Efficient approximation, error estimation, and adaptive computation for randomly perturbed elliptic problems

Donald Estep

University Interdisciplinary Research Scholar
Department of Statistics
Department of Mathematics
Colorado State University

Research supported by
Collaborators

Axel Målqvist
University of Uppsala

Simon Tavener
Colorado State University
The Problem
A Randomly Perturbed Elliptic Problem

Compute

Quantity of Interest: $Q(U)$

where $U$ solves (a.s.)

$$\begin{cases} -\nabla \cdot A \nabla U = f, & x \in \Omega, \\ U = 0, & x \text{ in } \partial \Omega, \end{cases}$$

- $f \in L^2(\Omega)$
- $\Omega$ is a convex polygonal domain with boundary $\partial \Omega$
- $A$ is a piecewise smooth, coercive, bounded, stochastic function with continuous and bounded covariance functions
- $Q(U)$ is given as a linear functional, e.g. average value over $\Omega$ or value at a point in $\Omega$
The output quantity $Q(U)$ is a random variable

The problem:

*Given the probability distribution describing $A$, compute the probability distribution of $Q(U)$*

**Nonparametric solution:** Compute an approximate distribution for $Q(U)$ using a finite sample of values

This is appropriate when the output distribution is unknown or complicated
The formal Monte Carlo solution algorithm is

1: Draw samples \( \{ A^n \}_{n=1}^N \) from the distribution of \( A \)

2: \textbf{for} \( n = 1, \cdots, N \) (number of samples) \textbf{do}

3: \hspace{1em} Compute solutions \( \{ U^n \} \) to produce samples \( \{ Q(U^n) \} \)

4: \textbf{end for}

5: Approximate the output distribution using a standard nonparametric technique, e.g. via binning

This algorithm is characterized by putting the statistics computation “in the outside loop”
This approach is affected by several issues:

- Limited knowledge about the stochastic nature of $A$
- Sample solutions are computationally expensive
- Using a finite number of sample numerical solutions introduces various errors that interact
- For uncertainty quantification, we require an estimate of the error in the computed distribution
- The smooth behavior of the solution operator carries information about sensitivity of the solution
Overview of this Work

Results in this work include:

- An efficient algorithm that reduces total computational work
- An \textit{a posteriori} error estimate for the approximate distribution that accounts for both stochastic and deterministic errors
- An efficient adaptive error control strategy that balances all sources of error
- Derivation of a modeling error expression
- Convergence analysis
A Model for the Stochastic Perturbation
A Modeling Assumption

The stochastic diffusion coefficient can be written

\[ A = a + A, \]

- \( a \) is a uniformly coercive, bounded, piecewise smooth function
- \( A \) is the random perturbation and \( |A(x)| < a(x) \)
- \[ A(x) = \sum_{\kappa \in \mathcal{K}} A^\kappa \chi_{\kappa}(x), \quad x \in \Omega, \]

- \( \mathcal{K} \) is a finite nonoverlapping polygonal partition of \( \Omega \)
- \( \chi_{\kappa} \) is the characteristic function for the set \( \mathcal{K} \)
- \( (A^\kappa) \) is a random vector
Geometry of the Random Perturbation

\[ A = A^K \chi_K(x) \]
$a$ is $0.1$ on a “cross” centered at the origin and $1$ elsewhere.
A introduces random perturbations on a $9 \times 9$ grid
Interpretations:

- $A$ is a stochastic perturbation to a known deterministic function $a$.
- $A$ can be measured at a finite set of points, $a$ is the expected value $E_A$ and $A$ is the variation around $a$.
- $A$ is a multiscale expression, where $a$ is the engineering scale value and $A$ is the (stochastic) microscale perturbation.
The value of $A$ at a quadrature point in a cell or element is determined by integrating the solution of a (stochastic) microscale model over representative volume element.
A Modeling Assumption

In all cases, we can measure values of $A$ at a finite set of points.

We choose to represent $A$ as a piecewise constant function.

Improving this model involves choosing a finer partition and measuring $A$ at more points.

Improving the accuracy of a given model involves acquiring more samples of $\{A^K\}$.

We can also represent $A$ using piecewise polynomials of higher order, including continuous functions.
The Numerical Method
Observation and Question

The deterministic part $a$ of the elliptic coefficient does not vary with the samples

The random perturbations are localized to subregions of the domain and are constant on those subregions

Can we somehow exploit this localized nature of the stochastic perturbations?

Note that properties of elliptic problems means that the effects are not completely local
Non-overlapping Domain Decomposition

Replace the problem by a system of local problems coupled by continuity of state and normal flux across interior boundaries.

