



LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

MULTILEVEL MONTE CARLO (MLMC) SIMULATIONS: PERFORMANCE RESULTS FOR SPE10 (XY SLICES)

D. Kalchev, P. S. Vassilevski

February 26, 2016

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

**MULTILEVEL MONTE CARLO (MLMC) SIMULATIONS: PERFORMANCE RESULTS
FOR SPE10 (XY SLICES)**

DELYAN KALCHEV AND PANAYOT S. VASSILEVSKI

ABSTRACT. In this report we first describe a generic multilevel Monte Carlo method and then illustrate its superior performance over a traditional single-level Monte Carlo method for second order elliptic PDEs corresponding to two-dimensional layers in (x, y) -direction of the Tenth SPE Comparative Solution project (SPE 10) which gives high-contrast permeability coefficients. The SPE10 data set is used as a coarse level in the Monte Carlo method and the respective permeability coefficient k (provided in the SPE10 dataset) is used as a mean in the simulation. The actual coefficients are drawn based on a KL-expansion assuming that the log-mean is perturbed by a log-normal distributed samples.

1. A GENERIC MULTILEVEL MONTE CARLO METHOD

We present here a short description of the multilevel Monte Carlo (MLMC) method originated in its present form in [2] (although these ideas have been around since the introduction of multigrid, cf. e.g., [1]). The exposition below is based on [3] for verification purposes.

Let $\mathbf{X}_M : \Omega \rightarrow \mathbb{R}^M$ be a random vector over some probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and consider a quantity of interest $Q_M = \mathcal{G}(\mathbf{X}_M)$, for a given functional $\mathcal{G}(\cdot)$. Assume also that $\mathbb{E}[Q_M]$ can be made arbitrarily close to $\mathbb{E}[Q]$ by choosing M sufficiently large. Our goal is to efficiently estimate $\mathbb{E}[Q_M]$ (which approximates the unknown value $\mathbb{E}[Q]$). For this purpose, we compute an estimator \widehat{Q}_M and quantify its accuracy using the root mean square error (RMSE)

$$e(\widehat{Q}_M) = \left(\mathbb{E} \left[\left(\widehat{Q}_M - \mathbb{E}[Q] \right)^2 \right] \right)^{1/2}.$$

The quantity $e^2(\widehat{Q}_M)$ is sometimes called mean square error (MSE). For some given small parameter ε , we construct an estimator which is accurate enough so that $e(\widehat{Q}_M) \leq \varepsilon$. The computational cost, $\mathcal{C}_\varepsilon(\widehat{Q}_M)$, of achieving this accuracy is called ε -cost.

1.1. Monte Carlo simulation. The standard Monte Carlo (MC) estimator is defined as

$$\widehat{Q}_{M,N}^{\text{MC}} = \frac{1}{N} \sum_{i=1}^N Q_M^{(i)},$$

where $Q_M^{(i)}$, $i = 1, \dots, N$, are independent samples of Q_M . We assume that the cost of computing one sample is $\mathcal{C}(Q_M^{(i)}) = \mathcal{O}(M^\gamma)$, for some $\gamma > 0$. The optimal cost is when $\gamma = 1$.

We have for the error

$$(1.1.1) \quad e^2(\widehat{Q}_{M,N}^{\text{MC}}) = \mathbb{V}[\widehat{Q}_{M,N}^{\text{MC}}] + (\mathbb{E}[Q_M - Q])^2 = N^{-1}\mathbb{V}[Q_M] + (\mathbb{E}[Q_M - Q])^2.$$

Date: August 20, 2012–beginning; Today is December 16, 2013.

1991 Mathematics Subject Classification. 65F10, 65N20, 65N30.

Key words and phrases. multilevel Monte Carlo, KL modes, SPE10.

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

The first term in (1.1.1) is the variance of the estimator. In the applications we consider, M is a spatial discretization parameter and Q_M is a quantity that approximates the inaccessible quantity Q . Q_M is computed by solving a discretized PDE (partial differential equation) whereas Q corresponds to the solution of the respective continuous PDE. Thus, the second term in (1.1.1) represents the error due to discretization and represents a bias of the estimator. We concentrate on the first term in the MSE (1.1.1) and we note that the second term can be analyzed in some model situations ([3]). Under the assumption that M is sufficiently large we can assume that $(\mathbb{E}[Q_M - Q])^2 \leq \varepsilon^2/2$. Then, $N^{-1}\mathbb{V}[Q_M] \leq \varepsilon^2/2$ is sufficient for the desired $e(\widehat{Q}_{M,N}^{\text{MC}}) \leq \varepsilon$. Clearly, the ε -cost for achieving this is

$$C_\varepsilon(\widehat{Q}_{M,N}^{\text{MC}}) \simeq \frac{2\mathbb{V}[Q_M]\mathcal{C}(Q_M^{(i)})}{\varepsilon^2}.$$

1.2. Multilevel Monte Carlo simulation. Neglecting the bias term (the second term) in (1.1.1), we see that the RMSE of the Monte Carlo estimator, as an approximation of $\mathbb{E}[Q_M]$, is of order $N^{-1/2}$. This reflects the (very) slow convergence of the standard Monte Carlo method – a main disadvantage additionally amplified by the high cost of sampling from Q_M which involves solving discretized PDE for each sample (which can be a challenging task on its own).

Here we present the multilevel Monte Carlo (MLMC) method as a variance reduction technique. The latter means that for a fixed cost the MLMC method has the ability to produce approximations with lower variance compared to the standard Monte Carlo method (for the same cost). This is achieved by reducing the cost of the simulation using multiple approximations of Q on a hierarchy of levels. In contrast, the standard Monte Carlo uses one approximation, Q_M . In terms of our PDE applications this means that multiple (coarse) levels (meshes) are used for sampling instead of using only the finest one. The heuristic behind this approach is that the uncertainty can be captured accurately enough on coarse grids. Thus, the majority of the samples can be produced for lower cost on coarser levels and only a little correction between levels is necessary resulting in an overall smaller number of fine-level (high-cost) samples.

