NUMERICAL LINEAR ALGEBRA WITH APPLICATIONS
Numer. Linear Algebra Appl. 0000; **00**:1–10
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nla

Adaptive Multilevel AMG for Sequences of Problems with Gradually Changing Random Coefficients

Delyan Kalchev[1], Christian Ketelsen[1] and Panayot S. Vassilevski[1*]

[1]*Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, P.O. Box 808, L-560, Livermore, CA 94550, U.S.A.*

SUMMARY

This paper describes an adaptive algebraic multigrid method based on [8] and [14]. The target is to efficiently solve large sequences of problems. To this purpose, already built hierarchies of spaces are reused and adapted so that efficient algebraic multilevel preconditioners are quickly constructed for nearby problems. The main considered application is the solution of linear systems arising in Markov chain Monte Carlo simulations of subsurface flow with uncertainty in the conductivity field. A set of numerical experiments demonstrate the efficiency of the method for the target application. Copyright © 0000 John Wiley & Sons, Ltd.

Received ...

KEY WORDS: iterative methods; multigrid; algebraic multigrid; adaptive algebraic multigrid; Monte Carlo; Markov chain Monte Carlo; subsurface flow

1. INTRODUCTION

In this paper we treat the construction and adaptation of algebraic multigrid iterative methods (or preconditioners) for solving sequences of linear systems originating from finite-element discretization of elliptic partial differential equations (PDEs) where the matrices change gradually throughout the sequence. Such sequences arise in the iterative solution of nonlinear PDE [1], topology optimization in material design [2], and the quantification of uncertainty via the Markov chain Monte Carlo (MCMC) method [3].

We concentrate particularly on the case of uncertainty quantification in subsurface flow simulation when geological models of the hydraulic conductivity are conditioned to observed data [4]. The MCMC method is used to sample from the conditional distribution. This requires the numerical solution of thousands of PDEs with gradually changing diffusion coefficient realizations varying several orders of magnitude within small subdomains. Furthermore, the solver must handle higher-order discretizations and anisotropic media. The linear system solves are the most computationally intensive part of such simulations [4]. Thus, robust and efficient solvers are crucial to carrying out efficient simulations.

The proposed method attempts to reuse a multilevel hierarchy constructed for a previous linear system by quickly adapting it so that it would be efficient for solving a system with a nearby matrix. It is founded upon the element-based algebraic multigrid (AMGe) method [5, 6] in the sense that

*Correspondence to: Panayot S. Vassilevski, Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, 7000 East avenue, Mail-Stop L-560, Livermore, CA 94551, U.S.A., E-mail: panayot@llnl.gov

Contract/grant sponsor: U.S. Department of Energy; contract/grant number: DE-AC52-07NA27344

the proposed method uses local problems to obtain coarse basis that capture local features of the problem. The method also utilizes ideas from smoothed aggregation (SA) multigrid [7] to developed hierarchy of spaces that has good approximation properties independent of the coefficient contrast [8]. Additionally, it incorporates the adaptive AMG methodology (see [9, 10], and in AMGe context see [11, 12, 13]) to automatically identify the modes that should be included into the new solver to produce accurate enough coarse spaces. A two-grid version of the proposed method and efficient adaptation strategies are developed in [8, 14]. In this paper we use the so-called minimal intersection sets [6, 15] to define coarse degrees of freedom (dofs) and element matrices which allows for the multilevel extension of the method.

The remainder of the paper is organized as follows: In Section 2 we describe the model problem we are solving, the finite-element discretization, and the basics of MCMC subsurface flow simulation. In Section 3 we present the algebraic multigrid method and the adaptive approach that we apply to efficiently solve the sequence of equations. In Section 4 we present a numerical experiments based on MCMC simulations showing the efficacy of the proposed method. Finally, we make some concluding remarks in Section 5.

2. BACKGROUND

2.1. Model Problems and Discretization

A simple model problem close to subsurface flow is the single-phase steady-state flow equations

$$\begin{aligned} \mathbf{q}(\mathbf{x},\omega) + k(\mathbf{x},\omega)\nabla p(\mathbf{x},\omega) &= \mathbf{g}(\mathbf{x}) \text{ in } D\times\Omega, \\ \nabla\cdot\mathbf{q}(\mathbf{x},\omega) &= f(\mathbf{x}) \text{ in } D\times\Omega, \end{aligned} \tag{1}$$

subject to suitable boundary conditions. Here, $\mathbf{q}$ is the Darcy flux, k is the hydraulic conductivity field, and p is the pressure head. The stochasticity in the PDE coefficient is denoted by ω that, in uncertainty quantification of subsurface flow, may represent a random variable on a probability space Ω which describes the geological properties of the subsurface. Notice that since p and $\mathbf{q}$ depend on k, they also depend on ω.

