Received: 29 January 2020 | Revised: 9 June 2020 | Accepted: 22 July 2020

DOI: 10.1002/nla.2333

RESEARCH ARTICLE

WILEY

Algebraic multigrid preconditioning of the Hessian in optimization constrained by a partial differential equation

Andrew T. Barker[1] | **Andrei Drăgănescu**[2]

[1]Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, California

[2]University of Maryland, Baltimore County, Baltimore, Maryland

Correspondence
Andrew T. Barker, Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA.
Email: atb@llnl.gov

Funding information
National Science Foundation, Grant/Award Number: DMS-1913201; U.S. Department of Energy, Grant/Award Numbers: DE-AC52-07NA27344, DE-SC0005455

Summary

We construct an algebraic multigrid (AMG) based preconditioner for the reduced Hessian of a linear-quadratic optimization problem constrained by an elliptic partial differential equation. While the preconditioner generalizes a geometric multigrid preconditioner introduced in earlier works, its construction relies entirely on a standard AMG infrastructure built for solving the forward elliptic equation, thus allowing for it to be implemented using a variety of AMG methods and standard packages. Our analysis establishes a clear connection between the quality of the preconditioner and the AMG method used. The proposed strategy has a broad and robust applicability to problems with unstructured grids, complex geometry, and varying coefficients. The method is implemented using the Hypre package and several numerical examples are presented.

KEYWORDS

algebraic multigrid, elliptic equations, finite element methods, PDE-constrained optimization

1 | INTRODUCTION

The purpose of this work is to construct and analyze algebraic multigrid (AMG) based preconditioners for the reduced Hessian for optimal control problems constrained by elliptic partial differential equations (PDEs). Due to the explosive development of computational resources over the last two decades, the computational science and engineering communities have shown an ever growing interest in solving large-scale optimization problems constrained by PDEs. Since PDEs require multiple input parameters for well-posedness (initial and boundary values, forcing terms, and various model parameters), PDE-constrained optimization can be employed to identify parameters that are optimal in some sense. Applications range from the optimization of manufacturing processes of various components[1] to data assimilation for weather prediction[2] and petroleum reservoir simulations.[3] A common question is whether in some cases an optimal control problem can be solved with the same high resolution as the forward model, ideally with a similar computational

cost. Relatedly, practitioners would like to use solvers and preconditioners developed and optimized for the forward problem also for the optimal control problem, as a way to limit programming effort and to exploit some knowledge and understanding of the model that is somehow embodied in the forward solver. In this article we show how AMG-based methods can be successfully used to solve very simple PDE-constrained optimization problems as a first step toward using such techniques more generally.

Efficient solvers for PDE-constrained optimization problems usually target the first-order optimality conditions, namely the Karush–Kuhn–Tucker (KKT) system. For linear-quadratic optimization problems the KKT is an indefinite linear system of two coupled PDEs: one is the original PDE (which represents the constraint) and the other is the adjoint equation, which is a PDE of essentially the same type as the original. For that reason, certain solver and preconditioning techniques that are appropriate for the PDE may be appropriate for the KKT system as well, while taking into account its indefinitedness. An alternative strategy is to solve reduced optimization problems obtained by removing the constraints from the original problem using the control-to-state map. For linear-quadratic problems this method results in a symmetric, positive system—the reduced Hessian—that usually has a dense matrix representation.

Regardless of whether one chooses to work with the full KKT system or the reduced system, multigrid methods have long been regarded among the most efficient for certain classes of PDEs, and their potential for PDE-constrained optimization is well established.[4] In particular, algebraic multigrid (AMG) methods have long been a well-developed strategy for solving the forward problem for challenging cases including unstructured grids, varying coefficients, and complicated geometries. The literature here is too large to even scratch the surface, but a reasonable starting point is the recent review.[5] However, the literature on AMG-based preconditioners for optimal control of PDEs is much sparser, and most, if not all of it, addresses the full KKT systems. For example, in Reference 6 a few V-cycles drawn from an AMG preconditioner are used to approximate the inverse of a stiffness matrix as part of a Schur-complement approximation. AMG takes center stage in very few papers[7,8] on optimal control, where the target is the entire KKT system as well.

In this article we focus on the reduced Hessian rather than the full KKT system. In this setting, an *unpreconditioned* conjugate gradient algorithm already can be shown to converge independently of the mesh size h (see Reference 9, ch. 7, also Reference 10). Even though this system is easy to precondition in some sense, each iteration is extremely expensive, requiring a forward and an adjoint solve of the underlying PDE. Since the absolute number of iterations can be large, and each iteration is expensive, in practice it may be desirable to have an efficient preconditioner to reduce the number of iterations.

Multilevel preconditioners for this problem setting have been discussed in References 10-16, among many other references, but to date there is a lack of robust practical implementations for cases when the forward problems, that is, the constraints, require an AMG treatment. However, reduced Hessians for optimal control problems constrained by PDEs are not good candidates for the direct application of AMG methods, primarily because they come in the form of dense matrices with no obvious sparse approximations, which is due to the fact that they represent integral operators that are nonlocal. The main novelty in this article is to show that, even though standard AMG methodology is not applicable, the AMG framework provides all the elements needed for preconditioning the reduced Hessian.

In particular, we implement an extension of the multilevel framework of Reference 10 that allows the use of an algebraic multigrid hierarchy in place of the geometric multigrid hierarchy. The preconditioner for the reduced Hessian can be constructed systematically based on the interpolation and restriction operators for the forward problem, which are readily available in most algebraic multigrid software implementations. The convergence theory depends on the underlying approximation properties of the multigrid hierarchy. In a standard algebraic multigrid context, these approximation properties may not allow for the same improvement with grid refinement that we see in the geometric multigrid setting, but the approach below is flexible and allows for a Hessian preconditioner for any multilevel discretization of the underlying forward problem. In particular, if an algebraic hierarchy with appropriate approximation properties is available for the forward problem, the preconditioning approach below will recover the appropriate convergence for the PDE-constrained optimization problem.

The numerical results show that the algebraic variant can be quite effective in reducing iteration counts even when the theory does not apply cleanly. Since algebraic preconditioners allow easier application to unstructured grids and problems with varying coefficients, the approach is practically useful in a wide variety of situations even apart from the convergence theory.

Our PDE-constrained optimization setting is introduced in Section 2, followed by a description of the practical algorithm in Section 3. A theoretical framework is developed in Section 4, followed by numerical results in Section 5. Some conclusions are formulated in Section 6.

2 | THE OPTIMIZATION PROBLEM

To fix ideas we focus on the standard distributed elliptic-constrained problem

$$\min_{y,u} J(y,u) = \frac{1}{2}\|y - y_d\|^2 + \frac{\beta}{2}\|u\|^2 \tag{1}$$

subject to

$$-\nabla \cdot (\kappa \nabla y) = u \text{ in } \Omega,\ y|_{\partial\Omega} = 0, \tag{2}$$

where $\Omega \subset \mathbb{R}^d$ ($d = 2,3$) is a sufficiently regular bounded domain, $y_d \in L^2(\Omega)$ is a desired state, and $\|\cdot\|$ is the $L^2(\Omega)$-norm. The coefficient $\kappa(x) \in \mathbb{R}$ is assumed to satisfy $0 < \kappa_0 \leq \kappa(x) \leq \kappa_1 < \infty$ for all $x \in \Omega$, for some positive constants κ_0, κ_1. We refer to u as the control and to y as the state. The problem of interest is a discrete counterpart of (1) and (2) resulted from discretizing (2) using a standard finite element formulation. To that end consider a finite element space $Y \subset H_0^1(\Omega)$ with basis $(\varphi_i)_{1 \leq i \leq m}$ and a discrete control space $U \subset L^2(\Omega)$ with basis $(\psi_i)_{1 \leq i \leq n}$. The standard discrete Galerkin formulation of (2) is:

$$\text{Find } y \in Y \text{ so that } a(y,v) = (u,v)\ \forall v \in Y, \tag{3}$$

where

$$a(y,v) = \int_\Omega \kappa \nabla y \cdot \nabla v\ \forall y, v \in H_0^1(\Omega), \tag{4}$$

and $(\cdot\,,\cdot)$ is the standard L^2-inner product. We introduce the solution operator $\mathcal{K} \in \mathfrak{L}(U,Y)$ defined by

$$\mathcal{K}u = y \text{ if } y \text{ satisfies (3).}$$

Since we prefer to distinguish between operators (acting on the finite element spaces) and their matrix representations, we will denote operators with caligraphic font and matrices using bold font. Furthermore, for a discrete function we use bold notation to denote the vector of coordinates with respect to the basis introduced in the space where the function resides. For example, if $y \in Y$, then $\mathbf{y} = [y_1, y_2, \ldots, y_m]^T \in \mathbb{R}^m$ is defined so that $y = \sum_{i=1}^m y_i \varphi_i$. The matrices needed for the discrete control problem are the *stiffness matrix* $[\mathbf{A}]_{ij} = a(\varphi_j, \varphi_i)$, the *state mass matrix* $[\mathbf{M}_y]_{ij} = (\varphi_j, \varphi_i)$, the *control mass matrix* $[\mathbf{M}_u]_{ij} = (\psi_j, \psi_i)$, and the *control-to-state mass matrix* $[\mathbf{M}_{yu}]_{i,j} = (\varphi_i, \psi_j)$. Note that both Y and U inherit the inner-product from $L^2(\Omega)$, and that $\|y\|^2 = \mathbf{y}^T \mathbf{M}_y \mathbf{y}$ for $y \in Y$, and $\|u\|^2 = \mathbf{u}^T \mathbf{M}_u \mathbf{u}$ for $u \in U$. The actual discrete problem to be solved is

