ACCELERATING MARKOV CHAIN MONTE CARLO WITH ACTIVE SUBSPACES[*]

PAUL G. CONSTANTINE[†], CARSON KENT[‡], AND TAN BUI-THANH[§]

Abstract. The Markov chain Monte Carlo (MCMC) method is the computational workhorse for Bayesian inverse problems. However, MCMC struggles in high-dimensional parameter spaces, since its iterates must sequentially explore the high-dimensional space. This struggle is compounded in physical applications when the nonlinear forward model is computationally expensive. One approach to accelerate MCMC is to reduce the dimension of the state space. *Active subspaces* are part of an emerging set of tools for subspace-based dimension reduction. An active subspace in a given inverse problem indicates a separation between a low-dimensional subspace that is informed by the data and its orthogonal complement that is constrained by the prior. With this information, one can run the sequential MCMC on the active variables while sampling independently according to the prior on the inactive variables. However, this approach to increase efficiency may introduce bias. We provide a bound on the Hellinger distance between the true posterior and its active subspace-exploiting approximation. And we demonstrate the active subspace-accelerated MCMC on two computational examples: (i) a two-dimensional parameter space with a quadratic forward model and one-dimensional active subspace and (ii) a 100-dimensional parameter space with a PDE-based forward model and a two-dimensional active subspace.

Key words. MCMC, active subspaces, dimension reduction

AMS subject classifications. 65C40, 65D15

DOI. 10.1137/15M1042127

1. Introduction.

Modern computing enables powerful Bayesian inference methods to quantify uncertainty in complex science and engineering models. The workhorse of these methods is Markov chain Monte Carlo (MCMC), which constructs a Markov chain whose steps produce correlated samples from the conditional posterior density of the parameters given data. Several excellent references introduce and develop the theory and practice of MCMC in the context of statistical inverse problems [3, 22, 27]. However, standard MCMC remains an inefficient tool when (i) the dimension of the parameter space is large and/or (ii) the forward model in the likelihood is expensive. Recent papers have proposed strategies to increase MCMC's efficiency by introducing structure-exploiting proposal densities [1, 20, 28, 18, 24, 5]. When the forward model numerically approximates a PDE solution, the differential operators may enable MCMC variants whose state space dimension is independent of the PDE discretization [4, 5, 24, 14].

[*] Submitted to the journal's Methods and Algorithms for Scientific Computing section October 1, 2015; accepted for publication (in revised form) July 1, 2016; published electronically September 1, 2016.

http://www.siam.org/journals/sisc/38-5/M104212.html

Funding: This work was supported by the U.S. Department of Energy Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under award DE-SC-0011077 and the Defense Advanced Research Projects Agency's Enabling Quantification of Uncertainty in Physical Systems. The first author's 2015 summer visit to the Institute for Computational Engineering and Sciences at UT Austin was supported by the J. Tinsley Oden Faculty Fellowship Research Program.

[†] Department of Applied Mathematics and Statistics, Colorado School of Mines, Golden, CO 80401 (paul.constantine@mines.edu).

[‡] Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305 (crkent@stanford.edu).

[§] Department of Aerospace Engineering and Engineering Mechanics, Institute for Computational Engineering and Sciences, University of Texas at Austin, Austin, TX 78705 (tanbui@ices.utexas.edu).

MCMC struggles in high dimensions because the Markov chain must explore the high-dimensional space to find regions of relatively high posterior probability. The recent paper by Cui et al. [14] seeks to reduce the Markov chain's state space dimension by identifying a low-dimensional *likelihood-informed subspace*. When an inverse problem is ill-posed, the data does not inform all parameters; Cui et al. use the eigendecomposition of the negative log-likelihood's Hessian—averaged over the parameter space with respect to the posterior—to identify a data-informed subspace; its orthogonal complement is informed by the prior. This separation leads to an efficient MCMC method when the forward model is twice differentiable, the prior is Gaussian, and the noise is Gaussian. They have extended these ideas to develop optimal posterior covariance approximations for linear forward models [26] and dimension-independent methods for infinite dimensional problems [13].

Our recent work in *active subspaces* resembles the subspace-based dimension reduction of Cui et al. [14], though the context is broader than statistical inverse problems. The active subspace is defined for a scalar-valued, multivariate function; the active subspace basis consists of the principal eigenvectors of the average outer product of the gradient with itself. When a function admits an active subspace, one can exploit the low-dimensional structure to enable approximation, integration, and optimization in high dimensions [9, 11, 8]. We have applied this technique to help study aerospace models [10, 23], hydrological models [21], and solar cell models [12]. We review active subspaces and the resulting approximations in section 2.

In this paper, we seek to discover the active subspace from the negative log-likelihood in a Bayesian inverse problem. If the negative log-likelihood admits an active subspace, then we use the active subspace to construct a function that approximates the Bayesian posterior. The approximate posterior enables an efficient MCMC that exploits the low-dimensional structure; the Markov chain operates on the active variables (i.e., the coordinates of the active subspace), while the inactive variables are drawn independently from their derived prior. In contrast to the likelihood-informed subspace of Cui et al. [14], the active subspace is defined by the gradient of the negative log-likelihood instead of a Gauss–Newton-based approximation of its Hessian. Additionally, the matrix that defines the active subspace is an average with respect to the prior as opposed to the posterior. This choice is computationally appealing, since one can draw independent samples from the prior and estimate integrals with standard Monte Carlo. Consequently, computing the active subspace is a preprocessing step that occurs before the MCMC.

In section 3, we analyze the approximation error in the posterior, measured by the Hellinger distance. We detail the MCMC algorithm with the active subspace in section 4. We demonstrate the approach in two synthetic Bayesian inverse problems: (i) a quadratic forward model with a two-dimensional parameter space and (ii) a forward model governed by Poisson's equation, where the data are the PDE solution—perturbed by noise—at points on the boundary, and the parameters are the coefficients of the differential operator. We describe the models and show results in section 5. In section 6, we summarize the method and discuss some practical limitations and future work.

2. Active subspaces.

The following description of active subspaces can be found in [9, 8]; we include this section to keep the paper reasonably self-contained. Let $f = f(\mathbf{x})$ be a function from $\mathbb{R}^m$ to $\mathbb{R}$; the input vector $\mathbf{x} \in \mathbb{R}^m$ has m independent components. Let $\rho : \mathbb{R} \to \mathbb{R}_+$ be a given probability density function. Assume that

$\rho = \rho(\mathbf{x})$ and $\mathbf{x}$ are such that

$$\int \mathbf{x}\,\rho\,d\mathbf{x} \;=\; 0, \qquad \int \mathbf{x}\,\mathbf{x}^T\,\rho\,d\mathbf{x} \;=\; \boldsymbol{I}, \tag{2.1}$$

where $\boldsymbol{I}$ is the $m \times m$ identity matrix.[1] Also assume that f is differentiable with gradient vector $\nabla f(\mathbf{x}) \in \mathbb{R}^m$, whose components are square-integrable with respect to ρ. Define the $m \times m$ symmetric, positive semidefinite matrix $\boldsymbol{C}$ and its eigenvalue decomposition as

$$\boldsymbol{C} \;=\; \int \nabla f(\mathbf{x})\,\nabla f(\mathbf{x})^T\,\rho(\mathbf{x})\,d\mathbf{x} \;=\; \boldsymbol{W}\Lambda\boldsymbol{W}^T, \tag{2.2}$$

where $\boldsymbol{W}$ is the orthogonal matrix of eigenvectors, and Λ is the diagonal matrix of eigenvalues ordered in descending order. The ith eigenvalue λ_i satisfies

$$\lambda_i \;=\; \mathbf{w}_i^T\,\boldsymbol{C}\,\mathbf{w}_i \;=\; \int \left(\mathbf{w}_i^T\nabla f(\mathbf{x})\right)^2 \rho(\mathbf{x})\,d\mathbf{x}. \tag{2.3}$$

In words, (2.3) means that the ith eigenvalue measures the average, squared directional derivative of f along the corresponding eigenvector $\mathbf{w}_i$. Thus, $\lambda_i = 0$ if and only if f is constant along the direction $\mathbf{w}_i$ in $\mathbb{R}^m$.

To define the active subspace, assume $\lambda_n > \lambda_{n+1}$ for some $n < m$ and partition the eigenvalues and eigenvectors as

$$\Lambda \;=\; \begin{bmatrix}\Lambda_1 & \\ & \Lambda_2\end{bmatrix}, \qquad \boldsymbol{W} \;=\; \begin{bmatrix}\boldsymbol{W}_1 & \boldsymbol{W}_2\end{bmatrix}, \tag{2.4}$$

where Λ_1 contains the first n eigenvalues, and the columns of $\boldsymbol{W}_1$ are the first n eigenvectors. The *active subspace* is the span of the columns of $\boldsymbol{W}_1$. However, the active subspace is not necessarily a subset of the domain of f—even when the domain of f is $\mathbb{R}^m$. Instead, the columns of $\boldsymbol{W}_1$ are a set of directions; perturbing $\mathbf{x}$ along these directions changes $f(\mathbf{x})$ more, on average, than perturbing $\mathbf{x}$ along the directions corresponding to the columns of $\boldsymbol{W}_2$. Any $\mathbf{x} \in \mathbb{R}^m$ can be written

$$\mathbf{x} \;=\; \boldsymbol{W}_1\boldsymbol{W}_1^T\mathbf{x} + \boldsymbol{W}_2\boldsymbol{W}_2^T\mathbf{x} \;=\; \boldsymbol{W}_1\mathbf{y} + \boldsymbol{W}_2\mathbf{z}, \tag{2.5}$$

where $\mathbf{y} = \boldsymbol{W}_1^T\mathbf{x}$ are the *active variables* and $\mathbf{z} = \boldsymbol{W}_2^T\mathbf{x}$ are the *inactive variables.* The density function ρ begets a joint density between the active and inactive variables,

$$\rho(\mathbf{x}) \;=\; \rho(\boldsymbol{W}_1\mathbf{y} + \boldsymbol{W}_2\mathbf{z}) \;=\; \rho(\mathbf{y}, \mathbf{z}), \tag{2.6}$$

which leads to marginal and conditional densities under the standard construction. If ρ is a standard Gaussian density on $\mathbf{x}$, then, due to the orthogonal columns of $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$, the marginal and conditional densities on $\mathbf{y}$ and $\mathbf{z}$ are standard Gaussian densities on $\mathbb{R}^n$ and $\mathbb{R}^{m-n}$, respectively.

2.1. Approximation with the active subspace.

If $\lambda_1, \ldots, \lambda_n$ are much larger than $\lambda_{n+1}, \ldots, \lambda_m$, then we can approximate f by a function of $n < m$ linear combinations of $\mathbf{x}$. To construct this approximation, define $g : \mathbb{R}^n \to \mathbb{R}$ by the conditional average of f given $\mathbf{y}$,

$$g(\mathbf{y}) \;=\; \int f(\boldsymbol{W}_1\mathbf{y} + \boldsymbol{W}_2\mathbf{z})\,\rho(\mathbf{z}|\mathbf{y})\,d\mathbf{z}, \tag{2.7}$$

[1] Any random vector with a full rank covariance can be shifted and linearly transformed to satisfy the zero-mean and identity covariance assumptions.

where $\rho(\mathbf{z}|\mathbf{y})$ is the conditional density of $\mathbf{z}$ given $\mathbf{y}$. Under this construction, we have the following bound on the root-mean-squared error of the approximation $f(\mathbf{x}) \approx g(\boldsymbol{W}_1^T\mathbf{x})$.

Theorem 2.1 (Theorem 3.1 in [9]).

$$\left(\int (f(\mathbf{x}) - g(\boldsymbol{W}_1^T\mathbf{x}))^2\,\rho\,d\mathbf{x}\right)^{1/2} \;\leq\; C\,(\lambda_{n+1} + \cdots + \lambda_m)^{1/2}, \tag{2.8}$$

where C is the Poincaré constant associated with the density ρ.

If ρ is a standard Gaussian density, then C in (2.8) is 1 [7]. The conditional expectation in (2.7) is not useful for computation, since evaluating $g(\mathbf{y})$ involves computing an $(n-m)$-dimensional integral. To move toward a useful computational tool, we introduce the Monte Carlo approximation, $\hat{g} \approx g$, defined as

$$\hat{g}(\mathbf{y}) \;=\; \frac{1}{M}\sum_{i=1}^{M} f(\boldsymbol{W}_1\mathbf{y} + \boldsymbol{W}_2\mathbf{z}_i), \tag{2.9}$$

where $\mathbf{z}_i$ are drawn independently according to the conditional density $\rho(\mathbf{z}|\mathbf{y})$. The approximation $f(\mathbf{x}) \approx \hat{g}(\boldsymbol{W}_1^T\mathbf{x})$ admits the following root-mean-squared error estimate.

Theorem 2.2 (Theorem 3.2 in [9]).

$$\left(\int (f(\mathbf{x}) - \hat{g}(\boldsymbol{W}_1^T\mathbf{x}))^2\,\rho\,d\mathbf{x}\right)^{1/2} \;\leq\; C(1 + M^{-1/2})(\lambda_{n+1} + \cdots + \lambda_m)^{1/2}, \tag{2.10}$$

where C is the constant from Theorem 2.1.

If f is such that the eigenvalues $\lambda_{n+1} = \cdots = \lambda_m = 0$, then the Monte Carlo estimate is exact for any number $M > 0$ of samples. Another way to see this is that $\lambda_{n+1} = \cdots = \lambda_m = 0$ implies f is constant along the directions corresponding to $\boldsymbol{W}_2$'s columns, and the average of a constant is the constant.

