INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2009; **80**:846–867
Published online 16 January 2009 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme.2547

Nonparametric density estimation for randomly perturbed elliptic problems II: Applications and adaptive modeling

D. Estep[1,*,†], A. Målqvist[2] and S. Tavener[3]

[1]*Department of Mathematics and Department of Statistics, Colorado State University, Fort Collins, CO 80523, U.S.A.*
[2]*Department of Information Technology, Uppsala University, SE-751 05 Uppsala, Sweden*
[3]*Department of Mathematics, Colorado State University, Fort Collins, CO 80523, U.S.A.*

SUMMARY

In this paper, we develop and apply an efficient adaptive algorithm for computing the propagation of uncertainty into a quantity of interest computed from numerical solutions of an elliptic partial differential equation with a randomly perturbed diffusion coefficient. The algorithm is well-suited for problems for which limited information about the random perturbations is available and when an approximation of the probability distribution of the output is desired. We employ a nonparametric density estimation approach based on a very efficient method for computing random samples of elliptic problems described and analyzed in (*SIAM J. Sci. Comput.* 2008. DOI: JCOMP-D-08-00261). We fully develop the adaptive algorithm suggested by the analysis in that paper, discuss details of its implementation, and illustrate its behavior using a realistic data set. Finally, we extend the analysis to include a 'modeling error' term that accounts for the effects of the resolution of the statistical description of the random variation. We modify the adaptive algorithm to adapt the resolution of the statistical description and illustrate the behavior of the adaptive algorithm in several examples. Copyright © 2009 John Wiley & Sons, Ltd.

Received 22 July 2008; Revised 28 October 2008; Accepted 1 December 2008

KEY WORDS: *a posteriori* error analysis; adaptive modeling; adjoint problem; density estimation; elliptic problem; nonparametric density estimation; random diffusion; sensitivity analysis; random perturbation

*Correspondence to: D. Estep, Department of Mathematics and Department of Statistics, Colorado State University, Fort Collins, CO 80523, U.S.A.
†E-mail: estep@math.colostate.edu

Contract/grant sponsor: Department of Energy; contract/grant numbers: DE-FG02-04ER25620, DE-FG02-05ER25699, DE-FC02-07ER54909
Contract/grant sponsor: Idaho National Laboratory; contract/grant number: 00069249
Contract/grant sponsor: Lawrence Livermore National Laboratory; contract/grant number: B573139
Contract/grant sponsor: National Aeronautics and Space Administration; contract/grant number: NNG04GH63G
Contract/grant sponsor: National Science Foundation; contract/grant numbers: DMS-0107832, DMS-0715135, DGE-0221595003, MSPA-CSE-0434354, ECCS-0700559
Contract/grant sponsor: Sandia Corporation; contract/grant number: PO299784

1. INTRODUCTION

The practical application of differential equations to model physical phenomena presents problems in both computational mathematics and statistics. The mathematical issues arise because of the need to compute approximate solutions of difficult problems while statistics arises because of the need to incorporate experimental data and model uncertainty. In this paper, we consider the propagation of uncertainty into a quantity of interest computed from numerical solutions of an elliptic partial differential equation with a randomly perturbed diffusion coefficient. The problem is to compute a quantity of interest $Q(U)$, expressed as a linear functional, of the solution $U \in \mathscr{H}_0^1(\Omega)$ of

$$\begin{aligned} -\nabla \cdot \mathscr{A} \nabla U &= f, \quad x \in \Omega \\ U &= 0, \quad x \text{ in } \partial\Omega \end{aligned} \tag{1}$$

where $f \in L^2(\Omega)$ is a given deterministic function, Ω is a convex polygonal domain with boundary $\partial\Omega$ and $\mathscr{A}(x)$ is a stochastic function that varies randomly according to some given probability structure. The problem (1) is interpreted to hold almost surely (a.s.), i.e. with probability 1. Under suitable assumptions, e.g. $\mathscr{A}$ is uniformly bounded and uniformly coercive and has piecewise smooth dependence on its inputs (a.s.) with continuous and bounded covariance functions, $Q(U)$ is a random variable. The approach we use extends to problems with more general Dirichlet or Robin boundary conditions in which both the right-hand side and data for the boundary conditions are randomly perturbed as well as problems with more general elliptic operators in a straightforward way.

There are several factors that affect the choice of an approach to this problem.

1. The available information about the stochastic nature of $\mathscr{A}$.
2. The information that is needed about the output uncertainty, e.g. one or two moments or the (approximate) probability distribution.
3. The fact that only a finite number $\mathscr{N}$ of solution realizations can be computed.
4. The solution of (1) has to be computed numerically, which is both expensive and leads to significant variation in the numerical error as the coefficients and data vary.

We consider problems for which the output distribution is unknown and/or complicated, e.g. multimodal, and for which there is a limited availability of information about the diffusion coefficient, e.g. it is only possible to determine a set of representative values at certain locations in the domain. Moreover, we desire to compute an approximate probability distribution rather than one or two moments. This is important in the context of multiphysics problems, for example, where the output of this model might enter as input into yet other models.

One powerful approach for uncertainty quantification for elliptic problems is based on the use of Karhunen-Loéve, Polynomial Chaos, or other orthogonal expansions of the random vectors [1, 2]. This approach requires detailed knowledge of the probability distributions for the input variables that is often not available. In many situations, e.g. some oil reservoir simulations [3], we have only a relatively few experimental values available. In this case, it is useful to consider the propagation of uncertainty as a nonparametric density estimation problem in which we seek to compute an approximate distribution for the output random variable using a Monte Carlo sampling method. Samples $\{\mathscr{A}^n\}$ are drawn from its distribution, solutions $\{U^n\}$ are computed to produce samples $\{Q(U^n)\}$, and the output distribution is approximated using a binning strategy coupled with smoothing.

DOI: 10.1002/nme

The two main ingredients determining the cost of this computational approach are the number of samples required for the Monte Carlo simulation and the cost of computing each sample. There is a large, and growing, literature on efficient random sampling. We do not address that issue here. Rather in [4], we address the cost of computing samples by developing a numerical procedure for computing randomly generated sample solutions of (1), which has the property that only a fixed number of linear systems need to be inverted, regardless of the number of samples that are computed. Since the cost of solving (1) is largely determined by the linear systems that must be solved, this approach greatly reduces the cost of random sampling. In [4] we also derive computable *a posteriori* error estimates that account for the significant effects of numerical error and uncertainty and which provide the basis for an adaptive error control algorithm that guides the adjustment of *all* computational parameters affecting accuracy.

In this paper, we fully develop the adaptive algorithm and discuss details of its implementation. Then we illustrate the adaptive algorithm using a realistic data set. In addition, we explore the modeling assumption underlying the basis for this new approach. The method depends on an assumption that the random variation of the diffusion coefficient is described as a piecewise constant function. We show that the *a posteriori* estimate can be extended to include a 'modeling error' term that accounts for the effects of varying the resolution of the statistical description of the random variation. We then describe a modification to the adaptive algorithm that provides a mechanism for adapting the resolution of the statistical description in order to compute a desired quantity of interest to a specified accuracy. We illustrate the behavior of this algorithm with several examples.

In [5], we give a theoretical convergence analysis of a generalization of the method presented in this paper. In particular, we relax the assumption that A is a piecewise constant and instead assume that it is a piecewise polynomial function. In particular, this makes it possible to use continuous perturbations.

2. A MODELING ASSUMPTION

We assume that the stochastic diffusion coefficient can be written

$$\mathscr{A} = a + A$$

where the uniformly coercive, bounded deterministic function a, may have multiscale behavior, and A is a relatively small piecewise constant function with random coefficients. We assume that $a(x) \geqslant a_0 > 0$ for $x \in \Omega$ and that $|A(x)| \leqslant \delta a(x)$ for some $0 < \delta < 1$.

To describe A, we let $\{\Omega_d,\ d = 1, \ldots, \mathscr{D}\}$, be a decomposition of Ω into a finite set of nonoverlapping polygonal sub-domains with $\bigcup \Omega_d = \Omega$. We denote the boundaries by $\partial\Omega_d$ and outward normals by $\mathbf{n}_d$. We let χ_{Ω_d} denote the characteristic function for the set Ω_d. We assume that

$$A(x) = \sum_{d=1}^{\mathscr{D}} A^d\ \chi_{\Omega_d}(x), \quad x \in \Omega \tag{2}$$

where (A^d) is a random vector and each coefficient A^d is associated with a given probability distribution. We plot a diffusion coefficient a so randomly perturbed in Figure 1.

DOI: 10.1002/nme

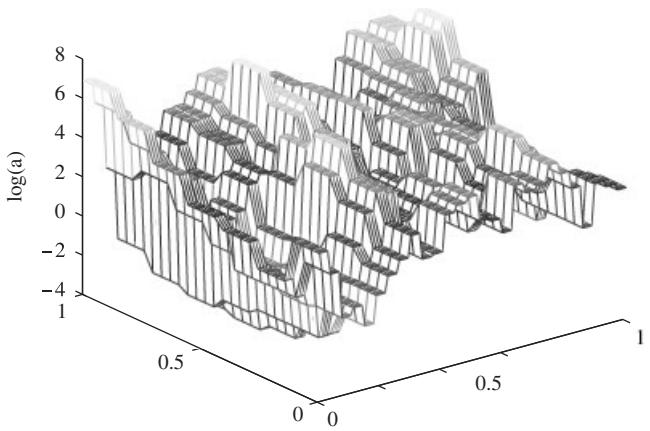

Figure 1. Illustration of the modeling assumption (2). The unit square is partitioned into 27×7 identically shaped squares. The diffusion coefficient a varies considerably through the domain. The random perturbations are uniform on the intervals determined by $\pm 20\%$ of the value of a in each sub-domain.

The coefficients A^d are quantities that may be determined experimentally, e.g. by measurements taken at specific points in the domain Ω. Improving the model requires choosing a finer partition and taking more measurements A^d.

