INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2010; **81**:1581–1608
Published online 18 September 2009 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme.2746

Non-linear model reduction for uncertainty quantification in large-scale inverse problems

D. Galbally[1], K. Fidkowski[2,*,†], K. Willcox[3] and O. Ghattas[4]

[1]*Iberdrola, Madrid 28033, Spain*
[2]*University of Michigan, 1320 Beal Avenue, 3029 François-Xavier Bagnoud Building,*
Ann Arbor, MI 48109, U.S.A.
[3]*Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A.*
[4]*University of Texas, Austin, TX 78712, U.S.A.*

SUMMARY

We present a model reduction approach to the solution of large-scale statistical inverse problems in a Bayesian inference setting. A key to the model reduction is an efficient representation of the non-linear terms in the reduced model. To achieve this, we present a formulation that employs masked projection of the discrete equations; that is, we compute an approximation of the non-linear term using a select subset of interpolation points. Further, through this formulation we show similarities among the existing techniques of gappy proper orthogonal decomposition, missing point estimation, and empirical interpolation via coefficient-function approximation. The resulting model reduction methodology is applied to a highly non-linear combustion problem governed by an advection–diffusion-reaction partial differential equation (PDE). Our reduced model is used as a surrogate for a finite element discretization of the non-linear PDE within the Markov chain Monte Carlo sampling employed by the Bayesian inference approach. In two spatial dimensions, we show that this approach yields accurate results while reducing the computational cost by several orders of magnitude. For the full three-dimensional problem, a forward solve using a reduced model that has high fidelity over the input parameter space is more than two million times faster than the full-order finite element model, making tractable the solution of the statistical inverse problem that would otherwise require many years of CPU time. Copyright © 2009 John Wiley & Sons, Ltd.

Received 10 February 2009; Revised 12 June 2009; Accepted 29 July 2009

KEY WORDS: non-linear model reduction; inverse problems; Bayesian inference; advection–diffusion-reaction PDE

[*] Correspondence to: K. Fidkowski, University of Michigan, 1320 Beal Avenue, 3029 François-Xavier Bagnoud Building, Ann Arbor, MI 48109, U.S.A.

[†] E-mail: kfid@umich.edu

Contract/grant sponsor: Singapore-MIT Alliance Computational Engineering Programme
Contract/grant sponsor: Air Force Office of Sponsored Research; contract/grant number: FA9550-06-0271
Contract/grant sponsor: National Science Foundation under DDDAS; contract/grant numbers: CNS-0540372, CNS-0540186
Contract/grant sponsor: Graduate Fellowship from the Caja Madrid Foundation

1. INTRODUCTION

Effective computational tools to support decision-making under uncertainty are essential for the design and operation of engineered systems, as well as for predictive modeling of scientific phenomena. However, characterizing uncertainty for large-scale models is a significant challenge—even with simplified (e.g. two-dimensional) analyses, the many thousands of function evaluations required to sample the uncertainty space (e.g. via Monte Carlo sampling) present an intractable computational burden. Here, we present a model reduction approach, which is a key enabler for uncertainty quantification, via Monte Carlo sampling for problems described by large-scale non-linear systems.

The problem of estimating inputs to a simulation model from observations is fundamentally an *inverse problem*. The *forward problem* seeks to predict the system outputs given the inputs (including parameter, state, or source inputs) by solving the governing equations. The inverse problem, on the other hand, reverses this relationship by seeking to estimate uncertain inputs from measurements or observations. The inverse problem is often formulated as a deterministic non-linear optimization problem that seeks to minimize the misfit between the observed and predicted outputs in an appropriate norm while also minimizing a regularization term that penalizes unwanted features of the inputs. When the forward model is governed by a set of partial differential equations (PDEs), the result is an optimization problem that is large-scale in the state variables, even if the number of inversion variables is small.

Estimation of inputs using the regularization approach to inverse problems as described above will yield an estimate of the 'best' input values that simultaneously fit the data and minimize the regularization penalty term. However, we are interested in not just point estimates of the best-fit inputs, but a complete statistical description of the input values that is consistent with the data. The Bayesian approach does this by reformulating the inverse problem as a problem in statistical inference, incorporating the forward model, prior information on the inputs, and uncertainties in the measurements. The solution of this inverse problem is the 'posterior' joint probability density of the inputs, which reflects the degree of confidence in their values [1, 2]. Thus, we are able to quantify the resulting uncertainty in the inputs, taking into account the model, prior information, and uncertainties in the data.

The Bayesian inverse problem framework, in conjunction with Markov chain Monte Carlo (MCMC) methods for sampling the posterior density, has been employed successfully in numerous settings; for representative applications, see [1]. However, it remains an outstanding challenge to apply these methods to large-scale problems with high-dimensional input spaces, such as those arising from discretization of PDEs, due to the necessity of solving the forward problem at every sample point. Recently, the use of lower-fidelity models has been proposed to decrease the cost of sampling [3, 4], by using spatial discretizations that are coarser than those usually necessary for accurate solution of the forward problem. Unfortunately, the level of mesh coarsening required for decreasing the computational cost of these problems to acceptable levels typically results in large errors that are often hard to quantify or that even yield unstable numerical schemes. Recent work has explored acceleration of the Bayesian solution of inverse problems through the use of stochastic spectral methods [5], exploiting advances in polynomial chaos representations of random variables and processes for forward uncertainty propagation [6, 7]. In [8], a coarse model is used to pre-screen acceptance during MCMC sampling, thereby avoiding the cost of a full model solve on proposals that would be rejected. In this approach, convergence is proven by showing that 'false negatives—that is, rejecting a proposal that would have been accepted if it were evaluated with

DOI: 10.1002/nme

the fine model—do not occur. This property holds only for certain problems with coarse models of a specific nature.

In this work, we pursue the approach of accelerating the solution of the forward problem using projection-based model reduction, in which a model of reduced complexity is derived using projection onto a reduced subspace (as opposed to the coarse-grid reduced models employed in [3, 4]). In particular, we employ the proper orthogonal decomposition (POD) method, which uses an ensemble of representative high-fidelity solutions to derive a set of basis vectors [9–11] onto which the governing equations are projected. Projection of any non-linear terms must be handled carefully, or the result is a reduced model that is of small dimension but whose solution cost still depends on the dimension of the original full model. In [12], model reduction based on the POD is used to accelerate forward model simulations for the solution of a radiative source statistical inversion problem; however, a specific approach to reduce the complexity of the projected non-linear term was not employed in that work.

Efficient non-linear model reduction has been addressed by several authors. If the problem under consideration is weakly non-linear, efficient reduced models can be obtained by retaining low-order terms in the Taylor series expansion of non-linear terms [13]. Another approach is to use the trajectory piecewise-linear scheme, which employs a weighted combination of various linear models, obtained by linearizing the non-linear system at selected points along a state trajectory [14]. Other approaches propose approximating the non-linear term through selective sampling of a subset of the original equations [15–18]. In [16], the missing point estimation approach, based on the theory of gappy POD [19], is used to approximate non-linear terms in the reduced model with selective spatial sampling, while in [17, 18] the empirical interpolation method (EIM) is used to approximate the non-linear terms by a linear combination of empirical basis functions for which the coefficients are determined using interpolation. Here, we present a formulation that employs masked projection of the discrete equations; that is, we compute an approximation of the non-linear term using a select subset of interpolation points. We use this formulation to show similarities among the existing non-linear model reduction techniques discussed above.

Section 2 of this paper describes the form of the general non-linear models considered, while Section 3 presents the inverse problem formulation and solution approach via MCMC sampling. Section 4 describes the non-linear model reduction approach, which in Section 5 is applied to advection–diffusion-reaction systems representing a jet diffusion flame in a combustor. Three-dimensional results for a system of over 8.5 million degrees of freedom demonstrate the scalability of our approach and the immense computational gains that can be achieved using model reduction. Finally, Section 6 concludes the paper.

2. GENERAL NON-LINEAR FORWARD MODEL

We first describe the forward problem that represents the system of interest. To illustrate the applicability of our approach, we present a general non-linear model in discrete form. This general form might result from spatial discretization of a PDE (using, for example, a finite element or finite volume method), or it might represent a set of differential–algebraic equations (such as arises in circuit modeling). The non-linear model is written as

$$\mathbf{R}(\mathbf{u};\boldsymbol{\mu})=\mathbf{R}_0+\mathbf{A}\mathbf{u}+\mathbf{s}(\mathbf{u};\boldsymbol{\mu})=\mathbf{0} \tag{1}$$

DOI: 10.1002/nme

where $\mathbf{u} \in \mathbb{R}^N$ is the discrete state vector of N unknowns, and $\boldsymbol{\mu} \in \mathbb{R}^p$ is the vector of parametric inputs that varies over some parameter domain $\mathscr{D}$. The non-linear residual vector $\mathbf{R} \in \mathbb{R}^N$ in (1) is divided into three components as follows. First, $\mathbf{R}_0 \in \mathbb{R}^N$ is a source term that is independent of the state. Second, $\mathbf{Au}$ is the contribution from those terms in the governing equations that have linear state dependence, where $\mathbf{A} \in \mathbb{R}^{N\times N}$ is a constant (usually sparse) matrix. The third component, $\mathbf{s}(\mathbf{u};\boldsymbol{\mu}) \in \mathbb{R}^N$, is the contribution of all non-linear terms. The specific form of the non-linear term depends on the particular governing equations at hand. The computational cost of solving (1) depends on both the number of states, N, and on the complexity of the non-linear term (which might, for example, involve quadrature approximations of integrals requiring evaluation of the non-linear term at many quadrature points).

Outputs of interest are assumed to be the linear combinations of states (general non-linear outputs could also be considered). Thus, the output equation is given by

$$\mathbf{y}(\mathbf{u}(\boldsymbol{\mu})) = \mathbf{Cu}(\boldsymbol{\mu}) \tag{2}$$

where $\mathbf{y} \in \mathbb{R}^q$ is a vector of q outputs of interest and $\mathbf{C} \in \mathbb{R}^{q\times N}$ is a constant matrix that maps the states to outputs. For simplicity of notation, we will write $\mathbf{y}(\mathbf{u}(\boldsymbol{\mu}))$ as $\mathbf{y}(\boldsymbol{\mu})$.

3. INVERSE PROBLEM

Given a parameter vector $\boldsymbol{\mu} \in \mathscr{D}$, the forward problem involves solving (1) to determine the state $\mathbf{u}$ and applying (2) to calculate the output vector $\mathbf{y} \in \mathbb{R}^q$. Conversely, the associated inverse problem involves determining an unknown parameter vector $\boldsymbol{\mu}^*$ given a set of measurements of the outputs $\bar{\mathbf{y}} \in \mathbb{R}^q$.

A deterministic approach to solving the inverse problem leads to an optimization problem, which yields a 'best' value of the parameter vector by minimizing the discrepancy between the measurements and the numerical prediction. However, to handle the effects of experimental error and/or incomplete knowledge of the parameters due to sparse measurements, a deterministic formulation can at best incorporate error covariances as a weighting in the optimization objective. Thus, deterministic results fail to capture comprehensively how experimental errors propagate through the numerical model underlying the inverse problem, and they do not quantify the uncertainty associated with the point estimate that results from the solution of the deterministic optimization problem. These shortcomings motivate a statistical approach to the inverse problem. In particular, we will pursue a Bayesian formulation, which, rather than determining a single 'best' value, incorporates the effects of measurement uncertainty to yield a probability density function for the parameters $\boldsymbol{\mu}^*$. This section first presents the deterministic inverse problem formulation, and then describes the Bayesian formulation and solution method for the statistical inverse problem. Additional details on the approach can be found in [2, 20].

3.1. Deterministic inverse problem

Given a set of measurements of the output, $\bar{\mathbf{y}} \in \mathbb{R}^q$, the deterministic inverse problem seeks to estimate the input parameters by solving the optimization problem

$$\min_{\mathbf{u},\boldsymbol{\mu}} \quad \|\mathbf{y} - \bar{\mathbf{y}}\|_2^2 \tag{3}$$

$$\text{s. t.}\quad \mathbf{R}(\mathbf{u};\boldsymbol{\mu})=\mathbf{0} \tag{3a}$$

$$\mathbf{y}=\mathbf{C}\mathbf{u} \tag{3b}$$

$$\boldsymbol{\mu}\in\mathscr{D} \tag{3c}$$

This inverse problem is often ill-posed and some form of regularization is needed to ensure a unique solution.