Consider a sequence of discretization parameters $\{M_\ell \in \mathbb{N} : \ell = 0, \dots, L \text{ and } M_0 < M_1 < \dots < M_L = M\}$, and the corresponding quantities $\{Q_{M_\ell}\}_{\ell=0}^L$. Let $s \geq 2$ be the coarsening factor, i.e. $M_\ell \simeq sM_{\ell-1}$, for $\ell = 1, \dots, L$. Define $Y_\ell = Q_{M_\ell} - Q_{M_{\ell-1}}$, for $\ell = 1, \dots, L$, and $Y_0 = Q_{M_0}$. Using the additivity of the expectation, \mathbb{E} , we get the following telescoping sum

$$(1.2.1) \quad \mathbb{E}[Q_M] = \mathbb{E}[Q_{M_0}] + \sum_{\ell=1}^L \mathbb{E}[Q_{M_\ell} - Q_{M_{\ell-1}}] = \sum_{\ell=0}^L \mathbb{E}[Y_\ell].$$

The terms $\mathbb{E}[Y_\ell]$ in (1.2.1) are approximated using standard MC estimators \widehat{Y}_ℓ . Namely,

$$(1.2.2) \quad \widehat{Y}_\ell = \frac{1}{N_\ell} \sum_{i=1}^{N_\ell} Y_\ell^{(i)} = \frac{1}{N_\ell} \sum_{i=1}^{N_\ell} (Q_{M_\ell}^{(i)} - Q_{M_{\ell-1}}^{(i)}), \text{ for } \ell = 1, \dots, L$$

and

$$\widehat{Y}_0 = \frac{1}{N_0} \sum_{i=1}^{N_0} Y_0^{(i)} = \frac{1}{N_0} \sum_{i=1}^{N_0} Q_{M_0}^{(i)},$$

where N_ℓ , $\ell = 0, \dots, L$, are the numbers of samples on the respective level. Note that the estimators (1.2.2) require sampling from $Y_\ell = Q_{M_\ell} - Q_{M_{\ell-1}}$, which formally means that the quantities $Q_{M_\ell}^{(i)}$ and $Q_{M_{\ell-1}}^{(i)}$ in $Y_\ell^{(i)} = Q_{M_\ell}^{(i)} - Q_{M_{\ell-1}}^{(i)}$ are computed using the same random sample.

Finally, the multilevel Monte Carlo estimator is defined as

$$\widehat{Q}_M^{\text{ML}} = \sum_{\ell=0}^L \widehat{Y}_\ell.$$

Similarly, for the MSE, we have

$$(1.2.3) \quad e^2(\widehat{Q}_M^{\text{ML}}) = \mathbb{V}[\widehat{Q}_M^{\text{ML}}] + (\mathbb{E}[Q_M - Q])^2 = \sum_{\ell=0}^L N_\ell^{-1} \mathbb{V}[Y_\ell] + (\mathbb{E}[Q_M - Q])^2.$$

Like before, the first term in (1.2.3) is the variance of the estimator and the second term is the bias which is the same as the bias term in (1.1.1). That is, $\widehat{Q}_M^{\text{ML}}$ is a biased estimator of $\mathbb{E}[Q]$ with the same bias as the standard MC estimator. However, (like in the standard MC) $\widehat{Q}_M^{\text{ML}}$ is an unbiased estimator of $\mathbb{E}[Q_M]$.

The major source of **cost reduction** is the fact that $\mathbb{V}[Y_\ell]$ becomes **smaller** as M_ℓ becomes finer. Thus, **fewer finer (higher cost)** samples are necessary to achieve the desired magnitude of the error.

Denote by $\mathcal{C}_\ell = \mathcal{C}(Y_\ell^{(i)})$ the cost of computing a single sample of Y_ℓ . Note that, for $\ell \geq 1$, \mathcal{C}_ℓ includes sampling once on level ℓ and once on $\ell - 1$. Thus, we can estimate \mathcal{C}_ℓ using the sum $M_\ell^\gamma + M_{\ell-1}^\gamma$ and also M_0^γ can be used as an estimate of \mathcal{C}_0 .

Based on optimality considerations ([3]), we can assume that the order of N_ℓ should be $\sqrt{\mathbb{V}[Y_\ell]/\mathcal{C}_\ell}$, i.e. $N_\ell \approx \sqrt{\mathbb{V}[Y_\ell]/\mathcal{C}_\ell}$. Hence, $N_\ell^{-1} \mathbb{V}[Y_\ell] \approx \sqrt{\mathbb{V}[Y_\ell]\mathcal{C}_\ell}$ and

$$\sum_{\ell=0}^L N_\ell^{-1} \mathbb{V}[Y_\ell] \approx \sum_{\ell=0}^L \sqrt{\mathbb{V}[Y_\ell]\mathcal{C}_\ell}.$$

The unknown $\mathbb{V}[Y_\ell]$ can be estimated using the unbiased sample variance estimator (using the fact that the samples $Y_\ell^{(i)}$ are independent)

$$\mathbb{V}[Y_\ell] \approx s_{Y_\ell}^2 = \frac{1}{N_\ell - 1} \sum_{i=1}^{N_\ell} (Y_\ell^{(i)} - \widehat{Y}_\ell)^2 = \frac{1}{N_\ell - 1} \left(\sum_{i=1}^{N_\ell} (Y_\ell^{(i)})^2 - N_\ell \widehat{Y}_\ell^2 \right).$$

1.3. The MLMC for a model problem. Consider the PDE

$$(1.3.1) \quad -\nabla \cdot (k(\mathbf{x}, \omega) \nabla p(\mathbf{x}, \omega)) = f(\mathbf{x}) \text{ in } D \times \Omega,$$

subject to suitable boundary conditions. Here $D \subset \mathbb{R}^2$ is a rectangular spatial domain. Both the coefficient (the hydraulic conductivity) k and the solution p are random fields on $D \times \Omega$. We discretize (1.3.1) using the finite element method and thus obtain discrete solution p^h . Particularly, we use bilinear finite elements on rectangular meshes and, also, for the multilevel estimator we use standard geometric levels constructed by uniform mesh refinement.

The quantity Q is generally some functional of the solution p and the coefficient k . Similarly, the quantity Q_M is the same functional of the discrete solution p^h and the possibly also discretized coefficient k . Then, the parameter M represents the number of degrees of freedom (dofs) of the finite element space. Note that sampling from Q_M , in the general case, requires numerical solution of the discretized PDE, which is the reason for the high cost of drawing samples of Q_M . Also, obtaining samples of the solution in turn requires sampling from the coefficient.

A popular choice is to model k as a log-normal random field with a two-point correlation structure. To this purpose, we use the following two-point covariance function

$$(1.3.2) \quad C(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(-\frac{|x_1 - x'_1|}{\lambda_1} - \frac{|x_2 - x'_2|}{\lambda_2}\right), \text{ for } \mathbf{x}, \mathbf{x}' \in D,$$

where σ^2 is the variance of the stochastic field, λ_j is the correlation length in the j^{th} direction, and the notation $\mathbf{x} = [x_1, x_2]^T$ is used.

Consider the Gaussian random field $Z(\mathbf{x}, \omega) = \ln[k(\mathbf{x}, \omega)]$. Using the Karhunen-Loève (KL) expansion ([4]), we have

$$(1.3.3) \quad Z(\mathbf{x}, \omega) = \mathbb{E}[Z(\mathbf{x}, \cdot)] + \sum_{n=1}^{\infty} \sqrt{\theta_n} \xi_n(\omega) b_n(\mathbf{x}),$$

where $\{\xi_n\}_{n \in \mathbb{N}}$ are independent standard normal random variables and $(\theta_n, b_n)_{n \in \mathbb{N}}$ are eigenpairs of the integral equation

$$(1.3.4) \quad \int_D C(\mathbf{x}, \mathbf{x}') b_n(\mathbf{x}') d\mathbf{x}' = \theta_n b_n(\mathbf{x}), \text{ for } \mathbf{x} \in D,$$

and $\|b_n\|_{L_2(D)} = 1$, for $n \in \mathbb{N}$. Note that $\mathbb{E}[Z(\mathbf{x}, \cdot)]$ in (1.3.3) is a function of \mathbf{x} and represents the mean of the random field Z .