2.2. Computing the active subspace with Monte Carlo. We assume that the dimension m is sufficiently large that Monte Carlo is the most practical choice to estimate the matrix $\boldsymbol{C}$ in (2.2). (The two-parameter example in section 5.1 uses a tensor product Gauss–Hermite quadrature rule.) Our recent work analyzes the Monte Carlo approximation [11]. Let $\mathbf{x}_j$ with $j = 1, \ldots, N$ be drawn independently according to the density ρ. For each $\mathbf{x}_j$, compute the gradient $\nabla f_j = \nabla f(\mathbf{x}_j)$. Then approximate

$$\boldsymbol{C} \;\approx\; \hat{\boldsymbol{C}} \;=\; \frac{1}{N}\sum_{j=1}^{N} \nabla f_j\,\nabla f_j^T \;=\; \hat{\boldsymbol{W}}\hat{\Lambda}\hat{\boldsymbol{W}}^T. \tag{2.11}$$

Let the estimated eigenvalues $\hat{\Lambda}$ and eigenvectors $\hat{\boldsymbol{W}}$ be partitioned as in (2.4). Let ε be the error in the estimated active subspace,

$$\varepsilon \;=\; \|\boldsymbol{W}_1\boldsymbol{W}_1^T - \hat{\boldsymbol{W}}_1\hat{\boldsymbol{W}}_1^T\| \;=\; \|\hat{\boldsymbol{W}}_1^T\boldsymbol{W}_2\|, \tag{2.12}$$

where $\|\cdot\|$ is the matrix 2-norm; see [19, section 2.5.3]. In [11], we show that when the number N of samples in (2.11) is greater than a problem dependent lower bound, the relative error in the estimated eigenvalues $\hat{\Lambda}$ falls below a user-specified tolerance with high probability; the lower bound grows as the log of the dimension m. With sufficient samples, the subspace error ε in (2.12) satisfies [11, Corollary 3.7]

$$\varepsilon \;\leq\; \frac{4\,\lambda_1\,\delta}{\lambda_n - \lambda_{n+1}}, \tag{2.13}$$

where δ is the user-specified error tolerance for the eigenvalue estimates. In practice, if we observe a gap between $\hat{\lambda}_n$ and $\hat{\lambda}_{n+1}$, then the Monte Carlo procedure gives a good estimate of the n-dimensional subspace.

2.3. Approximation with the estimated active subspace.

Practical versions of the approximations in (2.7) and (2.9) use the estimated eigenvectors $\hat{\boldsymbol{W}}_1$. The estimated active and inactive variables are $\hat{\mathbf{y}} = \hat{\boldsymbol{W}}_1^T\mathbf{x}$ and $\hat{\mathbf{z}} = \hat{\boldsymbol{W}}_2^T\mathbf{x}$, respectively. And the joint density is similar to (2.6), since $\hat{\boldsymbol{W}}$ is an orthogonal matrix. The conditional average using the estimated eigenvectors is

$$g_\varepsilon(\hat{\mathbf{y}}) \;=\; \int f(\hat{\boldsymbol{W}}_1\hat{\mathbf{y}} + \hat{\boldsymbol{W}}_2\hat{\mathbf{z}})\,\rho(\hat{\mathbf{z}}|\hat{\mathbf{y}})\,d\hat{\mathbf{z}}. \tag{2.14}$$

The root-mean-squared error in the approximation $f(\mathbf{x}) \approx g_\varepsilon(\hat{\boldsymbol{W}}_1^T\mathbf{x})$ is given in the next theorem.

Theorem 2.3 (Theorem 3.5 in [9]).

$$\begin{aligned}&\left(\int (f(\mathbf{x}) - g_\varepsilon(\hat{\boldsymbol{W}}_1^T\mathbf{x}))^2\,\rho\,d\mathbf{x}\right)^{1/2}\\ &\qquad\leq\; C\left(\varepsilon(\lambda_1+\cdots+\lambda_n)^{1/2} + (\lambda_{n+1}+\cdots+\lambda_m)^{1/2}\right),\end{aligned} \tag{2.15}$$

where C is from Theorem 2.1 *and ε is the subspace error from* (2.12).

When the error ε in the estimated subspace (see (2.12)) is not zero, the error estimate includes contributions from the larger eigenvalues. The Monte Carlo estimate $\hat{g}_\varepsilon$ of the conditional average g_ε is

$$\hat{g}_\varepsilon(\hat{\mathbf{y}}) \;=\; \frac{1}{M}\sum_{i=1}^{M} f(\hat{\boldsymbol{W}}_1\hat{\mathbf{y}} + \hat{\boldsymbol{W}}_2\hat{\mathbf{z}}_i), \tag{2.16}$$

where $\hat{\mathbf{z}}_i$ are drawn independently from the conditional density $\rho(\hat{\mathbf{z}}|\hat{\mathbf{y}})$. The next theorem bounds the root-mean-squared error in the Monte Carlo approximation.

Theorem 2.4 (Theorem 3.6 in [9]).

$$\begin{aligned}&\left(\int (f(\mathbf{x}) - \hat{g}_\varepsilon(\hat{\boldsymbol{W}}_1^T\mathbf{x}))^2\,\rho\,d\mathbf{x}\right)^{1/2}\\ &\qquad\leq\; C(1+M^{-1/2})\left(\varepsilon(\lambda_1+\cdots+\lambda_n)^{1/2} + (\lambda_{n+1}+\cdots+\lambda_m)^{1/2}\right),\end{aligned} \tag{2.17}$$

where C is from Theorem 2.1 *and ε is the subspace error from* (2.12).

To summarize, we introduce four low-dimensional approximations for $f(\mathbf{x})$ that exploit the active subspace: (i) the conditional average g (2.7), (ii) a Monte Carlo approximation $\hat{g}$ (2.9) of the conditional average, (iii) the conditional average g_ε (2.14) constructed with estimated eigenvectors $\hat{\boldsymbol{W}}_1$, and (iv) the Monte Carlo approximation $\hat{g}_\varepsilon$ (2.16) of the conditional average g_ε. For each approximation, we bound the root-mean-squared error by $\boldsymbol{C}$'s eigenvalues and the subspace error. Practical computations use $\hat{g}_\varepsilon$.

3. Approximating the Bayesian posterior.

We consider the following Bayesian inverse problem. Assume an additive noise model,

$$\mathbf{d} \;=\; \mathbf{m}(\mathbf{x}) + \mathbf{e}, \tag{3.1}$$

where $\mathbf{d} \in \mathbb{R}^d$ are the random data, $\mathbf{x} \in \mathbb{R}^m$ are the random parameters, $\mathbf{m} : \mathbb{R}^m \to \mathbb{R}^d$ is the deterministic map from parameters to observables (e.g., the observation operator composed with the forward model in PDE-based inverse problems), and $\mathbf{e} \in \mathbb{R}^d$ is the random noise. We assume that $\mathbf{m}$ is a differentiable function of $\mathbf{x}$. For simplicity, we assume $\mathbf{e}$ is a zero-mean Gaussian random vector with covariance $\sigma^2 \boldsymbol{I}$; correlated noise can be transformed to uncorrelated noise with standard linear transformations using the square root of the noise covariance matrix. The Gaussian noise assumption leads to the likelihood,

$$\rho_{\text{lik}}(\mathbf{d}, \mathbf{x}) \;=\; \exp\left(\frac{-\|\mathbf{d} - \mathbf{m}(\mathbf{x})\|^2}{2\sigma^2}\right), \tag{3.2}$$

where $\|\cdot\|$ is the Euclidean norm. Denote the prior density on the parameters by $\rho_{\text{pri}}(\mathbf{x})$, and let $\rho_{\text{pos}}(\mathbf{x})$ be the conditional density of $\mathbf{x}$ given $\mathbf{d}$. The Bayes Theorem implies

$$\rho_{\text{pos}}(\mathbf{x}) \;=\; c_{\text{pos}}^{-1}\, \rho_{\text{lik}}(\mathbf{d}, \mathbf{x})\, \rho_{\text{pri}}(\mathbf{x}), \qquad c_{\text{pos}} \;=\; \int \rho_{\text{lik}}(\mathbf{d}, \mathbf{x})\, \rho_{\text{pri}}(\mathbf{x})\, d\mathbf{x}. \tag{3.3}$$

3.1. Identifying the active subspace.

To apply the active subspace machinery from section 2, we must identify the scalar-valued function $f(\mathbf{x})$ and the density function $\rho(\mathbf{x})$ in (2.2). Similar to the likelihood-informed subspace of Cui et al. [14], we choose $f(\mathbf{x})$ to be the negative log-likelihood,

$$f(\mathbf{x}) \;=\; \frac{1}{2\sigma^2}\, \|\mathbf{d} - \mathbf{m}(\mathbf{x})\|^2. \tag{3.4}$$

This function is often called the *data misfit* function, or just *misfit*, and it is closely related to an optimizer's objective function in deterministic inverse problems. For a given point in the parameter space, $f(\mathbf{x})$ measures how far the modeled observations are from the given data. If we use the misfit to define the active subspace, then the orthogonal complement—i.e., the inactive subspace—identifies directions along which the likelihood is relatively flat. Perturbing the parameters along the inactive subspace changes the likelihood relatively little, on average. We wish to exploit this structure, when present, to accelerate the MCMC. The gradient of the misfit is

$$\nabla f(\mathbf{x}) \;=\; \frac{1}{\sigma^2}\, \nabla \mathbf{m}(\mathbf{x})^T\, (\mathbf{d} - \mathbf{m}(\mathbf{x})), \tag{3.5}$$

where $\nabla \mathbf{m} \in \mathbb{R}^{d\times m}$ is the Jacobian of the parameter-to-observable map. The misfit and its gradient depend on the data $\mathbf{d}$; this becomes important when we choose ρ.

3.2. Integrating against the prior.

Cui et al. [14] average the misfit's prior-preconditioned Gauss–Newton Hessian with respect to the posterior density in (3.3) to estimate the likelihood-informed subspace. Their subspace is then conditioned on the data. In contrast, we compute the averages defining $\boldsymbol{C}$ in (2.2) using $\rho = \rho_{\text{pri}}$, which requires careful interpretation of the data $\mathbf{d}$. In the model (3.1), $\mathbf{d}$ is a random variable whose mean depends on $\mathbf{x}$; in other words, $\mathbf{d}$ and $\mathbf{x}$ are not independent. Therefore, we cannot integrate against the prior without $\mathbf{d}$ changing as $\mathbf{x}$ varies. However, if we treat the realization $\mathbf{d}$ as a fixed and constant vector, then we can integrate $\nabla f \nabla f^T$, which depends on $\mathbf{d}$, against a density function equal to ρ_{pri} without issue. The integrals defining $\boldsymbol{C}$ from (2.2) and the approximation g from (2.7) are well-defined. We sacrifice the probabilistic interpretation of the data $\mathbf{d}$. Hence, we also sacrifice the interpretation of $\boldsymbol{C}$ and all derived quantities as random variables conditioned on $\mathbf{d}$; instead, they are functions of the fixed vector $\mathbf{d}$. Additionally, if we are given a new set of data, then we must recompute the active subspace. But we gain the practical advantage of estimating $\boldsymbol{C}$ with simple Monte Carlo as in section 2.2, since we can sample independently from the density ρ_{pri}. With this choice of ρ, the Monte Carlo–based eigenvector estimates $\hat{\boldsymbol{W}}$ from (2.11) are computable, and approximations to the misfit $f(\mathbf{x})$, namely, $\hat{g}$ from (2.9), g_ε from (2.14), and $\hat{g}_\varepsilon$ from (2.16), are also well-defined.

Consider the approximation $f(\mathbf{x}) \approx g(\boldsymbol{W}_1^T\mathbf{x})$ from (2.7). The prior can be factored as

$$\rho_{\text{pri}}(\mathbf{x}) \;=\; \rho_{\text{pri}}(\mathbf{y},\mathbf{z}) \;=\; \rho_{\text{pri}}(\mathbf{y})\,\rho_{\text{pri}}(\mathbf{z}|\mathbf{y}), \tag{3.6}$$

where $\rho_{\text{pri}}(\mathbf{y})$ is the marginal density of $\mathbf{y}$ and $\rho_{\text{pri}}(\mathbf{z}|\mathbf{y})$ is the conditional density of $\mathbf{z}$ given $\mathbf{y}$. If $\rho_{\text{pri}}(\mathbf{x})$ is a standard Gaussian on $\mathbb{R}^m$, then $\rho_{\text{pri}}(\mathbf{y})$ is a standard Gaussian on $\mathbb{R}^n$, and $\rho_{\text{pri}}(\mathbf{z}|\mathbf{y})$ is a standard Gaussian on $\mathbb{R}^{m-n}$ that is independent of $\mathbf{y}$. We construct an approximate posterior $\pi(\mathbf{x})$ as

$$\begin{aligned}\rho_{\text{pos}}(\mathbf{x}) &\approx \pi(\mathbf{x})\\ &= c_\pi^{-1}\,\exp(-g(\boldsymbol{W}_1^T\mathbf{x}))\,\rho_{\text{pri}}(\mathbf{x})\\ &= c_\pi^{-1}\,\exp(-g(\mathbf{y}))\,\rho_{\text{pri}}(\mathbf{y})\,\rho_{\text{pri}}(\mathbf{z}|\mathbf{y}),\end{aligned} \tag{3.7}$$

where

$$c_\pi \;=\; \int \exp(-g(\boldsymbol{W}_1^T\mathbf{x}))\,\rho_{\text{pri}}(\mathbf{x})\,d\mathbf{x}. \tag{3.8}$$

Since $\exp(\cdot) > 0$, $c_\pi > 0$. Also, $g \geq 0$ implies $\exp(-g) \leq 1$, so $c_\pi \leq 1$. The approximation in (3.7) suggests a strategy for MCMC that runs the Markov chain only on the active variables $\mathbf{y}$ while sampling independently from the prior $\rho_{\text{pri}}(\mathbf{z}|\mathbf{y})$ on the inactive variables; we explore this strategy in section 4. Before the computational exploration, we study the approximation properties of π and similar constructions using $\hat{g}$, g_ε, and $\hat{g}_\varepsilon$ in place of g. We use the Hellinger distance [17] to quantify the approximation errors.