Re-writing (1) as a 2^{nd}-order elliptic PDE, and adding boundary conditions, we have

$$\begin{aligned} -\text{div}\left[k(\mathbf{x},\omega)\nabla p(\mathbf{x},\omega)\right] &= f(\mathbf{x}) - \nabla\cdot\mathbf{g}(\mathbf{x}) \text{ in } D\times\Omega \\ p(\mathbf{x}) &= g_{_D}(\mathbf{x}) && \text{on } \partial D_D \\ \nabla p(\mathbf{x})\cdot\mathbf{n} &= g_{_N}(\mathbf{x}) && \text{on } \partial D_N. \end{aligned} \tag{2}$$

The proposed method is also applicable when the source term and boundary conditions depend on ω.

We are particularly interested in highly anisotropic conductivity fields. In this case, $k(\mathbf{x},\omega)$ can be written as a d-dimensional tensor, i.e. for $d=2$ we have

$$k(\mathbf{x},\omega) = \begin{bmatrix} k_{11}(\mathbf{x},\omega) & k_{12}(\mathbf{x},\omega) \\ k_{21}(\mathbf{x},\omega) & k_{22}(\mathbf{x},\omega) \end{bmatrix}.$$

Consider the polygonal domain $D\subset\mathbb{R}^2$. Let $\mathcal{T}_h$ be a quasi-uniform triangulation of D with mesh size h and collection of elements $\{\tau\}$. Let $\mathcal{V}_h$ be the finite element space associated with Lagrangian dofs $\mathcal{N}_h$. We assume that each realization of conductivity field is represented by piecewise-polynomial functions of degree one less than the degree of the polynomial basis in $\mathcal{V}_h$. For example, if we take $\mathcal{V}_h$ to be the space of piecewise linear functions over $\mathcal{T}_h$, then $k(\mathbf{x},\omega)$ is represented by piecewise constants over $\mathcal{T}_h$.

For fixed ω, the bilinear form corresponding to (2) is

$$a(u,v) = \int_D k(\mathbf{x},\omega)\nabla u\cdot\nabla v\, d\mathbf{x}.$$

DOI: 10.1002/nla

Denote by φ_j the basis function corresponding to dof j. Define the element stiffness matrix for τ as

$$[A_\tau]_{ij} = \int_\tau k(\mathbf{x},\omega)\,\nabla\varphi_i \cdot \nabla\varphi_j\, d\mathbf{x}, \forall i,j \in \tau.$$

Finally, the global fine-grid stiffness matrix A can be defined and assembled according to

$$u^T A v = \sum_{\tau\in\mathcal{T}_h} u_\tau^T A_\tau v_\tau,$$

where u_τ is the restriction of global finite-element function u to τ.

2.2. *Classical and Markov Chain Monte Carlo Simulations*

The goal of subsurface simulation is to compute moments of some quantity of interest given reasonable assumptions on the uncertainty in the data. Suppose that k has probability distribution $\pi(k)$ and we wish to compute moments of some quantity of interest Q which, for the model problem, is a function of k and p. The mean of Q can be approximated using a Monte Carlo estimator

$$\widehat{Q}_{MC} = \frac{1}{N}\sum_{i=1}^{N} Q^{(i)},$$

by drawing samples $k^{(i)}$ from $\pi(k)$, obtaining corresponding solutions $p^{(i)}$, and computing $Q^{(i)} \equiv Q^{(i)}\left(p^{(i)};k^{(i)}\right)$.

A popular choice is to model k as a log-normal random field with two-point correlation structure [16]. Then, k is expanded in a so-called *truncated Karhunen-Loève Expansion (KLE)*

$$k(\mathbf{x},\omega) = \exp\left[Y_M(\mathbf{x},\omega)\right], \tag{3}$$

where

$$Y_M(\mathbf{x},\omega) = Y_0(\mathbf{x}) + \sum_{j=1}^{M}\sqrt{\lambda_j}\phi_j(\mathbf{x})\,\xi_j(\omega). \tag{4}$$

Here, M is the number of terms in the expansion, $Y_0(\mathbf{x})$ is the mean, ξ_j are independent standard normal random variables, and (λ_j,ϕ_j) are eigenpairs of the integral equation with *covariance function* $C(\mathbf{x},\mathbf{x}')$ as a kernel. Take

$$C(\mathbf{x},\mathbf{x}') = \sigma_k^2 \exp\left(-\frac{|x_1-x_1'|}{\gamma_1} - \frac{|x_2-x_2'|}{\gamma_2}\right), \tag{5}$$

where σ_k^2 is the variance of the stochastic field and γ_j is the correlation length in the j^{th} direction. For this choice there exist analytic expressions for (λ_j,ϕ_j) [16]. Then, to draw a random sample $k^{(i)}$ only requires generating M independent random variables $\xi_j \sim \mathcal{N}(0,1)$ and computing $k(\mathbf{x},\omega)$. Having $k^{(i)}$, the remaining steps to obtain $Q^{(i)}$ are familiar.

However, when $\pi(k)$ is complex or not given explicitly (e.g. incorporating dynamic data like observed pressure values or flow rates), computing $\widehat{Q}_{MC}$ is not straight-forward. In such cases MCMC methods are necessary since they give a feasible way of sampling from such distributions.