2.1. Notation

We let $\Omega\subset\mathbf{R}^d,\ d=2,3$, denote the piecewise polygonal computational domain with boundary $\partial\Omega$. For an arbitrary domain $\omega\subset\Omega$ we denote $L^2(\omega)$ scalar product by $(v,w)_\omega=\int_\omega vw\,\mathrm{d}x$ in the domain and $\langle v,w\rangle_{\partial\omega}=\int_{\partial\omega}vw\,\mathrm{d}s$ on the boundary, with associated norms $\|\ \|_\omega$ and $|\ |_\omega$. We let $\mathscr{H}^s(\omega)$ denote the standard Sobolev space of smoothness s for $s\geqslant 0$. In particular, $\mathscr{H}^1_0(\Omega)$ denotes the space of functions in $\mathscr{H}^1(\Omega)$ for which the trace is 0 on the boundary. In addition,

$$\mathscr{H}^1_{0,\partial\Omega}(\Omega_d)=\{v\in\mathscr{H}^1(\Omega_d):v|_{\partial\Omega_{\mathscr{D}}\cap\partial\Omega}=0\}$$

If $\omega=\Omega$, we drop ω and also if $s=0$ we drop s, i.e. $\|\cdot\|$ denotes the $L^2(\Omega)$-norm.

Generally, capital letters denote random variables, or samples, and lower case letters represent deterministic variables or functions. We assume that any random vector X is associated with a probability space $(\Lambda,\mathscr{B},P)$ in the usual way. We let $\{X^n,n=1,\ldots,\mathscr{N}\}$ denote a collection of samples. We assume that it is understood how to draw these samples. We let $E(X)$ denote the expected value, $\mathrm{Var}(X)$ denote the variance, and $F(t)=P(X<t)$ denote the cumulative distribution function. We compute approximate cumulative distribution functions in order to determine the probability distribution of a random variable. Let $\mathscr{A}^n=a+A^n$ be a particular sample of the diffusion coefficient with corresponding solution U^n. On Ω_d, we denote a finite set of samples by $\{A^{n,d},n=1,\ldots,\mathscr{N}\}$.

For a function $\mathscr{A}^n$ on Ω, $\mathscr{A}^{n,d}$ means $\mathscr{A}^n$ restricted to Ω_d. For $d=1,\ldots,\mathscr{D}$, d' denotes the set of indices in $\{1,2,\ldots,\mathscr{D}\}\backslash\{d\}$ for which the corresponding domains $\Omega_{d'}$ share a common boundary with Ω_d. Following the notation above, we let $(\cdot,\cdot)_d$ denote the $L^2(\Omega_d)$ scalar product, and $\langle\cdot,\cdot\rangle_{d\cap\tilde{d}}$ denote the $L^2(\partial\Omega_d\cap\partial\Omega_{\tilde{d}})$ scalar product for $\tilde{d}\in d'$.

DOI: 10.1002/nme

We use the finite element method to compute numerical solutions. Let $\mathscr{T}_h = \{\tau\}$ be a quasi-uniform partition into elements that $\bigcup \tau = \Omega$. We assume that the finite element discretization $\mathscr{T}_h$ is obtained by refinement of $\{\Omega_d\}$. This is natural when the diffusion coefficient a and the data vary on a scale finer than the partition $\{\Omega_d\}$. Associated with $\mathscr{T}_h$, we define the discrete finite element space $\mathscr{V}_h$ consisting of continuous, piecewise linear functions on $\mathscr{T}$ satisfying Dirichlet boundary conditions, with mesh size function $h_\tau = \mathrm{diam}(\tau)$ for $x \in \tau$ and $h = \max_{\tau \in \mathscr{T}_h} h_\tau$. In some situations, we use a more accurate numerical solution computed using a finite element space $\mathscr{V}_{\tilde{h}}$ that is either the space of continuous, piecewise quadratic functions $\mathscr{V}_h^2$ or involves a refinement $\mathscr{T}_{\tilde{h}}$ of $\mathscr{T}_h$ where $\tilde{h} \ll h$.

3. THE COMPUTATIONAL METHOD

The method is based on a nonoverlapping domain decomposition method [6] and is iterative, hence, for a function $\mathscr{U}$ involved in the iteration, $\mathscr{U}_i$ denotes the value at the ith iteration. We let $\{U_0^{n,d}, d = 1, \ldots, \mathscr{D}\}$ denote a set of initial guesses for solutions in the sub-domains. Given the initial conditions, for each $i \geqslant 1$, we solve the $\mathscr{D}$ problems

$$
\begin{aligned}
-\nabla \cdot \mathscr{A}^n \nabla U_i^{n,d} &= f, \quad x \in \Omega_d \\
U_i^{n,d} &= 0, \quad x \in \partial\Omega_d \cap \partial\Omega \\
\frac{1}{\lambda} U_i^{n,d} + \mathbf{n}_d \cdot \mathscr{A}^n \nabla U_i^{n,d} &= \frac{1}{\lambda} U_{i-1}^{n,\tilde{d}} - \mathbf{n}_{\tilde{d}} \cdot \mathscr{A}^n \nabla U_{i-1}^{n,\tilde{d}}, \quad x \in \partial\Omega_d \cap \partial\Omega_{\tilde{d}}, \ \ \tilde{d} \in d'
\end{aligned}
$$

where the parameter $\lambda \in \mathbf{R}$ is chosen to minimize the number of iterations. In practice, we compute $\mathscr{I}$ iterations. Note that the problems can be solved independently.

To discretize, we let $\mathscr{V}_{h,d} \subset \mathscr{H}^1_{0,\partial\Omega}(\Omega_d)$ be a finite element approximation space corresponding to the mesh $\mathscr{T}_d$ on Ω_d. For each $i \geqslant 1$, we compute $U_i^{n,d} \in \mathscr{V}_{h,d}$, $d = 1, \ldots, \mathscr{D}$, solving

$$
\begin{aligned}
&(\mathscr{A}^n \nabla U_i^{n,d}, \nabla v)_d + \sum_{\tilde{d} \in d'} \left(\frac{1}{\lambda} \langle U_i^{n,d}, v \rangle_{d \cap \tilde{d}} - \langle \mathbf{n}_{\tilde{d}} \cdot \mathscr{A}^n \nabla U_i^{n,d}, v \rangle_{d \cap \tilde{d}} \right) \\
&\quad = (f, v)_d + \sum_{\tilde{d} \in d'} \left(\frac{1}{\lambda} \langle U_{i-1}^{n,\tilde{d}}, v \rangle_{d \cap \tilde{d}} - \langle \mathbf{n}_{\tilde{d}} \cdot \mathscr{A}^n \nabla U_{i-1}^{n,\tilde{d}}, v \rangle_{d \cap \tilde{d}} \right) \quad \text{all } v \in \mathscr{V}_{h,d}
\end{aligned} \tag{3}
$$

It is convenient to use the matrix form of (3). We let $\{\varphi_m^d, m = 1, \ldots, \mathbf{n}_d\}$ be the finite element basis functions for the space $\mathscr{V}_{h,d}$, $d = 1, \ldots, \mathscr{D}$. We let $\mathbf{U}_i^{n,d}$ denote the vector of basis coefficients of $U_i^{n,d}$ with respect to $\{\varphi_m^d\}$. On each domain Ω_d,

$$
(\mathbf{k}^{a,d} + \mathbf{k}^{n,d}) \mathbf{U}_i^{n,d} = \mathbf{b}^d(f) + \mathbf{b}^{n,d}(\mathscr{A}^n, U_{i-1}^{n,d'})
$$

where

$$
\begin{aligned}
(\mathbf{k}^{a,d})_{lk} &= (a \nabla \varphi_l^d, \nabla \varphi_k^d)_d + \sum_{\tilde{d} \in d'} \left(\frac{1}{\lambda} \langle \varphi_l^d, \varphi_k^d \rangle_{d \cap \tilde{d}} - \langle \mathbf{n}_{\tilde{d}} \cdot a \nabla \varphi_l^d, \varphi_k^d \rangle_{d \cap \tilde{d}} \right) \\
(\mathbf{k}^{n,d})_{lk} &= (A^{n,d} \nabla \varphi_l^d, \nabla \varphi_k^d)_d - \sum_{\tilde{d} \in d'} \langle \mathbf{n}_{\tilde{d}} \cdot A^{n,d} \nabla \varphi_l^d, \varphi_k^d \rangle_{d \cap d'}
\end{aligned}
$$

DOI: 10.1002/nme

$$(\mathbf{b}^d)_k = (f, \varphi_k^d)_d$$

$$(\mathbf{b}^{n,d})_k = \sum_{\tilde{d}\in d'} \left(\frac{1}{\lambda} \langle U_{i-1}^{n,\tilde{d}}, \varphi_k^d \rangle_{d\cap\tilde{d}} - \langle \mathbf{n}_{\tilde{d}} \cdot \mathscr{A}^n \nabla U_{i-1}^{n,\tilde{d}}, \varphi_k^d \rangle_{d\cap\tilde{d}} \right)$$

for $1 \leqslant l, k \leqslant M$. We abuse notation mildly by denoting the dependence of the data $\mathbf{b}^{n,d}(\mathscr{A}^n, U_{i-1}^{n,d'})$ on the values of $U_{i-1}^{n,\tilde{d}}$ for $\tilde{d} \in d'$.