$$\min_{y,u} J(y,u) = \frac{1}{2}\|y - y_d\|^2 + \frac{\beta}{2}\|u\|^2 = \frac{1}{2}(\mathbf{y} - \mathbf{y_d})^T \mathbf{M}_y (\mathbf{y} - \mathbf{y_d}) + \frac{\beta}{2}\mathbf{u}^T \mathbf{M}_u \mathbf{u} \tag{5}$$

subject to the constraint

$$\mathbf{A}\mathbf{y} = \mathbf{M}_{yu}\mathbf{u}. \tag{6}$$

We should remark that the problem (1) and (2) and its discretization (5) and (6) is among the most commonly studied in the PDE-constrained optimization literature;[17] therefore it is a natural example to showcase our method. However, the technique presented in this article can be directly applied to a variety of other linear-quadratic optimal control problems constrained by PDEs, such as boundary control of elliptic equations, initial value control and/or space-time distributed optimal control of parabolic equations, and so forth, as long as their discretizations can be expressed as (5) and (6). As with the geometric form of the multigrid algorithm, the performance of the presented method will vary from one model problem to another.

We reformulate the optimal control problem (5) and (6) as an unconstrained problem by defining a solution matrix for (6)

$$\mathbf{K} = \mathbf{A}^{-1}\mathbf{M}_{yu}, \tag{7}$$

which is precisely the matrix representation of the operator $\mathcal{K}$. Using $\mathbf{K}$ we eliminate the state y (or $\mathbf{y}$) in (5) and obtain the first-order optimality condition by setting the gradient of $\hat{J}(\mathbf{u}) = J(\mathbf{Ku}, \mathbf{u})$ to zero, followed by left-multiplying with $\mathbf{M}_u^{-1}$:

$$(\mathbf{M}_u^{-1}\mathbf{K}^T\mathbf{M}_y\mathbf{K} + \beta\mathbf{I})\mathbf{u} = \mathbf{M}_u^{-1}\mathbf{K}^T\mathbf{M}_y\mathbf{y_d}. \tag{8}$$

Equation (8) has a more simplified (and meaningful) form when using adjoint operators. By definition, $\mathcal{K}^*$ and its matrix $\mathbf{K}^*$ satisfy

$$\mathbf{y}^T\mathbf{M}_y\mathbf{Ku} = (\mathcal{K}u, y) = (u, \mathcal{K}^*y) = (\mathbf{K}^*\mathbf{y})^T\mathbf{M}_u\mathbf{u} = \mathbf{y}^T(\mathbf{K}^*)^T\mathbf{M}_u\mathbf{u} \quad \forall u \in U,\ y \in Y.$$

Hence, we obtain

$$\mathbf{K}^* = \mathbf{M}_u^{-1}\mathbf{K}^T\mathbf{M}_y = \mathbf{M}_u^{-1}\mathbf{M}_{yu}^T\mathbf{A}^{-T}\mathbf{M}_y.$$

This allows us to rewrite (8) in the familiar form

$$\mathbf{Gu} \overset{\text{def}}{=} (\mathbf{K}^*\mathbf{K} + \beta\mathbf{I})\mathbf{u} = \mathbf{K}^*\mathbf{y_d}. \tag{9}$$

The matrix $\mathbf{G}$ on the left side of (9) is the Hessian of the reduced cost function $\hat{J}$, usually referred to as the reduced Hessian. The goal of this article is to construct multilevel preconditioners for $\mathbf{G}$.

In general, $\mathbf{G}$ is dense and for large- and even medium-scale problems it would be extremely expensive, perhaps impossible, to form $\mathbf{G}$ explicitly. We can apply it (and even this operation is fairly expensive), but we cannot use its entries to construct an AMG preconditioner in the usual way. In what follows we show how to use the matrices required (and available) for building an AMG preconditioner for the stiffness matrix $\mathbf{A}$ in order to build a preconditioning algorithm for $\mathbf{G}$.

3 | PRECONDITIONING THE HESSIAN

Our construction follows closely the algorithm in Reference 10. We first present the construction of the two-level preconditioner which then leads us to the multilevel version.

3.1 | Two-level preconditioner

In the two-level setting, we define a coarse state space $Y_H \subset Y$ and a coarse control spaces $U_H \subset U$. To fix ideas we focus on a specific form of AMG, namely smoothed aggregation,[18,19] where the coarse basis functions for both state and controls are defined by prolongator matrices $\mathbf{S}$ and $\mathbf{P}$, respectively:

$$\Phi_k = \sum_{j=1}^{m}[\mathbf{S}]_{jk}\varphi_j,\ 1 \le k \le M, \quad \text{and} \quad \Psi_k = \sum_{j=1}^{n}[\mathbf{P}]_{jk}\psi_j,\ 1 \le k \le N.$$

The coarse state space Y_H is simply the span of $(\Phi_k)_{1\le k\le M}$, and the coarse control space U_H is the span of $(\Psi_k)_{1\le k\le N}$. Since $\Phi_k \in Y$ for $1 \le k \le M$, and $\Psi_k \in U$ $1 \le k \le N$ we indeed have that $Y_H \subset Y$,$U_H \subset U$. For future reference we remark that the prolongator matrices represent the embedding operators $\mathcal{S} \in \mathfrak{L}(Y_H, Y)$ and $\mathcal{P} \in \mathfrak{L}(U_H, U)$. The specific form of the prolongator is not relevant at this point.

At the coarse level we formulate the discrete PDE by replacing Y with Y_H in (3), thus obtaining a coarse stiffness matrix

$$[\mathbf{A}_H]_{kl} = a(\Phi_l, \Phi_k) = \sum_{i,j=1}^{m} a([\mathbf{S}]_{jl}\varphi_j, [\mathbf{S}]_{ik}\varphi_i) = \sum_{i,j=1}^{m}[\mathbf{S}^T]_{ki}[\mathbf{A}]_{ij}[\mathbf{S}]_{jl} = [\mathbf{S}^T\mathbf{AS}]_{kl}.$$

Therefore

$$\mathbf{A}_H = \mathbf{S}^T\mathbf{A}\mathbf{S}. \tag{10}$$

A similar calculation leads to the definition of the analogous coarse-level matrices

$$\mathbf{M}_{y,H} = \mathbf{S}^T\mathbf{M}_y\mathbf{S}, \quad \mathbf{M}_{u,H} = \mathbf{P}^T\mathbf{M}_u\mathbf{P}, \quad \mathbf{M}_{yu,H} = \mathbf{S}^T\mathbf{M}_{yu}\mathbf{P}. \tag{11}$$

Hence, a coarse version of the state equation (6) can now be written as

$$\mathbf{A}_H\mathbf{y}_H = \mathbf{M}_{yu,H}\mathbf{u}_H$$

and we can define a coarse solution matrix by

$$\mathbf{K}_H = (\mathbf{S}^T\mathbf{A}\mathbf{S})^{-1}(\mathbf{S}^T\mathbf{M}_{yu}\mathbf{P}) = (\mathbf{A}_H)^{-1}\mathbf{M}_{yu,H},$$

and a coarse Hessian

$$\mathbf{G}_H = \mathbf{K}_H^*\mathbf{K}_H + \beta\mathbf{I},$$

with $\mathbf{K}_H^* = \mathbf{M}_{u,H}^{-1}\mathbf{K}_H^T\mathbf{M}_{y,H}$.

To define the two-level preconditioner we need the L^2-projection $\Pi : U \to U_H$. Since Π is the adjoint of the embedding, the matrix representation of Π is

$$\mathbf{\Pi} = \mathbf{P}^* = \mathbf{M}_{u,H}^{-1}\mathbf{P}^T\mathbf{M}_u.$$

Note that in classical multigrid, and in particular in AMG, the usual restriction matrix is $\mathbf{P}^T$. One of the key features of our approach is to use $\mathbf{\Pi}$ instead of $\mathbf{P}^T$. Then the two-grid preconditioner is defined as in Reference [10, (4.1)] by

$$\mathbf{T} = \mathbf{P}(\mathbf{G}_H)\mathbf{\Pi} + \beta(\mathbf{I} - \mathbf{P}\mathbf{\Pi}). \tag{12}$$

Actually, in practice the inverse $\mathbf{T}^{-1}$ is needed. Since

$$(\mathbf{P}\mathbf{\Pi})^2 = \mathbf{P}(\mathbf{M}_{u,H}^{-1}\mathbf{P}^T\mathbf{M}_u)\mathbf{P}(\mathbf{M}_{u,H}^{-1}\mathbf{P}^T\mathbf{M}_u) = \mathbf{P}\mathbf{M}_{u,H}^{-1}\mathbf{M}_{u,H}\mathbf{M}_{u,H}^{-1}\mathbf{P}^T\mathbf{M}_u = \mathbf{P}\mathbf{\Pi},$$

it follows that $\mathbf{P}\mathbf{\Pi}$ is a projection matrix. Also note that $\mathbf{\Pi}\mathbf{P} = I$. Hence

$$\mathbf{T}^{-1} = \mathbf{P}(\mathbf{G}_H)^{-1}\mathbf{\Pi} + \beta^{-1}(\mathbf{I} - \mathbf{P}\mathbf{\Pi}), \tag{13}$$

since it can be easily verified that $\mathbf{T}^{-1}\mathbf{T} = I$. Notably, neither $\mathbf{T}$ nor $\mathbf{T}^{-1}$ is explicitly formed, but $\mathbf{T}^{-1}$ can be practically applied to a vector on the fine grid, provided the action of the inverse of the coarse Hessian $\mathbf{G}_H$ is available. Also note that, due to (11), $\mathbf{P}\mathbf{\Pi}$ is a projection matrix having as range the coarse space; applying this matrix only involves inverting the coarse control mass matrix. The definition (13) has an additive Schwarz structure, with a coarse grid correction and a kind of "smoother." Since the operator we are preconditioning is a "smoothing" operator (i.e., it involves the inverse of an elliptic operator), no real smoothing is required, just the above projection.

