NUMERICAL LINEAR ALGEBRA WITH APPLICATIONS
Numer. Linear Algebra Appl. 2009; **16**:19–37
Published online 16 June 2008 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nla.593

A new smoothed aggregation multigrid method for anisotropic problems

Michael W. Gee[1, *, †, ‡], Jonathan J. Hu[2] and Raymond S. Tuminaro[2]

[1] *Chair of Computational Mechanics, Technical University of Munich, Boltzmannstrasse 15, D-85747 Garching b. München, Germany*
[2] *Sandia National Laboratories, Scalable Algorithms, P.O. Box 969, MS 9159, Livermore, CA 94551, U.S.A.*

SUMMARY

A new prolongator is proposed for smoothed aggregation (SA) multigrid. The proposed prolongator addresses a limitation of standard SA when it is applied to anisotropic problems. For anisotropic problems, it is fairly standard to generate small aggregates (used to mimic semi-coarsening) in order to coarsen only in directions of strong coupling. Although beneficial to convergence, this can lead to a prohibitively large number of non-zeros in the standard SA prolongator and the corresponding coarse discretization operator. To avoid this, the new prolongator modifies the standard prolongator by shifting support (non-zeros within a prolongator column) from one aggregate to another to satisfy a specified non-zero pattern. This leads to a sparser operator that can be used effectively within a multigrid V-cycle. The key to this algorithm is that it preserves certain null space interpolation properties that are central to SA for both scalar and systems of partial differential equations (PDEs). We present two-dimensional and three-dimensional numerical experiments to demonstrate that the new method is competitive with standard SA for scalar problems, and significantly better for problems arising from PDE systems. Copyright © 2008 John Wiley & Sons, Ltd.

Received 13 August 2007; Revised 11 February 2008; Accepted 18 February 2008

KEY WORDS: AMG; multigrid; algebraic multigrid; smoothed aggregation; anisotropy

[*] Correspondence to: Michael W. Gee, Chair of Computational Mechanics, Technical University of Munich, Boltzmannstrasse 15, D-85747 Garching b. München, Germany.

[†] E-mail: gee@lnm.mw.tum.de

[‡] This work was performed while the author was at Sandia National Laboratories.

Contract/grant sponsor: United States Department of Energy; contract/grant number: DE-AC04-94-AL85000

1. INTRODUCTION

Anisotropic phenomena arise in many physical simulations for a variety of reasons. These phenomena may reflect the underlying physical model, such as is the case with anisotropic material laws in elasticity or anisotropic heat conduction. Anisotropies can also arise from discretizations with stretched grids, where the discretization is intentionally chosen coarser in one spatial direction than the other to capture certain physical characteristics (e.g. boundary layers in fluid flow). Such problems pose a challenge to standard algebraic multigrid (AMG) methods, and have been considered in [1–3] and recently in [4], among others.

The basic problem is that errors in certain directions are not smoothed by standard relaxation techniques and so it is inappropriate to coarsen in these directions. Thus, the general AMG goal is to detect directions where smoothing is not effective and to take this into account within the algorithms for both coarsening and generating prolongators.

The actual detection of directions where smoothing is ineffective remains an active research topic and is not the focus of this paper. Here, we use fairly standard algorithms. Instead, this paper addresses properly choosing the prolongator in the context of smoothed aggregation (SA). Standard SA proceeds by first selecting a set of aggregates (coarsening the problem) and then improving the derived simple prolongator via a prolongator smoothing step. One limitation is that the sparsity pattern of the final prolongator is essentially fixed once the aggregates are chosen. Within standard isotropic elliptic partial differential equations (PDEs), this limitation is not a problem as the resulting prolongator sparsity pattern gives rise to a method that is effective in cost per iteration and convergence rate. For anisotropic problems, however, small aggregates are often created to improve convergence, and this may give rise to a sparsity pattern considerably too dense to yield a cost-effective method. To rectify this, we consider a new algorithm that modifies the standard SA prolongator so that it conforms to a desired sparsity pattern. The key feature to this new algorithm is that it preserves the exact interpolation of near null space components. The near null space is the kernel of the ‘principal part’ of the underlying PDE neglecting any Dirichlet boundary conditions present [2]. This exact interpolation of near null space components is an essential feature of SA multigrid and is considered necessary for mesh-independent convergence rates. The basic idea is to perform a standard prolongator smoothing step to generate a standard SA prolongator. As in classical SA, this yields a prolongator whose basis functions are of low energy, where the energy of a basis function ϕ is defined as $\frac{1}{2}\sqrt{\phi^{\mathrm{T}} A \phi}$ and A is the symmetric positive-definite discrete differential operator. The low-energy property of coarse basis functions is needed to ensure mesh-independent convergence, see [5]. Owing to small aggregates, however, there may be too many non-zeros in the prolongator. To prevent this, the prolongator is modified by moving support among columns to sparsify the prolongator. The key idea is that this modification maintains the exact interpolation of the null space while not significantly altering the energy of the basis functions. Local orthogonalization of the tentative prolongator basis functions can be incorporated in the shifting process, and the resulting shifted prolongator preserves the interpolation properties for the near null space.

The remainder of this paper is organized as follows. Section 2.1 gives a brief overview of multigrid principals. Section 2.2 gives a summary of SA AMG. Section 2.3 discusses how coarse basis function support would ideally be chosen for anisotropic problems. Our new approach is detailed in Section 3. Section 4 discusses how SA AMG traditionally addresses problems with anisotropic properties and the associated shortcomings. Numerical experiments are presented in Section 5. Some concluding remarks are given in Section 6.

2. BACKGROUND

In this section, we give an overview of multigrid, as well as a brief introduction to SA multigrid applied to symmetric positive-definite operators.

2.1. *Multigrid overview*

Multigrid methods (e.g. [6–8]) are among the most efficient iterative algorithms for solving the linear system, $Ax=f$, associated with elliptic PDEs. The basic idea is to damp errors by utilizing multiple resolutions in the iterative scheme. High-energy (or oscillatory) components are efficiently reduced through a simple smoothing procedure, while the low-energy (or smooth) components are tackled using an auxiliary lower-resolution version of the problem (coarse grid). The idea is applied recursively on the next coarser level. An example multigrid iteration is given in Algorithm 1 to solve

$$A_1 u_1 = f_1 \tag{1}$$

Algorithm 1 Multigrid V-cycle consisting of N_{levels} grids to solve $A_1 u_1 = f_1$.

```
{Solve $A_k u_k = f_k$}
procedure multilevel($A_k, f_k, u_k, k$)
if ($k \neq N_{\text{levels}}$) then
    $u_k = \hat{S}_k(A_k, f_k, u_k)$;
    $r_k = f_k - A_k u_k$;
    $A_{k+1} = P_k^{\mathrm{T}} A_k P_k$;
    $u_{k+1} = 0$;
    multilevel($A_{k+1}, P_k^{\mathrm{T}} r_k, u_{k+1}, k+1$);
    $u_k = u_k + P_k u_{k+1}$;
    $u_k = \hat{S}_k(A_k, f_k, u_k)$;
else
    $u_k = A_k^{-1} f_k$;
end if
```

The two operators needed to specify the multigrid method fully are the relaxation (smoothing) procedures, $\hat{S}_k$, $k=1,\ldots,N_{\text{levels}}$, and the grid transfers, P_k, $k=1,\ldots,N_{\text{levels}}-1$. Note that P_k is an interpolation operator that transfers grid information from level $k+1$ to level k. The coarse grid discretization operator A_{k+1} ($k \geqslant 1$) is specified by the Galerkin product

$$A_{k+1} = P_k^{\mathrm{T}} A_k P_k \tag{2}$$

The key to fast convergence is the complementary nature of these two operators. That is, errors not reduced by $\hat{S}_k$ must be well interpolated by P_k. While constructing multigrid methods via algebraic concepts presents certain challenges, AMG can be used for several problem classes without requiring a major effort for each application. In this paper, we focus on a strategy to determine the P_k's based on algebraic principles. It is assumed that A_1 and f_1 are given.

