LLNL-JRNL-412928

Algebraic Multigrid Interpolation Operators for LinearElasticity Problems

A. H. Baker, Tz. V. Kolev, U. M. Yang

May 11, 2009

Numerical Linear Algebra with Applications

Disclaimer

Improving Algebraic Multigrid Interpolation Operators for Linear Elasticity Problems

A. H. Baker *, Tz. V. Kolev, U. M. Yang

Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Box 808 L-560, Livermore, CA 94551

SUMMARY

Linear systems arising from discretizations of systems of partial differential equations can be challenging for algebraic multigrid (AMG), since the design of AMG relies on assumptions based on the near-nullspace properties of scalar diffusion problems. For elasticity applications, the near-nullspace of the operator includes the so-called rigid body modes, which are not adequately represented by the classical AMG interpolation operators. In this paper we investigate several approaches for improving AMG convergence on linear elasticity problems by explicitly incorporating the near-nullspace modes in the range of the interpolation. In particular, we propose two new methods for extending any initial AMG interpolation operator to exactly fit the rigid body modes based on the introduction of additional coarse degrees of freedom at each node. Though the methodology is general and can be used to fit any set of near-nullspace vectors, we focus on the rigid body modes of linear elasticity in this paper. The new methods can be incorporated easily into existing AMG codes, do not require matrix inversions, and do not assume an aggregation approach or a finite element framework. We demonstrate the effectiveness of the new interpolation operators on several 2D and 3D elasticity problems.

KEY WORDS: algebraic multigrid, linear elasticity, rigid body mode interpolation

1. Introduction

Algebraic multigrid (AMG) methods [3, 2, 19, 22] are increasingly popular in scientific computing due to their robustness when solving large unstructured sparse linear systems of equations, $Ax = b$, particularly when A results from the discretization of a scalar second-order elliptic partial differential equation (PDE). A distinguishing feature of AMG is that problem geometry is not needed; the "grid" is simply the set of variables. This flexibility is useful for situations when the grid is not known explicitly or is unstructured. The key to an effective multigrid method, whether geometric or algebraic, is that the error not reduced by the relaxation (or smoothing) process must be eliminated by the coarse grid correction process. Therefore, the coarse grid correction must be designed to eliminate the type of error that relaxation leaves behind, and this "algebraically smooth error" must be approximated well in the

*Correspondence to: abaker@llnl.gov.

Contract/grant sponsor: This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract No. DE-AC52-07NA27344 (LLNL-JRNL-412928).

range of the interpolation operator (see, e.g., [4]). In particular, if e is a smooth error vector in AMG that satisfies $Ae = 0$ in all interior nodes, then e should be interpolated exactly on the fine grid.

In general, a linear system of equations resulting from the discretization of a system of PDEs consists of multiple scalar functions. We refer to the original AMG algorithm designed for scalar PDEs as classical AMG, which typically fails when applied without modification to a system of PDEs. Some of the assumptions for classical AMG simply do not carry over to systems of PDEs (see, e.g., the discussions in [18, 14, 8]). For example, interpolation schemes for classical AMG assume that the nullspace of the operator is comprised only of constant vectors. For a system of PDEs, however, the nullspace is typically larger than this, and its elements may not necessarily be "constant-like". In addition, because classical AMG does not distinguish between variables corresponding to different functions, its coarsening process applied to a system of PDEs may not necessarily produce a grid suitable for effective relaxation. Intuitively, a scalar approach would only be effective for a PDE system if the coupling between the scalar functions is quite weak (see, e.g., [8]).

In this paper we address the efficient solution of linear systems arising from linear elasticity problems, which are challenging for AMG due to the presence of the rigid body modes in the nullspace of the continuous operator. Because elements in the near-nullspace of the operator should lie in the range of interpolation to obtain good AMG convergence, the rigid body modes need to be interpolated well (see, e.g. [8, 21, 14]). Since rigid body modes are typically available from application codes, our aim is to develop techniques for improving existing AMG interpolation operators by explicitly incorporating this given set of (smooth) vectors in their range.

The remainder of the paper is organized as follows. In Section 2, we review some facts regarding AMG interpolation and discuss the linear elasticity model. We also provide an overview of relevant prior work and detail our goals and strategy. In Section 3, we discuss a least-squares approach for improving the interpolation weights, which leads to approximate rigid body mode interpolation. In Sections 4 and 5, we present two new methods that exactly interpolate the rigid body modes using additional coarse degrees of freedom. We give the results from numerical experiments with the two new methods in Section 6. Finally, we make some concluding remarks in Section 7.

2. Background

2.1. AMG interpolation

A principal task in AMG when solving $Ax = b$ is the construction of the interpolation operator, which determines the values of the fine degrees of freedom (dofs) given the values of the coarse dofs. In what follows, we assume the typical situations where the coarse dofs are a subset of the fine dofs and the interpolation restricted to the coarse grid is simply the identity matrix. Furthermore, we assume that the coarse degrees of freedom have already been determined, e.g. by one of the coarsening algorithms from [22] or [10].

Let i be a fixed fine degree of freedom, the value at which we want to interpolate from the coarse grid. Define the following neighborhood sets based on the sparsity pattern of A:

$$C_i = \{\text{coarse neighbors of } i\}, \quad F_i = \{\text{fine neighbors of } i\}, \quad N_i = C_i \cup F_i, \quad X_i = \{i\} \cup C_i. \qquad (1)$$

Below we will drop the subscript i when it is implied from the context.

Let e be an arbitrary vector of algebraically smooth error, which satisfies $Ae \approx 0$. Because

interpolation should represent such vectors well, ideally the value e_i will be determined from

$$A_{ii}e_i + A_{iC}e_C + A_{iF}e_F = 0\,. \tag{2}$$

Here, and in the rest of the paper, we use a subscript to denote the restriction of vectors and matrices to the corresponding sets of indices.

Since the error at the fine neighbors, e_F, is not known, it must be eliminated from (2) using "stencil collapsing" [12], or by defining an "extension operator" [15]. An extension operator is simply a linear representation of e_F in terms of e_C and possibly e_i:

$$e_F = W_{FC}e_C + W_{Fi}e_i\,. \tag{3}$$

Because (3) is linear, it can be viewed as an interpolation process from e_X to e_F. For example, when $W_{Fi} = 0$, we have the two-step interpolation procedure $e_C \mapsto e_N \mapsto e_i$. From this perspective, it is natural to require that the local "interpolation" (3) also represents well the near-nullspace components, though probably in some weaker sense. We will make use of this point in Section 5.

Once the local interpolation matrices W_{FC} and W_{Fi} in (3) are given, the interpolation operator for the degree of freedom i is defined by the following harmonic extension from (2):

$$e_i = -(A_{ii} + A_{iF}W_{Fi})^{-1}(A_{iC} + A_{iF}W_{FC})e_C\,.$$

Note that if W_{Fi} is not zero, it must be chosen such that $A_{ii} + A_{iF}W_{Fi}$ is invertible, see [15]. However, the most commonly-used interpolation schemes have $W_{Fi} = 0$, yielding

$$e_i = -A_{ii}^{-1}(A_{iC} + A_{iF}W_{FC})e_C\,, \tag{4}$$

or equivalently,

$$e_i = -\frac{1}{a_{ii}}\sum_{j\in C_i}(a_{ij} + \sum_{k\in F_i} a_{ik}w_{kj})e_j\,.$$

Therefore, if we use P to denote the matrix representation of the corresponding global interpolation operator, then $P_{ij} = -(a_{ij} + \sum_{k\in F_i} a_{ik}w_{kj})/a_{ii}$. The coefficients $\{w\}$ above can be viewed as the weights in the local interpolation of each fine neighbor $k \in F_i$:

$$e_k = \sum_{j\in C_i} w_{kj}e_j\,. \tag{5}$$

As an example, the simple interpolation scheme known as direct interpolation (see, e.g., [22]) defines w_{kj} such that

$$e_k = \sum_{j\in C_i} \frac{a_{ij}}{\sum_{n\in C_i} a_{in}}e_j\,.$$

A more effective interpolation is the commonly used "classical" AMG interpolation (see, e.g., [6]), which uses the concept of "strength of connection" to divide the fine neighbors into those that strongly influence i, F_i^s, and those that weakly influence i, F_i^w. Then, w_{kj} in (5) are defined such that

$$e_k = \sum_{j\in C_i} \frac{a_{kj}}{\sum_{n\in C_i} a_{kn}}e_j \quad \text{for } k \in F_i^s\,, \quad e_k = e_i \quad \text{for } k \in F_i^w\,.$$

In other words, we set e_k equal to e_i in the weak fine neighbors and define it for the rest using an operator-weighted averaging.