Next we use the fact that $A^{n,d}$ is constant on each Ω_d. Consequently, the matrix $\mathbf{k}^{n,d}$ has coefficients

$$(\mathbf{k}^{n,d})_{lk} = (A^{n,d}\nabla\varphi_l^d, \nabla\varphi_k^d)_d = A^{n,d}(\nabla\varphi_l^d, \nabla\varphi_k^d)_d = A^{n,d}(\mathbf{k}^d)_{lk}$$

where $\mathbf{k}^d$ is the standard stiffness matrix with coefficients $(\mathbf{k}^d)_{lk} = (\nabla\varphi_l^d, \nabla\varphi_k^d)_d$. As $A^{n,d}$ is relatively small, the Neumann series for the inverse of a perturbation of the identity matrix gives

$$(\mathbf{k}^{a,d} + A^{n,d}\mathbf{k}^d)^{-1} = \sum_{p=0}^{\infty} (-A^{n,d})^p ((\mathbf{k}^{a,d})^{-1}\mathbf{k}^d)^p (\mathbf{k}^{a,d})^{-1}$$

where id is the identity matrix. We compute only $\mathscr{P}$ terms in the Neumann expansion to generate the approximation,

$$\mathbf{U}_{\mathscr{P},i}^{n,d} = \sum_{p=0}^{\mathscr{P}-1} ((-A^{n,d})^p ((\mathbf{k}^{a,d})^{-1}\mathbf{k}^d)^p)\ (\mathbf{k}^{a,d})^{-1}(\mathbf{b}^d(f) + \mathbf{b}^{n,d}(\mathscr{A}^n, U_{\mathscr{P},i-1}^{n,d'})) \tag{4}$$

Note that $\mathbf{b}^{n,d}$ is nonzero only at boundary nodes. If $\mathscr{W}_{h,d}$ denotes the set of vectors determined by the finite element basis functions associated with the boundary nodes on Ω_d, then $\mathbf{b}^{n,d}$ is in the span of $\mathscr{W}_{h,d}$. We let $U_{\mathscr{P},\mathscr{J}}^{n,d}$ denote the finite element functions determined by $\mathbf{U}_{\mathscr{P},\mathscr{J}}^{n,d}$ for $n = 1, \ldots, \mathscr{N}$ and $d = 1, \ldots, \mathscr{D}$. We let $U_{\mathscr{P},\mathscr{J}}^{n}$ denote the finite element function that is equal to $U_{\mathscr{P},\mathscr{J}}^{n,d}$ on Ω_d.

We summarize as an Algorithm in Algorithm 1. Note that the number of linear systems that have to be solved in Algorithm 1 is independent of $\mathscr{N}$. Hence, there is a potential for enormous savings when the number of samples is large. We discuss this below in Section 4.

It is crucial for the method that the Neumann series converges. The following theorem from [4] shows convergence under the reasonable assumption that the random perturbations $A^{n,d}$ to the diffusion coefficient are smaller than the coefficient. We let $|||v|||_d^2 = \|\nabla v\|_{\Omega_d}^2 + \varepsilon|v|_{\partial\Omega_d}^2$ for some $\varepsilon > 0$. We define the matrices $\mathbf{c}^{n,d} = -A^{n,d}(\mathbf{k}^{a,d})^{-1}\mathbf{k}^d$ and denote the corresponding operators on the finite element spaces by $c^{n,d} : \mathscr{V}_{h,d} \to \mathscr{V}_{h,d}$.

Theorem 3.1
If $\eta = |\max\{A^{n,d}\}/a_0| < 1$, then

$$\left|\left|\left| \left((1-c^{n,d})^{-1} - \sum_{p=0}^{\mathscr{P}-1} (c^{n,d})^p \right) v \right|\right|\right|_d \leqslant \frac{\eta^{\mathscr{P}}}{1-\eta^{\mathscr{P}}} \left|\left|\left| \sum_{p=0}^{\mathscr{P}-1} (c^{n,d})^p v \right|\right|\right|_d \tag{5}$$

for any $v \in \mathscr{V}_{h,d}$.

DOI: 10.1002/nme

Algorithm 1 Monte Carlo domain decomposition finite element method

for $d=1,\ldots,D$ (number of domains) **do**
  for $p=1,\ldots,\mathscr{P}$ (number of terms) **do**
    Compute $\mathbf{y}=((\mathbf{k}^{a,d})^{-1}\mathbf{k}^{d})^{p}(\mathbf{k}^{a,d})^{-1}\mathbf{b}^{d}(f)$
    Compute $\mathbf{y}^{p}=((\mathbf{k}^{a,d})^{-1}\mathbf{k}^{d})^{p}(\mathbf{k}^{a,d})^{-1}\mathscr{W}_{h,d}$
  end for
end for
for $i=1,\ldots,\mathscr{I}$ (number of iterations) **do**
  for $d=1,\ldots,D$ (number of domains) **do**
    for $p=1,\ldots,\mathscr{P}$ (number of terms) **do**
      for $n=1,\ldots,\mathscr{N}$ (number of samples) **do**
        Compute $\mathbf{U}^{n,d}_{\mathscr{P},i}=\sum_{p=0}^{\mathscr{P}-1}(-A^{n,d})^{p}(\mathbf{y}^{p}\mathbf{b}^{n,d}(\mathscr{A}^{n},U^{n,d'}_{\mathscr{P},i-1})+\mathbf{y})$
      end for
    end for
  end for
end for

4. IMPLEMENTATION

As both the number of samples and the number of nodes in the discretization are likely to be large, storage and efficiency of computation are crucial issues.

As noted, the n-dependent part of $\mathbf{b}^{n,d}$ is nonzero only at boundary nodes and $\mathbf{b}^{n,d}$ is in the span of $\mathscr{W}_{h,d}$. In the first iterations, we can pre-compute

$$((\mathbf{k}^{a,d})^{-1}\mathbf{k}^{d})^{p}(\mathbf{k}^{a,d})^{-1}\mathscr{W}_{h,d}$$

efficiently, e.g. using PLU factorization. Note that this computation is independent of $\mathscr{N}$ and of $\mathscr{I}$ and must be performed only once for each sub-domain. However, it is crucial that the matrices $\mathbf{k}^{a,d}$ and $\mathbf{k}^{d}$ be kept fairly small.

During the time consuming inner loop over samples, we must compute

$$\mathbf{y}^{p}\mathbf{b}^{n,d}(\mathscr{A}^{n},U^{n,d'}_{\mathscr{P},i-1})$$

for each $p=0,\ldots,\mathscr{P}$, domain d, and sample n. The matrix $\mathbf{y}^{p}$ is an $\mathscr{L}^{d}\times\mathscr{J}^{d}$ matrix, where $\mathscr{L}^{d}$ is the number of nodes in the sub-domain and $\mathscr{J}^{d}$ is the number of nodes located on the faces of the sub-domain Ω_{d}. We also know that $\mathbf{b}^{d,n}(\mathscr{A}^{n},\mathbf{U}^{n,d}_{\mathscr{P},i-1})$ only has $\mathscr{J}^{d}$ nonzero entries. Furthermore, we only need to update $\mathbf{U}^{n,d}_{\mathscr{P},i}$ at the boundaries. This means that for each sample, we need to multiply a matrix of size $\mathscr{J}^{d}\times\mathscr{J}^{d}$ with a vector of length $\mathscr{J}^{d}$. Note that $\mathscr{J}^{d}\sim(\mathscr{L}^{d})^{(\dim-1)/\dim}$, dim is the space dimension. As there are $\mathscr{D}$ domains and we use $\mathscr{I}$ iterations, the total work required is of the order

$$\mathscr{N}\cdot\mathscr{I}\cdot\mathscr{D}\cdot(\mathscr{L}^{d})^{2(\dim-1)/\dim}$$

DOI: 10.1002/nme

As long as the size of the local problems are fairly small, this is a very fast way to compute solutions to the randomly perturbed Poisson equation.

Remark 4.1
It may seem surprising to claim that only boundary values are needed in the computations since $\mathbf{n}\cdot\nabla\mathbf{U}^{n,d}_{\mathscr{P},i}$ is present in the n-dependent part of the right-hand side. However, the normal derivative is never computed in practice. A trick with an initial guess at the boundary on a particular form that is updated through the computation eliminates the normal derivative from the algorithm. We guide the interested reader to [7] for a careful description of this idea.

Remark 4.2
The convergence rate of nonoverlapping domain decomposition algorithms suffers as the number of sub-domains increases. This can be addressing using a global coarse-scale correction. An altered approximation involves the sum of the solution from the previous iteration, a solution computed on a coarse grid, and a solution computed on a fine-scale using Algorithm 1.

5. *A POSTERIORI* ERROR ANALYSIS

We present two *a posteriori* results. The first yields an error estimate in the sample quantities of interest arising from numerical discretization. The second gives an error estimate for the error in the distribution function.

To obtain the estimate for each sample, we use the adjoint-based *a posteriori* approach [8, 9]. We express the error in the quantity of interest as a convolution of the computable residual of the finite element approximation and weights involving the generalized Green's function solving the adjoint problem corresponding to the quantity of interest. The adjoint weights scale the residuals to become errors, while the estimate takes into account cancellation, accumulation, and propagation of error throughout the domain. We start with the error in each sample linear functional value (U^n,ψ). We introduce a corresponding adjoint problem,

$$\begin{aligned} -\nabla\cdot\mathscr{A}\nabla\Phi &= \psi, \quad x\in\Omega \\ \Phi &= 0, \quad x\in\partial\Omega \end{aligned} \tag{6}$$

We compute $\mathscr{N}$ sample solutions $\{\Phi^n, n=1,\ldots,\mathscr{N}\}$ of (6) corresponding to the samples $\{\mathscr{A}^n, n=1,\ldots,\mathscr{N}\}$. To obtain estimates, we compute numerical solutions $\{\Phi^{n,d}_{\tilde{\mathscr{P}},\tilde{\mathscr{J}}}, d=1,\ldots,\mathscr{D}\}$ of (6) using Algorithm 1 with the more accurate space $\mathscr{V}_{\tilde{h}}$ and possibly different values $\tilde{\mathscr{P}}$ and $\tilde{\mathscr{J}}$. We denote the approximation on Ω by $\Phi^n_{\tilde{\mathscr{P}},\tilde{\mathscr{J}}}$.

The following result includes a remainder term that measures the effect of using a numerical solution of the adjoint problem.

Theorem 5.1
For each $n\in 1,\ldots,\mathscr{N}$,

$$|(U^n-U^n_{\mathscr{P},\mathscr{J}},\psi)|\lessapprox|(f,\Phi^n_{\tilde{\mathscr{P}},\tilde{\mathscr{J}}})-(\mathscr{A}^n\nabla U^n_{\mathscr{P},\mathscr{J}},\nabla\Phi^n_{\tilde{\mathscr{P}},\tilde{\mathscr{J}}})|+\mathscr{R}(\tilde{h},\tilde{\mathscr{P}},\tilde{\mathscr{J}}) \tag{7}$$

DOI: 10.1002/nme

where

$$\mathscr{R}(\tilde{h},\tilde{\mathscr{P}},\tilde{\mathscr{I}})=\begin{cases}\mathbf{O}(h^3), & \mathscr{V}_{\tilde{h}}=\mathscr{V}_h^2\\ \mathbf{O}(\tilde{h}^2), & \mathscr{T}_{\tilde{h}} \text{ is a refinement of } \mathscr{T}_h\end{cases}$$

Next, we present an *a posteriori* error analysis for the approximate cumulative distribution function obtained from $\mathscr{N}$ approximate sample values of a linear functional of a solution of a partial differential equation with random perturbations.