2.2. *SA multigrid*

We describe a special type of AMG called SA multigrid. For a more detailed description, see [2, 5, 9, 10]. Specifically, we focus on the construction of SA interpolation operators P_k ($k\geqslant 1$).

The interpolation P_k is defined as a product of a given prolongator smoother S_k and a tentative prolongator $\hat{P}_k$

$$P_k = S_k \hat{P}_k, \quad k=1,\ldots,N_{\text{levels}}-1 \tag{3}$$

The basic idea of the tentative prolongator is that it must accurately interpolate certain near null space (kernel) components of the discrete operator A_k. Once constructed, the tentative prolongator is then improved by the prolongator smoother in a way that reduces the energy or smooths the basis functions associated with the tentative prolongator. Constructing $\hat{P}_k$ consists of deriving its sparsity pattern and then specifying its non-zero values. The sparsity pattern is determined by decomposing the set of nodal blocks of degrees of freedom (DOFs) associated with A_k into a set of so-called *aggregates* $\mathscr{A}_k^i$, such that

$$\bigcup_{i=1}^{N_{k+1}} \mathscr{A}_k^i = \{1,\ldots,N_k\}, \quad \mathscr{A}_k^i \cap \mathscr{A}_k^j = \emptyset, \quad 1\leqslant i<j\leqslant N_{k+1} \tag{4}$$

where N_k denotes the number of nodal blocks on level k. A nodal block refers to the submatrix that couples all DOFs defined at the same grid node. For example, a nodal block in the case of a scalar Laplace-type problem would contain $m_k=1$ DOF. In two (three)-dimensional elasticity a nodal block would consist of $m_k=2(3)$ DOFs per node, respectively.

The ideal ith aggregate $\mathscr{A}_k^i$ on level k would formally be defined by

$$\mathscr{A}_k^i = \{j_i\} \cup \mathbb{N}(j_i) \tag{5}$$

where j_i is a so-called root nodal block in A_k and

$$\mathbb{N}(j) = \{d : \|(A_k)_{jd}\| \neq 0 \text{ and } d \neq j\} \tag{6}$$

is the neighborhood of nodal blocks, $(A_k)_{jn}$ (corresponding to neighboring mesh nodes) that share a non-zero off-diagonal block entry with node j. While ideal aggregates would only consist of a root nodal block and its immediate neighboring blocks, it is usually not possible to entirely decompose a problem into ideal aggregates. Instead, some aggregates that are a little larger or smaller than an ideal aggregate must be created. Figure 1 gives an illustration of aggregates on an unstructured grid.

For this paper, each nodal block contains m_k DOFs, where for simplicity we assume that the nodal block size m_k is constant throughout A_k. Thus, the dimension of A_k is $n_k = N_k m_k$. As we will frequently refer to these dimensions in the following, we separately list them in Table I for clarity. Aggregates $\mathscr{A}_k^i$ can be formed based on the connectivity and the strength of the connections in A_k. For an overview of serial and parallel aggregation techniques, we refer to [5, 10].

Although we speak of 'nodal blocks' and 'connectivity' in an analogy to finite element discretizations here, it shall be stressed that a node is a strictly algebraic entity consisting of a list of DOFs. In fact, this analogy is only possible on the finest level; on coarser levels, $k>1$, a node denotes a set of DOFs associated with the coarse basis functions whose supports contain the same aggregate on level $k-1$. Hence, each aggregate $\mathscr{A}_k^i$ on level k gives rise to one node on level $k+1$, and

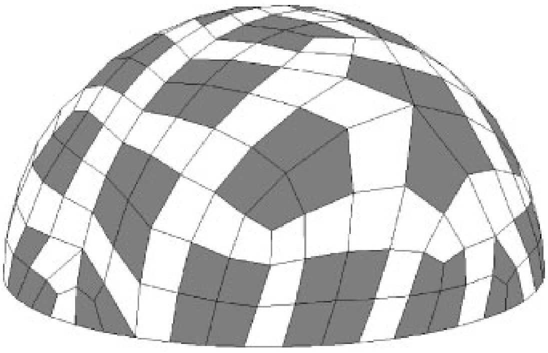

Figure 1. Aggregates of nodes on an unstructured discretization of a half sphere. Gray regions indicate aggregates containing all nodes in and adjacent to that region.

Table I. Overview of dimensions.

k	Level index; $k=1$ indicates fine grid
n_k	Number of scalar equations
m_k	Size of a nodal block (degrees of freedom per node)
N_k	Number of nodal blocks in the system on level k
$n_{\rm ns}$	Dimension of the near null space
a^i	Number of degrees of freedom in aggregate $\mathscr{A}^i$

each DOF associated with that node is a coefficient of a particular basis function associated with $\mathscr{A}_k^i$ [11]. Populating the sparsity structure of $\hat{P}_k$ derived from aggregation with appropriate values is the second step. This is done using a matrix $B_k \in \mathbb{R}^{n_k \times n_{\rm ns}}$ which represents the near null space of A_k. For a scalar PDE, the dimension of the near null space is $n_{\rm ns}=1$ and consists of a constant function, while for a vector-valued PDE corresponding to two (three)-dimensional elasticity, $n_{\rm ns}=3$ ($n_{\rm ns}=6$) and the near null space consists of two (three) constants and one (three) linear functions, respectively. In an elasticity problem, these functions are referred to as rigid body translations and rotations. On the finest mesh, it is assumed that B_k is given and that it satisfies $\tilde{A}_k B_k=0$, where $\tilde{A}_k$ differs from A_k in that Dirichlet boundary conditions are replaced by natural boundary conditions.