Note that the interpolation weights for both classical and direct interpolation are chosen such that constant vectors are interpolated exactly, both locally on N_i, and globally in all interior points where the matrix has a zero row sum. This is a good choice for a scalar elliptic PDE, where the nullspace is typically represented only by constant vectors, but it is not sufficient when AMG is applied to systems of PDEs.

2.2. AMG for systems

Consider a linear system of equations resulting from the discretization of a system of PDEs with p scalar functions (or "unknowns"). Each unknown corresponds to a different physical quantity, such as a component of the displacement field in the case of elasticity. We refer to the n physical points on the grid as nodes or points, and we use the term variable or dof to refer to a specific unknown at a specific point. When solving systems of PDEs with AMG, the two standard approaches are the *unknown-based* approach and the *node-based* approach.

The unknown-based approach was first described in the early works [18] and [19]. Its strategy is to treat variables corresponding to the same unknown separately. For example, consider the unknown-wise ordering of a matrix with three unknowns u, v, and w:

$$A = \begin{bmatrix} A_{uu} & A_{uv} & A_{uw} \\ A_{vu} & A_{vv} & A_{vw} \\ A_{wu} & A_{wv} & A_{ww} \end{bmatrix}. \tag{6}$$

Treating each unknown independently translates into applying classical AMG coarsening and interpolation strategies only to the diagonal blocks of matrix (6). In other words, inter-variable couplings, such as A_{uv}, are ignored, and the resulting interpolation and restriction operators are block-diagonal. For example, with an unknown-based approach, the interpolation matrix for (6) has the form

$$P = \begin{bmatrix} P_u & 0 & 0 \\ 0 & P_v & 0 \\ 0 & 0 & P_w \end{bmatrix}.$$

The coarse grid operators are formed using the standard Galerkin formulation and the full operator matrix A. The unknown-based approach is popular due in part to its simplicity to implement and its low computational cost. Intuitively, an unknown-based approach will be effective for problems for which the cross-couplings between unknowns are not too strong and the block-diagonals of A, e.g. in (6), are matrices that are amenable to classical AMG. If this is the case, and the smoother produces error that is algebraically smooth within each unknown, then one would expect an unknown-based approach to be quite effective.

A second approach, called the node-based approach, considers all of the unknowns at a physical grid point together such that the coarsening, relaxation, and interpolation all occur in a point-wise fashion. This approach is outlined in early AMG papers (see, e.g., [18, 19]) and studied more recently in [14] and extensively in [8]. Specifically, if each of the p unknowns is discretized on the same n grid points, then we have p variables at each point and can write A with a point-wise ordering:

$$A = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{bmatrix}. \tag{7}$$

Note that the "nodal" matrices A_{ij} in (7) are size $p \times p$. A node-based coarsening results in each unknown sharing the same set of coarse grids, so the variables at each point are either all coarse or all fine. The key difference between a node-based coarsening and a classical AMG coarsening is that determining strength of connection between points i and j, for example, involves comparing the nodal matrices A_{ii} and A_{ij}. This comparison is most easily done by "condensing" the nodal matrices in (7) to

scalars

$$\begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{bmatrix} \Longrightarrow \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1n} \\ c_{21} & c_{22} & \cdots & c_{2n} \\ \vdots & \vdots & & \vdots \\ c_{n1} & c_{n2} & \cdots & c_{nn} \end{bmatrix} = C$$

via an inexpensive norm like the Frobenius-norm $\|\cdot\|_F$ or the row-sum norm $\|\cdot\|_\infty$ (see, e.g., [14, 13, 8]), though other useful metrics are possible (see examples in [8]). The scalar entries $\{c_{ij}\}$ of the "condensed" matrix C are then used to determine strength of connection as in classical AMG. Note that a node-based coarsening permits the use of a block smoother, where all variables at each point are relaxed simultaneously. This may be advantageous for strongly-coupled problems (see, e.g., [8]). The node-based approach for forming interpolation consists of the straightforward generalization of classical AMG interpolation schemes to use nodal matrices instead of scalars.

We refer to the combination of a node-based coarsening with an unknown-based interpolation as a "hybrid" approach. The two new methods for extending AMG interpolation operators that we present in Sections 4 and 5 are based on this hybrid approach.

2.3. Linear elasticity

Mathematical models of elasticity are concerned with the deformation, stress, and strain of materials due to applied external forces, which are of interest in many practical applications. The Lamé equations for linearized elasticity on a domain Ω (see, e.g., [20] or Chapter 6 of [1]) are

$$\mu\Delta\vec{U} + (\lambda+\mu)\nabla(\nabla\cdot\vec{U}) = \vec{f} \quad \text{in} \quad \Omega, \tag{8}$$

where $\vec{U}$ is the displacement vector, and λ, μ are the so-called Lamé constants. These can be expressed in terms of the material properties described by the Poisson ratio, ν, and the Young's modulus, E, as follows:

$$\mu = \frac{E}{2(1+\nu)}, \qquad \lambda = \frac{\nu E}{(1+\nu)(1-2\nu)}.$$

Typically, the boundary conditions are a mix of fixed boundary (Dirichlet) conditions, e.g., $\vec{U} = \vec{0}$ on Γ_0, and a force applied on the boundary, $\sigma(\vec{U})\cdot\vec{n} = \vec{g}$ on Γ_1, which reduces to a free boundary condition when $\vec{g} = \vec{0}$. Here, σ is the stress tensor and $\vec{n}$ is the outward unit normal. The Lamé formulation is valid for modeling small displacements due to external forces, and the equations are elliptic. When a material is nearly incompressible ($\lambda \gg \mu$, or equivalently $\nu \to 0.5$) the problem becomes very ill-conditioned, and one has to be careful with the discretization [1].

When (8) is discretized, e.g., with a linear finite element method (FEM) in 3D, the resulting matrix has the form given in (6), where $\vec{U} = (u, v, w)$ are the components of the displacement field. If $\mu \gg \lambda$, the matrix A is spectrally equivalent to its block-diagonal, which consists of the Laplacian matrices for the unknowns u, v and w. In this case, the unknown-based AMG approach discussed in the previous section will be an efficient solver for problems with A. Note that the nullspace of the block-diagonal contains only component-wise constants (i.e. displacements which are translations) and can be handled well by classical AMG.

In general, however, A will not be strongly block-diagonally dominated, and when applying the unknown-based approach one has to take into account the full nullspace of the matrix. In the case of

free boundary conditions, this nullspace, known as the set of rigid body modes (RBMs), has dimension 6 in $3D$ (3 in 2D) and can be defined by:

$$\{RBMs\} = \left\{ \vec{a} + \vec{b} \times \vec{X} \ : \ \vec{a}, \vec{b} \in \mathbb{R}^3 \right\}.$$

Besides the previously mentioned translations, i.e. displacement discretizations of $(1,0,0)$, $(0,1,0)$ and $(0,0,1)$, its basis also contains the discretizations of the so-called "rotations" $(-y,x,0)$, $(z,0,-x)$ and $(0,-z,y)$. Here $\vec{X} = (x,y,z)$ is the coordinate of a point in Ω, which is restricted to the vertices of the mesh in the linear finite element discretization. The computation of the vectors of dofs representing the RBMs is straightforward in this situation and only uses the coordinates of the mesh vertices.

Recall that for good AMG convergence it is crucial to construct the interpolation operator such that near-nullspace elements are well represented in the range. However, while unknown-based classical AMG interpolation schemes interpolate the translations exactly (constants are interpolated exactly within each unknown), the linear rotations are generally not reproduced. In fact, it is well-known that AMG convergence tends to degrade for elasticity problems as the number of free boundaries increases (and A is nearly singular) due, in part, to poor interpolation of the rotation near-nullspace modes (see, e.g. [19, 8, 7]). In other words, unknown-based classical AMG interpolation schemes will address part of the RBMs space, but not all of it. This is the motivation to consider extending such interpolations so that they also represent the "rotation" modes.

2.4. Relevant prior work

Here we describe several works related to the goal of improving AMG interpolation for linear elasticity problems, a number of which explicitly use the RBMs space. We begin by noting that the general idea of modifying classical AMG interpolation to better fit a particular set of test vectors was first proposed in the early work [19]. Later, in [21], Stüben discusses improving AMG interpolation specifically for elasticity problems based on using coordinate information. Essentially, he describes a method for improving the interpolation of the RBMs by doing a local least-squares fit and modifying the initial interpolation weights. His preliminary results on a cantilever beam appear to be an improvement over standard AMG, and we investigate this idea further in Section 3. The work in [25] proposes the use of geometric grid information to build a new AMG interpolation for 2D elasticity problems that reproduces the two translations and one rotation mode. It is not clear that this technique could be easily extended to 3D problems.

