Multilevel Monte Carlo Methods for Uncertainty Quantification in Reservoir Simulation

Øystein S. Klemetsdal

PhD Trial Lecture
November 27, 2019, NTNU, Trondheim, Norway

Introdution

- Many problems are in principle deterministic, but we don't know the parameters
  - From computational finance to plasma physics
  - … and reservoir simulation, where subsurface properties are uncertain

Introdution

- Many problems are in principle deterministic, but we don't know the parameters
  - From computational finance to plasma physics
  - ... and reservoir simulation, where subsurface properties are uncertain
- Need a method to *quantify uncertainty*
  - Method of moments, collocation methods, stochastic Galerkin
  - Uncertainty with high dimension and highly nonlinear effect
    $\rightarrow$ Monte Carlo methods

Monte Carlo Method

- Let u be random variable with expected value $\mathbb{E}[u]$ and variance $\mathbb{V}[u]$

Monte Carlo Method

- Let u be random variable with expected value $\mathbb{E}[u]$ and variance $\mathbb{V}[u]$
- Approximate $\mathbb{E}[u]$ from independent, identically distributed samples $u^1, \ldots, u^N$

$$\mathbb{E}[u] \approx E(u) = \frac{1}{N} \sum_{i=1}^{N} u^i,$$

Monte Carlo Method

- Let u be random variable with expected value $\mathbb{E}[u]$ and variance $\mathbb{V}[u]$
- Approximate $\mathbb{E}[u]$ from independent, identically distributed samples $u^1, \ldots, u^N$

$$\mathbb{E}[u] \approx E(u) = \frac{1}{N}\sum_{i=1}^{N} u^i, \quad \mathbb{V}\left[E(u)\right] = \mathbb{E}\left[\left(E(u) - \mathbb{E}\left[E(u)\right]\right)^2\right] = \frac{1}{N}\mathbb{V}[u]$$

Monte Carlo Method

- Let u be random variable with expected value $\mathbb{E}[u]$ and variance $\mathbb{V}[u]$
- Approximate $\mathbb{E}[u]$ from independent, identically distributed samples $u^1, \ldots, u^N$

$$\mathbb{E}[u] \approx E(u) = \frac{1}{N}\sum_{i=1}^{N} u^i, \quad \mathbb{V}\left[E(u)\right] = \mathbb{E}\left[\left(E(u) - \mathbb{E}\left[E(u)\right]\right)^2\right] = \frac{1}{N}\mathbb{V}[u]$$

- **Upsides:** Easy to implement and easy to parallelize

Monte Carlo Method

- Let u be random variable with expected value $\mathbb{E}[u]$ and variance $\mathbb{V}[u]$
- Approximate $\mathbb{E}[u]$ from independent, identically distributed samples $u^1, \ldots, u^N$

$$\mathbb{E}[u] \approx E(u) = \frac{1}{N} \sum_{i=1}^{N} u^i, \quad \mathbb{V}\left[E(u)\right] = \mathbb{E}\left[\left(E(u) - \mathbb{E}\left[E(u)\right]\right)^2\right] = \frac{1}{N}\mathbb{V}[u]$$

- **Upsides:** Easy to implement and easy to parallelize
- **Downside:** Root mean square error (RMSE) of the estimator is

$$\mathcal{E}^{\mathrm{MC}}(u) = \sqrt{\mathbb{V}\left[E(u)\right]} = \mathcal{O}(N^{-1/2})$$

$\rightarrow$ Accuracy $\mathcal{E}^{\mathrm{MC}} < \varepsilon$ requires $N = \mathcal{O}(\varepsilon^{-2})$ samples!

Two-Level Monte Carlo Method

- Premise: we can obtain inexpensive approximation u_0 of $u_1 = u$

Two-Level Monte Carlo Method

- Premise: we can obtain inexpensive approximation u_0 of $u_1 = u$
- Express expected value as

$$\mathbb{E}[u_1] = \mathbb{E}[u_0] + \mathbb{E}[u_1 - u_0]$$

Two-Level Monte Carlo Method

- Premise: we can obtain inexpensive approximation u_0 of $u_1 = u$
- Express expected value as

$$\mathbb{E}[u_1] = \mathbb{E}[u_0] + \mathbb{E}[u_1 - u_0]$$

- ... with unbiased estimator

$$\mathbb{E}[u_1] \approx E(u_1) = \frac{1}{N_0} \sum_{i=1}^{N_0} u_0^i + \frac{1}{N_1} \sum_{i=1}^{N_1} (u_1^i - u_0^i)$$

Two-Level Monte Carlo Method

- Premise: we can obtain inexpensive approximation u_0 of $u_1 = u$
- Express expected value as

$$\mathbb{E}[u_1] = \mathbb{E}[u_0] + \mathbb{E}[u_1 - u_0]$$

- ... with unbiased estimator

$$\mathbb{E}[u_1] \approx E(u_1) = \frac{1}{N_0} \sum_{i=1}^{N_0} u_0^i + \frac{1}{N_1} \sum_{i=1}^{N_1} (u_1^i - u_0^i)$$

  - Quantities u_1^i and u_0^i come from the *same* random sample i

Two-Level Monte Carlo Method

- Total cost C (e.g., CPU time) and total variance V:

$$C = N_0 C_0 + N_1 C_1, \qquad V = \frac{V_0}{N_0} + \frac{V_1}{N_1}$$

C_0	Cost of computing single sample of u_0	V_0	Variance $\mathbb{V}[u_0]$
C_1	Cost of computing single sample of $u_1 - u_0$	V_1	Variance $\mathbb{V}[u_1 - u_0]$

Two-Level Monte Carlo Method

- Total cost C (e.g., CPU time) and total variance V:

$$C = N_0 C_0 + N_1 C_1, \qquad V = \frac{V_0}{N_0} + \frac{V_1}{N_1}$$

C_0	Cost of computing single sample of u_0	V_0	Variance $\mathbb{V}[u_0]$
C_1	Cost of computing single sample of $u_1 - u_0$	V_1	Variance $\mathbb{V}[u_1 - u_0]$

- Minimize total cost C for a fixed variance ε^2 (see blackboard)

$$\min \; N_0 C_0 + N_1 C_1 \quad \text{s.t.} \quad \frac{V_0}{N_0} + \frac{V_1}{N_1} = \varepsilon^2$$

Two-Level Monte Carlo Method

- Total cost C (e.g., CPU time) and total variance V:

$$C = N_0 C_0 + N_1 C_1, \qquad V = \frac{V_0}{N_0} + \frac{V_1}{N_1}$$

C_0	Cost of computing single sample of u_0	V_0	Variance $\mathbb{V}[u_0]$
C_1	Cost of computing single sample of $u_1 - u_0$	V_1	Variance $\mathbb{V}[u_1 - u_0]$

- Minimize total cost C for a fixed variance ε^2 (see blackboard)

$$\min\ N_0 C_0 + N_1 C_1 \quad \text{s.t.} \quad \frac{V_0}{N_0} + \frac{V_1}{N_1} = \varepsilon^2 \qquad \rightarrow N_\ell = \varepsilon^{-2} \left(\sqrt{V_0 C_0} + \sqrt{V_1 C_1} \right) \sqrt{\frac{V_\ell}{C_\ell}}$$

Multilevel Monte Carlo Method

- Premise: we can obtain less expensive approximation $u_{\ell-1}$ of u_ℓ for $u_0, \ldots, u_L = u$

Multilevel Monte Carlo Method

- Premise: we can obtain less expensive approximation $u_{\ell-1}$ of u_ℓ for $u_0, \ldots, u_L = u$
- Express expected value as telescopic sum (with $u_{-1} \equiv 0$)

$$\mathbb{E}[u_L] = \sum_{\ell=0}^{L} \mathbb{E}[u_\ell - u_{\ell-1}]$$

Multilevel Monte Carlo Method

- Premise: we can obtain less expensive approximation $u_{\ell-1}$ of u_ℓ for $u_0, \ldots, u_L = u$
- Express expected value as telescopic sum (with $u_{-1} \equiv 0$)

$$\mathbb{E}[u_L] = \sum_{\ell=0}^{L} \mathbb{E}[u_\ell - u_{\ell-1}]$$

- ... with unbiased estimator

$$\mathbb{E}[u_L] \approx E(u_L) = \sum_{\ell=0}^{L} \left(\frac{1}{N_\ell} \sum_{i=1}^{N_\ell} \left(u_\ell^{(\ell,i)} - u_{\ell-1}^{(\ell,i)} \right) \right)$$