Suppose that the subsurface is modeled by a simple (e.g. log-normal) *prior* distribution $P(k)$ and we have some observed data F, accurate up to some level of measurement error. Then, we wish to draw samples of k from the prior distribution *conditioned* to the observed data, i.e. from the *posterior* distribution $\pi(k) = P(k|F)$. Bayes' Law gives

$$P(k|F) \propto P(F|k)\,P(k),$$

where $P(F|k)$ is the likelihood that data F is observed given k. Finally, we model the likelihood function as a Gaussian

DOI: 10.1002/nla

$$P\left(F|k\right) \propto \exp\left(-\frac{\|F-F_k\|^2}{\sigma_f^2}\right),$$

where σ_f^2 is the likelihood variance and F_k is the model response obtained by solving (2) with k as a fixed coefficient.

In this paper we use the standard Metropolis-Hastings MCMC method [3] shown in Algorithm 2.1. Notice that we require a transition probability $q\left(k'|k\right)$ defining a transition from field k to k'.

Algorithm 2.1 Metropolis-Hastings MCMC

PROCEDURE: $k_{n+1} \leftarrow \mathbf{MH}\left(k_n\right)$.
INPUT: Current conductivity sample k_n.
OUTPUT: New conductivity sample k_{n+1}.
Generate proposal k' from transition probability $q\left(k'|k_n\right)$ and compute acceptance probability

$$\alpha\left(k_n,k'\right) = \min\left(1, \frac{\pi\left(k'\right)q\left(k_n|k'\right)}{\pi\left(k_n\right)q\left(k'|k_n\right)}\right).$$

Take $k_{n+1}=k'$ with probability $\alpha\left(k_n,k'\right)$, and $k_{n+1}=k_n$ with probability $1-\alpha\left(k_n,k'\right)$.

It is well-known [3] that after sufficiently long *burn-in*, samples $\{k_n\}$ come from the desired distribution $\pi\left(k\right)$. They are then used in a Monte Carlo estimator. Note again that each evaluation of $\pi\left(k\right)$ requires solution of (2) which is the main computational hardship in the simulation.

The transition probability that we consider is the *random walker*. Namely, each random coefficient in the KLE of $k\left(\mathbf{x},\omega\right)$ is perturbed by an independent random variable from $\mathcal{N}\left(0,\delta_k^2\right)$, where δ_k is the step-size parameter. That is, for $j=1,\ldots,M$ draw $\eta_j \sim \mathcal{N}\left(0,\delta_k^2\right)$ and set $\xi_j' = \xi_j + \eta_j$. The choice of δ_k affects the acceptance rate in Algorithm 2.1. Namely, increasing δ_k results in decreased acceptance and vice versa.

3. AMG FOR SEQUENCES OF LINEAR SYSTEMS

The foundation of the proposed adaptive method is the algorithm introduced in [8] (see also [17]), which combines the spectral AMGe (ρAMGe) [15] and SA-AMG [7] methods. In this section we give an algorithmic description, present the extensions we use to build a multilevel version of the method, and discuss the adaptive methodology.

3.1. Multilevel SA-ρAMGe

Multigrid methods are popular due to their potential to solve $N\times N$ sparse linear systems like $A\mathbf{x}=\mathbf{b}$ in $\mathcal{O}(N)$ time and space. Their efficiency is due to the synergy of *relaxation* and *coarse-grid correction* [1]. A general two-grid cycle is given in Algorithm 3.1 which provides the s.p.d. mapping $\mathbf{b}\mapsto B_{TG}^{-1}\mathbf{b}$ with $\mathbf{x}_0=0$. We generally choose relaxation which is A-convergent (i.e. $\|I-A^{1/2}M^{-1}A^{1/2}\|<1$) and inexpensive to compute. Applying Algorithm 3.1 recursively for the coarse problems results in the well-known V-cycle.

Consider the collection of elements $\mathcal{T}=\{\tau\}$, where each element is a set of small number of dofs and their union covers $\mathcal{N}$ – the set of all dofs. Note that elements are overlapping in terms of dofs. We assume that non-overlapping partitioning of the set $\mathcal{T}$ in *agglomerated elements (AEs)* $\{T\}$ has been constructed, where each T is a (connected) union of elements τ. Clearly, AEs are also overlapping in terms of dofs and their union covers $\mathcal{N}$. In addition, the set of dofs $\mathcal{N}$ is partitioned in a corresponding number of non-overlapping *aggregates* $\{\mathcal{A}\}$ such that each $\mathcal{A}$ is contained in a unique agglomerate T. This is accomplished by arbitrating dofs lying on the interface between AEs. Namely, we assign a shared dof to the neighboring aggregate to which it is most *strongly connected*.

