Error estimation and adaptive computation for elliptic problems with randomly perturbed coefficients

Axel Målqvist,

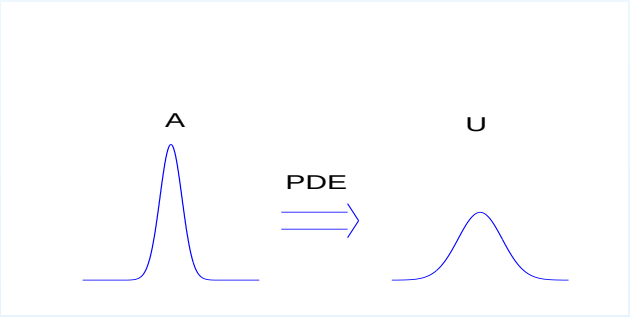
Donald Estep and Simon Tavener

axel@cam.ucsd.edu

Department of Mathematics, University of California, San Diego

Outline

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



Given samples of A the goal is to cheaply compute samples of U.

The model problem

Strong form:

The Poisson Equation with randomly perturbed diffusion coefficient, $\mathcal{A}^s = a + A^s > 0$, a deterministic with multiscale features, A^s are assumed to be piecewise constant random perturbation, $s \in \Lambda$, A^s are iid, $f \in L^2(\Omega)$ is deterministic,

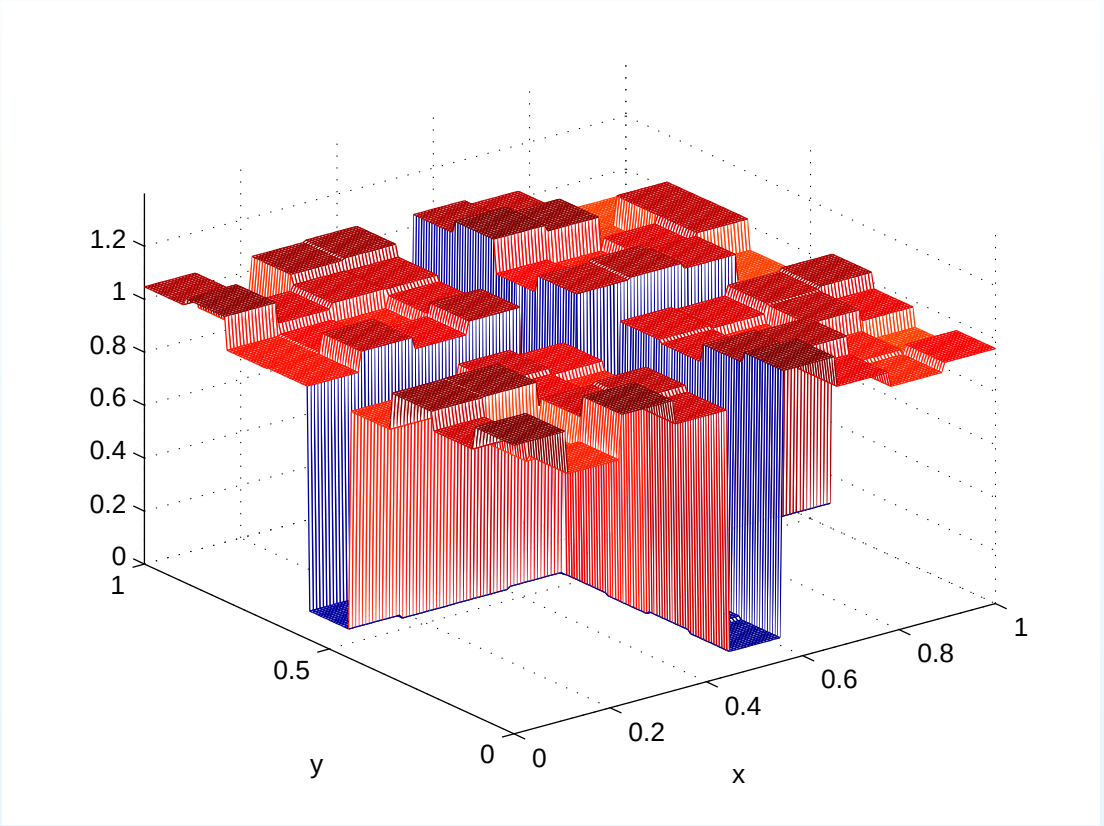
$$-\nabla \cdot \mathcal{A}^s \nabla U^s = f \quad \text{in } \Omega,$$
$$U^s = 0 \quad \text{on } \Gamma.$$

Weak form:

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

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

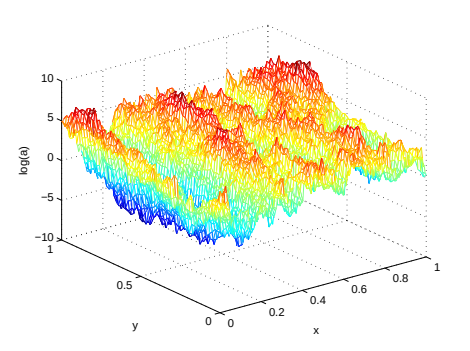
Diffusion coefficient



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

Motivation

- Field data has in most cases measurement errors.
- A natural way to model these errors are as random perturbations of the data.
- The sensitivity in the solution to these perturbations is important to understand if want to be able to rely on the solution.



- 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 $a^* = 1/E[1/\mathcal{A}^s]$. In general there is no simple expression. This means that even finding $E[U^s]$ for this problem is non-trivial.

Montecarlo finite element method

We solve one PDE for each sample of $\mathcal{A}^s$.

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.

We would like to find a method for getting this kind of information to a much lower computational cost.

The main idea

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

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

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

where $K^a_{i,j} = (a \nabla \phi_i, \nabla \phi_j)$, $K_{i,j} = (\nabla \phi_i, \nabla \phi_j)$, and $b_j = (f, \phi_j)$, given a finite element space $V_h = \text{span}(\{\phi_i\}_{i \in \mathcal{N}})$.

- Positive: If the Neumann series convergence quickly we can use a truncated version using a few terms. We only need to invert the matrix in the right hand side once.
- Negative: It will be expensive to solve these systems since a has multiscale features, which means that we will need high resolution to get an accurate solution.

The main idea

However, when A^s (not $\mathcal{A}^s = a + A^s$) is piecewise constant we can use a non-overlapping domain decomposition algorithm with domains that coincide with the regions where A^s are constant.

- If the Neumann series convergence quickly we can still compute samples of the solution by just multiplying and adding random numbers and vectors, now individually on the domains.
- We only need to invert the matrices on each domain which is much cheeper.
- Any multiscale problem need to be solved with some kind of parallel algorithm and Domain Decomposition is one of the most commonly used so this choice is not a huge limitation.

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.

Montecarlo with Lions domain decomposition solver

We will derive the method in three steps starting from the montecarlo finite element method. First we introduce Lions non-overlapping domain decomposition as the solver.

for s from 1 to S **do**
  for i from 1 to I **do**
    for d from 1 to D **do**
      Compute $U^{s,d}_{(i)} = (K^a + A^{s,d}K)^{-1} b^s(f, \mathcal{A}^s, U^s_{(i-1)})$.
    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.