In practice we use a truncated version of the KL-expansion where a finite number, m_{KL} , of terms of the expansion are computed. That is, we have

$$(1.3.5) \quad Z_{m_{\text{KL}}}(\mathbf{x}, \omega) = \mathbb{E}[Z(\mathbf{x}, \cdot)] + \sum_{n=1}^{m_{\text{KL}}} \sqrt{\theta_n} \xi_n(\omega) b_n(\mathbf{x}),$$

where the first m_{KL} eigenpairs with dominating eigenvalues θ_n , $n \leq m_{\text{KL}}$, are taken. Thus, drawing (approximate) samples of k requires simply drawing m_{KL} independent samples from standard normal distribution for $\{\xi_n\}_{n=1}^{m_{\text{KL}}}$, then evaluating $Z_{m_{\text{KL}}}$ from (1.3.5), and finally computing the sampled coefficient

$$k_{m_{\text{KL}}} = \exp[Z_{m_{\text{KL}}}] .$$

Note that the number of terms in the truncated KL-expansion (1.3.5) affects the bias ([3]). Therefore, to have a good approximation, large m_{KL} is required (related to the choice of the discretization parameter M).

To perform the above computation we need the eigenpairs of (1.3.4). For our special choice of kernel (1.3.2) the eigenpairs can be computed analytically ([3, 4]) by using separation of variables reducing to one-dimensional computations. More specifically, let $\sigma^2 = 1$ and denote the intervals $(0, a)$ by I_a , $a > 0$. Then, in the 1D case $D = I_1$ we have

$$C_{(I_1, \lambda)}(x, x') = \exp\left(-\frac{|x - x'|}{\lambda}\right), \text{ for } x, x' \in I_1,$$

and

$$\int_0^1 C_{(I_1, \lambda)}(x, x') b_n^{(I_1, \lambda)}(x') dx' = \theta_n^{(I_1, \lambda)} b_n^{(I_1, \lambda)}(x), \text{ for } x \in I_1,$$

where

$$\theta_n^{(I_1, \lambda)} = \frac{2\lambda}{\lambda^2 \omega_n^2 + 1},$$

$$b_n^{(I_1, \lambda)}(x) = A_n (\sin(\omega_n x) + \lambda \omega_n \cos(\omega_n x)), \text{ for } x \in I_1.$$

Here A_n is a normalization constant so that $\|b_n^{(I_1, \lambda)}\|_{L_2(0,1)} = 1$ and $\{\omega_n\}_{n \in \mathbb{N}}$ are the real solutions of the transcendental equation

$$\tan(\omega) = \frac{2\lambda\omega}{\lambda^2\omega^2 - 1}.$$

Consider now the case $D = I_a$ and the transformation $x = a\xi$, where $\xi \in I_1$ and $x \in I_a$. Clearly, we have

$$C_{(I_a, \lambda)}(x, x') = C_{(I_a, \lambda)}(a\xi, a\xi') = C_{(I_1, \lambda/a)}(\xi, \xi').$$

We are interested in the following eigenproblem

$$\int_0^a C_{(I_a, \lambda)}(x, x') b_n^{(I_a, \lambda)}(x') dx' = \theta_n^{(I_a, \lambda)} b_n^{(I_a, \lambda)}(x), \text{ for } x \in I_a.$$

Since

$$\int_0^a C_{(I_a, \lambda)}(x, x') b_n^{(I_a, \lambda)}(x') dx' = a \int_0^1 C_{(I_1, \lambda/a)}(\xi, \xi') b_n^{(I_a, \lambda)}(a\xi') d\xi',$$

we see that

$$(1.3.6) \quad \begin{aligned} \theta_n^{(I_a, \lambda)} &= a \theta_n^{(I_1, \lambda/a)}, \\ b_n^{(I_a, \lambda)}(x) &= \frac{1}{a} b_n^{(I_1, \lambda/a)}\left(\frac{x}{a}\right), \text{ for } x \in I_a. \end{aligned}$$

Note that $1/a$ in (1.3.6) normalizes the eigenfunctions so that $\|b_n^{(I_a, \lambda)}\|_{L_2(0, a)} = 1$.

Finally, consider the 2D case (that we are interested in), $D = I_{a_1} \times I_{a_2}$, and kernel (1.3.2) (with $\sigma^2 = 1$) and the eigenproblem (1.3.4). Clearly, for the eigenpairs we have

$$(1.3.7) \quad \begin{aligned} \theta_n &= \theta_{i_n}^{(I_{a_1}, \lambda_1)} \theta_{j_n}^{(I_{a_2}, \lambda_2)}, \\ b_n(\mathbf{x}) &= b_{i_n}^{(I_{a_1}, \lambda_1)}(x_1) b_{j_n}^{(I_{a_2}, \lambda_2)}(x_2), \text{ for } \mathbf{x} = (x_1, x_2) \in D, \end{aligned}$$

for some $i_n, j_n \in \mathbb{N}$. Note that in the general case when $\sigma^2 \neq 1$, the only change necessary is to multiply the eigenvalues in (1.3.7) by σ^2 .

The heuristic used in MLMC, i.e., that the uncertainty can be captured accurately enough on coarse grids in this model problem implies the requirement that the coarsest mesh must provide some level of resolution of the problem with respect to the correlation length used in the covariance function (cf., e.g., [3]), that is, the coarsest mesh size, h_0 , has to be smaller than the correlation length.

In the following experiments, for illustration, we assume $\gamma = 1$ (optimal solver) which is realistic for multigrid solvers. In contrast for sparse direct solvers we can let $\gamma = 1.5$. It is clear that when M grows (the finest mesh is refined) we can expect that the speedup of the MLMC method compared to the standard MC improves. While clearly the ε -costs of both the MLMC and the standard MC estimators as functions of ε (with fixed $M = M_L$ and L) are of order ε^{-2} , in practice M needs to be increased as $\varepsilon \downarrow 0$ to maintain the bias smaller than ε . Consequently, the speedup of the MLMC estimator in comparison to the standard MC estimator will increase as $\varepsilon \downarrow 0$.

2. NUMERICAL EXPERIMENTS

2.1. Basic experiments. Here we perform experiments similar to the ones in [3] for verification purposes. We consider equation (1.3.1) with $D = I_1 \times I_1 = (0, 1) \times (0, 1)$, $f \equiv 0$, subject to the following boundary conditions

$$\begin{aligned} p|_{x_1=0} &= 1, \\ p|_{x_1=1} &= 0, \\ \frac{\partial p}{\partial \mathbf{n}} \Big|_{x_2=0} &= 0, \\ \frac{\partial p}{\partial \mathbf{n}} \Big|_{x_2=1} &= 0. \end{aligned}$$

Additionally, we use an uniform rectangular mesh ($m_\ell \times m_\ell$ elements on level M_ℓ) and bilinear finite elements to discretize the PDE. The quantity of interest we use is the (horizontal) flux through the ‘‘outflow’’ part of the boundary, also referred to as effective (horizontal) conductivity of the region D , defined as

$$k_{\text{eff}} = - \int_0^1 k \frac{\partial p}{\partial x_1} \Big|_{x_1=1} dx_2.$$

Also, we choose $\mathbb{E}[Z(\mathbf{x}, \cdot)] \equiv 0$ in (1.3.5).