We let $U=U(X)$ be a solution of an elliptic problem that is randomly perturbed by a random variable X on a probability space $(\Omega,\mathscr{B},P)$ and $Q(U)=(U,\psi)$ be a quantity of interest, for some $\psi\in L^2(\Omega)$. We with to approximate the probability distribution function of $Q=Q(X)$,

$$F(t)=P(\{X:Q(U(X))\leqslant t\})=P(Q\leqslant t)$$

We use the sample distribution function computed from a finite collection of approximate sample values $\{\tilde{Q}^n,n=1,\ldots,\mathscr{N}\}=\{(\tilde{U}_n,\psi),n=1,\ldots,\mathscr{N}\}$,

$$\tilde{F}_{\mathscr{N}}(t)=\frac{1}{\mathscr{N}}\sum_{n=1}^{\mathscr{N}}I(\tilde{Q}^n\leqslant t)$$

where I is the indicator function. Here, $\tilde{U}^n$ is a numerical approximation for a true solution U^n corresponding to a sample X^n. We assume that there is an error estimate

$$\tilde{Q}^n-Q^n\approx\mathscr{E}^n$$

with $Q^n=(U^n,\psi)$, as provided by Theorem 5.1.

The two sources of error in the distribution function are finite sampling and the numerical approximation of the differential equation solutions. We define the sample distribution function

$$F_{\mathscr{N}}(t)=\frac{1}{\mathscr{N}}\sum_{n=1}^{\mathscr{N}}I(Q^n\leqslant t)$$

and decompose the error

$$|F(t)-\tilde{F}_{\mathscr{N}}(t)|\leqslant|F(t)-F_{\mathscr{N}}(t)|+|F_{\mathscr{N}}(t)-\tilde{F}_{\mathscr{N}}(t)|=I+II \tag{8}$$

There is an extensive statistics literature treating I, e.g. see [10]. We note that $F_{\mathscr{N}}$ has very desirable properties, e.g. it is itself distribution function, it is an unbiased estimator, $\mathscr{N}F_{\mathscr{N}}(t)$ has exact binomial distribution for $\mathscr{N}$ trials and probability of success $F(t)$, and $F_{\mathscr{N}}$ converges in several ways to F. As far as II, the effect of the error $\mathscr{E}^n$ is to introduce a random shift at each sample point. The resulting error can be estimated in terms of the continuity properties of F and $F_{\mathscr{N}}$ using the error estimate $\mathscr{E}^n$. We use I to denote the usual indicator function.

The following result presents two useful estimates, both of which have the flavor of bounding the error with high probability in the limit of large $\mathscr{N}$.

Theorem 5.2 (*Computable estimate*)
For any $\varepsilon>0$,

$$|F(t)-\tilde{F}_{\mathscr{N}}(t)|\leqslant\left(\frac{\tilde{F}_{\mathscr{N}}(t)(1-\tilde{F}_{\mathscr{N}}(t))}{\mathscr{N}\varepsilon}\right)^{1/2}+2\left|\frac{1}{\mathscr{N}}\sum_{n=1}^{\mathscr{N}}(I(\tilde{Q}^n-|\mathscr{E}^n|\leqslant t\leqslant\tilde{Q}^n+|\mathscr{E}^n|))\right|+\frac{1}{2\mathscr{N}\varepsilon} \tag{9}$$

with probability greater than $1-\varepsilon$.

DOI: 10.1002/nme

A posteriori bound: With L denoting the Lipschitz constant of F, for any $\varepsilon>0$,

$$|F(t)-\tilde{F}_{\mathscr{N}}(t)|\leqslant\left(\frac{F(t)(1-F(t))}{\mathscr{N}\varepsilon}\right)^{1/2}+L\max_{1\leqslant n\leqslant\mathscr{N}}\mathscr{E}^n+2\left(\frac{\log(\varepsilon^{-1})}{2\mathscr{N}}\right)^{1/2} \tag{10}$$

with probability greater than $1-\varepsilon$.

The leading order bounding terms in (9) are computable while the remainder tends to zero more rapidly in the limit of large $\mathscr{N}$. The bound (10) is useful for the design of adaptive algorithms. Assuming that the solutions of the elliptic problems are in $\mathscr{H}^2$, it indicates that the error in the computed distribution function is bounded by an expression whose leading order is proportional to

$$\frac{1}{\sqrt{\varepsilon\mathscr{N}}}+Lh^2$$

with probability $1-\varepsilon$. This suggests that in order to balance the error arising from finite sampling against the error in each computed sample, we typically should choose

$$\mathscr{N}\sim h^{-4}$$

This presents a compelling argument for seeking efficient ways to compute samples and control the accuracy.

6. ADAPTIVE ERROR CONTROL

Theorems 5.1 and 5.2 can be used as a basis for an adaptive error control algorithm. The computational parameters to be controlled are the mesh size h, number of terms in the truncated Neumann series $\mathscr{P}$, the number of iterations in the domain decomposition algorithm $\mathscr{I}$, and number of samples $\mathscr{N}$. To do this, we introduce $\Delta\mathscr{I}$ and $\Delta\mathscr{P}$ as positive integers and use the approximations

$$U^n_{\mathscr{P}+\Delta\mathscr{P},\mathscr{I}+\Delta\mathscr{I}}\approx U^n_{\infty,\infty}$$

where we use the obvious notation to denote the quantities obtained by taking $\mathscr{I},\mathscr{P}\to\infty$. The accuracy of the estimates improves as $\Delta\mathscr{I}$ and $\Delta\mathscr{P}$ increase.

To guide the selection of the parameters, we express the error $\mathscr{E}$ as a sum of three estimates corresponding, respectively, to discretization error, error from the incomplete Neumann series, and error from the incomplete domain decomposition iteration

$$\mathscr{E}^n=\mathscr{E}^n_{\mathrm{I}}+\mathscr{E}^n_{\mathrm{II}}+\mathscr{E}^n_{\mathrm{III}} \tag{11}$$

where

$$\mathscr{E}^n_{\mathrm{I}}\approx|(f,\Phi^n_{\tilde{\mathscr{P}},\tilde{\mathscr{I}}})-(\mathscr{A}^n\nabla U^n_{\mathscr{P},\mathscr{I}},\nabla\Phi^n_{\tilde{\mathscr{P}},\tilde{\mathscr{I}}})-(\mathscr{A}^n\nabla(U^n_{\mathscr{P}+\Delta\mathscr{P},\mathscr{I}+\Delta\mathscr{I}}-U^n_{\mathscr{P},\mathscr{I}}),\nabla\Phi^n_{\tilde{\mathscr{P}},\tilde{\mathscr{I}}})| \tag{12}$$

$$\mathscr{E}^n_{\mathrm{II}}\approx|(\mathscr{A}^n\nabla(U^n_{\mathscr{P}+\Delta\mathscr{P},\mathscr{I}}-U^n_{\mathscr{P},\mathscr{I}}),\nabla\Phi^n_{\tilde{\mathscr{P}},\tilde{\mathscr{I}}})| \tag{13}$$

$$\mathscr{E}^n_{\mathrm{III}}\approx|(\mathscr{A}^n\nabla(U^n_{\mathscr{P}+\Delta\mathscr{P},\mathscr{I}}-U^n_{h,\mathscr{P},\mathscr{I}}),\nabla\Phi^n_{\tilde{\mathscr{P}},\tilde{\mathscr{I}}})| \tag{14}$$

DOI: 10.1002/nme

We set

$$\mathscr{E} = \max_n \mathscr{E}^n, \quad \mathscr{E}_{\mathrm{I}} = \max_n \mathscr{E}^n_{\mathrm{I}}, \quad \mathscr{E}_{\mathrm{II}} = \max_n \mathscr{E}^n_{\mathrm{II}}, \quad \mathscr{E}_{\mathrm{III}} = \max_n \mathscr{E}^n_{\mathrm{III}}$$

We define in addition

$$\mathscr{E}_{\mathrm{IV}} = \left(\frac{\tilde{F}_{\mathscr{N}}(t)(1-\tilde{F}_{\mathscr{N}}(t))}{\mathscr{N}\varepsilon} \right)^{1/2} \tag{15}$$

for a given $\varepsilon > 0$.

We present an adaptive error control strategy based on these estimates in Algorithm 2.

6.1. A computational example

We present a realistic example using boundary conditions that arise frequently in simulations of oil reservoirs. We replace the Dirichlet boundary conditions in Equation (1) with Neumann conditions except on a small part of the boundary near the injection site, which is situated at the lower left corner of the domain,

$$\begin{aligned} -\nabla \cdot \mathscr{A}^n \nabla U^n &= f, \quad x \in \Omega \\ \mathscr{A}^n \partial_n U^n &= 0, \quad x \in \partial\Gamma_N \\ U^n &= 0, \quad x \in \partial\Gamma_D \end{aligned} \tag{16}$$

where $\Gamma_N \cup \Gamma_D = \partial\Omega$. In this case, a is the local permeability and U^n represents the pressure field. The error representation formula (7) and the adaptive algorithm are identical to the Dirichlet setting after defining the adjoint boundary conditions with the equivalent homogeneous Neumann/Dirichlet boundary conditions.

We choose a coefficient a that has a 'band' of low permeability around $x \approx 0.2$, which creates a large pressure drop parallel to the y-axis at this location. We let $f = 1$ in the lower left corner, the 'injector', and $f = -1$ in the upper right corner, the 'producer'. We use the average value of the solution as the quantity of interest, and set $\psi \equiv 1$.

We define a 27×7 decomposition of the square into equal sized sub-squares. On each sub-domain, we add a random perturbation to a, the magnitude of which is uniformly distributed in the interval determined by 20% of the magnitude of the local permeability. We illustrate in Figure 1.

