Nonparametric density estimation for elliptic problems with random perturbations

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

axel.malqvist@it.uu.se

Colorado State University and Uppsala University

Outline

Given perturbations of a coefficient in a PDE, the goal is to compute the distribution function of a quantity of interest cheaply, without assuming a certain kind of distribution, i.e. non-parametric density estimation.

- A model problem with randomly perturbed coefficient
- Methods for computing samples of the solution
- A posteriori error representation formula and adaptivity
- Numerical examples
- Future work

Poisson equation with randomly perturbed coefficient

Strong form:

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

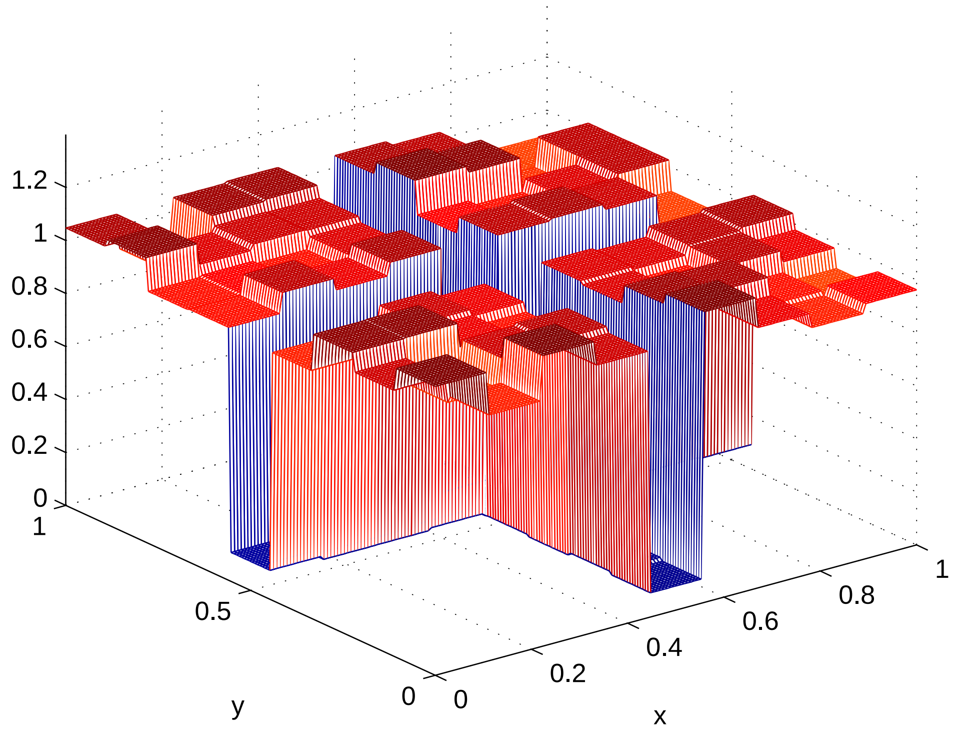
- We assume that $\mathcal{A}^s = a + A^s \geq \alpha > 0$,
- that a is deterministic with multiscale features,
- that A^s are piecewise constant, iid, random perturbations.
- and that $f \in L^2(\Omega)$ is deterministic.

Weak form:

For each s, find $u^s \in H_0^1(\Omega)$ such that,

$$(\mathcal{A}^s \nabla u^s, \nabla v) = (f, v) \quad \text{for all } v \in H_0^1(\Omega).$$

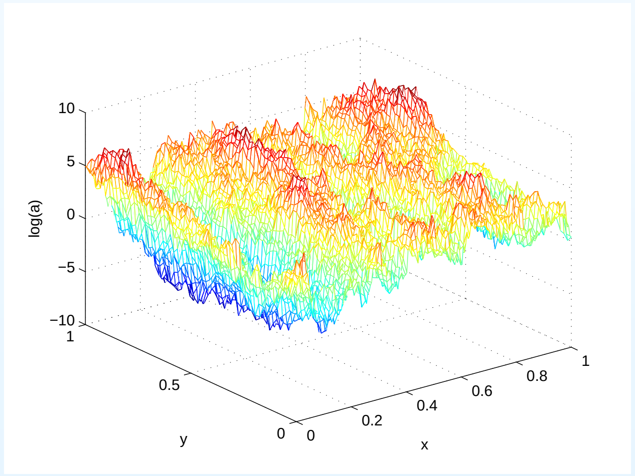
Diffusion coefficient



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

Motivation

- There is often measurement errors in field data.
- These errors can be modelled as random perturbations.
- Sensitivity in the solution to these perturbations is important to understand in order to get a reliable simulation.