One note on symmetry: both the Hessian $\mathbf{G}$ and the preconditioner $\mathbf{T}$ are symmetric with respect to the L^2-induced inner product on $\mathbb{R}^n$, that is, $\mathbf{G} = \mathbf{G}^*$ and $\mathbf{T} = \mathbf{T}^*$. This is not the same as saying they are symmetric matrices, but rather $\mathbf{G} = \mathbf{M}_u^{-1}\mathbf{G}^T\mathbf{M}_u$ and $\mathbf{T} = \mathbf{M}_u^{-1}\mathbf{T}^T\mathbf{M}_u$. Hence special care has to be taken when using (preconditioned) conjugate gradient to solve the system (9), a matter that is further discussed in Section 5.2. In addition, it is shown in Reference 10 that $\mathbf{T}$ is positive definite, a property that is not automatically shared by the multigrid preconditioner introduced in Section 3.2.

3.2 | Multilevel preconditioner

The multilevel preconditioner is not a straightforward recursion of the two-level preconditioner. Indeed, a short calculation shows that a simple V-cycle recursion in (13) results in just a two-grid method using an even coarser grid, and does not yield the desired optimality result in the geometric multigrid setting.[10]

To define a multilevel method precisely, we need some additional notation. An ℓ-level preconditioner involves a hierarchy of state and control space (numbered in the AMG tradition from fine to coarse) $Y = Y_0 \supseteq Y_1 \supseteq \ldots \supseteq Y_{\ell-1}$ and $U = U_0 \supseteq U_1 \supseteq \ldots \supseteq U_{\ell-1}$, together with prolongation matrices $\mathbf{S}_j$ corresponding to the embedding operators $S_j \in \mathfrak{L}(Y_{j+1}, Y_j)$, and $\mathbf{P}_j$ associated with embedding operators $\mathcal{P}_j \in \mathfrak{L}(U_{j+1}, U_j)$. Then coarse stiffness matrices $\mathbf{A}_j$ and mass matrices $\mathbf{M}_{y,j}$,$\mathbf{M}_{u,j}$,$\mathbf{M}_{uy,j}$ can be defined as in (10) and (11). The construction of the hierarchies of subspaces for controls may not be related to those for states, as the state space and the control space may involve completely different physical domains. For example, the controls may be supported only on the boundary of the domain.

The final multilevel preconditioner will involve a hierarchy of matrices (and operators) $\mathbf{W}_j$ each approximating $\mathbf{G}_j^{-1}$. Hence, for the multilevel case, the preconditioner approximates the inverse of the matrix rather than the matrix itself. At the coarsest level, for simplicity, we assume

$$\mathbf{W}_{\ell-1} = \mathbf{G}_{\ell-1}^{-1} = (\mathbf{K}_{\ell-1}^{*}\mathbf{K}_{\ell-1} + \beta\mathbf{I})^{-1}.$$

The construction of $\mathbf{W}_{\ell-1}$ may be no trivial matter, since it not only involves inverting the Hessian, but also building $\mathbf{G}_{\ell-1}$ by computing dense matrix-matrix products. We will see below in Section 5 that in practice the coarse-grid problem may be approximated, but the theory here assumes an exact inverse on the coarsest level.

To define the intermediate level operators $\mathbf{W}_j$ for $0 < j < \ell - 1$, we begin by writing the two-level preconditioner (13) at an arbitrary level in a multilevel hierarchy,

$$\mathbf{T}_j^{-1} = \mathbf{P}_j\mathbf{W}_{j+1}\mathbf{\Pi}_j + \beta^{-1}(\mathbf{I} - \mathbf{P}_j\mathbf{\Pi}_j), \tag{14}$$

where $\mathbf{\Pi}_j = \mathbf{P}_j^{*}$ is the matrix representation of the L^2-projection $\Pi_j \in \mathfrak{L}(U_j, U_{j+1})$. The actual preconditioner is the first Newton iterate for the matrix equation

$$\mathbf{X}^{-1} - \mathbf{G}_j = \mathbf{0}$$

with initial guess $\mathbf{T}_j^{-1}$, which, cf. remark 3.11 in Reference 10, takes the form

$$\mathbf{W}_j = 2\mathbf{T}_j^{-1} - \mathbf{T}_j^{-1}\mathbf{G}_j\mathbf{T}_j^{-1}. \tag{15}$$

At the finest level—and this is the actual multilevel preconditioner for $\mathbf{G}_0$—we define

$$\mathbf{W}_0 = \mathbf{T}_0^{-1}. \tag{16}$$

Naturally, none of the operators $\mathbf{W}_j$, $0 \le j < \ell - 1$ should be built. Instead, the action of $\mathbf{u}_j \leftarrow \mathbf{W}_j\mathbf{b}_j$ can be easily implemented following[10] Algorithm MLAS:

1. if $j = \ell - 1$
2. $\mathbf{u}_j \leftarrow \mathbf{G}_j^{-1}\mathbf{b}_j$
3. else
4. $\mathbf{u}_j \leftarrow \mathbf{T}_j^{-1}\mathbf{b}_j$.
5. if $j > 0$
6. $\mathbf{u}_j \leftarrow \mathbf{u}_j + \mathbf{T}_j^{-1}(\mathbf{b}_j - \mathbf{G}_j\mathbf{u}_j)$
7. end if
8. end if

Since the action of $\mathbf{T}_j^{-1}$ requires the action of $\mathbf{W}_{j+1}$, the algorithm above has a W-cycle structure, due to the two calls to $\mathbf{T}_j^{-1}$ in lines 4 and 6 for intermediate levels (when $0 < j < \ell - 1$). Also at intermediate levels, the action of the Hessian $\mathbf{G}_j$ is required (line 6), and usually this is the most cost-intensive component.

It should be noted here that the number of levels involved in the multilevel preconditioner is usually not as large as the number of levels that are in principle available from the AMG infrastructure for solving the forward problem. There is no guarantee that the multilevel preconditioner remains positive definite, except for special circumstances (although the two-level one always is). However, in the presence of an aggressive coarsening strategy and three spatial dimensions, in practice three or four levels may often suffice to achieve a significant speedup over unpreconditioned CG.

4 | ANALYSIS

So far we have shown that the definition of a multilevel preconditioner for the Hessian extends naturally from the geometric to the AMG context. The analysis follows suit to some degree, though certain details depend on the particular problem and properties of the coarsening strategy. In this section we prefer to focus on the operators defined in Section 3.2 rather than their associated matrices. Recall that the Hessian operator on U_j is given by

$$\mathcal{G}_j = \mathcal{K}_j^* \mathcal{K}_j + \beta I \in \mathfrak{L}(U_j), \tag{17}$$

whose matrix representation is $\mathbf{G}_j$. The goal is to estimate the spectral distance between the inverse of the finest-level hessian $\mathcal{G}_0^{-1}$ and the multilevel preconditioner $\mathcal{W}_0$ corresponding to the matrix-based definition in Section 3.2. The main result is Theorem 2.

The spectral distance between two symmetric positive definite operators $\mathcal{X}$ and $\mathcal{Y}$ on a Hilbert space U is defined in Reference 10 as

$$d_\sigma(\mathcal{X}, \mathcal{Y}) = \sup_{u \in U \setminus \{0\}} |\ln(\mathcal{X}u, u) - \ln(\mathcal{Y}u, u)|, \tag{18}$$

and is shown to be a good quality measure for the convergence of preconditioned iterative methods. In particular, for two symmetric positive definite operators $\mathcal{X}, \mathcal{Y}$, we have

$$\ln \operatorname{cond}(\mathcal{Y}^{-1}\mathcal{X}) \leq d_\sigma(\mathcal{X}, \mathcal{Y}).$$

In our context, where the construction of the preconditioner $\mathcal{W}_0$ changes with the number of levels ℓ, if $d_\sigma(\mathcal{G}_0^{-1}, \mathcal{W}_0)$ is bounded with respect to ℓ, then the w iterations is also bounded with respect to ℓ.

