Numerical Analysis and Adaptive Computation for Solutions of Elliptic Problems with Randomly Perturbed Coefficients

Donald Estep **Axel Målqvist** **Simon Tavener**

Department of Mathematics

Colorado State University and University of California at San Diego

Research supported by

Department of Energy: DE-FG02-04ER25620 and DE-FG02-05ER25699;

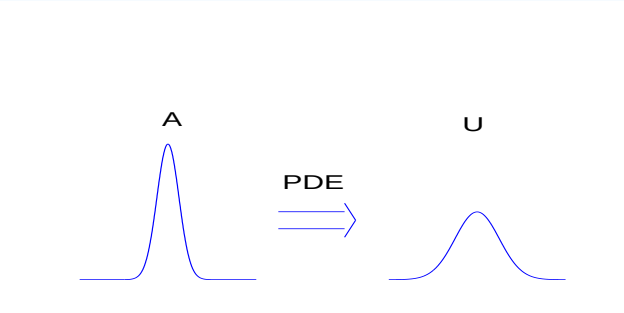
Sandia Corporation: PO299784;

National Aeronautics and Space Administration: NNG04GH63G;

National Science Foundation: DMS-0107832, DGE-0221595003, MSPA-CSE-0434354

Outline

Given samples of the data A to a PDE, the goal is to compute samples of the solution U cheaply.



- A model problem with randomly perturbed coefficient
- A method for computing samples of the solution
- Convergence and duality based a posteriori error analysis
- Adaptive algorithm
- Numerical examples
- Conclusions and future work

Poisson equation with randomly perturbed coefficient

Strong form:

$$-\nabla \cdot \mathcal{A}^s \nabla U^s = f \quad \text{in } \Omega,$$
$$U^s = 0 \quad \text{on } \Gamma.$$

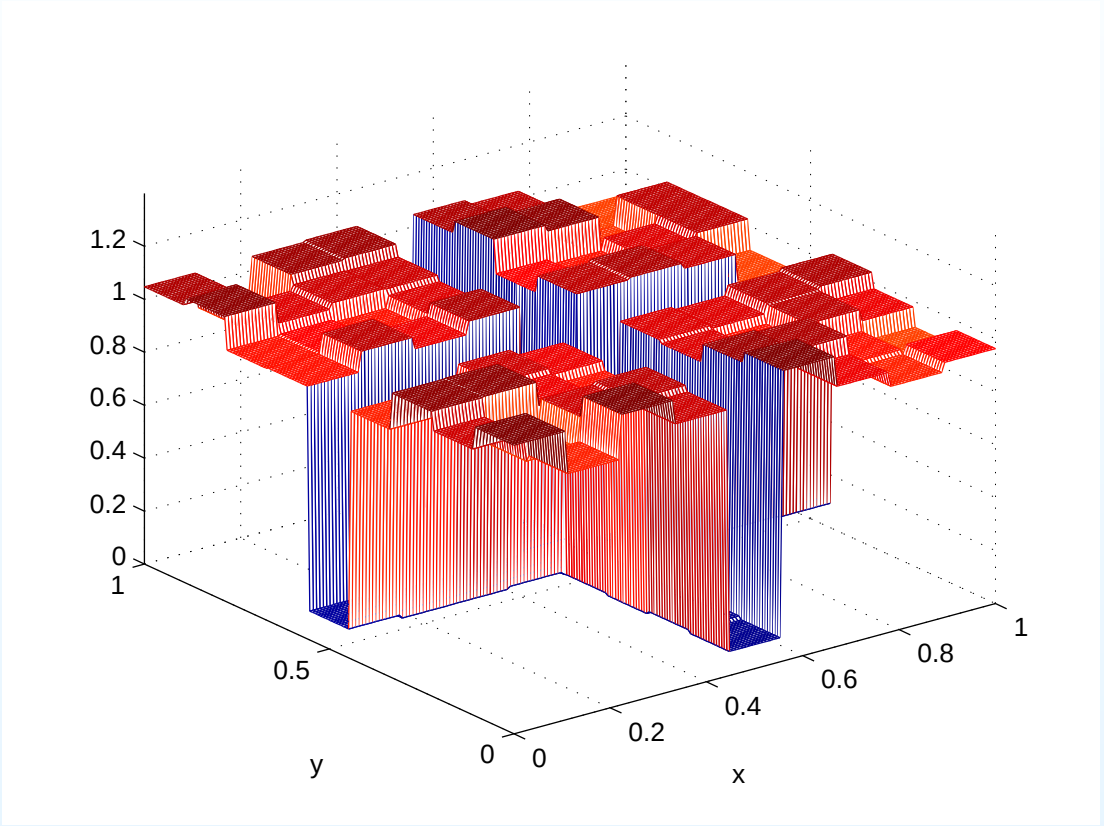
- We assume that $\mathcal{A}^s = a + A^s \geq \alpha > 0$,
- that a is deterministic with multiscale features,
- that A^s are piecewise constant, iid, random pert., $s \in \Lambda$,
- and that $f \in L^2(\Omega)$ is deterministic,