Engineers at e.g. Sandia and in oil reservoir simulation are interested in this kind of sensitivity analysis.

Monte Carlo finite element method

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

```
**for** $s$ from $1$ to $S$ **do**
  $\mathcal{A}^s = a + A^s$
  $u^s =$ solver$(f, \mathcal{A}^s)$
**end for**
```

- 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 high resolution is necessary.

Want same kind of information to much lower cost.

Improvements in the proposed Monte Carlo algorithm

1. Since a has multiscale features it is preferable to split the problem into smaller subproblems that can be solved in parallel.
   - Domain Decomposition, see overview in Smith et.al. 96.
   - Adaptive varaiational multiscale methods, see Larson et.al. 07, Nordbotten 08.
   - Multiscale finite element method, Hou et.al. 97, Aarnes et.al. 07.
2. We should take advantage of the fact that the subproblems look very similar for all samples of A^s.

$$[K_{\text{loc}}(a) + K_{\text{loc}}(A^s)]u^s_{\text{loc}} = b_{\text{loc}}(f, A^s, \dots),$$

where $K_{\text{loc}}(g) = (g\nabla\varphi_j, \nabla\varphi_i)_{\text{loc}}$ together with a boundary term if applicable.

Truncated Neumann series

We assume that the subdomains are chosen so that the perturbation is piecewise constant on them,

$$\begin{aligned}[K_{\mathsf{loc}}(a) + A^s_{\mathsf{loc}} K_{\mathsf{loc}}(1)]^{-1} &= [I + A^s_{\mathsf{loc}} K_{\mathsf{loc}}(a)^{-1} K_{\mathsf{loc}}(1)]^{-1} K_{\mathsf{loc}}(a)^{-1} \\ &= \sum_{t=0}^{\infty} [-A^s_{\mathsf{loc}} K_{\mathsf{loc}}(a)^{-1} K_{\mathsf{loc}}(1)]^t K_{\mathsf{loc}}(a)^{-1} \\ &\approx \sum_{t=0}^{T-1} [-A^s_{\mathsf{loc}} K_{\mathsf{loc}}(a)^{-1} K_{\mathsf{loc}}(1)]^t K_{\mathsf{loc}}(a)^{-1},\end{aligned}$$

if we have $|A^s_{\mathsf{loc}} / \min a_{\mathsf{loc}}| < 1$ for all samples.

The idea is to precompute as much as possible and only multiply random numbers A^s_{loc} with vectors.

Algorithm with subdomain solvers and truncation

We replace,

```
for $s$ from $1$ to $S$ do
  for $d$ from $1$ to $D$ do
    $u^s_{\mathrm{loc}} = [K_{\mathrm{loc}}(a) + A^s_{\mathrm{loc}} K_{\mathrm{loc}}(1)]^{-1} b_{\mathrm{loc}}(f, \mathcal{A}^s, \dots)$
  end for
end for
```

by

```
for $d$ from $1$ to $D$ do
  for $t$ from $0$ to $T-1$ do
    Compute $C^t = [K_{\mathrm{loc}}(a)^{-1} K_{\mathrm{loc}}(1)]^t [K_{\mathrm{loc}}(a)]^{-1}$
  end for
  for $s$ from $1$ to $S$ do
    $u^s_{\mathrm{loc}} \approx \sum_{t=0}^{T-1} (-A^s_{\mathrm{loc}})^t C^t b_{\mathrm{loc}}(f, \mathcal{A}^s, \dots)$
  end for
end for
```

Example: non-overlapping domain decomposition

To illustrate this we use Lion's overlapping Domain Decomposition algorithm.