For $\mathcal{X}, \mathcal{Y} \in \mathfrak{L}(U)$ denote $\mathcal{N}_\mathcal{X}(\mathcal{Y}) = 2\mathcal{Y} - \mathcal{Y}\mathcal{X}\mathcal{Y}$, and let $\mathcal{E}_j : \mathfrak{L}(U_{j+1}) \to \mathfrak{L}(U_j)$ be given by

$$\mathcal{E}_j(\mathcal{X}) = \mathcal{P}_j \mathcal{X} \Pi_j + \beta^{-1}(I - \mathcal{P}_j \Pi_j).$$

The multilevel operator whose matrix representation is $\mathbf{W}_j$ can be defined recursively as

$$\mathcal{W}_{\ell-1} = \mathcal{G}_{\ell-1}^{-1}, \tag{19}$$

$$\mathcal{W}_j = \mathcal{N}_{\mathcal{G}_j}(\mathcal{E}_j(\mathcal{W}_{j+1})), \quad j = 1, \ldots, \ell - 2, \tag{20}$$

$$\mathcal{W}_0 = \mathcal{E}_0(\mathcal{W}_1). \tag{21}$$

Denote by $\mathfrak{L}_+(U)$ the set of symmetric positive definite operators on the Hilbert space U. We recall from Reference 10 a set of basic facts about the spectral distance.

Theorem 1. *The function d_σ is a distance on the set of symmetric positive operators and satisfies the following.*

(a) If $\mathcal{X}, \mathcal{Y} \in \mathfrak{L}_+(U)$, then

$$d_\sigma(\mathcal{X}, \mathcal{Y}) = d_\sigma(\mathcal{X}^{-1}, \mathcal{Y}^{-1}). \tag{22}$$

(b) If $\mathcal{X}, \mathcal{G} \in \mathfrak{L}_+(U)$ *and* $d_\sigma(\mathcal{X}, \mathcal{G}^{-1}) < 0.4$*, then*

$$d_\sigma(\mathcal{N}_{\mathcal{G}}(\mathcal{X}), \mathcal{G}^{-1}) \leq 2d_\sigma(\mathcal{X}, \mathcal{G}^{-1})^2. \tag{23}$$

(c) If $\mathcal{X}, \mathcal{Y} \in \mathfrak{L}_+(U_{j+1})$*, then* $\mathcal{E}_j(\mathcal{X}), \mathcal{E}_j(\mathcal{Y}) \in \mathfrak{L}_+(U_j)$ *and*

$$d_\sigma(\mathcal{E}_j(\mathcal{X}), \mathcal{E}_j(\mathcal{Y})) \leq d_\sigma(\mathcal{X}, \mathcal{Y}). \tag{24}$$

To be precise, part (a) appears as lemma 3.10 in Reference 10, part (c) is lemma 5.1 in Reference 10, while part (b) is theorem 3.12 combined with the beginning the proof of theorem 5.4 in Reference 10.

For the analysis of our multilevel preconditioner we introduce the two-level operator

$$\mathcal{V}_j = \mathcal{P}_j \mathcal{G}_{j+1} \Pi_j + \beta(I - \mathcal{P}_j \Pi_j) \tag{25}$$

whose inverse is

$$\mathcal{V}_j^{-1} = \mathcal{P}_j \mathcal{G}_{j+1}^{-1} \Pi_j + \beta^{-1}(I - \mathcal{P}_j \Pi_j) = \mathcal{E}_j(\mathcal{G}_{j+1}^{-1}). \tag{26}$$

We denote

$$\|\mathcal{K}_j - \mathcal{S}_j \mathcal{K}_{j+1} \Pi_j\| = a_j,$$

where for an operator $\mathcal{L} \in \mathfrak{L}(U, V)$ with $U, V \subseteq L^2(\Omega)$

$$\|\mathcal{L}\| = \sup_{u \in U \setminus \{0\}} \frac{\|\mathcal{L}u\|}{\|u\|}.$$

Assumption 1. We assume that the solution operators satisfy the following stability and approximation conditions:

There exists a level-independent constant $C_{\mathcal{K}} > 0$ so that

$$\|\mathcal{K}_j\| \leq C_{\mathcal{K}}, \quad j = 0, 1, \ldots, \ell - 1 \tag{27}$$

$$4C_{\mathcal{K}} a_j \leq \beta, \quad j = 0, 1, \ldots, \ell - 1. \tag{28}$$

The inequality (28) in Assumption 1 clearly poses a lower-bound restriction on β which is problem dependent. Also note that in some contexts, as is the case for hierarchies based on geometric mesh refinement, the quantities a_j are naturally ordered in the sense that $a_0 \leq a_1 \leq \ldots \leq a_{\ell-1}$; in these cases (28) effectively poses a restriction on the coarsest grid used in association to a certain β.

Lemma 1. *If Assumption 1 holds and* $b_j = 4\beta^{-1} C_{\mathcal{K}} a_j$*, then*

$$\|\mathcal{G}_j - \mathcal{V}_j\| \leq 2C_{\mathcal{K}} a_j, \quad j = 0, 1, \ldots, \ell - 2, \tag{29}$$

$$d_\sigma(\mathcal{G}_j, \mathcal{V}_j) \leq b_j, \quad j = 0, 1, \ldots, \ell - 2. \tag{30}$$

Proof. For $u \in U_j$ we have

$$\begin{aligned}
&|((\mathcal{G}_j - \mathcal{V}_j)u, u)| \\
&\quad = |((\mathcal{K}_j^* \mathcal{K}_j - \mathcal{P}_j \mathcal{K}_{j+1}^* \mathcal{K}_{j+1} \Pi_j)u, u)| = |\|\mathcal{K}_j u\|^2 - \|\mathcal{K}_{j+1} \Pi_j u\|^2| \\
&\quad = |\|\mathcal{K}_j u\| - \|\mathcal{K}_{j+1} \Pi_j u\|| \cdot (\|\mathcal{K}_j u\| + \|\mathcal{K}_{j+1} \Pi_j u\|) \\
&\quad \leq 2C_{\mathcal{K}} \|\mathcal{K}_j u - \mathcal{S}_j \mathcal{K}_{j+1} \Pi_j u\| \cdot \|u\| \leq 2C_{\mathcal{K}} a_j \|u\|^2,
\end{aligned}$$

where we used that S_j is an embedding and $\|\Pi_j u\| \leq \|u\|$. The conclusion (29) follows from the symmetry of the operator $(\mathcal{G}_j - \mathcal{V}_j)$. Note that

$$|\ln x| \leq 2|x-1|, \quad \text{if } |x-1| \leq \frac{1}{2}. \tag{31}$$

If $u \neq 0$ then the assumption $4C_{\mathcal{K}} a_j \leq \beta$ implies

$$\left| \frac{(\mathcal{V}_j u, u)}{(\mathcal{G}_j u, u)} - 1 \right| = \left| \frac{((\mathcal{V}_j - \mathcal{G}_j)u, u)}{(\mathcal{G}_j u, u)} \right| \leq \beta^{-1} \|\mathcal{V}_j - \mathcal{G}_j\| \leq \frac{1}{2},$$

since $(\mathcal{G}_j u, u) \geq \beta(u, u)$. Using (31) we obtain

$$\left| \ln \frac{(\mathcal{V}_j u, u)}{(\mathcal{G}_j u, u)} \right| \leq 2 \left| \frac{(\mathcal{V}_j u, u)}{(\mathcal{G}_j u, u)} - 1 \right| \leq 2\beta^{-1} \|\mathcal{V}_j - \mathcal{G}_j\| \leq 4\beta^{-1} C_{\mathcal{K}} a_j. \tag{32}$$

The conclusion (30) follows by taking the max in (32) over $u \in U_j \setminus \{0\}$. ∎

We should point out that in the context of geometric multigrid both state and control spaces are constructed using classical finite elements corresponding to a sequence of meshes $\mathcal{T}_{h_j}$; if the finer grids are obtained by, say, uniform mesh-refinement, we have a sequence of mesh sizes $h_j = h_0 2^j$ (h_0 corresponds to the finest space, $h_{\ell-1}$ to the coarsest). Under standard assumptions on the elliptic equation (2), such as quasiuniformity of the meshes and full elliptic regularity, and by using continuous piecewise linear elements, it is known that the following approximation holds: there exists a constant $C_{mg} > 0$ so that

$$a_j \leq C_{mg} h_j^2. \tag{33}$$

This follows from the standard finite element a priori estimate[20]

$$\|(\mathcal{K}_j - \mathcal{K})u\| \leq C h_j^2 \|u\|, \quad \forall u \in L^2(\Omega). \tag{34}$$

If the domain is not convex and full elliptic regularity does not hold, then the power of h_j in (34) may be diminished. However, if (33) holds, we expect that, at least asymptotically

$$a_j \approx a_{j+1}/4, \quad j = 0, 1, \ldots, \ell - 3. \tag{35}$$

An approximation property as in assumption (28) where the norm decreases like h/H is referred to as a "strong approximation property" in the algebraic multigrid literature, and generally speaking most AMG algorithms do not provide such a property, as weaker approximation is all that is necessary for two-grid convergence. See the discussion in Reference 21, and for an example of an AMG method with strong approximation property see Reference 22.