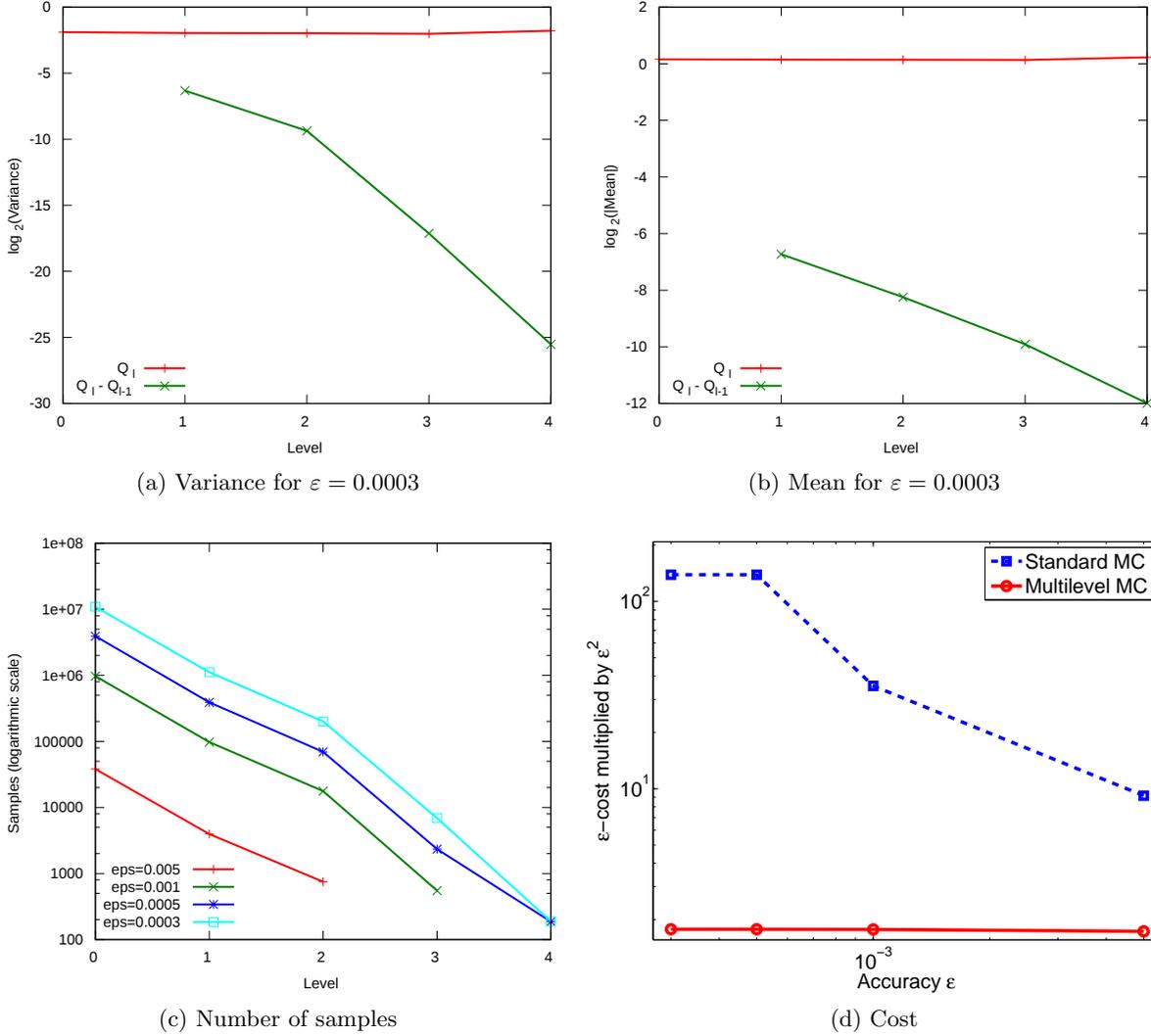


Figure 2.1.1: Variance, mean, number of samples and cost for $\gamma = 1$ (optimal solver), $\lambda_1 = \lambda_2 = 0.3$, $\sigma^2 = 1$, $m_{KL} = 1444$, $m_0 = 8$.

Fig. 2.1.1a shows that the variance of Q_ℓ on different levels is close to constant and the variance of Y_ℓ gets smaller with finer levels and accordingly, as seen in fig. 2.1.1c, the number of samples decreases as we move to finer and finer levels. Fig. 2.1.1d illustrates the ϵ -cost of the standard and multilevel MC methods. The total cost of the simulation is “normalized” in the sense that it is measured in term of one forward solve on the coarsest grid (which is the same in each case). Additionally, the y -axis in fig. 2.1.1d is scaled by ϵ^2 . Note that, as mentioned above, in fig. 2.1.1d we take into consideration the fact that smaller values of ϵ require simulations on finer grids (larger M).

Consider the case $\epsilon = 0.005$ on a 32×32 mesh. Then, the 3-level MLMC estimator gives $\mathbb{E}[Q_M] \approx 1.10079$ whereas the standard MC estimator is $\mathbb{E}[Q_M] \approx 1.09337$. The difference is 0.00742, which is reasonable for the choice of $\epsilon = 0.005$.

2.2. SPE10 experiments. Here we show experiments incorporating data from the SPE10 dataset.

The 3D domain for the SPE10 dataset has dimensions $1200 \times 2200 \times 170$ feet and it is divided into cells of size $20 \times 10 \times 2$. Thus, the 3D mesh has $60 \times 220 \times 85$ cells. We consider the 3D domain cut

into 85 horizontal (parallel to the xy plane) slices and we solve 2D problems corresponding to such 2D slices. The 2D domain has dimensions 1200×2200 and it is divided into cells of size 20×10 resulting in a mesh with 60×220 elements (13200 fine-grid elements). The PDE coefficient on each slice is a scalar function. The original 60×220 mesh is the coarsest one in our MLMC experiments. To produce the other (finer) levels, we uniformly refine the initial 2D mesh several times.