Multilevel Monte Carlo Method

- Premise: we can obtain less expensive approximation $u_{\ell-1}$ of u_ℓ for $u_0, \ldots, u_L = u$
- Express expected value as telescopic sum (with $u_{-1} \equiv 0$)

$$\mathbb{E}[u_L] = \sum_{\ell=0}^{L} \mathbb{E}[u_\ell - u_{\ell-1}]$$

- ... with unbiased estimator

$$\mathbb{E}[u_L] \approx E(u_L) = \sum_{\ell=0}^{L} \left(\frac{1}{N_\ell} \sum_{i=1}^{N_\ell} \left(u_\ell^{(\ell,i)} - u_{\ell-1}^{(\ell,i)} \right) \right)$$

  - Quantities $u_\ell^{(\ell,i)}$ and $u_{\ell-1}^{(\ell,i)}$ from the *same* random sample i, *different* for each ℓ

Multilevel Monte Carlo Method

- Total cost C (e.g., CPU time) and total variance V:

$$C = \sum_{\ell=0}^{L} N_\ell C_\ell, \qquad V = \sum_{\ell=0}^{L} \frac{V_\ell}{N_\ell}$$

C_ℓ	Cost of computing single sample of $u_\ell - u_{\ell-1}$	V_ℓ	Variance $\mathbb{V}[u_\ell - u_{\ell-1}]$

Multilevel Monte Carlo Method

- Total cost C (e.g., CPU time) and total variance V:

$$C = \sum_{\ell=0}^{L} N_\ell C_\ell, \qquad V = \sum_{\ell=0}^{L} \frac{V_\ell}{N_\ell}$$

C_ℓ	Cost of computing single sample of $u_\ell - u_{\ell-1}$	V_ℓ	Variance $\mathbb{V}[u_\ell - u_{\ell-1}]$

- Minimize total cost C for a fixed variance ε^2

$$\min \sum_{\ell=0}^{L} N_\ell C_\ell \quad \text{s.t.} \quad \sum_{\ell=0}^{L} \frac{V_\ell}{N_\ell} = \varepsilon^2 \quad \rightarrow \quad N_\ell = \varepsilon^{-2} \left(\sum_{k=0}^{L} \sqrt{V_k C_k} \right) \sqrt{\frac{V_\ell}{C_\ell}}$$

Multilevel Monte Carlo Method

- Generally, u_ℓ is obtained by simulation, so that u_L is approximation of u

Multilevel Monte Carlo Method

- Generally, u_ℓ is obtained by simulation, so that u_L is approximation of u
- Mean square error is now

$$\mathcal{E}^{\mathsf{ML}}(u_L)^2 = \mathbb{E}\left[(E(u_L) - \mathbb{E}[u])^2\right]$$

Multilevel Monte Carlo Method

- Generally, u_ℓ is obtained by simulation, so that u_L is approximation of u
- Mean square error is now

$$
\begin{aligned}
\mathcal{E}^{\mathrm{ML}}(u_L)^2 &= \mathbb{E}\left[(E(u_L) - \mathbb{E}[u])^2\right] \\
&= \mathbb{E}\left[(E(u_L) - \mathbb{E}\left[E(u_L)\right] + \mathbb{E}\left[E(u_L)\right] - \mathbb{E}[u])^2\right]
\end{aligned}
$$

Multilevel Monte Carlo Method

- Generally, u_ℓ is obtained by simulation, so that u_L is approximation of u
- Mean square error is now

$$\begin{aligned}
\mathcal{E}^{\mathrm{ML}}(u_L)^2 &= \mathbb{E}\left[\left(E(u_L) - \mathbb{E}[u]\right)^2\right] \\
&= \mathbb{E}\left[\left(E(u_L) - \mathbb{E}\left[E(u_L)\right] + \mathbb{E}\left[E(u_L)\right] - \mathbb{E}[u]\right)^2\right] \\
&= \underbrace{\mathbb{E}\left[\left(E(u_L) - \mathbb{E}\left[E(u_L)\right]\right)^2\right]}_{\text{Sampling error}} + \underbrace{\left(\mathbb{E}\left[E(u_L)\right] - \mathbb{E}[u]\right)^2}_{\text{Approximation error}}
\end{aligned}$$

Multilevel Monte Carlo Method

- Generally, u_ℓ is obtained by simulation, so that u_L is approximation of u
- Mean square error is now

$$\mathcal{E}^{\mathrm{ML}}(u_L)^2 = \underbrace{\mathbb{V}\left[E(u_L)\right]}_{\text{Sampling error}} + \underbrace{\left(\mathbb{E}\left[E(u_L)\right] - \mathbb{E}[u]\right)^2}_{\text{Approximation error}}$$

Multilevel Monte Carlo Method

- Generally, u_ℓ is obtained by simulation, so that u_L is approximation of u
- Mean square error is now

$$\mathcal{E}^{\mathrm{ML}}(u_L)^2 = \underbrace{\mathbb{V}\left[E(u_L)\right]}_{\text{Sampling error}} + \underbrace{\left(\mathbb{E}\left[E(u_L)\right] - \mathbb{E}[u]\right)^2}_{\text{Approximation error}}$$

- Sampling error $< \varepsilon^2/2$ and approximation error $< \varepsilon^2/2$ ensures $\mathcal{E}^{\mathrm{ML}}(u_L) < \varepsilon$

Multilevel Monte Carlo Method

- Generally, u_ℓ is obtained by simulation, so that u_L is approximation of u
- Mean square error is now

$$\mathcal{E}^{\mathrm{ML}}(u_L)^2 = \underbrace{\mathbb{V}\left[E(u_L)\right]}_{\text{Sampling error}} + \underbrace{\left(\mathbb{E}\left[E(u_L)\right] - \mathbb{E}[u]\right)^2}_{\text{Approximation error}}$$

- Sampling error $< \varepsilon^2/2$ and approximation error $< \varepsilon^2/2$ ensures $\mathcal{E}^{\mathrm{ML}}(u_L) < \varepsilon$
- Simplify notation: let E_ℓ be MC estimator of $u_\ell - u_{\ell-1}$

$$E_\ell = \frac{1}{N_\ell} \sum_{i=1}^{N_\ell} \left(u_\ell^{(\ell,i)} - u_{\ell-1}^{(\ell,i)} \right), \quad E(u_L) = \sum_{\ell=0}^{L} E_\ell$$

Multilevel Monte Carlo Method

Theorem

If there exists independent estimators E_ℓ based on N_ℓ MC samples, with expected cost C_ℓ and variance V_ℓ, and $\alpha, \beta, \gamma, c_1, c_2, c_3 > 0$ such that $\alpha \geq \min(\beta, \gamma)/2$, and

1. $|\mathbb{E}[u_\ell - u]| \leq c_1 2^{-\alpha\ell}$ *(Increase in accuracy)*
2. $V_\ell \leq c_2 2^{-\beta\ell}$ *(Decrease in variance)*
3. $C_\ell \leq c_3 2^{\gamma\ell}$ *(Increase in cost)*

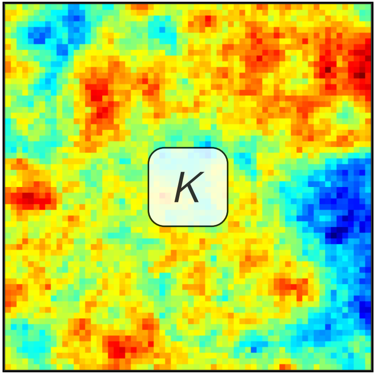
Then, there exists $c_4 > 0$ such that for $\varepsilon < e^{-1}$, there are L, N_ℓ for which the estimator $E(u_L) = \sum_\ell E_\ell$ has $\mathcal{E}^{\mathrm{ML}}(u_L) < \varepsilon$, and

$$\mathbb{E}[C] \leq \begin{cases} c_4 \varepsilon^{-2} & \beta > \gamma \\ c_4 \varepsilon^{-2} \log(\varepsilon)^2 & \beta = \gamma \\ c_4 \varepsilon^{-2-(\gamma-\beta)/\alpha} & \beta < \gamma \end{cases}$$