In [14], the authors investigate the node-based AMG approach for linear elasticity problems. They show that for a specific simple mesh and set of boundary conditions, the RBMs are reproduced by a node-based interpolation scheme. Their numerical results, however, generally demonstrate that an unknown-based interpolation (which they call direct-point interpolation) does as well or better than the node-based interpolation. In our experience, node-based interpolation is generally too costly and suffers from problems with inverting ill-conditioned matrices.

The smoothed aggregation (SA) variant of AMG is quite effective on linear elasticity problems as the RBMs are explicitly incorporated in and are exactly reproduced by the interpolation [23]. One of the more recent developments in this direction is the work in [5], where a parallel generalized variant of SA is developed to solve large-scale elasticity problems. This method requires a FEM framework and improves interpolation based on low-energy eigenvectors.

Another AMG approach that has been applied with success to elasticity is AMGe [4]. This method requires access to the element stiffness matrices and attempts to improve interpolation by determining the smooth error that needs to be interpolated well (approximations of the local near-nullspace

components are determined from the element stiffness matrices). However, the computational and storage costs for AMGe can be high and constructing coarse elements is non-trivial. A similar approach that does not require the individual element stiffness matrices is element-free AMGe [15]. Element-free AMGe builds "pseudo-element" matrices that require defining an extended neighborhood for each fine point and using an extension operator scheme to obtain interpolation coefficients. As with AMGe, this approach can also be effective, but it is computationally expensive (especially for parallel computing). Also notable is the work in [24] which proposes how to incorporate any given set of test vectors into an agglomeration-based AMG method.

Finally, one of the more promising non-aggregated AMG approaches to date was proposed by Dohrmann in [11]. His method is similar to element-free AMGe, but directly incorporates the rigid body modes in the local interpolation. The coarse grid nodes have both translational and rotational degrees of freedom, and local minimization problems are solved such that the rigid body modes are exactly interpolated. The results presented for this approach as compared with smoothed aggregation AMG are encouraging, though this approach is somewhat expensive and may be more difficult to parallelize. We discuss a connection with Dohrmann's approach in Section 5.

2.5. Discussion of goals and strategy

While the unknown-based approach is the typical choice for linear elasticity problems, its performance is often not good enough for use in practical application codes. Our goal is to improve its robustness by explicitly incorporating the RBMs into the unknown-based interpolation operators. As we noted earlier, RBMs are often available directly from the application code or indirectly via the grid coordinates. While the use of additional RBMs information takes away from the "black-box" nature of AMG, it is worthwhile to exploit in the case of elasticity.

Our strategy is to pursue algorithms that fit easily into an existing AMG framework so that we can utilize AMG machinery that is already in place (i.e., coarsening and interpolation algorithms). We also require that the approaches considered can be easily parallelized and are computationally inexpensive like the unknown-based approach. In particular, we avoid solving local minimization problems, and, more generally, performing matrix inversions. Furthermore, we do not assume a finite element framework or use an aggregation approach.

In the next three sections, we describe three methods that execute the above general strategy by modifying and/or augmenting an initial interpolation operator to fit the near-nullspace modes. The three methods are: a least-squares based approach, a global matrix approach, and a local neighborhood approach. The first approach is suggested in [21] and fits the near-nullspace modes only approximately. In contrast, our second approach fits the near nullspace modes exactly in a SA-like fashion by adding additional dofs to the coarse grids. Our third approach is similar to the second in that exact interpolation of the nullspace modes is achieved via additional coarse dofs, but the overall approach is more localized as the modes are fit via the extension operator.

3. A least-squares based approximation of rigid body modes

First, we explore the idea suggested in [21] of improving existing interpolation weights by doing a local least-squares (LS) fit that incorporates the RBMs space. Specifically, we form an initial interpolation matrix P in a standard way, e.g., by classical interpolation, and then apply a post-processing step that performs the LS fit. This LS fit does not change the size, or the sparsity pattern, of P and is done in a

row-wise manner, i.e. the minimization is performed over the neighborhood of each point i.

In general, we assume that we have k "smooth vectors" to fit,

$$S = [s_1\ s_2\ \cdots\ s_k],$$

where S is a column matrix and s_r, $r = 1:k$, are the RBMs. For each fine point i, we have m coarse interpolatory points, $\{c_j\}_{j=1}^m$, that correspond to the new interpolation weights,

$$w = [w_1, w_2, \cdots, w_m].$$

We let $a_i = S(i)$ be the $k \times 1$ vector corresponding to the values of the k RBMs at fine point i and $B_{ij} = S(c_j)$ be the $k \times m$ matrix corresponding to the values of the k RBMs at the m coarse interpolatory points of i. Then, for each fine point i, we determine the vector w as the minimizer of

$$\delta\|a - Bw\|^2 + (1-\delta)\|w - w_{old}\|^2 \mapsto \min,$$

where w_{old} denotes the original interpolation weights and $\delta \in [0,1]$ is a parameter which balances the deviation from the original matrix with the quality of approximation of a. Note that by the Woodbury formula, when $\delta \neq 1$, this minimization problem has a unique solution that can be written as

$$w = w_{old} + \delta B^T M_k^{-1}(a - Bw_{old}),$$

where $M_k = \delta BB^T + (1-\delta)I_k$, and I_k denotes the $k \times k$ identity matrix. In other words, we correct the original interpolation weights to better fit the smooth vectors in S. For elasticity we determine our initial P via an unknown-based interpolation, so we only fit the non-zero components of the "rotations", i.e. $k = 1$ in 2D and $k = 2$ in 3D. Therefore, the inversion of M_k is cheap. Furthermore, one can introduce different weighting of the different smooth vector fits in the LS minimization by simply replacing I_k in the definition of M_k with a diagonal weight matrix D_k.

Overall, the AMG-LS approach and its implementation are very simple (even in parallel). In terms of effectiveness, however, while using $\delta = 0.5$ generally gives the best results, we find minimal improvements, if any, compared to $\delta = 0$. For example, for the 3D linear elasticity problem described below with size $n = 27027$, we list results in Table I comparing AMG performance to that of AMG-LS.

Problem 1*: 3D cube* - This simple 3D elasticity problem is discretized on the unit cube using linear FEM and contains 3 subdomains, one with $\nu = 0.2$ and $E = 100$, and two pyramids with $\nu = 0.4$ and $E = 1$. The bottom face is fixed, and there is push at an angle on two of the adjacent boundary planes. The material subdomains and the mesh with the solution are shown in Figure 1.

Note that all experiments in this paper are done using a modified implementation of AMG based on the version available in *hypre* 2.4 [16]. In Table I, we vary the coarsening (RS = Ruge-Stüben or HMIS [9]), relaxation (GS = Gauss-Seidel, Sym = symmetric, CF = coarse-fine relaxation) and interpolation (Ext+i(5) = extended interpolation [10] with a max of 5 elements per row) choices to demonstrate that the results for this approach are sensitive to parameter choices. We note that the results for Problem 1 are representative of the typical results we observe in a range of test problems. Using the modified strength of connection scheme described in [21, 8], or an expanded neighborhood that includes all unknowns at a grid point, did not lead to improved performance.

The numerical results in Table I support the conclusion that, while the least-squares modification of existing weights is cheap, it does not seem to have a significant impact on convergence. Therefore, it is natural to consider schemes which may be more expensive (have more coarse degrees of freedom), but allow for the *exact interpolation* of the RBMs. We propose two such schemes in the next sections.

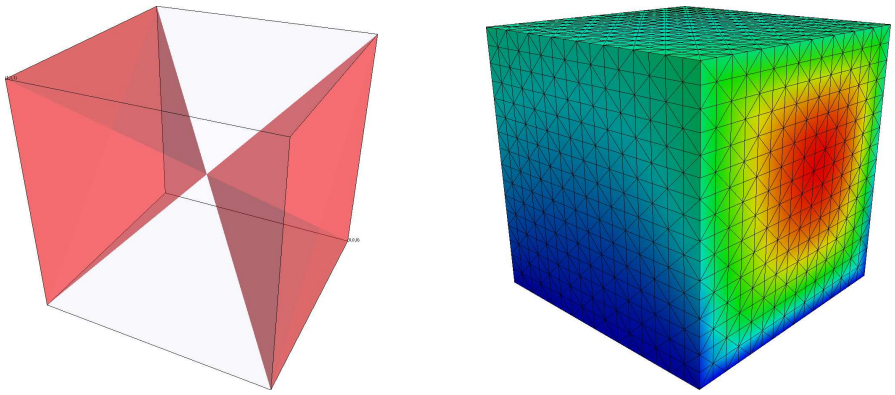

Figure 1. The (pyramidal) parts of the domain with $\nu = 0.4$ (left) and the mesh together with the magnitude of the deformation (right) for Problem 1.

Table I. A comparison of AMG to AMG-LS for Problem 1 with $n = 27027$.

