A Multilevel Stochastic Collocation Method

Peter Jantsch

University of Tennessee, Knoxville

April 2, 2014

Joint work with Aretha Teckentrup (FSU), Clayton Webster (ORNL/UT), and Max Gunzburger (FSU)

DOE Advanced Simulation Computing Research (ASCR) — #ERKJ259

PDEs with Random Coefficients

- Stochastic sampling methods for random PDEs are computationally expensive: each sample point corresponds to a PDE solve.
- Traditionally, we try to improve single level methods by reducing the number of samples/solves: Quasi Monte Carlo, (anisotropic) sparse grids, adaptive grids
- For a given accuracy, multilevel methods seek to reduce the complexity by spreading computational cost evenly across several resolutions of the spatial discretization.

History of the Multilevel Method

Multilevel methods for SPDEs derive from multigrid methods for the FEM, and have been used most commonly in the context of Monte Carlo methods:

- Multilevel Monte Carlo for numerical integration (S. Heinrich, 2001)
- Multilevel Monte Carlo path simulations for computational finance (M. Giles, 2008)
- Since applied to a variety of SPDEs, MLQMC, etc.

Model Problem - Linear Elliptic SPDE

Find $u \in L^2_\rho(\Gamma, H^1_0(D))$ such that for almost every $\mathbf{y} \in \Gamma$

$$\nabla \cdot (a(\mathbf{y}, x) \cdot \nabla u(\mathbf{y}, x)) = f(\mathbf{y}, x) \tag{1}$$

We assume that a, f are such that this problem has a unique solution represented in terms of $\mathbf{y} \in \Gamma$, a finite dimensional random vector.

Model Problem - Linear Elliptic SPDE

Find $u \in L^2_\rho(\Gamma, H^1_0(D))$ such that for almost every $\mathbf{y} \in \Gamma$

$$\nabla \cdot (a(\mathbf{y}, x) \cdot \nabla u(\mathbf{y}, x)) = f(\mathbf{y}, x) \tag{1}$$

We assume that a, f are such that this problem has a unique solution represented in terms of $\mathbf{y} \in \Gamma$, a finite dimensional random vector.

Such a PDE might represent ground water flow, etc.

Model Problem - Linear Elliptic SPDE

Find $u \in L^2_\rho(\Gamma, H^1_0(D))$ such that for almost every $\mathbf{y} \in \Gamma$

$$\nabla \cdot (a(\mathbf{y}, x) \cdot \nabla u(\mathbf{y}, x)) = f(\mathbf{y}, x) \tag{1}$$

We assume that a, f are such that this problem has a unique solution represented in terms of $\mathbf{y} \in \Gamma$, a finite dimensional random vector.

Such a PDE might represent ground water flow, etc.

Common Single Level Methods

Monte Carlo Method

- Most popular method
- Simple to implement, easily parallelizable
- Convergence rate $\mathcal{O}(M^{-1/2})$ is dimension independant, but relatively slow

Spectral Galerkin Methods

- Higher rate of convergence
- Degrees of freedom are coupled, leading to a large linear system
- Suffers from the curse of dimensionality

Stochastic Collocation

For stochastic collocation we choose a set of (interpolatory) points $\{\mathbf{y}_j\}_{j=1}^{M} \subset \Gamma$, and for each $\mathbf{y}_j$ solve the deterministic PDE

$$\nabla \cdot (a(\mathbf{y}_j, x) \cdot \nabla u(\mathbf{y}_j, x)) = f(\mathbf{y}_j, x), \tag{2}$$

using the finite element method to obtain a solution $u_h(\mathbf{y}_j, x)$.

Stochastic Collocation

For stochastic collocation we choose a set of (interpolatory) points $\{\mathbf{y}_j\}_{j=1}^M \subset \Gamma$, and for each $\mathbf{y}_j$ solve the deterministic PDE

$$\nabla \cdot (a(\mathbf{y}_j, x) \cdot \nabla u(\mathbf{y}_j, x)) = f(\mathbf{y}_j, x), \tag{2}$$

using the finite element method to obtain a solution $u_h(\mathbf{y}_j, x)$. Finally, we construct our approximation by interpolation:

$$\mathcal{I}_M u_h(\mathbf{y}, x) = \sum_{j=1}^{M} c_j(x) \Psi_j(\mathbf{y}) \tag{3}$$