THEOREM 3.1. *Let π be defined as in* (3.7)*, and define the approximate posteriors $\hat{\pi}$, π_ε, and $\hat{\pi}_\varepsilon$ using $\hat{g}$, g_ε, and $\hat{g}_\varepsilon$, respectively, in place of g—all constructed with $\rho = \rho_{\text{pri}}$. Define the constant L as*

$$L^2 \;=\; \frac{1}{8}\left[\left(\int \exp(-f)\,\rho_{\text{pri}}\,d\mathbf{x}\right)\left(\exp\left(-\int f\,\rho_{\text{pri}}\,d\mathbf{x}\right)\right)\right]^{-1/2}. \tag{3.9}$$

Then the Hellinger distances between the approximate posteriors and the true posterior ρ_{pos} are bounded as follows:

$$H(\rho_{\text{pos}},\pi) \leq \; L\,C\,(\lambda_{n+1}+\cdots+\lambda_m)^{1/2}, \tag{3.10}$$

$$H(\rho_{\text{pos}},\hat{\pi}) \leq \; L\,C\,\left(1+M^{-1/2}\right)(\lambda_{n+1}+\cdots+\lambda_m)^{1/2}, \tag{3.11}$$

$$H(\rho_{\text{pos}},\pi_\varepsilon) \leq \; L\,C\,\left(\varepsilon\,(\lambda_1+\cdots+\lambda_n)^{1/2}+(\lambda_{n+1}+\cdots+\lambda_m)^{1/2}\right), \tag{3.12}$$

$$H(\rho_{\text{pos}},\hat{\pi}_\varepsilon) \leq \; L\,C\,\left(1+M^{-1/2}\right)\left(\varepsilon\,(\lambda_1+\cdots+\lambda_n)^{1/2}+(\lambda_{n+1}+\cdots+\lambda_m)^{1/2}\right), \tag{3.13}$$

where C is the Poincaré constant associated with ρ_{pri}, M is from (2.9) *and* (2.16), *and ε is from* (2.12).

The bound (3.10) is an improved and extended version of Theorem 4.9 in [8]. The proof of Theorem 3.1 is in Appendix A. If the eigenvalues $\lambda_{n+1},\ldots,\lambda_m$ are small, and if the error ε in the numerically estimated active subspace is small, then the Hellinger distances between the posterior and its approximations are small. The Hellinger distance is a useful metric, because it provides an upper bound on the posterior mean and covariance; see [27, Lemma 6.37].

3.3. Linear forward model. Consider the case where the forward model is linear in the parameters,

$$\mathbf{m}(\mathbf{x}) \;=\; \boldsymbol{M}\mathbf{x}, \qquad \boldsymbol{M}\in\mathbb{R}^{d\times m}. \tag{3.14}$$

Assume $\boldsymbol{M}$ has rank r. If the prior and measurement noise are Gaussian, then the posterior is a Gaussian density whose mean and covariance have closed-form expressions in terms of the data, the prior covariance, and the noise covariance. A Gaussian density is completely characterized by its mean and covariance, so this case is not interesting to study with MCMC. Nevertheless, we can examine how the posterior approximation π from (3.7) compares to the true posterior. The gradient of the misfit (3.4) with the linear forward model is

$$\nabla f(\mathbf{x}) \;=\; \frac{1}{\sigma^2}\,\boldsymbol{M}^T(\boldsymbol{M}\mathbf{x}-\mathbf{d}). \tag{3.15}$$

Consider the case where ρ_{pri} is a standard Gaussian, i.e.,

$$\rho_{\text{pri}}(\mathbf{x}) \;=\; (2\pi)^{\frac{-m}{2}}\,\exp\left(\frac{-\mathbf{x}^T\mathbf{x}}{2}\right). \tag{3.16}$$

Using $\rho=\rho_{\text{pri}}$, the matrix $\boldsymbol{C}$ from (2.2) is

$$\boldsymbol{C} \;=\; \frac{1}{\sigma^4}\,\boldsymbol{M}^T(\boldsymbol{M}\boldsymbol{M}^T+\mathbf{d}\mathbf{d}^T)\boldsymbol{M}. \tag{3.17}$$

If $\boldsymbol{W}_1$ are the first $n<m$ eigenvectors of $\boldsymbol{C}$ from (3.17), then the conditional average $g(\mathbf{y})$ from (2.7) is

$$\begin{aligned} g(\mathbf{y}) &= \frac{1}{2\sigma^2}\int \|\boldsymbol{M}(\boldsymbol{W}_1\mathbf{y}+\boldsymbol{W}_2\mathbf{z})-\mathbf{d}\|^2\;(2\pi)^{\frac{-(m-n)}{2}}\;\exp\left(\frac{-\mathbf{z}^T\mathbf{z}}{2}\right)\,d\mathbf{z}\\ &= \frac{1}{2\sigma^2}\left(\|\boldsymbol{M}\boldsymbol{W}_1\mathbf{y}-\mathbf{d}\|^2+\gamma^2\right), \end{aligned} \tag{3.18}$$

where

$$\gamma^2 \;=\; \int \mathbf{z}^T \boldsymbol{W}_2^T \boldsymbol{M}^T \boldsymbol{M} \boldsymbol{W}_2 \mathbf{z}\, (2\pi)^{\frac{-(m-n)}{2}} \exp\left(\frac{-\mathbf{z}^T\mathbf{z}}{2}\right) d\mathbf{z} \tag{3.19}$$

is independent of $\mathbf{y}$. The posterior approximation π is

$$\pi(\mathbf{x}) \;=\; c_\pi^{-1} \exp\left(\frac{-\gamma^2}{2\sigma^2}\right) \exp\left(\frac{-\|\boldsymbol{M}\boldsymbol{W}_1\boldsymbol{W}_1^T\mathbf{x} - \mathbf{d}\|^2}{2\sigma^2}\right) \rho_{\text{pri}}(\mathbf{x}). \tag{3.20}$$

Note that $\exp(-\gamma^2/2\sigma^2)$ is independent of $\mathbf{x}$. Using standard manipulations as in [6, Chapter 8], we can write down π's mean μ and covariance matrix Γ,

$$\begin{aligned} \mu &= \tilde{\boldsymbol{M}}^T(\tilde{\boldsymbol{M}}\tilde{\boldsymbol{M}}^T + \sigma^2\boldsymbol{I})^{-1}\mathbf{d},\\ \Gamma &= \boldsymbol{I} - \tilde{\boldsymbol{M}}^T(\tilde{\boldsymbol{M}}\tilde{\boldsymbol{M}}^T + \sigma^2\boldsymbol{I})^{-1}\tilde{\boldsymbol{M}}, \end{aligned} \tag{3.21}$$

where

$$\tilde{\boldsymbol{M}} \;=\; \boldsymbol{M}\boldsymbol{W}_1\boldsymbol{W}_1^T. \tag{3.22}$$

Since $\text{rank}(\boldsymbol{M})$ is r, $\text{rank}(\boldsymbol{C}) \leq r$ for $\boldsymbol{C}$ from (3.17). Therefore, $\boldsymbol{C}$'s eigenvalues $\lambda_{r+1}, \dots, \lambda_m$ are zero. Applying Theorem 3.1, the Hellinger distance between π and ρ_{pos} is zero when the number n of active variables is greater than or equal to r. In that case, the mean and covariance approximations in (3.21) are exact.

Flath et al. [15] and Bui-Thanh et al. [4] construct an approximation to the posterior covariance with a rank-k update of the prior covariance. They derive the rank-k update from the k dominant eigenpairs of the negative log-likelihood's Hessian, preconditioned by the prior. Spantini et al. [26] recently showed that such an update is optimal in a general class of matrix norms that implies an optimal posterior approximation in the Hellinger distance. The approximation Γ in (3.21) is indeed a rank-n update to the prior covariance $\boldsymbol{I}$. However, it is difficult to compare this update to the Hessian-based constructions when $n < r$ because the eigenvectors $\boldsymbol{W}_1$ depend on the fixed data vector $\mathbf{d}$, while the eigenpairs of the negative log-likelihood's Hessian do not. It is possible to construct simple cases where the rank-n update in (3.21) is a poor approximation for a particular fixed $\mathbf{d}$. However, such a deficiency reveals little about the inverse problem with nonlinear forward models—where MCMC methods are most appropriate. For nonlinear forward models, it may be that data-dependent approximations (i.e., approximations that depend on $\mathbf{d}$) are preferable to data-independent approximations for a particular realization of the data.

4. MCMC with the active subspace.

Recall that choosing $\rho = \rho_{\text{pri}}$ in (2.2) and (2.7) requires us to interpret the data $\mathbf{d}$ as a fixed and constant vector; we sacrifice the probabilistic interpretation of $\mathbf{d}$ and any quantities that depend on $\mathbf{d}$. However, using $\rho = \rho_{\text{pri}}$ allows us to use simple Monte Carlo to estimate the integrals, which can be done in parallel. Theorem 3.1 contains error estimates for all likelihood approximations: the conditional expectation g, its Monte Carlo estimate $\hat{g}$, and the analogous approximations using the estimated eigenvectors, g_ε and $\hat{g}_\varepsilon$. We restrict attention to the computable approximation $\hat{g}_\varepsilon(\hat{\boldsymbol{W}}_1^T\mathbf{x})$ from (2.16) of the misfit $f(\mathbf{x})$ from (3.4). Recall that $\hat{g}_\varepsilon$ is a Monte Carlo estimate of the conditional expectation of f given the estimated active variables $\hat{\mathbf{y}} = \hat{\boldsymbol{W}}_1^T\mathbf{x}$.

We first compute the eigenpair estimates $\hat{\boldsymbol{W}}$ and $\hat{\Lambda}$ with Monte Carlo as in section 2.2 using the misfit's gradient (3.5), where the samples $\mathbf{x}_j$ are drawn independently according to the prior. One can view this step as preprocessing before running any MCMC. This preprocessing step checks for exploitable, low-dimensional structure indicated by (i) a gap in the estimated eigenvalues and (ii) small eigenvalues following the gap. A gap in the spectrum indicates that the Monte Carlo procedure in section 2.2 can accurately estimate the active subspace; see (2.13). In [11, section 4], we describe a practical bootstrap procedure that can aid in assessing the quality of the estimated active subspace. If the eigenvalues following the gap are small, then Theorem 3.1 gives confidence that the approximate posterior is close to the true posterior. If these two conditions are not satisfied, then the problem may not be a good candidate for the proposed active subspace-accelerated MCMC, since there is no evidence that an exploitable active subspace exists based on the misfit function $f(\mathbf{x})$ and the prior density $\rho = \rho_{\text{pri}}$.

Assuming we have identified an active subspace, we propose an MCMC method that exploits the low-dimensional structure. Algorithm 1 is an active subspace-exploiting variant of the Metropolis–Hastings method, sometimes called the *random walk* method, outlined by Kaipio and Somersalo [22, Chapter 3]. The essential idea is to run the Markov chain only on the n active variables instead of all m variables. Therefore, we expect the chain to mix faster than MCMC on all m variables. We do not compare to more sophisticated sampling schemes, because they can be adapted to exploit the active subspace in the same way.

Algorithm 1. MCMC with the active subspace.

Pick an initial value $\hat{\mathbf{y}}_1$, and compute $\hat{g}_\varepsilon(\hat{\mathbf{y}}_1)$. Set $k = 1$.

1. Draw $\hat{\mathbf{y}}' \in \mathbb{R}^n$ from a symmetric proposal density centered at $\hat{\mathbf{y}}_k$.
2. Compute $\hat{g}_\varepsilon(\hat{\mathbf{y}}')$ as in (2.16), where f is the misfit function (3.4), and $\hat{\mathbf{z}}_i$ are drawn independently according to $\rho(\hat{\mathbf{z}}|\hat{\mathbf{y}})$, which is a standard Gaussian on $\mathbb{R}^{m-n}$.
3. Compute the acceptance ratio

$$\gamma(\hat{\mathbf{y}}_k, \hat{\mathbf{y}}') \; = \; \text{minimum} \left(1, \, \frac{\exp(-\hat{g}_\varepsilon(\hat{\mathbf{y}}'))\rho_{\text{pri}}(\hat{\mathbf{y}}')}{\exp(-\hat{g}_\varepsilon(\hat{\mathbf{y}}_k))\rho_{\text{pri}}(\hat{\mathbf{y}}_k)} \right). \tag{4.1}$$

4. Draw t uniformly from $[0, 1]$.
5. If $\gamma(\hat{\mathbf{y}}_k, \hat{\mathbf{y}}') \geq t$, set $\hat{\mathbf{y}}_{k+1} = \hat{\mathbf{y}}'$. Otherwise, set $\hat{\mathbf{y}}_{k+1} = \hat{\mathbf{y}}_k$.
6. Increment k and repeat.

Step 2 computes $\hat{g}_\varepsilon(\hat{\mathbf{y}}')$ from (2.16). Each sample in (2.16) requires an independent evaluation of the misfit $f(\mathbf{x})$, and hence the parameter-to-observable map $\mathbf{m}(\mathbf{x})$. Therefore, each step of the Markov chain in the n active variables uses M forward model evaluations—compared to one forward model evaluation for each step in the standard Metropolis–Hastings. We expect that in many problems (such as the example in section 5), the dimension reduction enabled by the active subspace is far more valuable—in terms of forward model evaluations needed to sufficiently sample the space—than the penalty of a factor of M increase. A factor M increase is much smaller than the exponential growth of the parameter space with dimension.