3.2. Statistical inverse problem

The solution to the statistical inverse problem takes the form of a *posterior probability*, $p(\boldsymbol{\mu}^*|\bar{\mathbf{y}})$, which encapsulates the uncertainty in inferring the inputs from the set of measured outputs. Using Bayes' theorem, the posterior probability can be expressed as

$$p(\boldsymbol{\mu}^*|\bar{\mathbf{y}})=\frac{1}{p(\bar{\mathbf{y}})}p(\bar{\mathbf{y}}|\boldsymbol{\mu}^*)p(\boldsymbol{\mu}^*) \tag{4}$$

where $p(\bar{\mathbf{y}}|\boldsymbol{\mu}^*)$ is the *likelihood function*, which is the conditional probability of the outputs given a set of input parameters. The *prior probability*, $p(\boldsymbol{\mu}^*)$, contains all available information on the distribution of $\boldsymbol{\mu}^*$ before any measurements have been incorporated into the inference. The remaining factor $p(\bar{\mathbf{y}})$ is determined by normalization of the posterior probability.

We consider the case where the error in each output measurement is normally distributed with zero mean, that is,

$$\bar{\mathbf{y}}=\mathbf{y}(\boldsymbol{\mu}^e)+\boldsymbol{\varepsilon} \tag{5}$$

where $\boldsymbol{\mu}^e$ is the 'exact' parameter vector that would have generated the measurements in the case of zero noise, and $\boldsymbol{\varepsilon}\in\mathbb{R}^q$ are the normally distributed errors with zero mean and covariance $\boldsymbol{\Gamma}_\varepsilon\in\mathbb{R}^{q\times q}$. In the case that the errors are uncorrelated and all the measurements have the same standard deviation, σ, then $\boldsymbol{\Gamma}_\varepsilon=\sigma^2\mathbf{I}$ and the components of $\boldsymbol{\varepsilon}$ each have distribution $N(0,\sigma^2)$. In this case, the likelihood function is given by

$$p(\bar{\mathbf{y}}|\boldsymbol{\mu})\propto\exp\left[-\frac{1}{2\sigma^2}(\bar{\mathbf{y}}-\mathbf{y}(\boldsymbol{\mu}))^{\mathrm{T}}(\bar{\mathbf{y}}-\mathbf{y}(\boldsymbol{\mu}))\right] \tag{6}$$

In the general case, no information is assumed about $\boldsymbol{\mu}$ except that it must reside in the parameter domain $\mathscr{D}$. Thus, the prior $p(\boldsymbol{\mu})$ is a uniform distribution over $\boldsymbol{\mu}\in\mathscr{D}$. (If additional information about the distribution of $\boldsymbol{\mu}$ were available, it would be incorporated into the prior.) Applying (4), the posterior probability density in the case of uniform prior and Gaussian noise is

$$p(\boldsymbol{\mu}|\bar{\mathbf{y}})\propto\begin{cases}\exp\left[-\dfrac{1}{2\sigma^2}(\bar{\mathbf{y}}-\mathbf{y}(\boldsymbol{\mu}))^{\mathrm{T}}(\bar{\mathbf{y}}-\mathbf{y}(\boldsymbol{\mu}))\right] & \text{if } \boldsymbol{\mu}\in\mathscr{D}\\ 0 & \text{otherwise}\end{cases} \tag{7}$$

3.3. MCMC sampling

The Metropolis–Hastings MCMC algorithm [21] is used to numerically sample the posterior distribution in (7), with the goal of using these samples to estimate statistics of interest. Denote by

$\{\boldsymbol{\mu}^i\}_{i=1}^{N_m}$ a Markov chain of N_m parameter samples. Given a parameter sample $\boldsymbol{\mu}=\boldsymbol{\mu}^i$, the Markov chain algorithm generates a new sample $\boldsymbol{\mu}^{i+1}$ by the following two steps:

1. *Generation step*: Sample $\boldsymbol{\mu}'$ from a proposal distribution, $q(\boldsymbol{\mu}'|\boldsymbol{\mu})$. Also calculate $q(\boldsymbol{\mu}|\boldsymbol{\mu}')$.
2. *Acceptance step*: With probability

$$\alpha(\boldsymbol{\mu}'|\boldsymbol{\mu})=\min\left[1,\frac{p(\boldsymbol{\mu}'|\bar{\mathbf{y}})q(\boldsymbol{\mu}'|\boldsymbol{\mu})}{p(\boldsymbol{\mu}|\bar{\mathbf{y}})q(\boldsymbol{\mu}|\boldsymbol{\mu}')}\right] \tag{8}$$

accept the proposal ($\boldsymbol{\mu}^{i+1}=\boldsymbol{\mu}'$). Otherwise, reject the proposal ($\boldsymbol{\mu}^{i+1}=\boldsymbol{\mu}$).

Note that if the proposal $\boldsymbol{\mu}'$ falls outside the parameter domain $\mathscr{D}$, $p(\boldsymbol{\mu}'|\bar{\mathbf{y}})=0$, and $\boldsymbol{\mu}'$ is immediately rejected. Otherwise, the calculation of $p(\boldsymbol{\mu}'|\bar{\mathbf{y}})$ by (7) requires a forward solve to determine $\mathbf{y}(\boldsymbol{\mu}')$. This forward solve at every step is the most expensive part of the MCMC algorithm, often making MCMC impractical for problems governed by large-scale models. We address this issue in the next section, by deriving an efficient reduced-order model that will be used for the forward solve, cutting the computational time by several orders of magnitude.

The proposal distribution $q(\boldsymbol{\mu}'|\boldsymbol{\mu})$ used in the generation step governs the exploration of the parameter space and has a direct impact on the acceptance probability, α. If $q(\boldsymbol{\mu}'|\boldsymbol{\mu})$ is a very narrow distribution around $\boldsymbol{\mu}$, exploration of the parameter space will proceed very slowly, requiring a large N_m (and thus many forward solves) to obtain useful statistics. Conversely, if $q(\boldsymbol{\mu}'|\boldsymbol{\mu})$ is close to uniform over the entire parameter domain, the acceptance ratio will likely be very small, and N_m will again need to be unnecessarily high for useful statistics.

Two different proposal distributions are used in this work. The first is a uniform distribution over a hypercube of dimension $\boldsymbol{\Delta}\in\mathbb{R}^p$ centered at $\boldsymbol{\mu}$ in parameter space, so that $\boldsymbol{\mu}'$ is generated by

$$\boldsymbol{\mu}'=\boldsymbol{\mu}+\mathrm{diag}(\boldsymbol{\Delta})\mathbf{r} \tag{9}$$

where the components of $\mathbf{r}\in\mathbb{R}^p$ are drawn from a uniform distribution on $[\frac{-1}{2},\frac{1}{2}]$. Note that for this proposal, $q(\boldsymbol{\mu}'|\boldsymbol{\mu})=q(\boldsymbol{\mu}|\boldsymbol{\mu}')$, so that these terms cancel in (8).

The locally uniform proposal distribution described above will be inefficient at sampling a posterior distribution that exhibits a high degree of anisotropy. For example, for the posterior distribution shown in Figure 1, only proposals aligned with the stretched direction have a reasonable chance of being accepted; the majority of proposals will have $\alpha\ll 1$. To improve the acceptance rate, a second proposal distribution is considered, in the form of an anisotropic Gaussian centered at $\boldsymbol{\mu}$:

$$q(\boldsymbol{\mu}'|\boldsymbol{\mu})\propto\exp\left[-\frac{1}{2\delta^2}(\boldsymbol{\mu}'-\boldsymbol{\mu})^{\mathrm{T}}\mathbf{H}(\boldsymbol{\mu})(\boldsymbol{\mu}'-\boldsymbol{\mu})\right] \tag{10}$$

where δ is a dimensionless step-size parameter that can be varied to tune the acceptance rate, similar in role to the size of $\boldsymbol{\Delta}$ for the uniform proposal. The matrix $\mathbf{H}(\boldsymbol{\mu})\in\mathbb{R}^{p\times p}$ represents the anisotropy of the posterior distribution: its eigenvectors point in the principal stretching directions, and its eigenvalues control the magnitude of the stretching. It is given by

$$\mathbf{H}(\boldsymbol{\mu})=\frac{1}{2\sigma^2}\left[\frac{\partial\mathbf{y}}{\partial\boldsymbol{\mu}}(\boldsymbol{\mu})\right]^{\mathrm{T}}\frac{\partial\mathbf{y}}{\partial\boldsymbol{\mu}}(\boldsymbol{\mu}) \tag{11}$$

where the Jacobian $\partial\mathbf{y}/\partial\boldsymbol{\mu}$ can be calculated using finite differencing or analytical sensitivities (e.g. computed using an adjoint approach). The choice of $\mathbf{H}(\boldsymbol{\mu})$ in (11) corresponds to a Gauss–Newton

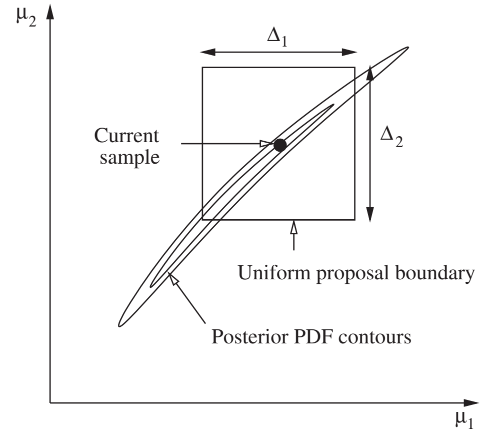

Figure 1. Inefficiency of uniform proposal for a sample anisotropic posterior distribution in a two-dimensional parameter space. Black dot denotes the current sample value in the Markov chain, $\boldsymbol{\mu}=\boldsymbol{\mu}^i$. Outside the curved contours, the posterior PDF is assumed negligible, so that the acceptance probability of a random proposal within the square boundary is very small.

approximation of the Hessian of the outputs with respect to the inputs, and it makes $q(\boldsymbol{\mu}'|\boldsymbol{\mu})$ in (10) a local approximation to the posterior distribution in (7). Specifically, the iso-probability curves of $q(\boldsymbol{\mu}'|\boldsymbol{\mu})$ in parameter space will be stretched ellipses, with the stretching controlled by the Hessian approximation. Details on the derivation can be found in [22]. For this proposal distribution, $\boldsymbol{\mu}'$ is generated by

$$\boldsymbol{\mu}'=\boldsymbol{\mu}+\delta\mathbf{L}^{-1}\mathbf{w}$$

where $\mathbf{H}(\boldsymbol{\mu})=\mathbf{L}^{\mathrm{T}}\mathbf{L}$ is the Cholesky decomposition, and each component of $\mathbf{w}\in\mathbb{R}^p$ is drawn from $N(0,1)$. Note that now $q(\boldsymbol{\mu}'|\boldsymbol{\mu})\neq q(\boldsymbol{\mu}|\boldsymbol{\mu}')$, so that these terms do not cancel in (8). However, the calculated $\mathbf{H}(\boldsymbol{\mu}')$ matrix can be reused following an accepted proposal, so that the number of forward solves per MCMC step using the stretched Gaussian proposal is $1+\dim(\mathscr{D})$ if $\partial\mathbf{y}/\partial\boldsymbol{\mu}$ is approximated by finite differences. Although the per-step expense of using this proposal is greater than that for the locally uniform distribution, a net computational decrease is expected for highly anisotropic posteriors due to significantly fewer rejected proposals.

3.4. Sampling statistics

The samples $\{\boldsymbol{\mu}^i\}_{i=1}^{N_m}$ are drawn from the posterior probability distribution, so that for large N_m, the distribution of the samples is expected to converge to the actual probability distribution of $\boldsymbol{\mu}^*$. However, for any finite N_m, statistical quantities calculated from $\{\boldsymbol{\mu}^i\}_{i=1}^{N_m}$ will only be estimators of the true statistics. Confidence in these estimators is measured by treating them as random variables and estimating their variances, as described below.