MLMC for Uncertainty Quantification in Reservoir Simulation

- Incompressible flow in porous media

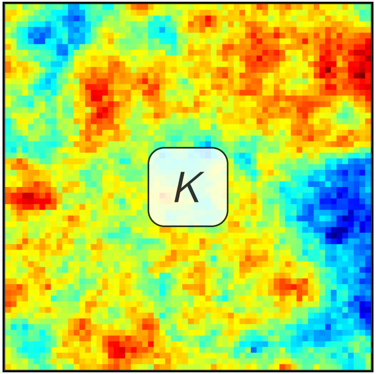
$$\vec{v}(\boldsymbol{x}) = -K(\boldsymbol{x})\nabla p(\boldsymbol{x}), \quad \nabla \cdot \vec{v}(\boldsymbol{x}) = q(\boldsymbol{x})$$



MLMC for Uncertainty Quantification in Reservoir Simulation

- Incompressible flow in porous media

$$-\nabla \cdot \big(K(\boldsymbol{x})\nabla p(\boldsymbol{x})\big) = q(\boldsymbol{x})$$

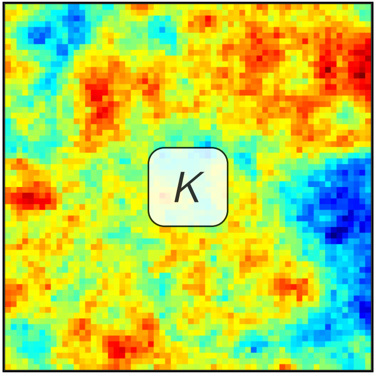


MLMC for Uncertainty Quantification in Reservoir Simulation

- Incompressible flow in porous media

$$-\nabla \cdot \big(K(\boldsymbol{x})\nabla p(\boldsymbol{x})\big) = q(\boldsymbol{x})$$

- Permeability K is *uncertain*
  - Few physical samples, uncertain seismic data

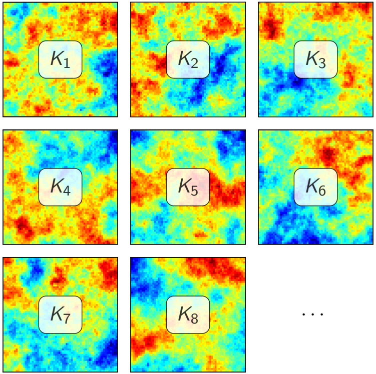


MLMC for Uncertainty Quantification in Reservoir Simulation

- Incompressible flow in porous media – SPDE

$$-\nabla \cdot \big(K(\boldsymbol{x},\omega)\nabla p(\boldsymbol{x},\omega)\big) = q(\boldsymbol{x},\omega)$$

- Permeability K is *uncertain*
  - Few physical samples, uncertain seismic data
- Modelled as random field $K(\boldsymbol{x},\omega)$
  - Spatial *and* stochastic variable $(\boldsymbol{x},\omega) \in D \times \Omega$

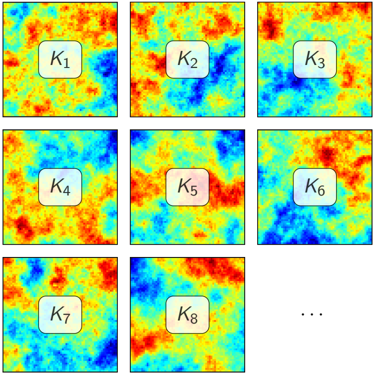


MLMC for Uncertainty Quantification in Reservoir Simulation

- Incompressible flow in porous media – SPDE

$$-\nabla \cdot \big(K(\boldsymbol{x},\omega)\nabla p(\boldsymbol{x},\omega)\big) = q(\boldsymbol{x},\omega)$$

- Permeability K is *uncertain*
  - Few physical samples, uncertain seismic data
- Modelled as random field $K(\boldsymbol{x},\omega)$
  - Spatial *and* stochastic variable $(\boldsymbol{x},\omega) \in D \times \Omega$
- Fixing ω gives a deterministic PDE



Two-phase flow in porous media

$$\partial_t(\phi S_\alpha) + \nabla \cdot \vec{v}_\alpha = q_\alpha, \quad \vec{v}_\alpha = -\lambda_\alpha K \nabla p, \quad \alpha = w, o$$

ϕ: porosity S_α: saturation $\vec{v}_\alpha$: Darcy velocity q_α: sources/sinks λ_α: mobility

Two-phase flow in porous media

$$\partial_t(\phi S_\alpha) + \nabla \cdot \vec{v}_\alpha = q_\alpha, \quad \vec{v}_\alpha = -\lambda_\alpha K \nabla p, \quad \alpha = w, o$$

ϕ: porosity S_α: saturation $\vec{v}_\alpha$: Darcy velocity q_α: sources/sinks λ_α: mobility

- Quantity of interest u typically derived from simulation results, e.g.

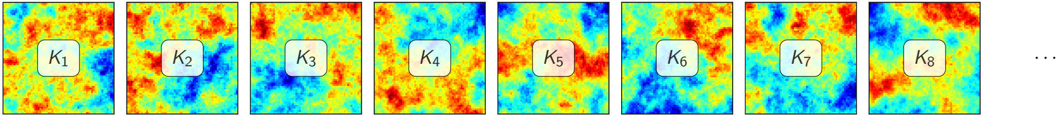
Saturation at time t'	$u = S_w(\boldsymbol{x}, t')$
Water production rate	$u = q_w(t)$
Total oil production	$u = Q_o$

Two-phase flow in porous media

$$\partial_t(\phi S_\alpha) + \nabla \cdot \vec{v}_\alpha = q_\alpha, \quad \vec{v}_\alpha = -\lambda_\alpha K \nabla p, \quad \alpha = w, o$$

ϕ: porosity $\quad S_\alpha$: saturation $\quad \vec{v}_\alpha$: Darcy velocity $\quad q_\alpha$: sources/sinks $\quad \lambda_\alpha$: mobility

- Quantity of interest u typically derived from simulation results
  ... which are all random variables



MLMC for Uncertainty Quantification in Reservoir Simulation

- Quantity of interest u generally not scalar, but defined over domain Λ

Saturation at time t'	$u = S_w(\boldsymbol{x}, t')$	$\Lambda = D$ (physical domain)
Water production rate	$u = q_w(t)$	$\Lambda = [0, T]$ (time domain)
Total oil production	$u = Q_o$	Λ not applicable

MLMC for Uncertainty Quantification in Reservoir Simulation

- Quantity of interest u generally not scalar, but defined over domain Λ

Saturation at time t'	$u = S_w(\boldsymbol{x}, t')$	$\Lambda = D$ (physical domain)
Water production rate	$u = q_w(t)$	$\Lambda = [0, T]$ (time domain)
Total oil production	$u = Q_o$	Λ not applicable

- Necessary substitutions with appropriate norm $\|\cdot\|$ over Λ

$$|\mathbb{E}[u_\ell] - \mathbb{E}[u]| \to \|\mathbb{E}[u_\ell] - \mathbb{E}[u]\|, \quad \mathbb{V}[u_\ell - u_{\ell-1}] \to \mathbb{E}\left[\|u_\ell - u_{\ell-1} - \mathbb{E}[u_\ell - u_{\ell-1}]\|^2\right]$$

MLMC for Uncertainty Quantification in Reservoir Simulation

- Quantity of interest u generally not scalar, but defined over domain Λ

Saturation at time t'	$u = S_w(\boldsymbol{x}, t')$	$\Lambda = D$ (physical domain)
Water production rate	$u = q_w(t)$	$\Lambda = [0, T]$ (time domain)
Total oil production	$u = Q_o$	Λ not applicable

- Necessary substitutions with appropriate norm $\|\cdot\|$ over Λ

$$|\mathbb{E}[u_\ell] - \mathbb{E}[u]| \to \|\mathbb{E}[u_\ell] - \mathbb{E}[u]\|, \quad \mathbb{V}[u_\ell - u_{\ell-1}] \to \mathbb{E}\left[\|u_\ell - u_{\ell-1} - \mathbb{E}[u_\ell - u_{\ell-1}]\|^2\right]$$