![Diagram showing a 2x2 grid of local elliptic problems with state flux between them.]

Donald Estep: University of Chicago, 11/15/2011
We denote the subdomains of $\mathcal{K}$ by $\Omega_d$, $d = 1, \cdots, D$

$d'$ denotes the set of indices in $\{1, 2, \cdots, D\} \setminus \{d\}$ where $\Omega_{d'}$ intersects at a boundary with $\Omega_d$

$A^n = a + A^n$ is a particular sample
Non-overlapping Domain Decomposition

Replace the original problem by a system of $\mathcal{D}$ local problems

\[
\begin{cases}
-\nabla \cdot A^\lambda \nabla U^{n,d} = f, & x \in \Omega_d, \\
U^{n,d} = 0, & x \in \partial \Omega_d \cap \partial \Omega, \\
\frac{1}{\lambda} U^{n,d} + n_{\tilde{d}} \cdot A^\lambda \nabla U^{n,d} = \frac{1}{\lambda} U^{n,\tilde{d}} - n_{\tilde{d}} \cdot A^\lambda \nabla U^{n,\tilde{d}}, & x \in \partial \Omega_d \cap \partial \Omega_{\tilde{d}}, \quad \tilde{d} \in d',
\end{cases}
\]

$\lambda$ is a parameter and $n_{\tilde{d}}$ denotes a unit normal vector

On interior boundaries, we are prescribing a kind of continuity of state and normal flux

Each subdomain is provided with sufficient boundary conditions
Iterative Solution of the System

The resulting system is solved iteratively

Choose initial guesses \( \{ U_{0,1}^n, \ldots, U_{0,D}^n \} \)

For \( 1 \leq i \leq I \), solve local elliptic problems

\[
\begin{align*}
-\nabla \cdot A^n \nabla U_{i,d}^n &= f, & x \in \Omega_d, \\
U_{i,d}^n &= 0, & x \in \partial \Omega_d \cap \partial \Omega, \\
\frac{1}{\lambda} U_{i,d}^n + n_{\tilde{d}} \cdot A^n \nabla U_{i,d}^n &= \frac{1}{\lambda} U_{i-1,d}^{n,\tilde{d}} - n_{\tilde{d}} \cdot A^n \nabla U_{i-1,d}^{n,\tilde{d}}, & x \in \partial \Omega_d \cap \partial \Omega_{\tilde{d}}, \\
\tilde{d} &\in d', \\
d &= 1, \ldots, D
\end{align*}
\]

The boundary data is “lagged” from the previous iterate
Finite Element Discretization

We **subdivide** the model mesh $\mathcal{K}$ to create a finite element mesh.

Mesh for $\mathcal{K}$
The finite element approximation is a piecewise polynomial function on the finite element mesh.

It satisfies an “average” formulation of the differential equation.

The average, or variational formulation, of the problem is obtained by multiplying the equation by a test function and integrating over the spatial domain.

Integration by parts is used to move derivatives onto the test function, and we have to choose appropriate function spaces.
Finite Element Discretization

\( V_{h,d} \subset H^1_{0,\partial \Omega}(\Omega_d) \) is a finite element approximation space corresponding to a mesh \( \mathcal{T}_d \) on \( \Omega_d \)

The finite element method: \( U_i^{n,d} \in V_{h,d}, \ d = 1, \cdots, D, \) solves

\[
(A^n \nabla U_i^{n,d}, \nabla v)_d + \frac{1}{\lambda} \langle U_i^{n,d}, v \rangle_{d \cap \tilde{d}} = (f, v)_d + \sum_{\tilde{d} \in d'} \left( \frac{1}{\lambda} \langle U_i^{n,\tilde{d}}, v \rangle_{d \cap \tilde{d}} - \langle \mathbf{n}_{\tilde{d}} \cdot A^n \nabla U_i^{n,\tilde{d}}, v \rangle_{d \cap \tilde{d}} \right),
\]

all \( v \in V_{h,d} \)

\((\ , \ , \ )_d\) denotes the \( L^2 \) inner product over \( d \) subdomain

\(\langle \ , \ , \ \rangle_{d \cap \tilde{d}}\) denotes the \( L^2 \) inner product over the indicated boundary
We choose a basis for the piecewise polynomial space for the approximation and test functions.

We substitute a basis expansion for the finite element solution into the equation and cycle the test functions through the variational formulation.