Montecarlo with domain decomposition: 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 $U_{(i)}^{s,d} = (K^a + A^{s,d}K)^{-1} b^s(f, \mathcal{A}^s, U_{(i-1)}^s)$
    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 $C^t = (K^t (K^a)^{-1})^t$
    end for
    for s from 1 to S **do**
      $U^{s,d}_{(i)} \approx \sum_{t=0}^{T-1} (-A^{s,d})^t C^t b^s (f, \mathcal{A}^s, U^s_{(i-1)})$
    end for
  end for
end for

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,

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

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

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

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

Proof of (i)

Let $\delta = A^{s,d}$ and $L = -\delta(K^a)^{-1}K : V_h^d \to V_h^d$ and $z = Lx$ for an arbitrary $x \in V_h^d$. Then z solves the following problem: find $z \in V_h^d$ such that,

$$(\frac{a}{\delta}\nabla z, \nabla v) + \frac{1}{\delta\lambda}\langle z, v\rangle = -(\nabla x, \nabla v),$$

for all $v \in V_h^d$. If we choose $v = z$ we get,
$\frac{1}{\delta}\|\sqrt{a}\nabla z\|_{\Omega_d}^2 + \frac{1}{\delta\lambda}\|z\|_{\partial\Omega_d}^2 = -(\nabla x, \nabla z)$. If we use that
$a^* = \min_{x\in\Omega_d} a \leq a$ and use the Cauchy-Schwartz inequality we further have,

$$\frac{a^*}{\delta}\|\nabla z\|_{\Omega_d}^2 + \frac{1}{\delta\lambda}\|z\|_{\partial\Omega_d}^2 \leq \|\nabla x\|_{\Omega_d}\|\nabla z\|_{\Omega_d}.$$

We define $\|z\|_{V_h^d}^2 = \|\nabla z\|_{\Omega_d}^2 + \epsilon\|z\|_{\partial\Omega_d}^2$ and choose $\epsilon = 2/(\lambda a^*)$,

$$\|z\|_{V_h^d}^2 = \|\nabla z\|_{\Omega_d}^2 + \frac{2}{\lambda a^*}\|z\|_{\partial\Omega_d}^2 \leq \left(\frac{\delta}{a^*}\right)^2 \|\nabla x\|_{\Omega_d}^2 \leq \left(\frac{\delta}{a^*}\right)^2 \|x\|_{V_h^d}^2.$$

Proof of (i)

This means that,

$$\|Lx\|_{V_h^d} \leq \frac{\delta}{a^*}\|x\|_{V_h^d}.$$

Using the same argument on Lz instead of Lx we get,

$$\|L^2x\|_{V_h^d} = \|Lz\|_{V_h^d} \leq \frac{\delta}{a^*}\|z\|_{V_h^d} \leq \left(\frac{\delta}{a^*}\right)^2 \|x\|_{V_h^d},$$

and further by induction, $\|L^tx\|_{V_h^d} \leq \left(\frac{\delta}{a^*}\right)^t \|x\|_{V_h^d}$. In the corresponding operator norm,

$$(i) \qquad \|L^t\|_{V_h^d, V_h^d} = \sup_{\|w\|_{V_h^d}=1} \|L^tw\|_{V_h^d} \leq \left(\frac{\delta}{a^*}\right)^t.$$

Norms on finite dimensional spaces are equivalent, $\|\cdot\| \leq C\|\cdot\|_{V_h^d, V_h^d}$.

Proof of (ii) and (iii)

For part (ii) we start with an identity,

$$I - L^T = (I - L) \sum_{t=0}^{T-1} L^t.$$

We know from (i) that $\lim_{t\to\infty} L^t = 0$ since $\lim_{t\to\infty} \|L^t\|_{V_h^d, V_h^d} \leq \lim_{t\to\infty} (\frac{\delta}{a^*})^t = 0$. If we take the limit as T tends to infinity we end up with, $I = (I - L) \sum_{t=0}^{\infty} L^t$, or

$$(ii) \qquad (I - L)^{-1} = \sum_{t=0}^{\infty} L^t.$$

Given (ii) statement (iii) follows easily since,

$$(iii) \qquad (I - L)^{-1} - \sum_{t=0}^{T-1} L^t = \sum_{t=T}^{\infty} L^t = L^T \sum_{t=0}^{\infty} L^t = L^T (I - L)^{-1}.$$

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 five 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.
5. Modeling error when assuming the random perturbation to be piecewise constant (D).

The goal is to equidistribute the error between these components.

Duality based a posteriori error analysis

We start by studying the first three types of error, (h, I, T). Let U^s be the exact solution and let $U^s_{h,I,T}$ be the approximate solution for each s.

We construct the corresponding dual problem,

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

where ψ is deterministic. We construct this problem in order to get estimates of $(U^s - U^s_{h,I,T}, \psi)$ for each individual sample, s.

Note that Φ is now also stochastic.

Error representation formula

We introduce some more notations, let $e^s = U^s - U^s_{h,I,T} = U^s_{0,\infty,\infty} - U^s_{h,I,T}$. We proceed with a standard calculation,

$$\begin{aligned}
(e^s, \psi) &= (e^s, -\nabla \cdot \mathcal{A}^s \nabla \Phi^s) \\
&= (f, \Phi^s) - (\mathcal{A}^s \nabla U^s_{h,I,T}, \nabla \Phi^s) \\
&= (f, \Phi^s) - (\mathcal{A}^s \nabla U^s_{h,\infty,\infty}, \nabla \Phi^s) \\
&\quad - (\mathcal{A}^s \nabla (U^s_{h,I,T} - U^s_{h,\infty,\infty}), \nabla \Phi^s) \\
&= (f, \Phi^s) - (\mathcal{A}^s \nabla U^s_{h,\infty,\infty}, \nabla \Phi^s) \\
&\quad + (\mathcal{A}^s \nabla (U^s_{h,\infty,T} - U^s_{h,I,T}), \nabla \Phi^s) \\
&\quad + (\mathcal{A}^s \nabla (U^s_{h,\infty,\infty} - U^s_{h,\infty,T}), \nabla \Phi^s).
\end{aligned}$$

Remember that s is fix which means that we can view the problem as being deterministic which means that the usual approach works.

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

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

First term (h)

The natural thing would be to say,

$$(f,\Phi^s)-(\mathcal{A}^s\nabla U^s_{0,\infty,\infty},\nabla\Phi^s)\approx(f,\Phi^s_{h',T',I'})-(\mathcal{A}^s\nabla U^s_{h,I,T},\nabla\Phi^s_{h',T',I'}).$$

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,

$$\begin{aligned}(f,\Phi^s)-(\mathcal{A}^s\nabla U^s_{0,\infty,\infty},\nabla\Phi^s)\approx{}&(f,\Phi^s_{h',T',I'})-(\mathcal{A}^s\nabla U^s_{h,I,T},\nabla\Phi^s_{h',T',I'})\\&-(\mathcal{A}^s\nabla(U^s_{h,I+\Delta I,T}-U^s_{h,I,T}),\nabla\Phi^s_{h',I',T'})\\&-(\mathcal{A}^s\nabla(-A^{s,d})^T((K^a)^{-1}K)^TU^s_{h,I+\Delta I,T},\nabla\Phi^s_{h',I',T'}).\end{aligned}$$

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.

Fourth error type (S)

Now we have access to $\{(U^s_{h,I,T}, \psi)\}_{s=1}^S$ and estimates of $\{(e^s, \psi)\}_{s=1}^S$.

We want to estimate the error in a stochastic quantity and we choose the distribution function $F(x)$,

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

where Λ is the entire distribution of samples. In order to estimate this error we introduce the notation $e_M = \sup_{s\in\Lambda} |(e^s, \psi)|$ and assume that S is large enough for the Central Limit Theorem to be valid. Then,

$$|F(x) - \tilde{F}_s(x)| \leq \tau \sqrt{\frac{\tilde{F}(x)(1-\tilde{F}(x))}{S}} + e_M \max_{y\in[x-e_M, x+e_M]} \tilde{f}(y),$$

with probability $\int_{-\infty}^{\tau} e^{-t^2/2}\, dt/\sqrt{2\pi}$, where $\tilde{F}(x) = P(\{(U^s_{h,I,T})\}_{s\in\Lambda} < x)$ and $\tilde{F}(x) = \int_{-\infty}^{x} \tilde{f}(x)\, dx$.