Treating the components of $\boldsymbol{\mu}$ as random variables, an estimator for a statistical quantity is defined by

$$\bar{f} \equiv \frac{1}{N_m} \sum_{i=1}^{N_m} g(\boldsymbol{\mu}^i) \tag{12}$$

where $g(\boldsymbol{\mu})$ is the kernel of the estimator. For example, for an estimator of the mean of parameter μ_j, where $j=1,2,\ldots,p$, $g(\boldsymbol{\mu})=\mu_j$. For the variance of μ_j, $g(\boldsymbol{\mu})=(\mu_j-\bar{\mu}_j)^2$, where $\bar{\mu}_j=\sum_{i=1}^{N_m}\mu_j^i/N_m$ is the sample mean. The estimator defined in (12) is itself a random variable in the sense that different MCMC chains will produce different $\bar{f}$. The variance of $\bar{f}$ with respect to different MCMC chain realizations is a measure of accuracy of the estimator for a given finite N_m. For $N_m\to\infty$, this variance can be calculated as [20]

$$\operatorname{var}(\bar{f}) = \frac{\tau_g \operatorname{var}(g(\boldsymbol{\mu}^i))}{N_m} \tag{13}$$

where $\tau_g>0$ is the integrated autocorrelation time,

$$\tau_g = 1+2\sum_{s=1}^{\infty} \rho_g(s) \tag{14}$$

and $\rho_g(s)$ is the normalized autocovariance function for a lag of s,

$$\rho_g(s) = \frac{1}{\operatorname{var}(g(\boldsymbol{\mu}^i))} \operatorname{cov}(g(\boldsymbol{\mu}^i), g(\boldsymbol{\mu}^{i+s})) \tag{15}$$

For finite N_m, the estimates for $\rho_g(s)$ become noisy with increasing s, and hence the sum in (14) is truncated when the noise begins to dominate. In this work, this truncation is performed heuristically by monitoring the decay of $\rho_g(s)$ versus s. Such a procedure is deemed sufficient as the purpose of the estimator variance calculation is to gain insight into the credibility of the estimator for finite N_m. Note that the calculation of a statistic estimator itself requires only (12).

The MCMC approach to solving the statistical inverse problem may require tens of thousands of forward problem solutions—making it computationally intractable for the large-scale systems that result from PDE discretizations. In the next section, we present a model reduction approach that provides dramatic reductions in the cost of a forward solve. The reduced model will be used in place of the full-order system within the MCMC sampling.

4. MODEL REDUCTION

This section presents a reduced-basis approximation of the general non-linear system in (1). The key to the success of the model reduction is an online operation count that is independent of the number of full-order degrees of freedom and the number of evaluations of the non-linear term $\mathbf{s}(\cdot,\boldsymbol{\mu})$ in the full-order system.

4.1. *Reduced-basis approximation*

In the reduced-basis approximation, the state is constrained to lie in an n-dimensional space spanned by basis functions $\boldsymbol{\phi}_j \in \mathbb{R}^N$, $1 \leqslant j \leqslant n$ where $n \ll N$. The expansion can be written in compact form as

$$\mathbf{u} \approx \bar{\mathbf{u}} + \boldsymbol{\Phi}\mathbf{u}_r \tag{16}$$

where $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_n] \in \mathbb{R}^{N \times n}$, and $\mathbf{u}_r \in \mathbb{R}^n$ is the vector of reduced states. The particular solution $\bar{\mathbf{u}} \in \mathbb{R}^N$ is included for discretizations that enforce the Dirichlet boundary condition strongly, as the basis functions are chosen to satisfy homogeneous boundary conditions.

The discrete reduced model is obtained by substituting (16) into (1) and, in the standard Galerkin projection fashion, requiring the resulting residual to be orthogonal to the space spanned by the basis vectors $\boldsymbol{\phi}_j$, yielding

$$\mathbf{R}_r + \mathbf{A}_r \mathbf{u}_r + \boldsymbol{\Phi}^{\mathrm{T}} \mathbf{s}(\bar{\mathbf{u}} + \boldsymbol{\Phi}\mathbf{u}_r; \boldsymbol{\mu}) = \mathbf{0} \tag{17}$$

where $\mathbf{R}_r = \boldsymbol{\Phi}^{\mathrm{T}}(\mathbf{R}_0 + \mathbf{A}\bar{\mathbf{u}}) \in \mathbb{R}^n$, and $\mathbf{A}_r = \boldsymbol{\Phi}^{\mathrm{T}}\mathbf{A}\boldsymbol{\Phi} \in \mathbb{R}^{n \times n}$. Efficient evaluation of the non-linear term will be addressed in the next subsection. The reduced model approximation of the outputs defined by (2) is given by

$$\mathbf{y}_r(\mathbf{u}_r(\boldsymbol{\mu})) = \mathbf{C}_r \mathbf{u}_r(\boldsymbol{\mu}) \tag{18}$$

where $\mathbf{y}_r \in \mathbb{R}^q$ is the reduced model approximation of the q outputs of interest and $\mathbf{C}_r = \mathbf{C}\boldsymbol{\Phi}$.

The n basis vectors $\boldsymbol{\phi}_j$ can be obtained by many methods. In this work, we use the POD [9–11]. Using the method of snapshots [11], the basis vectors are computed from a set of $K > n$ solution snapshots, $\{\mathbf{u}^k\}_{k=1}^K$, where $\mathbf{u}^k \equiv \mathbf{u}(\boldsymbol{\mu}^k)$ is the state solution computed for the kth parameter sample, $\boldsymbol{\mu}^k$. The parameter sample points are chosen by grid-sampling when the parameter space dimension is low ($\leqslant 3$), or an optimization-based adaptive sampling technique [23, 24] otherwise. The POD procedure consists of finding a set of orthonormal basis vectors that most accurately represent the snapshots in a mean-square error sense. Specifically, the basis vectors are chosen such that the sum of the square-magnitudes of the snapshot projections onto the basis set is maximum. In this work, the continuous L_2 inner product over the domain Ω is used for the projections.

4.2. *Interpolation of non-linear term by masked projection*

While (17) represents n discrete equations in n unknowns, the cost of its solution still depends on the size of the original problem through the evaluation of $\mathbf{s}(\cdot; \boldsymbol{\mu})$ and its projection onto the reduced basis. For large problems, these computations will dominate the online cost of solving the reduced model. In this work, an online cost independent of the size of the original problem is achieved by combining the reduced basis approximation with the EIM of [17, 18]. Here, we present an interpretation of that method as a masked projection, that is, evaluations of the non-linear term over a subset of points, as described below.

The starting point for this construction is a set of K snapshots of the non-linear term, $\{\mathbf{s}^k\}_{k=1}^K$, where $\mathbf{s}^k = \mathbf{s}(\mathbf{u}^k; \boldsymbol{\mu}^k)$ is the non-linear term corresponding to the kth state snapshot. From these snapshots of the non-linear term, a set of $m < K$ POD basis vectors is computed, where m is chosen so that the basis vectors accurately span the space of the snapshots. The resulting expansion of a non-linear vector $\mathbf{s} \in \mathbb{R}^N$ takes the form,

$$\mathbf{s} \approx \boldsymbol{\Phi}^s \mathbf{s}_r \tag{19}$$

DOI: 10.1002/nme

where $\mathbf{\Phi}^s = [\boldsymbol{\phi}_1^s, \ldots, \boldsymbol{\phi}_m^s] \in \mathbb{R}^{N\times m}$ contains the orthonormal basis functions resulting from the POD applied to the snapshots of the non-linear vector, and $\mathbf{s}_r \in \mathbb{R}^m$ is the vector of expansion coefficients. As in (16), the approximation sign indicates that a general non-linear vector $\mathbf{s}$ will not lie exactly in the span of $\{\boldsymbol{\phi}_i^s\}_{i=1}^m$.

The approximation in (19) does not in itself yield an efficient reduced model because calculation of the expansion coefficients via projection, $\mathbf{s}_r = (\mathbf{\Phi}^s)^{\mathrm{T}}\mathbf{s}$, depends on N. An N-independent reduced model is obtained by applying masked projection [16, 19, 25, 26] to calculate $\mathbf{s}_r$ approximately; that is, an approximation of $\mathbf{s}_r$ is computed using a small subset, $m \ll N$, of the components of $\mathbf{s}$. Note that the number of components of $\mathbf{s}$ used to calculate $\mathbf{s}_r$ is the same as the number of basis vectors for the non-linear term. Using fewer then m components of $\mathbf{s}$ would result in an under-determined system for $\mathbf{s}_r$. On the other hand, using more than m components would result in an over-determined system, which could yield a more robust solution. This possibility has not been explored in the present work. We denote the indices of the sampled components of $\mathbf{s}$ used for creating $\mathbf{s}_r$ by $z_i \in 1, \ldots, N$, where $i \in 1, \ldots, m$. That is, for each i in $1, \ldots, m$, the *sampled index* z_i is an integer that gives the row number at which the column vector $\mathbf{s} \in \mathbb{R}^N$ is sampled. The full set of these sampled indices is then defined by the mask matrix $\mathbf{Z} \in \mathbb{R}^{m\times N}$ whose ijth entry is

$$\begin{aligned} Z_{ij} &= 1 \quad \text{if } j = z_i \\ Z_{ij} &= 0 \quad \text{otherwise} \end{aligned}$$

We now compute $\widetilde{\mathbf{s}}_r$, the approximation to $\mathbf{s}_r$, using just the m sampled indices in the mask. Multiplying (19) by $\mathbf{Z}$ yields

$$\mathbf{Z}\mathbf{s} \approx \mathbf{Z}\mathbf{\Phi}^s\mathbf{s}_r \tag{20}$$

where $\mathbf{Z}\mathbf{\Phi}^s$ is a square matrix. Assuming that this matrix is invertible, we solve for $\mathbf{s}_r$,

$$\widetilde{\mathbf{s}}_r \approx (\mathbf{Z}\mathbf{\Phi}^s)^{-1}\mathbf{Z}\mathbf{s} \tag{21}$$

Substituting back into (19) yields the approximate representation

$$\widetilde{\mathbf{s}} \approx \mathbf{\Psi}^s\mathbf{Z}\mathbf{s} \tag{22}$$

where $\mathbf{\Psi}^s = \mathbf{\Phi}^s(\mathbf{Z}\mathbf{\Phi}^s)^{-1} \in \mathbb{R}^{N\times m}$, and $\widetilde{\mathbf{s}}$ denotes the approximation to $\mathbf{s}$ due to masked projection. Note that the columns of $\mathbf{\Psi}^s$ form a cardinal basis set with respect to the sampled indices in the sense that $\mathbf{Z}\mathbf{\Psi}^s = \mathbf{I}^{m\times m}$, where $\mathbf{I}$ is the identity matrix. This property explains the use of 'interpolation' to describe the approximation of the non-linear term.

Finally, we substitute $\widetilde{\mathbf{s}}$ from (22) for $\mathbf{s}$ in (17), to yield the reduced model

$$\mathbf{R}_r + \mathbf{A}_r\mathbf{u}_r + \mathbf{\Phi}^{\mathrm{T}}\mathbf{\Psi}^s\mathbf{Z}\mathbf{s} = 0 \tag{23}$$

Evaluation of $\mathbf{Z}\mathbf{s}$ is efficient when the calculation of each component of $\mathbf{s}$ is local (that is, based on only a few components of $\mathbf{u}$ at a time). Then only $O(m)$ components of $\mathbf{u}$ are used to compute $\mathbf{Z}\mathbf{s}$. This is the case in many problems where interpolation is used to compute the non-linear contributions. For example, quadrature is often employed to approximate integrals that arise from a finite element discretization of non-linear governing equations. In that case, the full discretized model might have a non-linear term of the form

$$\mathbf{s}(\mathbf{u}; \boldsymbol{\mu}) = \mathbf{E}\,\mathbf{f}(\mathbf{D}\mathbf{u}; \boldsymbol{\mu}) \tag{24}$$

where any integrals contributing to this non-linear component are assumed to be evaluated using numerical quadrature with a total of M quadrature points over the numerical domain. Thus, $\mathbf{D}\in\mathbb{R}^{M\times N}$ in (24) is an interpolation matrix mapping the discrete unknown states (e.g. nodal values of u) to values of u at the M quadrature points. The vector $\mathbf{f}(\cdot,\boldsymbol{\mu})$ is the non-linear term: it is evaluated at all components of its first input argument. Thus, $\mathbf{f}(\cdot,\boldsymbol{\mu})\in\mathbb{R}^M$ contains the evaluation of the non-linear term at the quadrature points, and $\mathbf{E}\in\mathbb{R}^{N\times M}$ prescribes how the M evaluations of the non-linear term enter into each algebraic equation. Both $\mathbf{D}$ and $\mathbf{E}$ are typically sparse, and they are not constructed in practice. However, the cost of solving the full model in this case depends on both the number of states, N, and the number of evaluations of the non-linear term, M. The reduced model in this case is constructed using snapshots of $\mathbf{f}(\cdot,\boldsymbol{\mu})$ to compute the basis $\boldsymbol{\Psi}^s$, and reads

$$\mathbf{R}_r+\mathbf{A}_r\mathbf{u}_r+\mathbf{E}_r\mathbf{f}_r(\bar{\mathbf{u}}_r+\mathbf{D}_r\mathbf{u}_r;\boldsymbol{\mu})=0 \tag{25}$$

where

$$\begin{aligned}\mathbf{E}_r&=\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{E}\boldsymbol{\Psi}\in\mathbb{R}^{n\times m}\\ \mathbf{D}_r&=\mathbf{Z}\mathbf{D}\boldsymbol{\Phi}\in\mathbb{R}^{m\times n}\\ \bar{\mathbf{u}}_r&=\mathbf{Z}\mathbf{D}\bar{\mathbf{u}}\in\mathbb{R}^{m}\end{aligned}$$

and $\mathbf{f}_r\in\mathbb{R}^m$ contains the non-linear terms evaluated only at the mask sampled indices, that is, $\mathbf{f}_r=\mathbf{Z}\mathbf{f}$.