This results in a matrix-vector system for the coefficients of the finite element solution.
\( \vec{U}_{i}^{n,d} \) is the vector of basis coefficients of \( \vec{U}_{i}^{n,d} \)

\[
(\mathbf{k}^{a,d} + \mathbf{k}^{n,d}) \vec{U}_{i}^{n,d} = \vec{b}^{d}(f) + \vec{b}^{n,d}(A^{n}, U_{i-1}^{n,d'})
\]

\( (\mathbf{k}^{a,d})_{lk} = (a \nabla \varphi_{d,l}, \nabla \varphi_{d,k})_{d} \), deterministic stiffness matrix

\( (\mathbf{k}^{n,d})_{lk} = (A^{n,d} \nabla \varphi_{d,l}, \nabla \varphi_{d,k})_{d} \), stochastic stiffness matrix

\( (\vec{b}^{d})_{k} = (f, \varphi_{d,k})_{d} \),

\( (\vec{b}^{n,d})_{k} = \sum_{\tilde{d} \in d'} \left( \frac{1}{\lambda} \langle U_{i-1}^{n,\tilde{d}} \varphi_{d,k} \rangle_{d \cap \tilde{d}} - \langle n_{\tilde{d}} \cdot A^{n} \nabla U_{i-1}^{n,\tilde{d}} \varphi_{d,k} \rangle_{d \cap \tilde{d}} \right) \),

for \( 1 \leq l, k \leq M \), where \( \{ \varphi_{d,m} \}_{m=1}^{n_d} \) is a basis for \( V_{h,d} \)
A Monte-Carlo Algorithm for the Approximation

1: Draw samples \( \{ A^n \}_{n=1}^N \) from the distribution of \( A \)

2: for \( n = 1, \cdots, N \) (number of samples) do

3: for \( i = 1, \cdots, I \) (iterations in domain decomposition) do

4: for \( d = 1, \cdots, D \) (domains in domain decomposition) do

5: Solve \((k^a,d + k^n,d)\vec{U}_{i}^{n,d} = \vec{b}^d(f) + \vec{b}^{n,d}(A^n, \vec{U}_{i-1}^{n,d'})\)

6: end for

7: end for

8: end for

9: Approximate the output distribution

This algorithm is expensive for a large number of realizations

The main expense comes from the matrix inversions in the core of the algorithm
Recall the finite element mesh is a refinement of the mesh $\mathcal{K}$ defining the random perturbation

$A^{n,d}$ is constant on each $\Omega_d \implies (k^{n,d})_{lk} = A^{n,d}(\nabla \varphi_{d,l}, \nabla \varphi_{d,k})_{d} = A^{n,d}(k^{d})_{lk},$

$k^d$ is the standard stiffness matrix

$k^d$ is independent of $A^{n,d}$

Can we exploit this?
A Neumann Series Approximation

Recall the Neumann series for an operator

$$(I - C)^{-1} = I + C + C^2 + C^3 + \cdots \quad \|C\| < 1$$

Applying this under suitable assumptions yields

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

We take a finite number of terms to define the approximation

$$\bar{U}_{\mathcal{P},i}^{n,d} = \sum_{p=0}^{\mathcal{P}-1} (-A^{n,d})^p \left((k^{a,d})^{-1}k^d\right)^p (k^{a,d})^{-1} \left(\bar{b}^d(f) + \bar{b}^{n,d}(A^n, U_{\mathcal{P},i-1}^{n,d'})\right)$$

Only uses inversions of the fixed deterministic stiffness matrix

Let $\mathcal{V}_{h,d} = \text{set of vectors determined by the basis functions for the boundary nodes on } \Omega_d$
An Efficient Monte Carlo Algorithm

1: Draw samples \( \{A^n\}_{n=1}^N \) from the distribution of \( A \)
2: for \( i = 1, \cdots, I \) (iterations in domain decomposition) do
3: for \( d = 1, \cdots, D \) (domains in domain decomposition) do
4: for \( p = 1, \cdots, P \) (terms in Neumann series) do
5: Compute \( \vec{y} = ((k^{a,d})^{-1}k^d)^p (k^{a,d})^{-1}b^d(f) \)
6: Compute \( y^p = ((k^{a,d})^{-1}k^d)^p (k^{a,d})^{-1}W_{h,d} \)
7: end for
8: for \( n = 1, \cdots, N \) (number of samples) do
9: \( \vec{U}^{n,d}_{P,i} = \sum_{p=0}^{P-1} (-A^{n,d})^p (y^p \vec{b}^{n,d}(A^n, U^{n,d'}_{P,i-1}) + \vec{y}) \)
10: end for
11: end for
12: end for
13: Approximate the output distribution
An Efficient Monte Carlo Algorithm

This algorithm moves the statistics computation into the “inside loop”

Two observations

- The number of linear systems that have to be solved is independent of the number of samples $N$!