Proof of the estimate of the stochastic error

We use the triangle inequality, $|F(x) - \tilde{F}_S(x)| \leq |F(x) - \tilde{F}(x)| + |\tilde{F}(x) - \tilde{F}_S(x)|$, and start by studying the second term in the right hand side. Let,

$$X^s(x) = \begin{cases} 1 & \text{if } (U^s_{h,I,T}, \psi) < x, \\ 0 & \text{otherwise.} \end{cases}$$

This means that, $\tilde{F}_S(x) = P(\{(U^s_{h,I,T}, \psi)\}_{s=1}^S < x) = \frac{1}{S}\sum_{s=1}^S X^s(x)$, and, $\tilde{F}(x) = P(\{(U^s_{h,I,T}, \psi)\}_{s\in\Lambda} < x) = E[X](x)$. For fix x we use CLT to get the following bound which is valid for large numbers S,

$$P\left(\frac{1}{\sigma(X)\sqrt{S}}\sum_{s=1}^{S}(X^s - E[X]) \leq \tau\right) \approx \frac{1}{2\pi}\int_{-\infty}^{\tau} e^{-t^2/2}\,dt.$$

Further since X has binomial distribution we know that, $\sigma^2(X)(x) = \tilde{F}(x)(1 - \tilde{F}(x))$.

Proof of estimate of stochastic error

$$P\left(|\tilde{F}(x) - \tilde{F}_S(x)| \leq \tau\sqrt{\frac{\tilde{F}(x)(1-\tilde{F}(x))}{S}}\right) \approx \frac{1}{2\pi}\int_{-\infty}^{\tau} e^{-t^2/2}\,dt.$$

We now return to the first term ($|F(x) - \tilde{F}(x)|$). We have,

$$\begin{aligned}
F(x) - \tilde{F}(x) &= P(\{(U^s,\psi)\}_{s\in\Lambda} < x) - P(\{(U^s_{h,I,T},\psi)\}_{s\in\Lambda} < x) \\
&= P(\{(U^s_{h,I,T},\psi) + (e^s,\psi)\}_{s\in\Lambda} < x) - P(\{(U^s_{h,I,T},\psi)\}_{s\in\Lambda} < x) \\
&\leq P(\{(U^s_{h,I,T},\psi)\}_{s\in\Lambda} < x + e_M) - P(\{(U^s_{h,I,T},\psi)\}_{s\in\Lambda} < x) \\
&= \int_x^{x+e_M} \tilde{f}(y)\,dy \leq e_M \max_{y\in[x,x+e_M]} \tilde{f}(y).
\end{aligned}$$

We also have, $\tilde{F}(x) - F(x) \leq e_M \max_{y\in[x-e_M,x]} \tilde{f}(y)$. We finally get,

$$|F(x) - \tilde{F}(x)| \leq e_M \max_{y\in[x-e_M,x+e_M]} \tilde{f}(y).$$

The a posteriori estimate

With probability $\int_{-\infty}^{\tau} e^{-t^2/2}\,dt/\sqrt{2\pi}$ we have,

$$|F(x) - \tilde{F}_S(x)| \leq \tau\sqrt{\frac{\tilde{F}(x)(1-\tilde{F}_s(x))}{S}} + e_M \cdot \max_{y\in[x-e_M,x+e_M]} \tilde{f}(y),$$

where $e_M = \sup_{s\in\Lambda} |(e_s, \psi)|$. However, in practise we are going to use an approximation,

$$|F(x) - \tilde{F}_S(x)| \leq \tau\sqrt{\frac{\tilde{F}_S(x)(1-\tilde{F}_S(x))}{S}} + e_m \cdot \tilde{F}'_S(x),$$

where $e_m = \max_{s=1,\ldots,S} |(e^s, \psi)|$, which we have an estimate of given our three error indicators for each sample $s \in 1, \ldots, S$, and $\tilde{F}'_S(x)$ will be computed using a central difference approximation.

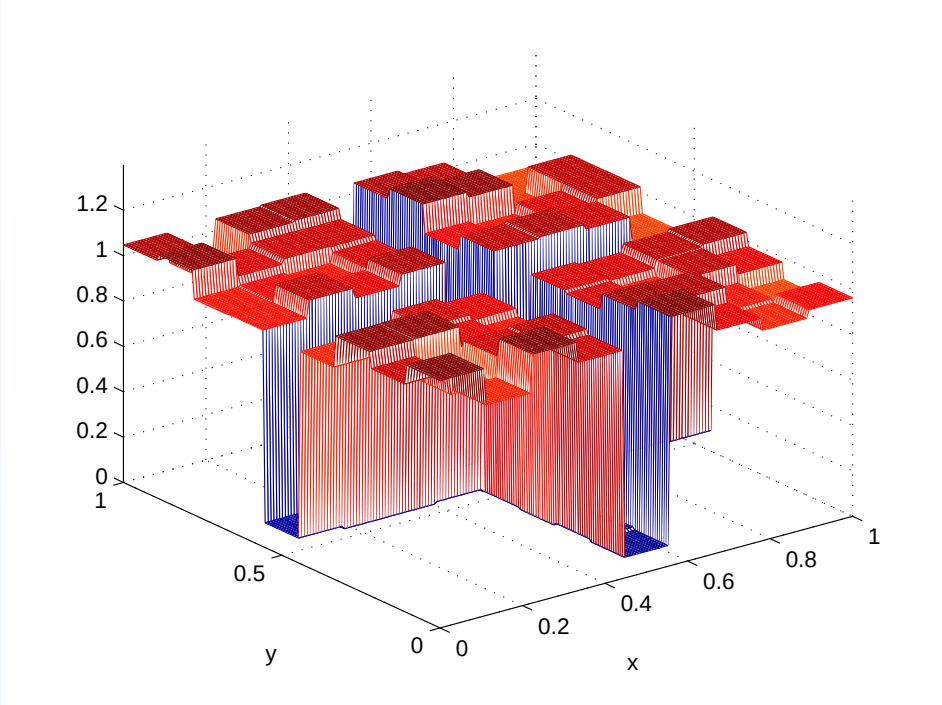
Given this estimate we can present an adaptive algorithm.

Adaptive algorithm

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

Numerical examples

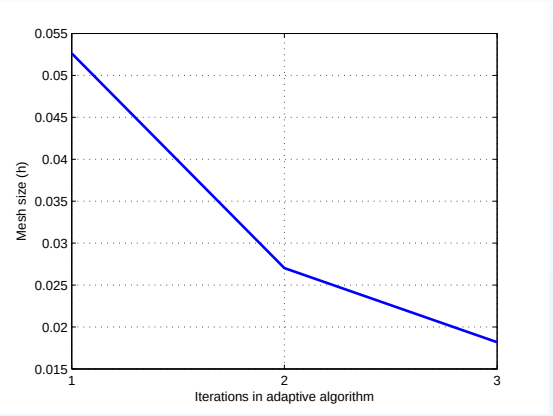
We partition the domain into 9×9 subdomains for the domain decomposition algorithm and let A^s be piecewise constant on the partition.

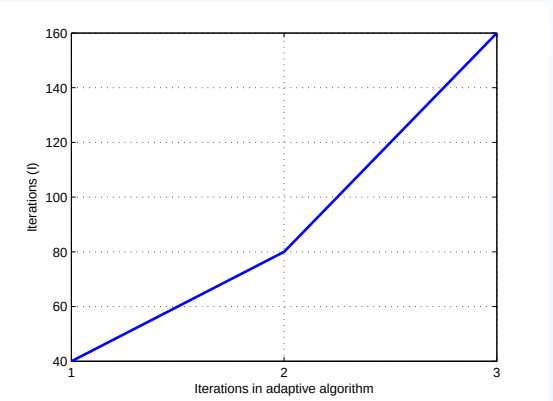


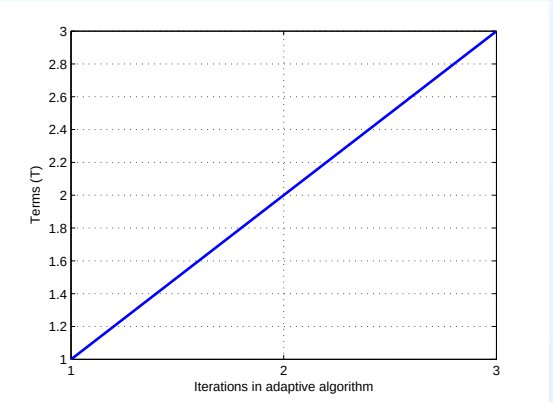
The maximum size of the perturbation A^s is 10% of the value in a. We let $f = \psi = 1$.