Stochastic Collocation

For stochastic collocation we choose a set of (interpolatory) points $\{\mathbf{y}_j\}_{j=1}^M \subset \Gamma$, and for each $\mathbf{y}_j$ solve the deterministic PDE

$$\nabla \cdot (a(\mathbf{y}_j, x) \cdot \nabla u(\mathbf{y}_j, x)) = f(\mathbf{y}_j, x), \tag{2}$$

using the finite element method to obtain a solution $u_h(\mathbf{y}_j, x)$. Finally, we construct our approximation by interpolation:

$$\mathcal{I}_M u_h(\mathbf{y}, x) = \sum_{j=1}^{M} c_j(x) \Psi_j(\mathbf{y}) \tag{3}$$

For this scheme, we need to solve M systems of size n_h. For high dimensional spaces Γ, the number of points M needed to obtain a good approximation can be huge!

Main Idea: Suppose we have a sequence of finite element solutions $u_{h_k}(\mathbf{y}) \in V_{h_k}$, (with $u_{-1} = 0$). Multilevel methods are based on the following simple identity:

$$u_{h_K}(\mathbf{y}) = \sum_{k=0}^{K} u_{h_k}(\mathbf{y}) - u_{h_{k-1}}(\mathbf{y}).$$

Main Idea: Suppose we have a sequence of finite element solutions $u_{h_k}(\mathbf{y}) \in V_{h_k}$, (with $u_{-1} = 0$). Multilevel methods are based on the following simple identity:

$$u_{h_K}(\mathbf{y}) = \sum_{k=0}^{K} u_{h_k}(\mathbf{y}) - u_{h_{k-1}}(\mathbf{y}).$$

With Monte Carlo methods, we can compute expectation using sample averages:

$$\mathbb{E}(u_{h_K}(\mathbf{y})) \approx u_{h_K}^{MLMC} = \sum_{k=0}^{K} \frac{1}{M_{K-k}} \sum_{j=1}^{M_{K-k}} \left(u_{h_k}(\mathbf{y}_j) - u_{h_{k-1}}(\mathbf{y}_j) \right).$$

For stochastic collocation, we interpolate the differences at different resolutions. Suppose we have a sequence of interpolation operators $\{\mathcal{I}_{l_k}\}$ with increasing approximation properties. Now the multilevel approximation is given by:

$$u_K^{ML} = \sum_{k=0}^{K} \mathcal{I}_{l_{K-k}} \left(u_{h_k} - u_{h_{k-1}} \right).$$

For stochastic collocation, we interpolate the differences at different resolutions. Suppose we have a sequence of interpolation operators $\{\mathcal{I}_{l_k}\}$ with increasing approximation properties. Now the multilevel approximation is given by:

$$u_K^{ML} = \sum_{k=0}^{K} \mathcal{I}_{l_{K-k}} \left(u_{h_k} - u_{h_{k-1}} \right).$$

We can also approximate functionals of the solution $\varphi[u]$ in a similar way:

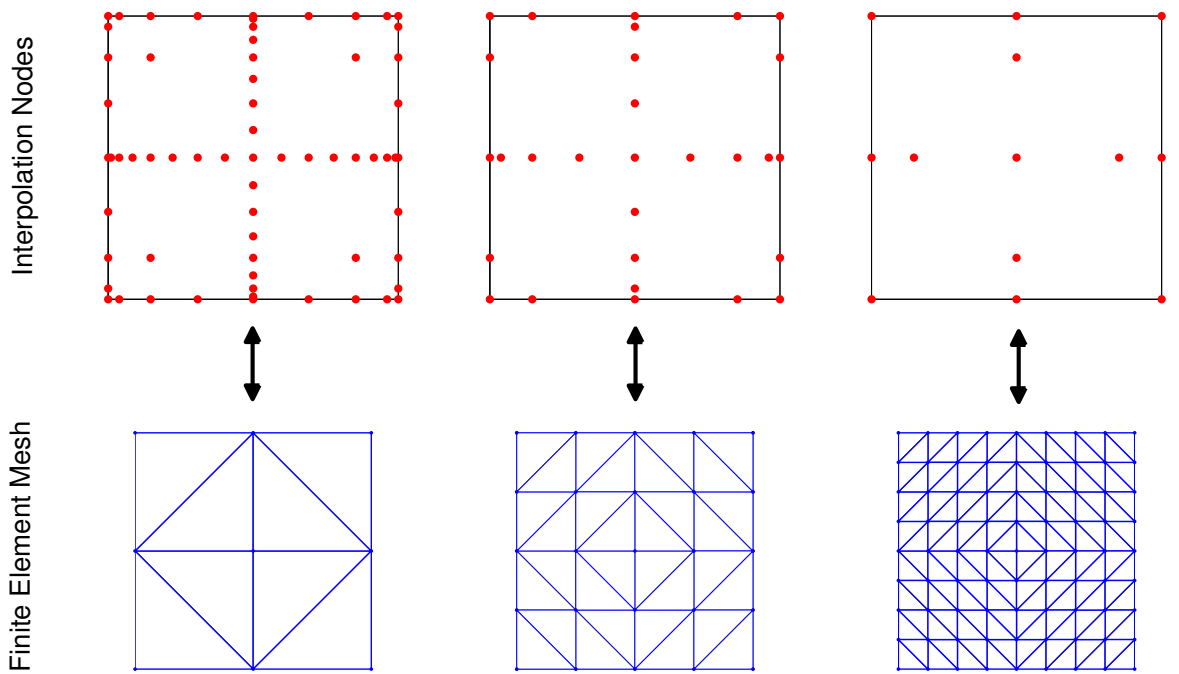
$$\varphi_K^{ML}[u] = \sum_{k=0}^{K} \mathcal{I}_{l_{K-k}} \left(\varphi[u_{h_k}] - \varphi[u_{h_{k-1}}] \right).$$