DOI: 10.1002/nla

Algorithm 3.1 Two-Level (TL) Algorithm

PROCEDURE: $\mathbf{x}_{TG} \leftarrow \mathbf{TL}\left(A, \mathbf{b}, \mathbf{x}_0, M, P\right)$
INPUT: Matrix A, vector $\mathbf{b}$, initial iterate $\mathbf{x}_0$, relaxation operator M, interpolation operator P.
OUTPUT: Two-grid iterate $\mathbf{x}_{\mathrm{TG}} = \mathbf{x}$.
Initialize: $\mathbf{x} = \mathbf{x}_0$.
Pre-relax: $\mathbf{x} \leftarrow \mathbf{x} + M^{-1}\left(\mathbf{b} - A\mathbf{x}\right)$.
Correct: $\mathbf{x} \leftarrow \mathbf{x} + P\left(P^T A P\right)^{-1} P^T\left(\mathbf{b} - A\mathbf{x}\right)$.
Post-relax: $\mathbf{x} \leftarrow \mathbf{x} + M^{-T}\left(\mathbf{b} - A\mathbf{x}\right)$.

More precisely, let $a_{ij} = [A]_{ij}$ and for given dof i and aggregate $\mathcal{A}$ compute

$$\sigma_i(\mathcal{A}) = \max_{j \in \mathcal{A}} \left(\frac{|a_{ij}|}{\sqrt{a_{ii} a_{jj}}} \right).$$

Then, assign i to the candidate aggregate $\mathcal{A}$ with largest $\sigma_i(\mathcal{A})$.

Furthermore, $\mathcal{N}$ is partitioned in *minimal intersection sets* $\{\mathcal{I}\}$ [6, 15]. By definition, the minimal intersection sets are equivalence classes defined by the relation that two dofs as equivalent if they belong to identical sets of AEs. Then, intersecting the aggregates $\{\mathcal{A}\}$ with the sets $\{\mathcal{I}\}$ gives the partitioning of $\mathcal{N}$ in *refined aggregates* $\{\alpha\}$. Appropriately selected collections of these refined aggregates cover exactly each agglomerate T and each (big) aggregate $\mathcal{A}$.

Assume that we have the element stiffness matrices for the elements in $\mathcal{T}$. Thus, for each T we assemble the local stiffness matrix A_T. Consider the Schur complement $S_{\mathcal{A}}$ based on A_T and corresponding to the dofs of the aggregate $\mathcal{A} \subset T$. We solve the generalized eigenvalue problem

$$S_{\mathcal{A}} \mathbf{q}_k = \lambda_k D_{\mathcal{A}} \mathbf{q}_k, \ k = 1, \ldots, n_{\mathcal{A}},$$

where $D_{\mathcal{A}}$ is a diagonal matrix and $n_{\mathcal{A}}$ is the size of $\mathcal{A}$. Assuming that the eigenpairs are sorted in ascending order according to their eigenvalues, for a prescribed tolerance θ, we select the first $m_{\mathcal{A}} \leq n_{\mathcal{A}}$ eigenvectors such that $\lambda_k \leq \theta \lambda_{n_{\mathcal{A}}}$ for all $k \leq m_{\mathcal{A}}$. In practice, we take $D_{\mathcal{A}}$ to be the restriction to $\mathcal{A}$ of the weighted ℓ_1-smoother D_T corresponding to A_T. That is, $D_T = \mathrm{diag}\left(d_i\right)$, where $d_i = \sum_j |a_{ij}| \sqrt{a_{ii}/a_{jj}}$ and $a_{ij} = [A_T]_{ij}$. With this choice we are guaranteed that $\lambda_{n_{\mathcal{A}}} \leq 1$.

Having computed the eigenvectors, we describe two possible ways to proceed. The first way is to simply take the local tentative interpolant $\widehat{P}_{\mathcal{A}} = [\mathbf{q}_1, \ldots, \mathbf{q}_{m_{\mathcal{A}}}]$. Note that this approach does not make use of the refined aggregates. The second method allows the construction of multilevel hierarchies. Namely, for every $\alpha \subset \mathcal{A}$ take the restriction of the eigenvectors on α, i.e. $Q_\alpha = [\mathbf{q}_1|_\alpha, \ldots, \mathbf{q}_{m_{\mathcal{A}}}|_\alpha]$. Then, by othogonalizing the columns of Q_α, we arrive at the local tentative interpolant $\widehat{P}_\alpha$ whose $m_\alpha \leq m_{\mathcal{A}}$ columns span the column space of Q_α. Finally, depending on the choice of local tentative interpolant constructions described above, we build one of the following global tentative interpolants

$$\widehat{P}_{\{\mathcal{A}\}} = \begin{bmatrix} \widehat{P}_{\mathcal{A}_1} & 0 & & 0 \\ 0 & \widehat{P}_{\mathcal{A}_2} & & 0 \\ & & \ddots & \vdots \\ 0 & 0 & \ldots & \widehat{P}_{\mathcal{A}_{n_c}} \end{bmatrix}, \quad \widehat{P}_{\{\alpha\}} = \begin{bmatrix} \widehat{P}_{\alpha_1} & 0 & & 0 \\ 0 & \widehat{P}_{\alpha_2} & & 0 \\ & & \ddots & \vdots \\ 0 & 0 & \ldots & \widehat{P}_{\alpha_{N_c}} \end{bmatrix}.$$