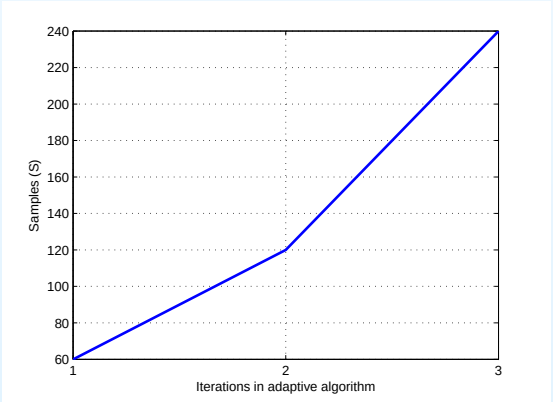
Numerical example: adaptivity

We start with $h = 1/18$, $I = 40$, $T = 1$, and $S = 60$ and let the adaptive algorithm choose refine the parameters. Let TOL $= 0.15$ and $\tau = 1.645$ which means 15% error with 95% probability.



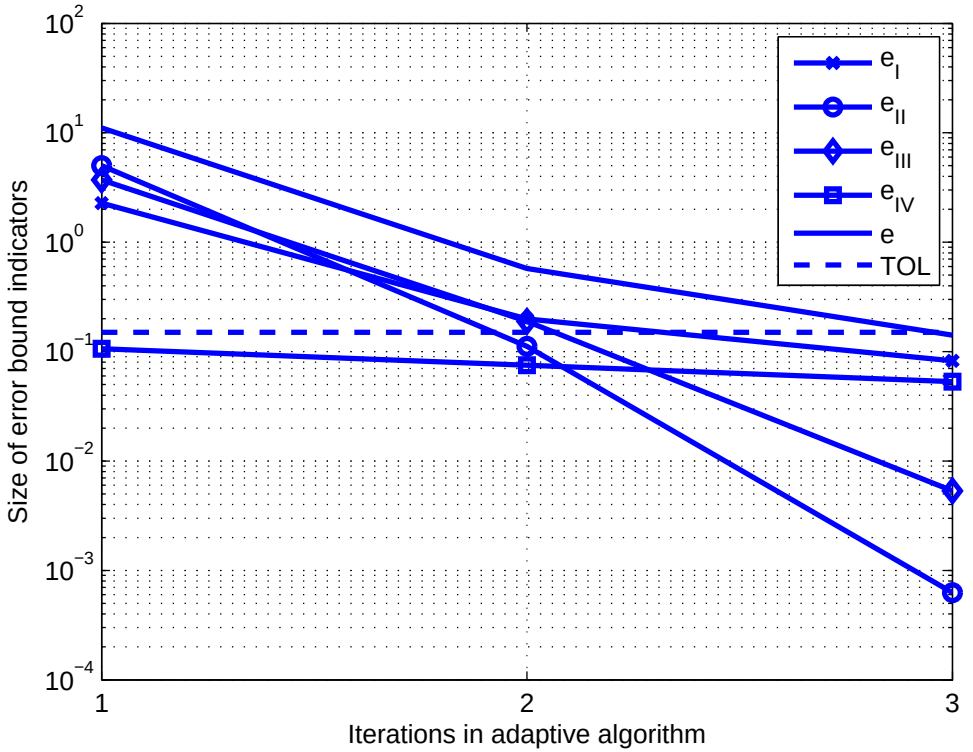






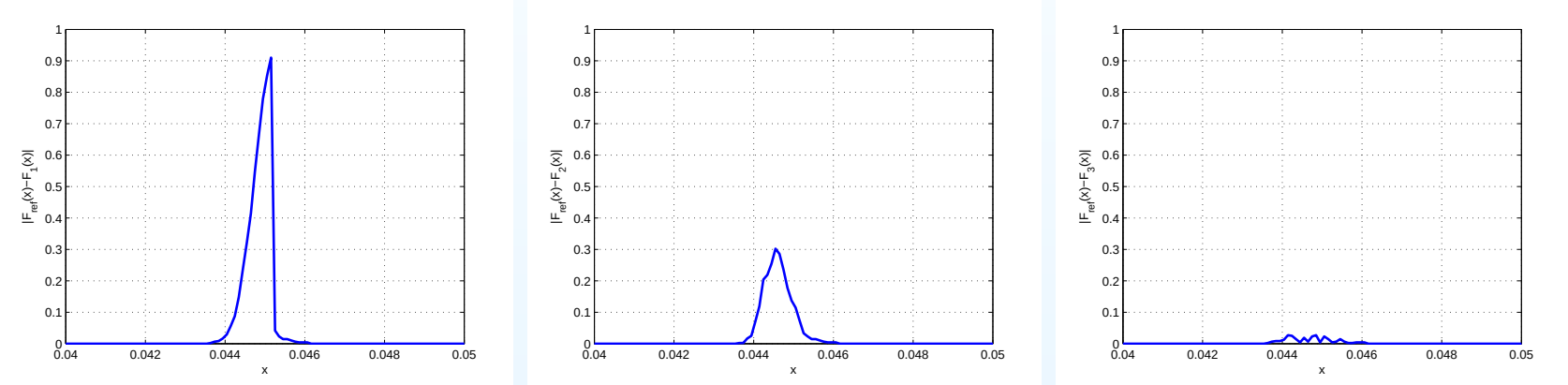
Numerical example: adaptivity

Here we refine if $e_{\mathrm{I}} < \mathrm{TOL}/2$, $e_{\mathrm{II}} < \mathrm{TOL}/8$, $e_{\mathrm{III}} < \mathrm{TOL}/8$, and $e_{\mathrm{IV}} < \mathrm{TOL}/4$ since it is cheaper to improve the second and the third error term compared to the first and the fourth.



Numerical example: adaptivity

We compare the computed version of $F(x)$ with the reference solution, i.e. $|\tilde{F}_S(x) - \tilde{F}_{\text{ref}}(x)|$ ($h_{\text{ref}} = 1/72$, $I_{\text{ref}} = 300$, $T_{\text{ref}} = 5$, and $S_{\text{ref}} = 480$) we get,



We see that the error goes from almost 100%, which means that we have missed the critical area completely to an error of around 30% to finally an error less then 3%.

Numerical example: oil reservoir data

We now study a more realistic problem using boundary conditions that arise frequently in simulations of oil reservoirs.

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

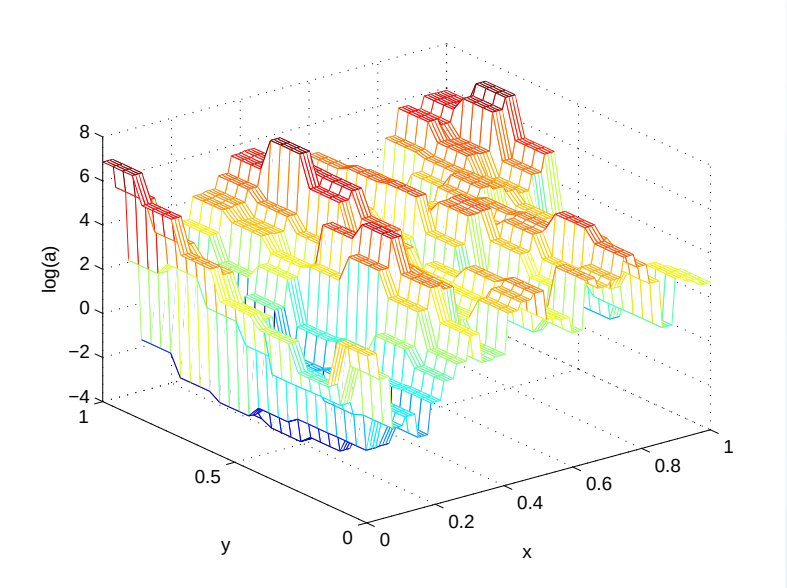
where $\Gamma_N \cup \Gamma_D = \Gamma$. Here U^s represents the pressure field, and a is the local permeability.