$$u_K^{ML} = \sum_{k=0}^{K} \mathcal{I}_{l_{K-k}} \left(u_{h_k} - u_{h_{k-1}} \right), \qquad u_{h_{-1}} = 0.$$

- **k = 0:** $\mathcal{I}_{l_K}[u_{h_0}]$

  $\vdots$

- **k = K:** $\mathcal{I}_{l_0}[u_{h_K} - u_{h_{K-1}}]$



Error Splitting

We examine the method by considering the discretization errors independently:

$$\|u - u_K^{ML}\| \leq \underbrace{\|u - u_{h_K}\|}_{I} + \underbrace{\|u_{h_K} - u_K^{ML}\|}_{II} \leq \varepsilon.$$

Error Splitting

We examine the method by considering the discretization errors independently:

$$\|u - u_K^{ML}\| \leq \underbrace{\|u - u_{h_K}\|}_{I} + \underbrace{\|u_{h_K} - u_K^{ML}\|}_{II} \leq \varepsilon.$$

Now to compute the computational cost, we assume that the spatial discretization converges in h as

$$I \leq C_s h_K^{\alpha},$$

The term II can be further split apart using the triangle inequality:

$$
\begin{aligned}
II &= \| \sum_{k=0}^{K} (u_{h_k} - u_{h_{k-1}}) - \mathcal{I}_{l_{K-k}} (u_{h_k} - u_{h_{k-1}}) \| \\
&\leq \sum_{k=0}^{K} \| (\mathrm{I} - \mathcal{I}_{l_{K-k}}) (u_{h_k} - u_{h_{k-1}}) \|.
\end{aligned}
$$

Assume that the stochastic interpolation operators converge according to

$$
\begin{aligned}
\| (\mathrm{I} - \mathcal{I}_{l_{K-k}}) (u_{h_k} - u_{h_{k-1}}) \| &\leq C_I \, M_{K-k}^{-\mu} \, h_k^{\beta}, \\
\implies II &\leq \sum_{k=0}^{K} C_I \, M_{K-k}^{-\mu} \, h_k^{\beta}.
\end{aligned}
$$

Error Balancing

Finally, we compute the cost of the multilevel method using the metric

$$Cost = \sum_{k=0}^{K} M_{K-k}\, C_k^{FEM} \approx \sum_{k=0}^{K} M_{K-k}\, h_k^{-\gamma}. \tag{4}$$

Error Balancing

Finally, we compute the cost of the multilevel method using the metric

$$Cost = \sum_{k=0}^{K} M_{K-k}\, C_k^{FEM} \approx \sum_{k=0}^{K} M_{K-k}\, h_k^{-\gamma}. \tag{4}$$

We can ensure $I \leq \varepsilon/2$ by requiring

$$C_s\, h_K^{\alpha} \leq \frac{\varepsilon}{2}$$

This fixes the mesh maximum mesh size (and the lower level meshes).