Denote $\widehat{P} = \{\widehat{P}_{\{\mathcal{A}\}} \text{ or } \widehat{P}_{\{\alpha\}}\}$. In order to make the interpolation operator P stable in energy norm, $\widehat{P}$ is smoothed by an appropriate polynomial smoother. That is, $P = S\widehat{P}$, where matrix S is a polynomial in $D^{-1}A$ and D is a diagonal matrix spectrally equivalent to the diagonal of A. Particularly, we use $S = s_{\nu_p}\left(b^{-1} D^{-1} A\right)$, where $\|D^{-1/2} A D^{-1/2}\| \leq b = \mathcal{O}\left(1\right)$ and

$$s_\nu\left(t\right) = \left(-1\right)^\nu \frac{1}{2\nu + 1} \frac{T_{2\nu+1}\left(\sqrt{t}\right)}{\sqrt{t}},$$

Prepared using nlaauth.cls
DOI: 10.1002/nla

is the well-known smoothed aggregation polynomial [13]. Here, $T_l(t)$ is the degree-l first-kind Chebyshev polynomial over $[-1,+1]$. For relaxation we use $I - M^{-1}A = p_{\nu_r}\left(b^{-1}D^{-1}A\right)$ where

$$p_\nu(t) = \left(1 - T^2_{2\nu+1}(\sqrt{t})\right) s_\nu(t).$$

In practice, we choose D to be the weighted ℓ_1-smoother corresponding to A which allows us to take $b = 1$.

Clearly, for identical sets of eigenvectors, $\mathrm{range}(\widehat{P}_{\{\mathcal{A}\}}) \subset \mathrm{range}(\widehat{P}_{\{\alpha\}})$ and hence the results in [8] hold for both choices of tentative interpolants above. Thus, using $\widehat{P}_{\{\alpha\}}$ results in larger coarse spaces than using $\widehat{P}_{\{\mathcal{A}\}}$ but the former allows for multilevel hierarchies. However, it is still practical to apply the latter for building the coarsest level in the hierarchy.

To produce multiple levels we want to apply the described method recursively. Although it is easy to see that agglomerates are appropriate choice for coarse elements, we still need to adequately define coarse dofs, coarse elements in terms of coarse dofs, and their corresponding coarse element matrices. These ingredients are sufficient to apply the above approach recursively and thus build multilevel preconditioners. To this purpose we use $\widehat{P} = \widehat{P}_{\{\alpha\}}$ and identify the *coarse dofs* with the columns of $\widehat{P}$. Then, for any agglomerate T, its coarse dofs are all columns of $\widehat{P}$ coming from all refined aggregates $\alpha \subset T$. Consider the agglomerate interpolant $\widehat{P}_T = [\widehat{P}_\alpha]_{\alpha\subset T}$ where the columns of $\widehat{P}_\alpha$ are extended by zero on $T \setminus \alpha$. This allows us to define *coarse element matrices* based on agglomerate stiffness matrices A_T. Namely, for coarse element $\tau := T$, its coarse element stiffness matrix is given by the product

$$A_\tau = (\widehat{P}_T)^T A_T \widehat{P}_T.$$

Having all the pieces, we invoke the above procedure to recursively build $P_0, \ldots, P_{l-1}$, where the finest grid is defined to be level 0.

3.2. Adaptive SA-ρAMGe

An extensive description of the adaptive strategies that we use is given in [14]. Here we present a concise discussion of the methodology applied to the first coarse level in the hierarchy. That is, only the first interpolant P_0 is adapted while the rest of the hierarchy is always rebuilt from scratch. Thus, we save substantial fine-grid work which is the most expensive component of the hierarchy construction.

Assume that we have a multilevel preconditioner B built as described in Section 3.1 for matrix A and that a new nearby matrix A' is given. We want to quickly produce preconditioner B' efficient for solving $A'\mathbf{x} = \mathbf{b}$. Using A' and updated M in Algorithm 3.1 (as a recursive V-cycle) we perform ν_a iterations

$$\mathbf{x}_k = \left(I - B^{-1}A'\right)\mathbf{x}_{k-1}, \tag{6}$$

with random $\mathbf{x}_0$, and we monitor the convergence $\|\mathbf{x}_k\|_{A'} \to 0$. If the convergence rate is better than a prescribed ρ^{target}, then we use $B' = B$ as the new preconditioner. Otherwise, we set $\mathbf{x}^{\text{bad}} = \mathbf{x}_{\nu_a}$ and take its restriction $\mathbf{x}^{\text{bad}}_{\mathcal{A}} = \mathbf{x}^{\text{bad}}|_{\mathcal{A}}$ on each aggregate $\mathcal{A}$. Here, $\mathbf{x}^{\text{bad}}$ exposes the algebraically smooth components of the error on the finest grid that are not efficiently damped by the coarse-grid correction. Then, we solve the modified generalized eigenvalue problem for each $\mathcal{A}$