How large should M be to compute $\hat{g}_\varepsilon$ in step 2? The error estimate (3.13) in Theorem 3.1 provides some guidance. Note that M enters the error bound through the multiplicative term $1 + M^{-1/2}$. Thus, the effect of M is bounded between 1 (for very large M) and 2 (for $M = 1$). In other words, M has relatively little effect on the error estimate; the eigenvalues matter much more. If the eigenvalues $\lambda_{n+1}, \dots, \lambda_m$ are small, and if the gap $\lambda_n - \lambda_{n+1}$ is large—implying a small subspace error ε—then M can be surprisingly small with little effect on the error. However, it is difficult to provide a universal numerical condition (e.g., a tolerance) for the eigenvalues that directly leads to a useful M. In the experiment in section 5, we perform a preliminary computational experiment that justifies $M = 10$; we recommend such an experiment in practice.

Algorithm 1 generates the set $\{\hat{\mathbf{y}}_k\}$. These samples must be transformed to the space of the original parameters $\mathbf{x}$ for inference. For each $\hat{\mathbf{y}}_k$, draw independent realizations of $\hat{\mathbf{z}}_{k,\ell}$ from the conditional density $\rho_{\text{pri}}(\hat{\mathbf{z}}|\hat{\mathbf{y}}_k)$ with $\ell = 1, \dots, P$. Then construct

$$\mathbf{x}_{k,\ell} \;=\; \hat{\boldsymbol{W}}_1\hat{\mathbf{y}}_k + \hat{\boldsymbol{W}}_2\hat{\mathbf{z}}_{k,\ell}. \tag{4.2}$$

To be sure, constructing $\{\mathbf{x}_{k,\ell}\}$ from $\{\hat{\mathbf{y}}_k\}$ requires no forward model evaluations; it only requires that one be able to draw independent samples from the conditional density $\rho_{\text{pri}}(\hat{\mathbf{z}}|\hat{\mathbf{y}})$. If $\rho_{\text{pri}}(\mathbf{x})$ is a standard Gaussian as in our numerical examples, then $\rho_{\text{pri}}(\hat{\mathbf{z}}|\hat{\mathbf{y}})$ is a standard Gaussian, so drawing independent samples is straightforward. By a derivation similar to (3.7), the set $\{\mathbf{x}_{k,\ell}\}$ contains correlated samples from the approximate posterior $\hat{\pi}_\varepsilon$ defined as

$$\hat{\pi}_\varepsilon(\mathbf{x}) \;=\; c_{\hat{\pi}_\varepsilon}^{-1}\,\exp(-\hat{g}_\varepsilon(\hat{\boldsymbol{W}}_1^T\mathbf{x}))\,\rho_{\text{pri}}(\mathbf{x}), \tag{4.3}$$

where

$$c_{\hat{\pi}_\varepsilon} \;=\; \int \exp(-\hat{g}_\varepsilon(\hat{\boldsymbol{W}}_1^T\mathbf{x}))\,\rho_{\text{pri}}(\mathbf{x})\,d\mathbf{x}. \tag{4.4}$$

Recall that Theorem 3.1 bounds the Hellinger distance between $\hat{\pi}_\varepsilon$ and the true posterior in (3.13). We expect the correlation in the set of samples $\{\mathbf{x}_{k,\ell}\}$ to be much smaller than a set of samples drawn with MCMC directly on the parameters $\mathbf{x}$, since the $\mathbf{x}_{k,\ell}$'s from (4.2) contain many independently sampled components, $\hat{\mathbf{z}}_{k,\ell}$. The problem in section 5.2 shows an example of such behavior.

5. Numerical experiments.

The following experiments used MATLAB 2015b and Enthought Canopy Python 2.7.9 on a 2013 MacBook Air with 8GB of RAM for the computations with the quadratic model from section 5.1 and most postprocessing and plotting. The PDE-based experiment from section 5.2 ran on two processors from one node of Colorado School of Mines' Mio cluster (inside.mines.edu/mio) using the same Python distribution. The scripts and data to produce the figures for the following numerical experiments can be found at bitbucket.org/paulcon/accelerating-mcmc-with-active-subspaces.

5.1. Two-parameter model.

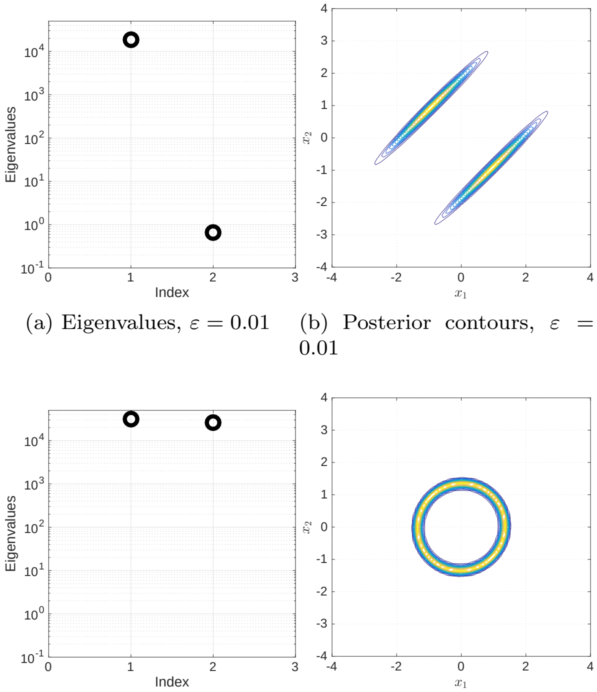
We first demonstrate the approach on a simple two-parameter quadratic forward model with a scalar output; in this case $\mathbf{d}$ has one component. Let $\mathbf{x} = [x_1, x_2]^T$, and we assume a standard Gaussian prior. Define the parameter-to-observable map $\mathbf{m}(\mathbf{x})$ as

$$\mathbf{m}(\mathbf{x}) \;=\; \frac{1}{2}\,\mathbf{x}^T\boldsymbol{A}\mathbf{x}, \tag{5.1}$$

where

$$\boldsymbol{A} = \boldsymbol{Q}\begin{bmatrix}1 & \\ & \varepsilon\end{bmatrix}\boldsymbol{Q}^T, \qquad \boldsymbol{Q} = \frac{1}{2}\begin{bmatrix}\sqrt{2} & \sqrt{2}\\ -\sqrt{2} & \sqrt{2}\end{bmatrix}. \tag{5.2}$$

This map goes into the likelihood (3.2) with noise parameter $\sigma^2 = 0.1$, and we set the fixed data $\mathbf{d} = 0.9$. The parameter ε in (5.2) controls how active the one-dimensional active subspace is. Figure 5.1 shows two cases: (i) $\varepsilon = 0.01$ and (ii) $\varepsilon = 0.95$. We



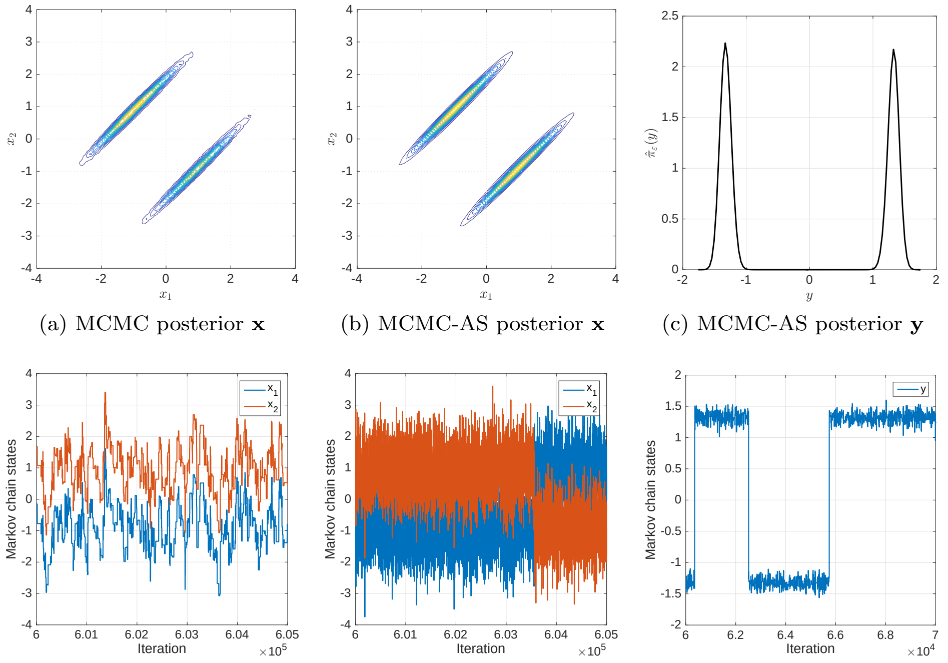
(a) Eigenvalues, $\varepsilon = 0.01$ (b) Posterior contours, $\varepsilon = 0.01$

(c) Eigenvalues, $\varepsilon = 0.95$ (d) Posterior contours, $\varepsilon = 0.95$

FIG. 5.1. *Two cases of* $\boldsymbol{A}$ *in the forward map* (5.1)*. The first uses* $\varepsilon = 0.01$ *in* (5.2) *(top row) and the second uses* $\varepsilon = 0.95$ *(bottom row). The leftmost figures show the two eigenvalues from* $\boldsymbol{C}$ *in* (2.2) *computed with a high order Gauss–Hermite quadrature rule. The rightmost figures show the contours of the posterior density conditioned on* $\mathbf{d} = 0.9$*. (Colors are visible in the electronic version.)*

compute the elements of $\boldsymbol{C}$—each a two-dimensional integral—using a tensor product Gauss–Hermite quadrature rule with 50 points in each dimension. The high order integration rule eliminates the finite sampling errors present in Monte Carlo estimates. The eigenvalues of $\boldsymbol{C}$ with $\varepsilon = 0.01$ are shown in Figure 5.1(a). The gap between the eigenvalues suggests an active one-dimensional active subspace. This is confirmed in Figure 5.1(b), which plots the skewed contours of the posterior density conditioned on $\mathbf{d} = 0.9$. The comparable figures for $\varepsilon = 0.95$ are shown in Figures 5.1(c) and 5.1(d), respectively. In this case, both eigenvalues have the same order of magnitude and the smallest eigenvalue is large, so they do not suggest an exploitable active subspace. This is confirmed by the posterior contours that vary significantly in all directions.

Next we compare standard Metropolis–Hastings MCMC on the two-dimensional parameter space to an active subspace-accelerated MCMC; we use the $\varepsilon = 0.01$ case of the forward model (5.1) with data $\mathbf{d} = 0.9$. The active subspace-accelerated method (i) runs Metropolis–Hastings on the estimated active variables $\hat{\mathbf{y}}$ and (ii) samples independently from the prior on the inactive variables $\hat{\mathbf{z}}$ for each sample of $\hat{\mathbf{y}}$. In this case, $\hat{\mathbf{y}}$ and $\hat{\mathbf{z}}$ each have one component. The bivariate standard Gaussian prior on $\mathbf{x}$ implies that the conditional density of $\hat{\mathbf{z}}$ given $\hat{\mathbf{y}}$ is a univariate standard Gaussian prior.



(a) MCMC posterior **x**

(b) MCMC-AS posterior **x**

(c) MCMC-AS posterior **y**

(d) MCMC trace **x**

(e) MCMC-AS trace **x**

(f) MCMC-AS trace **y**

FIG. 5.2. *Comparison of standard Metropolis–Hastings MCMC to the active subspace accelerated MCMC.* (a) *Density estimate of the posterior contours for the standard MCMC.* (b) *The same contours for the samples from Algorithm* 1 *using the reconstruction* (4.2)*. The plots directly beneath them show trace plots for steps* 600000 *to* 605000*; notice that the active subspace-accelerated chain mixes better. The rightmost plots show the univariate posterior on the active variable* $\hat{\mathbf{y}}$ *(above) and its trace plot for* 10000 *representative steps (below). (Colors are visible in the electronic version.)*

We run the standard MCMC for 10^6 steps using a two-dimensional standard Gaussian proposal density with variance $\sigma^2 = 0.5$. The acceptance rate was 12%, which is not surprising since the true posterior in Figure 5.1(b) shows two small, separated regions of large posterior probability. Figure 5.2(a) shows the contours of a bivariate Gaussian kernel density estimate [2] using all 10^6 steps. Figure 5.2(d) shows the trace plots for steps 600000 to 605000.

We run the active subspace-accelerated MCMC for 10^5 steps. At each step, we evaluate the misfit approximation $\hat{g}_\varepsilon(\hat{\mathbf{y}})$ with a 10-point Gauss–Hermite quadrature rule, which is appropriate because (i) $\hat{\mathbf{z}}$ has one component and (ii) the conditional density of $\hat{\mathbf{z}}$ given $\hat{\mathbf{y}}$ is a standard Gaussian. Since each approximate misfit uses 10 forward model evaluations, the total number of forward model evaluations is the same for each MCMC method. The proposal density for Metropolis–Hastings on $\hat{\mathbf{y}}$ is a univariate standard Gaussian with variance $\sigma^2 = 0.5$, and the acceptance rate was 13%. We draw $P = 10$ independent samples from the Gaussian prior on $\hat{\mathbf{z}}$ for each step in the chain as in (4.2); recall that drawing these samples needs no forward model evaluations. Figure 5.2(b) shows the contours of the approximate posterior density constructed with the same bivariate Gaussian kernel density estimation from samples of the active subspace-accelerated MCMC; these contours compare favorably to the true posterior contours in Figure 5.1(b). The trace plots for the **x** components are shown in Figure 5.2(e) for steps 600000 to 605000. Notice the difference between the trace plots in Figure 5.2(d) and those in Figure 5.2(e). Since $\hat{\mathbf{z}}$ are drawn independently according to their Gaussian prior, the samples from the active subspace-accelerated MCMC are much less correlated than the samples from MCMC on the two-dimensional $\mathbf{x}$ space. Figure 5.2(c) shows a univariate Gaussian kernel density estimate of the posterior on $\hat{\mathbf{y}}$ computed with the 10^5 samples from the Metropolis–Hastings; Figure 5.2(f) shows the trace plot for steps 60001 to 70000.