Hence we conduct the multigrid analysis both for the case when the approximation properties improve with resolution, as in the geometric multigrid case, or simply stay bounded. The precise assumption on a_j (or rather b_j) will be made clear in Theorem 2. We begin with a few technical results. The following lemma is closely related to lemma 5.3 from Reference 10; for completeness and consistency of notation we prefer to give the short technical proof.

Lemma 2. *Under the hypotheses and notation of Lemma 1, let* $d_j = d_\sigma(\mathcal{W}_j, \mathcal{G}_j^{-1})$. *If* $b_j \leq 0.1$ *for* $j = 0,1, \ldots, \ell - 1$, *then* $d_j < 0.2$ *for* $j = 1, \ldots, \ell - 1$, *and the following recursion holds:*

$$d_j \leq 2(d_{j+1} + b_j)^2, \quad j = 0, 1, \ldots, \ell - 2, \tag{36}$$

$$d_0 \leq d_1 + b_0. \tag{37}$$

Proof. The proof proceeds by induction starting from the coarsest level, where we have $d_{\ell-1}=0$. Assume for some $j<\ell-2$ that $d_{j+1}<0.2$. Then by (24)

$$d_\sigma(\mathcal{E}_j(\mathcal{W}_{j+1}),\quad \mathcal{E}_j(\mathcal{G}_{j+1}^{-1}))\le d_{j+1}<0.2.$$

If $j\ge 1$ then

$$\begin{aligned}
d_j &= d_\sigma(\mathcal{W}_j,\mathcal{G}_j^{-1}) = d_\sigma(\mathcal{N}_{\mathcal{G}_j}(\mathcal{E}_j(\mathcal{W}_{j+1})),\mathcal{G}_j^{-1}) \overset{(23)}{\le} 2(d_\sigma(\mathcal{E}_j(\mathcal{W}_{j+1}),\mathcal{G}_j^{-1}))^2\\
&\le 2(d_\sigma(\mathcal{E}_j(\mathcal{W}_{j+1}),\mathcal{V}_j^{-1})+d_\sigma(\mathcal{V}_j^{-1},\mathcal{G}_j^{-1}))^2\\
&\overset{(22)}{=} 2(d_\sigma(\mathcal{E}_j(\mathcal{W}_{j+1}),\mathcal{E}_j(\mathcal{G}_{j+1}^{-1}))+d_\sigma(\mathcal{V}_j,\mathcal{G}_j))^2\\
&\overset{(24)}{\le} 2(d_\sigma(\mathcal{W}_{j+1},\mathcal{G}_{j+1}^{-1})+d_\sigma(\mathcal{V}_j,\mathcal{G}_j))^2 \overset{(30)}{\le} 2(d_{j+1}+b_j)^2<2\cdot 0.3^2=0.18<0.2.
\end{aligned}$$

Finally for $j=0$

$$\begin{aligned}
d_0 &= d_\sigma(\mathcal{W}_0,\mathcal{G}_0^{-1}) = d_\sigma(\mathcal{E}_0(\mathcal{W}_1),\mathcal{G}_0^{-1})\le d_\sigma(\mathcal{E}_0(\mathcal{W}_1),\mathcal{V}_0^{-1})+d_\sigma(\mathcal{V}_0^{-1},\mathcal{G}_0^{-1})\\
&\overset{(22)}{=} d_\sigma(\mathcal{E}_0(\mathcal{W}_1),\mathcal{E}_0(\mathcal{G}_1^{-1}))+d_\sigma(\mathcal{V}_0,\mathcal{G}_0)\overset{(24)}{\le} d_\sigma(\mathcal{W}_1,\mathcal{G}_1^{-1})+d_\sigma(\mathcal{V}_0,\mathcal{G}_0)\overset{(30)}{\le} d_1+b_0,
\end{aligned}$$

which completes the proof. ∎

Lemma 3. *If* $0<\alpha\le 1/8$ *then the inequality*

$$2(\alpha x+1)^2\le x \tag{38}$$

is satisfied by $x_{opt}=(1-4\alpha)/(4\alpha^2)$, *and* $x_{opt}>0$.

Proof. The inequality (38) is equivalent to

$$q(x)=2\alpha^2x^2+(4\alpha-1)x+2=2(\alpha x+1)^2-x\le 0.$$

The quadratic q above has a minimum at $x_{opt}=(1-4\alpha)/(4\alpha^2)$, and the minimal value is

$$q_{\min}=-\frac{(4\alpha-1)^2-16\alpha^2}{8\alpha^2}=\frac{8\alpha-1}{8\alpha^2}.$$

Clearly $q_{\min}\le 0$ and $x_{opt}>0$ if $\alpha\le 1/8$. ∎

Theorem 2. *If Assumption 1 holds and* $b_j=4\beta^{-1}C_{\mathcal{K}}a_j$ *satisfies*

$$b_j\le \mathfrak{K}\mathfrak{f}^{\ell-1-j},\quad j=0,1,\ldots,\ell-1. \tag{39}$$

with $0<\mathfrak{f}\le 1$ *and* $\mathfrak{K}\le\min(0.1,\mathfrak{f}/8)$ *then*

$$d_\sigma(\mathcal{W}_0,\mathcal{G}_0^{-1})\le\left(\frac{1}{4}+\mathfrak{K}\right)\mathfrak{f}^{\ell-1}. \tag{40}$$

Proof. Let $d_j=d_\sigma(\mathcal{W}_j,\mathcal{G}_j^{-1})$. We are looking for $\mathfrak{C}>0$ so that

$$d_j\le\mathfrak{C}(\mathfrak{K}\mathfrak{f}^{\ell-1-j})^2,\quad j=1,2,\ldots,\ell-1. \tag{41}$$

We perform an inductive argument from $j=\ell-1$ down to $j=1$. The estimate for the case $j=0$ will then follow.

Since $d_{\ell-1}=0$, the case $j=\ell-1$ is trivial. Assume that (41) holds for $(j+1)$ with some $1\le j\le \ell-2$. Using (36) and $\mathfrak{f}\le 1$ we obtain

$$\begin{aligned} d_j &\le 2(d_{j+1}+b_j)^2 \le 2(\mathfrak{C}(\mathfrak{K}\mathfrak{f}^{\ell-1-(j+1)})^2+\mathfrak{K}\mathfrak{f}^{\ell-1-j})^2 = 2(\mathfrak{C}\mathfrak{K}^2\mathfrak{f}^{2(\ell-j-2)}+\mathfrak{K}\mathfrak{f}^{\ell-1-j})^2\\ &= 2(\mathfrak{C}\mathfrak{K}\mathfrak{f}^{\ell-j-3}+1)^2(\mathfrak{K}\mathfrak{f}^{\ell-1-j})^2 \le 2(\mathfrak{C}\mathfrak{K}\mathfrak{f}^{-1}+1)^2(\mathfrak{K}\mathfrak{f}^{\ell-1-j})^2. \end{aligned}$$

We apply (38) with $\alpha=\mathfrak{K}\mathfrak{f}^{-1}$ and $x=\mathfrak{C}$ to conclude that for

$$\mathfrak{C}=x_{opt}=\frac{1-4\alpha}{4\alpha^2}=\frac{1-4\mathfrak{K}\mathfrak{f}^{-1}}{4(\mathfrak{K}\mathfrak{f}^{-1})^2}$$

we have

$$d_j\le \mathfrak{C}(\mathfrak{K}\mathfrak{f}^{\ell-1-j})^2,$$

which concludes the proof by induction. Hence, for $j=1,\ \dots\ ,\ell-1$

$$d_j\le \frac{1-4\mathfrak{K}\mathfrak{f}^{-1}}{4(\mathfrak{K}\mathfrak{f}^{-1})^2}(\mathfrak{K}\mathfrak{f}^{\ell-1-j})^2<\frac{1}{4}\mathfrak{f}^{2(\ell-j)}.$$

Finally, by (37)

$$d_0\le d_1+b_0<\frac{1}{4}\mathfrak{f}^{2(\ell-1)}+\mathfrak{K}\mathfrak{f}^{\ell-1}\le\left(\frac{1}{4}\mathfrak{f}^{\ell-1}+\mathfrak{K}\right)\mathfrak{f}^{\ell-1}\le\left(\frac{1}{4}+\mathfrak{K}\right)\mathfrak{f}^{\ell-1}.$$

Hence (40) holds. ∎

A few remarks are in order. First, the hypothesis (39) on b_j can be written as

$$a_j\le\beta\frac{\mathfrak{K}}{4C_{\mathcal{K}}}\mathfrak{f}^{\ell-1-j},\quad j=0,1,\ \dots\ ,\ell-1. \tag{42}$$

The latter form of the assumption is consistent with the approximation properties of the coarser spaces for the case when they represent a geometric multigrid hierarchy. For example, the estimates (34) and (35) are consistent with $\mathfrak{f}=1/4$. It is conceivable that for some cases AMG hierarchies of spaces could also satisfy (42) with $\mathfrak{f}<1$, meaning that the finer spaces have better approximation properties than the coarser spaces. The case $\mathfrak{f}=1$ was not addressed in the earlier works on geometric multigrid, and translates into saying that there is a uniform upper bound for the two-level approximation as expressed in (42). In the absence of sufficient theoretical results in the AMG literature regarding the successive two-grid approximations, we conducted numerical experiments in Section 5.1 to verify the relative behavior of the sequence a_j for a few cases of interest.