- Theory for scalar variables follows directly – particularly with $L^2(\Lambda)$-norm:

$$\mathbb{V}\left[E(u_L)\right] = \sum_{\ell=0}^{L} \frac{V_\ell}{N_\ell}, \quad \text{where } V_\ell = \mathbb{E}\left[\|u_\ell - u_{\ell-1} - \mathbb{E}[u_\ell - u_{\ell-1}]\|^2\right]$$

Algorithm: Multilevel Monte Carlo Method

```
Input: Hierarchy of approximations ℓ = 0, ..., L, tolerance ε
for ℓ = 0, ..., L do                                              // Warmup
    Compute N_w samples of u_ℓ − u_{ℓ−1};                          // Local MC
end
Estimate V_ℓ, C_ℓ and optimal N_ℓ = N_w + N'_ℓ given desired tolerance ε
while any extra samples needed, (N'_ℓ > 0) do                     // Multilevel MC
    for ℓ = 0, ..., L do
        Compute N'_ℓ more samples of u_ℓ − u_{ℓ−1};                // Local MC
    end
    Update estimates V_ℓ, C_ℓ and optimal N_ℓ = N_ℓ + N'_ℓ given desired accuracy ε
end
```

Algorithm: Multilevel Monte Carlo Method

Input: Hierarchy of approximations $\ell = 0, \ldots, L$, tolerance ε

```
for ℓ = 0,...,L do                                                  // Warmup
    Compute N_w samples of u_ℓ − u_{ℓ−1};                           // Local MC
end
Estimate V_ℓ, C_ℓ and optimal N_ℓ = N_w + N'_ℓ given desired tolerance ε
while any extra samples needed, (N'_ℓ > 0) do                       // Multilevel MC
    for ℓ = 0,...,L do
        Compute N'_ℓ more samples of u_ℓ − u_{ℓ−1};                 // Local MC
    end
    Update estimates V_ℓ, C_ℓ and optimal N_ℓ = N_ℓ + N'_ℓ given desired accuracy ε
end
```

MLMC for Uncertainty Quantification in Reservoir Simulation

Algorithm: Multilevel Monte Carlo Method

```
Input: Hierarchy of approximations ℓ = 0, ..., L, tolerance ε
for ℓ = 0, ..., L do                                                    // Warmup
    Compute N_w samples of u_ℓ − u_{ℓ−1};                               // Local MC
end
Estimate V_ℓ, C_ℓ and optimal N_ℓ = N_w + N'_ℓ given desired tolerance ε
while any extra samples needed, (N'_ℓ > 0) do                           // Multilevel MC
    for ℓ = 0, ..., L do
        Compute N'_ℓ more samples of u_ℓ − u_{ℓ−1};                     // Local MC
    end
    Update estimates V_ℓ, C_ℓ and optimal N_ℓ = N_ℓ + N'_ℓ given desired accuracy ε
end
```

MLMC for Uncertainty Quantification in Reservoir Simulation

Algorithm: Multilevel Monte Carlo Method

Input: Hierarchy of approximations $\ell = 0, \ldots, L$, tolerance ε

```
for ℓ = 0,...,L do                                                   // Warmup
    Compute N_w samples of u_ℓ − u_{ℓ−1};                            // Local MC
end
Estimate V_ℓ, C_ℓ and optimal N_ℓ = N_w + N'_ℓ given desired tolerance ε
while any extra samples needed, (N'_ℓ > 0) do                        // Multilevel MC
    for ℓ = 0,...,L do
        Compute N'_ℓ more samples of u_ℓ − u_{ℓ−1};                  // Local MC
    end
    Update estimates V_ℓ, C_ℓ and optimal N_ℓ = N_ℓ + N'_ℓ given desired accuracy ε
end
```

Two Illustrating Examples

Problem

- The `hello world` of reservoir simulation:
  - Quarter five-spot problem with water injection in oil-filled reservoir
- Incompressible flow, linear relative permeabilities, equal viscosities
- Assume $\log K$ is Gaussian with given covariance function $\rightarrow$ 1000 realizations

Two Illustrating Examples

Problem

- The `hello world` of reservoir simulation:
  - Quarter five-spot problem with water injection in oil-filled reservoir
- Incompressible flow, linear relative permeabilities, equal viscosities
- Assume $\log K$ is Gaussian with given covariance function $\rightarrow$ 1000 realizations

Strategy

1. Run MC simulation with 100 samples, compute RMSE $\mathcal{E}^{\mathrm{MC}}$
2. For layer $0 \ldots, L$, run 10 warmup samples to estimate C_ℓ and V_ℓ
3. Compute N_ℓ for desired tolerance $\varepsilon \approx \mathcal{E}^{\mathrm{MC}}$, run and compare with MC

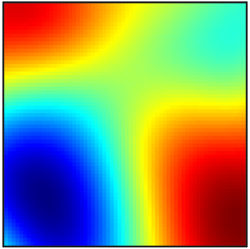
Example 1: Smooth Permeability

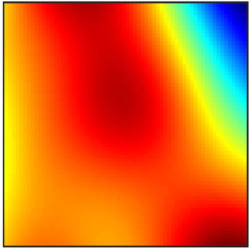
Covariance function

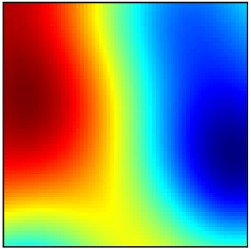
$$C(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{\lambda}\right)$$

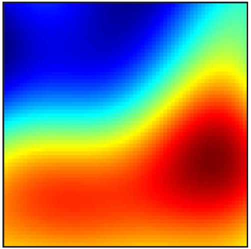
covariance $\sigma^2 = 1$

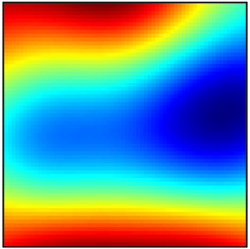
correlation length $\lambda = 0.3$











$\cdots$

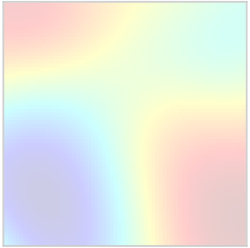
Example 1: Smooth Permeability

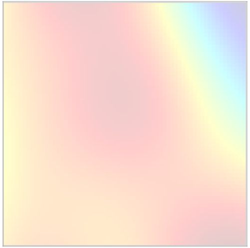
Covariance function

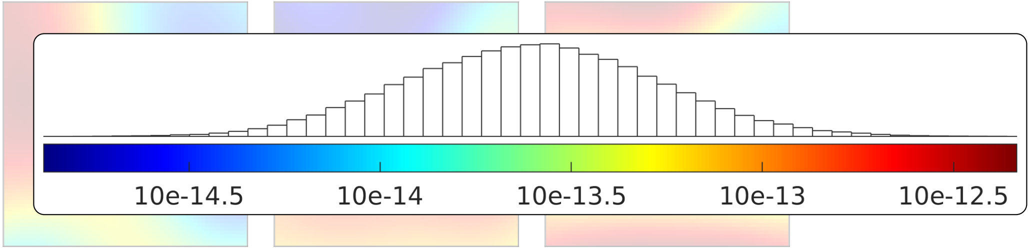
$$C(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{\lambda}\right)$$

covariance $\sigma^2 = 1$

correlation length $\lambda = 0.3$







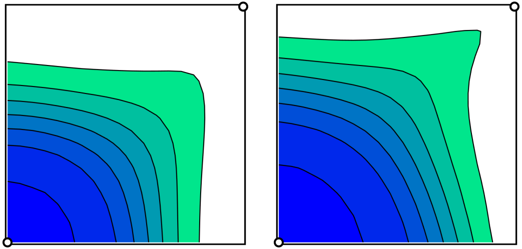
Example 1: Smooth Permeability

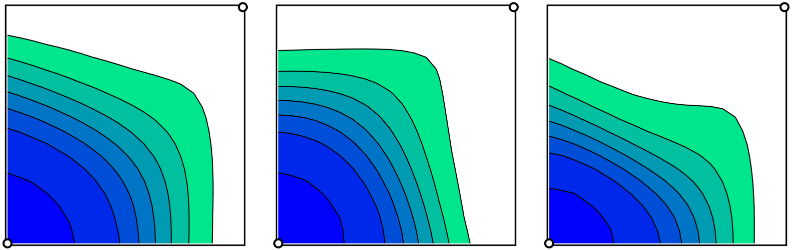
Covariance function

$$C(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{\lambda}\right)$$

covariance $\sigma^2 = 1$

correlation length $\lambda = 0.3$





. . .