- Since $(k_{a,d}^{-1}k_d^{-1})p(k_{a,d}^{-1})\mathcal{W}_{h,d}$ only involves internal boundary nodes, we can reduce computational cost further by pre-computing it, e.g. using PLU factorization

This is an example of the general idea of “switching the order of the loops” in a stochastic differential equation computation
Convergence Analysis
Convergence Study

We use a $9 \times 9$ uniform partition of $[0, 1] \times [0, 1]$

$a$ is .1 in the interior “cross”, 1 elsewhere

$A$ is uniformly distributed in $\pm 10\%$ of the value of $a$

Reference solution has $h = 1/72$, $I = 400$, $P = 5$, $N = 480$
Convergence Study

![Graphs showing relative error and approximate distribution for different numbers of samples.](image)
Convergence of the Iteration

The convergence of the domain decomposition method is well understood (Lion)

For the Neumann series

**Theorem** If \( \eta = \max\{A^{n,d}\}/a_0 < 1 \) and \( c^{n,d} = -A^{n,d}(k^{a,d})^{-1}k^d \), then

\[
\left\| \left( (1 - c^{n,d})^{-1} - \sum_{p=0}^{P-1} (c^{n,d})^p \right) v \right\| _d \leq \frac{\eta^P}{1 - \eta^P} \left\| \sum_{p=0}^{P-1} (c^{n,d})^p v \right\| _d,
\]

for any \( v \in V_{h,d} \) and a certain norm \( \|\|\|\|_d \).
Convergence of the Iteration

Given the convergence of the domain decomposition iteration and Neumann series,

The issue is to analyze the effect of the imprecise linear algebra on the convergence of the domain decomposition iteration.

It suffices to treat one realization, and we drop the subscripts $n$ and $\mathcal{P}$ for clarity.
Convergence of the Iteration

Let $G$ be the value (trace) of $U$ on the internal boundaries $\partial \Omega_d \cap \partial \Omega_{\bar{d}}$

We measure the error of the $i^{th}$ iterate $\{U_i, G_i\}$ by

$$\|\|\{U, G\} - \{U_i, G_i\}\|^2 = \sum_{d=1}^{D} \left( \|\sqrt{A}(U^d - U^d_i)\|_{L^2(\Omega_d)}^2 + \|G^d - G^d_i\|_{L^2(\partial \Omega_d)}^2 \right)$$
Convergence of the Iteration

Assume that the domain decomposition iteration converges at a geometric rate

\( T \) is the operator giving one iteration of the domain decomposition

\( L \) is a constant reflecting the “complexity” of the partition (winding number)

Theorem There is a fixed integer \( n_0 \) depending on \( L \) and constants \( 0 < \gamma < 1 \) and \( 0 < \delta < 1 \) such that for all \( P \) sufficiently large,

\[
\|\|\|\{U, G\} - (T^{n_0}(\{U_0, G_0\}))^k\|\|\| \leq \gamma^k C\|\|\|\{U, G\} - \{U_0, G_0\}\|\|\| + C\delta^P
\]

\( \boxed{\|\|\|\|\{U, G\} - \{U_0, G_0\}\|\|\|\|} \) Neumann series

domain decomposition
A Posteriori Error Analysis
Two important questions for analysis of a numerical method:

- (Classic) For a given class of problem and data, how does the approximation behave as the discretization is refined?
- (A Posteriori) What is the error in a particular numerical solution of a particular problem computed on a given fixed discretization with particular data?

The first question is relevant to evaluating the qualities of a proposed numerical method.

The second question is important for the practical use of a numerical method.

In many situations, we cannot compute highly resolved solutions or reach the asymptotic convergence behavior.
There are two approximations used for the computed distribution

- (Deterministic) We compute numerical solutions $\tilde{U} \approx U$
- (Stochastic) We compute only a finite number of sample values $Q(U(\lambda_i))$

These sources interact: As parameter values vary, solution behavior varies, and so does numerical accuracy.
Approximating a CDF

We estimate the error in an approximate cumulative distribution function for the quantity of interest $Q(U)$

The solution $U$ depends implicitly on a random vector $\lambda$

We approximate the CDF

$$F(t) = P(\{\lambda : Q(U(\lambda)) \leq t\}) = P(Q \leq t)$$

using a finite number of approximate sample values $\{\tilde{Q}^n\}$:

$$\tilde{F}_N(t) = \frac{1}{N} \sum_{n=1}^{N} I(\tilde{Q}^n \leq t),$$

$I$ is the indicator function

$$\tilde{Q}^n = Q(\tilde{U}^n)$$ is computed using a numerical solution $\tilde{U}^n \approx U^n$
A Decomposition of the Error