The second remark refers to the optimality of the result in Theorem 2. In case $\mathfrak{f}<1$, the Theorem shows that $d_\sigma(\mathcal{W}_0,\mathcal{G}_0^{-1})$ decreases at the same rate as b_0, albeit with a larger constant in front. Therefore the quality of the preconditioners increases with increasing resolution, resulting in a decreasing number of preconditioned CG iterations, as in the case of geometric multigrid.[10] However, if $\mathfrak{f}=1$, then we have a uniform bound of the spectral distance independent of the number of levels, which results in a mesh-independent number of iterations.

We also note the role of the regularization parameter β in the multilevel convergence estimates. As might be expected, a small β implies that better approximation is required from the coarse spaces, and in general the problem is harder to precondition if β is small.

Finally, the numbers a_j can be computed numerically based on matrix norms, since the translation between operator and matrix norms is straightforward. If $\mathcal{L}\in\mathfrak{L}(U,Y)$ has matrix representation $\mathbf{L}\in\mathbb{R}^{m\times n}$, then

$$\begin{aligned} \|\mathcal{L}\| &= \max_{u\in U}\frac{\|\mathcal{L}u\|}{\|u\|}=\max_{\mathbf{u}\in\mathbb{R}^n}\frac{((\mathbf{L}\mathbf{u})^T\mathbf{M}_y\mathbf{L}\mathbf{u})^{\frac{1}{2}}}{(\mathbf{u}^T\mathbf{M}_u\mathbf{u})^{\frac{1}{2}}}=\max_{\mathbf{v}\in\mathbb{R}^n}\frac{((\mathbf{M}_y^{\frac{1}{2}}\mathbf{L}\mathbf{M}_u^{-\frac{1}{2}}\mathbf{v})^T\mathbf{M}_y^{\frac{1}{2}}\mathbf{L}\mathbf{M}_u^{-\frac{1}{2}}\mathbf{v})^{\frac{1}{2}}}{(\mathbf{v}^T\mathbf{v})^{\frac{1}{2}}}\\ &= \|\mathbf{M}_y^{\frac{1}{2}}\mathbf{L}\mathbf{M}_u^{-\frac{1}{2}}\|_2, \end{aligned}$$

TABLE 1 Direct computation of a_j coefficient from (43) and $\mathfrak{f}$ in (42) for geometric and algebraic multigrid coarsening of a uniform quadrilateral mesh

	Geometric			AMG (aggressive)			AMG (no aggressive)		
Level j	a_j	$\tilde{a}_j$	$\mathfrak{f}$	a_j	$\tilde{a}_j$	$\mathfrak{f}$	a_j	$\tilde{a}_j$	$\mathfrak{f}$
0	3.24e−4	3.14e−4	—	2.63e−2	2.63e−2	—	3.22e−3	3.22e−3	—
1	1.29e−3	1.25e−3	0.25	3.86e−3	3.86e−3	6.81	2.20e−3	2.20e−3	1.46
2	4.95e−3	4.80e−3	0.26	6.82e−3	6.79e−3	0.57	7.03e−3	6.95e−3	0.32
3	1.71e−2	1.63e−2	0.29				1.60e−2	1.60e−2	0.43

Note: In this setting we can show numerically that (44) approximates (43) quite well.

where we substituted $\mathbf{u} = \mathbf{M}_u^{-\frac{1}{2}}\mathbf{v}$, and $\|\cdot\|_2$ is the matrix 2-norm. Hence, for prototype AMG methods, one can compute the numbers a_j to identify their behavior numerically by using the formula

$$a_j = \|\mathbf{M}_y^{\frac{1}{2}}(\mathbf{K}_j - \mathbf{S}_j\mathbf{K}_{j+1}\mathbf{\Pi}_j)\mathbf{M}_u^{-\frac{1}{2}}\|_2. \tag{43}$$

This formula is used in Section 5.1 to assess the behavior of a_j for standard AMG methods.

5 | IMPLEMENTATION AND NUMERICAL RESULTS

In this section we show numerical experiments to accompany our analysis (Section 5.1), followed in Section 5.2 by a discussion on implementation. In Section 5.3 we show numerical results for a constant coefficient case, and in Section 5.4 we compare our results with the geometric multigrid version of this method. In Sections 5.5 and 5.6 we show results for problems with complex geometries and varying coefficients, respectively.

The AMG package we use comes from Hypre,[23,24] specifically the BoomerAMG preconditioner.[25] Hypre is a well-established and widely used open source algebraic multigrid package with an emphasis on parallel performance and scalability, but our approach does not depend on any particular characteristics or algorithms in Hypre. We use the open source finite element package MFEM for our finite element discretization.[26]

5.1 | Numerical estimates of (43)

Since $\mathbf{M}_y$ and $\mathbf{M}_u$ can be thought of as discretizations of an identity operator, and in a finite element context these matrices have a spectrum that is bounded independently of the mesh size, it makes sense to approximate (43) by

$$\tilde{a}_j = \|\mathbf{K}_j - \mathbf{S}_j\mathbf{K}_{j+1}\mathbf{\Pi}_j\|, \tag{44}$$

which can be estimated effectively by approximating the dominant eigenvalue with the power method. Tests on small matrices in 2D, where explicit matrix square roots and norms are feasible, show that $\tilde{a}_j$ is quite a good approximation for a_j, as shown in Table 1. The problem here is based on the constraint (2) with $\kappa = 1$ on the domain $\Omega = [0, 1]^2$ discretized with a uniform mesh of first-order Lagrange quadrilateral elements. Except for boundary conditions, the same discretization is used for state and control spaces.

In Table 2, we use the estimate (44) for a uniform mesh of hexahedra on $\Omega = [0, 1]^3$, again with $\kappa = 1$ (the same setting used below in Section 5.4). In Tables 1 and 2, we see that the geometric multigrid has ratios of $\mathfrak{f} \approx \tilde{a}_j/\tilde{a}_{j+1}$ of about 1/4, as expected.

For algebraic multigrid, this ratio is generally below one with a significant technical exception. The AMG package we use, Hypre, by default performs one level of "aggressive coarsening" for efficiency reasons, which simply means that the second finest mesh is very much coarser than the finest mesh. This choice appears clearly in Tables 1 and 2, where the first ratio $\tilde{a}_0/\tilde{a}_1$ is quite large when this option is enabled. One emphasis of this work is its applicability to standard AMG

TABLE 2 Numerical estimates of the $\tilde{a}_j$ coefficient from (44) and $\mathfrak{f} \approx \tilde{a}_j/\tilde{a}_{j+1}$ in (42) for geometric and algebraic multigrid coarsening of a uniform hexahedral mesh

	Geometric		AMG (aggressive)		AMG (no aggressive)	
Level j	$\tilde{a}_j$	$\tilde{a}_j/\tilde{a}_{j+1}$	$\tilde{a}_j$	$\tilde{a}_j/\tilde{a}_{j+1}$	$\tilde{a}_j$	$\tilde{a}_j/\tilde{a}_{j+1}$
0	3.13e−4	—	1.65e−2	—	5.97e−4	—
1	1.23e−3	0.25	2.16e−3	7.64	1.65e−3	0.36
2	4.44e−3	0.28	3.58e−3	0.60	3.27e−3	0.50
3	1.37e−2	0.32	6.92e−3	0.52	5.43e−3	0.60

packages with minimal programmer effort, so in all of our other numerical results we use AMG in its default configuration with aggressive coarsening, even though this may not be the best theoretical choice in view of Theorem 2.

5.2 | Algorithm implementation

In practice we solve the problem

$$(\mathbf{K}^T\mathbf{M}_y\mathbf{K} + \beta\mathbf{M}_u)\mathbf{u} = \mathbf{K}^T\mathbf{M}_y\mathbf{y_d} \tag{45}$$

rather than (8). Note that the former can be obtained by multiplying the latter by $\mathbf{M}_u$ from the left. Then our actual implementation preconditions the operator $\mathbf{M}_u\mathbf{G}$. If (13) is a good preconditioner for $\mathbf{G}$, then $\mathbf{T}^{-1}\mathbf{M}_u^{-1}$ is a good preconditioner for $\mathbf{M}_u\mathbf{G}$. With some substitutions we can write

$$\mathbf{T}^{-1}\mathbf{M}_u^{-1} = \mathbf{P}(\mathbf{M}_{u,H}\mathbf{G}_H)^{-1}\mathbf{P}^T + \beta^{-1}(\mathbf{M}_u^{-1} - \mathbf{P}\mathbf{M}_{u,H}^{-1}\mathbf{P}^T), \tag{46}$$

with analogous modifications for the multilevel preconditioner.

Our algorithm requires the application of $\mathbf{\Pi}$ and therefore the inversion of mass matrices. We use conjugate gradient preconditioned with a symmetric Gauss–Seidel sweep to solve mass matrix problems, with a relative residual tolerance of 10^{-8}. Our emphasis in what follows is showing that the proposed algorithm works and is practical, not on tuning of parameters for maximum efficiency.