Weak form:

Find $U^s \in V = H_0^1(\Omega)$ such that,

$$(\mathcal{A}^s \nabla U^s, \nabla v) = (f, v) \quad \text{for all } v \in V.$$

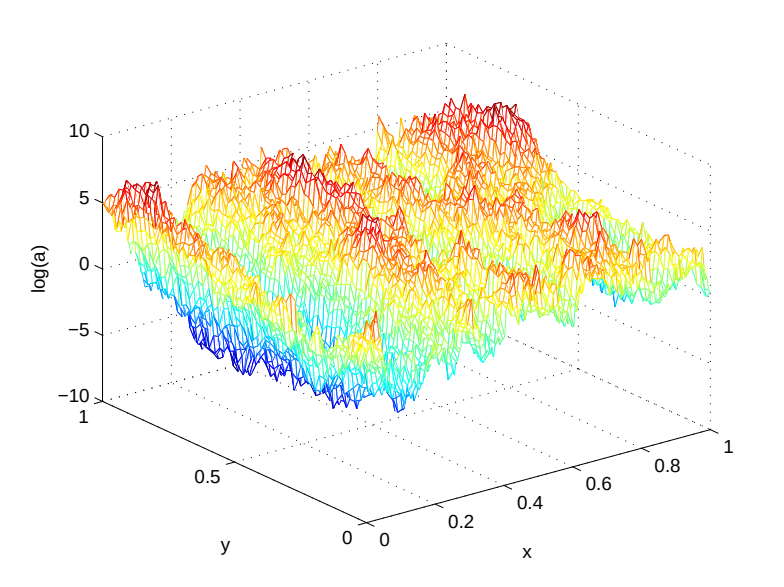
Diffusion coefficient



A piecewise constant random perturbation is added to a deterministic diffusion coefficient.

Motivation

- There are often measurement errors in field data.
- One way to model this uncertainty is as a random perturbations of the data.
- The sensitivity in the solution to these perturbations is important to understand if we want to be able to rely on the solution.



Simple observation

Note that if we want to find a^* such that,

$$-\nabla \cdot a^* \nabla E[U^s] = f$$

then $a^* \neq E[\mathcal{A}^s]$ in general, in fact if $\mathcal{A}^s$ is constant in space we have that,

$$a^* = \frac{1}{E[\frac{1}{\mathcal{A}^s}]}.$$

In general there is no simple expression.

This means that even finding $E[U^s]$ for this problem is non-trivial.

The goal of this work is to compute stochastic quantities of $\{U^s\}_{s\in\Lambda}$ cheaply given $\{A^s\}_{s\in\Lambda}$.

Monte Carlo finite element method

Here we solve one PDE for each sample $\mathcal{A}^s$.

for s from 1 to S **do**
$\mathcal{A}^s = a + A^s$
$U^s = \text{solver}(f, \mathcal{A}^s)$
end for
$E[U] \approx \sum_{s=1}^{S} U^s/S$ for example.

- Positive: We have full access to $\{U^s\}_{s=1}^{S}$. It is possible to get a good picture of how sensitive the solution is to the perturbations.
- Negative: Expensive since we need to solve S PDE's all with different operators and multiscale features which means that a high resolution is necessary.

Want same information to much lower cost.