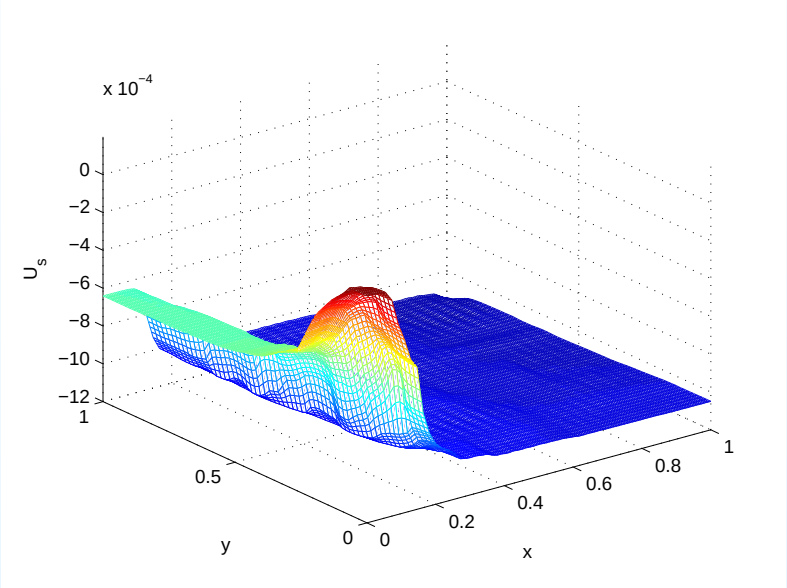
We have choosen Let $f = 1$ in the lower left corner, the injector, and $f = -1$ in the upper right corner, the producer.

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

Numerical example: oil reservoir data

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





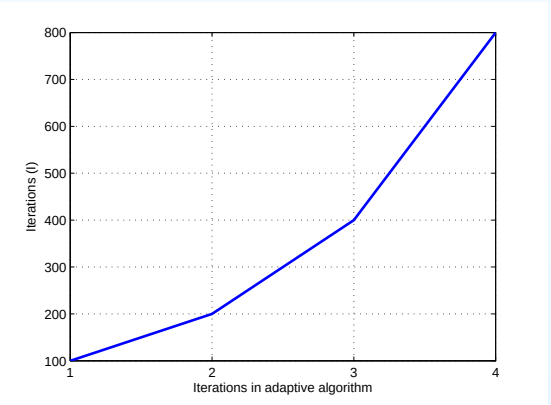
We add a random perturbation to a (20% of the magnitude of a). To the right: a typical solution U^s.

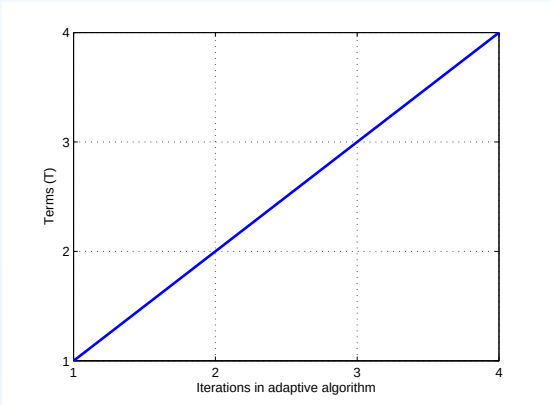
The band of low permeability at $x \approx 0.2$ creates a large pressure drop parallel to the y-axis at this location.

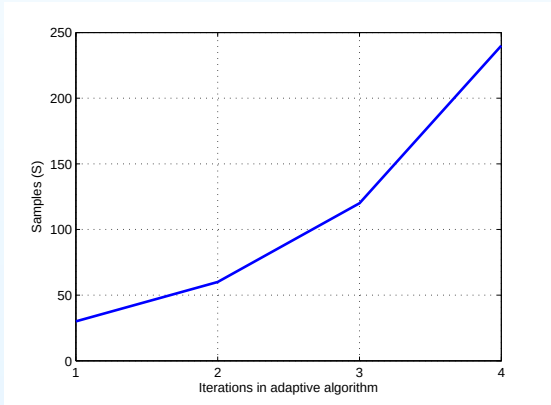
Numerical example: oil reservoir data

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

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



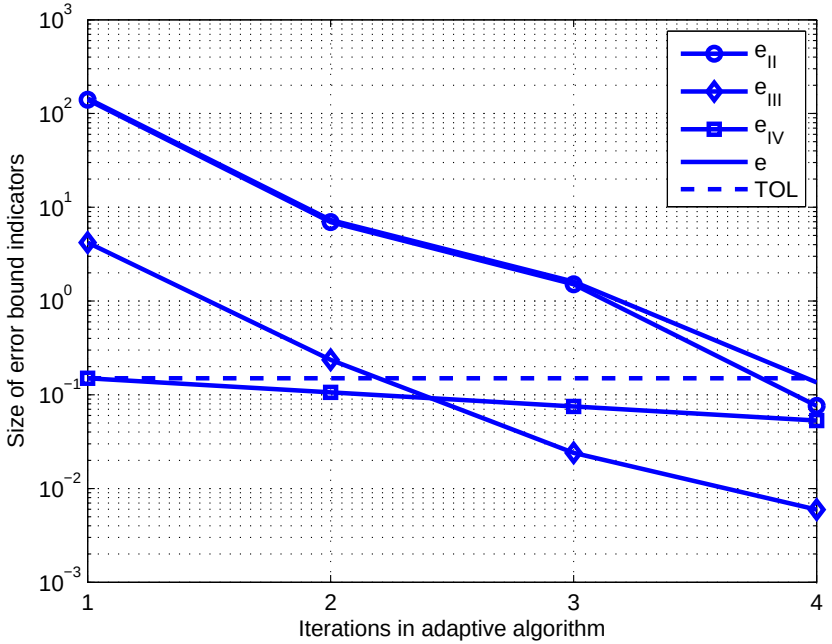




Since the mesh size is fix in this example it does not appear in the figure. The error tolerance is achieved when $I = 800$, $T = 4$, and $S = 240$.

Numerical example: oil reservoir data

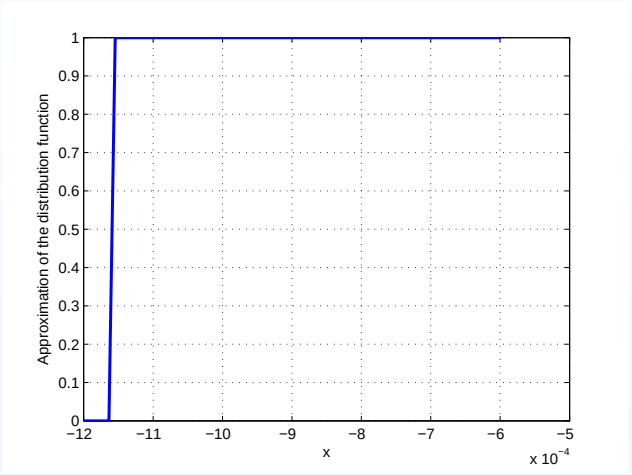
We plot error bound indicators after each iteration in the adaptive algorithm and the total error bound.