Similarly, whenever we apply the operators $\mathbf{K}$ or $\mathbf{K}^*$, we use conjugate gradient preconditioned with Hypre BoomerAMG to invert $\mathbf{A}$, solving to a relative residual tolerance of 10^{-8}. The coarsest optimization solve $\mathbf{G}_{\ell-1}^{-1}$ is done with unpreconditioned conjugate gradient and a relative residual tolerance of 10^{-4}. We compare the multilevel preconditioning technique with completely unpreconditioned CG, that is, solving (45) using the operator on the left-hand side just as it is. Even when the optimization solve is unpreconditioned, the same inner solver is used, that is, the inner solves in $\mathbf{K}, \mathbf{K}^*$ always use an algebraic multigrid method. As a result, the AMG setup cost for our preconditioner is comparable to that for the unpreconditioned case. We note that our implementation is fully parallel, though the emphasis in this paper is not on parallel performance or scalability.

5.3 | Constant coefficient elliptic constraint

The next set of experiments discretize (2) with $\kappa = 1$ on the domain $\Omega = [0, 1]^3$ and with homogeneous Dirichlet conditions on all of $\partial\Omega$. The mesh is based on a 474-element unstructured tetrahedral mesh that is refined uniformly (see Figure 1), and we use first-order Lagrange elements. The desired state is set as

$$y_d = \left(\frac{1}{3\pi^2} + 3\pi^2\beta\right)\sin(\pi x)\sin(\pi y)\sin(\pi z) \tag{47}$$

which results in a closed-form solution to the optimal control problem, see Figure 1. In these examples all the solution methods approximate the true solution with the expected order of accuracy, and in particular our preconditioner does not change the solution compared with unpreconditioned conjugate gradient. We solve the Hessian problem (8) with the conjugate gradient method, stopping when the relative residual is less than 10^{-8}, and compare use of the multilevel

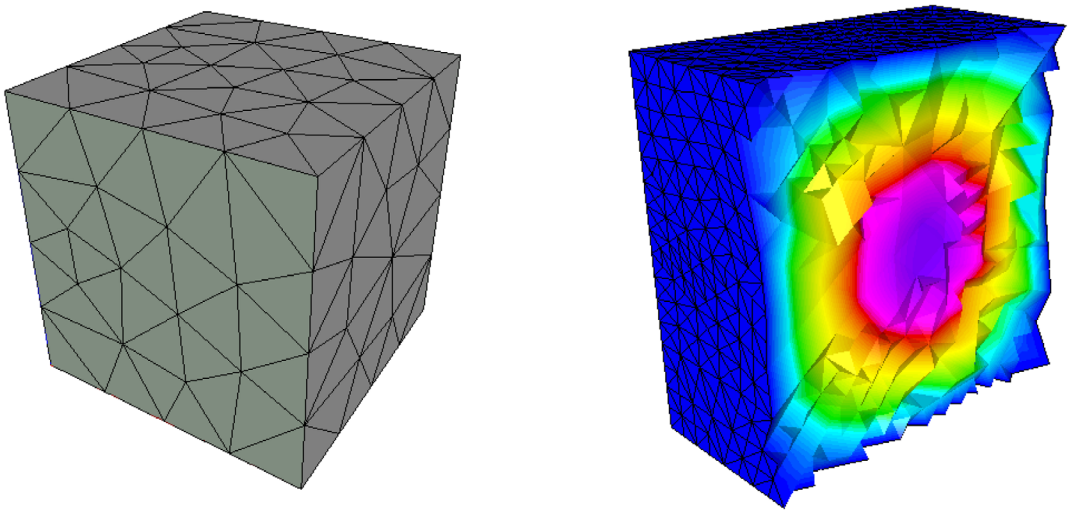

FIGURE 1 An unstructured tetrahedral mesh used for numerical experiments (left) and a typical achieved optimal control for the problem (47) on a refined mesh (right)

	β			
N	**0.0001**	**0.01**	**1.0**	**100.0**
5941	11 (0.411)	4 (0.157)	2 (0.0881)	2 (0.104)
43 881	12 (1.57)	4 (0.571)	2 (0.319)	2 (0.368)
337 105	10 (6.98)	4 (3.07)	2 (1.79)	2 (1.95)
2 642 337	11 (90.7)	4 (36.9)	2 (21.7)	2 (21.8)

TABLE 3 Number of outer conjugate gradient iterations and solve time (in seconds, in parentheses) for constant coefficient problem using the AMG-based preconditioner to the Hessian, based on the unstructured mesh in Figure 1 and running on eight processors

	β			
N	**0.0001**	**0.01**	**1.0**	**100.0**
5941	42 (0.604)	39 (0.525)	40 (0.554)	40 (0.535)
43 881	41 (2.58)	40 (2.54)	41 (2.60)	41 (2.58)
337 105	37 (16.3)	40 (18.3)	40 (17.8)	40 (18.2)
2 642 337	33 (211)	40 (257)	40 (254)	40 (261)

TABLE 4 Number of unpreconditioned conjugate gradient iterations and solve time (in seconds, in parentheses) for constant coefficient Hessian problem, based on the unstructured mesh in Figure 1 running on eight processors

preconditioner $\mathbf{W}^{-1}$ to the unpreconditioned conjugate gradient. In these examples we use as many levels in our multilevel Hessian preconditioner as Hypre's BoomerAMG algorithm generates for inverting the stiffness matrix $\mathbf{A}$ needed for the forward problem (7), as well as for the adjoint problem. In Table 3 we report results for the multilevel preconditioned Hessian for this problem, for different problem sizes N (which is the number of degrees of freedom or mesh nodes) and regularization parameters β. By comparing Table 3 with the corresponding results for the unpreconditioned optimization problem, Table 4, we see that the multilevel preconditioner provides a large efficiency gain over the unpreconditioned solver.

5.4 | Comparison of geometric and algebraic multigrid

Here we again consider problem (2) with known solution (47) and $\kappa = 1$ on $[0, 1]^3$, but on a structured regular hexahedral mesh where we can compare the algebraic multigrid approach to a geometric multigrid setting. To more closely reflect our analysis, we use only a two-grid hierarchy here. As expected, the results in Table 5 show that the geometric hierarchy

TABLE 5 Comparison of conjugate gradient iteration counts for geometric and algebraic hierarchies on a structured hexahedral mesh

	$\beta = 10^{-4}$		$\beta = 10^{-6}$	
N	AMG	Geometric	AMG	Geometric
81	7	5	24	26
289	9	4	40	12
1089	9	3	42	7
4225	7	3	24	4
16 641	6	3	16	3
66 049	6	3	13	2

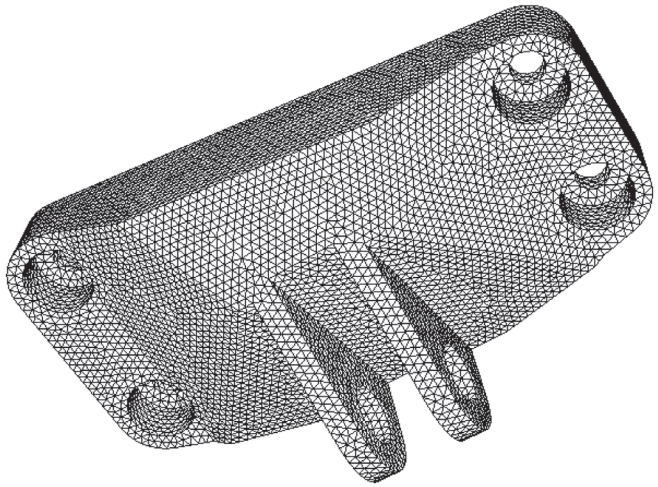

FIGURE 2 A mesh for an engine bracket challenge problem.[27] We refine this complex unstructured mesh uniformly once to run our computational examples on a mesh with two million tetrahedra

has better approximation properties and faster convergence, but in this simple setting the AMG hierarchy also shares the key property that convergence improves as $h \to 0$.

5.5 | Complex geometry

As an example of applying this approach to a complex geometry for which it is difficult to apply geometric multigrid techniques, we use as a test geometry a mesh for an engine bracket used for a design challenge problem in 2014.[27] The mesh we use for computations has two million tetrahedral elements. For the state equation, a zero Dirichlet boundary condition is imposed on the inside surface of the two eyelets pictured at the bottom of the mesh in Figure 2, with a natural Neumann condition on the remainder of the boundary. The desired state $y_d = 1$ throughout the domain, and $\kappa = 1$. The optimal control when $\beta = 1$ is shown in Figure 3. For this example, we solve on a single core and compare the results without preconditioning to a multilevel preconditioner for various regularization parameters in Table 6, which shows some speedup for the AMG-based preconditioner. We note that the number of levels used varies in this example—a full multilevel hierarchy is very effective for $\beta = 1$ but for the smaller β values we are restricted to only three levels, because using more levels leads to an indefinite preconditioner in the conjugate gradient solve.