We use the sample distribution function

\[ F_N(t) = \frac{1}{N} \sum_{n=1}^{N} I(Q^n \leq t) \]

The error is decomposed into the stochastic and deterministic components

\[ |F(t) - \tilde{F}_N(t)| \leq |F(t) - F_N(t)| + |F_N(t) - \tilde{F}_N(t)| = I + II \]

We assume that there is an error estimate

\[ \tilde{Q}^n - Q^n \approx \mathcal{E}^n \]
Properties of a Sample CDF

\( F_N \) has very desirable properties for numerical analysis, e.g.

- As a function of \( t \), \( F_N(t) \) is a distribution function
- For each fixed \( t \), \( F_N(t) \) is a random variable corresponding to the sample
- It is an unbiased estimator, i.e., \( E(F_N) \equiv E(F) \)
- \( NF_N(t) \) has exact binomial distribution for \( N \) trials and probability of success \( F(t) \)
- \( \text{Var}(F_N(t)) = F(t)(1 - F(t))/N \to 0 \) as \( N \to \infty \), and \( F_N \) converges in mean square to \( F \) as \( N \to \infty \)

A standard measure of accuracy is the Kolmogorov–Smirnov distance

\[
\sup_{t \in \mathbb{R}} \left| F_N(t) - F(t) \right|
\]
Bounds on an Sample CDF

Since \( \{I(Q^n \leq t)\} \) are iid Bernoulli, the Chebyshev inequality implies that for \( \epsilon > 0 \),

\[
P \left( \left| F(t) - F_N(t) \right| \leq \left( \frac{F(t)(1 - F(t))}{N \epsilon} \right)^{1/2} \right) > 1 - \epsilon
\]

Since

\[F(t)(1-F(t)) = F_N(t)(1-F_N(t)) + (F(t)-F_N(t))(1-F(t)-F_N(t))\]

For \( \epsilon > 0 \),

\[
P \left( \left| F(t) - F_N(t) \right| \leq \left( \frac{F_N(t)(1 - F_N(t))}{N \epsilon} \right)^{1/2} + \frac{1}{2N \epsilon} \right) > 1 - \epsilon
\]
We next estimate the effect of using approximate values on the sample distribution function

\[ |F_N(t) - \tilde{F}_N(t)| = \left| \frac{1}{N} \sum_{n=1}^{N} \left( I(\tilde{Q}^n \leq t) - I(Q \leq t) \right) \right| \]

---

**Contribution to Error**

- **No**
- **Yes**
- **No**

<table>
<thead>
<tr>
<th>No</th>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>(t)</td>
<td>(\tilde{Q})</td>
<td>(t)</td>
</tr>
</tbody>
</table>
Effect of Approximate Values

Using \( \tilde{Q}^n = Q^n + \mathcal{E}^n \),

\[
|F_N(t) - \tilde{F}_N(t)| = \left| \frac{1}{N} \sum_{n=1}^{N} (I(Q^n + \mathcal{E}^n \leq t) - I(Q \leq t)) \right|
\]

\[
= \left| \frac{1}{N} \sum_{n=1}^{N} (I(Q^n \leq t \leq Q^n + |\mathcal{E}^n|)) \right|
\]

\[
+ \frac{1}{N} \sum_{n=1}^{N} (I(Q^n - |\mathcal{E}^n| \leq t \leq Q^n)) \right|
\]

We obtain the bound

\[
|F_N(t) - \tilde{F}_N(t)| \leq \left| \frac{1}{N} \sum_{n=1}^{N} (I(Q^n - |\mathcal{E}^n| \leq t \leq Q^n + |\mathcal{E}^n|)) \right|
\]
Expanding with $Q^n = \tilde{Q}^n - \mathcal{E}^n$ yields the computable estimate

$$|F_N(t) - \tilde{F}_N(t)| \leq \left| \frac{1}{N} \sum_{n=1}^{N} \left( I(\tilde{Q}^n - |\mathcal{E}^n| \leq t \leq \tilde{Q}^n + |\mathcal{E}^n|) \right) \right|$$