Relaxation	Sym-GS	GS	GS + CF	Sym-GS	GS + CF
Coarsening	RS	RS	RS (Node-based)	HMIS	HMIS
Interpolation	Classical	Classical	Classical	Ext+i(5)	Classical
AMG	64	74	63	60	138
AMG-LS ($\delta = 0.5$)	58	75	60	74	148

4. Exact interpolation using additional degrees of freedom: global matrix approach

In this section, we present a new method for extending a given interpolation, which we call the global matrix (GM) approach. The main idea of this approach is to add new coarse degrees of freedom, similarly to the smoothed aggregation variant of AMG, which ensure *exact* interpolation of the RBMs, instead of *approximate* interpolation as with a least-squares approach. One of the motivations behind the GM approach is that the effectiveness of smoothed aggregation on linear elasticity problems can be partially explained by the fact that it interpolates the RBMs exactly via additional degrees of freedom at each coarse node.

The name of the GM approach is derived from the fact that it takes any initial global AMG interpolation matrix P and augments it with a sequence of matrices Q, each of which is responsible for the exact interpolation of a specified near-nullspace (smooth) vector. In other words, if s^F is a fixed smooth vector on the fine grid, then the new interpolation operator is

$$\tilde{P} = \begin{bmatrix} P & Q \end{bmatrix} \quad \text{s.t.} \quad s^F \in \text{range}(\tilde{P}) \,. \tag{9}$$

When defining Q, we want to keep the new interpolation $\tilde{P}$ local, so we require that Q has (at most) the same sparsity pattern as the initial P.

There are a number of possible choices for Q that satisfy (9). Below we discuss the two that we found to be most general and useful in practice.

GM variant 1: As in SA, we let

$$s^F = \tilde{P}\begin{bmatrix} 0 \\ 1 \end{bmatrix} \implies \quad s^F = Q1 \quad \implies \quad s_i^F = \sum_{j \in C_i} Q_{ij}, \tag{10}$$

where i is a fine point, cf. (1), and 1 is a vector of ones. Then for each i, we let

$$Q_{ij} = \frac{P_{ij}}{\sum_{j \in C_i} P_{ij}} s_i^F. \tag{11}$$

Note that when P interpolates constants exactly (which is typically the case with unknown-based classical AMG interpolation schemes), we have $P1^C = 1^F$, so the above formula simplifies to $Q_{ij} = P_{ij} s_i^F$.

While other choices for Q that satisfy (10) are possible, we find (11) to be the most effective in practice. Note that if we were to restrict our initial P to have a single element per row, then the new interpolation defined by GM variant 1 is comparable to SA, where the aggregates are defined by the columns of Q.

A second choice for Q that satisfies (9) is as follows.
GM variant 2: Let s^C be the coarse grid restriction of s^F and let

$$s^F = \tilde{P}\begin{bmatrix} s^C \\ 1 \end{bmatrix}. \tag{12}$$

Now Q has the following form:

$$Q_{ij} = P_{ij}\left(\frac{s_i^F}{\sum_{j \in C_i} P_{ij}} - s_j^C\right), \tag{13}$$

which simplifies to $Q_{ij} = P_{ij}(s_i^F - s_j^C)$ when P interpolates constants exactly. If we let Q be defined as in (11), one can easily show that

$$\text{Range}\left(\begin{bmatrix} P & Q \end{bmatrix}\right) \;=\; \text{Range}\left(\begin{bmatrix} P & Q - Ps^C \end{bmatrix}\right).$$

Therefore, the two variants have the same range for the final interpolation matrix $\tilde{P}$ and thus perform equivalently for a 2-level method (where an exact solver is used on the first coarse grid). Their multilevel versions differ however, as will be discussed later.

Our approach for both variants is to determine the initial P using an unknown-based interpolation, so P is a block-diagonal matrix. We require a node-based coarsening such that all unknowns share the same grid. For example, a 2D linear elasticity problem with unknowns u and v has one "rotation" rigid body mode, $[y; -x]$. We let $s^F = [y; -x]$ and compute $\tilde{P}$ in the following form:

$$\tilde{P} = \begin{bmatrix} P_u & 0 & Q_u \\ 0 & P_v & Q_v \end{bmatrix}.$$

This new interpolation adds one degree of freedom at each coarse node which represents the rotation. Note that if node-based coarsening is not required, then, in the 2D case, two new unknowns would be added by $\tilde{P}$ instead of one, though the number of non-zeros would remain the same. We have not investigated this option thoroughly yet, and instead focus on node-based coarsening which is more natural in elasticity problems.

Now, if we have k smooth vectors to interpolate, then (9) becomes

$$\tilde{P} = \begin{bmatrix} P & Q^1 & \cdots & Q^k \end{bmatrix} \quad \text{s.t.} \quad S \in \text{range}(\tilde{P}), \tag{14}$$

where S is a column matrix of k smooth vectors. For both variants, matrix Q^r is determined using $s^F = \{S\}_r$, $r = 1:k$. For example, for a 3D linear elasticity problem, three rotation vectors need to be interpolated exactly. Therefore, we add a column of Q matrices for each rotation, resulting in the addition of three new dofs at each coarse node:

$$\tilde{P} = \begin{bmatrix} P_u & 0 & 0 & Q_u^1 & Q_u^2 & Q_u^3 \\ 0 & P_v & 0 & Q_v^1 & Q_v^2 & Q_v^3 \\ 0 & 0 & P_w & Q_w^1 & Q_w^2 & Q_w^3 \end{bmatrix}.$$

To extend the method beyond two levels, we must now consider the coarsening and interpolation for the coarse grid matrix A_1. The Galerkin coarse grid matrix has the following form:

$$A_1 = \tilde{P}^T A \tilde{P} = \begin{bmatrix} P^T A P & P^T A Q \\ Q^T A P & Q^T A Q \end{bmatrix}. \tag{15}$$

In this discussion, we use only the 2D problem to outline the algorithm, since the 3D case is analogous. As on the finest level, we perform a node-based coarsening on A_1 and form an initial P_1 with any unknown-based interpolation approach in the usual manner. This way, we obtain

$$P_1 = \begin{bmatrix} P_{1_u} & 0 & 0 \\ 0 & P_{1_v} & 0 \\ 0 & 0 & P_{1_{\tilde{C}}} \end{bmatrix}, \tag{16}$$

where $\tilde{C}$ indicates the set of new coarse degrees of freedom.

At this point, the difference between the two GM variants is notable. For variant 1, (10) indicates that P_1 must exactly interpolate the columns of

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{17}$$

Therefore, P_1 as defined in (16) is sufficient (provided the unknown-based interpolation preserves constants). In contrast, for variant 2, (12) indicates that P_1 must exactly interpolate the columns of

$$\begin{bmatrix} 1 & 0 & s_u^C \\ 0 & 1 & s_v^C \\ 0 & 0 & 1 \end{bmatrix}. \tag{18}$$

This means that we must augment P_1 from the unknown-based approach with Q on *all* coarse levels. In particular, we coarsen s^C to determine s^{C_2} and modify P_1:

$$\tilde{P_1} = \begin{bmatrix} P_{1_u} & 0 & Q_{1_u} \\ 0 & P_{1_v} & Q_{1_v} \\ 0 & 0 & P_{1_{\tilde{C}}} \end{bmatrix}. \tag{19}$$

As before, the entries of Q_1 are defined per (13):

$$(Q_1)_{ij} = (P_1)_{ij} \left(\frac{s_i^C}{\sum_{j \in C_{2,i}} (P_1)_{ij}} - s_j^{C_2} \right). \tag{20}$$

The key difference here is that variant 1 only needs to have the initial interpolation augmented with Q on the first coarse grid, whereas variant 2 requires augmentation on all levels as well as storage of

coarse grid versions of the smooth vectors. In addition, we find that for variant 2, the Q^TAQ block is not amenable to classical AMG interpolation. Therefore, we use the following definition for $P_{\tilde{C}}$ on each level:

$$(P_{\tilde{C}})_{ij} = (w_1 P_u + w_2 P_v)_{ij}, \tag{21}$$

with equal weights $w_1 = w_2 = \frac{1}{2}$. This formula ensures that constants are interpolated correctly by $P_{\tilde{C}}$ and incorporates information from both u and v to determine the rotations. While the above choice appears to work well in practice, it is not clear that this is the best manner in which to form interpolation for the Q^TAQ block, and we intend to further investigate this issue in subsequent work.

The addition of new coarse dofs naturally increases the operator complexity for AMG, particularly in the 3D case. To reduce the complexity, we truncate the entries in Q^r, $r = 1 : k$, either by using an absolute or relative threshold, or by specifying a maximum number of entries for Q^r in each row. The total value of the dropped entries is distributed equally among the remaining entries of Q^r such that the row sum of Q^r remains the same after truncation. Note that the blocks corresponding to the initial P are not modified, and the Q^r corresponding to each smooth vector (rotation) is truncated separately.