In this small example where the misfit admits a one-dimensional active subspace, the active subspace-accelerated MCMC performs very well. For the same number of forward model evaluations, the active subspace enables the Metropolis–Hastings to run on only the active variable $\hat{\mathbf{y}}$, while the inactive variable $\hat{\mathbf{z}}$ is sampled independently according to its prior. This procedure produces samples of $\mathbf{x}$ with very little correlation.

5.2. PDE model. Next we apply the active subspace-accelerated MCMC on a synthetic inverse problem derived from a PDE forward model. The PDE solution $u = u(\mathbf{s}, \mathbf{x})$ satisfies

$$-\nabla_{\mathbf{s}} \cdot (a\,\nabla_{\mathbf{s}} u) \;=\; 1, \qquad \mathbf{s} \in [0,1]^2. \tag{5.3}$$

The coordinates $\mathbf{s}$ represent the solution's spatial variables, and $\nabla_{\mathbf{s}}$ denotes differentiation with respect to $\mathbf{s}$. The boundary conditions are $u = 0$ on the bottom ($s_2 = 0$), left ($s_1 = 0$), and top ($s_2 = 1$) of the domain, and $\mathbf{n} \cdot (a\,\nabla_{\mathbf{s}} u) = 0$ on the right ($s_1 = 1$) of the domain, where $\mathbf{n}$ is the unit normal vector. The log of the coefficients $a = a(\mathbf{s}, \mathbf{x})$ is a truncated Karhunen–Loeve series,

$$\log(a(\mathbf{s}, \mathbf{x})) \;=\; \sum_{i=1}^{m} \sqrt{\sigma_i}\,\phi_i(\mathbf{s})\,x_i, \tag{5.4}$$

with $m = 100$ terms. The pairs $(\sigma_i, \phi_i(\mathbf{s}))$ are the eigenpairs of an exponential two-point correlation function,

$$\mathcal{C}(\mathbf{s}_1, \mathbf{s}_2) \;=\; \exp\left(-\frac{\|\mathbf{s}_1 - \mathbf{s}_2\|_1}{\beta}\right), \tag{5.5}$$

where $\|\cdot\|_1$ is the 1-norm on $\mathbb{R}^2$, and $\beta = 0.02$ is the correlation length. Note that the correlation length 0.02 is short compared to the size of the spatial domain, and the 1-norm generates rough coefficient fields. The Karhunen–Loeve eigenvalues σ_i are shown in Figure 5.4(b); their slow decay reflects the short correlation length. Given a point $\mathbf{x} = [x_1, \dots, x_m]^T$, the forward model evaluates the Karhunen–Loeve expansion (5.4) and solves the PDE (5.3) with second-order finite differences—discretized using 100 points in each spatial dimension. Figure 5.3 shows three realizations of the log-coefficients with parameters drawn according to a standard Gaussian (bottom row) and their corresponding PDE solutions (top row).

5.3. Parameters and data. The coefficients $\mathbf{x}$ in (5.4) are parameters in the statistical inverse problem. We use a standard Gaussian prior on $\mathbf{x}$, which implies a Gaussian random field prior on $\log(a)$ with a low-rank (m-term) correlation function approximating (5.5). This setup is similar to the a priori dimension reduction proposed by Marzouk and Najm [25], which, for this case, reduces the dimension of the parameter space from 10000 to 100 via the truncated Karhunen–Loeve series.

The forward model's observations come from the PDE solution u evaluated at seven points on the right boundary, $s_2 \in \{0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$. The

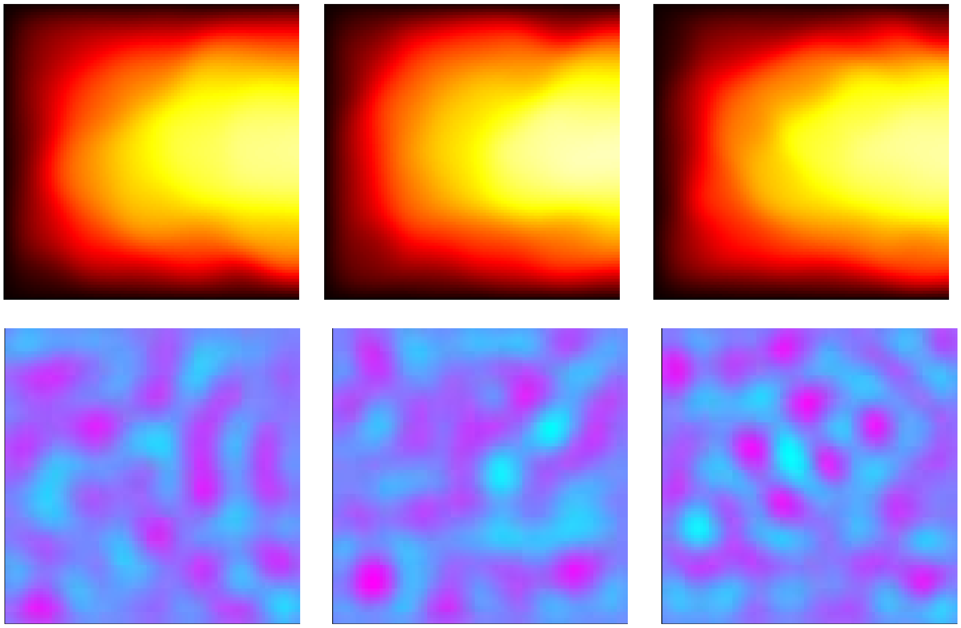
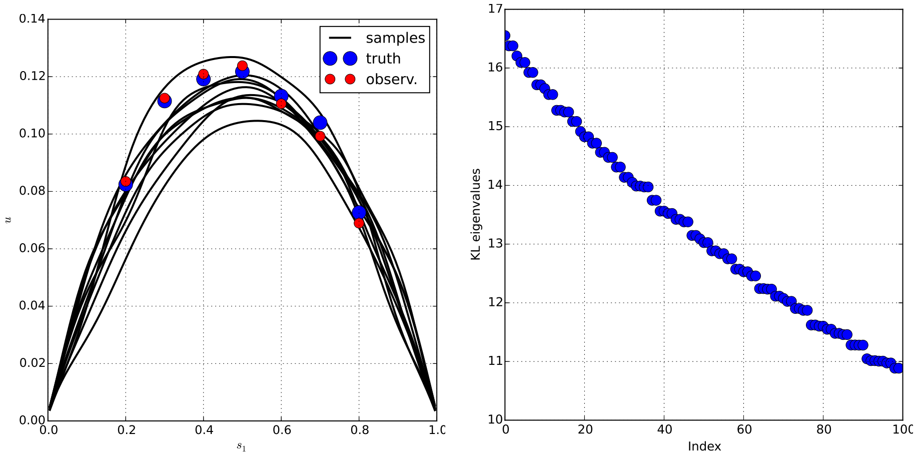


FIG. 5.3. *The bottom row shows three realizations of the coefficients* $\log(a)$ *with parameters* $\mathbf{x}$ *drawn according to the prior. The top row shows the corresponding PDE solutions. (Colors are visible in the electronic version.)*

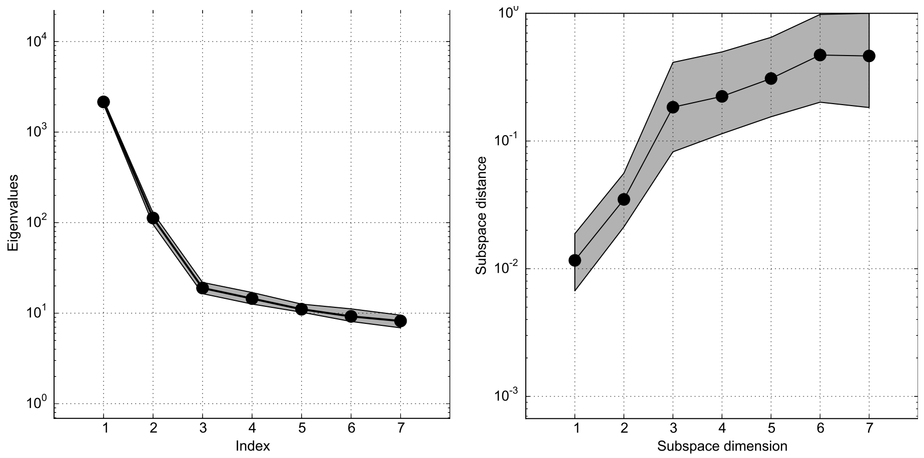
parameter-to-observable map $\mathbf{m}(\mathbf{x})$ from (3.1) takes the Karhunen–Loeve coefficients $\mathbf{x}$ and returns the PDE solution at those seven points on the boundary. To generate the data for the synthetic inverse problem, we draw $\mathbf{x}_{\text{true}}$ according to the prior and evaluate (3.1) with $\mathbf{x} = \mathbf{x}_{\text{true}}$. The noise is $\sigma^2 = 0.0001\, \|\mathbf{m}(\mathbf{x}_{\text{true}})\|_2^2$, which is roughly 1%. In Figure 5.4(a), the black lines are 10 realizations of the PDE solution u on the right boundary ($s_1 = 1$) with parameters $\mathbf{x}$ drawn according to the prior. The blue dots are the observations from the solution evaluated with $\mathbf{x} = \mathbf{x}_{\text{true}}$, and the red dots are the observations perturbed by the noise that constitute the synthetic data for the inverse problem.

5.4. Estimating the active subspace. The Python code that evaluates the forward model is built using Wang's automatic differentiation package `numpad` [29]. This allows us to compute the misfit's gradient (3.5) efficiently. With the prior $\rho_{\text{pri}}(\mathbf{x})$, misfit $f(\mathbf{x})$, and gradient $\nabla f(\mathbf{x})$ defined, we can estimate the active subspace. We use $N = 1000$ gradient samples in the computation (2.11). Figure 5.5(a) shows the first seven of the $m = 100$ eigenvalues and their bootstrap ranges. The gaps between the first two pairs of eigenvalues suggest that the Monte Carlo approximation (2.11) can accurately estimate a one- or two-dimensional active subspace. Figure 5.5(b) shows bootstrap estimates of the subspace error (2.12) as described in [11]. The relatively small error estimates for the one- and two-dimensional subspaces are consistent with the eigenvalue gaps. We choose to study a two-dimensional active subspace, since (i) the gap between eigenvalues $\hat{\lambda}_2$ and $\hat{\lambda}_3$ and the estimated subspace error suggest a good approximation of the subspace, and (ii) the error estimate in Theorem 3.1 then includes the last 98 eigenvalues instead of the last 99 for a one-dimensional subspace. In other words, the theoretical error bound for the approximate posterior is smaller



(a) Forward model realizations and data (b) Karhunen–Loeve eigenvalues

FIG. 5.4. (a) *Ten realizations of the PDE solution with the parameters drawn according to the prior. The blue dots are the observations corresponding to* $\mathbf{x} = \mathbf{x}_{true}$, *and the red dots show the observations perturbed by* 1% *noise to generate synthetic data for the inverse problem.* (b) *The eigenvalues of the Karhunen–Loeve expansion in* (5.4). *The slow decay corresponds to the short correlation length. (Colors are visible in the electronic version.)*

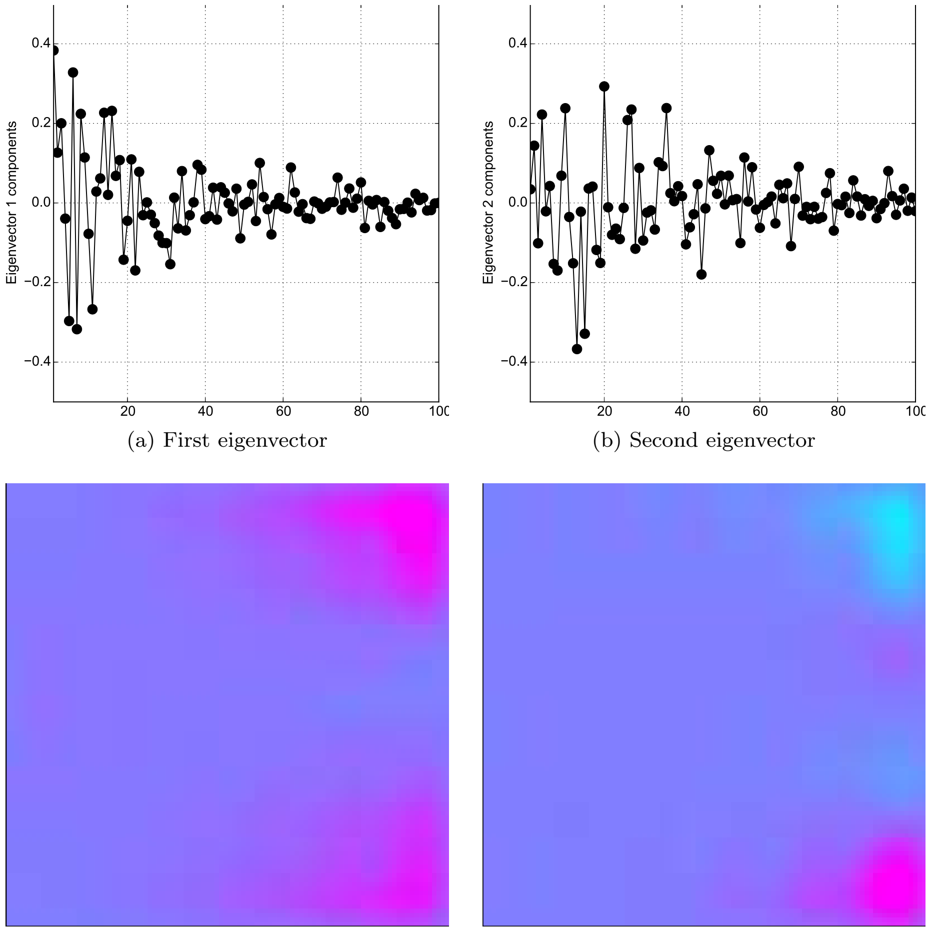


(a) Active subspace eigenvalues (b) Active subspace errors

FIG. 5.5. *The left figure shows the first seven of the* $m = 100$ *eigenvalues* $\hat{\lambda}_i$ *from the active subspace analysis; see* (2.2). *The large gaps between eigenvalues* $\hat{\lambda}_1$, $\hat{\lambda}_2$ *and eigenvalues* $\hat{\lambda}_2$, $\hat{\lambda}_3$ *indicate a one-dimensional and a two-dimensional active subspace. The right figure shows the estimated subspace error* ε *from* (2.12) *using the bootstrap as described in* [11].

for a two-dimensional active subspace than for a one-dimensional active subspace.