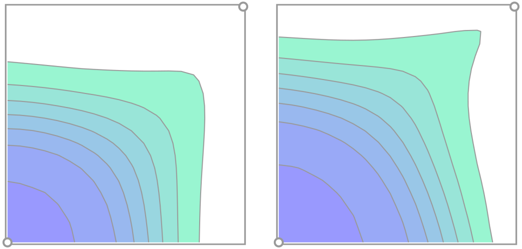
Example 1: Smooth Permeability

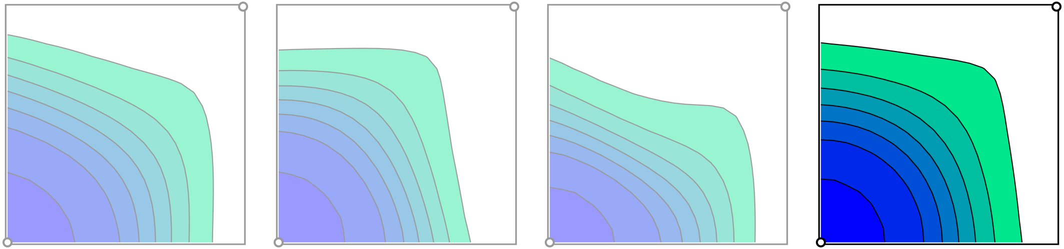
Covariance function

$$C(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{\lambda}\right)$$

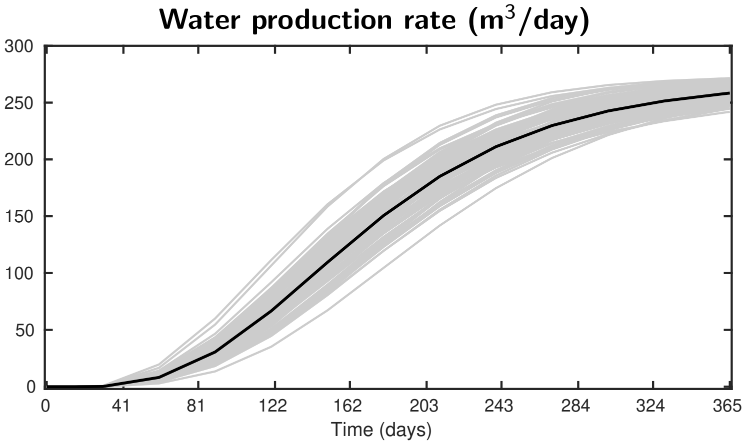
covariance $\sigma^2 = 1$

correlation length $\lambda = 0.3$



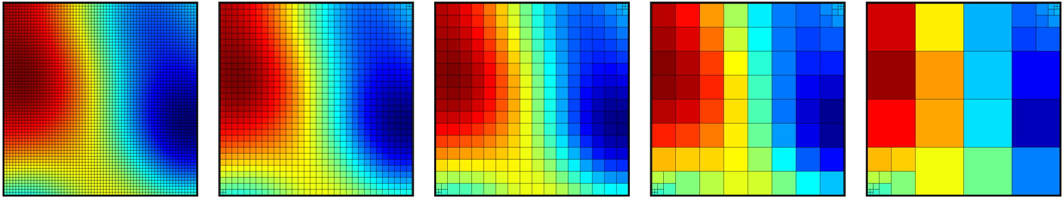


Example 1: Smooth Permeability



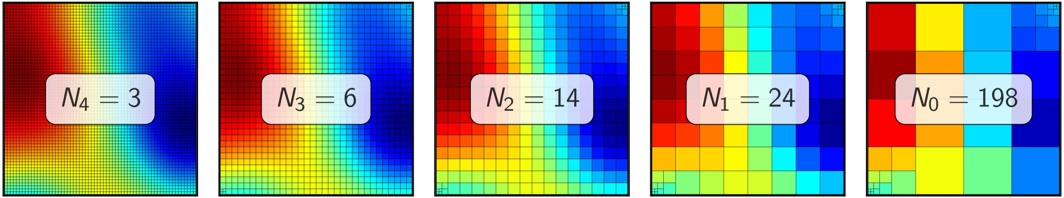
- Monte Carlo simulation with 100 samples: $\mathcal{E}^{\text{MC}}(q_w) = 1.0 \times 10^{-2}$

Example 1: Smooth Permeability



- Five levels with $\sim 4^2$, 8^2, 16^2, 32^2, 64^2 cells + refinement around wells

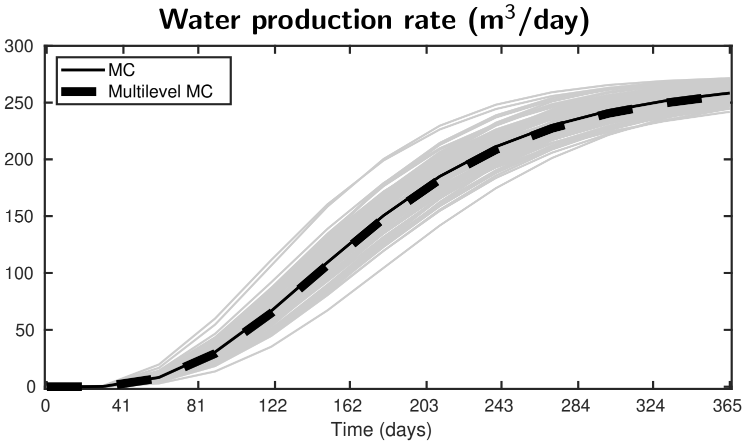
Example 1: Smooth Permeability



- Five levels with $\sim 4^2$, 8^2, 16^2, 32^2, 64^2 cells + refinement around wells
- Warmup: Run 10 samples on each level to estimate V_ℓ and C_ℓ
  $\rightarrow$ used to find optimal N_ℓ for tolerance $\varepsilon \approx \mathcal{E}^{\mathrm{MC}}$

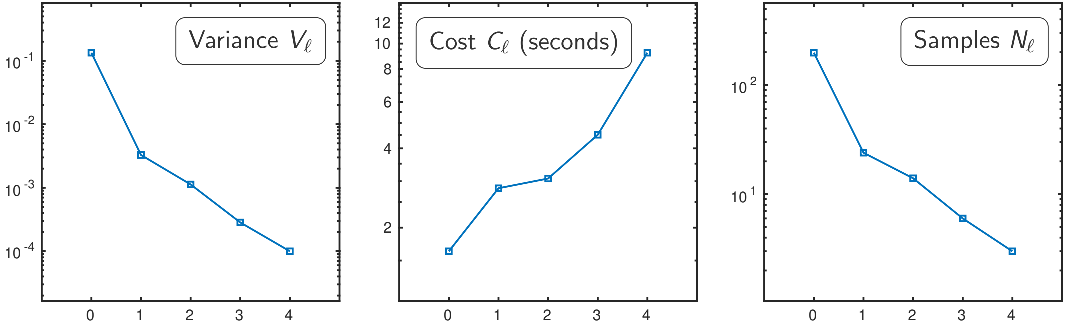
$$\min \sum_{\ell=0}^{L} N_\ell C_\ell \quad \text{s.t.} \quad \sum_{\ell=0}^{L} \frac{V_\ell}{N_\ell} = \varepsilon^2 \quad \rightarrow \quad N_\ell = \varepsilon^{-2} \left(\sum_{k=0}^{L} \sqrt{V_k C_k} \right) \sqrt{\frac{V_\ell}{C_\ell}}$$