To obtain a computable estimate on I, we use

$$F_N(t)(1 - F_N(t)) = \tilde{F}_N(t)(1 - \tilde{F}_N(t)) + (F_N(t) - \tilde{F}_N(t))(1 - F_N(t) - \tilde{F}_N(t))$$

along with bounds from the analysis of II
Theorem  For any $\epsilon > 0$, 

$$
|F(t) - \tilde{F}_N(t)| \leq \left( \frac{\tilde{F}_N(t)(1 - \tilde{F}_N(t))}{N\epsilon} \right)^{1/2}
+ 2 \left| \frac{1}{N} \sum_{n=1}^{N} \left( I(\tilde{Q}^n - |E^n| \leq t \leq \tilde{Q}^n + |E^n|) \right) \right| + \frac{1}{2N\epsilon}
$$

with probability greater than $1 - \epsilon$
Performance of Distribution Error Estimate

We generate a true distribution and then an approximation computed with finite sampling and adding random errors to each value.

The original distribution is exponential with parameter 1 with a relative 5% error.

\[ N = 500 \]
A Computational Point

Note that

\[ \left| \frac{1}{N} \sum_{n=1}^{N} \left( I(\tilde{Q}^n - |E^n| \leq t \leq \tilde{Q}^n + |E^n|) \right) \right| \]

is itself an (approximate) expected value.

If \( M < N \) and \( N' = \{n_1 < \cdots < n_M\} \) is a set chosen randomly from \( \{1, \ldots, N\} \), we can use the estimator

\[ \left| \frac{1}{M} \sum_{n \in N'} \left( I(\tilde{Q}^n - |E^n| \leq t \leq \tilde{Q}^n + |E^n|) \right) \right| \]

The error decreases as \( O\left(1/\sqrt{M}\right) \), which gives reasonable accuracy when \( N \) is large.
A Posteriori Error Estimate for the Samples

We solve a differential equation

\[ L(U) = f \]

for a quantity of interest given by a linear functional

\[ (U, \psi) \quad \psi \text{ determines the quantity of interest} \]

We compute a numerical approximation

\[ \tilde{U} \approx U \]

The local resolution of the approximation is measured using the residual

\[ R(\tilde{U}) = L(\tilde{U}) - f \]

The residual is not the error!
A Posteriori Error Estimate for the Samples

To build an error estimate, we solve a linear adjoint problem

\[(DL)^* \phi = \psi\]

Induced by the original problem, the adjoint problem is a map on the “space of quantities of interest”

We obtain the error representation

\[(\tilde{U} - U, \psi) = (R(\tilde{U}), W(\phi))\]

for a suitable adjoint weight \(W(\phi)\)

- The adjoint weight carries information about stability or sensitivity of \(U\) to local perturbations
- The estimate accounts for the accumulation, cancellation, and propagation of local error contributions

The stability information is specific to the quantity of interest
The sample linear functional value is \((U^n, \psi)\)

The adjoint problem:

\[
\begin{cases}
-\nabla \cdot A \nabla \Phi = \psi, & x \in \Omega, \\
\Phi = 0, & x \in \partial \Omega.
\end{cases}
\]

The sample adjoint solution \(\Phi^{n,d}_{\tilde{I},\tilde{P}}\) is computed using a higher order finite element space \(V_\tilde{h}\) to capture “Galerkin orthogonality”
A Posteriori Error Estimate for the Samples

Theorem

\[ |(U^n - U^n_{I,P}, \psi)| \lesssim |(f, \Phi^n_{I,\bar{P}}) - (A^n \nabla U^n_{I,P}, \nabla \Phi^n_{I,\bar{P}})| + R(\tilde{h}, \tilde{I}, \tilde{P}), \]

where

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

\(\mathcal{V}_h^2\) is the space of piecewise continuous quadratic functions on \(\mathcal{T}_h\).
Accuracy of the A Posteriori Estimate for Samples

Let \( a = 0.9 \) and \( A^1 = 0.1 \) on the unit square with a partition of \( 8 \times 8 \) subsquares, \( \psi = 1 \), \( h = 1/32 \) for the forward solution and \( h = 1/64 \) for the adjoint solve.

Accurate adjoint solution, \( \mathcal{P}_{adj} = 2\mathcal{P}_{fwd} \), \( \mathcal{I}_{adj} = 2\mathcal{I}_{fwd} \).