5.5. Spatial sensitivity. Figures 5.6(a) and 5.6(b) show the first and second eigenvector components from $\hat{\boldsymbol{C}}$ in (2.11). Recall the interpretation of these eigenvectors: perturbing $\mathbf{x}$ along some linear combination of the first two eigenvectors changes the misfit more, on average, than perturbing $\mathbf{x}$ in a direction orthogonal to the span



(a) First eigenvector

(b) Second eigenvector

(c) Coefficient field with first eigenvector

(d) Coefficient field with second eigenvector

FIG. 5.6. *The top two figures show the first and second eigenvector from* $\hat{\boldsymbol{C}}$ *in* (2.11)*. The bottom two figures show the corresponding log-coefficient fields when these eigenvectors are plugged into* $\mathbf{x}$ *in* (5.4)*. These fields can be treated as quantifying the spatial sensitivity of the misfit. (Colors are visible in the electronic version.)*

of the first two eigenvectors. These eigenvectors quantify global sensitivity of the misfit to the parameters. If we use the eigenvector components as $\mathbf{x}$ in (5.4), we get a spatially varying perturbation of the log-coefficients $\log(a)$ that changes the misfit the most, on average. The spatial perturbation corresponding to the first eigenvector is in Figure 5.6(c), and the spatial perturbation corresponding to the second eigenvector is in Figure 5.6(d). The sensitivity is larger near the boundary where the observations are taken.

5.6. Applying the active subspace-accelerated MCMC. With the eigenvectors $\hat{\boldsymbol{W}}$ from (2.11) and the choice of $n = 2$ (i.e., a two-dimensional active subspace)—justified by the eigenvalues (Figure 5.5(a)) and subspace error estimates (Figure 5.5(b))—we can apply the active subspace-accelerated MCMC in Algorithm 1. To choose the number M of Monte Carlo samples in step 2, we perform the following experiment:

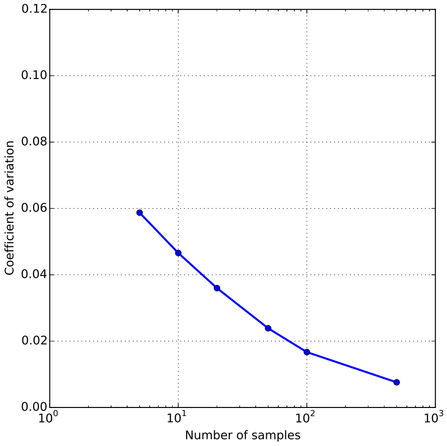


FIG. 5.7. *Average coefficient of variation as a function of the number M of samples for* 100 *Monte Carlo estimates of the conditional expectation $\hat{g}_\varepsilon(\hat{\mathbf{y}})$ from* (2.16)*. We use $M = 10$ in step* 2 *of Algorithm* 1*, which gives one-to-two digits of accuracy. (Colors are visible in the electronic version.)*

1. Draw 100 $\mathbf{x}$'s independently at random according to the Gaussian prior.
2. For each $\mathbf{x}$, let $\hat{\mathbf{y}} = \hat{\boldsymbol{W}}_1^T\mathbf{x}$, and compute the coefficient of variation for the Monte Carlo estimates,

$$\frac{\left(\frac{1}{M-1}\sum_{i=1}^{M}(f(\hat{\boldsymbol{W}}_1\hat{\mathbf{y}} + \hat{\boldsymbol{W}}_2\hat{\mathbf{z}}_i) - \hat{g}_\varepsilon(\hat{\mathbf{y}}))^2\right)^{1/2}}{\sqrt{M}\,\hat{g}_\varepsilon(\hat{\mathbf{y}})}, \tag{5.6}$$

   where $\hat{g}_\varepsilon(\hat{\mathbf{y}})$ is from (2.16), for $M =$1, 5, 10, 20, 50, 100, and 500.
3. For each M, average the coefficients of variation over all $\mathbf{x}$'s.

Figure 5.7 shows the average coefficients of variation as a function of M. We choose $M = 10$ for step 2 in Algorithm 1, which is sufficient for one to two digits of accuracy from the Monte Carlo estimates $\hat{g}_\varepsilon(\hat{\mathbf{y}})$.

We use a standard Gaussian proposal density in step 1 of Algorithm 1. We compare results from two variants of the active subspace-accelerated MCMC: (i) a two-dimensional active subspace and a proposal variance of 0.1, and (ii) a two-dimensional active subspace with a proposal variance of 0.3. Each case ran 50k steps of the Markov chain, which used 500k forward model evaluations and 120 CPU hours. We discard 10k steps as a burn-in. For each sample $\hat{\mathbf{y}}_k$ from the MCMC, we use $P = 10$ independent samples of the 98-dimensional inactive variables $\hat{\mathbf{z}}$ drawn according to their standard Gaussian prior as in (4.2) to construct a chain on the 100-dimensional parameter space.

We compare the results to a standard Metropolis–Hastings MCMC in all $m = 100$ dimensions with a Gaussian proposal density with variance 0.1. We refer to this chain as the *vanilla* case. The 100-dimensional Markov chain took 500k steps, which corresponds to 500k forward model evaluations, and we discarded 100k steps as a burn-in. We do not compare the active subspace-accelerated method to other MCMC variants, because any of those variants can exploit the active subspace the same way Algorithm 1 does.

Table 5.1 displays several characteristics of the Markov chains for the three cases. The first two rows show the proposal density dimension and variance. The two-dimensional chains on the active variables $\hat{\mathbf{y}}$ use 50k steps, while the 100-dimensional

Table 5.1

The first column shows statistics for standard Metropolis–Hastings MCMC on all 100 *dimensions, i.e., the vanilla case. The second two columns show cases of Algorithm* 1 *that exploit the active subspace. The first row is the dimension of the proposal density. The second row is the proposal variance. The third row is the number of steps in the MCMC, and the fourth row shows the number of forward model evaluations. The fifth row shows the acceptance rate. The sixth row shows the smallest effective sample size over the two active variables for chains operating in the active subspace. The last row shows the smallest effective sample size over* 100 *components of the chain in the original* 100*-dimensional space.*

	Vanilla	AS2, 0.1	AS2, 0.3
Proposal dimension	100	2	2
Proposal variance	0.1	0.3	0.1
Number of steps	500k	50k	50k
Number of forward models	500k	500k	500k
Acceptance rate	60%	80%	62%
Min. eff. sample size $\hat{\mathbf{y}}$	N/A	198	1053
Min. eff. sample size $\mathbf{x}$	604	81986	47281

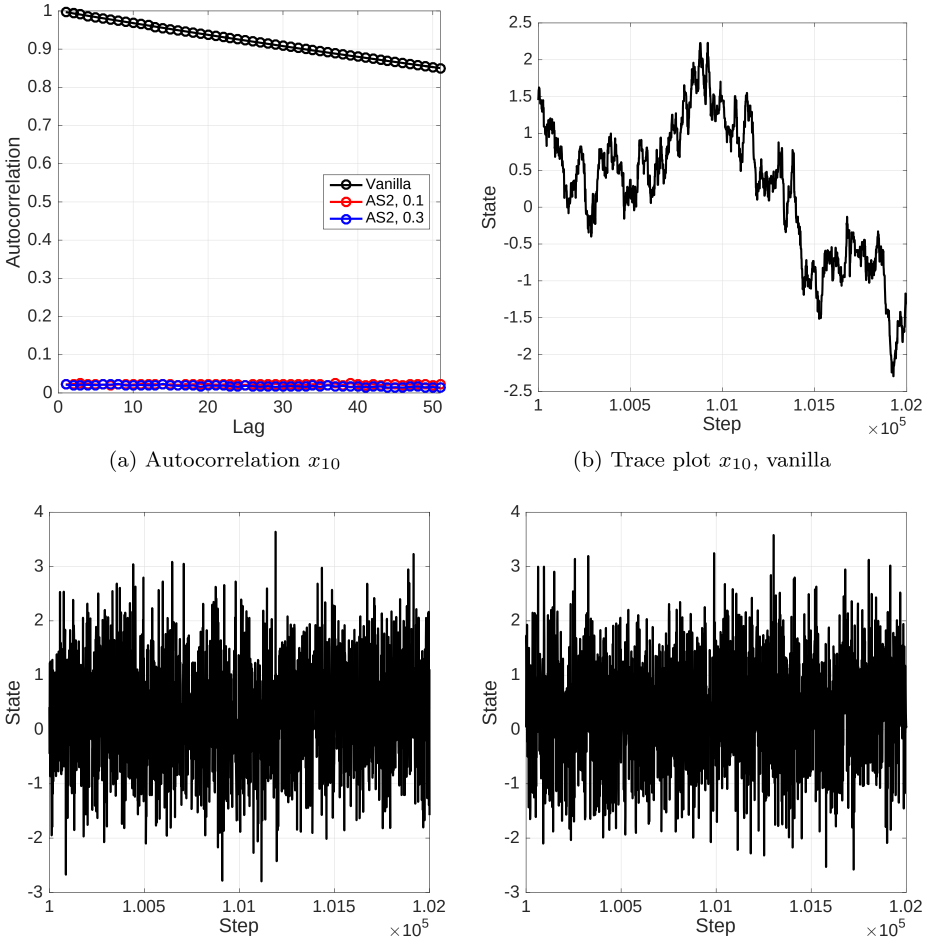
chain uses 500k steps. Each evaluation of the approximate likelihood $\hat{g}_\varepsilon(\hat{\mathbf{y}})$ in step 2 of Algorithm 1 uses $M = 10$ forward model evaluations, so the total number of forward model evaluations is the same (500k) across all chains. The fifth row shows the acceptance rates for the chains. Note that the two-dimensional chain can use a larger proposal variance while maintaining a comparable acceptance rate to the vanilla case. We compute the effective sample size of the chain's components as

$$\frac{N_{\text{steps}}}{1 + 2\sum_{k=1}^{2000} \rho_k}, \tag{5.7}$$

where N_{steps} is the number of steps in the chain, and ρ_k is the autocorrelation with lag k. The sixth row of Table 5.1 shows the minimum effective sample size over the two components of the two-dimensional chain in the coordinates of the active subspace. The last row shows the minimum effective sample size over all 100 components of the chains on the full 100-dimensional parameter space. The much larger effective sample sizes for the active subspace-accelerated chains is due to the independent sampling according to the prior on the inactive variables.

Figure 5.8(a) shows the autocorrelation function for the 10th component of $\mathbf{x}$ for all three chains on the 100-dimensional parameter space: the standard MCMC in 100 dimensions (vanilla), the active subspace-accelerated MCMC in 2 dimensions with proposal variance 0.1 (AS2, 0.1), and the active subspace-accelerated MCMC in 2 dimensions with proposal variance 0.3 (AS2, 0.3). The other components of $\mathbf{x}$ had similar autocorrelation functions. The slow decay in the vanilla case is due to the Markov chain operating in all 100 dimensions, while the active subspace accelerated cases run the Markov chain in only two dimensions and draw the remaining components independently according to the prior. Thus, the iterates in $\mathbf{x}$ appear uncorrelated. The remaining subplots in Figure 5.8 show trace plots of $\mathbf{x}$'s 10th component for all three cases. The active subspace-accelerated method mixes much better in the space of $\mathbf{x}$ due to the independent sampling of $\hat{\mathbf{z}}$; this faster mixing justifies the term *accelerated* in the title.