All of the matrices and calculations in (25) are independent of N, and hence the reduced model is inexpensive to solve. The matrices in (25) can be computed in an offline stage, so that each online calculation requires solving a system of just n non-linear equations, with just m evaluations of $\mathbf{f}(\cdot;\boldsymbol{\mu})$ per residual evaluation.

4.3. Mask construction

The idea of using masks has been explored previously in the contexts of reconstructing missing data and model reduction. Everson and Sirovich applied a least-squares variation of (21), in which $\mathbf{Z}$ contained more than m rows, to reconstruct facial images from ones with missing pixels in a technique termed gappy POD [19]. This technique was subsequently applied to steady and unsteady aerodynamic flow reconstruction problems using data from a limited number of sensors [25, 26]. Astrid *et al.* applied a similar idea to reduce the cost of an inner-product between two vectors by sampling the vectors at masked indices, using a least-squares variation of (21) to obtain the approximate orthonormal basis coefficients, and evaluating the dot product between the resulting coefficient vectors [16]. In a continuous setting, a mask consists of a set of interpolation points, and (22) becomes the coefficient-function approximation of [17, 18]. As mentioned in [16], reconstruction of information using gappy spatial data is a reasonable strategy since, for those applications for which model reduction is effective, the basis vectors are often relatively smooth in space and thus amenable to interpolation.

The accuracy of the reduced model depends on the choice of the mask $\mathbf{Z}$. Indeed certain choices can make $\mathbf{Z}\boldsymbol{\Phi}^s$ singular or severely ill-conditioned. Astrid *et al.* [16] and Willcox [26] use this observation directly to drive a greedy algorithm that chooses sampling points to minimize the condition number of $\mathbf{Z}\boldsymbol{\Phi}^s$. In a continuous setting, Nguyen *et al.* [27] propose two alternate mask-selection techniques: the best points interpolation method (BPIM), and the EIM. The BPIM

DOI: 10.1002/nme

consists of solving an optimization problem to minimize the error between the masked projection and the full projection of the K snapshots of the non-linear term. Applied to the problem at hand, this means choosing $\mathbf{Z}$ as

$$\mathbf{Z} = \arg\min_{\mathbf{Z}} \sum_{k=1}^{K} (\mathbf{\Phi}^{s\mathrm{T}} \boldsymbol{s}^k - (\mathbf{Z}\mathbf{\Phi}^s)^{-1} \mathbf{Z}\boldsymbol{s}^k)^2 \tag{26}$$

which can be an expensive least-squares minimization problem. Alternatively, the EIM is a greedy algorithm that is sub-optimal but less expensive. In the EIM, the mask is built in a recursive manner, with the $i+1$st sampling point chosen at the location where the error between $\boldsymbol{\phi}^s_{i+1}$ and its reconstruction using the first i sampling points is maximum. For example, provided with an additional basis function, $\boldsymbol{\phi}^s_{m+1}$, the $m+1$st sampling point would be chosen as the index where the norm of the following error vector is maximum:

$$|\boldsymbol{\phi}^s_{m+1} - \mathbf{\Phi}^s (\mathbf{Z}\mathbf{\Phi}^s)^{-1} \mathbf{Z}\boldsymbol{\phi}^s_{m+1}|$$

and $\mathbf{Z}$ would be augmented with an additional row accordingly. In this work, both the BPIM and the EIM are used.

5. RESULTS

In this section, we present the results for an advection–diffusion-reaction system that represents a jet diffusion flame in a combustor. The governing equations are given by

$$\begin{aligned} \nabla\cdot(\boldsymbol{U}u) - \nabla\cdot(v\nabla u) + f(u;\boldsymbol{\mu}) &= 0 \quad \text{in } \Omega \\ u &= u_D \quad \text{on } \partial\Omega_D \\ \nabla u\cdot\mathbf{n} &= 0 \quad \text{on } \partial\Omega\backslash\partial\Omega_D \end{aligned} \tag{27}$$

where the field variable u represents the concentration of fuel in a domain Ω. The velocity field, $\boldsymbol{U}$, and the diffusion coefficient, v, are taken as constant. The field is specified with $u = u_D$ on the Dirichlet inflow boundary $\partial\Omega_D$, and zero diffusive flux is prescribed on all other boundaries. The non-linear reaction term $f(u;\boldsymbol{\mu})$ is of Arrhenius type and is given by

$$f(u;\boldsymbol{\mu}) = Au(c-u)e^{-E/(d-u)} \tag{28}$$

where c and d are known constants and the system parameters $\boldsymbol{\mu} = (\ln(A), E)$ can vary within the parameter domain $\mathscr{D} \equiv [5.00,\ 7.25] \times [0.05,\ 0.15] \subset \mathbb{R}^2$. Physically, u represents fuel concentration, while $c-u$ is the oxidizer concentration. In practice, the Arrhenius parameters A and E are unknown and are inferred from experimental data, which often consist of average concentration measurements along lines or planes in the domain. Hence, this example is well-suited for the inverse problem context.

Figure 2 shows the combustor domain. A constant flow velocity of 0.2 m/s is imposed in the positive x direction. The diffusivity is set to $v = 5\times10^{-6}\,\mathrm{m}^2/\mathrm{s}$, and the fixed Arrhenius parameters are $c = 0.2$ and $d = 0.24$. At the Dirichlet boundary on the left of the domain, u_D is set to c at the fuel inflow, and to 0 at the oxidizer inflow. The output vector consists of average fuel concentrations at the measurement planes.

DOI: 10.1002/nme

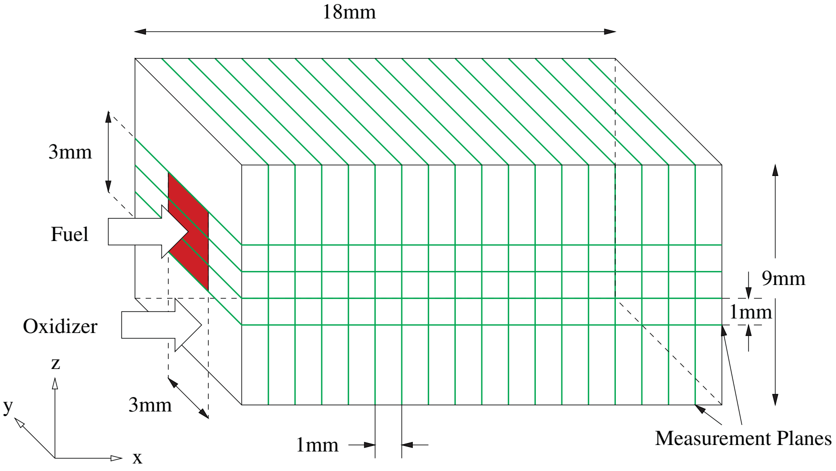

Figure 2. Combustor model domain and problem setup.

In the following, we first present a streamline upwind Petrov–Galerkin (SUPG) finite element discretization of (27) on a simplified domain in two dimensions. The resulting system of $N = 3383$ equations allows for detailed comparisons between full-order and reduced-order results. Second, we present a discontinuous Galerkin (DG) discretization of (27) on the full three-dimensional domain, which leads to a very large system with more than 8.5 million degrees of freedom. For both formulations, the discrete full-order and reduced-order equations are solved using a damped Newton method.

5.1. Model problem in two dimensions

The two-dimensional domain is formed from a vertical slice at constant y in the center of the combustor shown in Figure 2. Thus, the output vector consists of 17 average concentrations measured on vertical lines at the equally spaced x locations shown in the figure. All reduced-order approximations were constructed using a set of $K = 196$ snapshots generated with parameter vectors $\boldsymbol{\mu}^k$ on a uniform 14×14 grid in the parameter space $\mathscr{D}$ (one snapshot per parameter point). Although the choice of snapshots is important for adequately capturing the relevant physics, there are no rigorous error guarantees for the problem at hand. Instead, in this work we rely on grid-based sampling, and we check the errors *a posteriori*.

5.1.1. SUPG discretization. An SUPG finite element discretization of (27) in two dimensions is defined on standard linear triangles. The total number of degrees of freedom, N, is the number of nodes in the mesh, not including the nodes on the Dirichlet boundary. For the finite element mesh used in this work, $N = 3383$. The spatial resolution of the mesh is uniform along the x direction, as the location of the reaction front varies significantly for the parameter range considered. In the z direction, the mesh has less resolution away from the centerline since the reaction front extends minimally into this region [24]. The solutions at various parameter samples were examined visually to deduce that the chosen resolution adequately captured the physics of the problem, and the set

DOI: 10.1002/nme

of equations arising from the finite element discretization was treated as the 'true' system for the projection-based model reduction.

Denote by V_D the approximation space of piecewise linear functions satisfying the Dirichlet boundary conditions, and by V the piecewise linear space of test functions satisfying the homogenous form of the Dirichlet boundary conditions. For $w \in V_D$ and $v \in V$, the semilinear form obtained from (27) reads

$$\begin{aligned}\mathscr{R}(w, v; \boldsymbol{\mu}) = {} & \sum_{k=1}^{N_e} \int_{\Omega_k} (v\boldsymbol{U}\cdot\nabla w + \nu\nabla v\cdot\nabla w + v\, f(w; \boldsymbol{\mu}))\,\mathrm{d}\mathbf{x} \\ & + \sum_{k=1}^{N_e} \int_{\Omega_k} \tau_k \boldsymbol{U}\cdot\nabla v(\boldsymbol{U}\cdot\nabla w + f(w; \boldsymbol{\mu}))\,\mathrm{d}\mathbf{x}\end{aligned} \tag{29}$$

where N_e is the number of elements, $\boldsymbol{\Omega}_k$ denotes the kth element, and τ_k is a SUPG stabilization parameter required to prevent spurious solution oscillations for advection-dominated flows. In this work, the following choice of τ_k, due to Brooks and Hughes [28], is used

$$\tau_k = \frac{h_k}{2\|y\boldsymbol{U}\|}\left(\coth(Pe_k) - \frac{1}{Pe_k}\right), \quad Pe_k \equiv \|\boldsymbol{U}\| h_k/(2\nu)$$

where h_k is a measure of the size of element k.

In the general discrete form (1), this SUPG discretization leads to the following terms that comprise the non-linear residual vector: $\mathbf{R}_0$ is zero and $\mathbf{A}$ is formed from all the terms containing ∇w in (29). The non-linear contributions are approximated using quadrature, thus giving a term of the form (24), where $\mathbf{D}$ consists of the finite element basis functions evaluated at the element quadrature points, and $\mathbf{E}$ is $\mathbf{D}^{\mathrm{T}}$ multiplied by the quadrature weights. Four quadrature points are used for each element, giving $M = 26\,784$ for the mesh used here. One particular solution satisfying the Dirichlet boundary conditions is chosen for $\bar{\mathbf{u}}$ in (16).