Because "Q-truncation" is a necessity for controlling complexity, particularly for 3D problems, variant 2 has a distinct advantage over variant 1 in that is it more straightforward to effectively truncate. The elements of Q may vary greatly in size for variant 1, making truncation difficult. In contrast, for variant 2, a small value for Q indicates that the value of the smooth vector in a coarse neighbor is close to the value at the fine point, so it makes sense to drop such a value for Q. Therefore, while variant 1 is computationally cheaper and more directly correlated to SA, we use variant 2 in practice and for all results with the GM approach due to its amenability for truncation.

While an efficient parallel variant of the GM approach is the subject of future work, we note that its parallelization is straightforward. The primary issues are that the smooth vector information residing on neighbor processors must be shared, and the column partitioning of the new interpolation matrix will need to be adjusted. Neither of these are difficult to handle with the structures provided in modern parallel AMG libraries, such as [16].

Finally, we observe that when forming the first new interpolation operator $\tilde{P}$ in (9), $\tilde{P}$ is not guaranteed to be full rank. (We cannot easily orthogonalize the smooth vectors as is done in SA.) While the probability that the columns of Q are linearly dependent (and therefore $\tilde{P}^TA\tilde{P}$ is singular) is small, in general, it is safest to solve the coarse grid with an iterative method like the Conjugate Gradient (CG) method. In practice, we have not encountered rank deficiency issues when coarsening many times (we typically coarsen until the coarse grid size is less than 9 variables).

5. Exact interpolation using additional degrees of freedom: a local neighborhood approach

We next describe a "local" version of the GM approach which incorporates the exact representation of the RBMs in the local extension interpolation (3). We call this the local neighborhood (LN) approach, and we describe it below based on the notation introduced in Section 2.1. We also point out the connections with the previous works [11] and [17].

The GM approach presented in the last section finds a new interpolation operator $\tilde{P}$ that exactly interpolates the smooth vectors. Recall that

$$C = \{\text{coarse neighbors of } i\},$$

and let

$$\tilde{C} = \{\text{new coarse neighbors of } i\}.$$

As before, we let S be a column matrix of k smooth vectors and S_C be S at the original coarse dofs. We also set $S_{\tilde{C}} = 1$, i.e. the smooth vectors are extended to be 1 at the new coarse dofs, cf. (10), (12). For the GM approach, variant 2, we have

$$\tilde{P} = \begin{bmatrix} P & Q \end{bmatrix} \quad \text{s.t.} \quad S_i = P_{iC} S_C + Q_{i\tilde{C}} S_{\tilde{C}}.$$

In other words, $\tilde{P}$ exactly interpolates the smooth vectors in S. We now consider a "local neighborhood" approach to ensure the above property, where we assume that P is formed via harmonic extension as described in Section 2.1.

Specifically, we assume that the initial interpolation P has the form (4), where e_i is determined using the extension operator

$$e_F = W_{FC} e_C \,. \tag{22}$$

The key idea in the LN approach is to use the new coarse dofs to exactly interpolate the k smooth vectors via this extension operator:

$$e_F = W_{FC} e_C + W_{F\tilde{C}} e_{\tilde{C}} \quad \text{s.t.} \quad S_F = W_{FC} S_C + W_{F\tilde{C}} S_{\tilde{C}}. \tag{23}$$

We then define the LN interpolation $\tilde{P}$ by harmonic extension based on the local extension

$$\widetilde{W}_{FC} = [W_{FC}, W_{F\tilde{C}}] \,,$$

i.e. (cf. (4)),

$$e_i = -A_{ii}^{-1} (A_{iC} e_C + A_{iF} W_{FC} e_C + A_{iF} W_{F\tilde{C}} e_{\tilde{C}}) \,. \tag{24}$$

One can also view this approach in terms of a harmonic extension that includes the operator $A_{i\tilde{C}}$, i.e. we start from

$$A_{ii} e_i + A_{iF} e_F + A_{iC} e_C + A_{i\tilde{C}} e_{\tilde{C}} = 0,$$

where $A_{i\tilde{C}} = 0$. This local neighborhood is illustrated for 2D elasticity in Figure 2, where one new dof is added at each coarse node to represent the rotation (coarse points are denoted by solid circles).

We note that the idea of defining an extension operator (23) which preserves the RBMs was already proposed and developed by Dohrmann in [11]. His definition of W_{FC} and $W_{F\tilde{C}}$ however, is significantly more complicated and is based on solving local minimization problems.

Recall that for elasticity, S contains nullspace vectors for A, i.e. for any interior point i we have

$$A_{ii} S_i + A_{iF} S_F + A_{iC} S_C = 0 \,. \tag{25}$$

Combining this with (24) for $e = S$, we see that $\tilde{P}$ will interpolate the smooth vectors S exactly provided that (23) holds. In other words, we have localized the interpolation problem to the augmented extension operator $\widetilde{W}_{FC}$. We expect this local version to be better than the GM approach since any local approximation in $\widetilde{W}_{FC}$ does not affect the general harmonic extension principle (24), which is valid for any near-nullspace vector, not just those in S.

As in the GM approach, variant 2, in order to exactly interpolate the smooth vectors $S = \{s_r\}$, $r = 1 : k$, we define

$$W_{F\tilde{C}} = [D^F_{s_1} W_{FC} - W_{FC} D^C_{s_1}, \cdots, D^F_{s_k} W_{FC} - W_{FC} D^C_{s_k}], \tag{26}$$

where $D_s = \mathrm{diag}(s)$, and the superscript denotes restriction to the corresponding dofs. While this definition ensures (23), there are two important points regarding the LN approach that remain to be addressed: how to avoid the restriction on the form of the initial interpolation, and what to do if (25) does not hold exactly. These topics are discussed in the following paragraphs.

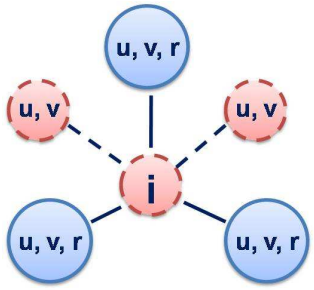

Figure 2. For 2D elasticity, one new dof (labeled "r") is added to each coarse point. Coarse connections to fine point i are denoted by solid lines and fine connections by dashed lines.

First, in order to be able to improve *any* interpolation operator determined by the AMG solver, we add a pre-processing step that will put P into the harmonic extension form required above. Specifically, given an initial interpolation matrix $P = \{P_{ij}\}$, we form a new interpolation matrix P_α via harmonic extension where the local extension operators are defined by the entries of P:

$$(P_\alpha)_{ij} = -\frac{1}{a_{ii}}\,(a_{ij} + \sum_{k \in F_i} a_{ik} w_{kj}), \tag{27}$$

where

$$w_{kj} = \frac{P_{kj}}{\sum_{n \in C_i} P_{kn}}.$$

A procedure of this type (e.g., "iterative weight interpolation" in [7]) can improve the interpolation weights. However, equally significantly for us, without any assumptions on the form of P, we now have a new interpolation matrix, P_α, that is based on harmonic extension, and from which we can determine the LN analog of Q. From (24) and (26), we have that

$$Q^r_{ij} = -\frac{1}{a_{ii}} \sum_{k \in F_i} a_{ik} w_{kj}\,((s_r)_k - (s_r)_j) \tag{28}$$

for $r = 1 : k$. The matrices P_α and Q^r together form the new LN interpolation operator

$$\tilde{P} = \left[\begin{array}{cccc} P_\alpha & Q^1 & \cdots & Q^k \end{array}\right].$$

The second point we want to address relates to our goal of a computationally inexpensive approach which does not require matrix inversions. This can lead to a situation where (25) is not true, and below we discuss how to handle this case.

Following the recent indirect Bootstrap AMG work by M. Park et al. [17], for a given smooth vector s we define the residual

$$r_i = A_{ii} s_i + A_{iF} s_F + A_{iC} s_C. \tag{29}$$

For the LN approach, we assume that $As = 0$, or $r_i = 0$. However, if we use something inexact for A, for example we let A be a diagonal block of the whole matrix, then we may have $r_i \neq 0$.