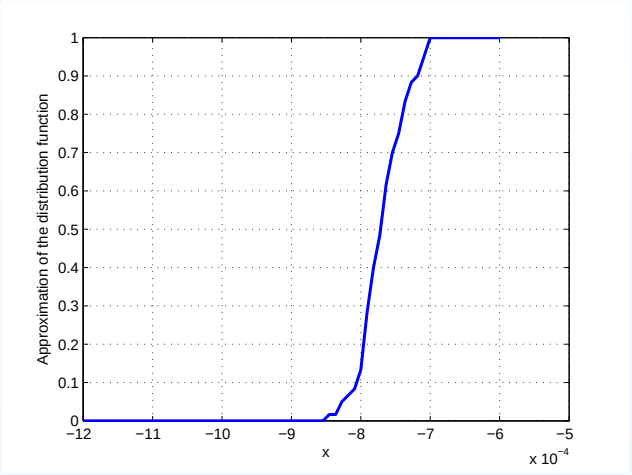


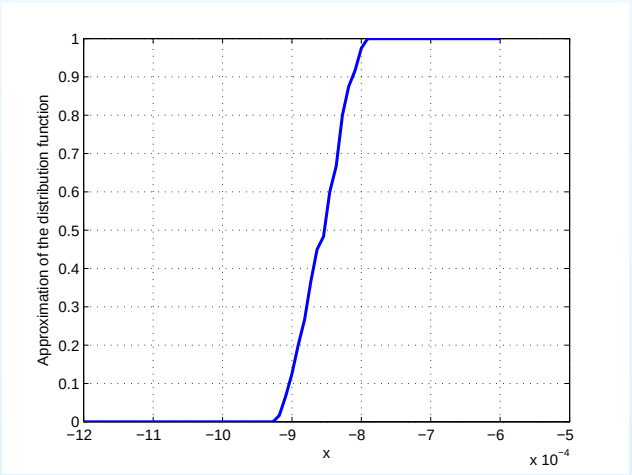
We solve the dual problem using the same mesh as the primal since we are not interested in refining the mesh. The number of iterations, terms, and samples is the same for the primal and the dual.

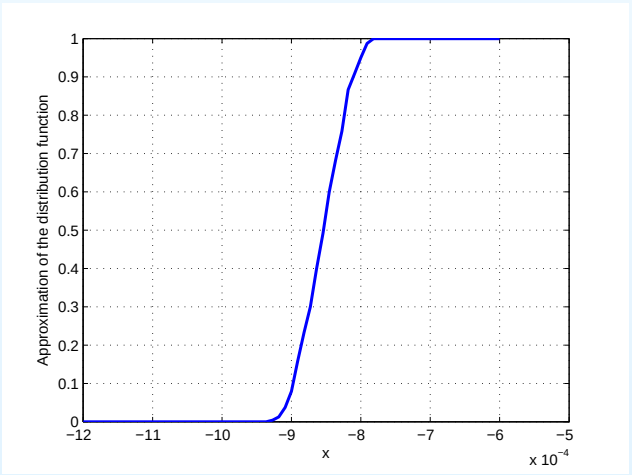
Numerical example: oil reservoir data

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









The fifth type of error

- So far we have neglected the fifth error type arising from the fact that we approximate the random perturbation as a piecewise constant function in space.
- Now instead we Assume that we have access to a very refined version of $\mathcal{A}^s$ but we only want to use a piecewise constant approximation $\bar{\mathcal{A}}^s$.
- Using to few domains in the dd algorithm means that the matrix we need to invert on each domain becomes huge. This can be avoided by choosing more domains in regions where $\mathcal{A}^s$ is constant.
- On the other hand since all computation in the inner loop (over s) are done on the boundary of the domains we want to minimize the length of the interior boundaries i.e. the number of domains. Thats why we need to avoid using to more domains, i.e. better representation of $\mathcal{A}^s$, then necessary. This is important both for efficiency and storage.

Modified a posteriori error representation formula

Let s be fix and assume we use an approximation to the correct perturbation, $\bar{A}^s \approx A^s$ defined as being piecewise constant on a partition on Ω, when we compute $U^s_{h,I,T,D}$. We have,

$$
\begin{aligned}
(e^s, \psi) &= (f, \Phi^s) - (\mathcal{A}^s \nabla U^s_{h,I,t,D}, \nabla \Phi^s) \\
&= (f, \Phi^s) - ((a + \bar{A}^s) \nabla U^s_{h,I,t,D}, \nabla \Phi^s) + ((\bar{A}^s - A^s) \nabla U^s_{h,I,T,D}, \nabla \Phi^s) \\
&= (f, \Phi^s) - ((a + \bar{A}^s) \nabla U^s_{h,I,t,D}, \nabla \Phi^s) + \sum_{d=1}^{D} ((\bar{A}^{s,d} - A^{s,d}) \nabla U^s_{h,I,T,D}, \nabla \Phi^s)_d.
\end{aligned}
$$

The first term can be approximated with the three error indicators already discussed, using the approximate perturbation. The sum will measure the effect of not using the correct random perturbation.

$$
\sum_{d=1}^{D} ((\bar{A}^{s,d} - A^{s,d}) \nabla U^s_{h,I,T,D}, \nabla \Phi^s)_d \approx \sum_{d=1}^{D} ((\bar{A}^{s,d} - A^{s,d}) \nabla U^s_{h,I,T,D}, \nabla \Phi^s_{h',I',T',D})_d
$$

Modified a posteriori error representation formula

By equidistributing the error on the five terms and then equidistribute the error in the fifth term between the domains we get a local indicator on how many sub domains we need in each region for the domain decomposition algorithm.

Remember,

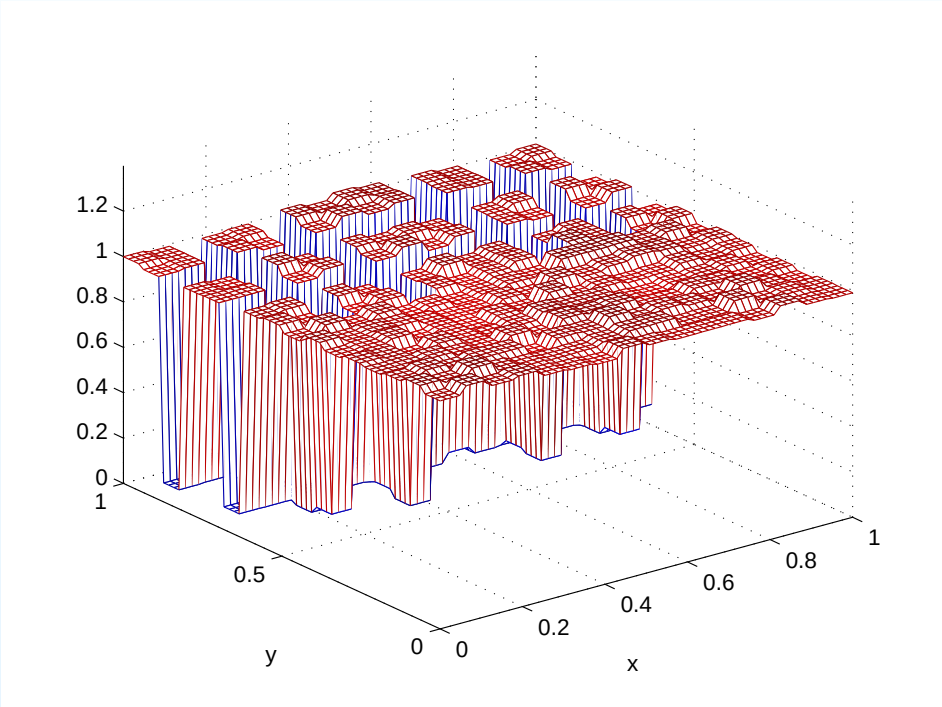
$$|F(x) - \tilde{F}_S(x)| \leq \tau \sqrt{\frac{\tilde{F}_S(x)(1 - \tilde{F}_s(x))}{S}} + e_m \cdot \tilde{F}'_S(x),$$

where $e_m = \max_{s=1,\ldots,S} |(e^s, \psi)|$.

Now $|(e^s, \psi)|$ will have one more contribution.

Numerical example: stochastic modeling error

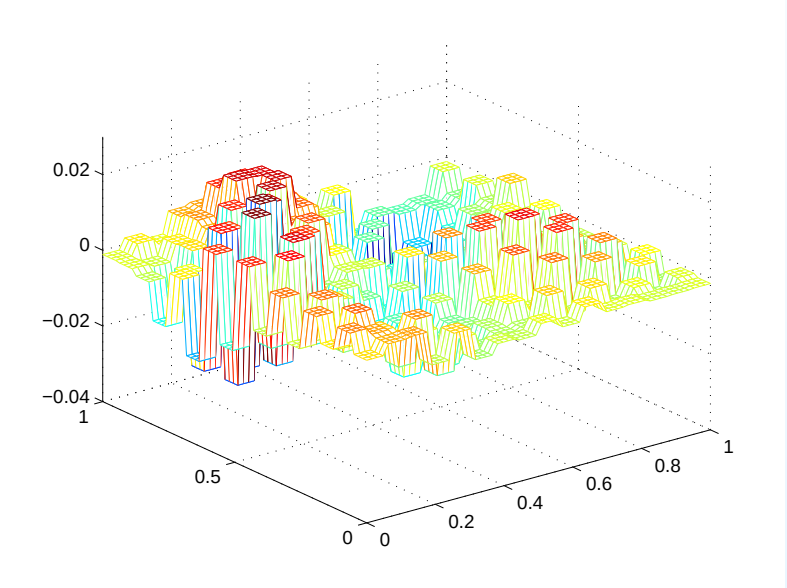
We let the random perturbation have 10% of the magnitude of a and plot a typical sample of $\mathcal{A}^s$.