$$S'_{\mathcal{A}}\mathbf{z}_k = \mu_k D'_{\mathcal{A}}\mathbf{z}_k,\ k = 1, \ldots, n'_{\mathcal{A}} \le m_{\mathcal{A}} + 1, \tag{7}$$

in the subspace spanned by $\{\mathbf{q}_1, \ldots, \mathbf{q}_{m_{\mathcal{A}}}, \mathbf{x}^{\text{bad}}_{\mathcal{A}}\}$, where $\{\mathbf{q}_1, \ldots, \mathbf{q}_{m_{\mathcal{A}}}\}$ is the local basis used for building the current P_0. Next, as above, we consider the lower part of the spectrum, i.e. we select the first $m'_{\mathcal{A}} \le n'_{\mathcal{A}} \le m_{\mathcal{A}} + 1$ eigenvectors such that $\mu_k \le \theta\|(D'_{\mathcal{A}})^{-1/2}S'_{\mathcal{A}}(D'_{\mathcal{A}})^{-1/2}\|$ for all $k \le m'_{\mathcal{A}}$. Note that (7) is solved in a small subspace which is much faster than solving the full local problem necessary when building P_0 from scratch.

After obtaining all eigenvectors, we proceed as before. That is, we assemble the global tentative interpolation operator and smooth it to produce the adapted interpolant P'_0. Then, using the

DOI: 10.1002/nla

procedure in Section 3.1 we rebuild the rest of the hierarchy from scratch, i.e. $P'_1, \ldots, P'_{l-1}$, and in the end we obtain the adapted preconditioner B'. The procedure is repeated, starting from (6), until the desired convergence determined by ρ^{target} is achieved. If the convergence fails to improve by some factor (e.g. 0.9) during the adaptation, we increase the tolerance θ to get more local vectors from $\mathbf{x}^{\text{bad}}$ in the coarse space. Namely, we take

$$\theta' = \frac{1}{n_c} \sum_{i=1}^{n_c} \mu_{n'_{\mathcal{A}_i}}^{(\mathcal{A}_i)},$$

where n_c is the number of aggregates. It is precisely the average of the maximal eigenvalues in problems (7).

Additionally, we use an adaptive approach to choose the initial tolerance θ for every linear system in the sequence. Denote by θ_i $(i = 1, \ldots, n)$ the final value of θ used for the i^{th} linear system. Then, if adaptation is needed for the $(n+1)^{\text{st}}$ linear system it is invoked with initial spectral tolerance

$$\theta_{n+1} = \kappa \theta_n + (1 - \kappa) \frac{1}{n-1} \sum_{i=1}^{n-1} \theta_i,$$

where $\kappa \in [0, 1]$ is a tuning parameter.

As explained, our final goal is to solve large sequences of linear systems where consecutive matrices are in a sense similar. In such case, instead of starting with convergence check (6) for each system by running the stationary iteration for $A\mathbf{x} = \mathbf{0}$, it is more practical to attempt to solve the linear system with the desired right hand side, i.e. $A\mathbf{x} = \mathbf{b}$. In the case that the method does not return a solution for a prescribed number of iterations, or the reduction factor of the residual is not satisfactory, only then the adaptation procedure is invoked to improve the method until the desired ρ^{target} is reached. Otherwise, the solution is obtained and used in the MCMC simulation. Furthermore, for difficult problems it may be more beneficial to execute a predefined number of adaptation steps per linear system whenever adaptation is necessary, instead of requiring convergence better than ρ^{target} regardless of the adaptivity cost.

4. NUMERICAL EXPERIMENTS

We apply the proposed method for solving sequences of linear systems coming from MCMC simulations. The model problem we consider is given in (8).

$$\begin{aligned} -\text{div}\left[k\left(\mathbf{x}, \omega\right) \nabla p\left(\mathbf{x}, \omega\right)\right] &= 0 \qquad \text{in } \left[0,1\right]^2 \times \Omega \\ p\left(\mathbf{x}\right) &= 1 - x_1 \text{ on } x_1 = \{0,1\} \\ \nabla p\left(\mathbf{x}\right) \cdot \mathbf{n} &= 0 \qquad \text{on } x_2 = \{0,1\} \end{aligned} \tag{8}$$

We use three different forms of k: corresponding to scalar, grid-aligned anisotropic, and non-grid-aligned anisotropic conductivity tensors. Also, linear finite-element discretizations on a uniform triangular mesh are used. In all cases, we assume that the observed data F comes from measurements of the pressure head at several fixed points in the domain. As described in Section 2.2, $Y_M\left(\mathbf{x}, \omega\right)$ (see (4)) is a Gaussian field with two-point correlation structure (5) expanded in a M-term truncated KLE. For the experiments, we choose $M = 1000$, $Y_0 \equiv 0$, $\sigma_k^2 = 3$, and $\gamma_1 = \gamma_2 = 0.1$. In the case of scalar conductivity, k is modeled as a log-normal random field (see (3)). For the parameters used, two sample realizations of Y_M are shown in Figure 1. It is visible that conductivity realizations vary as many as six orders of magnitude over small length scales.