For any interpolation operator P based on harmonic extension (4), we can rewrite r_i as

$$r_i = A_{ii}(s_i - P_{iC} s_C) + A_{iF}(s_F - W_{FC} s_C). \tag{30}$$

This holds for any matrix or vector s_C and arbitrary s_i and s_F. Equation (30) is particularly interesting because it illuminates a connection between local and global interpolation. In particular, when $r_i = 0$, then (30) indicates that if we make the local interpolation W_{FC} exact, i.e. $s_F = W_{FC}s_C$, then the global interpolation P_{iC} will also be exact, i.e. $s_i = (Ps_C)_i$, which is exactly our goal. Note that the converse is not true: exact global interpolation does not imply exact local interpolation due to the nullspace of A_{iF}. Similarly, (30) shows that the local least-squares fit $W_{FC}S_CS_C^T = S_FS_C^T$ implies the global least-squares minimization $P_{iC}S_CS_C^T = S_iS_C^T$, when $r_i = 0$, see [17].

Now, when we apply an unknown-based approach, the interpolation uses only the diagonal blocks of A. Because this choice yields $r_i \neq 0$, we need to distribute the non-zero residual in order to ensure that exact interpolation by W_{FC} results in exact interpolation by P_{iC}. One simple and effective option to do that is to incorporate the residual r_i into s_F by choosing a $\tilde{s}_F$ such that

$$A_{ii}(s_i - P_{iC}s_c) + A_{iF}(\tilde{s}_F - W_{FC}s_C) = 0,$$

where

$$A_{iF}\tilde{s}_F = A_{iF}s_F - r_i.$$

For example, let A_{uu} be the block of u-to-u connections of A, then

$$r_i = A_{uu}s = (A_{uu})_{ii}s_i + (A_{uu})_{iF}s_F + (A_{uu})_{iC}s_C \neq 0.$$

If we choose $\tilde{s}_F$ such that

$$(A_{uu})_{iF}\tilde{s}_F = (A_{uu})_{iF}s_F - r_i\,,$$

then we can replace (28) with the following:

$$Q^r_{ij} = -\frac{1}{(a_{uu})_{ii}} \sum_{k \in F_i} (a_{uu})_{ik} w_{kj} ((\tilde{s}_r)_k - (s_r)_j), \tag{31}$$

where, for example,

$$(\tilde{s}_r)_k = (s_r)_k - \frac{r_i}{\sum_{k \in F_i}(a_{uu})_{ik}}.$$

With the above modification, we can apply the LN approach in an unknown-based fashion and still guarantee exact interpolation of the RBMs, even though (25) is violated.

When extending the LN approach beyond two levels, the additional considerations are analogous to that of the GM approach (variant 2). Specifically, the initial P on all coarse levels must be modified to include Q. We also modify $P_{\tilde{C}}$ on each level as in (21). In addition, the requirements for a parallel implementation are straightforward as for the GM approach.

Complexity is also an issue, though how to best perform Q-truncation for the LN approach is currently under investigation. The most straightforward manner of truncation is to use the same technique as for the GM approach: truncate the entries in Q^r, $r = 1 : k$, and distribute the value of the dropped entries equally such that the row sum of Q^r remains the same. This is the technique we used for the experimental results in the next section. However, an alternative truncation approach that is more in line with the local nature of the LN approach is to reduce the sparsity of Q by removing columns of $W_{F\tilde{C}}$. One possibility for implementing this is to examine the connections for each fine point and determine whether one or more coarse connections can be removed such that all the fine neighbors remain connected to at least one coarse neighbor. In a sense, this technique will determine a locally optimal coarse set. Investigating local truncation approaches is the subject of future work, and, while the setup costs would be slightly higher, preserving the harmonic extension principle and the local nature of the algorithm may be beneficial.

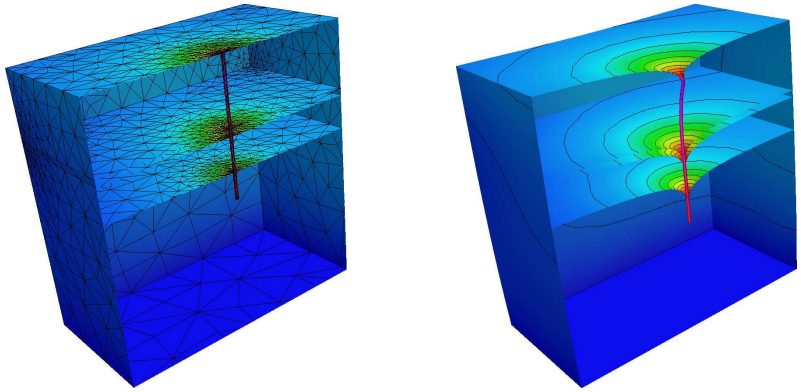

Figure 3. Cross-section of the initial mesh (left) and the deformed configuration (right) for the 3D pile problem.

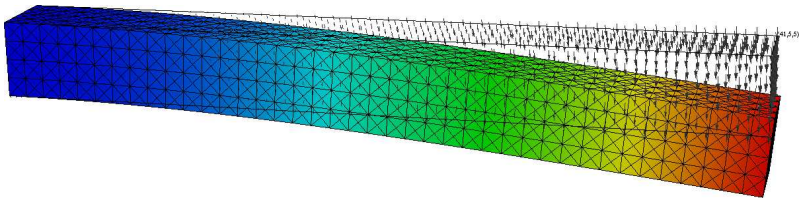

Figure 4. The deformed mesh for the 3D beam problem.

6. Numerical experiments

We first briefly describe the 2D and 3D linear elasticity problems that we use for the numerical experiments in this section. We then present 2-level and multilevel results for the GM and LN interpolations.

6.1. Example problems

All of these problems use linear elements and were discretized with the unstructured finite element package aFEM. (Recall that Problem 1 is described in Section 3.)

Problem 2*: 3D pile* - This 3D elasticity problem is discretized on a cube with 5 subdomains (4 soil layers plus a pile driver). For the pile, $\nu = 0.4$ and $E = 3150$, and for the surrounding domains $\nu = 0.1, 0.3, 0.3$, and 0.2 and $E = 0.71$. The mesh in the initial and deformed state is given in Figure 3. The bottom is fixed and the pile is pushed downward at a slight angle.

Problem 3: *3D beam* - This 3D elasticity problem on a long beam has an aspect ratio of 8:1. There is one material with $\nu = 0.2$ and $E = 1$. One end of the beam is fixed, and the opposite end is pushed downward. Tetrahedral elements are used as shown in Figure 4.

Problem 4: *2D square* - This simple 2D linear elasticity problem has zero Dirichlet boundary conditions and $\nu = 0.25$ and $E = 1$ on a single subdomain. The problem is discretized on an unstructured grid on the unit square. The mesh is shown on the left in Figure 5.

Problem 5: *2D pile* - This 2D linear elasticity problem has 3 subdomains: 2 pile drivers plus the remaining soil material. The piles have $\nu = 0.1$ and $E = 10^3$ and the soil area has $\nu = 0.4$ and $E = 1$.

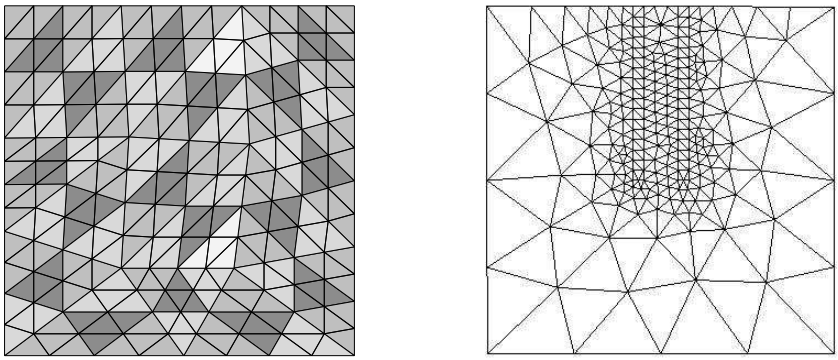

Figure 5. Unstructured mesh for 2D square problem (left) and for the 2D pile problem (right).

The piles are being pushed downwards, and the outside edges of the domain are fixed. The mesh is shown on the right in Figure 5.

Problem 6: *2D beam* - This 2D elasticity problem on a long beam has an aspect ratio of 10:1. The beam has $\nu = 0.2$ and $E = 1$. One end of the beam is fixed, and the opposite end is pushed downward. The elements are triangular, and the mesh and deformation is analogous to that of the 3D beam.

6.1.1. Two-level results We first show 2-level results on small discretizations of the six test problems to illustrate the effect of incorporating the rotations into the interpolation operator. For these problems, we used CG on the coarse grid, unknown-based classical interpolation, node-based Ruge-Stüben coarsening, and a relative convergence tolerance of 10^{-8}. We denote the AMG method by H-AMG (H = hybrid). CG preconditioned with AMG is the solver for the beam and pile problems (2,3,5, and 6), while the easier square and cube problems (1 and 4) are solved with AMG alone. We denote H-AMG with GM interpolation by H-AMG-GM and with LN interpolation by H-AMG-LN. We do not perform any Q-truncation here as we aim to illustrate the maximum benefit of the new approach. (We focus on practicalities with the multilevel results in the next section.)