In case of constant perturbation

If A^s was a constant perturbation (in space) we could use a Neumann series to compute the inverse of the matrix,

$$U^s = (\mathbf{k}^a + A^s\mathbf{k})^{-1}b = (I + A^s(\mathbf{k}^a)^{-1}\mathbf{k})^{-1}(\mathbf{k}^a)^{-1}b$$

$$= \sum_{t=0}^{\infty}(-A^s(\mathbf{k}^a)^{-1}\mathbf{k})^t(\mathbf{k}^a)^{-1}b,$$

where $\mathbf{k}^a_{i,j} = (a\nabla\phi_i, \nabla\phi_j)$, $\mathbf{k}_{i,j} = (\nabla\phi_i, \nabla\phi_j)$, and $b_j = (f, \phi_j)$, given a set of finite element basis function $\{\phi_i\}$.

- Positive: If Neumann series converges quickly we use truncated version. We only need to invert the matrix once.
- Negative: a has multiscale features $\rightarrow$ expensive to solve, since we need high resolution to get an accurate solution.

The main idea

However, when A^s ($\mathcal{A}^s = a + A^s$) is piecewise constant we can use a non-overlapping domain decomposition algorithm with domains that coincide with the regions where A^s is constant.

- If the Neumann series converges quickly we can still compute samples of the solution by just multiplying and adding random numbers and vectors (computed using matrix vector multiplication), now individually on the domains.
- We only need to invert the matrices on each domain which is much cheeper.
- When computing good approximations of a stochastic quantity the number of samples needed to get accurate solution is a bottle neck.
- Domain decomposition is very suited for solving multiscale problems.

Lions' non-overlapping domain decomposition method

We use the following non-overlapping domain decomposition algorithm proposed by Lions, here presented on two domains $\Omega = \Omega_1 \cup \Omega_2$ for simplicity,

$$
\begin{aligned}
-\nabla \cdot \mathcal{A}^s \nabla U^{s,1}_{(i)} &= f, \quad \text{in } \Omega_1, \\
U^{s,1}_{(i)} &= 0, \quad \text{on } \partial\Omega_1 \cap \Gamma, \\
U^{s,1}_{(i)} + \lambda \boldsymbol{n}_1 \cdot \mathcal{A}^s \nabla U^{s,1}_{(i)} &= U^{s,2}_{(i-1)} - \lambda \boldsymbol{n}_2 \cdot \mathcal{A}^s \nabla U^{s,2}_{(i-1)}, \quad \text{on } \partial\Omega_1 \cap \partial\Omega_2, \\
-\nabla \cdot \mathcal{A}^s \nabla U^{s,2}_{(i)} &= f, \quad \text{in } \Omega_2, \\
U^{s,2}_{(i)} &= 0, \quad \text{on } \partial\Omega_2 \cap \Gamma, \\
U^{s,2}_{(i)} + \lambda \boldsymbol{n}_2 \cdot \mathcal{A}^s \nabla U^{s,2}_{(i)} &= U^{s,1}_{(i-1)} - \lambda \boldsymbol{n}_1 \cdot \mathcal{A}^s \nabla U^{s,1}_{(i-1)},
\end{aligned}
$$

where (i) is the iterate in the domain decomposition algorithm.

Monte Carlo with Lions' Domain Decomposition solver

We introduce Lions' non-overlapping domain decomposition method as the solver in the Monte Carlo finite element method.

for s from 1 to S **do**
  for i from 1 to I **do**
    for d from 1 to D **do**
      Compute $U_{(i)}^{s,d} = (\mathbf{k}^a + A^{s,d}\mathbf{k})^{-1} b^s(f, \mathcal{A}^s, U_{(i-1)}^s)$.
    end for
  end for
end for

Here d indicates a certain domain in the DD algorithm.

Note that the loop over s is independent of the other loops.

Monte Carlo with DD: reversed order

Since the loops are independent we can reverse the order,

for i from 1 to I **do**
  for d from 1 to D **do**
    for s from 1 to S **do**
      Compute $U_{(i)}^{s,d} = (\mathbf{k}^a + A^{s,d}\mathbf{k})^{-1} b^s(f, \mathcal{A}^s, U_{(i-1)}^s)$
    end for
  end for
end for

Remember that the random perturbation $A^{s,d}$ is a constant. This means that on each domain d we want to solve S problems with very similar matrices, that can be approximately inverted using a truncated Neumann series.

The proposed method

We approximate the inverse of the
perturbed matrix with a truncated Neumann series using T terms,

for i from 1 to I **do**
  for d from 1 to D **do**
    for t from 0 to T **do**
      Compute $\mathbf{c}^t = ((\mathbf{k}^a)^{-1}\mathbf{k})^t(\mathbf{k}^a)^{-1}$
    end for
    for s from 1 to S **do**
      $U^{s,d}_{(i)} \approx \sum_{t=0}^{T-1}(-A^{s,d})^t\mathbf{c}^t b^s(f, \mathcal{A}^s, U^s_{(i-1)})$
    end for
  end for
end for

We want to minimize the work in the inner loop.