Figure 5.8 and Table 5.1 suggest that the active subspace-accelerated MCMC mixes much faster than the vanilla MCMC. But are its iterates producing correlated samples from a density close to the true posterior? MCMC convergence metrics can be difficult to interpret for a single chain, so comparing results from different chains is especially challenging. The reader should treat the following results as qualitative,



(a) Autocorrelation x_{10}

(b) Trace plot x_{10}, vanilla

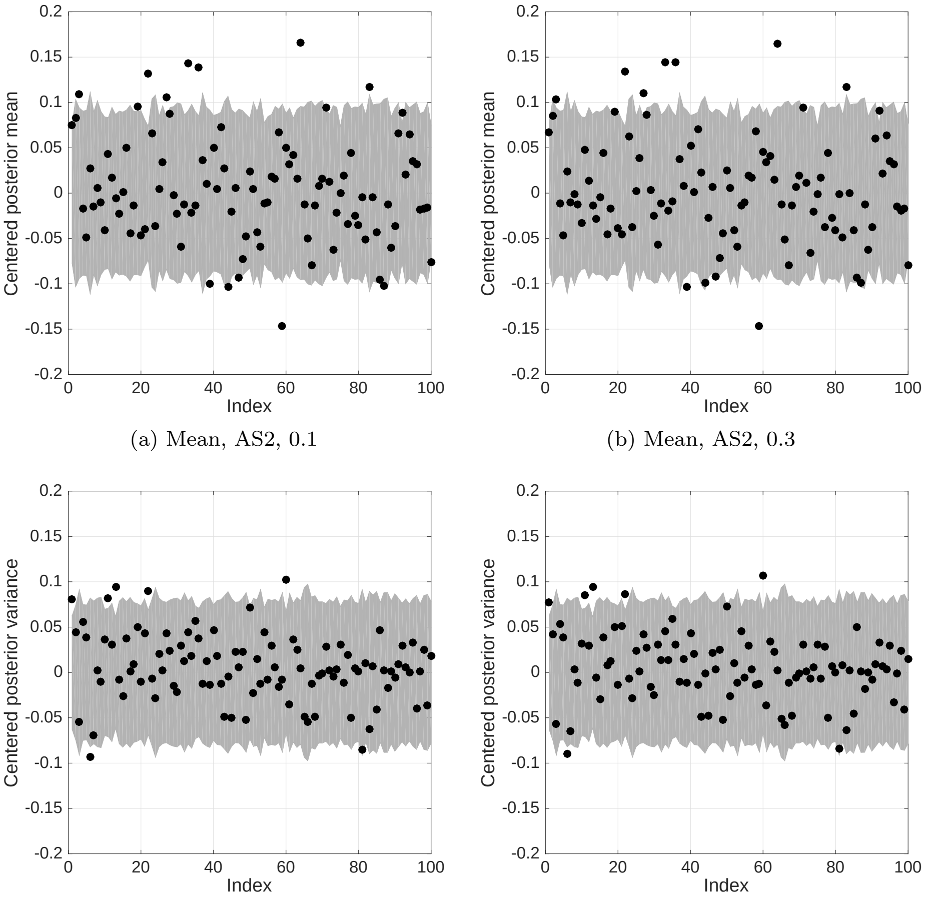
(c) Trace plot x_{10}, AS2, 0.1

(d) Trace plot x_{10}, AS2, 0.3

Fig. 5.8. *The top left figure shows the autocorrelation function for the* 10*th component of* $\mathbf{x}$ *in three cases:* (i) *the Metropolis–Hastings MCMC in* 100 *dimensions (vanilla),* (ii) *Algorithm* 1 *with a two-dimensional subspace and proposal variance* 0.1 (AS2, 0.1)*, and* (iii) *Algorithm* 1 *with a two-dimensional subspace and proposal variance* 0.3 (AS2, 0.3)*. The remaining figures show trace plots of* x_{10} *for all three cases. The active subspace-accelerated MCMC produces Markov chain iterates with much smaller correlation. (Colors are visible in the electronic version.)*

since any quality metrics for the vanilla MCMC are computed from a 100-dimensional chain with high autocorrelation and low effective sample size; see Table 5.1. We perform the following test to check the mean and variance of the iterates produced by the active subspace-accelerated MCMC. We first compute asymptotically valid 99% confidence intervals on the posterior mean and variance from the vanilla MCMC using *consistent batch means* as in section 3.1 of Flegal, Haran, and Jones [16] with parameter $\theta = 2/3$. Let $\hat{\mu} \in \mathbb{R}^{100}$ be the sample posterior mean, and let $\hat{\sigma}^2 \in \mathbb{R}^{100}$ be the sample posterior variance—both computed from the vanilla MCMC. Denote the confidence intervals

$$\hat{\mu}_\ell \leq \hat{\mu} \leq \hat{\mu}_u, \qquad \hat{\sigma}^2_\ell \leq \hat{\sigma}^2 \leq \hat{\sigma}^2_u, \tag{5.8}$$



(a) Mean, AS2, 0.1

(b) Mean, AS2, 0.3

(c) Variance, AS2, 0.1

(d) Variance, AS2, 0.3

FIG. 5.9. *These figures show the sample moments (black dots) of the active subspace-accelerated chains shifted by the sample moments from the vanilla chain; the top row shows sample means and the bottom row shows sample variances. The gray regions are the asymptotically valid* 99% *confidence intervals, shifted by the sample moments, computed with consistent batch means as in* [16]. *The left column shows the accelerated chain with proposal variance* 0.1, *and the right column shows the chain with proposal variance* 0.3.

where the inequalities are interpreted componentwise. Denote the sample mean and variance from the active subspace-accelerated chains as $\hat{\mu}_{\mathrm{as}}$ and $\hat{\sigma}^2_{\mathrm{as}}$, respectively. Figure 5.9 compares the shifted moments, $\hat{\mu}_{\mathrm{as}}-\hat{\mu}$ and $\hat{\sigma}^2_{\mathrm{as}}-\hat{\sigma}^2$, to the shifted confidence intervals,

$$(5.9) \qquad [\hat{\mu}_\ell - \hat{\mu},\, \hat{\mu}_u - \hat{\mu}], \qquad [\hat{\sigma}^2_\ell - \hat{\sigma}^2,\, \hat{\sigma}^2_u - \hat{\sigma}^2].$$

Shifting by the sample mean and variance allows easier visual comparison. Figures 5.9(a) and 5.9(c) show (i) the shifted mean and variance (black dots), respectively, for the active subspace-accelerated chain with proposal variance 0.1 and (ii) the shifted confidence intervals (gray region). Figures 5.9(b) and 5.9(d) show the same shifted moments for the accelerated chain with proposal variance 0.3. We also computed the consistent batch means-based confidence intervals for the accelerated chain's moments, but the intervals were very small (within the marker size), so we do not include them in the plots. The small confidence intervals for the accelerated chain's moments are

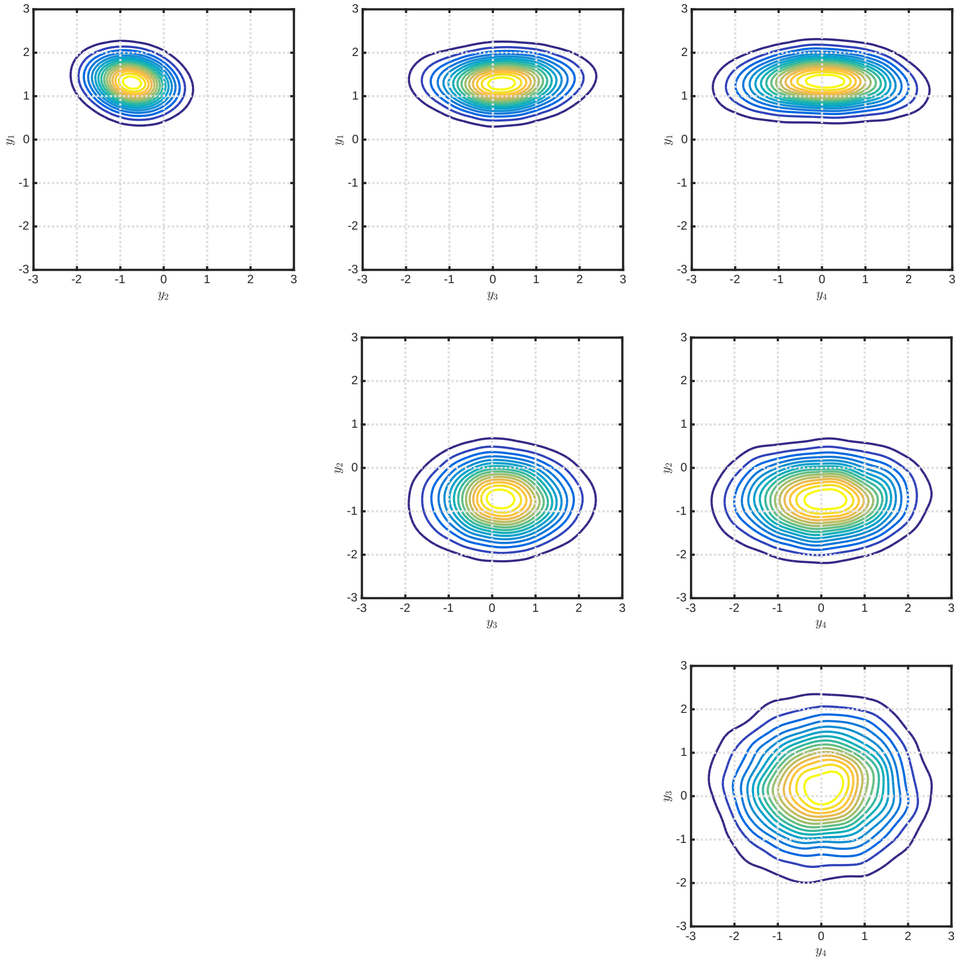


Fig. 5.10. *Bivariate posterior marginals for the first four components from the transformed variables* $\hat{\mathbf{y}}_k = \hat{\boldsymbol{W}}^T\mathbf{x}_k$*, where the set* $\{\mathbf{x}_k\}$ *contains samples from the vanilla MCMC in* 100 *dimensions with proposal variance* 0.1*. (Colors are visible in the electronic version.)*

consistent with the observed rapid mixing. The bulk of the black dots falls within the confidence region, which indicates general agreement in the sample moments. The dots that fall outside the confidence region may indicate bias introduced by the active subspace-based dimension reduction.

We perform one final qualitative check for the two-dimensional structure using the iterates from the 100-dimensional vanilla MCMC. With the eigenvectors $\hat{\boldsymbol{W}}$ from (2.11), we can transform all iterates $\mathbf{x}_k$ from the vanilla chain to the space of active variables, $\hat{\mathbf{y}}_k = \hat{\boldsymbol{W}}^T\mathbf{x}_k$. In this transformation, $\hat{\mathbf{y}}_k$ has 100 components, but they are ordered like the eigenvalues of $\hat{\boldsymbol{C}}$. In other words, the first two components of $\hat{\mathbf{y}}_k$ are comparable to the active variables. Figure 5.10 shows contours of the kernel density estimates of the bivariate marginals using the first four components from the samples $\hat{\mathbf{y}}_k$; the axis labels indicate which components are shown. Note that the active subspace was not employed to generate these samples; they are from the vanilla MCMC. Nevertheless, the density estimate for first two active variable components, $\hat{y}_1$

and $\hat{y}_2$, departs from the standard Gaussian prior. In other words, the data updates this density from the prior. In contrast, the density estimate for components $\hat{y}_3$ and $\hat{y}_4$ more closely resembles the standard Gaussian prior; this pattern continues for the remaining $\hat{\mathbf{y}}$ components. Postprocessing the vanilla MCMC iterates with the eigenvectors $\hat{\boldsymbol{W}}$ validates the two-dimensional structure suggested by the eigenvalues in Figure 5.5(a), providing further confirmation that the data informs only a two-dimensional subspace in the 100-dimensional parameter space.

6. Summary and conclusions.

We have shown how to use the active subspace—derived from the scalar-valued data misfit and defined with respect to the prior—to accelerate MCMC for high-dimensional Bayesian inverse problems with nonlinear parameter-to-observable maps. One can estimate the active subspace as a preprocessing step before beginning the MCMC. Since the active subspace is defined with respect to the prior, its components can be estimated in parallel. If the low-dimensional structure is apparent, then the MCMC can be run on only the active variables, which correspond to the subspace informed by the data. The inactive variables are constrained by the prior; they can be sampled independently as a postprocessing step to reconstruct a chain on the original parameter space. We have bounded the error in the Hellinger distance between the true posterior and its approximation with the active subspace. The bound is in terms of the eigenvalues and the error the numerically estimated active subspace basis. We demonstrated the approach on (i) a two-dimensional example with a quadratic forward model and one-dimensional active subspace and (ii) a 100-dimensional example with a PDE-based forward model and a two-dimensional active subspace.

The proposed approach has several limitations that practitioners should consider when evaluating its appropriateness for their own Bayesian inverse problems. The eigenvalues of $\boldsymbol{C}$ derived from the misfit's gradient provide evidence of exploitable low-dimensional structure. However, these eigenvalues may be misleading for a suitably irregular parameter-to-observable map. The eigenvectors associated with small eigenvalues are reasonable directions to ignore in the parameter space *on average*. A parameter-to-observable map with large, localized variability in a high-dimensional parameter space may yield an estimated active subspace that misses important variation in the misfit. (Such irregular models cause problems for most inversion methods.) Moreover, since the active subspace is derived from the misfit's gradient, a misfit with large gradients but small variation (such as an oscillating function) may have an active subspace that favors directions of oscillation—even when other directions are more appropriate for exploring the range of misfit values. We have not encountered this scenario in practice, but it is possible to construct such functions; see [8, Chapter 1]. For a linear forward model, when the given data is in the tails of the forward model's likely outputs (i.e., propagated according to the prior), the active subspace may be unduly influenced by the data vector. This insight is not directly relevant for MCMC, since MCMC is not appropriate for linear forward models. However, it suggests that deep understanding of how the data affect the active subspace may be needed for some nonlinear forward models; we intend to pursue such analysis in future studies.

The practitioner should also be aware of scaling limitations of the proposed approach as the problem's dimensions increase. Without additional acceleration (e.g., surrogate forward models or alternative structure-exploiting techniques), random walk MCMC is practically limited to inference on a handful of parameters in expensive nonlinear simulation models. If the active subspace for a given problem has dimension greater than 5 to 10, then the practitioner should consider a more sophisticated approach than random walk MCMC for efficient inference. For PDE-based inverse problems, the holy grail is to infer a spatially varying parameter field, where the number of parameters is the number of spatial discretization cells—which may be in the millions for modern computational models. In the example from section 5.2, we first assumed we could reduce the number of parameters from 10000 to 100 using known correlation structure in the parameters. Without this reduction, we may have needed thousands of realizations of the 10000-component gradient vector to estimate the eigenpairs of $\boldsymbol{C}$. Such computations are large enough to stress workstation-sized computers. A more spatially refined computation—or one that originates from a PDE in three spatial dimensions—would cause more stress. Moreover, there is no guarantee that the misfit's active subspace dimension would be small enough to permit efficient MCMC. Therefore, for practical PDE-based inverse problems, we expect that estimating and exploiting an active subspace in the misfit would be one tool in a comprehensive toolbox for Bayesian inference.