On each subdomain we solve,

$$\begin{aligned} -\nabla\cdot(a+A^s)\nabla u_{\text{loc}}^{s,i+1} &= f, \quad \text{for all } x\in\Omega_{\text{loc}}, \\ u_{\text{loc}}^{s,i+1} &= 0, \quad \text{for all } x\in\Gamma, \\ u_{\text{loc}}^{s,i+1} + n\cdot(a+A_{\text{loc}}^s)\nabla u_{\text{loc}}^{s,i+1} &= u_{\text{loc}'}^{s,i} - n\cdot(a+A_{\text{loc}'}^s)\nabla u_{\text{loc}'}^{s,i}, \text{ on } \partial\Omega_{\text{loc}}\setminus\Gamma. \end{aligned}$$

this means that,

$$b_{\text{loc}}^j = (f,\varphi_j) + (u_{\text{loc}'}^{s,i} - n\cdot(a+A_{\text{loc}}^s)\nabla u_{\text{loc}'}^{s,i},\varphi_j)_{\partial\Omega_D}, \quad j=1,\dots,n_D.$$

Implementation issues

- Inverses are not computed explicitly, linear systems in the small local problems are solved repeatedly.
- Since the subdomains are small coarse grid correction can be needed to get quick convergence. On the coarse grid standard brute force computation is used.
- The solution to each sample needs to be stored at the interior boundaries together with desired output quantities.
- The idea is that the method is going to be more and more effective as the number of sample increases.
- Piecewise polynomial perturbations can easily be covered.

Error estimate for a single sample

If we for the moment fix s we can construct an adjoint problem,

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

to derive an a posteriori error estimate in terms of critical parameters of the method,

$$(u^s_{\text{exact}} - u^s, \psi) = (f, \phi^s) - (\mathcal{A}^s \nabla u^s, \nabla \phi^s) \approx e_{\text{I}} + e_{\text{II}} + \dots,$$

in this case we have parameters h, I, and T, and terms,
$e_{\text{II}}(I) = (\mathcal{A}^s \nabla (u^s_{h,I,T} - u^s_{h,I+\Delta I,T}), \nabla \phi^s)$,

$e_{\text{III}}(T) = (\mathcal{A}^s \nabla (u^s_{h,I+\Delta I,T} - u^s_{h,I+\Delta I,\infty}), \nabla \phi^s) \sim \left(\frac{A^s_{\text{loc}}}{\min a_{\text{loc}}} \right)^T$, and

$e_{\text{I}}(h) = (f, \phi^s) - (\mathcal{A}^s \nabla u^s_{h,I,T}, \nabla \phi^s) - e_{\text{II}} - e_{\text{III}}$.

Error in cumulative distribution function

For each sample we have,

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

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

$$F(x) - F_S(x) = P(\{(u^s_{\text{exact}}, \psi)\}_{s\in\Lambda} < x) - P(\{(u^s_{h,I,T}, \psi)\}_{s=1}^S < x).$$

Using the Central Limit Theorem we get essentially,

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

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 $F_S(x)$ and $DF_S(x)$.
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 $DF_S(x)$.
4. Compute the error indicator associated with the sample size, $e_{IV} = \tau\sqrt{F_S(1 - F_S)/S}$.
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.