Convergence of the Neumann series

Let $\|\cdot\|$ be an operator norm. If we assume $|A^{s,d}| < a^* = \min_{x\in\Omega_d} a$ then,

$$(i) \qquad \|(A^{s,d}(\mathbf{k}^a)^{-1}\mathbf{k})^t\| \leq C\left(\frac{A^{s,d}}{a^*}\right)^t,$$

$$(ii) \qquad (I + A^{s,d}(\mathbf{k}^a)^{-1}\mathbf{k})^{-1} = \sum_{t=0}^{\infty}(-A^{s,d}(\mathbf{k}^a)^{-1}\mathbf{k})^t,$$

$$(iii) \qquad \|(I + A^{s,d}(\mathbf{k}^a)^{-1}\mathbf{k})^{-1} - \sum_{t=0}^{T-1}(-A^{s,d}(\mathbf{k}^a)^{-1}\mathbf{k})^t\|$$

$$\leq C\left(\frac{A^{s,d}}{a^*}\right)^T \|(I + A^{s,d}(\mathbf{k}^a)^{-1}\mathbf{k})^{-1}\|.$$

Error estimation

We are going to focus on estimating the error of some stochastic quantity, i.e. the distribution function, of a linear functional of the solution.

There are four different error contributions in this method:

1. Space discretization error (h).
2. Error committed by not converging in the domain decomposition algorithm (I).
3. Error committed by truncating the Neumann series (T).
4. Error committed by only using (S) realizations of the solutions in order to compute the desired stochastic quantity.

The goal is to equidistribute the error between these components.

Duality based a posteriori error analysis

The dual problem gives us an error representation formula for each sample,

$$\begin{aligned} -\nabla \cdot \mathcal{A}^s \nabla \Phi^s &= \psi \quad \text{in } \Omega, \\ \Phi^s &= 0 \quad \text{on } \Gamma, \end{aligned}$$

where ψ is deterministic.

$$\begin{aligned} |(U^s - U^s_{h,I,T}, \psi)| = |&(f, \Phi^s) - (\mathcal{A}^s \nabla U^s_{h,\infty,\infty}, \nabla \Phi^s) \\ &+ (\mathcal{A}^s \nabla (U^s_{h,\infty,T} - U^s_{h,I,T}), \nabla \Phi^s) \\ &+ (\mathcal{A}^s \nabla (U^s_{h,\infty,\infty} - U^s_{h,\infty,T}), \nabla \Phi^s)| \\ &\approx e^s_I + e^s_{II} + e^s_{III}, \end{aligned}$$

where $U^s_{h,I,T}$ is the approximate solution.

Second (I) and third (T) term

Let ΔI be some positive number. Given an approximation to the dual solution corresponding to U^s, $\Phi^s_{h',I',T'}$ we assume

$$(\mathcal{A}^s\nabla(U^s_{h,\infty,T}-U^s_{h,I,T}),\nabla\Phi^s)\approx(\mathcal{A}^s\nabla(U^s_{h,I+\Delta I,T}-U^s_{h,I,T}),\nabla\Phi^s_{h',I',T'}).$$

For the third term we use the summation formula, $U^{s,d}_{h,\infty,\infty}=\sum_{t=0}^{\infty}[(-A^{s,d})^t((\mathbf{k}^a)^{-1}\mathbf{k})^t](\mathbf{k}^a)^{-1}b^s$, and, $U^{s,d}_{h,\infty,T}=\sum_{t=0}^{T-1}[(-A^{s,d})^t((\mathbf{k}^a)^{-1}\mathbf{k})^t](\mathbf{k}^a)^{-1}b^s$. This means that, $U^s_{h,\infty,\infty}-U^s_{h,\infty,T}=\sum_{t=T}^{\infty}[(-A^{s,d})^t((\mathbf{k}^a)^{-1}\mathbf{k})^t](\mathbf{k}^a)^{-1}b^s=$ $[(-A^{s,d})^T((\mathbf{k}^a)^{-1}\mathbf{k})^T]U^s_{h,\infty,\infty}$. To approximate this quantity we assume, $U^s_{h,\infty,\infty}-U^s_{h,\infty,T}\approx[(-A^{s,d})^T((\mathbf{k}^a)^{-1}\mathbf{k})^T]U^s_{h,I+\Delta I,T}$,

$$(\mathcal{A}^s\nabla(U^s_{h,\infty,\infty}-U^s_{h,\infty,T}),\nabla\Phi^s)$$
$$\approx(\mathcal{A}^s\nabla(-A^{s,d})^T((\mathbf{k}^a)^{-1}\mathbf{k})^TU^s_{h,I+\Delta I,T},\nabla\Phi^s_{h',I',T'}).$$