Table II. A comparison of iteration counts for the 2-level methods H-AMG, H-AMG-GM, and H-AMG-LN. (Problems 2,3,5 and 6 use H-AMG/H-AMG-GM/H-AMG-LN as a preconditioner for CG.)

Problem	Description	Problem Size	H-AMG	H-AMG-GM	H-AMG-LN
1	3D cube	3723	31	14	10
2	3D pile	4225	42	7	7
3	3D beam	3995	43	9	8
4	2D square	1178	15	11	11
5	2D pile	4225	34	12	11
6	2D beam	3995	51	10	8

In general, Table II shows nice improvements in iteration counts for both H-AMG-GM and H-AMG-LN, particularly for the more difficult beam and pile problems. The 2-level results for the LN approach

are all either comparable or slightly better than the GM approach, which supports the claim in Section 5 that the LN approach has the potential to be even more effective than the GM approach.

6.1.2. Multilevel results In this section, we show experimental results for H-AMG, H-AMG-GM, and H-AMG-LN on larger versions of the six example problems. We also compare with an unknown-based approach (U-AMG). For these problems, we coarsen such that the coarse grid size is less than 50 variables for the beams (Problems 3 and 6) and less than 9 variables for the remaining problems. We use a relative convergence tolerance of 10^{-6} and use a direct solver (Gaussian elimination) on the coarse grid. As before, CG preconditioned with AMG is the solver for the more difficult beam and pile problems, while the square and cube problems are solved with AMG alone. Relaxation is symmetric Gauss-Seidel for the problems solved by CG and Gauss-Seidel otherwise. Block smoothing is used with the hybrid approaches (H-AMG, H-AMG-GM,and H-AMG-LN) for some of the problems (the "blk. sm." column in Tables III - VIII). In addition, the type of node-based coarsening (see Section 2.2) is indicated in the "nodal" column.

For these problems, we made an effort to control the operator complexity. The initial interpolation matrix P is typically truncated for all four approaches (U-AMG, H-AMG, H-AMG-GM, and H-AMG-LN) using either a relative threshold or a maximum number of elements per row; a value of $P_{max/tr}$ below one indicates a truncation threshold, whereas integers greater than one indicate the maximum number of elements per row. To compare iteration counts between methods in a meaningful way, we attempted as much as possible to obtain reasonably similar complexities for all four (or at least two) approaches. In particular, for H-AMG-GM and H-AMG-LN, we typically perform a large amount of Q-truncation using an absolute threshold (Q-th) and/or a maximum number of Q elements per smooth vector (Q-max). In most cases we also truncated the initial P more for H-AMG-GM and H-AMG-LN than for the other two approaches. Of course, if we do not Q-truncate H-AMG-GM and H-AMG-LN (and use the same P-truncation as H-AMG), then lower iteration counts will be obtained. However, the complexity will be higher, and our emphasis in this section is on practical results. To illustrate the trade-offs between low complexity and robustness, we also include some "less-truncated" results for H-AMG-GM in the tables.

We begin with the 2D problems, and results for Problem 4 with $n = 70338$ are given in Table III. This 2D square is a very easy problem for AMG, and thus does not benefit much from the GM or LN interpolation.

Table III. Problem 4: 2D square size $n = 70338$

solver	blk. sm.	nodal	coarsen	interp. ($P_{max/tr}$)	Q-th	Its (C_{op})
U-AMG	–	–	HMIS	Ext+i(.3)	–	13 (2.4)
H-AMG	no	Frobenius	HMIS	Ext+i(4)	–	12 (2.4)
H-AMG-GM	no	Frobenius	HMIS	Ext+i(4)	0.1	11 (2.5)
H-AMG-LN	no	Frobenius	HMIS	Ext+i(4)	0.1	11 (2.5)
H-AMG-GM	no	Frobenius	HMIS	Ext+i(4)	none	10 (4.4)

Results for Problem 5, the 2D pile problem, are shown in Table IV. This 2D problem is more difficult than the previous problem, even with CG, and the new approaches are more beneficial. For this problem we list results for two different types of node-based coarsening, one with block smoothing

and one without. Note that "max-element" denotes that the condensed matrix is formed using the maximum element of the corresponding nodal matrix. A comparison of U-AMG to H-AMG indicates that node-based coarsening is not as effective for this problem as unknown-based coarsening, which is a disadvantage for H-AMG-GM and H-AMG-LN. We are interested in further investigating node-based coarsening for systems, but that work is beyond the scope of the current paper.

Table IV. Problem 4: 2D pile $n = 111314$

solver	blk. sm.	nodal	coarsen	interp. ($P_{max/tr}$)	Q-th	Its (C_{op})
U-AMG/CG	–	–	HMIS	Ext+i(.2)	–	45 (2.7)
H-AMG/CG	yes	Frobenius	HMIS	Ext+i(5)	–	51 (2.7)
H-AMG-GM/CG	yes	Frobenius	HMIS	Ext+i(4)	0.001	33 (2.7)
H-AMG-LN/CG	yes	Frobenius	HMIS	Ext+i(4)	0.0008	32 (2.7)
H-AMG-GM/CG	yes	Frobenius	HMIS	Ext+i(4)	none	27 (4.3)
U-AMG/CG	–	–	HMIS	Ext+i(6)	–	43 (3.0)
H-AMG/CG	no	max-element	HMIS	Ext+i(5)	–	49 (2.9)
H-AMG-GM/CG	no	max-element	HMIS	Ext+i(4)	0.001	34 (3.1)
H-AMG-LN/CG	no	max-element	HMIS	Ext+i(4)	0.00075	33 (3.1)
H-AMG-GM/CG	no	max-element	HMIS	Ext+i(4)	none	30 (5.0)

Results for the 2D beam with $n = 83330$, Problem 6, are given in Table V. The new methods H-AMG-GM and H-AMG-LN show nice improvements for this problem with both classical interpolation and extended interpolation.

Table V. Problem 6: 2D beam $n = 83330$

solver	blk. sm.	nodal	coarsen	interp.($P_{max/tr}$)	Q-th	Its (C_{op})
U-AMG/CG	–	–	RS	Classical(.5)	–	23 (2.4)
H-AMG/CG	no	row-sum	RS	Classical	–	20 (2.3)
H-AMG-GM/CG	no	row-sum	RS	Classical	0.07	13 (2.4)
H-AMG-LN/CG	no	row-sum	RS	Classical	0.07	13 (2.4)
H-AMG-GM/CG	no	row-sum	RS	Classical	none	10 (4.4)
U-AMG/CG	–	–	HMIS	Ext+i(.22)	–	20 (2.8)
H-AMG/CG	no	row-sum	HMIS	Ext+i(5)	–	19 (2.8)
H-AMG-GM/CG	no	row-sum	HMIS	Ext+i(4)	0.1	14 (2.8)
H-AMG-LN/CG	no	row-sum	HMIS	Ext+i(4)	0.1	14 (2.8)
H-AMG-GM/CG	no	row-sum	HMIS	Ext+i(4)	none	10 (5.1)

We now consider the 3D problems, and for the easiest one, the 3D cube with $n = 27027$, Table VI shows that node-based coarsening is beneficial: H-AMG performs better than U-AMG with both classical interpolation and extended interpolation. As in the case of the 2D square, the new methods with Q-truncation provide only modest improvement for this simple problem.

The 3D pile Problem 2 with $n = 32079$ is a fairly challenging problem, and we obtain nice reductions in iterations using both H-AMG-GM and H-AMG-LN in Table VII. We show results for both classical and extended interpolation, for which U-AMG is better with extended and H-AMG is better with

Table VI. Problem 1: 3D cube $n = 27027$

solver	blk. sm.	nodal	coarsen	interp. ($P_{max/tr}$)	Q-th	Its (C_{op})
U-AMG	–	–	HMIS	Ext+i(5)	–	69 (2.4)
H-AMG	yes	row-sum	HMIS	Ext+i(5)	–	56 (1.7)
H-AMG	yes	row-sum	HMIS	Ext+i	–	49 (2.5)
H-AMG-GM	yes	row-sum	HMIS	Ext+i(3)	0.01	37 (2.5)
H-AMG-LN	yes	row-sum	HMIS	Ext+i(4)	0.006	33 (2.5)
H-AMG-GM	yes	row-sum	HMIS	Ext+i(3)	none	36 (3.1)
U-AMG	–	–	RS	Classical	–	70 (3.6)
U-AMG	–	–	RS	Classical(.7)	–	184 (2.0)
H-AMG	no	row-sum	RS	Classical	–	65 (1.6)
H-AMG-GM	no	row-sum	RS	Classical	0.1	54 (1.8)
H-AMG-LN	no	row-sum	RS	Classical	0.1	54 (1.8)
H-AMG-GM	no	row-sum	RS	Classical	none	28 (3.7)

classical interpolation. The GM and LN interpolations are helpful even with the extra truncation that has been applied to the initial P.