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

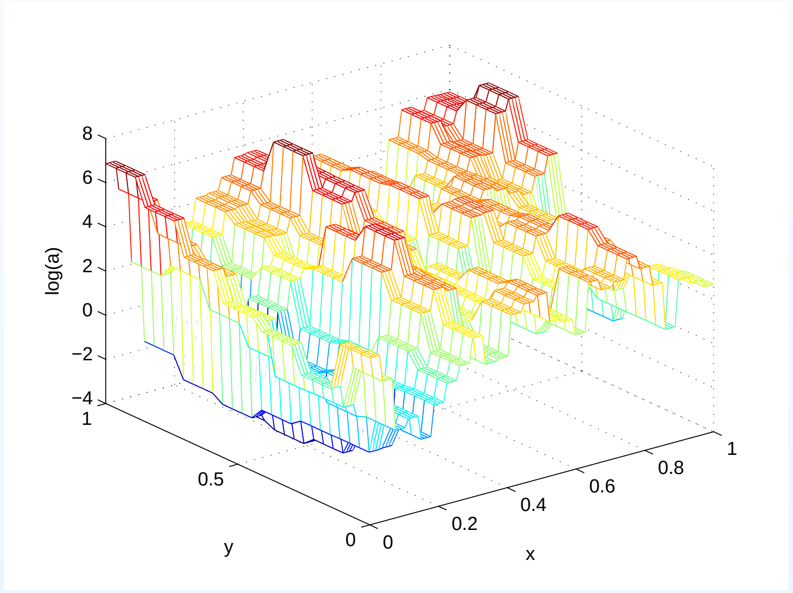
where $\Gamma_N \cup \Gamma_D = \Gamma$. Here u^s represents the pressure field, and a is the local permeability.

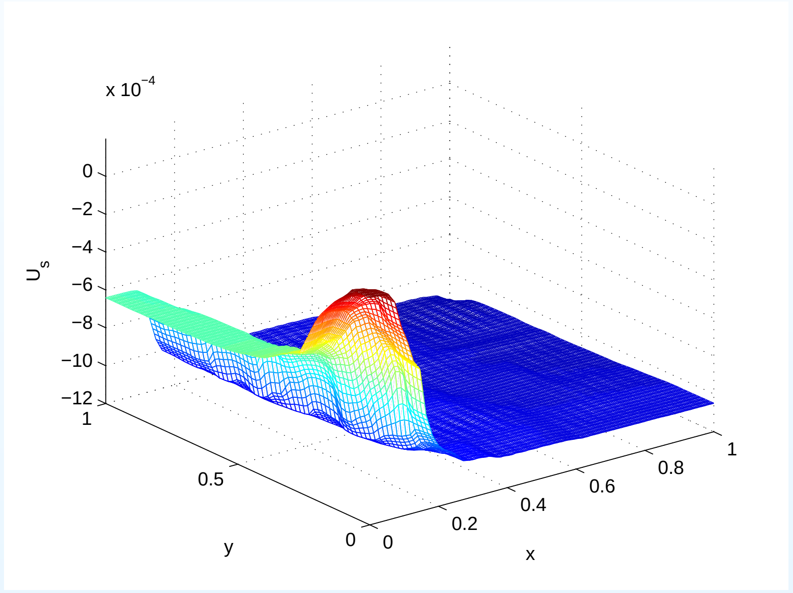
We have chosen $f = 1$ in the lower left corner, the injector, and $f = -1$ in the upper right corner, the producer.

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.





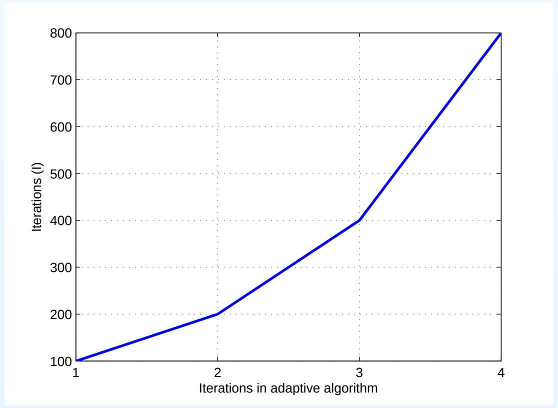
We add a random perturbation to a (20% of the magnitude of a). To the right: a typical solution u^s.