First term (h)

The natural thing would be to say,

$$(f,\Phi^s)-(\mathcal{A}^s\nabla U^s_{h,\infty,\infty},\nabla\Phi^s)\approx(f,\Phi^s_{h',T',I'})-(\mathcal{A}^s\nabla U^s_{h,I,T},\nabla\Phi^s_{h',T',I'}).$$

However, $(f,\Phi^s)-(\mathcal{A}^s\nabla U^s_{h,I,T},\nabla\Phi^s)$ is the entire error i.e. the sum of the three parts. So instead we let,

$$\begin{aligned}(f,\Phi^s)-(\mathcal{A}^s\nabla U^s_{h,\infty,\infty},\nabla\Phi^s)\approx{}&(f,\Phi^s_{h',T',I'})-(\mathcal{A}^s\nabla U^s_{h,I,T},\nabla\Phi^s_{h',T',I'})\\&-(\mathcal{A}^s\nabla(U^s_{h,I+\Delta I,T}-U^s_{h,I,T}),\nabla\Phi^s_{h',I',T'})\\&-(\mathcal{A}^s\nabla(-A^{s,d})^T((\mathbf{k}^a)^{-1}\mathbf{k})^TU^s_{h,I+\Delta I,T},\nabla\Phi^s_{h',I',T'}).\end{aligned}$$

This gives us computable approximations to the three first error terms. Note that $\Delta I>0$ to get a non-zeros contribution to the second term and more importantly $h^{'}<h$ in order for the approximation of the first term to make sense.

Stochastic error contribution

For each sample we have,

$$|(U^s - U^s_{h,I,T}, \psi)| \approx e^s_I + e^s_{II} + e^s_{III} = e^s.$$

We want to minimize the error in the distribution function $F(x)$:

$$F(x) - \tilde{F}_S(x) = P(\{(U^s, \psi)\}_{s\in\Lambda} < x) - P(\{(U^s_{h,I,T}, \psi)\}^S_{s=1} < x).$$

We prove, using the Central Limit Theorem, that,

$$|F(x) - \tilde{F}_s(x)| \leq \tau \sqrt{\frac{\tilde{F}_S(x)(1-\tilde{F}_S(x))}{S}} + \max_{s\in\{1,\ldots,S\}} |e^s| \tilde{F}'_S(x),$$

with approximate probability $\int_{-\infty}^{\tau} e^{-t^2/2}\, dt/\sqrt{2\pi}$. This estimate is valid for large values S and can be used in an adaptive algorithm.

Adaptive algorithm

1. Compute $\{U^s_{h,I,T}\}_{s=1}^S$ and $\{\Phi^s_{h',I',T'}\}_{s=1}^S$ given $\{\mathcal{A}^s\}_{s=1}^S$.
2. Compute $\tilde{F}_S(x)$ and an approximation to $\tilde{F}'_S(x)$ using central differences.
3. Compute approximations to the three first parts of the error indicator e_I that depends on h, e_{II} that depends on I, and e_{III} that depends on T, and multiply these by $\tilde{F}'_S(x)$.
4. Compute the error indicator associated with the sample size, $e_{IV} = \tau\sqrt{\tilde{F}(1-\tilde{F})/S}$.
5. If the error is small enough, stop.
6. Otherwise improve h, I, T, and S according to the error indicators.
7. Return to 1.