<table>
<thead>
<tr>
<th>( \mathcal{P} )</th>
<th>( \mathcal{I} )</th>
<th>Ratio</th>
<th>( \mathcal{P} )</th>
<th>( \mathcal{I} )</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 50</td>
<td>0.992</td>
<td></td>
<td>3 10</td>
<td>2.78</td>
<td></td>
</tr>
<tr>
<td>2 50</td>
<td>1.03</td>
<td></td>
<td>3 25</td>
<td>1.02</td>
<td></td>
</tr>
<tr>
<td>3 50</td>
<td>1.01</td>
<td></td>
<td>3 50</td>
<td>1.01</td>
<td></td>
</tr>
</tbody>
</table>

Inaccurate adjoint solution, \( \mathcal{P}_{adj} = 0.5\mathcal{P}_{fwd} \), \( \mathcal{I}_{adj} = 0.5\mathcal{I}_{fwd} \).

<table>
<thead>
<tr>
<th>( \mathcal{P} )</th>
<th>( \mathcal{I} )</th>
<th>Ratio</th>
<th>( \mathcal{P} )</th>
<th>( \mathcal{I} )</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 50</td>
<td>0.908</td>
<td></td>
<td>3 10</td>
<td>8.21</td>
<td></td>
</tr>
<tr>
<td>2 50</td>
<td>1.00</td>
<td></td>
<td>3 25</td>
<td>1.08</td>
<td></td>
</tr>
<tr>
<td>3 50</td>
<td>0.933</td>
<td></td>
<td>3 50</td>
<td>0.933</td>
<td></td>
</tr>
</tbody>
</table>
The complete result is

**Theorem** For any $\epsilon > 0$,

\[
|F(t) - \tilde{F}_N(t)| \leq \left( \frac{\tilde{F}_N(t)(1 - \tilde{F}_N(t))}{N\epsilon} \right)^{1/2} + 2 \left| \frac{1}{N} \sum_{n=1}^{N} \left( I(\tilde{Q}^n - |\mathcal{E}^n| \leq t \leq \tilde{Q}^n + |\mathcal{E}^n|) \right) \right| + \frac{1}{2N\epsilon}
\]

with probability greater than $1 - \epsilon$, where

\[
|\mathcal{E}^n| = \left| \langle f, \Phi^n_{\mathcal{I},\mathcal{P}} \rangle - (A^n \nabla U^n_{\mathcal{I},\mathcal{P}}, \nabla \Phi^n_{\mathcal{I},\mathcal{P}}) \right|
\]

This is an uncertainty quantification result.
Adaptive Error Control
Motivation

Despite the algorithm, computational cost is a concern

- We may have to take a large number of samples
- Domain decomposition and Neumann series approximations may converge slowly

We construct an *a posteriori* adaptive algorithm that iteratively adjusts each parameter used in the approximation algorithm to obtain a desired accuracy
Adaptive error control generally uses upper bounds that are easily evaluated.

We use

\[
2 \left| \frac{1}{N} \sum_{n=1}^{N} \left( I(\tilde{Q}^n - |\mathcal{E}^n| \leq t \leq \tilde{Q}^n + |\mathcal{E}^n|) \right) \right| \leq 2L \max_{1 \leq n \leq N} |\mathcal{E}^n|
\]

where \( L \) is the Lipschitz constant of \( F \).
Adaptive Error Control

We further manipulate the a posteriori bound

$$|F(t) - \tilde{F}_N(t)| \leq \mathcal{E}_1 + \mathcal{E}_{\text{II}} + \mathcal{E}_{\text{III}} + \mathcal{E}_{\text{IV}}$$

$$\mathcal{E}_1 \approx 2L \max_n |(f, \Phi^n_{\tilde{P}, \tilde{I}}) - (A^n \nabla U^n_{\tilde{P}, \tilde{I}}, \nabla \Phi^n_{\tilde{P}, \tilde{I}}) - (A^n \nabla (U^n_{\tilde{P}, \tilde{I}} + \Delta \tilde{P}, \tilde{I} - U^n_{\tilde{P}, \tilde{I}}), \nabla \Phi^n_{\tilde{P}, \tilde{I}})|$$

$$\mathcal{E}_{\text{II}} \approx 2L \max_n |(A^n \nabla (U^n_{\tilde{P}, \tilde{I}} + \Delta \tilde{I} - U^n_{\tilde{P}, \tilde{I}}), \nabla \Phi^n_{\tilde{P}, \tilde{I}})|$$

$$\mathcal{E}_{\text{III}} \approx 2L \max_n |(A^n \nabla (U^n_{\tilde{P}, \tilde{I}} + \Delta \tilde{P}, \tilde{I} - U^n_{h, \tilde{P}, \tilde{I}}), \nabla \Phi^n_{\tilde{P}, \tilde{I}})|$$

$$\mathcal{E}_{\text{IV}} = \left( \frac{\tilde{F}_N(t)(1 - \tilde{F}_N(t))}{N\epsilon} \right)^{1/2}$$

$\Delta \tilde{P}$, $\Delta \tilde{I}$ are incremental improvements
Adaptive Error Control via Optimization

We find discretization parameter values that produce a desired accuracy at a relatively low computational cost.