Tentative prolongators and a coarse representation of the near null space are constructed simultaneously and recursively to satisfy

$$B_k=\hat{P}_k B_{k+1}, \quad \hat{P}_k^{\rm T}\hat{P}_k=I, \;\; k=1,\ldots,N_{\rm levels}-1 \tag{7}$$

This guarantees exact interpolation of the near null space by the tentative prolongator. To do this, each aggregate is assigned a set of columns of $\hat{P}_k$ with a sparsity structure that is disjoint from all other columns. We define $I_k^m \in \mathbb{R}^{a^m \times n_k}$

$$I_k^m(i,j)=\begin{cases} 1 & \text{if } i+\sum\limits_{d=1}^{m-1} a^d=j, \;\; 1\leqslant i\leqslant a^m \\ 0 & \text{otherwise}\end{cases} \tag{8}$$

DOI: 10.1002/nla

to be the mth aggregate-wise row partition of the identity $I_k \in \mathbb{R}^{n_k \times n_k}$. Then,

$$B_k^m = I_k^m B_k, \quad m = 1, \ldots, N_{k+1} \tag{9}$$

is an aggregate-local block of the near null space. B_k is restricted to individual aggregates using (9) to form

$$\bar{B}_k = \begin{pmatrix} B_k^1 & & & \\ & B_k^2 & & \\ & & \ddots & \\ & & & B_k^{N_{k+1}} \end{pmatrix} \tag{10}$$

and an aggregate-local orthonormalization problem

$$B_k^i = Q_k^i R_k^i, \quad i = 1, \ldots, N_{k+1} \tag{11}$$

is solved by applying a QR algorithm. The resulting orthonormal basis Q_k^i forms the values of a block column of

$$\hat{P}_k = \begin{pmatrix} Q_k^1 & & & \\ & Q_k^2 & & \\ & & \ddots & \\ & & & Q_k^{N_{k+1}} \end{pmatrix} \tag{12}$$

whereas the coefficients R_k^i define the coarse representation of the near null space

$$B_{k+1} = \begin{pmatrix} R_k^1 \\ R_k^2 \\ \vdots \\ R_k^{N_{k+1}} \end{pmatrix} \tag{13}$$

The exact interpolation of the near null space, (7), is considered to be an essential property of an AMG grid transfer. It implies that error components in the near null space (which are not damped by conventional smoothers) are accurately approximated (and therefore eliminated) on coarse meshes. Unfortunately, (7) is not sufficient for an effective multigrid cycle. In addition, one needs to also bound the energy of the grid transfer basis functions. To do this, the tentative prolongator is improved via the prolongator smoother. The usual choice for the prolongator smoother is

$$S_k = \left(I - \frac{4}{3\lambda_k} D^{-1} A_k \right) \tag{14}$$

where $D = \text{diag}(A_k)$ and λ_k is an upper bound on the spectral radius of the matrix on level k, i.e. $\rho(D^{-1}A_k) \leqslant \lambda_k$. This corresponds to a damped Jacobi smoothing procedure applied to each column of the tentative prolongator.

DOI: 10.1002/nla

It can be easily shown that (7) holds for the smoothed prolongator when the near null space B_k is actually the true null space of A_k. In particular,

$$\begin{aligned} P_k B_{k+1} &= (I-\omega D^{-1}A_k)\hat{P}_k B_{k+1} \\ &= (I-\omega D^{-1}A_k)B_k \\ &= B_k \text{ as } A_k B_k = 0 \end{aligned} \tag{15}$$

where $\omega = 4/(3\lambda_k)$. It is emphasized that once $\hat{P}_k$ is chosen, the sparsity pattern of P_k is defined. In standard SA multigrid with well-shaped aggregates, the sparsity pattern of P_k leads to an attractive method. However, there are situations where it is beneficial to consider non-standard aggregates (e.g. for anisotropic problems). In these situations the sparsity pattern of P_k does not lead to a practical method.

With A_1, B_1 and b_1 given, the setup of the standard isotropic SA (ISA) multigrid hierarchy can be performed using (4), (11), (7), (3) and finally (2). For a more detailed discussion of SA we refer to [9, 10].

2.3. *SA with non-standard aggregates*

We give a simple two-dimensional example to illustrate details on the support of coarse-level basis functions, as these play a crucial role in understanding the new approach presented in Section 3.

Using the smoothed prolongator (3) and the Galerkin product (2) results in a coarse grid discretization of the following form:

$$A_{k+1} = \hat{P}_k^{\mathrm{T}}(I-\omega D^{-1}A_k)^{\mathrm{T}} A_k (I-\omega D^{-1}A_k)\hat{P}_k \tag{16}$$

If we neglect the D's (as they do not alter the sparsity pattern)

$$A_{k+1} = \hat{P}_k^{\mathrm{T}} q(A_k)\hat{P}_k \tag{17}$$

where $q(A_k)$ is a third degree polynomial. This implies that nodal blocks $(A_{k+1})_{ij}$ are non-zero for any aggregates $\mathscr{A}_k^i$ and $\mathscr{A}_k^j$ (or nodal block columns $(\hat{P}_k)_{\cdot i}$ and $(\hat{P}_k)_{\cdot j}$) that are less than a distance of 3 in the graph of A_k from each other.

In Figure 2(a), a symbolic visualization of aggregates and support of basis functions on a finite element grid as it is achieved with standard ISA is given. Note that the aggregates represent the support of basis functions in the tentative prolongator $\hat{P}_1$, while the visualized support of basis functions in the final prolongator results from the prolongator smoothing step (3).

Each multiplication by the matrix A_k extends the support by 1 in each direction. Thus, while $q(A_k)\hat{P}_k$ extends support by 3 in each direction, the resulting function still interacts with only the eight neighboring aggregates in Figure 2(a). This implies that the coarse matrix block row associated with the central aggregate will contain nine non-zero block entries corresponding to its immediate neighbors and itself. However, because the diameter of each aggregate is 3, the distance between the central aggregate and the non-neighboring aggregates is greater than 3; hence, no non-zero is created with non-neighboring aggregates.

Unfortunately, for an anisotropic problem, the support of basis functions in Figure 2(a) is not optimal. For the moment, let us consider the simple anisotropic Laplace equation $\varepsilon u_{xx} + u_{yy} = f, \varepsilon \ll 1$, with strong couplings in the vertical and weak couplings in the horizontal direction.

DOI: 10.1002/nla

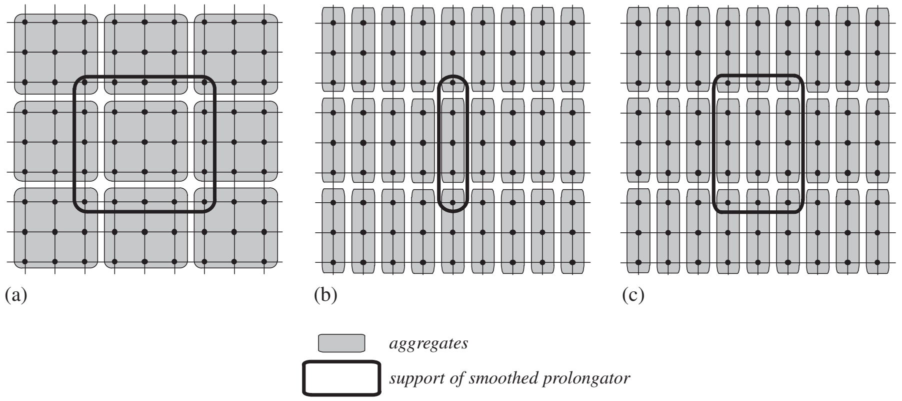

Figure 2. Symbolic visualization of aggregates and support of coarse grid basis functions: (a) isotropic behavior of standard smoothed aggregation multigrid; (b) desired behavior in anisotropic case where strong connections exist in vertical direction; and (c) anisotropic semi-coarsening aggregates with standard prolongator smoother leading to prohibitively high complexities.