We have $D = I_{1200} \times I_{2200}$ and perform similar to the previous model case tests but here we use horizontal slices of SPE10 as the mean of the random conductivity field. That is, we take $\mathbb{E}[Z(\mathbf{x}, \cdot)] = \ln[k_{\text{SPE10 slice}}(\mathbf{x})]$ in (1.3.5), where $k_{\text{SPE10 slice}}(\mathbf{x})$ is the SPE10 coefficient for the considered horizontal slice. We solve (1.3.1) with $f \equiv 0$ and subject to the following boundary conditions

$$\begin{aligned} p|_{x_2=2200} &= 1, \\ p|_{x_2=0} &= 0, \\ \frac{\partial p}{\partial \mathbf{n}}|_{x_1=0} &= 0, \\ \frac{\partial p}{\partial \mathbf{n}}|_{x_1=1200} &= 0. \end{aligned}$$

Furthermore, we use bilinear finite elements to discretize the PDE. The quantity of interest is again the effective (horizontal) conductivity of the region D which in this case is defined as

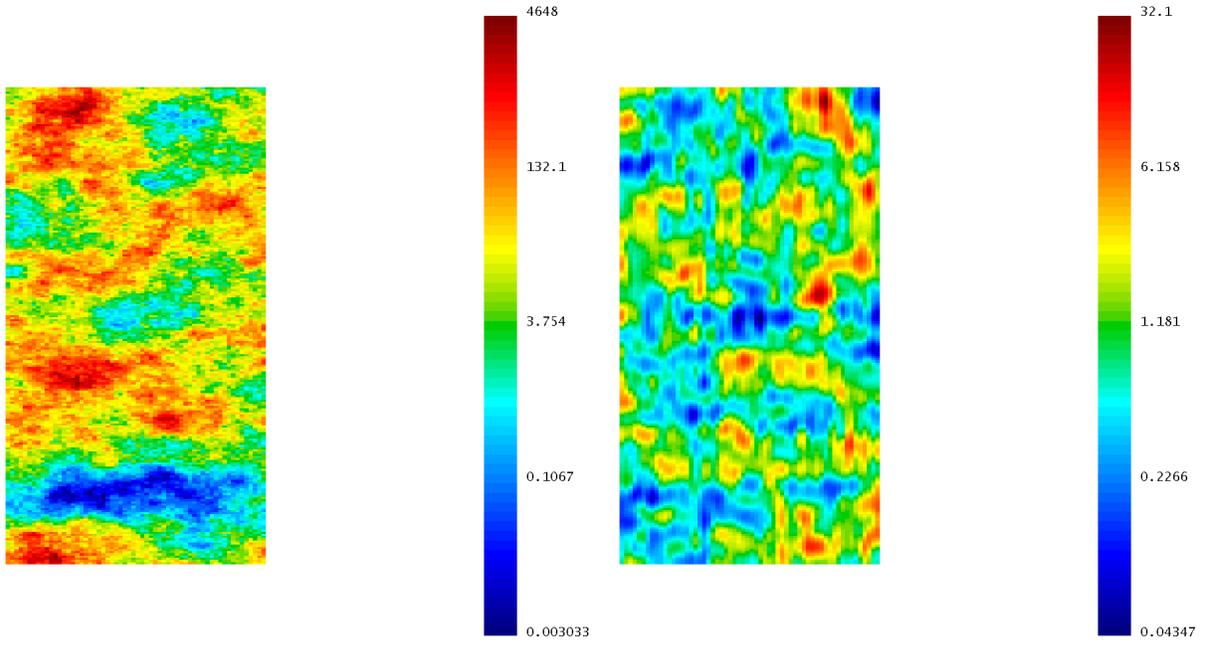
$$k_{\text{eff}} = \int_0^{1200} k \frac{\partial p}{\partial x_2} \Big|_{x_2=0} dx_1.$$

We also choose a correlation length of 100 feet, i.e. $\lambda_1 = \lambda_2 = 100$.

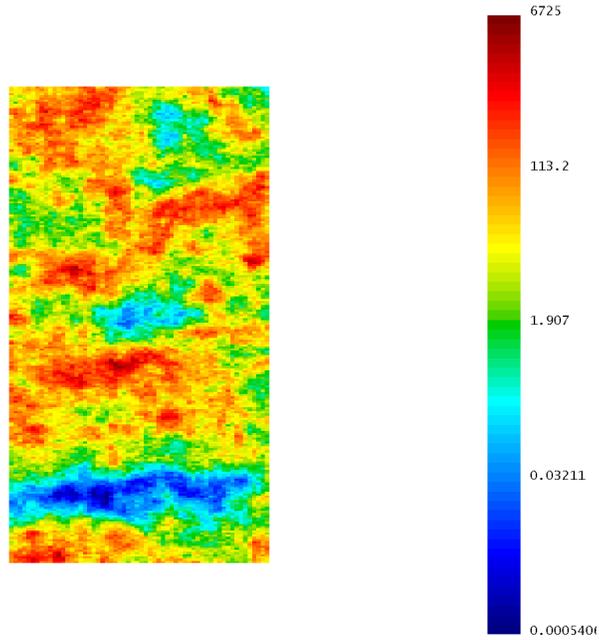
Clearly, the sampled coefficients in these experiments are simply products of SPE10 slices and coefficients sampled using zero mean in (1.3.5). The last is illustrated in fig. 2.2.1. Fig. 2.2.1c shows a typical sample of the conductivity field used in our simulations while fig. 2.2.1a and 2.2.1b show the components composing this coefficient. Namely, in fig. 2.2.1a we see the exponent of the mean of the Gaussian field in (1.3.5) and fig. 2.2.1b shows the exponent of the sum term in (1.3.5). Similarly, figures 2.2.2, 2.2.3, and 2.2.4 illustrate the same quantities when other slices are used as the mean.

Likewise, figures 2.2.5, 2.2.6, 2.2.7, and 2.2.8 show results from simulations using SPE10 coefficients of different slices as mean of the random conductivity fields. We see that the trend is similar to the previous basis model experiments. We note that the simulations may require smaller number of samples but a minimum number is enforced so that the variance is approximated accurately according to the formula. This explains behavior like in fig. 2.2.5c (graphs are “merged” together).

Tables 2.2.1, 2.2.2, 2.2.3, and 2.2.4 show comparison between standard and multilevel MC. We can observe from our experiments that MLMC demonstrates substantial speedup in comparison to the standard MC. We point out that executing standard MC for more complex experiments is highly infeasible as even in our demonstrations single level MC was impractical due to the large amount of time necessary to complete the simulation process.