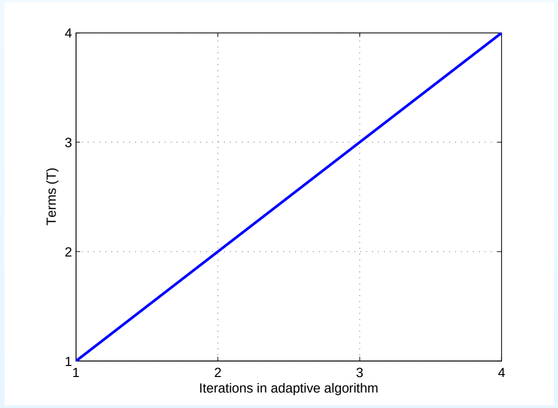
The band of low permeability at $x \approx 0.2$ creates a large pressure drop parallel to the y-axis at this location.

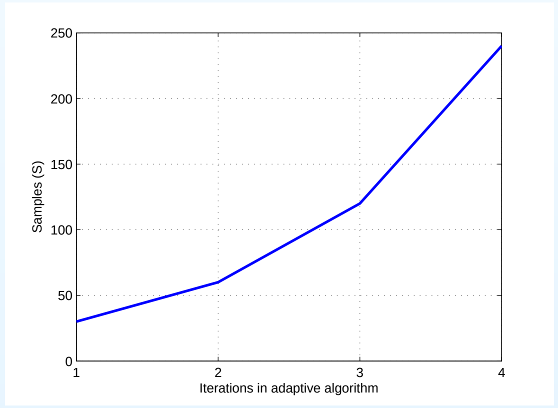
Numerical example: oil reservoir data

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 in 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 $= 0.15$.



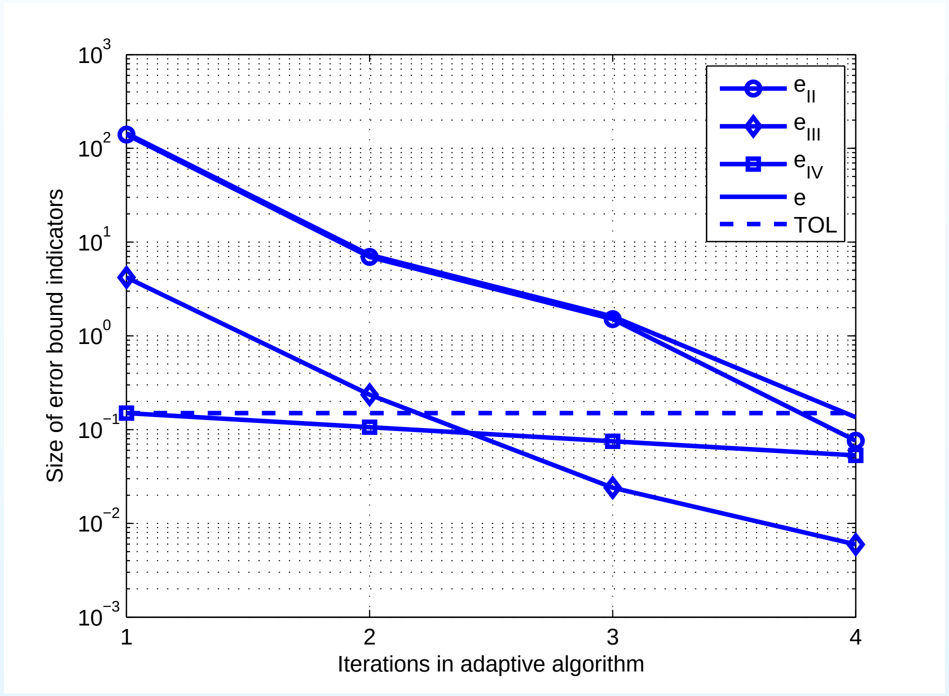




Since the mesh size is fix in this example h does not appear in the figure. The error tolerance is achieved when $I = 800$, $T = 4$, and $S = 240$.