5.6 | Varying coefficient

Algebraic multigrid methods are especially attractive when the coefficient κ is spatially varying. For this set of experiments we use the varying coefficient

$$\kappa(x) = \begin{cases} 1, & |x - x_c| > 1/4 \\ \alpha, & |x - x_c| \leq 1/4, \end{cases}$$

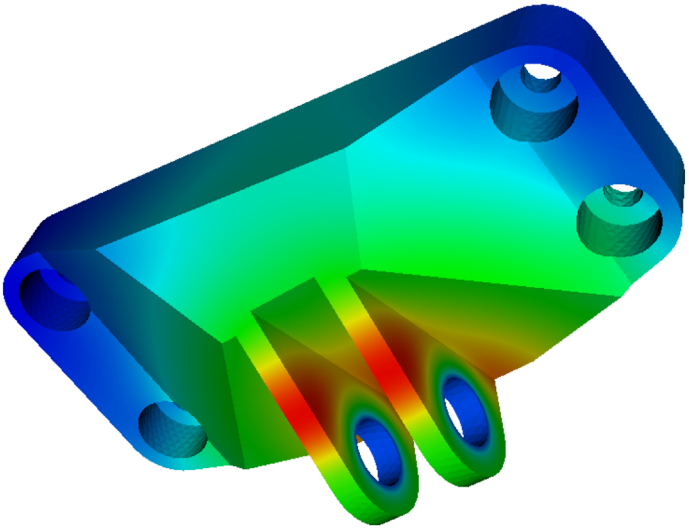

FIGURE 3 Achieved optimal control for the engine bracket mesh with $\beta = 1$

	Unpreconditioned		Multilevel	
β	Iterations	Solve time	Iterations	Solve time
1.0	43	206.23	11	64.31
0.5	42	190.55	17	97.07
0.25	45	215.32	17	118.83
0.125	50	234.07	21	144.50

TABLE 6 Conjugate gradient iteration counts and solve times (in seconds) for the unstructured engine bracket example (Figure 2) for various β parameters

		Multilevel			
α	None	2	3	4	5
0.0001	200 (1210)	57 (768.0)	—	—	—
0.001	87 (537.0)	16 (181.0)	15 (182.0)	15 (155.0)	16 (144.0)
0.01	70 (463.0)	5 (68.7)	5 (69.0)	5 (57.1)	5 (50.2)
0.1	70 (475.0)	3 (49.9)	3 (47.6)	3 (38.1)	3 (33.0)
1	70 (476.0)	3 (49.1)	3 (47.1)	3 (37.8)	3 (32.7)

Note: The original mesh is refined five times and this problem is run on eight processors.

TABLE 7 Number of outer conjugate gradient iterations and solve time (in seconds, in parentheses) for the varying coefficient problem for no preconditioning (none) and for the AMG-based preconditioner with different numbers of levels in the multilevel hierarchy

where $x_c = (0.5,0.5,0.5)$ is the center of $\Omega = [0,1]^3$, and α will vary in the experiments below. Note that the mesh does not align with the coefficient in these examples. A zero Dirichlet condition is applied on the boundary $z = 0$, while the other boundaries have homogeneous Neumann conditions. The desired state is a constant $y_d = 1$, and the regularization parameter is fixed at $\beta = 1$.

In these examples we use only the finest few levels of the multigrid hierarchy generated for the stiffness matrix $\mathbf{A}$ in our multilevel algorithm, because for a high contrast-coefficient κ (i.e., for a small α) using many levels leads to an non-convergent preconditioner $\mathbf{W}^{-1}$. In Table 7 we compare using different numbers of levels to the unpreconditioned ("none") case. We see again in this more challenging setting that our multilevel procedure has a fairly large efficiency advantage over unpreconditioned conjugate gradient.

6 | CONCLUSIONS

We have shown that previously developed geometric multigrid preconditioning techniques for optimal control of elliptic equations can be successfully extended to algebraic multigrid and implemented in standard packages. The novel

AMG-based preconditioner brings a significant algorithmic efficiency for problems where geometric multigrid based preconditioning is not applicable. In the future we expect to extend the approach to the constrained optimization case as in References 28,29, and further to optimal control of semilinear elliptic equations.

ACKNOWLEDGEMENTS

The work of the Andrew T. Barker was performed under the auspices of the U.S. Department of Energy under Contract DE-AC52-07NA27344 (LLNL-JRNL-802278). The work of the Andrei Drăgănescu is supported by the U.S. Department of Energy Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under Award Number DE-SC0005455, and by the National Science Foundation under award DMS-1913201.

CONFLICTS OF INTEREST

This work does not have any conflicts of interest.

ORCID

Andrew T. Barker https://orcid.org/0000-0003-3572-911X

REFERENCES

1. Hinze M, Pinnau R, Ulbrich M, Ulbrich S. Optimization with PDE constraints. Mathematical modelling: Theory and applications. Volume 23. New York, NY: Springer, 2009.
2. Asch M, Bocquet M, Nodet M. Data assimilation. Fundamentals of algorithms. Methods, algorithms, and applications. Volume 11. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM), 2016.
3. Brandman J, Denli H, Trenev D. Introduction to PDE-constrained optimization in the oil and gas industry. Frontiers in PDE-constrained optimization. New York, NY: Springer, 2018; p. 171–203.
4. Borzı` A, Schulz V. Computational optimization of systems governed by partial differential equations. vol. 8 Computational science & engineering. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA: 2012.
5. Xu J, Zikatanov L. Algebraic multigrid methods. Acta Numer. 2017;26:591–721.
6. Rees T, Stoll M, Wathen A. All-at-once preconditioning in PDE-constrained optimization. Kybernetika (Prague). 2010;46(2):341–360.
7. Borzı` A, Borzı` G. An efficient algebraic multigrid method for solving optimality systems. Comput Vis Sci 2004;7(3-4):183–188.
8. Borzı` A, Borzı` G. An algebraic multigrid method for a class of elliptic differential systems. SIAM J Sci Comput 2003;25(1):302–323.
9. Engl HW, Hanke M, Neubauer A. Regularization of inverse problems. Mathematics and its applications. Volume 375. Dordrecht, Netherlands: Kluwer Academic Publishers Group, 1996.
10. Drăgănescu A, Dupont TF. Optimal order multilevel preconditioners for regularized ill-posed problems. Math Comp. 2008;77:2001–2038.
11. King JT. Multilevel algorithms for ill-posed problems. Numer Math. 1992;61(3):311–334.
12. Hanke M, Vogel CR. Two-level preconditioners for regularized inverse problems I: Theory. Numer Math. 1999;83(3):385–402.
13. Kaltenbacher B. V-cycle convergence of some multigrid methods for ill-posed problems. Math Comp. 2003;72(244):1711–1730.
14. Biros G, Doğan G. A multilevel algorithm for inverse problems with elliptic PDE constraints. Inverse Probl. 2008;24(3):034010.
15. Gong W, Xie H, Yan N. A multilevel correction method for optimal controls of elliptic equations. SIAM J Sci Comput. 2015;37:A2198–A2221.
16. Hackbusch W. Fast solution of elliptic control problems. J Optim Theory Appl. 1980;31(4):565–581.
17. Tröltzsch F. Optimal control of partial differential equations. Graduate studies in mathematics. Volume 112. Providence, RI: American Mathematical Society, 2010 Theory, methods and applications, Translated from the 2005 German original by Jürgen Sprekels.
18. Vaněk P, Mandel J, Brezina M. Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. Computing. 1996;56(3):179–196.
19. Brezina M, Vaněk P, Vassilevski PS. An improved convergence analysis of smoothed aggregation algebraic multigrid. Numerical Linear Algebra Appl. 2012;19(3):441–469.
20. Brenner SC, Scott LR. The mathematical theory of finite element methods. New York, NY: Springer, 2008.
21. MacLachlan SP, Olson LN. Theoretical bounds for algebraic multigrid performance: Review and analysis. Numerical Linear Algebra Appl. 2014;21(2):194–220.
22. Hu X, Vassilevski PS. Modifying AMG coarse spaces with weak approximation property to exhibit approximation in energy norm. SIAM J Matrix Anal Appl. 2019;40(3):1131–1152.
23. hypre: High-performance preconditioners. computation.llnl.gov/casc/hypre.
24. Baker AH, Falgout RD, Kolev TV, Yang UM. Scaling hypre's Multigrid Solvers to 100,000 Cores. High performance scientific computing: Algorithms and applications. New York, NY: Springer, 2012.
25. Henson V, Yang UM. BoomerAMG: A parallel algebraic multi-grid solver and preconditioner. Appl Numer Math. 2002;41:155–177.
26. MFEM: Modular finite element methods. mfem.org.
27. Morgan H, Levatti H, Sienz J, Gil A, Bould DC. GE Jet engine bracket challenge: A case study in sustainable design. Sustainable design and manufacturing. New York, NY: Springer, 2014; p. 95–107.

28. Drăgănescu A, Petra C. Multigrid preconditioning of linear systems for interior point methods applied to a class of box-constrained optimal control problems. SIAM J Numer Anal. 2012;50(1):328–353.
29. Drăgănescu A, Saraswat J. Optimal-order preconditioners for linear systems arising in the semismooth Newton solution of a class of control-constrained problems. SIAM J Matrix Anal Appl. 2016;37(3):1038–1070.

How to cite this article: Barker AT, Drăgănescu A. Algebraic multigrid preconditioning of the Hessian in optimization constrained by a partial differential equation. *Numer Linear Algebra Appl*. 2020;e2333. https://doi.org/10.1002/nla.2333