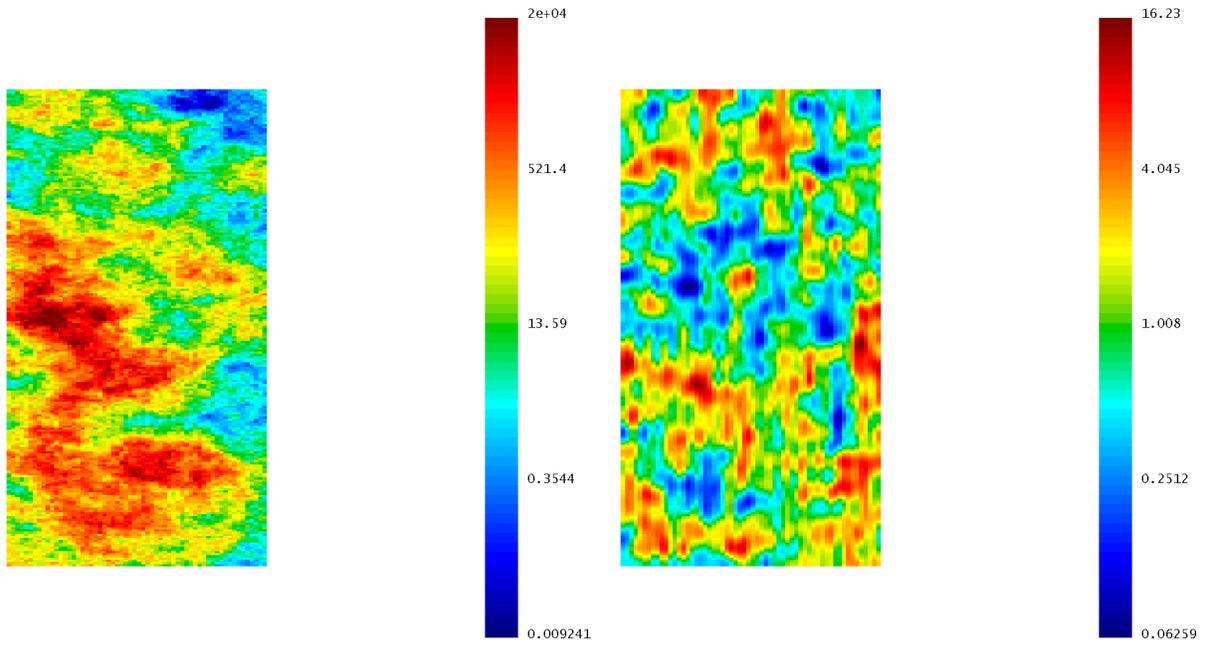


(a) The original slice 0 of SPE10 (logarithmic scale)

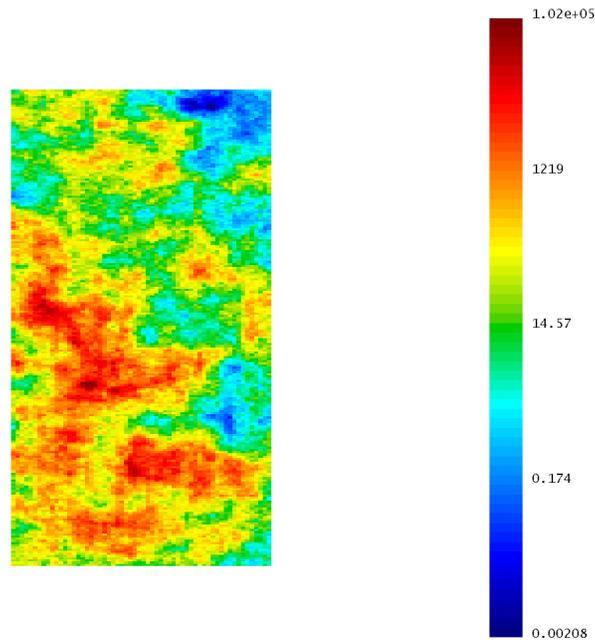
(b) Conductivity coefficient sampled using zero mean, $\sigma^2 = 1$, $\lambda_1 = \lambda_2 = 100$, and $m_{\text{KL}} = 1444$ (logarithmic scale)

(c) The final coefficient – a product of (a) and (b) (logarithmic scale)

Figure 2.2.1: A sample coefficient composed of slice 0 as mean and a random combination of the KL-modes, see (1.3.5), on the coarsest 60×220 mesh.

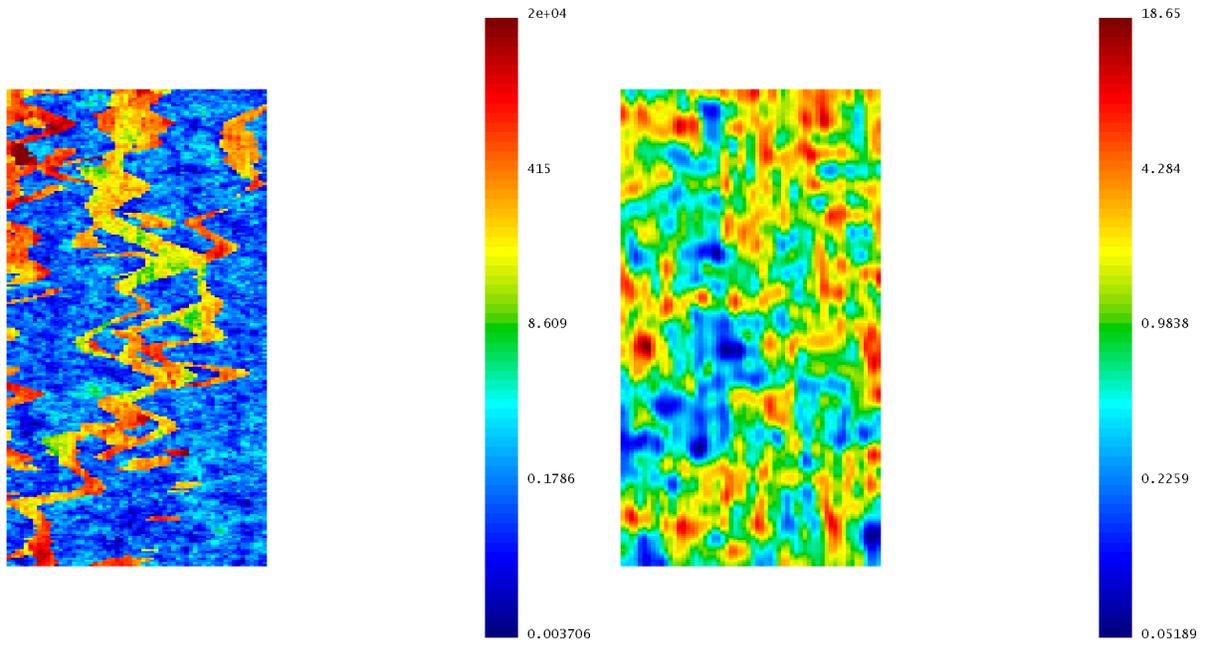


(a) The original slice 12 of SPE10 (logarithmic scale) (b) Conductivity coefficient sampled using zero mean, $\sigma^2 = 1$, $\lambda_1 = \lambda_2 = 100$, and $m_{\text{KL}} = 1444$ (logarithmic scale)

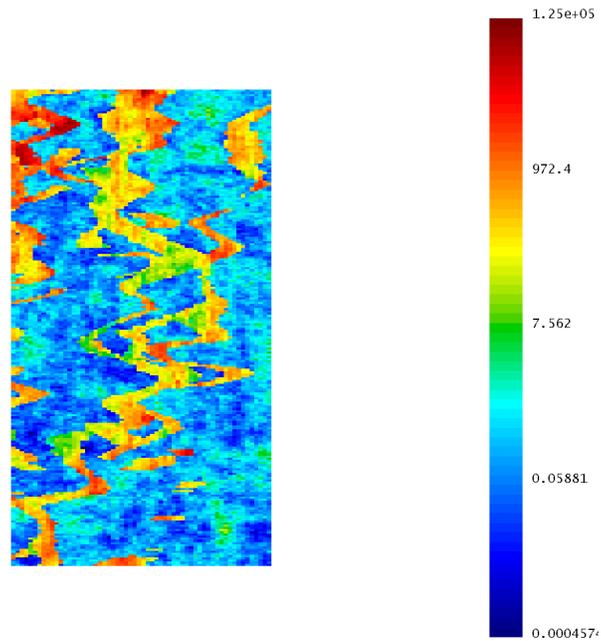


(c) The final coefficient – a product of (a) and (b) (logarithmic scale)

Figure 2.2.2: A sample coefficient composed of slice 12 as mean and a random combination of the KL-modes, see (1.3.5), on the coarsest 60×220 mesh.