Standard point-wise smoothing would damp error components in the vertical direction much better than in the horizontal direction. Therefore, error components in the horizontal direction cannot be expected to show smoothness and therefore coarsening in this direction is inappropriate. It becomes obvious that coarse basis function support should be laid out as depicted in Figure 2(b), such that the vertical direction is coarsened but not the horizontal direction. Further, basis function energy $\frac{1}{2}\sqrt{\phi^{\mathrm{T}}A\phi}$ is dominated only by the y-derivative term as $\varepsilon \ll 1$. Thus, the tentative prolongator only needs to be smoothed in the y-direction by extending support vertically.

Creating the anisotropic aggregates in Figure 2(b) also implies modifying the prolongator smoother, or else significant undesired overlap between non-neighboring aggregates appears as depicted in Figure 2(c). In Figure 2(c) the same aggregates are shown in conjunction with the support of a standard smoothed prolongator basis function. The problem, however, is that the central aggregate has 20 aggregates that are a distance 3 or less from it. This implies that a nine-point operator on the fine mesh grows to a 21-point operator on the coarse mesh according to (17). This leads to a multigrid cycle that has prohibitively high cost in storage, setup and cost per iteration. This cost is commonly measured with the so-called multigrid operator complexity

$$c = \frac{\sum_{k=1}^{N_{\text{levels}}} \text{nz}(A_k)}{\text{nz}(A_1)} \tag{18}$$

where nz() denotes the total number of non-zero entries in a matrix.

To overcome the described shortcomings of Figure 2(c), we would like to modify the prolongator smoother step in a way that the supports look like that shown in 2(b). It is easy to see that the coarse mesh stencil remains a nine-point operator with these basis functions, thus leading to an attractive method in terms of storage, setup and cost per iteration. The main difficulty and topic of this contribution now is to find a method to modify the prolongator smoother to obtain smoothed prolongator basis function support as depicted in Figure 2(b) while maintaining exact interpolation of the near null space.

3. SMOOTHED AGGREGATION MULTIGRID WITH BASIS FUNCTION SHIFTING

As a first step, a desired sparsity pattern for the prolongator P_k has to be found. The pattern should well reflect the anisotropy of the underlying problem while at the same time be as sparse as possible to guarantee low complexity (18) of the overall multigrid hierarchy. Finding such patterns for anisotropic problems is an active field of research but not the topic of this contribution. Standard ways to find such patterns will therefore be briefly reviewed in Sections 4.1 and 4.2. For now we assume that such a pattern is given.

We denote the non-zero sparsity pattern of a matrix $T\in\mathbb{R}^{n\times m}$ by

$$\mathscr{N}(T)=\{(i,j):t_{ij}\neq 0\},\quad T\in\mathbb{R}^{n\times m},\ i=1,\dots,n,\ j=1,\dots,m \tag{19}$$

and define $\mathscr{N}_k^{\text{bs}}=\mathscr{N}(P_k^{\text{bs}})$ as the desired pattern of the still undefined basis shifting smoothed prolongator P_k^{bs}. Note that for this paper it is assumed that

$$\mathscr{N}(\hat{P}_k)\subset\mathscr{N}_k^{\text{bs}} \tag{20}$$

In the following, we omit the level index $(\cdot)_k$ for ease of notation and denote coarse-level quantities $(\cdot)_{k+1}$ by $(\cdot)_c$. We repeat (7)

$$B=\hat{P}B_c \tag{21}$$

with B_c still given by (13) and stress that the smoothed prolongator (3) also satisfies (21) by construction, see (3) and (14).

Consider an interpolation operator of the form

$$P=\hat{P}-\omega T\bar{B}_c^{-1} \tag{22}$$

where

$$\bar{B}_c^{-1}=\begin{pmatrix}(R^1)^{-1} & & & \\ & (R^2)^{-1} & & \\ & & \ddots & \\ & & & (R^{N_c})^{-1}\end{pmatrix} \tag{23}$$

contains the aggregate-wise inverses of the coarse representation of the near null space. Then,

$$PB_c=\hat{P}B_c-\omega T\bar{B}_c^{-1}B_c=B-TI_c \tag{24}$$

with

$$I_c = \begin{pmatrix} I^1 \\ I^2 \\ \vdots \\ I^{N_c} \end{pmatrix} = \bar{B}_c^{-1} B_c \tag{25}$$

being a block vector of stacked aggregate-wise identities $I^i \in \mathbb{R}^{n_{\text{ns}} \times n_{\text{ns}}}$. Thus, for (22) to satisfy the interpolation of the null space condition, we need

$$T I_c = 0 \tag{26}$$

For standard unsmoothed aggregation using $\hat{P}$, $T = 0$ and (26) is trivial. For standard SA as described in Section 2.2

$$T = T^{\text{sa}} = D^{-1} A \bar{B} \tag{27}$$

and therefore

$$T^{\text{sa}} I_c = D^{-1} A \bar{B} I_c = D^{-1} A B = 0 \tag{28}$$

Unfortunately, for non-standard aggregates T^{sa} has too many non-zero entries. Therefore, we would like to find a matrix T^{bs} such that $T^{\text{bs}} I_c = 0$, T^{bs} has a prescribed sparsity pattern $\mathcal{N}^{\text{bs}}$ and the energy of the resulting basis functions in P is minimized. A related approach was considered in [2]. In [2], the sum of the basis function energies is minimized subject to the constraint that the exact interpolation of the null space has to be preserved for scalar- and vector-valued problems. Although this approach has merits, it does require the non-trivial solution of a constrained minimization problem. We instead seek an alternative approach which is direct in that it does not require iteration nor the solution of a constrained minimization problem. To do this, we relax the condition of finding minimum energy basis functions and look for a matrix T^{bs} that is 'close' to T^{sa} in a way that does not significantly alter the energy of the original grid transfer basis functions.

We therefore introduce a splitting

$$T^{\text{sa}} = T^{\text{m}} + T^{\text{nm}} \tag{29}$$

where T^{m} corresponds to all non-zeros in T^{sa} matching the desired sparsity pattern $\mathcal{N}^{\text{bs}}$:

$$\mathcal{N}(T^{\text{m}}) \subset \mathcal{N}^{\text{bs}}, \quad \mathcal{N}(T^{\text{m}} \bar{B}_c^{-1}) \subset \mathcal{N}^{\text{bs}} \tag{30}$$

where T^{nm} corresponds to the remaining (not matching) non-zeros. We then replace T^{nm} by a new matrix T^{s} which also matches the target sparsity pattern. That is

$$T^{\text{bs}} = T^{\text{m}} + T^{\text{s}}, \quad \mathcal{N}(T^{\text{bs}}) = \mathcal{N}^{\text{bs}} \tag{31}$$

such that

$$T^{\text{bs}} I_c = 0 \tag{32}$$

DOI: 10.1002/nla

This implies that

$$
\begin{aligned}
T^{\mathrm{s}} I_c &= -T^{\mathrm{m}} I_c & (33)\\
&= (T^{\mathrm{nm}} - T^{\mathrm{sa}}) I_c \\
&= T^{\mathrm{nm}} I_c & (34)
\end{aligned}
$$

Equation (32) is a simple row sum condition stating that each row sum of T^{s} must equal the corresponding negative row in T^{m}. As it is under-determined, there are many solutions including the trivial choice $T^{\mathrm{s}} = -T^{\mathrm{m}}$ which would result in the tentative prolongator $\hat{P}$.