5.1.2. Performance of the reduced model. To test the accuracy of the model reduction technique for the forward problem, reduced models of various sizes, n and m, were constructed using the set of $K = 196$ snapshots. In addition, the EIM was compared with the BPIM for creating the projection mask for the non-linear term. The accuracy of each reduced model was assessed by computing an average relative error norm of outputs over a set of test parameters:

$$\varepsilon_{\mathrm{rel}} = \underset{\boldsymbol{\mu}\in\Xi^{\mathrm{test}}}{\mathrm{mean}} \frac{\|\boldsymbol{y}(\boldsymbol{\mu}) - \boldsymbol{y}_r(\boldsymbol{\mu})\|}{\|\boldsymbol{y}(\boldsymbol{\mu})\|} \tag{30}$$

where the test set Ξ^{test} consists of a 23×23 grid in parameter space and contains different parameter samples from those used to create the POD basis.

Figure 3 shows $\varepsilon_{\mathrm{rel}}$ as a function of n and m for reduced models using both the EIM and the BPIM. For a given value of m, the error decreases as n is increased until a point is reached where the convergence stalls: the error remains constant regardless of the number of basis functions added to the approximation space. Convergence stalls when the error in the solution is dominated by the error due to the approximation of the non-linear term. Using BPIM to create the reduced model approximation only has an effect on the point where convergence stalls. That is, for a given value of m, the minimum achievable error (for sufficiently large n) when using BPIM is less than

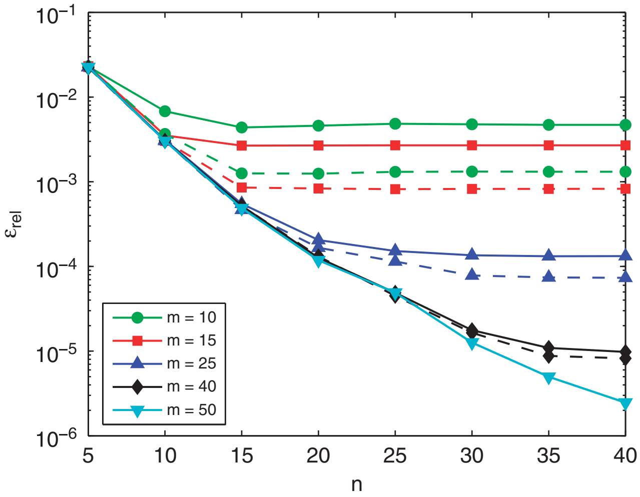

Figure 3. The average relative error norm of the output, ε_{rel}, for the two-dimensional problem as a function of n and m when the non-linear term projection mask is built using EIM (solid lines) versus BPIM (dashed lines).

when using EIM. Note that before the error curves level off both methods provide almost identical results.

To provide additional insight into the point at which the curves in Figure 3 start to level off, the singular values of the state snapshot matrix, $\{\mathbf{u}^k\}_{k=1}^K$, and of the non-linear term snapshot matrix, $\{\mathbf{s}^k\}_{k=1}^K$, are plotted in Figure 4. The singular values are normalized by the largest singular value in order to compare the decay rates. The slower decay rate in the singular values of $\{\mathbf{s}^k\}_{k=1}^K$ suggests that the space spanned by the non-linear term snapshots is richer than the space spanned by the state snapshots, and hence it is advantageous to use $m>n$.

Figure 5 shows sample concentration profiles for both full and reduced solutions, and Table I presents a summary of the numerical results obtained with the reduced models. Shown are the maximum and average relative errors in the output over the test space Ξ^{test}, and a comparison of the online computational time of the reduced model versus the full model. Note that outputs can be computed more than 50 000 times faster using the reduced model than using the full model for all values of n and m shown in this table. Furthermore, the convergence of the reduced-model approximation is excellent as n is increased.

The offline cost for building a reduced model in this example consists of 196 full model snapshot solutions, plus the (relatively small) computational cost of forming the POD bases that span the state and non-linear term spaces. In multiple-query contexts such as the MCMC sampling considered in the next section, this up-front cost is quickly amortized. Furthermore, in real-time applications, building a reduced model may be the only viable option for computing results in the required amount of time.

5.1.3. Inverse problem solution. The reduced model generated with $n=40$ and $m=50$ was used to solve the statistical inverse problem via the Bayesian inference approach. A set of measurements was generated according to (5) using $\boldsymbol{\mu}_e=(6.7386, 0.0727)$ and normally distributed errors with

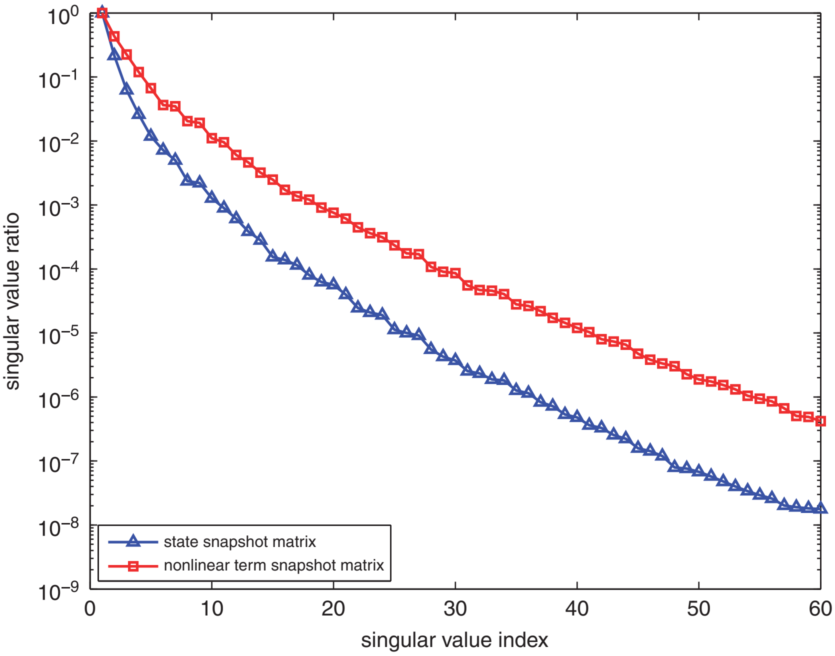

Figure 4. The singular values of the state snapshot matrix and of the non-linear term snapshot matrix, normalized by the largest singular value in each case.

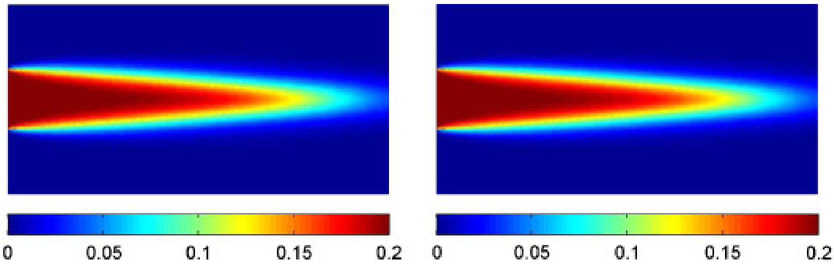

Figure 5. Comparison of full model (left) and reduced model (right) solutions for two-dimensional problem with $\boldsymbol{\mu}=(\ln(A), E)=(6.4318, 0.1091)$. The reduced model was constructed using $n=40$, $m=50$, and the EIM for creating the projection mask.

standard deviation $\sigma=3.54\times10^{-3}$ (1.5% of the output mean). The goal of the inverse problem was to determine the posterior probability distributions of the Arrhenius parameters $\mu_1^*=\log(A^*)$ and $\mu_2^*=E^*$. To this end, a Markov chain of size $N_m=50\,000$ was generated according to the MCMC algorithm presented in Section 3. The uniform proposal distribution was used for sample generation, and $\boldsymbol{\Delta}$ in (9) was set to one quarter of the dimensions of $\mathscr{D}$ based on empirical trials.

To assess convergence of the Markov chain, the integrated autocorrelation time, τ_g, was calculated for the mean and variance of both input parameters, according to (14). The values of τ_g were all between 222 and 304, significantly less than N_m. Figure 6 shows the normalized autocovariance

DOI: 10.1002/nme

Table I. Maximum and average relative errors in the output of the reduced model for the two-dimensional problem over the parameter test grid Ξ^{test} and online computational time required by the reduced-model solver as a function of n for $m=50$ and using the BPIM to create the projection mask. Computational times are normalized with respect to the time required by the full model to compute the solution at $\boldsymbol{\mu}=(7.25, 0.05)$.

n	m	Avg. rel. error	Max. rel. error	Online time
5	50	2.25E−02	9.73E−02	1.59E−05
10	50	3.03E−03	2.78E−02	1.61E−05
20	50	1.18E−04	2.00E−03	1.63E−05
30	50	1.26E−05	4.48E−04	1.71E−05
40	50	2.47E−06	1.34E−04	2.00E−05

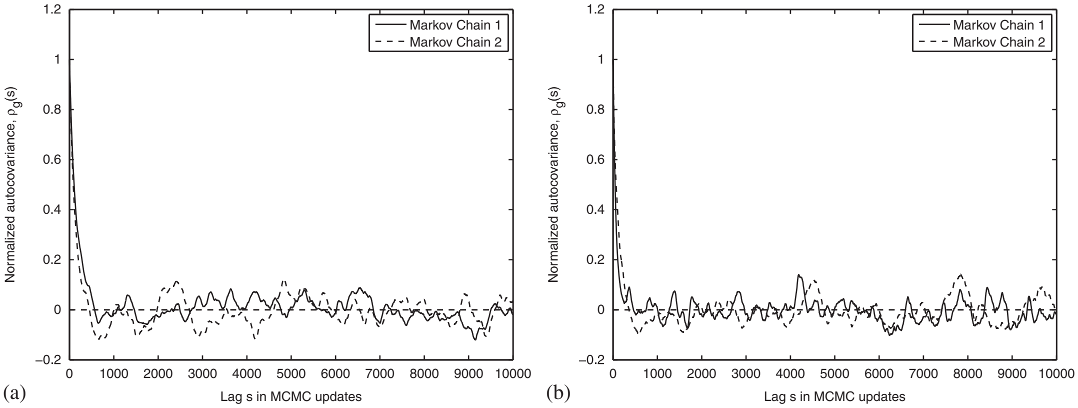

Figure 6. Normalized autocovariance functions for mean and variance estimators for the two-dimensional problem: (a) Mean and (b) Variance.

functions for mean and variance estimators, as defined by (15) for two different Markov chain realizations. The normalized autocovariance functions can be seen to approach zero smoothly and then to oscillate randomly due to noise. In addition, no obvious trend is apparent in the sample sequences, which are shown in Figure 7. These checks imply that the computed Markov chain is adequate for obtaining reliable posterior probability distributions for the unknown input parameters of the inverse problem.

The marginal posterior histograms for both the parameters are shown in Figure 8. The mean values associated with the calculated distribution of $\boldsymbol{\mu}^*$ deviate from the original values in $\boldsymbol{\mu}_e$ because the inversion was performed for one realization of the measurement errors. In particular, the mean estimates for the parameters computed from the Markov chain samples are 6.70 ± 0.013 for $\ln(A)$ and 0.068 ± 0.0015 for E. The standard deviation estimates are 0.086 ± 0.007 for $\ln(A)$ and 0.0095 ± 0.0007 for E. The accompanying intervals are the 95% credible intervals obtained from the estimator variance calculation outlined in Section 3.4.

An important question is the effect on the inference results of using the reduced model in place of the full CFD model for the sample evaluations. From Table I, it can be seen that the $n=40$, $m=50$ reduced model exhibits an average relative error of 2.47×10^{-6} and a maximum error of

DOI: 10.1002/nme

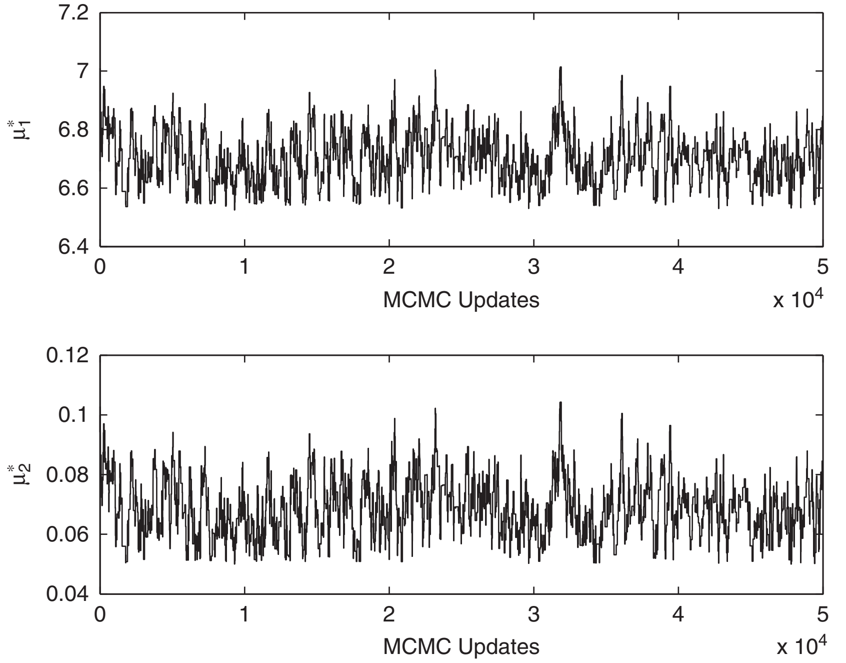

Figure 7. Markov chain samples from posterior probability distribution for the Arrhenius parameters obtained using the reduced model with $n=40$ and $m=50$ for the two-dimensional problem.