We present a typical sample solution U^n in Figure 2.

In this problem, we fix the number of discretization nodes on each of the sub-domains to be 5×5 and let the adaptive algorithm choose the other parameters to match the accuracy given by this mesh. It is often the case in oil reservoir simulations, that the spatial mesh is fixed. The goal is again to approximate the distribution function $F(t)$ to within $\mathrm{TOL} = 0.15$ with 95% confidence. We let $\varepsilon = 0.05$, $\Delta\mathscr{I} = 0.3\mathscr{I}$, and $\Delta\mathscr{P} = 1$ in the adaptive algorithm. We start with 100 iterations in the domain decomposition algorithm, 1 term in the approximate inverse expansion, and 30 samples. We solve the adjoint problem using the same mesh as for the forward problem since we are not interested in mesh refinement. The number of iterations, terms, and samples is the same for the forward and the adjoint problems.

DOI: 10.1002/nme

Algorithm 2 Adaptive algorithm

Choose ε in Theorem 5.2, which determines the reliability of the error control
Let TOL be the desired tolerance of the error $|F(t)-\tilde{F}_{\mathscr{N}}(t)|$
Let $\sigma_1+\sigma_2+\sigma_3+\sigma_4=1$ be positive numbers that are used to apportion the tolerance TOL between the four contributions to the error, with values chosen based on the computational cost associated with changing the four discretization parameters
Choose initial meshes $\mathscr{T}_h$, $\mathscr{T}_{\tilde{h}}$ and $\mathscr{P}$, $\mathscr{I}$, and $\mathscr{N}$

Compute $\{U^n_{\mathscr{P},\mathscr{I}}, n=1,\cdots,\mathscr{N}\}$ in the space $\mathscr{V}_h$ and the sample quantity of interest values
Compute $\{\Phi^n_{\tilde{\mathscr{P}},\tilde{\mathscr{I}}}, n=1,\cdots,\mathscr{N}\}$ in the space $\mathscr{V}_{\tilde{h}}$
Compute $\mathscr{E}^n_{\mathscr{I}}, \mathscr{E}^n_{\mathscr{I}\mathscr{I}}, \mathscr{E}^n_{\mathscr{I}\mathscr{I}\mathscr{I}}$ for $n=1,\cdots,\mathscr{N}$

Compute $\tilde{F}_{\mathscr{N}}(t)$
Estimate the Lipschitz constant L of F using $\tilde{F}_{\mathscr{N}}$
Compute $\mathscr{E}_{\mathscr{I}\mathscr{V}}$

while $\mathscr{E}_{\mathscr{I}\mathscr{V}}+\left|\frac{1}{\mathscr{N}}\sum_{n=1}^{\mathscr{N}}(I(\tilde{Q}^n-|\mathscr{E}^n|\leqslant t\leqslant\tilde{Q}^n+|\mathscr{E}^n|))\right|\geqslant$TOL **do**

if $L\mathscr{E}_{\mathscr{I}}>\sigma_1$TOL **then**
Refine $\mathscr{T}_h$ and $\mathscr{T}_{\tilde{h}}$ to meet the prediction that $L\mathscr{E}_{\mathscr{I}}\approx\sigma_1$TOL on the new mesh
end if
if $L\mathscr{E}_{\mathscr{I}\mathscr{I}}>\sigma_2$TOL **then**
Increase $\mathscr{P}$ to meet the prediction $\mathscr{E}_{\mathscr{I}\mathscr{I}}\approx\sigma_2$TOL
end if
if $L\mathscr{E}_{\mathscr{I}\mathscr{I}\mathscr{I}}>\sigma_3$TOL **then**
Increase $\mathscr{P}$ to meet the prediction $\mathscr{E}_{\mathscr{I}\mathscr{I}\mathscr{I}}\approx\sigma_3$TOL
end if
if $\mathscr{E}_{\mathscr{I}\mathscr{V}}>\sigma_4$TOL **then**
Increase $\mathscr{N}$ to meet the prediction $\mathscr{E}_{\mathscr{I}\mathscr{V}}\approx\sigma_4$TOL
end if

Compute $\{U^n_{\mathscr{P},\mathscr{I}}, n=1,\cdots,\mathscr{N}\}$ in the space $\mathscr{V}_h$ and the sample quantity of interest values
Compute $\{\Phi^n_{\tilde{\mathscr{P}},\tilde{\mathscr{I}}}, n=1,\ldots,\mathscr{N}\}$ in the space $\mathscr{V}_{\tilde{h}}$
Compute $\mathscr{E}^n_{\mathscr{I}}, \mathscr{E}^n_{\mathscr{I}\mathscr{I}}, \mathscr{E}^n_{\mathscr{I}\mathscr{I}\mathscr{I}}$ for $n=1,\cdots,\mathscr{N}$

Compute $\tilde{F}_{\mathscr{N}}(t)$
Estimate the Lipschitz constant L of F using $\tilde{F}_{\mathscr{N}}$
Compute $\mathscr{E}_{\mathscr{I}\mathscr{V}}$

end while

In Figure 3, we plot the parameter values during the four iterations. The error tolerance is achieved when $\mathscr{I}=800$, $\mathscr{P}=4$, and $\mathscr{N}=240$.

In Figure 4, we plot error estimators after each iteration in the adaptive algorithm along with the total estimate.

DOI: 10.1002/nme

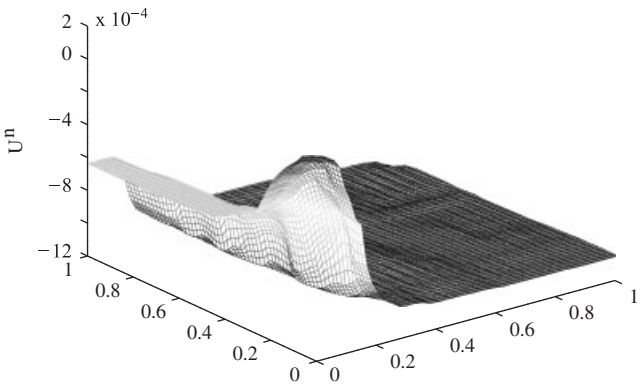

Figure 2. A typical sample solution.

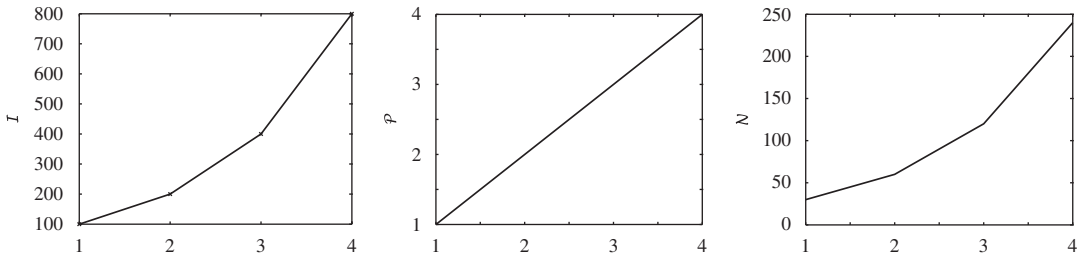

Figure 3. The parameter values chosen by four iterations of the adaptive algorithm. Left: $\mathscr{I}$; middle: $\mathscr{P}$; and right: $\mathscr{N}$.

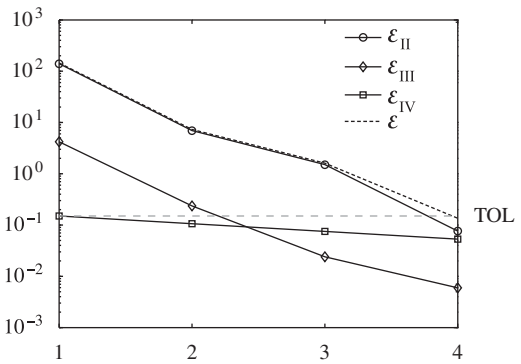

Figure 4. The error estimates computed after each iteration in the adaptive algorithm.

In Figure 5, we plot the approximation to $F(t)$ after each iteration. We see that the distribution function converges as the number of iterations increases, and there is a large error in the distribution function for the first two iterations.

DOI: 10.1002/nme

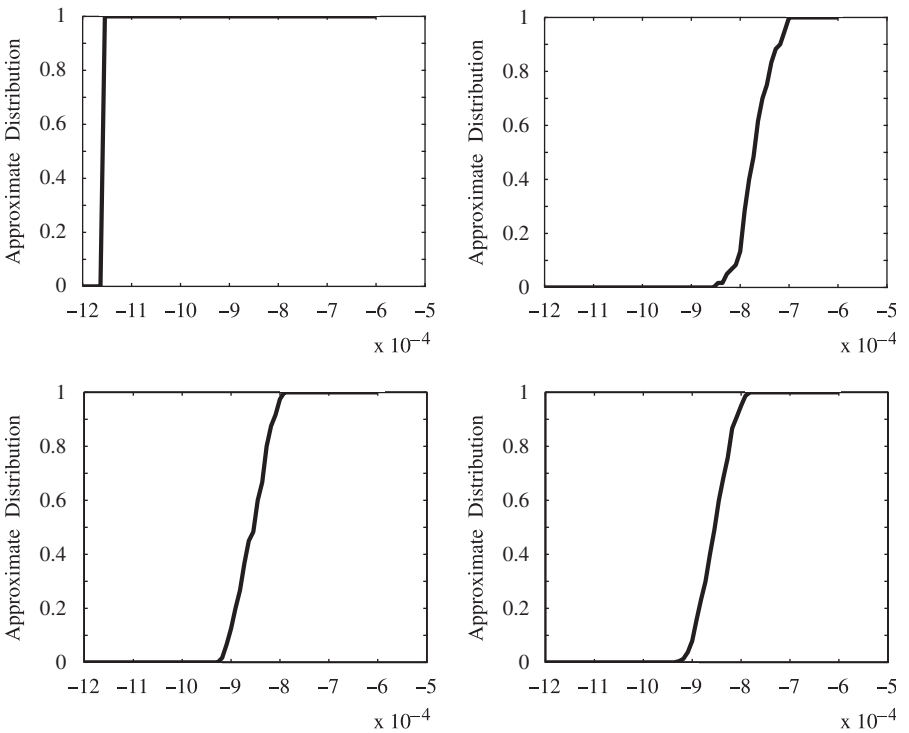

Figure 5. The approximate distribution functions for each of the four iterations in the adaptive algorithm, from left to right, top to bottom.