Another possible solution is

$$T^{\mathrm{s}}_{ij} = \frac{-\sum_k (T^{\mathrm{m}})_{ik}}{\mathrm{nz}((T^{\mathrm{s}})_{j\cdot})}, \quad (i,j) \in \mathcal{N}^{\mathrm{bs}} \tag{35}$$

where $\mathrm{nz}((T^{\mathrm{s}})_{j\cdot})$ denotes the number of non-zero entries in row j of T^{s}. This corresponds to the minimum 2-norm solution. The main problem with row-wise approaches is that each component of a basis function is modified independently. This could potentially degrade the low-energy character of the standard smoothed prolongator. To avoid this, we instead pursue a column-based approach. The basic idea is to assign column subsets of T^{nm} to column subsets of T^{s} such that (34) is satisfied. In particular, for each entry T^{nm}_{ij} in the jth column of T^{nm}, there exists at least one entry T^{s}_{ik} that is allowed to be non-zero. We add the value of T^{nm}_{ij} to T^{s}_{ik}. If there are in fact several possible k's, we choose a k that allows for a large number of entries in the jth column of T^{nm} to be assigned to the same kth column of T^{s}. The basic reasoning is that we desire to split basis function support into as few as possible parts to achieve good smoothness of the resulting shifted basis functions in directions where smoothness is desired.

Algorithm 2 performs the column-wise shifting starting from T^{sa} and results in T^{bs} that is guaranteed to satisfy the desired sparsity pattern $\mathcal{N}^{\mathrm{bs}}$. The final basis shifting prolongator is then constructed according to (22) and is guaranteed to have the null space within its range. For ease of notation Algorithm 2 presents the scalar case with 1 DOF per node. The actual vector-valued

Algorithm 2 Basis function shifting algorithm.

```
Given $T^{\mathrm{sa}}$ from (27), $\mathcal{N}^{\mathrm{sa}}$, empty matrix $T^{\mathrm{bs}}$ and $\mathcal{N}^{\mathrm{bs}}(T^{\mathrm{bs}})$
for $j = 1$ to $n_c$ do {Loop over columns $T^{\mathrm{sa}}_{\cdot j}$}
    $T^{\mathrm{bs}}_{ij} = T^{\mathrm{bs}}_{ij} + \{T^{\mathrm{sa}}_{ij} : (i,j) \in \mathcal{N}^{\mathrm{bs}}\}$ {Copy elements in $\mathcal{N}^{\mathrm{bs}}$}
    $I = \{i : (i,j) \notin \mathcal{N}^{\mathrm{bs}}, T^{\mathrm{sa}}_{ij} \neq 0\}$ {Find elements not in $\mathcal{N}^{\mathrm{bs}}$}
    while $I \neq \emptyset$ do
        Define $\mathcal{G}_d = \{(i,d) : i \in I\}$
        Find column $l$ that maximizes $|\mathcal{G}_l \cap \mathcal{N}(\hat{P})|$
        $I_l = \{i : i \in I$ and $(i,l) \in \mathcal{N}^{\mathrm{bs}}\}$
        $T^{\mathrm{bs}}_{I_l l} = T^{\mathrm{bs}}_{I_l l} + T^{\mathrm{sa}}_{I_l j}$
        $I = I \backslash I_l$
    end while
end for
```

case of interest is straightforward to derive by applying the algorithm to nodal block columns of T^{sa} and T^{bs}, respectively.

4. REVIEW OF STANDARD SMOOTHED AGGREGATION TECHNIQUES FOR ANISOTROPIC PROBLEMS

In order to test and compare the presented basis shifting approach we briefly review and discuss techniques that are commonly used on anisotropic problems to mimic anisotropic coarsening, find a desired sparsity pattern and modify the prolongator smoother.

4.1. Creation of anisotropic aggregates

One approach originally presented in [5] is to define a *strongly coupled neighborhood* of node i on level k

$$N_i(\varepsilon)=\{j:|a_{ij}|\geqslant\varepsilon\sqrt{a_{ii}a_{jj}}\} \tag{36}$$

where the choice of the drop tolerance ε depends on the problem. A *thresholded* or *filtered* matrix $A_k^{\text{F}}=(a_{ij}^{\text{F}})$ from A_k is constructed by

$$a_{ij}^{\text{F}}=\left\{\begin{matrix} a_{ij} & \text{if } j\in N_i(\varepsilon)\\ 0 & \text{otherwise}\end{matrix}\right\} \text{ if } i\neq j,\quad a_{ii}^{\text{F}}=a_{ii}+\sum_{j=1,j\neq i}^{n}(a_{ij}-a_{ij}^{\text{F}}) \tag{37}$$

The diagonal modification ensures that constant functions remain in the near null space of A_k^{F}.

A_k^{F} can then be used with a standard isotropic aggregation method to form anisotropic aggregates. For vector-valued PDEs such as elasticity, the definition of the strongly coupled neighborhood (36) has to be altered to incorporate norms of nodal block matrices, making the choice of a good thresholding strategy for such problems more complicated, see [5].

In the case where the anisotropy results from a stretched discretization, a second approach involves the construction of an auxiliary matrix A_k^{pL}, the *pseudo-Laplacian* (pL), that reflects distances among nodes of the discretization. Provided that nodal coordinate information $\mathbf{x}_1\in\mathbb{R}^{N_1\times 3}$ of the fine grid $k=1$ is available, one constructs a matrix $A_k^{\text{pL}}=(a_{ij}^{\text{pL}})_k$ such that

$$\begin{aligned} a_{ij}^{\text{pL}}&=-\frac{1}{d_{ij}},\quad d_{ij}=\sum_{l=1}^{3}(x_{il}-x_{jl})^2 \text{ if } a_{ij}\neq 0 \text{ and } i\neq j\\ a_{ii}^{\text{pL}}&=\sum_j\frac{1}{d_{ij}}\end{aligned} \tag{38}$$

Coarse-level matrices A_k^{pL}, $k>1$ are formed by recursively building aggregate coordinates $\mathbf{x}_k\in\mathbb{R}^{N_k\times 3}$ of all nodes in an aggregate $\mathscr{A}_k^i$ using averages of coordinates.

Off-diagonal entries of A_k^{pL} properly reflect decay with increasing distance between neighboring nodes while diagonal entries are formed such that row sums are zero. Thresholding (36) and (37) is then applied to A_k^{pL} to form a thresholded pseudo-Laplacian, A_k^{pLF}. It can then be used in the aggregation process to create anisotropic aggregates but must *not* be used in the prolongator

DOI: 10.1002/nla

smoothing process to be discussed in Section 4.2, as it is physically unrelated to the problem at hand. Instead a matrix A_k^{F} is constructed via (37) using A_k^{pLF} only for the sparsity pattern. The pseudo-Laplacian A_k^{pLF} from (38) is a scalar operator with 1 DOF per node, and has to be extended to a matching block node form if used as auxiliary operator for a vector-valued problem. This can easily be achieved by forming a new matrix with nodal blocks of appropriate size m_k, where the main diagonal of a nodal block replicates the scalar value.