1.34×10^{-4} compared with the full finite element model solution over the parameter test set. We can determine the effect of reduced model error on the computed statistics by using the full model to solve the statistical inverse problem for the same output measurement vector and realization of experimental error. Using the full model, a Markov chain of 10 000 samples was generated again using the uniform proposal distribution for sample generation. The number of samples was limited by computational time, as the full model is several orders of magnitude more expensive compared with the reduced model. Convergence of this Markov chain was verified using the criteria discussed in Section 3.4. Figure 9 illustrates the estimated means of the two parameters with 95% credible intervals for the full model MCMC. For reference, results from a 10 000-sample reduced model MCMC are also shown.

It can be seen from Figure 9 that for the 10 000-sample case both the parameter mean estimates and 95% credible intervals are in good agreement between the reduced and full-order models. Further, as expected, the credible intervals are wider for the five-fold shorter Markov chain. This empirical validation shows that a reduced model with high levels of accuracy over the parametric input space can provide an accurate solution to the statistical inverse problem; however, in the general case no rigorous guarantees can be made regarding the effects of model reduction error. An interesting future research direction is the inclusion of estimates of reduced model error as an additional source of uncertainty in the Bayesian formulation (although again, in the general non-linear case, any model of reduction error will be heuristic and provide no rigorous guarantees). However, despite the lack of *a priori* guarantees, it is important to note that the reduced model delivers the statistical inverse problem results approximately 4000 times faster than the full model.

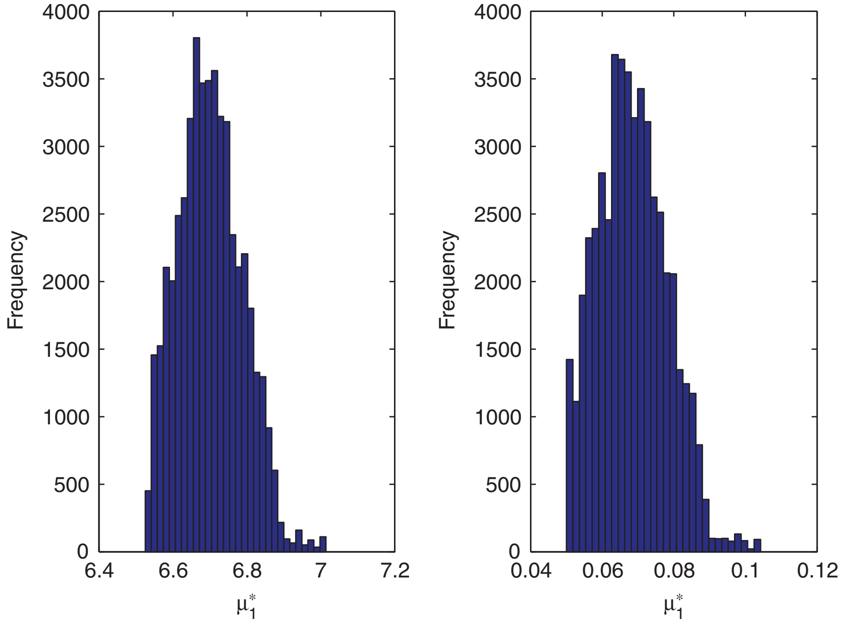

Figure 8. Marginal posterior histograms for $\mu_1^* = \log(A^*)$ and $\mu_2^* = E^*$ for the two-dimensional problem with an $N_M = 50\,000$ Markov chain generated using the reduced model with $n = 40$ and $m = 50$. Note that $\boldsymbol{\mu}^* = (\mu_1^*, \mu_2^*)$ can vary within the system parameters domain $\mathscr{D} \equiv [5.00, 7.25] \times [0.05, 0.15] \subset \mathbb{R}^2$.

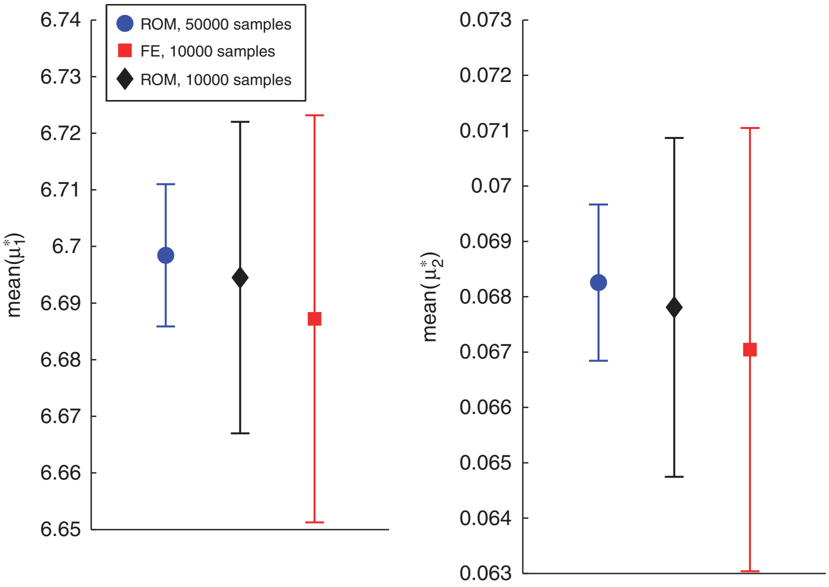

Figure 9. Ninty-five percent credible intervals for the mean of $\mu_1^* = \log(A^*)$ and $\mu_2^* = E^*$, computed using Markov chains with $10\,000$ and $50\,000$ samples for the two-dimensional problem. The reduced model was constructed using $n = 40$ and $m = 50$.

DOI: 10.1002/nme

This points to model reduction as a key enabler for solving statistical inverse problems involving large-scale systems that would otherwise be intractable. The full three-dimensional model of this combustor system is such a case, as described in the next section.

5.2. *Three-dimensional demonstration case*

In three dimensions, the computational domain is the parallelepiped shown in Figure 2. The flow velocity $\boldsymbol{U}$ is uniform in the positive x direction, and the inflow consists of a center square on the minimum-x boundary face. The output vector consists of 21 average concentrations measured on cut planes taken as shown in Figure 2. All reduced models were constructed using a set of $K=169$ snapshots generated with parameter vectors $\boldsymbol{\mu}^k$ on a uniform 13×13 grid in the parameter space $\mathscr{D}$. The size of this grid was in part driven by computational resource limits, although the errors were checked *a posteriori*.

5.2.1. DG discretization. A DG finite element discretization of (27) in three dimensions is performed with hexahedral elements. The solution/test space, V^{DG}, consists of piecewise-discontinuous tensor-product polynomials of order p_{DG} over the elements. On each element the basis is formed from tensor-product Lagrange functions, so that the total number of degrees of freedom is $N=N^e(p_{\mathrm{DG}}+1)^3$, where N^e is the number of elements. N discrete equations are obtained by multiplying (27) by the basis functions and integrating over the computational domain. Pure upwinding is used for the advection term, and interior-penalty stabilization is used for the diffusion term. Details on the DG discretization can be found in the literature [29, 30]. For completeness, the semilinear form for general $w,\ v\in V^{\mathrm{DG}}$ is given below:

$$\begin{aligned}\mathscr{R}(w,v;\boldsymbol{\mu}) = {} & \sum_{k=1}^{N_e}\int_{\Omega_k}(-\nabla v\cdot\boldsymbol{U}w+\nu\nabla v\cdot\nabla w+v\,f(w;\boldsymbol{\mu}))\,\mathrm{d}\mathbf{x}\\ & +\int_{\Gamma}(v^+-v^-)\widehat{\mathbf{F}}(w^+,w^-)\cdot\mathbf{n}\,\mathrm{d}s-\nu\int_{\Gamma}[\![w]\!]\cdot\{\nabla v\}\,\mathrm{d}s\\ & -\nu\int_{\Gamma}[\![v]\!]\cdot\{\nabla w\}\,\mathrm{d}s+\nu\int_{\Gamma\cup\Omega_D}\eta^f(h^f)^{-1}[\![v]\!]\cdot[\![w]\!]\,\mathrm{d}s\\ & +\int_{\partial\Omega_D}vu_D\boldsymbol{U}\cdot\mathbf{n}\,\mathrm{d}s-\nu\int_{\partial\Omega_D}(w-u_D)\nabla v\cdot\mathbf{n}\,\mathrm{d}s \end{aligned}\tag{31}$$

where Ω_k denotes the kth element, Γ denotes the set of interior faces, $(\cdot)^+/(\cdot)^-$ refers to quantities taken from the left/right sides of an interior face, $\mathbf{n}$ is the face normal vector, $[\![w]\!]=w^+\mathbf{n}^++w^-\mathbf{n}^-$, $\{w\}=(w^++w^-)/2$, $\widehat{\mathbf{F}}(w^+,w^-)$ is an upwind flux corresponding to $\boldsymbol{U}w$, h^f is a length measure for face f based on the face area and adjacent element volumes, and η^f is a stabilization constant [31].

Using (31) the discrete matrices and vectors in (1) are constructed similarly to the SUPG case, with the exception that $\mathbf{R}_0$ now contains the weakly enforced Dirichlet boundary data. Also, as the boundary conditions are enforced weakly, $\bar{\mathbf{u}}=0$ in (16). In the reduction of the non-linear term, the EIM was used to create the projection mask. The finite element mesh contains 314 928 quadratic elements, spaced uniformly over the domain, for a total of over 8.5 million unknowns. For each element, 64 quadrature points were used to approximate the non-linear term, giving $M=20,155,392$. The large number of unknowns ensures adequate resolution of the reaction front

DOI: 10.1002/nme

for the parameter range of interest and demonstrates the applicability of the model reduction technique to a large-scale problem.

5.2.2. Parallel implementation. To enable large-scale simulations, the DG solver and offline model reduction were implemented for parallel computation on distributed-memory clusters with MPI communication. The underlying premise of the implementation is that all operations involving N or M should be parallelized. Thus, solution of the full-scale system of equations during snapshot generation was performed using a parallelized Newton Generalized Minimal RESidual (GMRES) solver, using ParMetis for mesh partitioning [32]. All K snapshots were then loaded from disk and partitioned with each processor storing a partition of every snapshot vector. The POD coefficient matrix was constructed by performing K^2 inner products in parallel and gathering the resulting $K \times K$ matrix onto the root processor for eigenvalue calculation. The POD basis was then formed by broadcasting the eigenvectors of the POD correlation matrix to all the processors and calculating the appropriate linear combinations of snapshots in parallel.

The reduced quantities $\mathbf{R}_r$ and $\mathbf{A}_r$ in (25) were created in parallel by evaluating the linear residual $n+1$ times to obtain $\mathbf{R}_0$ and $\mathbf{A\Phi}$, and then dotting these vectors with $\mathbf{\Phi}$. Snapshots of the non-linear term were computed locally at element quadrature points on each processor, and then reduced via POD in the parallel manner described above to yield $\mathbf{\Phi}^s$. The mask $\mathbf{Z}$ was created via the EIM in parallel, using straightforward vector operations and maximizations. The matrix $\mathbf{E}_r$ in (25) was first formed locally on each processor by looping over the quadrature points of the processor-local elements, and then gathered/summed onto the root processor. Finally, rows of $\mathbf{D}_r$ were created by the processors containing the unknowns corresponding to the mask $\mathbf{Z}$, and they were assembled on the root processor. Note that $\bar{\mathbf{u}}_r$ is zero because $\bar{\mathbf{u}}$ is zero.