Error Balancing

Finally, we compute the cost of the multilevel method using the metric

$$Cost = \sum_{k=0}^{K} M_{K-k}\, C_k^{FEM} \eqsim \sum_{k=0}^{K} M_{K-k}\, h_k^{-\gamma}. \tag{4}$$

We can ensure $I \leq \varepsilon/2$ by requiring

$$C_s\, h_K^{\alpha} \leq \frac{\varepsilon}{2}$$

This fixes the mesh maximum mesh size (and the lower level meshes). Then we may choose the interpolation operators $\mathcal{I}_{l_k}$ to satisfy

$$II \leq \sum_{k=0}^{K} C_I\, M_{K-k}^{-\mu}\, h_k^{\alpha} \leq \frac{\varepsilon}{2}.$$

and such that they minimize the computational cost.

Theorem: [Gunzburger, J, Teckentrup, Webster]

Under our assumptions, for any $\varepsilon > 0$ there exists an integer K such that

$$\|u - u_K^{ML}\|_{L^2_\rho(\Gamma;H^1_0(D))} \leq \varepsilon$$

and

$$Cost_\varepsilon^{ML} \lesssim \begin{cases} \varepsilon^{-\frac{1}{\mu}} & \text{if } \beta > \mu\gamma, \\ \varepsilon^{-\frac{1}{\mu}} \left|\log \varepsilon\right|^{1+\frac{1}{\mu}} & \text{if } \beta = \mu\gamma, \\ \varepsilon^{-\frac{1}{\mu} - \frac{\gamma\mu - \beta}{\alpha\mu}} & \text{if } \beta < \mu\gamma. \end{cases} \tag{5}$$

Theorem: [Gunzburger, J, Teckentrup, Webster]

Under our assumptions, for any $\varepsilon > 0$ there exists an integer K such that

$$\|u - u_K^{ML}\|_{L^2_\rho(\Gamma;H^1_0(D))} \leq \varepsilon$$

and

$$Cost_\varepsilon^{ML} \lesssim \begin{cases} \varepsilon^{-\frac{1}{\mu}} & \text{if } \beta > \mu\gamma, \\ \varepsilon^{-\frac{1}{\mu}} \left|\log \varepsilon\right|^{1+\frac{1}{\mu}} & \text{if } \beta = \mu\gamma, \\ \varepsilon^{-\frac{1}{\mu} - \frac{\gamma\mu - \beta}{\alpha\mu}} & \text{if } \beta < \mu\gamma. \end{cases} \tag{5}$$

Compare to standard, single level SC:

$$Cost_\varepsilon^{SL} \eqsim h^{-\gamma} M \eqsim \varepsilon^{-\gamma/\alpha - 1/\mu}.$$

Case	$\beta > \mu\gamma$	$\beta = \mu\gamma$	$\beta = \alpha < \mu\gamma$
Cost Reduction	$\varepsilon^{-\gamma/\alpha}$	$\eqsim \varepsilon^{-\gamma/\alpha}$	$\varepsilon^{-1/\mu}$

Two Implementation Challenges

- First, the multilevel method predicts a number of points, but not an appropriate sparse grid. Thus, we may use more many points than necessary.

Two Implementation Challenges

- First, the multilevel method predicts a number of points, but not an appropriate sparse grid. Thus, we may use more many points than necessary.

  We could ameliorate this issue by using adaptive grids, solving a discrete optimization problem for grid levels, or various *ad hoc* rounding strategies. However, we still see gains in efficiency in the "worst" case.

Two Implementation Challenges

- First, the multilevel method predicts a number of points, but not an appropriate sparse grid. Thus, we may use more many points than necessary.

  We could ameliorate this issue by using adaptive grids, solving a discrete optimization problem for grid levels, or various *ad hoc* rounding strategies. However, we still see gains in efficiency in the "worst" case.

- Secondly, the method relies on *a priori* knowledge of the convergence rates and parameters $\alpha, \beta, \mu, \gamma$. We need a practical implementation of the method. (See *MLMC path simulation*, Giles, 2008).