7. ADAPTIVE MODELING

It is common to find meshes with large variations in element sizes through the domain when using adaptive error control based on goal-oriented error estimates. By extension, it is natural to conjecture that the degree to which a model must be specified may also have to vary through the domain. We show how the adaptive algorithm can be modified to estimate and control model uncertainty.

We assume that $\mathscr{A}^n$ is an approximation to an exact random perturbation $\bar{\mathscr{A}}^n$ that is used in a computation. The *a posteriori* estimate (7) becomes

$$|(U^n - U^n_{\mathscr{P}}, \psi)| \leqslant |(f, \Phi^n_{\tilde{h}}) - (\mathscr{A}^n \nabla U^n_{\mathscr{P}}, \nabla \Phi^n_{\tilde{h}})| + |((\bar{\mathscr{A}}^n - \mathscr{A}^n)\nabla U^n_{\mathscr{P}}, \nabla \Phi^n_{\tilde{h}})| + \mathscr{R}(\tilde{h}, \tilde{\mathscr{P}}, \tilde{\mathscr{I}}) \tag{17}$$

This observation gives a new term in the error representation formula, which we approximate in the following way,

$$\mathscr{E}_{\mathscr{A}} = |((\mathscr{A}^n - \bar{\mathscr{A}}^n)\nabla U^n_{h,\mathscr{P},\mathscr{I}}, \nabla \Phi^n)| \tag{18}$$

We modify the adaptive algorithm by choosing five apportioning constants $\sigma_1 + \cdots + \sigma_5 = 1$ and inserting an extra step where $L\mathscr{E}_{\mathscr{A}} \leqslant \sigma_5 \mathrm{TOL}$ is checked. If this condition is violated, then some

DOI: 10.1002/nme

subset of the domain elements Ω_d are marked for refinement, requiring the collection of additional sample values $\mathscr{A}^{n,d}$ for the next adaptive iteration.

7.1. Computational examples

We present three examples that illustrate the consequences of including a modeling error term in the adaptive algorithm.

7.1.1. A varying diffusion coefficient affected by a uniformly sized relative error. We solve (1) with $f \equiv 1$ and the diffusion coefficient a perturbed by the random perturbation uniformly distributed on an interval determined by 10% of the magnitude of the diffusion coefficient on each sub-domain. We show a typical realization of the diffusion coefficient in Figure 6.

The goal is to compute an accurate approximation of the solution in the upper right corner of the domain by setting $\psi = 1$ when $0.9 \leqslant x, y \leqslant 1$ and $\psi = 0$ otherwise. We require the error in the approximation of F to be smaller than 10% with 95% probability; hence, TOL $= 0.1$ and $\varepsilon = 0.05$. Initially, we choose $h = \frac{1}{32}$, $\mathscr{I} = 40$, $\mathscr{P} = 1$, and $\mathscr{N} = 40$. We compute the adjoint solution using a finer mesh and the same number of iterations, terms, samples, and the same random perturbation as the forward problem.

We follow the adaptive algorithm, Algorithm 2, modified to include the new error indicator $\mathscr{E}_{\mathscr{V}}$. We use $\sigma_2 = \sigma_3 = \sigma_5 = \frac{1}{6}$, $\sigma_4 = \frac{1}{2}$. We do not adapt the spatial mesh. For refining the description of the random perturbation, we proceed as follows. We start with a coarse 2×2 subdivision of the unit square on which A^n is described. For a given subdivision associated with $\mathscr{A}^{n,d}$, we refine each sub-domain into 2×2 children and these children further into 2×2 children. The random perturbation on any sub-domain or its children is chosen by the same rule, i.e. distributed uniformly on the interval determined by 10% of the value of a on that sub-domain. We now estimate $\mathscr{A}^{n,d} - \bar{\mathscr{A}}^{n,d}$ by approximating $\bar{\mathscr{A}}^{n,d}$ on the finest subdivision corresponding to the current subdivision for $\mathscr{A}^{n,d}$.

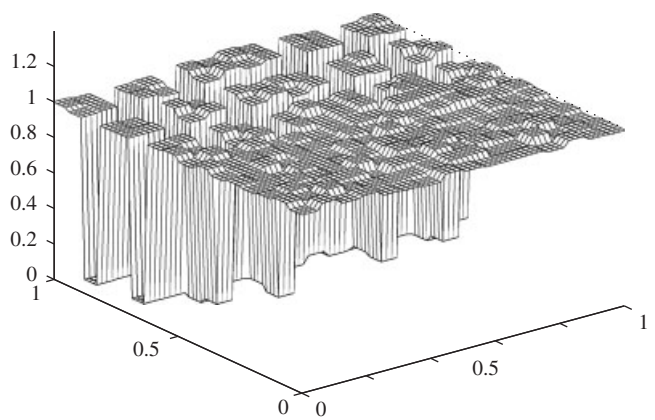

Figure 6. The diffusion coefficient with a piecewise constant perturbation. The random perturbation is uniformly distributed on an interval determined by 10% of the magnitude of the diffusion coefficient on each sub-domain.

DOI: 10.1002/nme

In Figure 7, we present the first three parameter values for each of the iterates. The tolerance was reached after three iterations with $\mathscr{I}=160$, $\mathscr{P}=4$, and $\mathscr{N}=320$.

In Figure 8, we plot the sequence of sub-domains chosen by the adaptive modeling component of the algorithm. Not surprisingly, the algorithm calls for a more detailed description of the random uncertainty in the spatial region where the quantity of interest is concentrated. In Figure 9, we plot the error estimators for each iteration in the adaptive algorithm along with the total error estimate. In Figure 10, we plot the approximate distribution functions. We use the distribution computed during the last iteration as a reference solution and compute approximate errors, which are plotted in Figure 11.

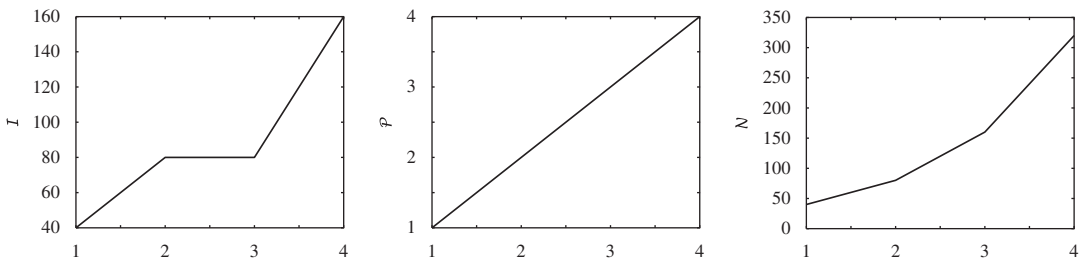

Figure 7. The method parameters chosen by the adaptive algorithm. Right: $\mathscr{I}$; middle: $\mathscr{P}$; and left: $\mathscr{N}$.

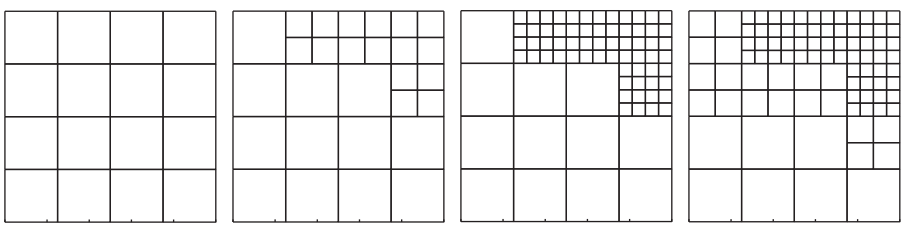

Figure 8. The sequence of sub-domains chosen by the adaptive modeling component of the algorithm.

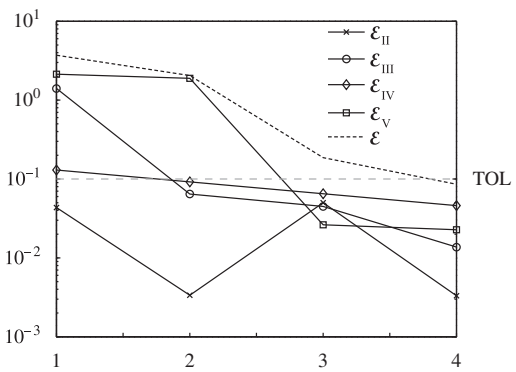

Figure 9. The error estimators for each iteration in the adaptive algorithm.

DOI: 10.1002/nme

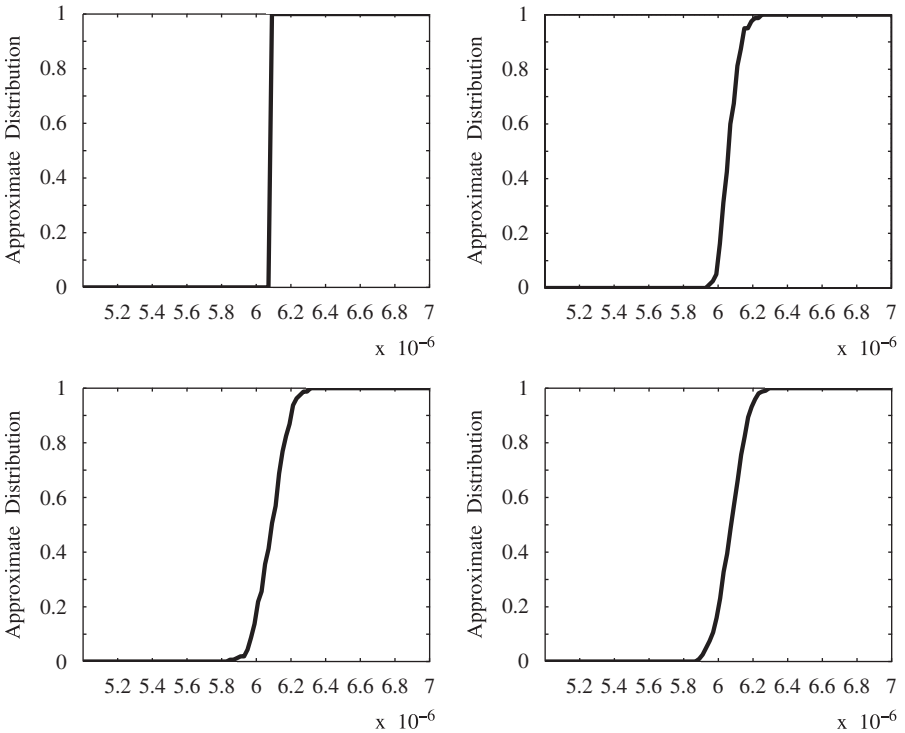

Figure 10. The approximate distribution functions computed by the adaptive algorithm.