5.2.3. Performance of the reduced model. Reduced models of various sizes, n and m, were constructed using the set of $K=169$ snapshots and the EIM. The accuracy of each reduced model was assessed by computing an average relative error norm of outputs according to (30), using a 17×17 grid in parameter space for Ξ^{test}. Sample concentration profiles are shown in Figure 10, for both full and reduced solutions.

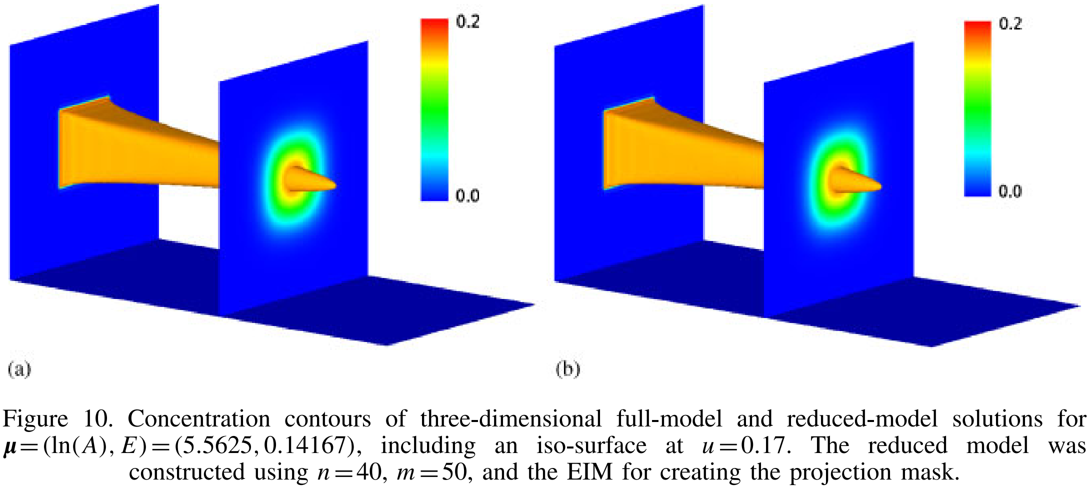

Figure 10. Concentration contours of three-dimensional full-model and reduced-model solutions for $\boldsymbol{\mu}=(\ln(A), E)=(5.5625, 0.14167)$, including an iso-surface at $u=0.17$. The reduced model was constructed using $n=40$, $m=50$, and the EIM for creating the projection mask.

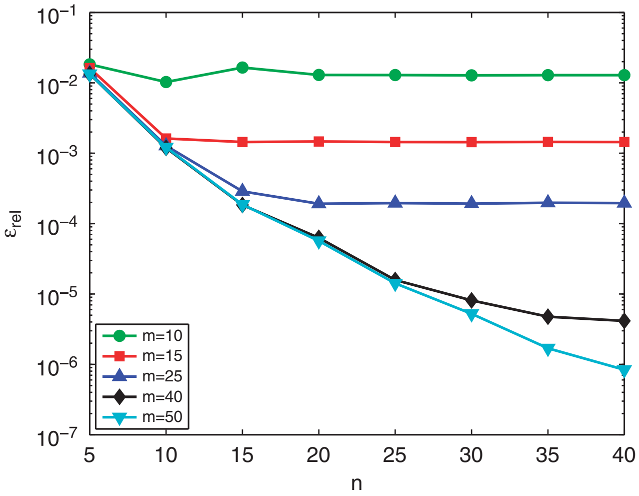

Figure 11. Average relative error norm of the output, ε_{rel}, as a function of n and m for the three-dimensional example. The EIM was used to construct the non-linear term projection mask.

Figure 11 shows ε_{rel} as a function of n and m. As in the two-dimensional results, for a given value of m, the error decreases as n is increased until a point is reached where the solution is dominated by the error due to the approximation of the non-linear term and convergence stalls.

As in two dimensions, the convergence of the reduced model with n is excellent. Furthermore, the relative computational time savings are even greater in three dimensions: all reduced models ran in under 0.023 s CPU time on a single processor, whereas each forward solve of the full-order model required a wall clock time of 13 h when run on 26 processors. Thus, an average relative error of 10^{-6} can be achieved with a reduced model that is more than two million times faster than the full-order model. Note that the symmetry of the problem chosen is such that only a quarter of the domain needs to be modeled. The fact that the entire domain is considered in these runs adversely affects the full-order computation time, whereas in the reduced model the symmetry is accounted for in the basis functions. Nevertheless, with a factor of four adjustment, or even one or two orders of magnitude if adaptive mesh refinement were employed, the speedup of the reduced model would still be crucial for the inverse problem application at hand.

5.2.4. Inverse problem solution. The statistical inverse problem was solved using the Bayesian inference approach with the reduced model generated with $n=40$ and $m=50$. The large cost of each full-order solve prohibited an inverse calculation with the full-order equations—even a 1000-sample MCMC would require approximately 1.5 years of CPU time. A set of measurements was generated according to (5) using $\boldsymbol{\mu}_e=(5.625, 0.1)$ and normally distributed errors with standard deviation set to 1.5% of the output mean. To determine the mean and variance of the input parameters, a Markov chain of size $N_m=50\,000$ was generated using the uniform proposal distribution with $\boldsymbol{\Delta}$ set to one quarter the dimensions of $\mathscr{D}$. The Markov chain was deemed converged, as the autocorrelation time, τ_g, was calculated to be less than 250, the sample sequences shown in Figure 12 show no obvious trends, and the autocovariance function was found to decrease smoothly with the lag before being dominated by noise. The marginal posterior histograms for both parameters are shown in Figure 13.

DOI: 10.1002/nme

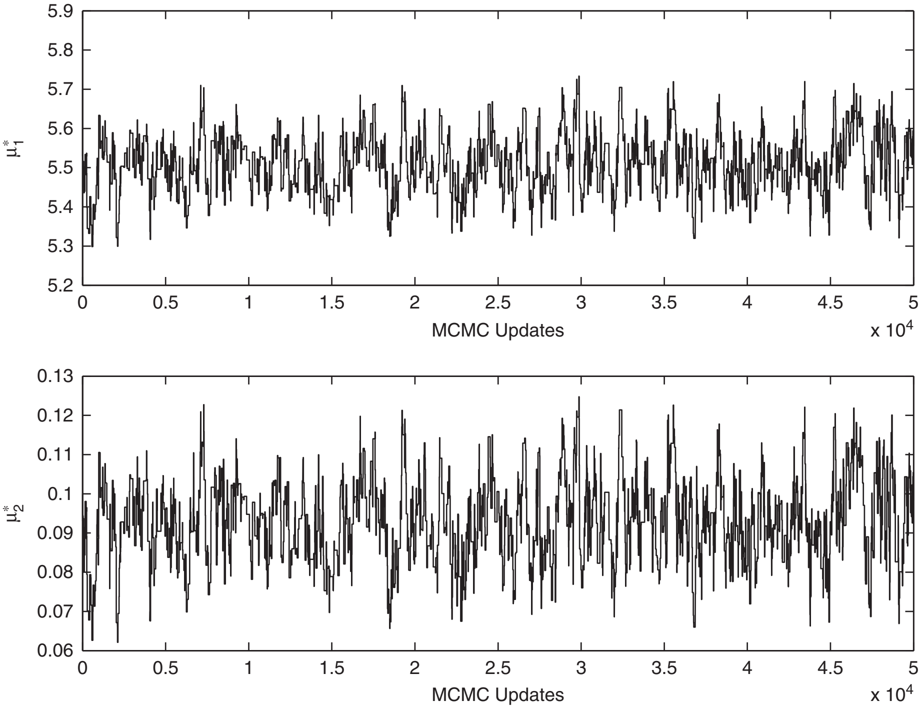

Figure 12. Markov chain samples from posterior probability distribution for the Arrhenius parameters obtained using the reduced model with $n=40$ and $m=50$ for the three-dimensional example. The uniform proposal was used in the Markov chain generation step, yielding an acceptance rate of 3.6%.

The mean estimates for the parameters computed from the Markov chain samples are 5.512 ± 0.011 for $\ln(A)$ and 0.093 ± 0.0015 for E. The standard deviation estimates are 0.075 ± 0.0056 for $\ln(A)$ and 0.0107 ± 0.0008 for E. The accompanying intervals are the 95% credible intervals obtained from the estimator variance calculations. As in the two-dimensional example, the mean values deviate from the original values in $\boldsymbol{\mu}_e$ because the inversion was performed for one realization of the measurement errors.

The Markov chain acceptance rate, defined as the number of accepted proposals divided by N_m, was observed to be 3.4% for the uniform proposal. This low acceptance rate can be attributed to the anisotropy in the posterior distribution, which is shown in Figure 14. To improve the acceptance rate, a stretched-Gaussian proposal was implemented, as described in Section 3.3, using $\delta = 1.5$ for the dimensionless step-size parameter and finite differencing for calculation of $\partial \mathbf{y}/\partial \boldsymbol{\mu}$ in (10). MCMC experiments indicated an acceptance rate of 27% using this stretched proposal, and hence many fewer samples were required for the desired statistics. Specifically, using $N_m = 5000$, the calculated statistics were: mean estimates of 5.514 ± 0.0075 for $\ln(A)$ and 0.0936 ± 0.0011 for E; standard deviation estimates of 0.078 ± 0.0054 for $\ln(A)$ and 0.0111 ± 0.00075 for E. Note that the estimator credible intervals are tighter compared with the $N_m = 50\,000$ Markov chain, even though an order of magnitude fewer samples was taken. Accounting for the finite differencing in the linearization calculations, using the stretched-Gaussian proposal yields a cost savings of over

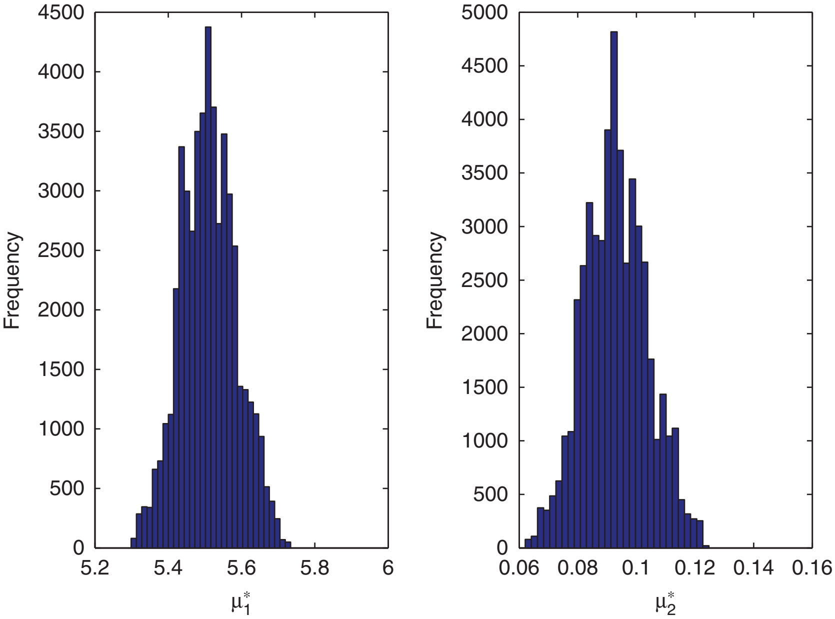

Figure 13. Marginal posterior histograms for $\mu_1^* = \log(A^*)$ and $\mu_2^* = E^*$ for the three-dimensional example with an $N_M = 50000$ Markov chain generated using the reduced model with $n = 40$ and $m = 50$ and the uniform proposal.

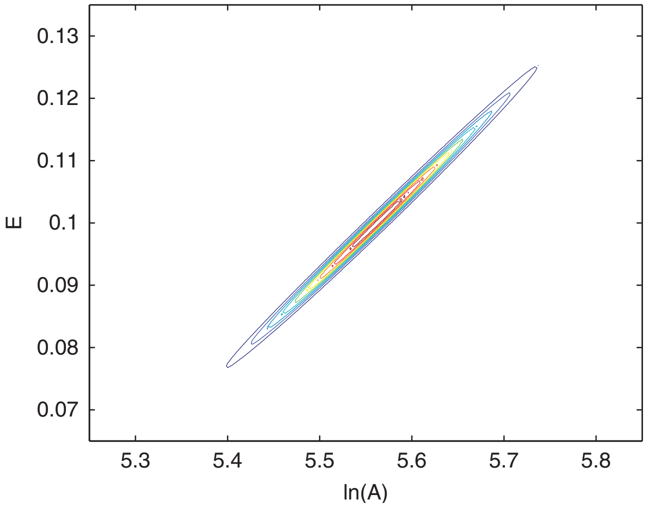

Figure 14. Iso-probability contours of the posterior distribution (7) in parameter space, generated by sampling the parameter space with a 256×256 grid, for the three-dimensional example with $\boldsymbol{\mu}_e = (5.625, 0.1)$ and $\sigma = 1.5\%$ of the output mean. The reduced model with $n = 40$ and $m = 50$ was used for the sampling.