Example 1: Smooth Permeability

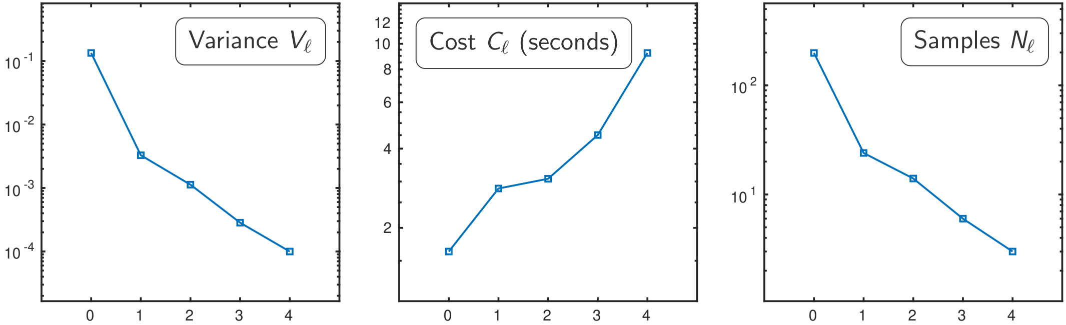


- Multilevel Monte Carlo simulation: $\mathcal{E}^{\mathrm{ML}}(q_w) = 1.5 \times 10^{-2}$

Example 1: Smooth Permeability

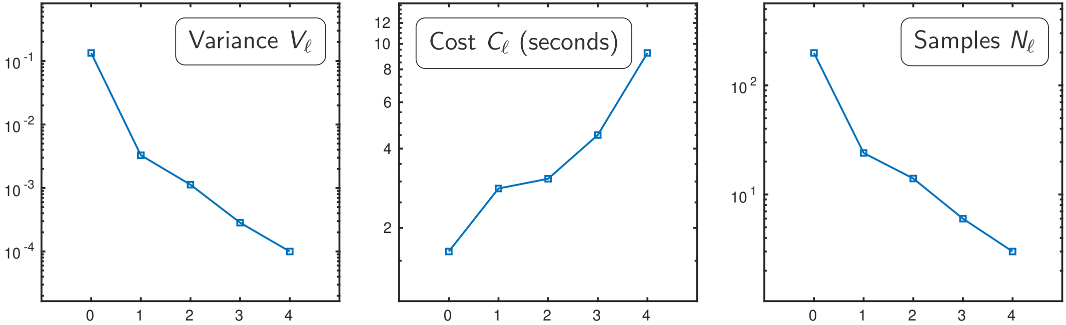


Example 1: Smooth Permeability



- Total cost of Multilevel Monte Carlo: $\sum_\ell N_\ell C_\ell \approx 488$ s

Example 1: Smooth Permeability



- Total cost of Multilevel Monte Carlo: $\sum_\ell N_\ell C_\ell \approx 488$ s
- Total cost of Monte Carlo (assuming C_4 = cost of $u_4 - u_3 \approx$ cost of u_4) ≈ 923 s
  $\to$ Similar accuracy with about half the cost

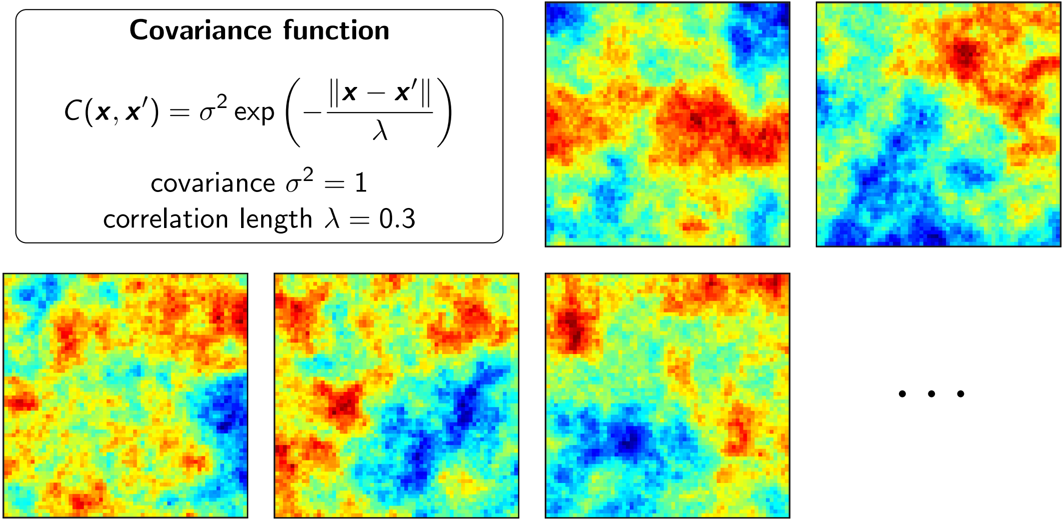
Example 2: High-Contrast Permeability

Covariance function

$$C(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|}{\lambda}\right)$$

covariance $\sigma^2 = 1$

correlation length $\lambda = 0.3$



. . .