Numerical example: oil reservoir data

We study a pressure equation that arises in oil reservoir simulation.

$$\begin{aligned} -\nabla \cdot \mathcal{A}^s \nabla U^s &= f \quad \text{in } \Omega, \\ \mathcal{A}^s \partial_n U^s &= 0 \quad \text{on } \Gamma_N, \\ U^s &= 0 \quad \text{on } \Gamma_D, \end{aligned}$$

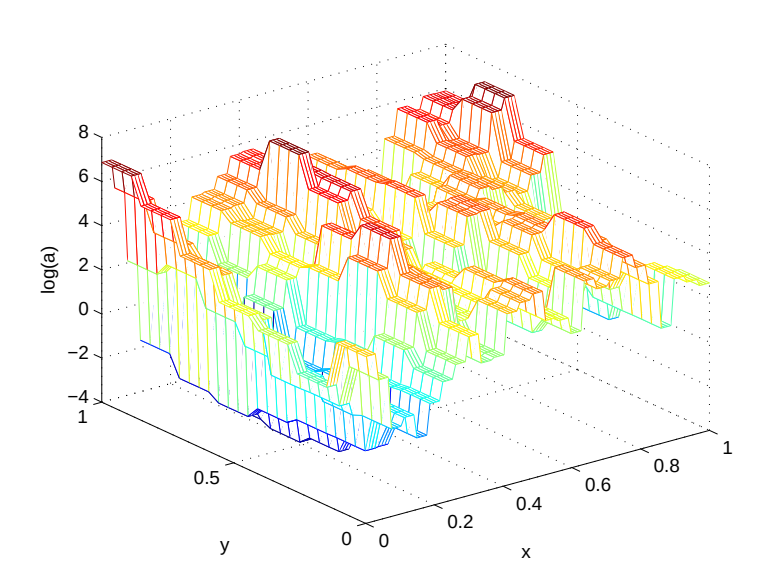
where $\Gamma_N \cup \Gamma_D = \Gamma$. Here U^s represents the pressure field, and a is the local permeability.

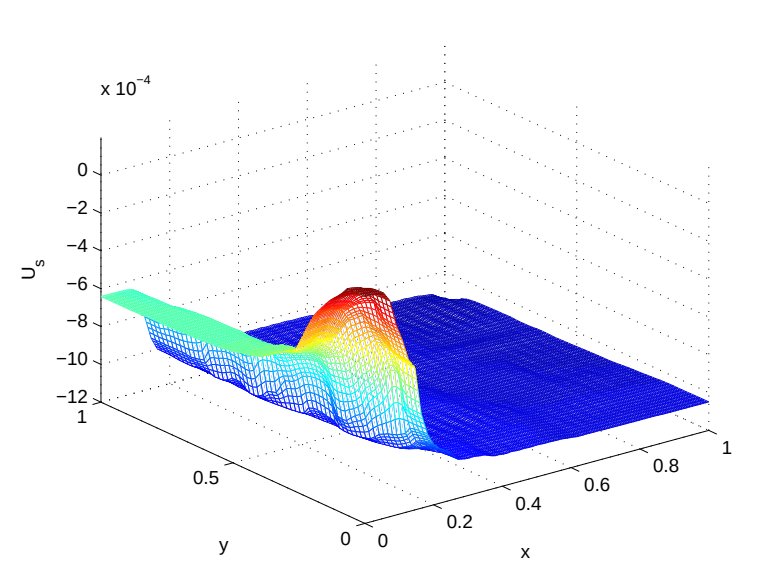
We have chosen $f = 1$ in the lower left corner, the injector, and $f = -1$ in the upper right corner, the producer.

Note that the a posteriori error analysis for this setting is almost identical to the pure Dirichlet case.

Numerical example: oil reservoir data

The permeability is piecewise constant on a 27×7 grid and is plotted in log-scale to the left.