Example Problem:

As an example, we consider the following boundary value problem on either $D = (0,1)$ or $D = (0,1)^2$:

$$\begin{aligned} -\nabla \cdot (a(\omega, x)\nabla u(\omega, x)) &= 1, && \text{for } x \in D, \\ u(\omega, x) &= 0, && \text{for } x \in \partial D\,. \end{aligned}$$

Example Problem:

As an example, we consider the following boundary value problem on either $D = (0,1)$ or $D = (0,1)^2$:

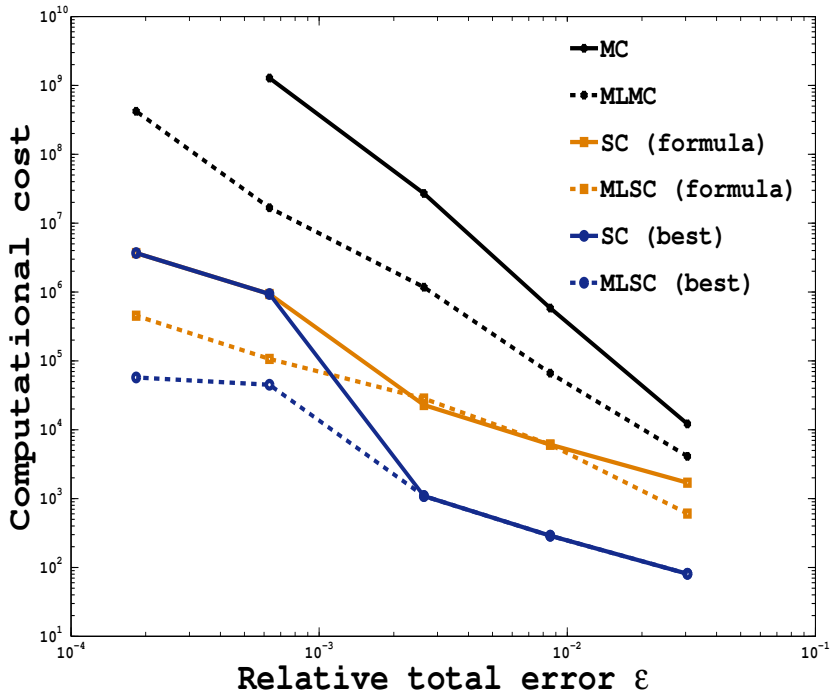
$$\begin{aligned} -\nabla \cdot (a(\omega, x)\nabla u(\omega, x)) &= 1, \qquad && \text{for } x \in D, \\ u(\omega, x) &= 0, && \text{for } x \in \partial D\,. \end{aligned}$$

We take the coefficient a to be of the form

$$a(\omega, x) = 0.5 + \exp\left[\sum_{n=1}^{N} \sqrt{\lambda_n} b_n(x) Y_n(\omega)\right],$$

where $\{Y_n\}_{n\in\mathbb{N}}$ is a sequence of independent, uniformly distributed random variables on [-1,1], and $\{\lambda_n\}_{n\in\mathbb{N}}$ and $\{b_n\}_{n\in\mathbb{N}}$ are the eigenvalues and eigenfunctions, resp., of the covariance operator with kernel function $C(x,y) = \exp[-\|x-y\|_1]$.