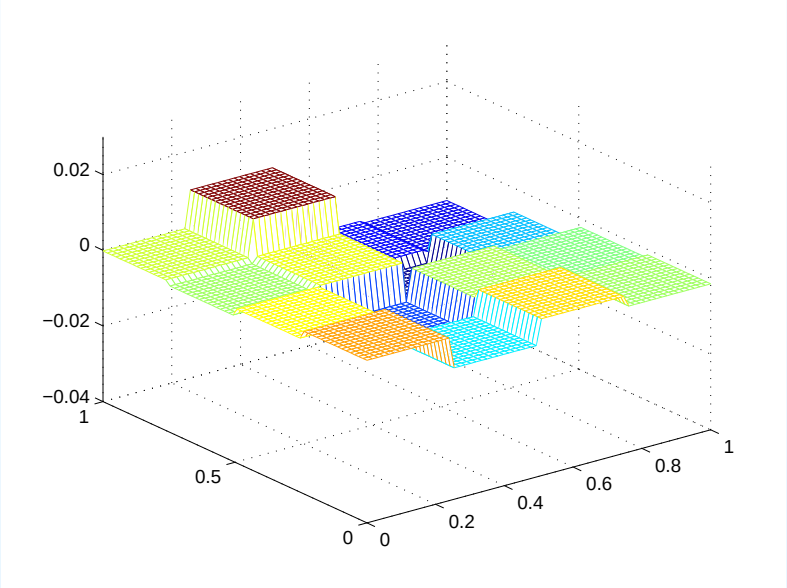


We assume the exact representation of the random perturbation is has 16×16 values. The dual load $\psi = 1$, $0.9 \leq x, y \leq 1$, and $\psi = 0$ otherwise, $f = 1$.

Numerical example: stochastic modeling error

We use a coarse (4×4) representation of the random perturbation by averaging A^s over the regions.

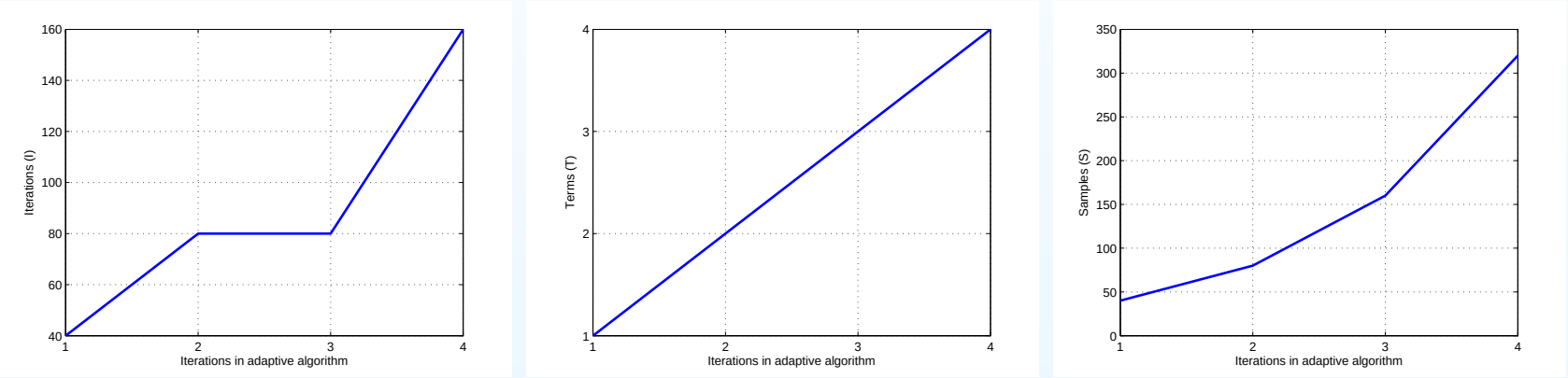




We assume some (weak) correlation in the measurement errors. If we do not have any correlation only full and none refinement will be considered.

Numerical example: stochastic modeling error

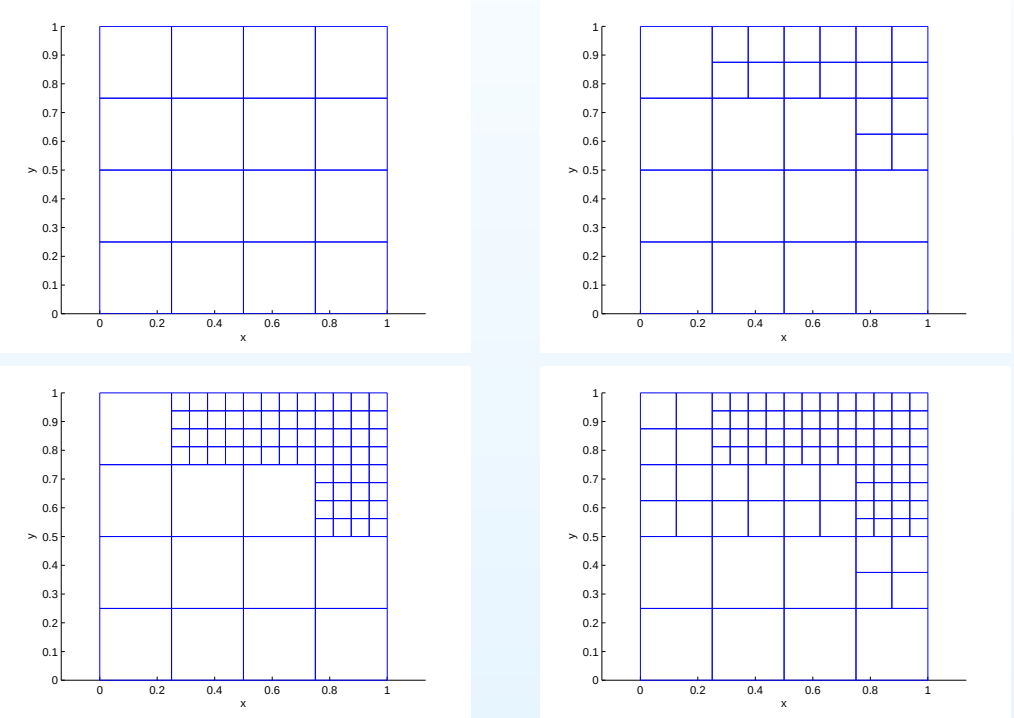
We let $h = 1/32$, $I = 40$, $T = 1$, $S = 40$, TOL $= 0.1$, and $\tau = 1.645$. We compute the dual solution using the same mesh as the primal. The adaptive algorithm chooses the following parameter values.



The tolerance was reached after three iterations with $I = 160$, $T = 4$, and $S = 320$.