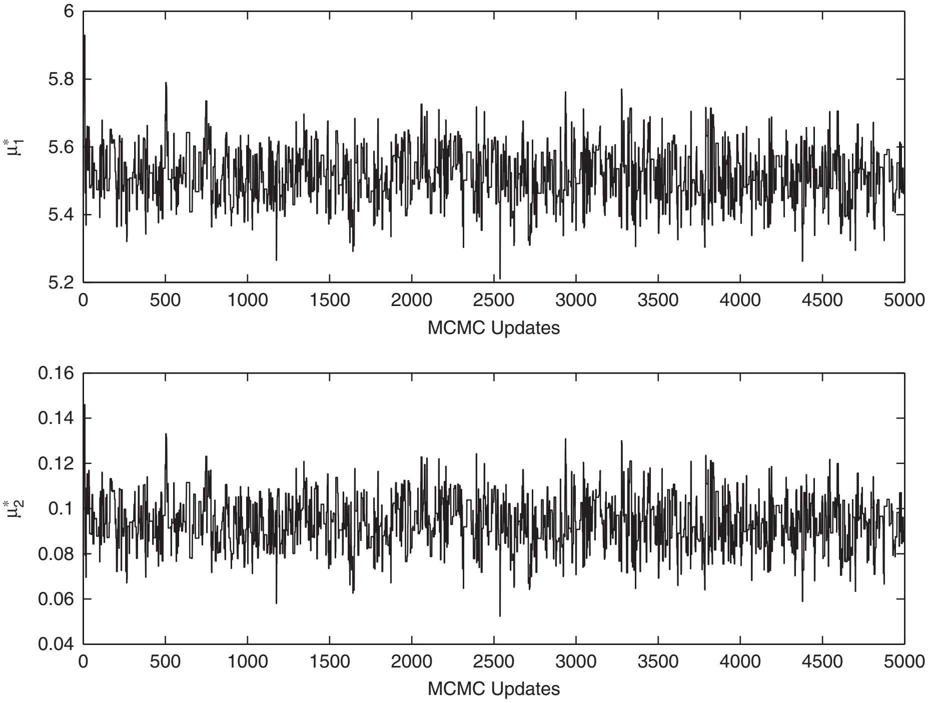

Figure 15. Markov chain samples from posterior probability distribution for the Arrhenius parameters obtained using the reduced model with $n=40$ and $m=50$ for the three-dimensional example. The stretched-Gaussian proposal was used in the Markov chain generation step, yielding an acceptance rate of 27%.

a factor of 3. Furthermore, the Markov chain was deemed converged, as the autocorrelation time, τ_g, was calculated to be less than 10, the sample sequences shown in Figure 15 show no obvious trends, and the autocovariance function was found to decrease smoothly with the lag before being dominated by noise. The resulting marginal posterior histograms for the two parameters are shown in Figure 16.

6. CONCLUSIONS

This paper presents an application of non-linear model reduction to an inverse problem solution in a Bayesian inference setting. The inherent multiple-query context of the Bayesian approach makes model reduction an attractive option for large-scale problems. Indeed accurate results are obtained by using the reduced model in MCMC simulations at computational costs several orders of magnitude less than that of the full-order model.

The non-linear model reduction is presented in a discrete projection framework in which masked projection is used to reduce the number of evaluations of the non-linear term. Similarities between this method and existing techniques, such as gappy POD, missing point estimation, and coefficient-function approximation, are identified. In addition, a parallel implementation of the reduction

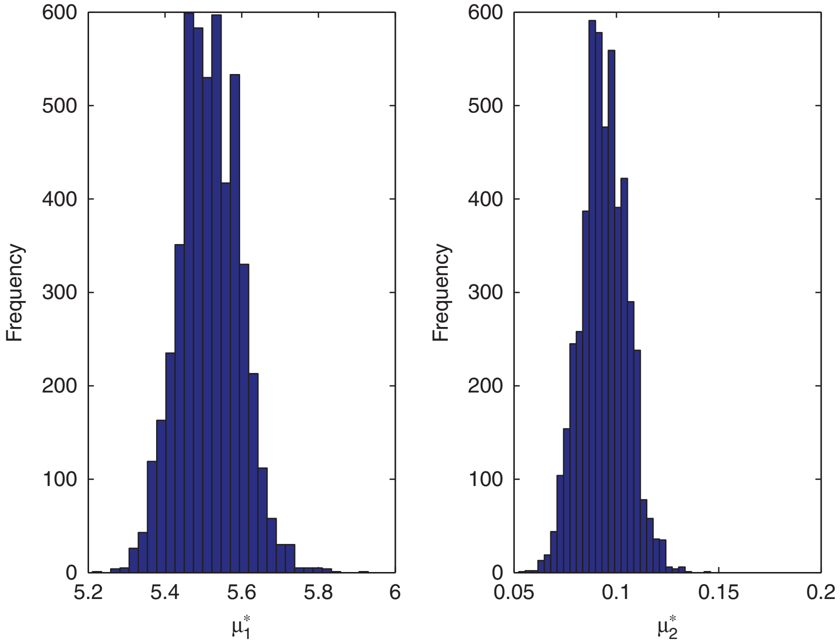

Figure 16. Marginal posterior histograms for $\mu_1^* = \log(A^*)$ and $\mu_2^* = E^*$ for the three-dimensional example with an $N_M = 5000$ Markov chain generated using the reduced model with $n = 40$ and $m = 50$ and the stretched-Gaussian proposal.

procedure is presented and demonstrated on a large-scale problem with 8.5 million degrees of freedom. Although only steady-state test problems are considered in this work, the discrete projection framework extends naturally to unsteady problems, where the discrete system is augmented by a mass matrix multiplying the time derivative of **u**. In this case, the accuracy of the reduced model will depend on how well the chosen basis represents the time history of **u**.

While grid-based parameter sampling for the snapshots was sufficient for the two parameters considered in this work, a model-constrained adaptive sampling approach [23, 24] will be necessary as the number of parameters is increased. In addition, although relatively high-accuracy reduced models were used in the MCMC approach, further work is required to quantify the effect of model reduction errors on the statistics of interest.

ACKNOWLEDGEMENTS

This work was supported in part by the Singapore-MIT Alliance Computational Engineering Programme, the Air Force Office of Sponsored Research under grant FA9550-06-0271 (program director Dr Fariba Fahroo), the National Science Foundation under DDDAS grants CNS-0540372 and CNS-0540186 (program directors Dr Frederica Darema and Dr Anita Lasalle), and a Graduate Fellowship from the Caja Madrid Foundation.

REFERENCES

1. Kaipio JP, Somersalo E. *Statistical and Computational Inverse Problems*. Springer: Berlin, 2005.
2. Tarantola A. *Inverse Problem Theory and Methods for Model Parameter Estimation*. SIAM: Philadelphia, 2005.
3. Arridge SR, Kaipio JP, Kolehmainen V, Schweiger M, Somersalo E, Tarvainen T, Vauhkonen M. Approximation errors and model reduction with an application in optical diffusion tomography. *Inverse Problems* 2006; **22**: 175–195.
4. Kaipio JP, Somersalo E. Statistical inverse problems: discretization, model reduction and inverse crimes. *Journal of Computational and Applied Mathematics* 2007; **198**:493–504.
5. Marzouk YM, Najm HN, Rahn LA. Stochastic spectral methods for efficient Bayesian solution of inverse problems. *Journal of Computational Physics* 2007; **224**(2):560–586.
6. Ghanem RG, Red-Horse J. Propagation of probabilistic uncertainty in complex physical systems using a stochastic finite element approach. *Physica D* 1999; **133**(1/4):137–144.
7. Babuška I, Nobile F, Tempone R. A stochastic collocation method for elliptic partial differential equations with random input data. *SIAM Journal on Numerical Analysis* 2007; **45**(3):1005–1034.
8. Efendiev Y, Hou T, Luo W. Preconditioned Markov chain Monte Carlo simulations using coarse-scale models. *SIAM Journal on Scientific Computing* 2007; **28**(2):776–803.
9. Kosambi DD. Statistics in function space. *Journal of Indian Mathematical Society* 1943; **7**:76–88.
10. Holmes P, Lumley JL, Berkooz G. *Turbulence*, *Coherent Structures*, *Dynamical Systems and Symmetry*. Cambridge University Press: Cambridge, U.K., 1996.
11. Sirovich L. Turbulence and the dynamics of coherent structures. Part 1: coherent structures. *Quarterly of Applied Mathematics* 1987; **45**(3):561–571.
12. Wang J, Zabaras N. Using Bayesian statistics in the estimation of heat source in radiation. *International Journal of Heat and Mass Transfer* 2005; **48**:15–29.
13. Chen Y. Model order reduction for nonlinear systems. *Master's Thesis*, MIT, Cambridge, MA, June 1999.
14. Rewienski M, White J. A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. *IEEE Transaction on Computer-aided Design of Integrated Circuits and Systems* 2003; **22**(2):155–170.
15. Bos R, Bombois X, van den Hof P. Accelerating large-scale nonlinear models for monitoring and control using spatial and temporal correlations. *Proceedings of American Control Conference*, Boston, U.S.A., 2004.
16. Astrid P, Weiland S, Willcox K, Backx T. Missing point estimation in models described by proper orthogonal decomposition. *IEEE Transactions on Automatic Control* 2008; **53**(10):2237–2251.
17. Barrault M, Maday Y, Nguyen NC, Patera AT. An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathématique Académie des Sciences. Paris* 2004; **I**:339–667.
18. Grepl MA, Maday Y, Nguyen NC, Patera AT. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *Mathematical Modelling and Numerical Analysis* (*M2AN*) 2007; **41**(3):575–605.
19. Everson R, Sirovich L. The Karhunen–Loéve procedure for gappy data. *Journal of the Optical Society of America* 1995; **12**(8):1657–1664.
20. Mosegaard K, Tarantola A. Probabilistic approach to inverse problems. *International Handbook of Earthquake and Engineering Seismology* (*Part A*). Academic Press: New York, 2002; 237–265.
21. Hastings WK. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 1970; **57**(1):97–109.
22. Calvetti D, Somersalo E. *Introduction to Bayesian Scientific Computing*: *Ten Lectures on Subjective Computing*. Springer: Berlin, 2008.
23. Bui-Thanh T, Willcox K, Ghattas O. Model reduction for large-scale systems with high-dimensional parametric input space. *SIAM Journal on Scientific Computing* 2008; **30**(6):3270–3288.
24. Galbally D. Nonlinear model reduction for uncertainty quantification in large-scale inverse problems: application to nonlinear convection–diffusion-reaction equations. *Master's Thesis*, MIT, Cambridge, MA, January 2008.
25. Bui-Thanh T, Damodaran M, Willcox K. Aerodynamic data reconstruction and inverse design using proper orthogonal decomposition. *AIAA Journal* 2004; **42**(8):1505–1516.
26. Willcox K. Unsteady flow sensing and estimation using the gappy proper orthogonal decomposition. *Computers and Fluids* 2006; **35**:208–226.
27. Nguyen NC, Patera AT, Peraire J. A 'best points' interpolation method for efficient approximation of parametrized functions. *International Journal for Numerical Methods in Engineering* 2007; DOI: 10.1002/nme.2086.

28. Brooks AN, Hughes TJR. Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 1982; **32**:199–259.
29. Cockburn B, Shu C-W. Runge–Kutta discontinuous Galerkin methods for convection-dominated problems. *Journal of Scientific Computing* 2001; **16**(3):173–261.
30. Arnold DN, Brezzi F, Cockburn B, Donatella Marini L. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM Journal on Numerical Analysis* 2002; **39**(5):1749–1779.
31. Shahbazi K. An explicit expression for the penalty parameter of the interior penalty method. *Journal of Computational Physics* 2005; **205**:401–407.
32. Karypis G, Kumar V. Parallel multilevel k-way partitioning scheme for irregular graphs. *SIAM Review* 1999; **41**(2):278–300.