4.2. Anisotropic prolongator smoothing

An anisotropic variant of the prolongator smoothing step (3) can be obtained using the thresholded matrix A_k^{F} from (37) in the prolongator smoother (14)

$$S_k^{\mathrm{F}} = \left(I - \frac{4}{3\lambda_k} (D^F)^{-1} A_k^{\mathrm{F}} \right) \tag{39}$$

where λ_k is given in Section 2.2. This approach leads to a correct interpolation of the near null space

$$B_k = S_k^{\mathrm{F}} \hat{P}_k B_{k+1} \tag{40}$$

for *scalar* problems, where the only near null space component is the constant. This is due to the diagonal modification in (37) which guarantees zero row sums of A_k^{F} away from Dirichlet boundary conditions. However, (40) in general is *not* satisfied in the case of vector-valued PDEs. In this case, zero row sums of A_k^{F} are a necessary but not sufficient condition to guarantee that A_k^{F} and A_k share the same near null space. Detailing (3) and (7) and utilizing a thresholded matrix in the construction of the prolongator smoother in the more general case of vector-valued PDEs leads to

$$B_k \neq (\hat{P}_k - \omega (D_k^{\mathrm{F}})^{-1} A_k^{\mathrm{F}} \hat{P}_k) B_{k+1} \tag{41}$$

as

$$A_k^{\mathrm{F}} \hat{P}_k B_{k+1} = A_k^{\mathrm{F}} B_k \neq 0 \tag{42}$$

Usually, this shortcoming of thresholded prolongator smoothing for vector-valued problems is simply ignored. This leads to deterioration of convergence rates. We will compare our basis shifting approach with it in Section 5.

5. NUMERICAL EXPERIMENTS

A series of numerical tests are performed to evaluate the modified prolongator smoother algorithm. In particular, the basis function shifting approach is applied to several anisotropic problems. The anisotropy is due to mesh stretching (i.e. elements with poor aspect ratios) in all cases. As mentioned earlier, the detection of general anisotropic behavior is still a research question. However, a pL works fairly well for mesh stretching and is used to generate anisotropic aggregates for these experiments in conjunction with a drop tolerance range of $0.18 \leqslant \varepsilon \leqslant 0.25$ in (36). The basis shifting SA (BSSA) approach is compared with plain aggregation using isotropic aggregates (IPA), ISA and SA with anisotropic aggregation (ASA). In IPA, the prolongator smoother step is skipped. This means that the prolongator is already fairly sparse and so there is no need to further sparsify it. ISA

Table II. Solution methods used in experiments.

IPA	Isotropic plain aggregation multigrid
ISA	Isotropic smoothed aggregation multigrid
ASA	Smoothed aggregation multigrid with anisotropy detection (Sections 4.1 and 4.2)
BSSA	Basis shifting smoothed aggregation multigrid

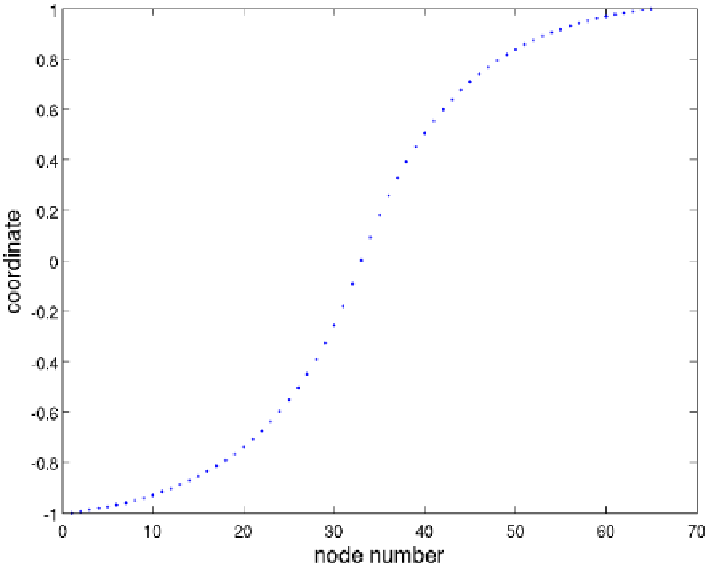

Figure 3. Distribution of x-coordinates for the IFISS-generated problem.

corresponds to standard SA with standard aggregates. ASA uses the pseudo-Laplacian to determine the sparsity pattern in conjunction with A_k^{F} defined by (37) within the prolongator smoother though this approach has not been explicitly designed for vector-valued problems. BSSA uses Algorithm 2 in conjunction with the pseudo-Laplacian. Different algorithms are summarized in Table II. All results correspond to a conjugate gradient iteration using a multigrid V-cycle preconditioner with one Gauss–Seidel iteration within the pre- and post-smoother.

5.1. Anisotropic Poisson's equation in two dimensions

In this example, we consider the two-dimensional Poisson equation on a square domain and discretized with Q1 elements. This problem was generated using the IFISS packages [12]. Q1 elements are particularly difficult for AMG methods as the matrix coefficients corresponding to weak coupling are still relatively large. The lower boundary corresponds to a Dirichlet boundary condition while the other three boundaries correspond to Neumann conditions. The finite element mesh is uniform in the center of the mesh and increasingly anisotropic toward the boundaries. The distribution of the x-coordinates for the 65×65 mesh is given in Figure 3, and the finite element mesh is given in Figure 4(a).

DOI: 10.1002/nla

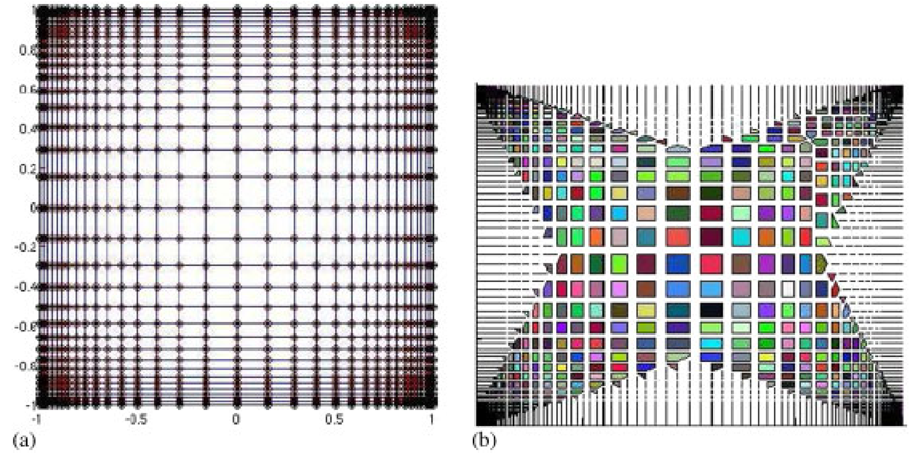

Figure 4. Mesh and aggregates for Poisson's equation on unit cube: (a) finite element mesh with Q1 elements and (b) fine grid aggregates.