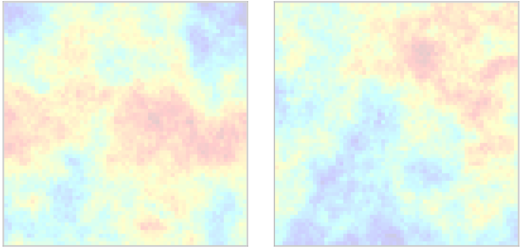
Example 2: High-Contrast Permeability

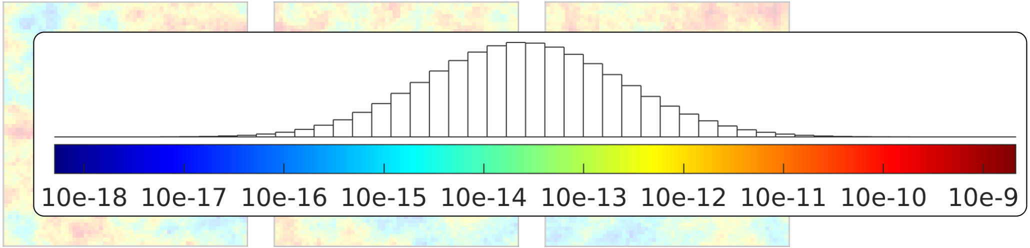
Covariance function

$$C(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|}{\lambda}\right)$$

covariance $\sigma^2 = 1$

correlation length $\lambda = 0.3$





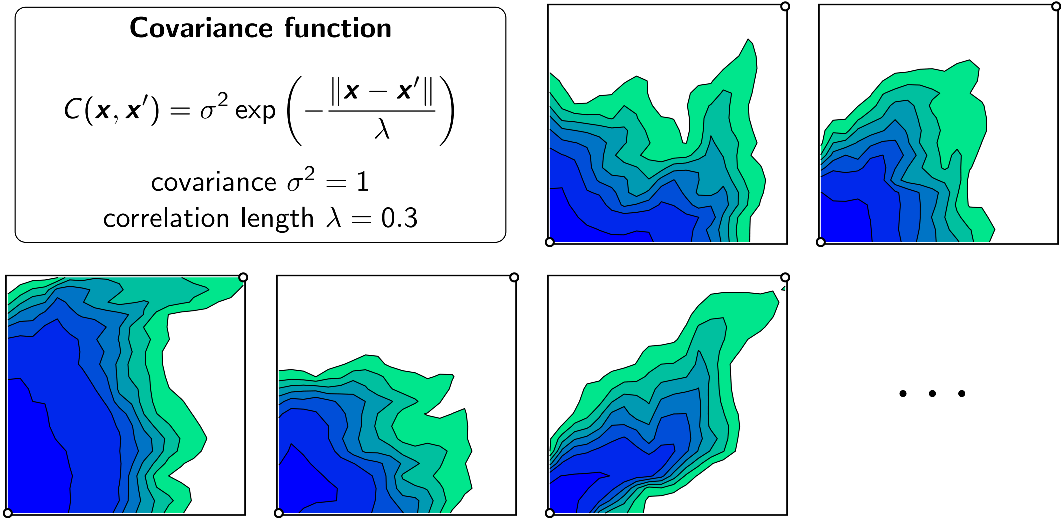
Example 2: High-Contrast Permeability

Covariance function

$$C(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|}{\lambda}\right)$$

covariance $\sigma^2 = 1$

correlation length $\lambda = 0.3$



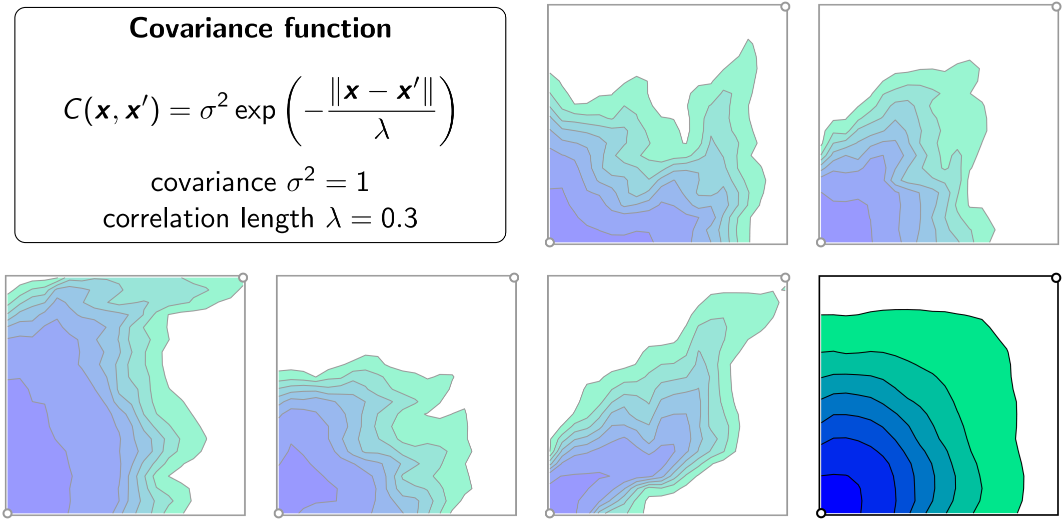
Example 2: High-Contrast Permeability

Covariance function

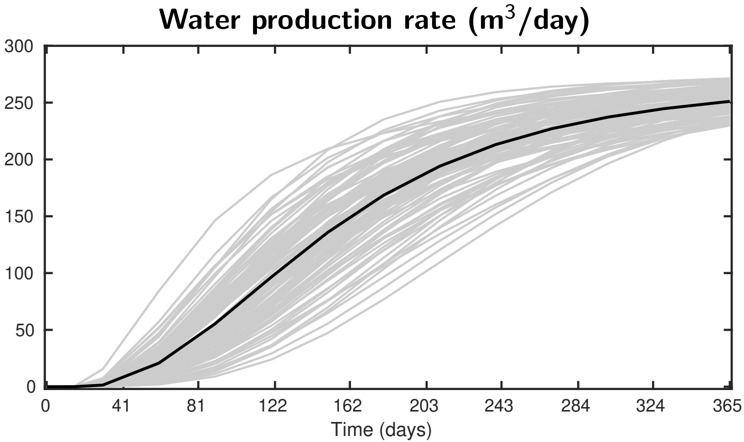
$$C(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|}{\lambda}\right)$$

covariance $\sigma^2 = 1$

correlation length $\lambda = 0.3$

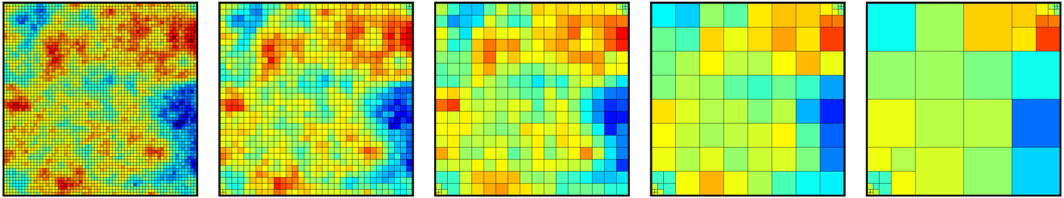


Example 2: High-Contrast Permeability



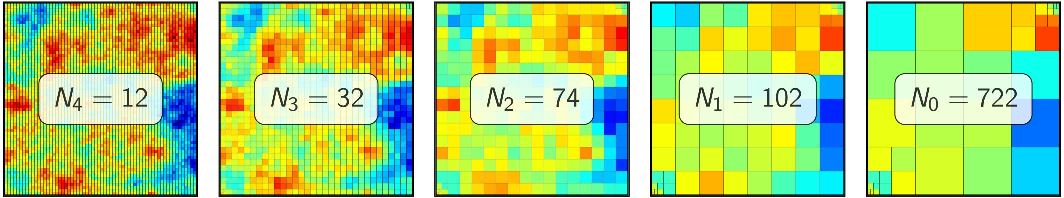
- Monte Carlo simulation with 100 samples: $\mathcal{E}^{\text{MC}}(q_w) = 2.1 \times 10^{-2}$

Example 2: High-Contrast Permeability



- Five levels with $\sim 4^2, 8^2, 16^2, 32^2, 64^2$ cells + refinement around wells

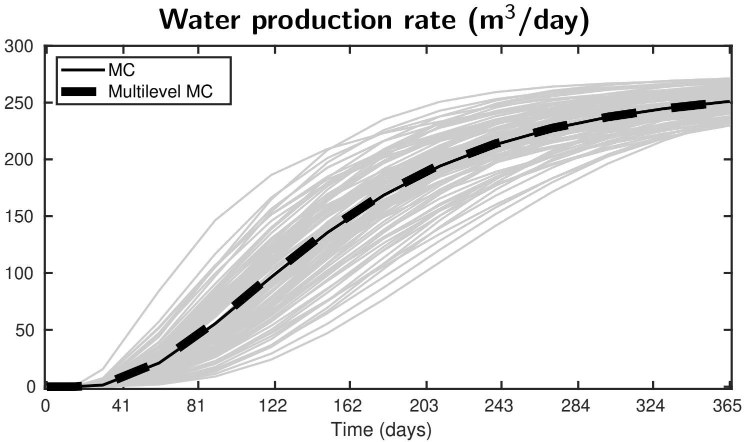
Example 2: High-Contrast Permeability



- Five levels with $\sim 4^2, 8^2, 16^2, 32^2, 64^2$ cells + refinement around wells
- Warmup: Run 10 samples on each level to estimate V_ℓ and C_ℓ
  $\rightarrow$ used to find optimal N_ℓ for tolerance $\varepsilon \approx \mathcal{E}^{\text{MC}}$

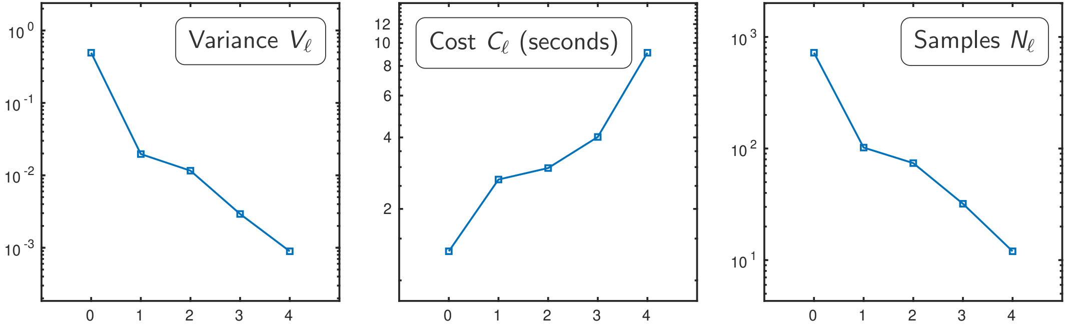
$$\min \sum_{\ell=0}^{L} N_\ell C_\ell \quad \text{s.t.} \quad \sum_{\ell=0}^{L} \frac{V_\ell}{N_\ell} = \varepsilon^2 \quad \rightarrow \quad N_\ell = \varepsilon^{-2} \left(\sum_{k=0}^{L} \sqrt{V_k C_k} \right) \sqrt{\frac{V_\ell}{C_\ell}}$$