Table VII. Problem 2: 3D pile $n = 32079$

solver	blk. sm.	nodal	coarsen	interp. ($P_{max/tr}$)	Q-th	Q-max	Its (C_{op})
U-AMG/CG	–	–	HMIS	Ext+i(4)	–	–	69 (1.9)
H-AMG/CG	no	Frobenius	HMIS	Ext+i(6)	–	–	82 (1.9)
H-AMG-GM/CG	no	Frobenius	HMIS	Ext+i(2)	0.001	1	43 (2.0)
H-AMG-LN/CG	no	Frobenius	HMIS	Ext+i(2)	0.001	1	37 (2.0)
H-AMG-GM/CG	no	Frobenius	HMIS	Ext+i(4)	none	none	30 (3.8)
U-AMG/CG	–	–	RS	Classical(.75)	–	–	92 (2.2)
H-AMG/CG	yes	Frobenius	RS	Classical(3)	–	–	69 (2.2)
H-AMG-GM/CG	yes	Frobenius	HMIS	Classical(2)	none	none	34 (2.1)
H-AMG-LN/CG	yes	Frobenius	HMIS	Classical(2)	0.001	none	35 (2.1)

Finally, Table VIII lists two different sizes and node-based coarsenings of the 3D beam Problem 3. Here we also obtain improvement with the new GM and LN interpolations.

In general, we are able to obtain good results with H-AMG-GM and H-AMG-LN, even with the extra efforts to control the complexity, making the new methods very competitive. While the 2-level results indicate an advantage for the LN approach, the distinction is not as sharp in the multilevel case, because the GM approach performs slightly better on some problems.

7. Summary

We have proposed two new methods for extending an initial AMG interpolation operator to exactly interpolate a given set of near-nullspace vectors. Our experiments indicate that interpolating near-

Table VIII. Problem 3: 3D beam $n = 19683$ and $n = 85155$

solver	size	blk. sm.	nodal	coarsen	interp. ($P_{max/tr}$)	Q-th	Q-max	Its (C_{op})
U-AMG/CG	19683	–	–	HMIS	Ext+i(3)	–	–	50 (3.2)
H-AMG/CG	19683	no	max-element	HMIS	Ext+i(4)	–	–	50 (3.0)
H-AMG-GM/CG	19683	no	max-element	HMIS	Ext+i(2)	0.1	1	24 (3.1)
H-AMG-LN/CG	19683	no	max-element	HMIS	Ext+i(2)	0.06	1	27 (3.1)
U-AMG/CG	85155	–	–	HMIS	Ext+i(3)	–	–	59 (3.0)
H-AMG/CG	85155	yes	row-sum	HMIS	Ext+i(7)	–	–	43 (2.2)
H-AMG-GM/CG	85155	yes	row-sum	HMIS	Ext+i(4)	0.1	1	31 (2.3)
H-AMG-LN/CG	85155	yes	row-sum	HMIS	Ext+i(5)	0.1	none	28 (2.3)

nullspace vectors exactly, rather than approximately, is important. While the addition of coarse dofs is required to obtain exact interpolation, we demonstrate that reasonable complexities can be achieved through aggressive truncation of the new interpolation operator. The new approaches have a number of practical advantages, including low-complexity (due to the unknown-based interpolation), low-cost (no matrix inversions or need to solve minimization problems), and generality (no requirement for a finite element framework). More significantly, though, is the ability to derive robust interpolation by modifying existing AMG interpolation operators, rather than creating new ones. This feature is of great practical importance because it enables us to take advantage of the numerous (parallel) interpolation strategies available in AMG libraries such as [16]. Finally, while the focus of this work is interpolating the rigid body modes for linear elasticity problems, the GM and LN techniques can be used for interpolating given near-nullspace vectors in general problems.

ACKNOWLEDGEMENT

We thank John Ruge for many insightful discussions and for inspiring variant 2 of the GM approach, and we thank Panayot Vassilevski for his helpful input and advice.

REFERENCES

1. Dietrich Braess. *Finite Elements*. Cambridge University Press, 1997.
2. A. Brandt. Algebraic multigrid theory: The symmetric case. *Applied Mathematics and Computation*, 19:23–56, 1986.
3. A. Brandt, S. McCormick, and J. Ruge. Algebraic multigrid (AMG) for sparse matrix equations. In D. J. Evans, editor, *Sparsity and its Applications*, pages 257–284, Cambridge, 1984. Cambridge University Press.
4. M. Brezina, A. J. Cleary, R. D. Falgout, V. E. Henson, J. E. Jones, T. A. Manteuffel, S. F. McCormick, and J. W .Ruge. Algebraic multigrid based on element interpolation (AMGe). *SIAM Journal on Scientific Computing*, pages 1570–1592, 2000.
5. Marian Brezina, Charles Tong, and Richard Becker. Parallel algebraic multigrid methods for structural mechanics. *SIAM Journal on Scientific Computing*, 27:1534–1554, 2006.
6. W. L. Briggs, V. E. Henson, and S.F. McCormick. *A Multigrid Tutorial*. SIAM, Philadelphia, second edition edition, 2000.
7. A. J. Cleary, R. D. Falgout, V. E. Henson, J. E. Jones, T. A. Manteuffel, S. F. McCormick, G. N. Miranda, and J. W. Ruge. Robustness and scalability of algebraic multigrid. *SIAM Journal on Scientific Computing*, 21:1886–1908, 2000.
8. Tanja Clees. *AMG Strategies for PDE Systems with Applications in Industrial Semiconductor Simulation*. PhD thesis, University of Köln, 2005.

9. H. De Sterck, U.M. Yang, and J.J. Heys. Reducing complexity in parallel algebraic multigrid preconditioners. *SIAM Journal on Matrix Analysis and Applications*, 27:1019–1039, 2006.
10. Hans De Sterck, Robert D. Falgout, Josh Nolting, and Ulrike Meier Yang. Distance-two interpolation for parallel algebraic multigrid. *Numerical Linear Algebra with Applications*, 15:115–139, 2008.
11. Clark R. Dohrmann. Interpolation operators for algebraic multigrid by local optimization. *SIAM Journal on Scientific Computing*, 29(5):2045–2058, 2007.
12. Robert D. Falgout. An introduction to algebraic multigrid. *Computing in Science and Engg.*, 8(6):24–33, 2006.
13. T. Füllenbach and K. Stüben. Algebraic multigrid for selected PDE systems. In *Proceedings of the 4th European Conference on Elliptic and Parabolic Problems, Rolduc and Gaeta 2001*, pages 399–410. World Scientific, London, 2002.
14. Micheal Griebel, Daniel Oeltz, and Alexander Schweitzer. An algebraic multigrid for linear elasticity. *SIAM Journal on Scientific Computing*, 25(2):385–407, 2003.
15. Van Emden Henson and Panayot S. Vassilevski. Element-free AMGe: General algorithms for computing interpolation weights in amg. *SIAM Journal on Scientific Computing*, 23:629–650, 2001.
16. hypre. High performance preconditioners. http://www.llnl.gov/CASC/linear_solvers/.
17. Minho Park, Stephen F. McCormick, Thomas Manteuffel, John Ruge, and Marian Brezina. A new least squares based AMG. Talk at the 2009 Copper Mountain Conference on Multigrid Methods, March 2009.
18. J. Ruge. AMG for problems of elasticity. *Applied Mathematics and Computation*, 19:293–309, 1986.
19. J.W. Ruge and K. Stüben. Algebraic multigrid (AMG). In S.F. McCormick, editor, *Multigrid Methods*, volume 3 of *Frontiers in Applied Mathematics*. SIAM, 1987.
20. William S. Slaughter. *The Linearized Theory of Elasticity*. Birkhäuser, 2002.
21. K. Stüben. Some studies on algebraic multigrid. Talk at the International Workshop on Algebraic Multigrid Methods, June 2000.
22. K. Stüben. *Multigrid*, chapter An Introduction to Algebraic Multigrid, pages 413–532. Academic Press, 2001.
23. P. Vaněk, J. Mandel, and M.Brezina. Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. *Computing*, 56:179–196, 1986.
24. P. S. Vassilevski and L. T. Zikatanov. Multiple vector preserving interpolation mappings in algebraic multigrid. *SIAM Journal on Matrix Analysis and its Applications*, 27(4):1040–1055, 2006.
25. Ying-Xiong Xiao, Ping Zhang, and Shi Shu. An algebraic multigrid method with interpolation reproducing rigid body modes for semi-definite problems in two-dimensional elasticity. *Journal of Computational and Applied Mathematics*, 200:637–652, 2007.