Results in 10D



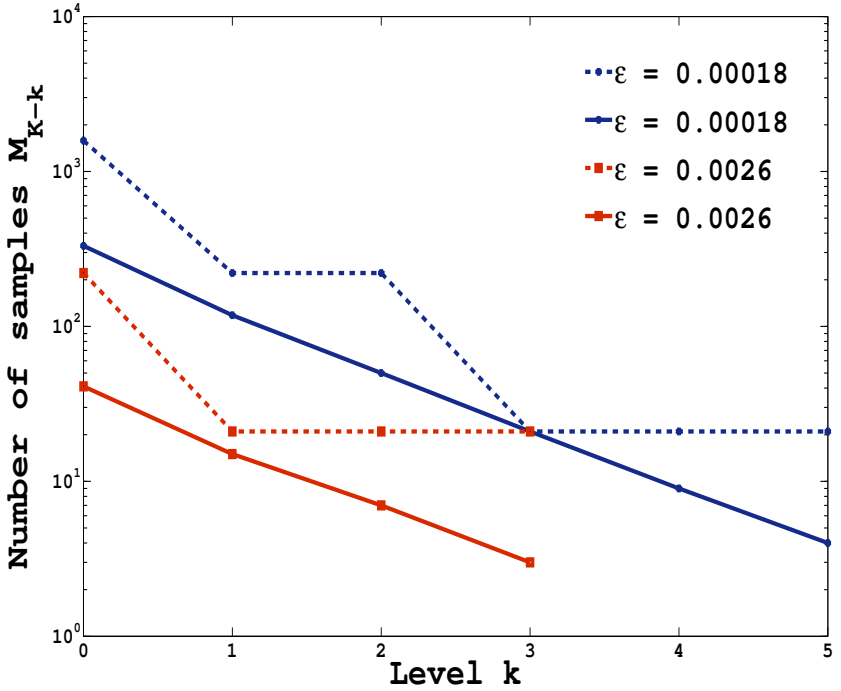
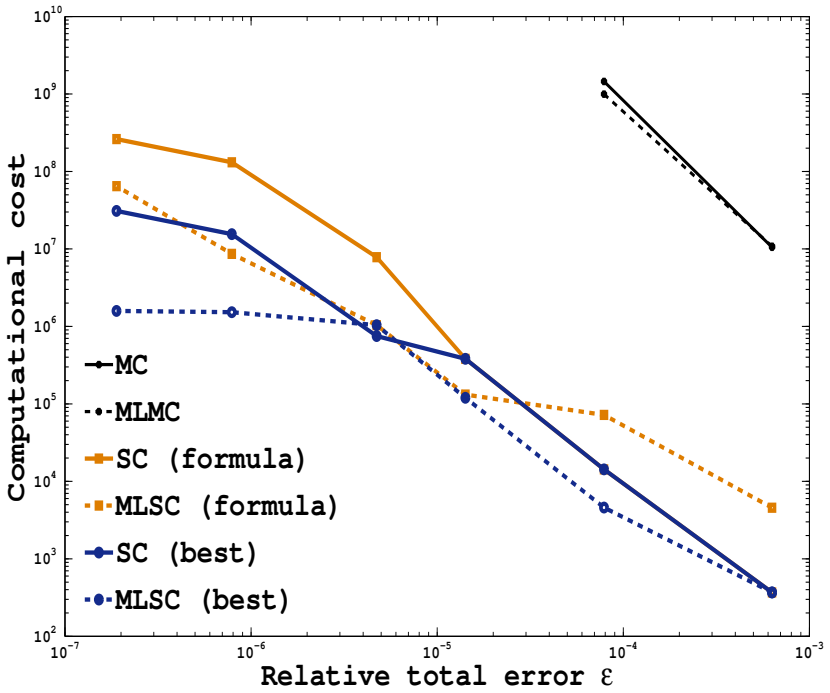


Figure : Left: Cost versus Error for $D = (0, 1)^2$, $N = 10$. Right: Number of samples per level (predicted vs actual).

Results in 20D



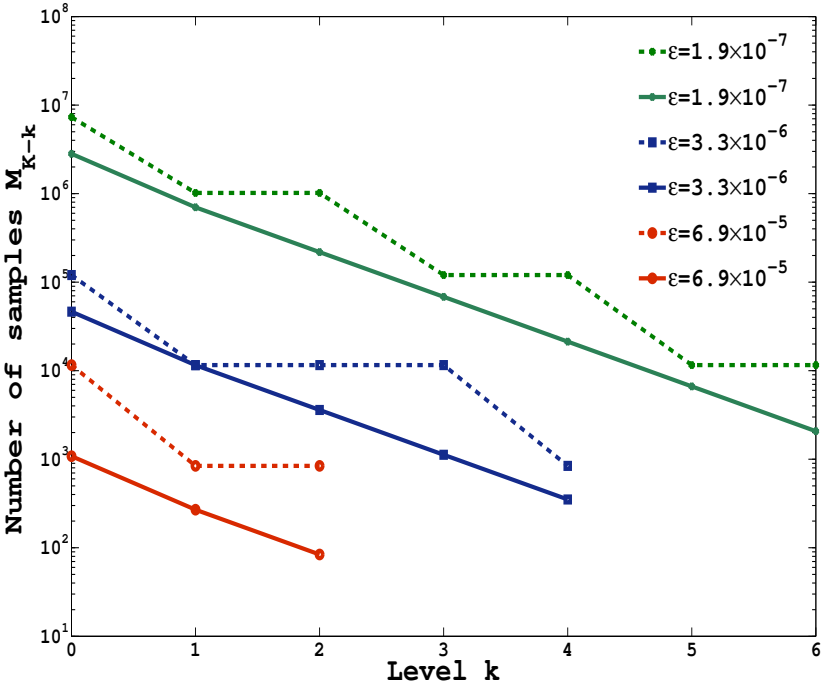


Figure : Left figures: Cost versus Error for $D = (0, 1)$, $N = 20$. Right figures: Number of samples per level (predicted vs actual).

Multilevel methods:

- Can be practically applied to SC methods based on sparse grids
- Reduce computational cost for a variety of stochastic sampling methods for SPDEs.
- Work to counteract the curse of dimensionality.
- Effective when applied to SC schemes even up to 20D.