There are other variants of active subspace-accelerated MCMC that are worth studying that we did not explore in this paper. For instance, one could define the active subspace using the posterior instead of the prior as the integration measure; this is similar to the likelihood-informed subspace [14]. Using the posterior would allow the proper interpretation of the subspace components as conditional random variables, but it would be more computationally expensive. Alternatively, one could use the joint density of the data and parameters in place of the prior, which would produce a data independent subspace. Another idea is to use the likelihood directly as the differentiable, scalar-valued function, instead of its negative log (i.e., the misfit). Lastly, there may be a way to combine subspaces from the state covariance, the matrix $\boldsymbol{C}$ from (2.2), and the average Hessian as in [14] to produce a more robust dimension reduction. A thorough quantitative comparison to likelihood-informed subspace is beyond the scope of the current manuscript, but such a comparison would be worthwhile.

Appendix A. Proof of Theorem 3.1. First, we carefully work out the derivation for (3.10), which uses the approximation g from (2.7). The following quantities depend on $\mathbf{x}$:

$$\rho_{\text{pos}} = \rho_{\text{pos}}(\mathbf{x}), \quad \pi = \pi(\mathbf{x}), \quad \rho_{\text{pri}} = \rho_{\text{pri}}(\mathbf{x}), \quad f = f(\mathbf{x}), \quad g = g(\boldsymbol{W}_1^T\mathbf{x}). \tag{A.1}$$

In the next derivation, we omit the explicit dependence on $\mathbf{x}$ to keep the notation clean.

(A.2)

$$H^2(\rho_{\text{pos}},\pi) = \frac{1}{2}\int \left((\rho_{\text{pos}})^{\frac{1}{2}} - (\pi)^{\frac{1}{2}} \right)^2 d\mathbf{x}$$

$$= \frac{1}{2}\int \left(\left(\frac{\exp(-f)\,\rho_{\text{pri}}}{c_{\text{pos}}} \right)^{\frac{1}{2}} - \left(\frac{\exp(-g)\,\rho_{\text{pri}}}{c_{\pi}} \right)^{\frac{1}{2}} \right)^2 d\mathbf{x} \tag{A.3}$$

$$= \frac{1}{2}\int \left(\left(\frac{\exp(-f)}{c_{\text{pos}}} \right)^{\frac{1}{2}} - \left(\frac{\exp(-g)}{c_{\pi}} \right)^{\frac{1}{2}} \right)^2 \rho_{\text{pri}}\, d\mathbf{x} \tag{A.4}$$

$$= \frac{1}{2(c_{\text{pos}}\,c_{\pi})^{\frac{1}{2}}} \left[\int \left((\exp(-f))^{\frac{1}{2}} - (\exp(-g))^{\frac{1}{2}} \right)^2 \rho_{\text{pri}}\, d\mathbf{x} - \left(c_{\text{pos}}^{\frac{1}{2}} - c_{\pi}^{\frac{1}{2}} \right)^2 \right] \tag{A.5}$$

$$\leq \frac{1}{2(c_{\text{pos}}\,c_{\pi})^{\frac{1}{2}}} \int \left((\exp(-f))^{\frac{1}{2}} - (\exp(-g))^{\frac{1}{2}} \right)^2 \rho_{\text{pri}}\, d\mathbf{x} \tag{A.6}$$

$$= \frac{1}{2(c_{\mathrm{pos}}\, c_\pi)^{\frac{1}{2}}} \int \left(\exp\left(\frac{-f}{2}\right) - \exp\left(\frac{-g}{2}\right)\right)^2 \rho_{\mathrm{pri}}\, d\mathbf{x} \tag{A.7}$$

$$\le \frac{1}{2(c_{\mathrm{pos}}\, c_\pi)^{\frac{1}{2}}} \int \left(\frac{1}{2}\,(f-g)\right)^2 \rho_{\mathrm{pri}}\, d\mathbf{x} \tag{A.8}$$

$$\le \frac{C}{8(c_{\mathrm{pos}}\, c_\pi)^{\frac{1}{2}}} (\lambda_{n+1} + \cdots + \lambda_m) \tag{A.9}$$

Line (A.2) is the definition of the squared Hellinger distance. Line (A.3) plugs in the definitions of the posterior ρ_{pos} and approximate posterior π in terms of the misfit and its approximation. Line (A.4) factors out the prior. Line (A.5) is verified by inspection using the definitions of c_{pos} (3.3) and c_π (3.8). Line (A.6) follows since the omitted squared term is positive. Line (A.8) follows from the mean value theorem and the fact that $|\exp(-x)| \le 1$ for $x \ge 0$. The last line follows from Theorem 2.1.

The constant c_π is bounded below using Jensen's inequality,

$$\begin{aligned} c_\pi &= \int \exp(-g)\, \rho_{\mathrm{pri}}\, d\mathbf{x} \\ &\ge \exp\left(- \int g\, \rho_{\mathrm{pri}}\, d\mathbf{x}\right) \\ &= \exp\left(- \int f\, \rho_{\mathrm{pri}}\, d\mathbf{x}\right), \end{aligned} \tag{A.10}$$

where the last line follows from the construction of g in (2.7). Then the constant from (A.9) can be bounded,

$$\frac{1}{8}(c_{\mathrm{pos}}\, c_\pi)^{\frac{-1}{2}} \;\le\; \frac{1}{8}\left[c_{\mathrm{pos}}\, \exp\left(- \int f\, \rho_{\mathrm{pri}}\, d\mathbf{x}\right)\right]^{\frac{-1}{2}}. \tag{A.11}$$

Recalling the definition of c_{pos} from (3.3) with the definition of the misfit,

$$c_{\mathrm{pos}} \;=\; \int \exp(-f)\, \rho_{\mathrm{pri}}\, d\mathbf{x}, \tag{A.12}$$

which completes the proof of (3.10).

Equations (A.2) through (A.8) are identical if $\hat{\pi}$ replaces π, $\hat{g}$ replaces g, and $c_{\hat{\pi}}$ replaces c_π. The constant $c_{\hat{\pi}}$ is bounded as

$$\begin{aligned} c_{\hat{\pi}} &= \int \exp(-\hat{g})\, \rho_{\mathrm{pri}}\, d\mathbf{x} \\ &\ge \exp\left(- \int \hat{g}\, \rho_{\mathrm{pri}}\, d\mathbf{x}\right) \\ &= \exp\left(- \int f\, \rho_{\mathrm{pri}}\, d\mathbf{x}\right), \end{aligned} \tag{A.13}$$

where the last line follows from the unbiasedness of the Monte Carlo approximation (2.9). Then (A.11) holds with $c_{\hat{\pi}}$ replacing c_π. Finally, using Theorem 2.2 recovers (3.11). Lines (3.12) and (3.13) use identical reasoning with the estimated eigenvectors $\hat{\boldsymbol{W}}$ and Theorems 2.3 and 2.4.

Acknowledgments. We thank Luis Tenorio and Aaron Porter at Colorado School of Mines and Youssef Marzouk and Tiangang Cui at MIT for their helpful comments.

REFERENCES

[1] A. Apte, M. Hairer, A. Stuart, and J. Voss, *Sampling the posterior: An approach to non-Gaussian data assimilation*, Phys. D, 230 (2007), pp. 50–64, http://dx.doi.org/10.1016/j.physd.2006.06.009.

[2] Z. Botev, *kde2d*. http://www.mathworks.com/matlabcentral/fileexchange/17204-kde2d-data-n-min-xy-max-xy, (2007).

[3] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, *Handbook of Markov Chain Monte Carlo*, CRC Press, Boca Raton, FL, 2011.

[4] T. Bui-Thanh, C. Burstedde, O. Ghattas, J. Martin, G. Stadler, and L. Wilcox, *Extreme-scale UQ for Bayesian inverse problems governed by PDEs*, in Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, Los Alamitos, CA, IEEE, 2012, pp. 3:1–3:11, http://dl.acm.org/citation.cfm?id=2388996.2389000.

[5] T. Bui-Thanh and M. Girolami, *Solving large-scale PDE-constrained Bayesian inverse problems with Riemann manifold Hamiltonian Monte Carlo*, Inverse Problems, 30 (2014), 114014, http://stacks.iop.org/0266-5611/30/i=11/a=114014.

[6] D. Calvetti and E. Somersalo, *Introduction to Bayesian Scientific Computing: Ten Lectures on Subjective Computing*, Springer, New York, 2007.

[7] L. Chen, *An inequality for the multivariate normal distribution*, J. Multivariate Anal., 12 (1982), pp. 306–315.

[8] P. Constantine, *Active Subspaces: Emerging Ideas in Dimension Reduction for Parameter Studies*, SIAM, Philadelphia, 2015.

[9] P. Constantine, E. Dow, and Q. Wang, *Active subspace methods in theory and practice: Applications to kriging surfaces*, SIAM J. Sci. Comput., 36 (2014), pp. A1500–A1524, http://dx.doi.org/10.1137/130916138.

[10] P. Constantine, M. Emory, J. Larsson, and G. Iaccarino, *Exploiting active subspaces to quantify uncertainty in the numerical simulation of the HyShot* II *scramjet*, J. Comput. Phys., 302 (2015), pp. 1–20, http://dx.doi.org/10.1016/j.jcp.2015.09.001.

[11] P. Constantine and D. Gleich, *Computing active subspaces with Monte Carlo*, preprint, arXiv:1408.0545v2, 2015.

[12] P. Constantine, B. Zaharatos, and M. Campanelli, *Discovering an active subspace in a single-diode solar cell model*, Stat. Anal. Data Min. (2015), http://dx.doi.org/10.1002/sam.11281

[13] T. Cui, K. Law, and Y. Marzouk, *Dimension-independent likelihood-informed MCMC*, J. Comput. Phys., 304 (2016), pp. 109–137, http://dx.doi.org/10.1016/j.jcp.2015.10.008.

[14] T. Cui, J. Martin, Y. Marzouk, A. Solonen, and A. Spantini, *Likelihood-informed dimension reduction for nonlinear inverse problems*, Inverse Problems, 30 (2014), 114015, http://stacks.iop.org/0266-5611/30/i=11/a=114015.

[15] H. Flath, L. Wilcox, V. Akçelik, J. Hill, B. van Bloemen Waanders, and O. Ghattas, *Fast algorithms for Bayesian uncertainty quantification in large-scale linear inverse problems based on low-rank partial Hessian approximations*, SIAM J. Sci. Comput., 33 (2011), pp. 407–432, http://dx.doi.org/10.1137/090780717.

[16] J. Flegal, M. Haran, and G. Jones, *Markov chain Monte Carlo: Can we trust the third significant figure?*, Statist. Sci., 23 (2008), pp. 250–260, http://www.jstor.org/stable/27645897.

[17] A. Gibbs and F. Su, *On choosing and bounding probability metrics*, Internat. Statistic. Rev., 70 (2002), pp. 419–435, http://dx.doi.org/10.1111/j.1751-5823.2002.tb00178.x.

[18] M. Girolami and B. Calderhead, *Riemann manifold Langevin and Hamiltonian Monte Carlo methods*, J. Roy. Statist. Soc. Ser. B, 73 (2011), pp. 123–214, http://dx.doi.org/10.1111/j.1467-9868.2010.00765.x.

[19] G. Golub and C. V. Loan, *Matrix Computations*, 4th ed., Johns Hopkins University Press, Baltimore, 2013.

[20] H. Haario, M. Laine, A. Mira, and E. Saksman, *DRAM: Efficient adaptive MCMC*, Statist. Comput., 16 (2006), pp. 339–354, http://dx.doi.org/10.1007/s11222-006-9438-0.

[21] J. Jefferson, J. Gilbert, P. Constantine, and R. Maxwell, *Active subspaces for sensitivity analysis and dimension reduction of an integrated hydrologic model*, Comput. Geosci., 83 (2015), pp. 127–138, http://dx.doi.org/10.1016/j.cageo.2015.07.001.

[22] J. Kaipio and E. Somersalo, *Statistical and Computational Inverse Problems*, Springer, New York, 2005.

[23] T. Lukaczyk, P. Constantine, F. Palacios, and J. Alonso, *Active Subspaces for Shape Optimization*, in Proceedings of the 10th AIAA Multidisciplinary Design Optimization Conference, National Harbor, Maryland, 2014, http://dx.doi.org/doi:10.2514/6.2014-1171.

[24] J. Martin, L. Wilcox, C. Burstedde, and O. Ghattas, *A stochastic Newton MCMC method for large-scale statistical inverse problems with application to seismic inversion*, SIAM J. Sci. Comput., 34 (2012), pp. A1460–A1487, http://dx.doi.org/10.1137/110845598.

[25] Y. Marzouk and H. Najm, *Dimensionality reduction and polynomial chaos acceleration of Bayesian inference in inverse problems*, J. Comput. Phys., 228 (2009), pp. 1862–1902, http://dx.doi.org/10.1016/j.jcp.2008.11.024.

[26] A. Spantini, A. Solonen, T. Cui, J. Martin, L. Tenorio, and Y. Marzouk, *Optimal low-rank approximations of Bayesian linear inverse problems*, SIAM J. Sci. Comput., 37 (2015), A2451–A2487, http://dx.doi.org/10.1137/140977308.

[27] A. Stuart, *Inverse problems: A Bayesian perspective*, Acta Numer., 19 (2010), pp. 451–559, http://dx.doi.org/10.1017/S0962492910000061.

[28] J. Vrugt, C. Ter Braak, C. Diks, B. Robinson, J. Hyman, and D. Higdon, *Accelerating Markov chain Monte Carlo simulation by differential evolution with self-adaptive randomized subspace sampling*, Int. J. Nonlinear Sci. Numer. Simul., 10 (2009), pp. 273–290.

[29] Q. Wang, *Numpad: Numerical prototyping in Python assisted by automatic differentiation.* https://github.com/qiqi/numpad (2014).