Table III. Iterations and complexity for two-dimensional Poisson equation discretized with Q1 elements on unit square.

No. elements	No. equations	IPA	ISA	ASA	BSSA
17^2	289	18/1.11	17/1.10	9/1.31	10/1.31
33^2	1089	34/1.11	27/1.11	15/1.37	12/1.37
65^2	4225	49/1.12	45/1.12	21/1.37	16/1.37
129^2	16641	99/1.12	77/1.12	31/1.38	18/1.38

The anisotropic fine grid aggregates using the pseudo-Laplacian are given in Figure 4(b). Note that the aggregate shapes accurately reflect the direction of grid anisotropy. Results are given in Table III corresponding to choosing a random right-hand side and a zero initial guess. Convergence is declared when the initial residual is reduced by 10 orders of magnitude. As expected IPA gives poor convergence rates as it employs sub-optimal piece-wise-constant interpolation. ISA generates misshaped aggregates that correspond to coarsening in all spatial dimensions including those of weak coupling. Thus, its poor convergence rates are also not surprising. ASA exhibits better iteration counts, but there is still h-dependence. BSSA, although not completely h-independent, shows the best iteration counts. Furthermore, the BSSA complexity is the same as ASA, as the ASA prolongator and the BSSA prolongator used identical target non-zero patterns.

5.2. Elasticity for a beam on anisotropic grid in three dimensions

This example describes a three-dimensional linear elastic beam with full Dirichlet boundary conditions on one end and a constant distributed surface load on the other. The initial guess was chosen to be zero. The geometry and material properties of the example and one of the investigated discretizations are given in Figure 5. It is generated using CARAT [13]. The discretization is

DOI: 10.1002/nla

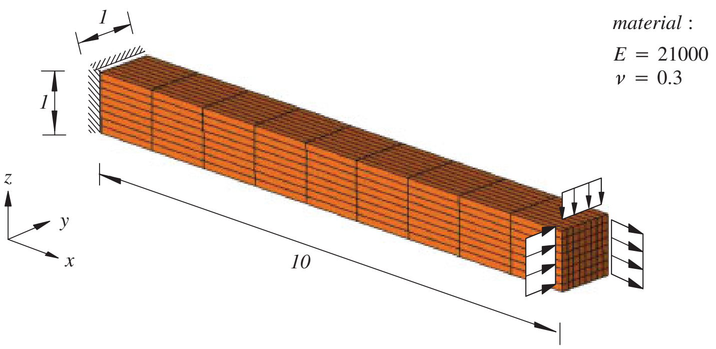

Figure 5. Three-dimensional beam with discretization $9\times8\times8$.

Table IV. Iterations and complexity for three-dimensional beam.

Discretization (elements $x\times y\times z$)	No. equations	IPA	ISA	ASA	BSSA
$3\times2\times2$	81	22/1.01	21/1.01	14/1.29	10/1.29
$9\times8\times8$	2187	117/1.10	109/1.12	42/1.35	14/1.35
$10\times9\times9$	3000	77/1.19	63/1.19	43/1.53	12/1.53
$15\times14\times14$	10125	161/1.13	117/1.15	39/1.43	14/1.43
$19\times18\times18$	20577	104/1.17	79/1.17	55/1.54	12/1.54

IPA, isotropic plain aggregation multigrid; ISA, isotropic smoothed aggregation multigrid; ASA, smoothed aggregation multigrid with anisotropy detection; BSSA, basis shifting smoothed aggregation multigrid.

chosen such that it is isotropic in the y–z plane (cross section of beam) and stretched along the main axis of the beam. Thus, coupling is 'weak' along the x-axis and isotropic in the y–z plane.

To obtain coarse grid discretizations of good approximation quality and low complexity here, it is desirable to have aggregates that are of planar shape and lie in the y–z plane. This is again achieved by using the pseudo-Laplacian operator together with thresholding as described in Section 4.1 in the aggregation process. In Table IV, we give iteration numbers and preconditioner complexities (18) for several levels of refinement.

Although IPA and ISA result in very low complexities due to the isotropic aggregation, iteration numbers are high. SA with anisotropy detection (ASA) results in slightly higher complexity numbers while exhibiting a slight advantage in iteration numbers compared with IPA and ISA. Although the trend is similar to the Laplace operator, the improvement is much more modest for ASA. This is due to the inexact interpolation of the near null space as pointed out in Section 4.2. In particular, the null space is six dimensional, corresponding to three translations and three rotations. The filtered matrix used in the prolongator smoother, however, is only adapted to properly maintain the constant within the null space. BSSA exhibits the same moderate complexity as the ASA approach throughout the investigated range of discretizations due to the limitation of the basis function support through shifting and usage of aggregates that lie in the y–z plane. It shows low iteration numbers and h-independence. This example illustrates the importance of maintaining null space properties and how this is easily accomplished with basis function shifting even for PDE systems, which is in contrast to the filtering approach.

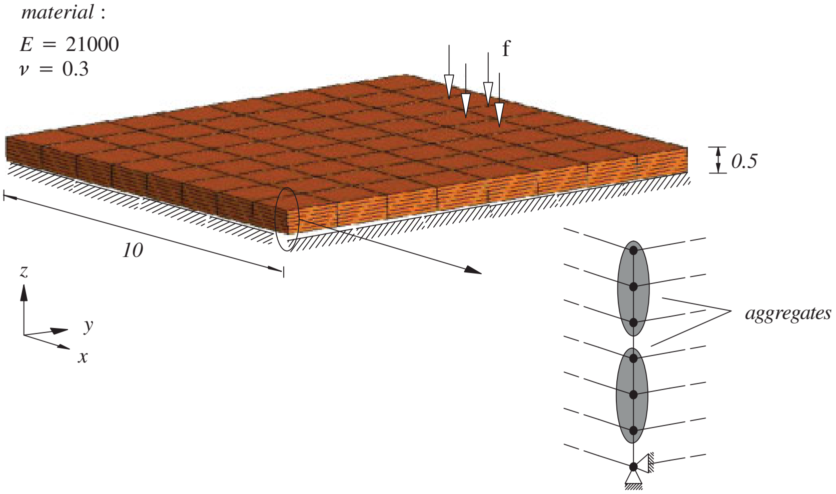

Figure 6. Three-dimensional thick plate with discretization $8\times 8\times 6$.

Table V. Iterations and complexity for three-dimensional thick plate.

Discretization (elements $x\times y\times z$)	No. equations	IPA	ISA	ASA	BSSA
$4\times 4\times 3$	252	70/1.05	67/1.05	14/1.98	15/1.98
$8\times 8\times 6$	1605	211/1.07	186/1.09	35/2.13	15/2.13
$16\times 16\times 12$	11079	346/1.08	270/1.11	61/2.29	14/2.29

IPA, isotropic plain aggregation multigrid; ISA, isotropic smoothed aggregation multigrid; ASA, smoothed aggregation multigrid with anisotropy detection; BSSA, basis shifting smoothed aggregation multigrid.