Example 2: High-Contrast Permeability

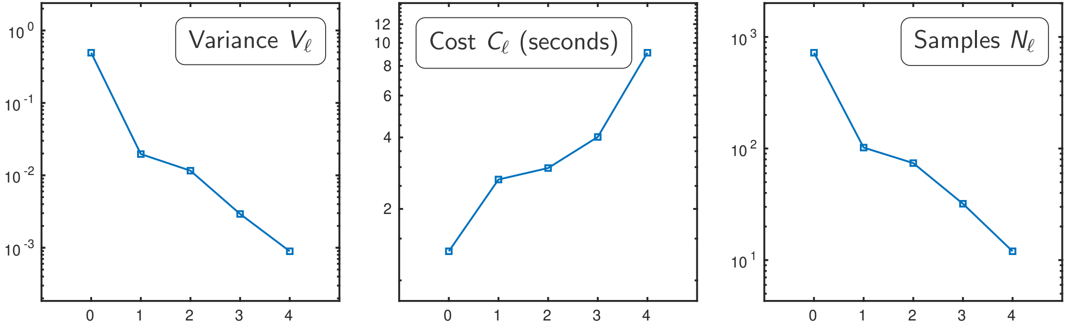


- Multilevel Monte Carlo simulation: $\mathcal{E}^{\mathrm{ML}}(q_w) = 1.2 \times 10^{-2}$

Example 2: High-Contrast Permeability

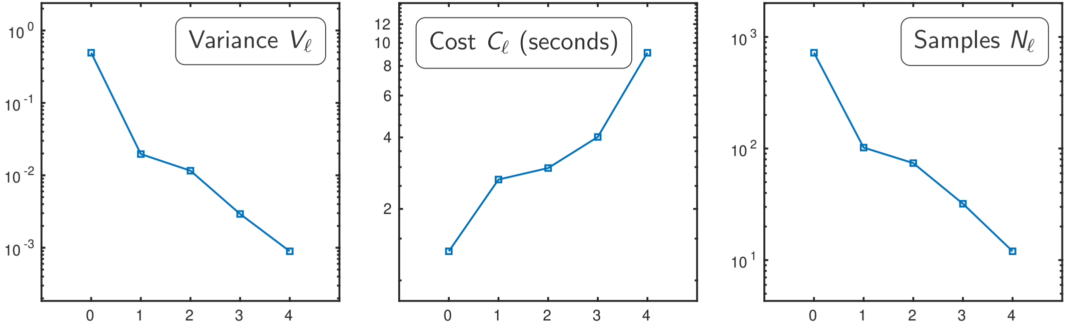


Example 2: High-Contrast Permeability



- Total cost of Multilevel Monte Carlo: $\sum_\ell N_\ell C_\ell \approx 1686$ s
- Total cost of Monte Carlo (assuming $C_4 =$ cost of $u_4 - u_3 \approx$ cost of u_4) ≈ 909 s
  $\rightarrow$ Similar variance with almost twice the cost

Example 2: High-Contrast Permeability



- Total cost of Multilevel Monte Carlo: $\sum_\ell N_\ell C_\ell \approx$ 1686 s
- Total cost of Monte Carlo (assuming C_4 = cost of $u_4 - u_3 \approx$ cost of u_4) $\approx$ 909 s
  $\rightarrow$ Similar variance with almost twice the cost

Concluding Remarks

A few pitfalls and shortcomings

Concluding Remarks

A few pitfalls and shortcomings

- Convergence theory with conditions in terms of unknown quantities
  - Number of samples needed to approximate V_ℓ, C_ℓ is problem-dependent

Concluding Remarks

A few pitfalls and shortcomings

- Convergence theory with conditions in terms of unknown quantities
  - Number of samples needed to approximate V_ℓ, C_ℓ is problem-dependent
- Use upscaling with care
  - Coarsest level $\ell = 0$ should have cell diameter $h \sim$ correlation length λ

Concluding Remarks

A few pitfalls and shortcomings

- Convergence theory with conditions in terms of unknown quantities
  - Number of samples needed to approximate V_ℓ, C_ℓ is problem-dependent
- Use upscaling with care
  - Coarsest level $\ell = 0$ should have cell diameter $h \sim$ correlation length λ
- Very challenging to upscale complex models in a meaningful way
  - Channelized reservoirs, different rock types, multiphase, etc.

    … and the best methods are expensive

Concluding Remarks

A few pitfalls and shortcomings

- Convergence theory with conditions in terms of unknown quantities
  - Number of samples needed to approximate V_ℓ, C_ℓ is problem-dependent
- Use upscaling with care
  - Coarsest level $\ell = 0$ should have cell diameter $h \sim$ correlation length λ
- Very challenging to upscale complex models in a meaningful way
  - Channelized reservoirs, different rock types, multiphase, etc.
    ... and the best methods are expensive
- Not all choices of u are appropriate!
  - Rule of thumb: average of u should "make sense" for the problem at hand

Concluding Remarks

Can we do better?

Concluding Remarks

Can we do better?

- *Level* does not necessarily mean spatial resolution!
  - Solver-based: use a more accurate solver for higher levels
    e.g., increasing accuracy of spatial/temporal discretization with level
  - Multiscale methods (compromise between upscaling and solver accuracy)

Concluding Remarks

Can we do better?

- *Level* does not necessarily mean spatial resolution!
  - Solver-based: use a more accurate solver for higher levels
    e.g., increasing accuracy of spatial/temporal discretization with level
  - Multiscale methods (compromise between upscaling and solver accuracy)
- MLMC does not require a geometric sequence of levels
  - Sufficient that accuracy and cost *increase* and variance *decrease* with ℓ

Concluding Remarks

Can we do better?

- *Level* does not necessarily mean spatial resolution!
  - Solver-based: use a more accurate solver for higher levels
    e.g., increasing accuracy of spatial/temporal discretization with level
  - Multiscale methods (compromise between upscaling and solver accuracy)
- MLMC does not require a geometric sequence of levels
  - Sufficient that accuracy and cost *increase* and variance *decrease* with ℓ
- Multi-index Monte Carlo – change multiple aspects of simulation with level
  - Example: resolution in space *and* time, $\ell \to \boldsymbol{\ell} = (\ell_{\boldsymbol{x}}, \ell_t)$

Suggested Literature

K. A. Cliffe, M. B. Giles, R. Scheichl, and A. L. Teckentrup. Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients. *Comput. Vis. Sci.*, 14(1):3–15, 2011. ISSN 14329360. doi: 10.1007/s00791-011-0160-x.

Y. Efendiev, O. Iliev, and C. Kronsbein. Multilevel Monte Carlo methods using ensemble level mixed MsFEM for two-phase flow and transport simulations. *Comput. Geosci.*, 17(5):833–850, 2013. ISSN 14200597. doi: 10.1007/s10596-013-9358-y.

M. B. Giles. Multilevel monte carlo methods. *Acta Numer.*, pages 259–328, 2015. doi: 10.1017/S096249291500001X.

F. Müller, P. Jenny, and D. W. Meyer. Multilevel Monte Carlo for two phase flow and Buckley-Leverett transport in random heterogeneous porous media. *J. Comput. Phys.*, 250:685–702, 2013. ISSN 10902716. doi: 10.1016/j.jcp.2013.03.023.

F. Müller, D. W. Meyer, and P. Jenny. Solver-based vs. grid-based multilevel Monte Carlo for two phase flow and transport in random heterogeneous porous media. *J. Comput. Phys.*, 268:39–50, 2014. ISSN 10902716. doi: 10.1016/j.jcp.2014.02.047. URL `http://dx.doi.org/10.1016/j.jcp.2014.02.047`.

A. L. Teckentrup, R. Scheichl, M. B. Giles, and E. Ullmann. Further analysis of multilevel Monte Carlo methods for elliptic PDEs with random coefficients. *Numer. Math.*, 125(3):569–600, 2013. ISSN 0029599X. doi: 10.1007/s00211-013-0546-4.

Developed by nuclear physicist Stanislaw Ulam during the Manhattan Project in the late 1940's

It was at that time that I suggested an obvious name for the statistical method – a suggestion not unrelated to the fact that Stan had an uncle who would borrow money from relatives because he "just had to go to Monte Carlo"

— Nicholas Metropolis, *The Beginning of the Monte Carlo Method* (1987)