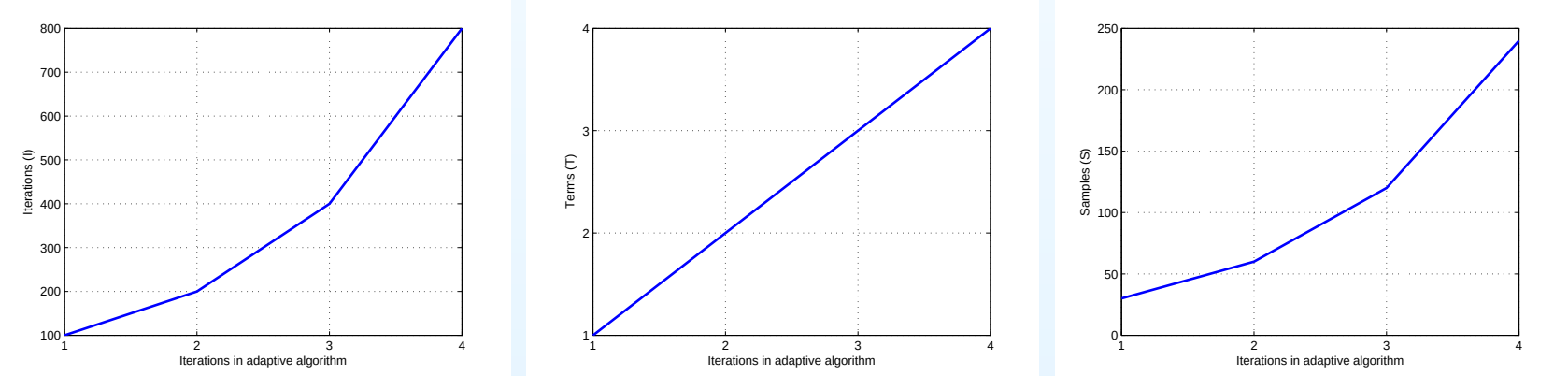
We add a random perturbation to a (20% of the magnitude of a). To the right: a typical solution U^s.

The band of low permeability at $x \approx 0.2$ creates a large pressure drop parallel to the y-axis at this location.

Numerical example: oil reservoir data

We assume the mesh is given and can not be refined due to the size of the problem (common in these applications).

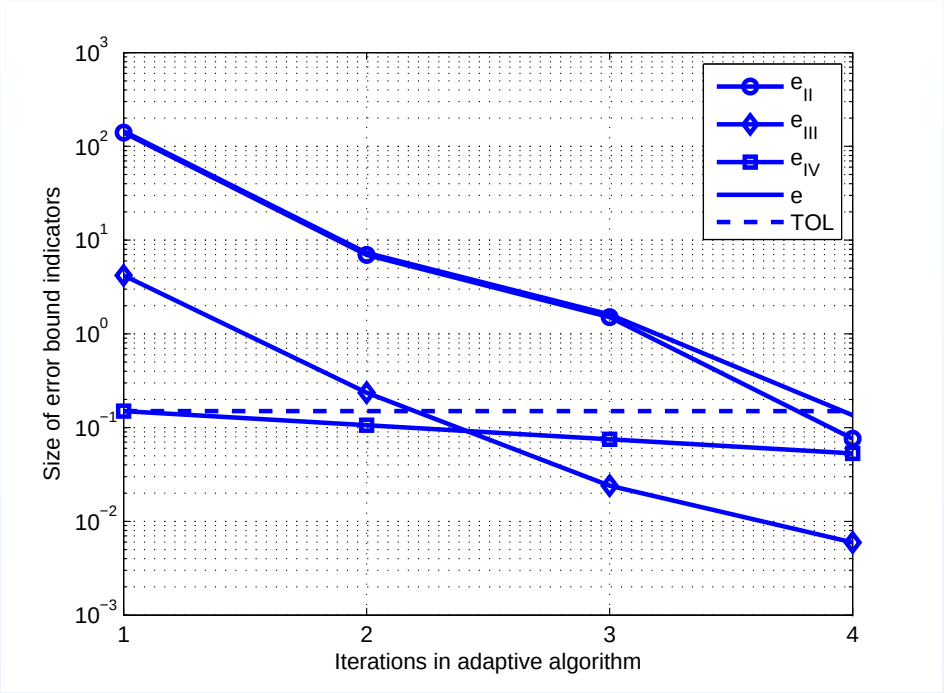
We fix the number of nodes on each of the 27×7 domains to be 5×5 and let $\psi = 1$. Let $I = 100$, $T = 1$, $S = 30$, $\tau = 1.645$ (95% probability), and TOL $= 0.15$.



Since the mesh size is fix in this example h does not appear in the figure. The error tolerance is achieved when $I = 800$, $T = 4$, and $S = 240$.