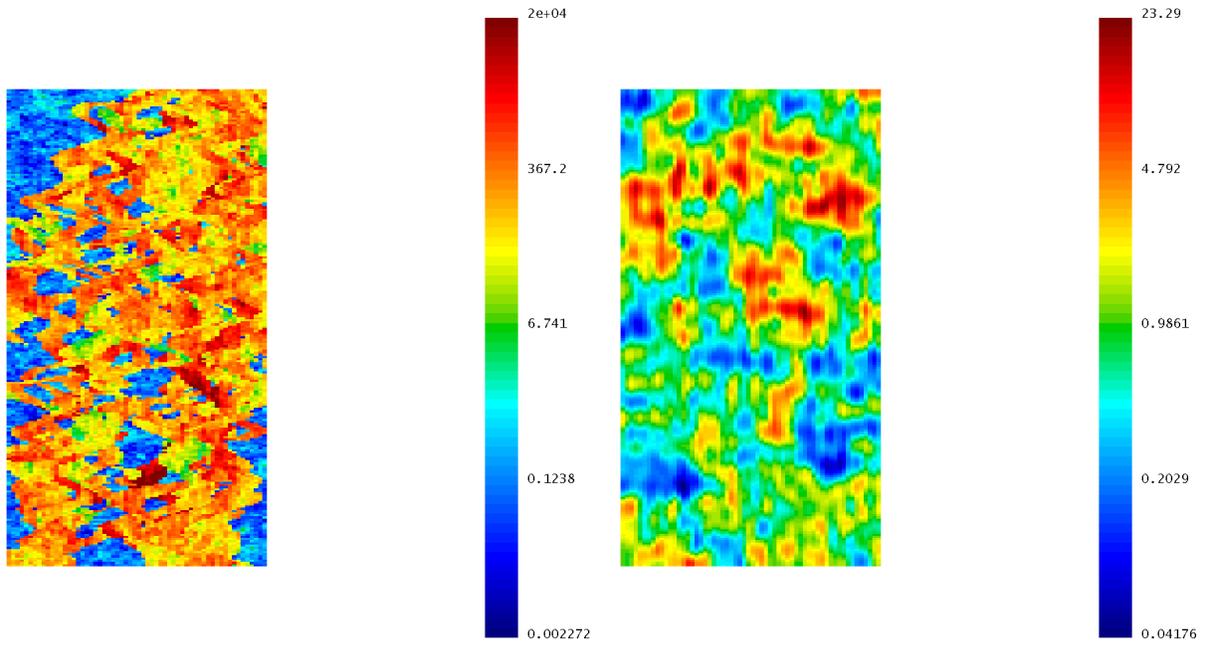


(a) The original slice 36 of SPE10 (logarithmic scale) (b) Conductivity coefficient sampled using zero mean, $\sigma^2 = 1$, $\lambda_1 = \lambda_2 = 100$, and $m_{\text{KL}} = 1444$ (logarithmic scale)

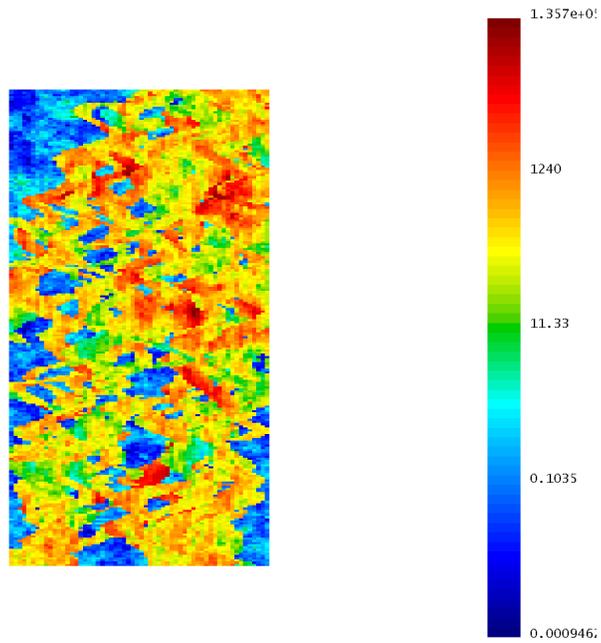


(c) The final coefficient – a product of (a) and (b) (logarithmic scale)

Figure 2.2.3: A sample coefficient composed of slice 36 as mean and a random combination of the KL-modes, see (1.3.5), on the coarsest 60×220 mesh.



(a) The original slice 84 of SPE10 (logarithmic scale) (b) Conductivity coefficient sampled using zero mean, $\sigma^2 = 1$, $\lambda_1 = \lambda_2 = 100$, and $m_{\text{KL}} = 1444$ (logarithmic scale)



(c) The final coefficient – a product of (a) and (b) (logarithmic scale)

Figure 2.2.4: A sample coefficient composed of slice 84 as mean and a random combination of the KL-modes, see (1.3.5), on the coarsest 60×220 mesh.