Numerical example: stochastic modeling error

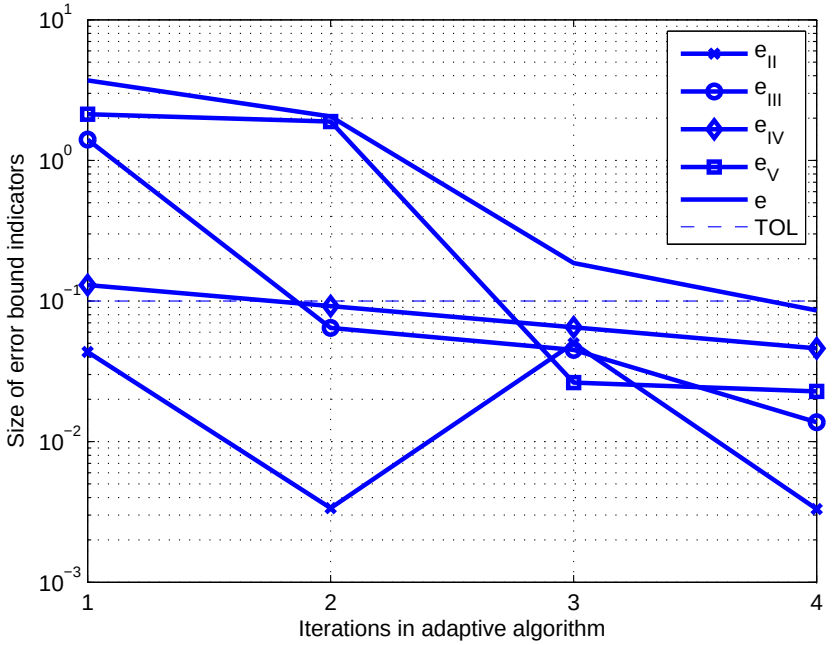
We plot a representation of the random perturbation after each step in the adaptive algorithm.



Remember $\psi = 1$ in the upper right corner and a was complicated in the upper half of the domain.

Numerical example: stochastic modeling error

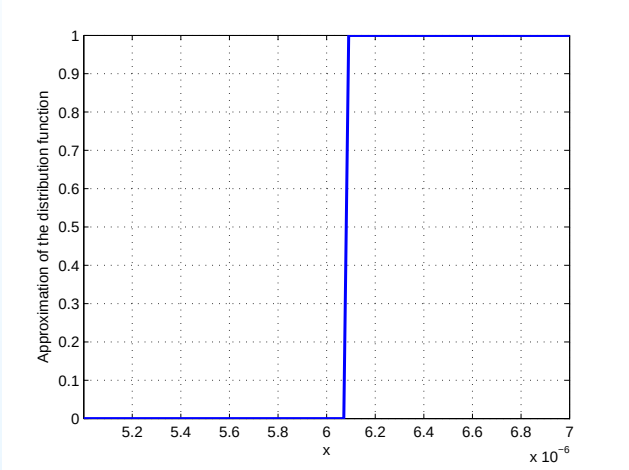
We plot error bound indicators after each iteration in the adaptive algorithm and the total error bound.

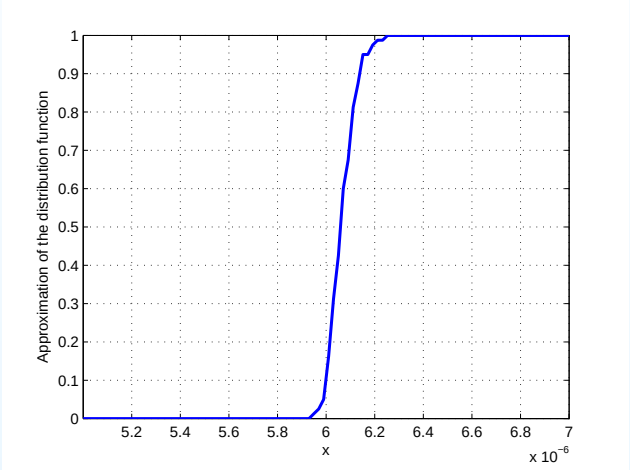


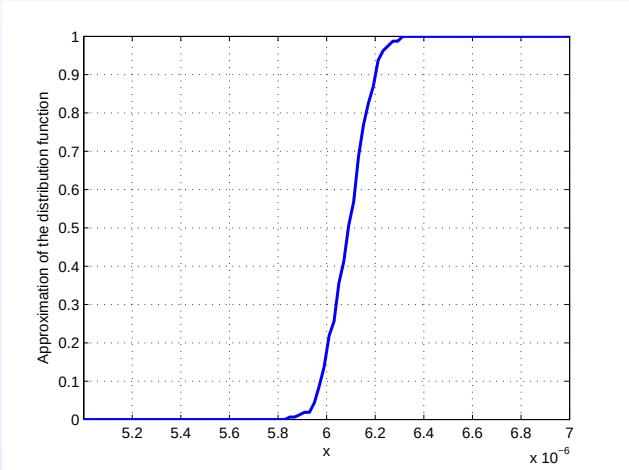
Between iteration 2 and 3 the number of iterations in the domain decomposition stays the same. At the same time the size of the smallest domain decreases by a factor two.

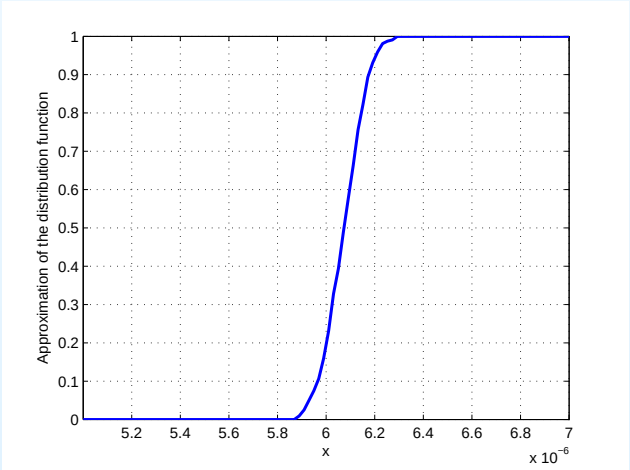
Numerical example: stochastic modeling error

We get the following distribution functions after each of the four iterations in the adaptive algorithm.



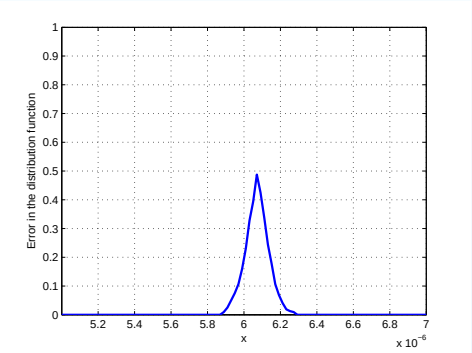


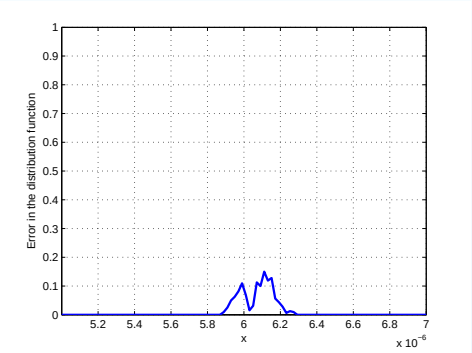


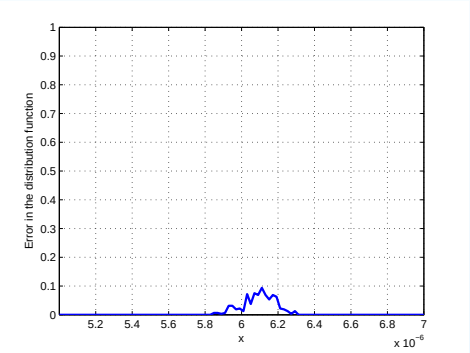


Numerical example: stochastic modeling error

If we let the last iterate in the adaptive algorithm serve as reference solution we get the following errors after the three first iterations.







In areas where we use a coarse representation of the random perturbation we almost replace it by its mean. We know that this is not ok in general (e.g. for constant perturbation the harmonic mean is accurate).

However, we only do this in regions with tiny impact on the desired output quantity. In crucial regions we resolve the perturbation. We detect these areas in the adaptive algorithm.

Conclusions and future work

- 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 which is a crucial result for the method
- We present an a posteriori error representation formula and an adaptive algorithm that tunes all parameters
- We discuss stochastic modeling error and include this in the adaptive algorithm.
- We apply the method to various problems including a model problem in oil reservoir simulation