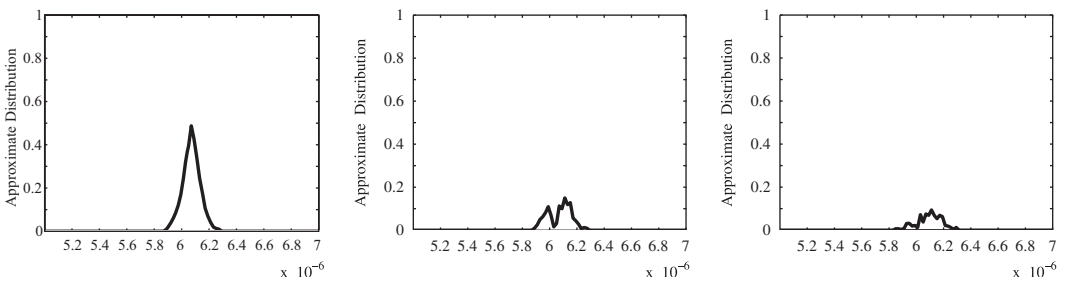

Figure 11. Approximate errors in the distributions from the first three iterations.

7.1.2. A constant diffusion coefficient affected by an error that varies throughout the domain. In this example, we let $a=1$ and $f=1$ and consider a random perturbation that has different sizes in different regions of the domain, more precisely it is uniformly distributed in an interval of size 10% for $0\leqslant x<0.75$ and 50% for $0.75\leqslant x\leqslant 1$. We plot a typical realization of $\mathscr{A}^s$ in Figure 12.

The goal is to compute an accurate solution in the upper right corner of the domain, hence, we set $\psi=1$ for $0.9\leqslant x, y\leqslant 1$ and zero otherwise. We require the error to be less than 0.1 with 95% probability so TOL$=0.1$ and $\varepsilon=0.05$. Initially, we choose $h=\frac{1}{32}$, $\mathscr{I}=40$, $\mathscr{P}=1$, and $\mathscr{N}=40$. We

DOI: 10.1002/nme

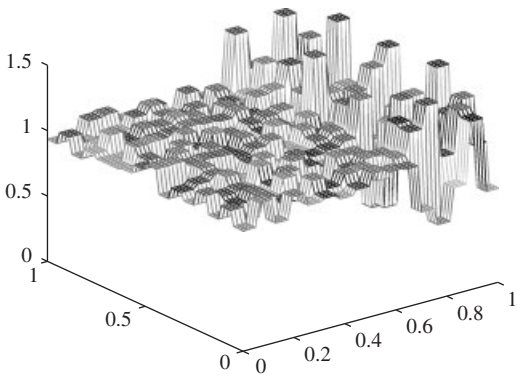

Figure 12. The diffusion coefficient $a=1$ is perturbed by a random error uniformly distributed in an interval of size 10% for $0\leqslant x<0.75$ and 50% for $0.75\leqslant x\leqslant 1$.

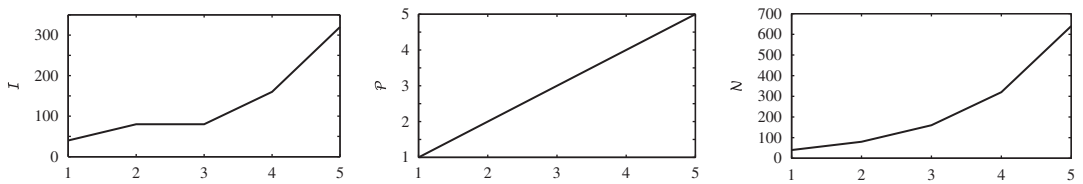

Figure 13. The method parameters chosen by the adaptive algorithm. Right: $\mathscr{I}$; middle: $\mathscr{P}$; and left: $\mathscr{N}$.

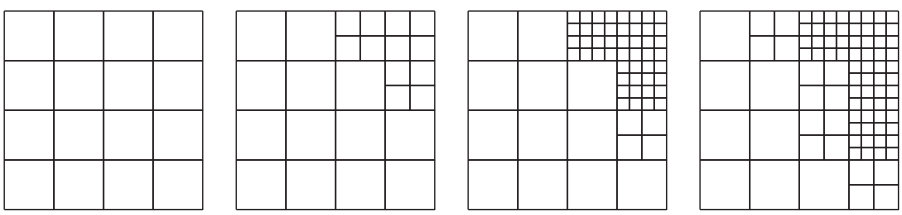

Figure 14. The first four sub-domains in the sequence chosen to control the modeling error. The sequence remains fixed after the third iteration.

compute the adjoint solution using a finer mesh and the same number of iterations, terms, samples, and the same random perturbation as the forward problem. We follow the adaptive algorithm, Algorithm 2, modified to include the new error indicator $\mathscr{E}_{\mathscr{V}}$. We use $\sigma_2=\sigma_3=\sigma_5=\frac{1}{6}$ and $\sigma_4=\frac{1}{2}$. We show the sequence of discretization parameters chosen by the algorithm in Figure 13. The tolerance is reached when $\mathscr{I}=320$, $\mathscr{P}=5$, and $\mathscr{N}=640$.

In Figure 14, we show the sequence of sub-domains chosen to control the modeling error. We observe that the algorithm uses a finer-scale representation of the random perturbation in the vicinity of the region of interest, where $\psi=1$, and the region in the domain where the perturbations are largest.

DOI: 10.1002/nme

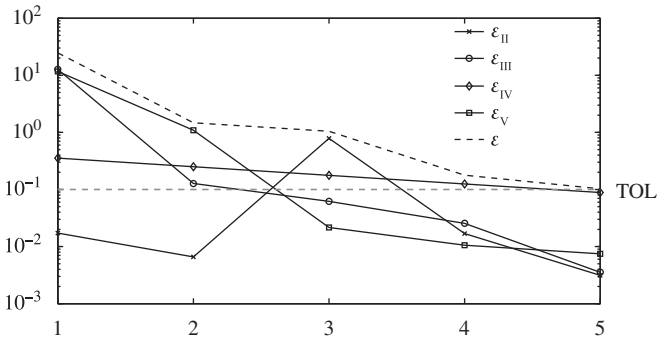

Figure 15. The error estimators for each iteration in the adaptive algorithm.

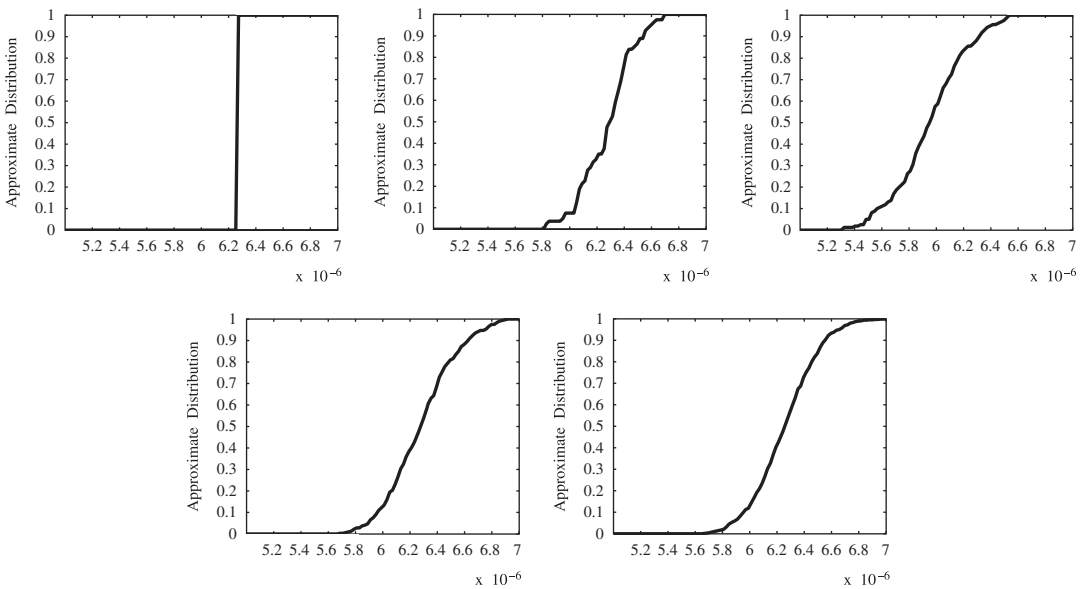

Figure 16. The approximate distribution functions computed by the adaptive algorithm.

In Figure 15 we plot the error indicators after each iteration. We see that the iteration error $\mathscr{E}_{\mathrm{II}}$ increases when the size of the smallest sub-domain is reduced between iterations 2 and 3. We also note that the truncation error term $\mathscr{E}_{\mathrm{III}}$ is larger now since the worst perturbation is larger, recalling that the truncation error is proportional to the size of the perturbation divided by the size of a to the power of $\mathscr{P}$.

In Figure 16, we plot the approximate distribution functions.