We pose this as an optimization problem that is solved iteratively.

We attempt to balance the relative costs of changing the various parameters.

We partition a global error tolerance into individual contribution tolerances based on relative cost estimates.
Adaptive Error Control via Optimization

1. Choose error tolerance TOL and compute contribution tolerances
2. Choose initial (coarse) discretization parameters
3. Compute initial (coarse) solution
4. Compute error estimate for initial solution
5. while Error Estimate > TOL do
6. Adjust discretization parameters depending on the contributions relative to the contribution tolerances
7. Compute solution
8. Compute error estimate for solution
9. end while
Examples
Example 1

We use a $9 \times 9$ uniform partition of $[0, 1] \times [0, 1]$

Here $a$ is .1 in the interior “cross”, 1 elsewhere, $A$ is uniformly distributed in $\pm 10\%$ of the value of $a$

We ask for 15% error with probability 0.95.

Initial: $h = 1/18, \mathcal{I} = 40, \mathcal{P} = 1, \text{ and } \mathcal{N} = 60, \text{ and } \Delta \mathcal{I} = 0.3 \mathcal{I}, \Delta \mathcal{P} = 1$
The tolerance was reached after three iterations with $h = 1/54$, $I = 160$, $P = 3$, and $N = 240$. 
The tolerance was reached after three iterations with $h = 1/54$, $I = 160$, $P = 3$, and $N = 240$. 
Example 1

Error in the approximate CDFs:

The tolerance was reached after three iterations with $h = 1/54$, $\mathcal{I} = 160$, $\mathcal{P} = 3$, and $\mathcal{N} = 240$
Example 2

$A$ is uniformly distributed in $\pm 20\%$ of a realistic value for $a$

$f = 1$ near $(0, 0)$ and $f = -1$ near $(1, 1)$

We use a $27 \times 27$ uniform partition of $[0, 1] \times [0, 1]$ and a fixed $135 \times 135$ finite element mesh
Example 2

Tolerance is \(0.15\) with confidence 95%

Initial: \(I = 100, P = 1, N = 30\)
Example 2

Decrease of Error Estimators

![Graph showing decrease of error estimators](image-url)
Example 2

Approximate Distributions

[Graphs showing approximate distributions with x-axis ranging from $-12 \times 10^{-4}$ to $-5 \times 10^{-4}$ and y-axis from 0 to 1.]
Modeling Error
Estimating Model Error

We consider the effect of representing the random perturbation by a piecewise constant function

We construct an improved approximation $\bar{A}^n$ on each subdomain and add a new term that includes a residual for the local approximation of $A^n$

This is estimated as

$$E_V = 2L \max_n \left| (A^n - \bar{A}^n) \nabla U^n_{h,P,I,\nabla \Phi} \right|$$

We modify the adaptive error control to include this term

Model error is treated similarly to quadrature error in a posteriori error analysis
Let $a = 1$ and $f = 1$, $\psi = 1$ for $0.9 \leq x, y \leq 1$ and zero otherwise.

The random perturbation is uniformly distributed in an interval of size $10\%$ for $0 \leq x < 0.75$ and $50\%$ for $0.75 \leq x \leq 1$. 
Adaptive Modeling Example

Parameters chosen by the adaptive algorithm

Right: $\mathcal{I}$. Middle: $\mathcal{P}$. Left: $\mathcal{N}$

The tolerance is reached when $\mathcal{I} = 320$, $\mathcal{P} = 5$, and $\mathcal{N} = 640$. 
Adaptive Modeling Example

Meshes for representing the random perturbation constructed by the adaptive algorithm for the first three iterations

Region of the Quantity of Interest is Marked in Red
Region of High Variance in the Perturbation is Marked in Green
Adaptive Modeling Example

The error contributions as the adaptive algorithm proceeds
Adaptive Modeling Example

The computed distributions
Conclusion
We study the nonparametric density estimation problem for an elliptic problem with a randomly perturbed diffusion coefficient.

Using a piecewise constant representation of the random perturbation, domain decomposition, and Neumann series, we construct a very efficient numerical method.

We derive an a posteriori error estimate for the output distribution that accounts for both stochastic and deterministic errors.

We construct an iterative adaptive algorithm that adjusts all discretization parameters to achieve efficiency.

We include an adaptive modeling term that reflects the accuracy of the representation of the random perturbation.
References