When anisotropic conductivity is used, the problem becomes significantly more difficult. Let $Y_{M,x} \equiv Y_{M,x}\left(\mathbf{x}, \omega\right)$ and $Y_{M,y} \equiv Y_{M,y}\left(\mathbf{x}, \omega\right)$ be two independent instances of the Gaussian process described above. Then, the grid-aligned and non-grid-aligned anisotropic conductivity tensors are respectively given as follows

$$k\left(\mathbf{x}, \omega\right) = \begin{bmatrix} Y_{M,x}^2 + \epsilon & 0 \\ 0 & Y_{M,y}^2 + \epsilon \end{bmatrix}, \quad k\left(\mathbf{x}, \omega\right) = \begin{bmatrix} Y_{M,x}^2 + \epsilon & Y_{M,y} Y_{M,x} \\ Y_{M,x} Y_{M,y} & Y_{M,y}^2 + \epsilon \end{bmatrix},$$

Prepared using nlaauth.cls
DOI: 10.1002/nla

where ϵ is a small parameter.

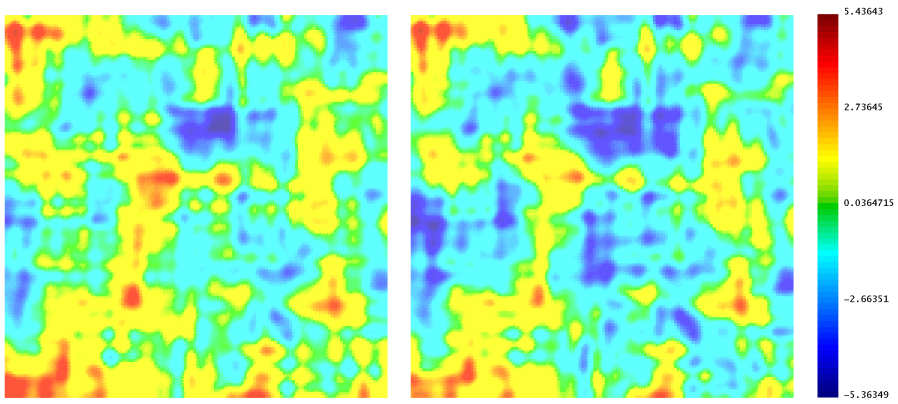

Figure 1. Two realizations of $Y_M(\mathbf{x}, \omega)$.

Table I. Average results for simulations with **scalar** conductivity field.

n_f	$\overline{\rho}$	$\overline{C_A}$	N_r	$\overline{N_b}$	L
66049	0.82 (0.38)	1.26	0.0%	NA	3
263169	0.88 (0.46)	1.28	0.3%	1.0	4

Table II. Average results for simulations with **grid-aligned anisotropic** conductivity field.

n_f	$\overline{\rho}$	$\overline{C_A}$	N_r	$\overline{N_b}$	L
66049	0.90 (0.50)	1.34	0.8%	3.0	3
263169	0.91 (0.51)	1.77	2.8%	3.0	4

Table III. Average results for simulations with **non-grid-aligned anisotropic** conductivity field.

n_f	$\overline{\rho}$	$\overline{C_A}$	N_r	$\overline{N_b}$	L
66049	0.91 (0.49)	1.97	10.8%	3.0	3
263169	0.92 (0.53)	1.91	36.4%	3.0	3

In all experiments, ρ^{target} is 0.9 and δ_k is tuned so that $20-25\%$ acceptance rate is observed in the MCMC algorithm. For each test case, we run several simulations for a total number of 5000 MCMC iterations. We report: the number of dofs on the finest grid (n_f); the number of levels (L); the average convergence factor ($\overline{\rho}$) for the stationary iteration and the conjugate gradient method (given in parentheses) preconditioned by the proposed multilevel preconditioner; the percentage of configurations for which adaptation is necessary (N_r); the average number of adaptive cycles performed when adaptation is necessary ($\overline{N_b}$); and the average operator complexity ($\overline{C_A}$), where

$$C_A = \frac{1}{\mathrm{nnz}\,(A_0)} \sum_{i=0}^{l} \mathrm{nnz}\,(A_i)\,.$$

Here, l is the coarsest level and A_i is the operator on level i. In the anisotropic cases, we set a fixed number of adaptive cycles to be performed when adaptation is necessary. Results of these numerical experiments are displayed in Tables I-III.