7.1.3. A problem with a sharply discontinuous diffusion coefficient and random perturbation. In this example, we consider a diffusion coefficient a with a sharp discontinuity between the values

DOI: 10.1002/nme

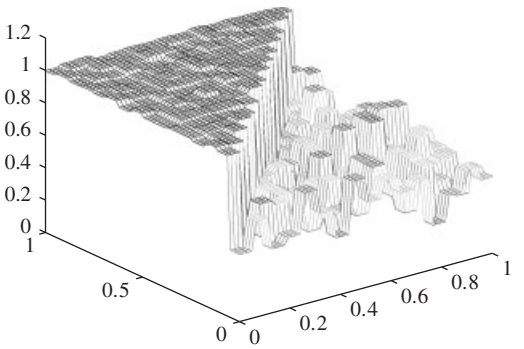

Figure 17. We let $a=1$ for $x+y>1$ and $a=0.5$ for $x+y\leqslant 1$. The random perturbation has normal distribution with mean zero. The standard deviation is 0.01 in the region where $a=1$ and 0.1 in the region where $a=0.5$.

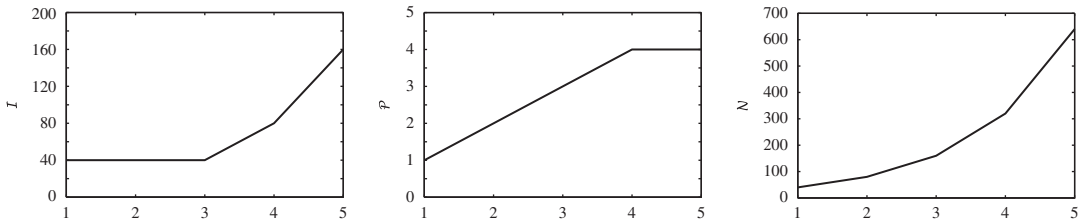

Figure 18. The method parameters chosen by the adaptive algorithm. Right: $\mathscr{I}$; middle: $\mathscr{P}$; and left: $\mathscr{N}$.

$a=1$ for $x+y>1$ and $a=0.5$ for $x+y\leqslant 1$. In the region where $a=0.5$, we impose a perturbation that has a truncated normal distribution with mean zero and standard deviation 0.1, while in the remaining region, we impose a perturbation with a truncated normal distribution with mean zero and smaller standard deviation 0.01. In both cases, we truncate the distribution so that $\mathscr{A}^s \geqslant 0.1$. We illustrate this by plotting a typical realization of $\mathscr{A}^s$ in Figure 17.

We use $f=1$, and set the adjoint data ψ equal to one in the lower right corner $0.9\leqslant x\leqslant 1$ and $0\leqslant y\leqslant 0.1$ and zero otherwise. We require that the error to be less than 0.1 with 95% probability so TOL $=0.1$ and $\varepsilon=0.05$. We compute the adjoint solution using a finer mesh and the same number of iterations, terms, samples, and the same random perturbation as the forward problem. We follow the adaptive algorithm, Algorithm 2, modified to include the new error indicator $\mathscr{E}_{\mathscr{V}}$. We use $\sigma_2=\sigma_3=\sigma_5=\frac{1}{6}$ and $\sigma_4=\frac{1}{2}$. We show the sequence of discretization parameters chosen by the algorithm in Figure 18. The tolerance is reached when $\mathscr{I}=160$, $\mathscr{P}=4$, and $\mathscr{N}=640$.

In Figure 19, we show the sequence of sub-domains chosen to control the modeling error. The algorithm uses a finer representation of the random perturbation where $\psi=1$ and below the diagonal where the perturbations are large.

In Figure 20 we plot the error indicators after each iteration. The iteration error $\mathscr{E}_{\mathscr{I}\mathscr{I}}$ increases when the size of the smallest sub-domain is reduced between iterations 2 and 3.

In Figure 21, we plot the approximate distribution functions.

DOI: 10.1002/nme

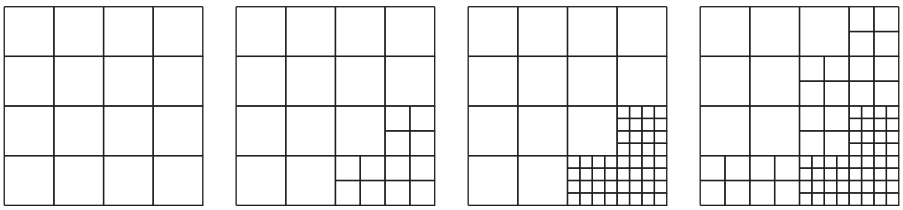

Figure 19. The first four sub-domains in the sequence chosen to control the modeling error. The sequence remains fixed after the third iteration.

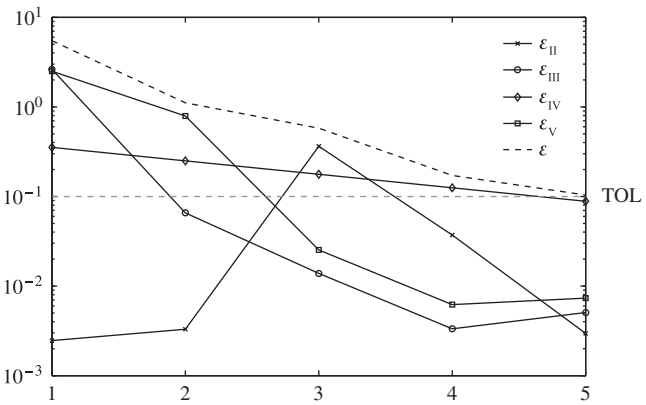

Figure 20. The error estimators for each iteration in the adaptive algorithm.

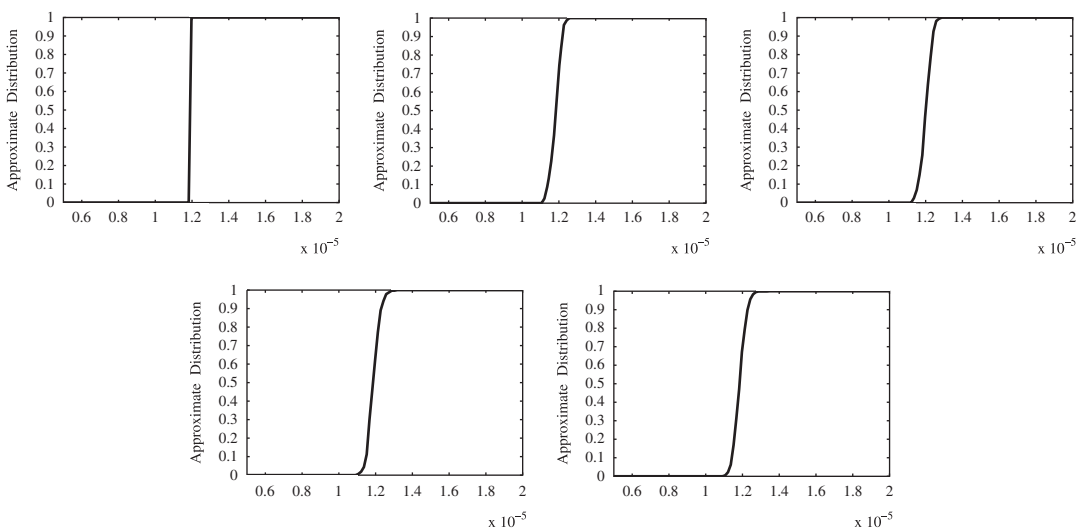

Figure 21. The approximate distribution functions computed by the adaptive algorithm.

DOI: 10.1002/nme

8. CONCLUSION

In this paper, we consider propagation of uncertainty into a quantity of interest computed from numerical solutions of an elliptic partial differential equation with a randomly perturbed diffusion coefficient. Our particular interest are problems for which limited knowledge of the random perturbations are known and for which we desire an approximation of the probability distribution of the output. We employ a nonparametric density estimation approach based on a very efficient method for computing random samples of elliptic problems [4]. We fully develop the adaptive algorithm suggested by the analysis in that paper and discuss details of its implementation. Then, we illustrate the adaptive algorithm using a realistic data set. Finally, we explore the modeling assumption underlying the basis for this new approach. In particular, we show that the analysis can be extended to include a 'modeling error' term that accounts for the effects of varying the resolution of the statistical description of the random variation. We then describe a modification to the adaptive algorithm that provides a mechanism for adapting the resolution of the statistical description in order to compute a desired quantity of interest to a specified accuracy. We illustrate the behavior of the adaptive algorithm in several examples.

REFERENCES

1. Ghanem R, Spanos P. *Stochastic Finite Elements*: *A Spectral Approach*. Dover: New York, 2003.
2. Babuška I, Tempone R, Zouraris GE. Solving elliptic boundary value problems with uncertain coefficients by the finite element method: the stochastic formulation. *Computer Methods in Applied Mechanics and Engineering* 2005; **194**:1251–1294.
3. Farmer CL. Upscaling: a review. *International Journal for Numerical Methods in Fluids* 2002; **40**:63–78. *ICFD Conference on Numerical Methods for Fluid Dynamics*, Oxford, 2001.
4. Estep D, Målqvist A, Tavener S. Nonparametric density estimation for randomly perturbed elliptic problems I: computational methods, a posteriori analysis, and adaptive error control. *SIAM Journal on Scientific Computing* 2008. DOI: JCOMP-D-08-00261.
5. Estep D, Holst MJ, Målqvist A. Nonparametric density estimation for randomly perturbed elliptic problems III: convergence and a priori analysis. 2008. DOI: JCOMP-D-08-00261.
6. Lions P-L. On the Schwarz alternating method. III. A variant for nonoverlapping subdomains. *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, Houston, TX, 1989. SIAM: Philadelphia, PA, 1990; 202–223.
7. Guo W, Hou LS. Generalizations and accelerations of Lions' nonoverlapping domain decomposition method for linear elliptic PDE. *SIAM Journal on Numerical Analysis* 2003; **41**:2056–2080 (electronic).
8. Estep D, Larson MG, Williams RD. *Estimating the Error of Numerical Solutions of Systems of Reaction–diffusion Equations*. Memoirs of the American Mathematical Society, vol. 146. American Mathematical Society: Providence, RI, 2000; viii+109.
9. Bangerth W, Rannacher R. *Adaptive Finite Element Methods for Differential Equations*. Lectures in Mathematics ETH Zürich. Birkhäuser Verlag: Basel, 2003.
10. Serfling R. *Approximation Theorems of Mathematical Statistics*. Wiley: New York, 1980.

DOI: 10.1002/nme