Numerical example: oil reservoir data

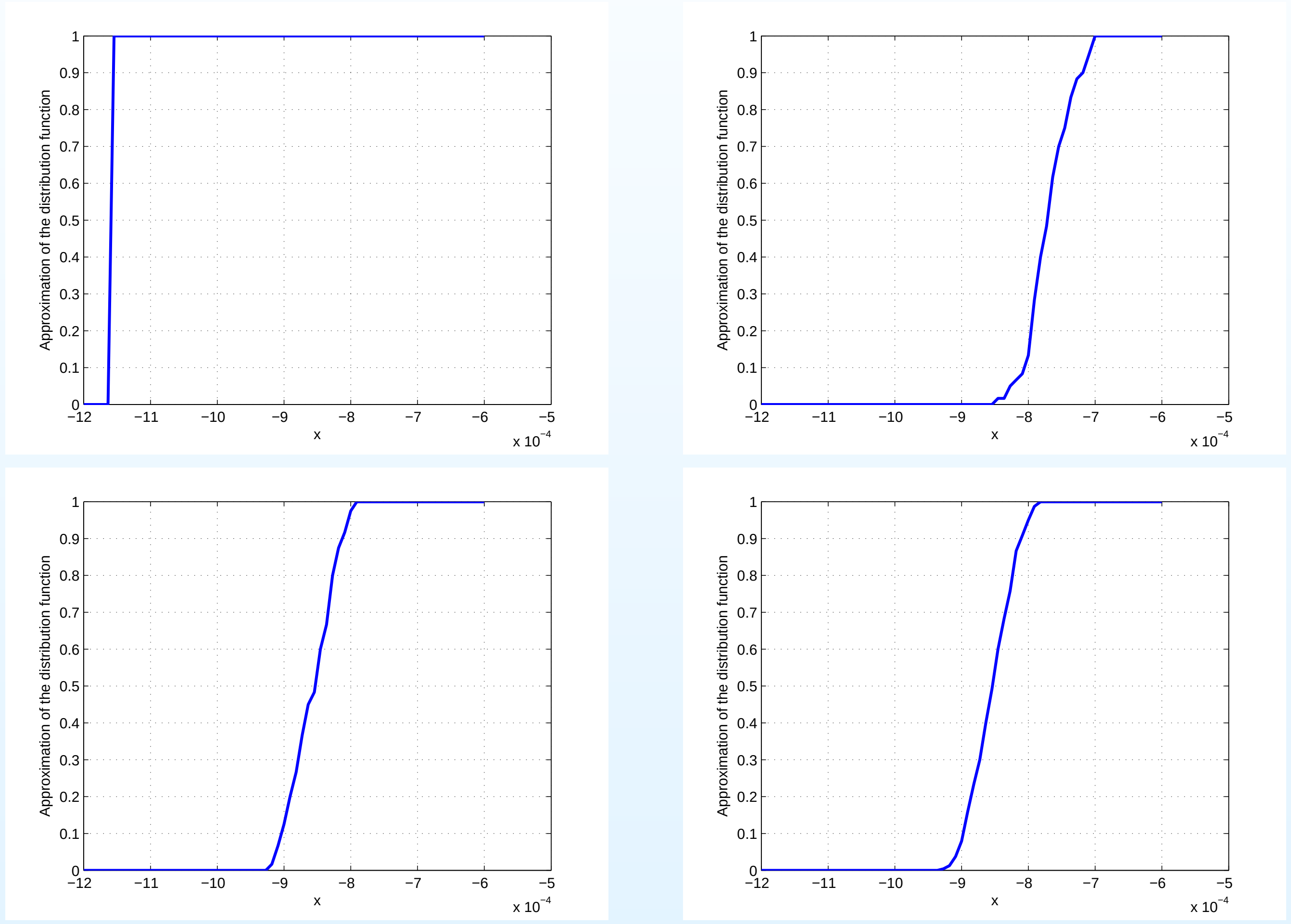
We plot error bound indicators after each iteration in the adaptive algorithm and the total error bound.



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 $F_S(x)$ after each iteration.



Future Work

- Study convergence of Lion’s method when the local solves are done approximately, this is partially done and also work in progress.
- Implement multiscale methods as a compliment to Lion’s method.
- More experimental tests on other data sets.
- Extend the technique and study more challenging equations such as the transport equation in oil reservoir simulation