Numerical example: oil reservoir data

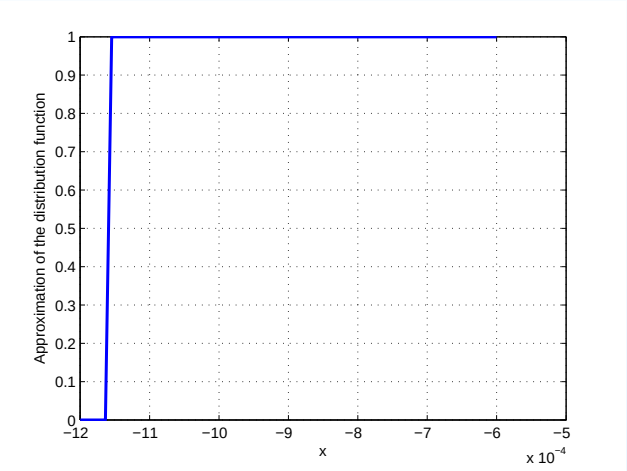
We plot error bound indicators after each iteration in the adaptive algorithm and the total error bound.

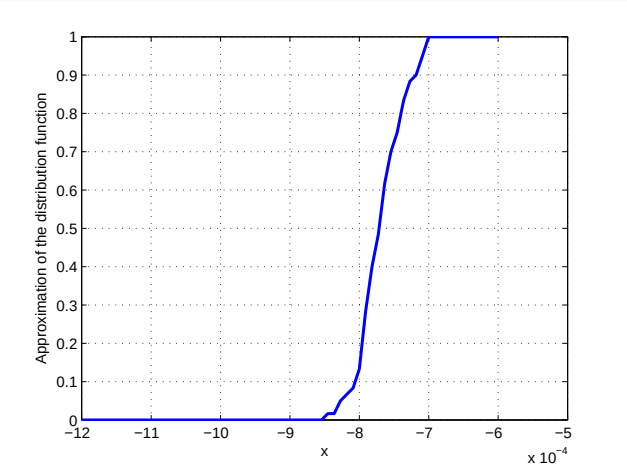


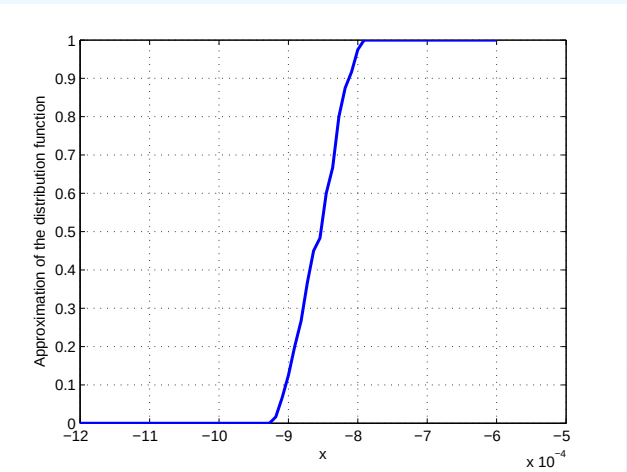
We solve the dual problem using the same parameter values as the primal since we are not interested in refining the mesh.

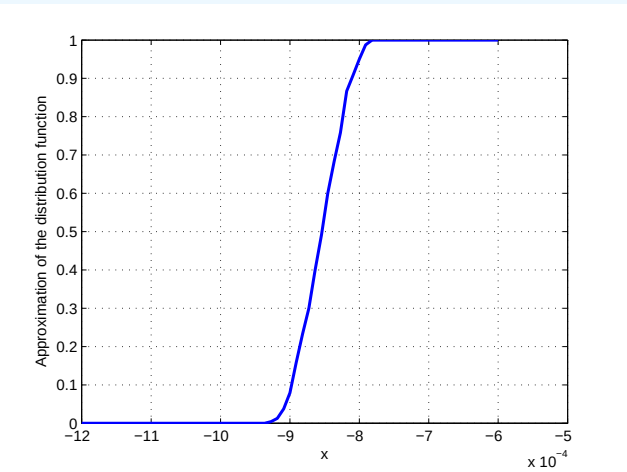
Numerical example: oil reservoir data

We plot the approximation to $F(x)$ after each iteration.









Conclusions

- We present a novel method for cheaply computing samples of the solution to an elliptic problem with randomly perturbed coefficient
- We prove the Neumann series expansion converges
- We present an a posteriori error representation formula and an adaptive algorithm that tunes all critical parameters
- We apply the method to a pressure equation that arises in oil reservoir simulation

Future Work

- The method easily extends to piecewise linear representation of the random perturbation. This is particulary useful when the deterministic part a is continuous. We can then avoid to add artificial low regularity to the solution.
- We will study the error in using coarser representation of the random perturbation locally
- Study more challenging equations such as the transport equation in oil reservoir simulation