For some difficult problems (e.g. for non-grid-aligned anisotropic conductivity field) we see notable increase in the number of configurations for which adaptation is necessary with (finest) mesh

DOI: 10.1002/nla

refinement. Note that the proposed adaptive approach is expected to be efficient when consecutive coefficient realizations are similar in a local sense, i.e. similarities are observed within agglomerates. Nevertheless, in our experiments, as we refine the finest mesh we keep M, δ_k, and $\frac{H}{h}$ fixed, where H is the diameter of the agglomerated elements on the finest mesh. Thus, the first coarse level also gets refined and while coefficient realizations seem globally similar, they become less locally similar with mesh refinement. Clearly, we can expect to preserve local similarity as h is decreased by appropriately increasing the number of KL modes, M, and accordingly decreasing δ_k. Here, decreasing δ_k is quite natural as we want to keep the acceptance rate within some desired bounds. However, even when adaptation is frequently necessary, the adaptive method may still be applicable depending on the cost of building the hierarchy from scratch.

5. CONCLUSION

In conclusion, we have described a multilevel AMG method that utilizes hierarchy adaptation to efficiently solve sequences of linear systems coming from finite-element discretizations of elliptic PDEs with gradually changing conductivity coefficients. The method constructs robust hierarchies that provide effective solvers for many linear systems where the conductivity field is highly heterogeneous or anisotropic. We particularly considered the application of the presented approach in Markov chain Monte Carlo simulation of steady-state subsurface flow. In this context we demonstrated by performing numerical experiments the potential of the proposed method for constructing efficient preconditioners for the arising sequences of linear systems.

ACKNOWLEDGEMENTS

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

REFERENCES

1. Briggs W, Henson V, McCormick S. *Multigrid Tutorial*. SIAM, 2000.
2. Lazarov B, Matzen R, Sigmund O. Efficient topology optimization in matlab using 88 lines of code. *Structural and Multidisciplinary Optimization* 2011; **43(1)**:1–16.
3. Robert C, Casella G. *Monte Carlo Statistical Methods, 2nd Ed.* Springer: New York, NY, 2004.
4. Efendiev Y, Datta-gupta A, Ginting V, Ma X, Mallick B. An efficient two-stage markov chain monte carlo method for dynamic data integration. *Water Resources Research* 2005; **41, W12423**.
5. Brezina M, Cleary A, Falgout R, Henson V, Jones J, Manteuffel T, McCormick S, Ruge J. Algebraic multigrid based on element interplation(AMGe). *SIAM J. Sci. Comp* 2000; **22**:1570–1592.
6. Jones J, Vassilevski P. AMGe based on element agglomeration. *SIAM J. Sci. Comput.* 2001; **23(1)**:109–133.
7. Vaněk P, Mandel J, Brezina M. Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. *Computing* 1996; **56**:179–196.
8. Brezina M, Vassilevski P. Smoothed aggregation spectral element agglomeration AMG: $SA-\rho$AMGe. *"Large-Scale Scientific Computing, 8th International Conference, LSSC 2011, Sozopol, Bulgaria, June 6-10th, 2011. Revised Selected Papers". Lecture Notes in Computer Science*, vol. 7116, Springer, 2012; 3–15.
9. Brezina M, Falgout R, MacLachlan S, Manteuffel T, McCormick S, Ruge J. Adaptive smoothed aggregation (αSA) multigrid. *SIAM Rev.* 2005; **47(2)**:317–346.
10. Brezina M, Falgout R, MacLachlan S, Manteuffel T, McCormick S, Ruge J. Adaptive algebraic multigrid. *SIAM J. Sci. Comp.* 2006; **27(4)**:1261–1286.
11. Vassilevski P, Zikatanov L. Multiple vector preserving interpolation mappings in algebraic multigrid. *SIAM Journal on Matrix Analysis and Applications* 2005-2006; **27**:1040–1055.
12. Lashuk I, Vassilevski PS. On some versions of the element agglomeration AMGe method. *Numerical Linear Algebra with Applications* 2008; **15**:595–620.
13. Vassilevski P. *Multilevel Block Factorization Preconditioners, Matrix-based Analysis and Algorithms for Solving Finite Element Equations*. Springer: New York, NY, 2008.
14. Kalchev D, Ketelsen C, Vassilevski P. Adaptive algebraic multigrid for sequence of problems with slowly varying random coefficients. *Submitted* 2012; .
15. Chartier T, Falgout R, Henson VE, Jones J, Manteuffel T, McCormick S, Ruge J, Vassilevski PS. Spectral element agglomerate AMGe. *Domain Decomposition Methods in Science and Engineering XVI*, *Lecture Notes in Computational Science and Engineering*, vol. 55, Springer, 2007; 515–524.
16. Le Maître O, Knio O. *Spectral Methods for Uncertainty Quantification*. Springer: New York, NY, 2010.

DOI: 10.1002/nla

17. Brezina M, Heberton C, Mandel J, Vaněk P. An iterative method with convergence rate chosen a priori April 1999. An earlier version presented at the 1998 Copper Mountain Conference on Iterative Methods, April 1998.

DOI: 10.1002/nla