5.3. Elasticity for a plate on anisotropic grid in three dimensions

In this second elasticity example, a thick plate shown in Figure 6 is discretized using stretched finite elements that are thin in one dimension. Strong connections exist in the z-axis direction (across the plate thickness) and therefore aggregation should take place in this direction as well.

In this case, pseudo-Laplacian yields one-dimensional aggregates that are aligned with the z-axis and have no geometrical extension in the y- and x-directions, see Figure 6. As such 'degenerate' aggregates cannot represent the full set of six rigid body modes, we limit the number of near null space components in the construction of the multigrid hierarchy to 5 and exclude the rotation around the z-axis. As such 'degenerate' aggregates are more likely to appear in aggregation designed for anisotropic problems, handling this case in a more general manner is subject to further investigation, but shall not be further addressed in this contribution. In Table V, iteration numbers and preconditioner complexity for the BSSA approach are compared with the same set of isotropic and anisotropic approaches already used in Section 5.2.

IPA and ISA do not perform well on this example and do not scale in problem size. ASA leads to the same complexities as the new basis shifting approach (BSSA) because they use the same desired sparsity pattern. As the ASA approach inherently cannot guarantee correct interpolation of the null space, iteration numbers are not satisfactory. Although complexities with the basis shifting

DOI: 10.1002/nla

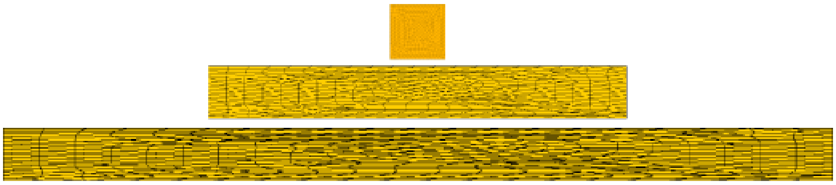

Figure 7. Rectangular domain with unstructured discretization with 980 unknowns. Top: ratio 1:1, middle: ratio 8:1, bottom: ratio 16:1.

Table VI. Iterations and complexity for two-dimensional stretched unstructured rectangle with dimension ratios $x:y=1:1, 8:1, 16:1$.

Discretization (no. equations)	Dimensions $(x:y)$	IPA	ISA	ASA	BSSA
980	1:1	19/1.20	13/1.33	16/1.35	9/1.35
3760	1:1	25/1.22	12/1.35	22/1.36	10/1.36
980	8:1	60/1.20	45/1.33	37/2.01	20/2.01
3760	8:1	105/1.22	67/1.35	57/2.03	26/2.03
980	16:1	81/1.20	66/1.33	45/2.14	25/2.14
3760	16:1	158/1.22	107/1.35	80/2.10	38/2.10

IPA, isotropic plain aggregation multigrid; ISA, isotropic smoothed aggregation multigrid; ASA, smoothed aggregation multigrid with anisotropy detection; BSSA, basis shifting smoothed aggregation multigrid.

method are higher than with any of the isotropic methods due to the one-dimensional shape of the aggregates, iteration numbers are significantly lower than with any other technique and scale well with problem size.

5.4. *Elasticity on anisotropic unstructured grid in two dimensions*

In this example, a two-dimensional elasticity problem on a rectangular domain is discretized with an unstructured stretched quadrilateral grid. The stretched discretization is obtained by first discretizing a unit square and then stretching it to obtain a rectangular domain, see Figure 7. Dirichlet boundary conditions are imposed along one edge of the rectangle corresponding to the y-direction. Two unstructured refinements are studied in Table VI.

None of the tested multigrid methods exhibit mesh-independent convergence on the stretched grids. BSSA and ISA lead to mesh-independent convergence rates for the unstretched meshes. The additional complications associated with anisotropic phenomena that are not aligned with the mesh have been observed in [14]. Generally, non-aligned anisotropic phenomena are considered much harder problems for multigrid methods. In this example, the use of a stretched unstructured discretization using quadrilaterals results in a poor discrete approximation to the PDE problem due to the high distortion of the elements. This affects not only the quality of the solution but also the multigrid performance as shown in Table VI and even the performance of basic relaxation methods. The new basis shifting method BSSA though converges significantly better than the other approaches in all stretched cases while maintaining the same complexity as the ASA approach.

DOI: 10.1002/nla

6. CONCLUSIONS

A new method to form SA interpolation operators for anisotropic problems is studied and compared with traditional approaches to handle anisotropy. We show that using this approach, a prolongation operator with a prescribed anisotropic sparsity pattern can be constructed that leads to moderate complexities and maintains exact interpolation of a given null space for scalar- and vector-valued problems. The latter property is important to obtain low and scalable iteration numbers and is demonstrated in the given examples.

ACKNOWLEDGEMENTS

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94-AL85000.

REFERENCES

1. Moulton JD, Dendy JJE, Hyman JM. The black box multigrid numerical homogenization algorithm. *Journal of Computational Physics* 1998; **142**:80–108.
2. Mandel J, Brezina M, Vaněk P. Energy optimization of algebraic multigrid bases. *Computing* 1999; **62**:205–228.
3. Wan J, Chan T, Smith B. An energy-minimizing interpolation for robust multigrid methods. *SIAM Journal on Scientific Computing* 2000; **21**:1632–1649.
4. Brezina M. Handling of anisotropies in smoothed aggregation code parsamis. *Proceedings of Thirteenth Copper Mountain Conference on Multigrid Methods*, 2007.
5. Vaněk P, Mandel J, Brezina M. Algebraic multigrid based on smoothed aggregation for second and fourth order problems. *Computing* 1996; **56**:179–196.
6. Hackbusch W. *Multigrid Methods and Applications*. Computational Mathematics, vol. 4. Springer: Berlin, 1985.
7. Trottenberg U, Oosterlee C, Schüller A. *Multigrid*. Academic Press: London, 2001.
8. Briggs WL, Henson VE, McCormick S. *A Multigrid Tutorial* (2nd edn). SIAM: Philadelphia, PA, 2000.
9. Vaněk P, Brezina M, Mandel J. Convergence of algebraic multigrid based on smoothed aggregation. *Numerische Mathematik* 2001; **88**:559–579.
10. Tuminaro R, Tong C. Parallel smoothed aggregation multigrid: aggregation strategies on massively parallel machines. In *SuperComputing 2000 Proceedings*, Donnelley J (ed.). 2000.
11. Brezina M, Falgout R, MacLachlan S, Manteuffel T, McCormick S, Ruge J. Adaptive smoothed aggregation (αSA) multigrid. *SIAM Review* 2005; **47**(2):317–346.
12. Elman H, Silvester D, Wathen A. *Finite Elements and Fast Iterative Solvers with Applications in Incompressible Fluid Dynamics*. Numerical Mathematics and Scientific Computation. Oxford University Press: Oxford, 2005.
13. Ramm E, Wall WA. Carat: a multifield parallel research finite element solver for structural and CFD problems. *Technical Report*, University of Stuttgart, Germany, 2000.
14. Brannick J, Brezina M, MacLachlan S, Manteuffel T, McCormick S, Ruge J. An energy-based AMG coarsening strategy. *Numerical Linear Algebra with Applications* 2006; **13**:133–148.

DOI: 10.1002/nla