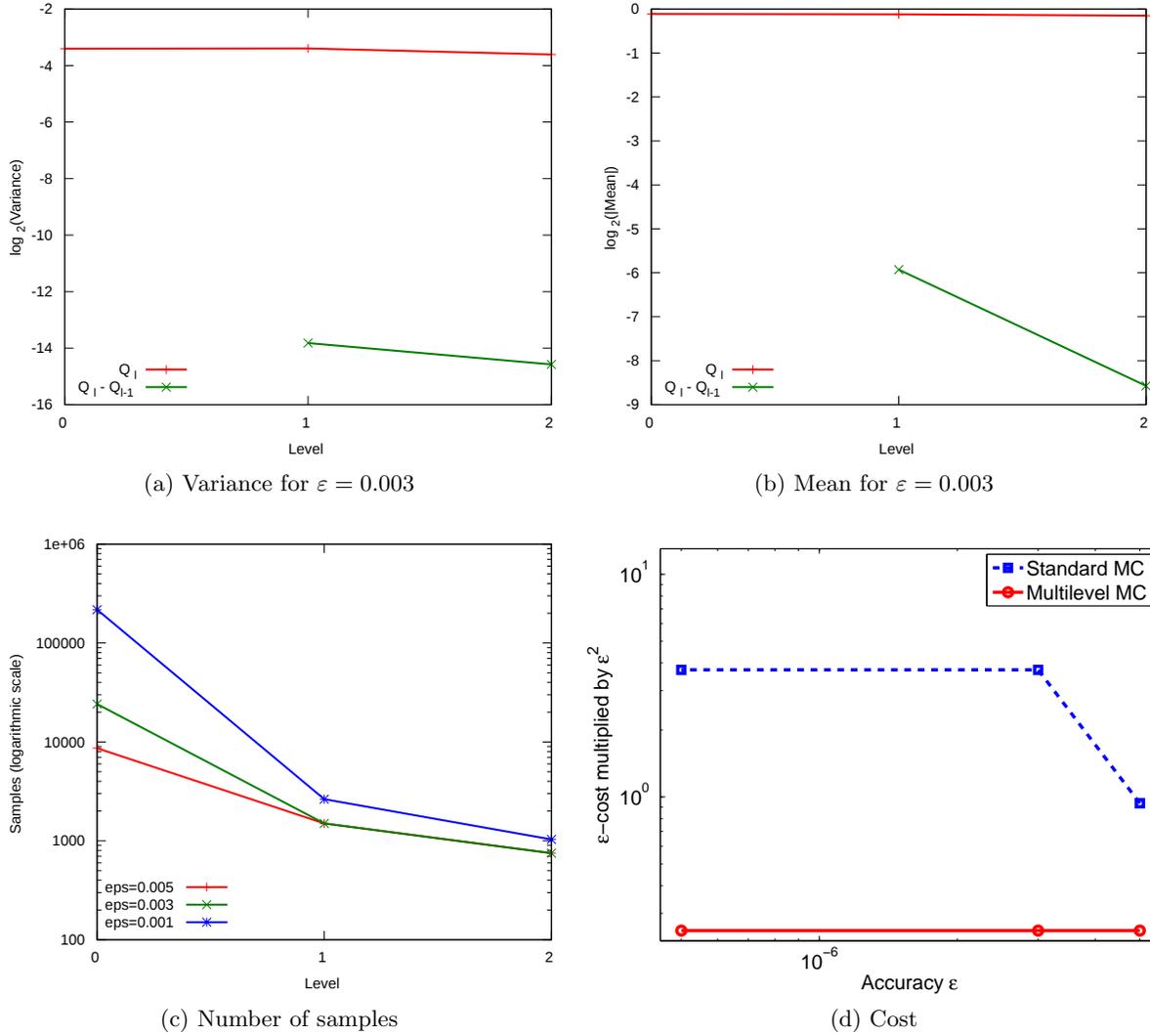


Figure 2.2.5: Variance, mean, number of samples and cost when slice 0 is the mean. Parameters: $\gamma = 1$ (optimal solver), $\lambda_1 = \lambda_2 = 100$, $\sigma^2 = 1$, $m_{\text{KL}} = 1444$.

	# dofs	# samples	Wall-clock time (hours)	$\mathbb{E}[Q_M] \approx$
# Standard MC	53361	20611	6.06	0.917358
# Multilevel MC	F: 53361 C: 13481	F: 1500 C: 22779	2.88	0.919248

Table 2.2.1: Comparison between standard MC and two-level MC when slice 0 is the mean. The fine level is the original SPE10 mesh refined once. Parameters: $\varepsilon = 0.003$, $\gamma = 1$ (optimal solver), $\lambda_1 = \lambda_2 = 100$, $\sigma^2 = 1$, $m_{\text{KL}} = 625$.

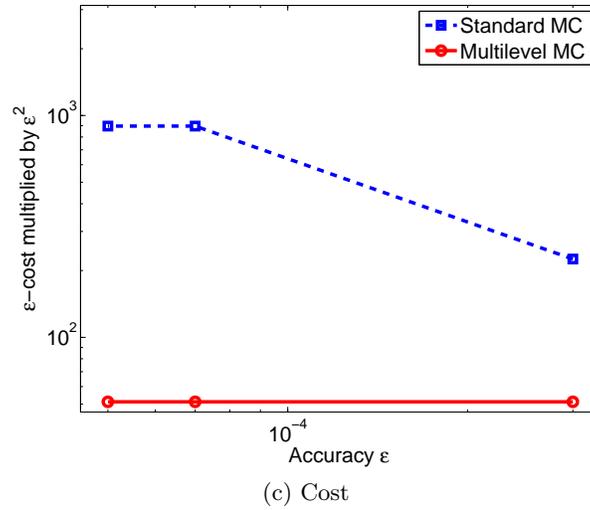
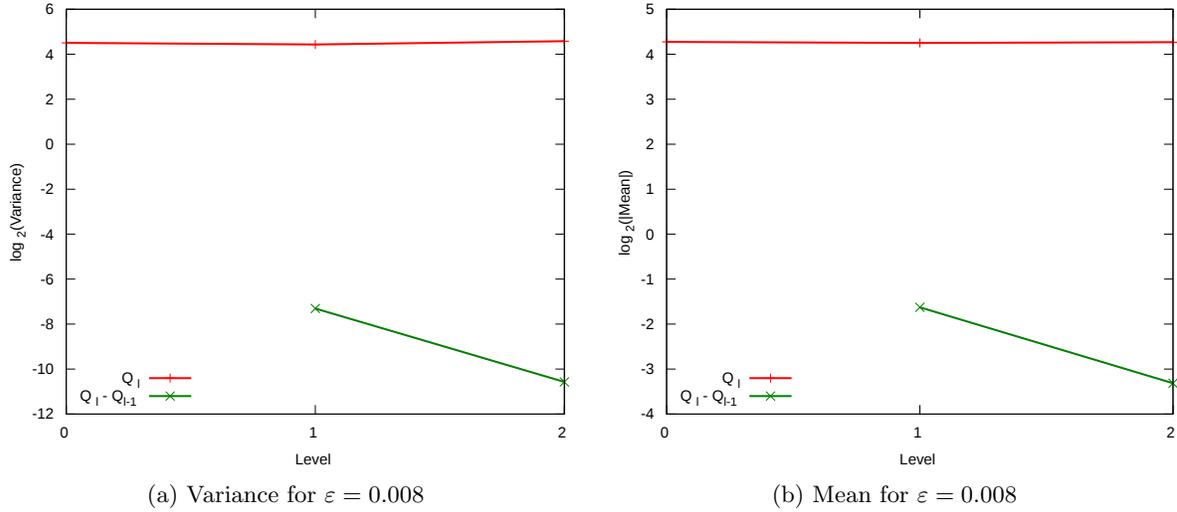


Figure 2.2.6: Variance, mean, number of samples and cost when slice 12 is the mean. Parameters: $\gamma = 1$ (optimal solver), $\lambda_1 = \lambda_2 = 100$, $\sigma^2 = 1$, $m_{KL} = 1444$.

	# dofs	# samples	Wall-clock time (hours)	$\mathbb{E}[Q_M] \approx$
# Standard MC	53361	36058	10.53	19.0903
# Multilevel MC	F: 53361 C: 13481	F: 1500 C: 38512	4.23	19.0937

Table 2.2.2: Comparison between standard MC and two-level MC when slice 12 is the mean. The fine level is the original SPE10 mesh refined once. Parameters: $\varepsilon = 0.035$, $\gamma = 1$ (optimal solver), $\lambda_1 = \lambda_2 = 100$, $\sigma^2 = 1$, $m_{KL} = 625$.

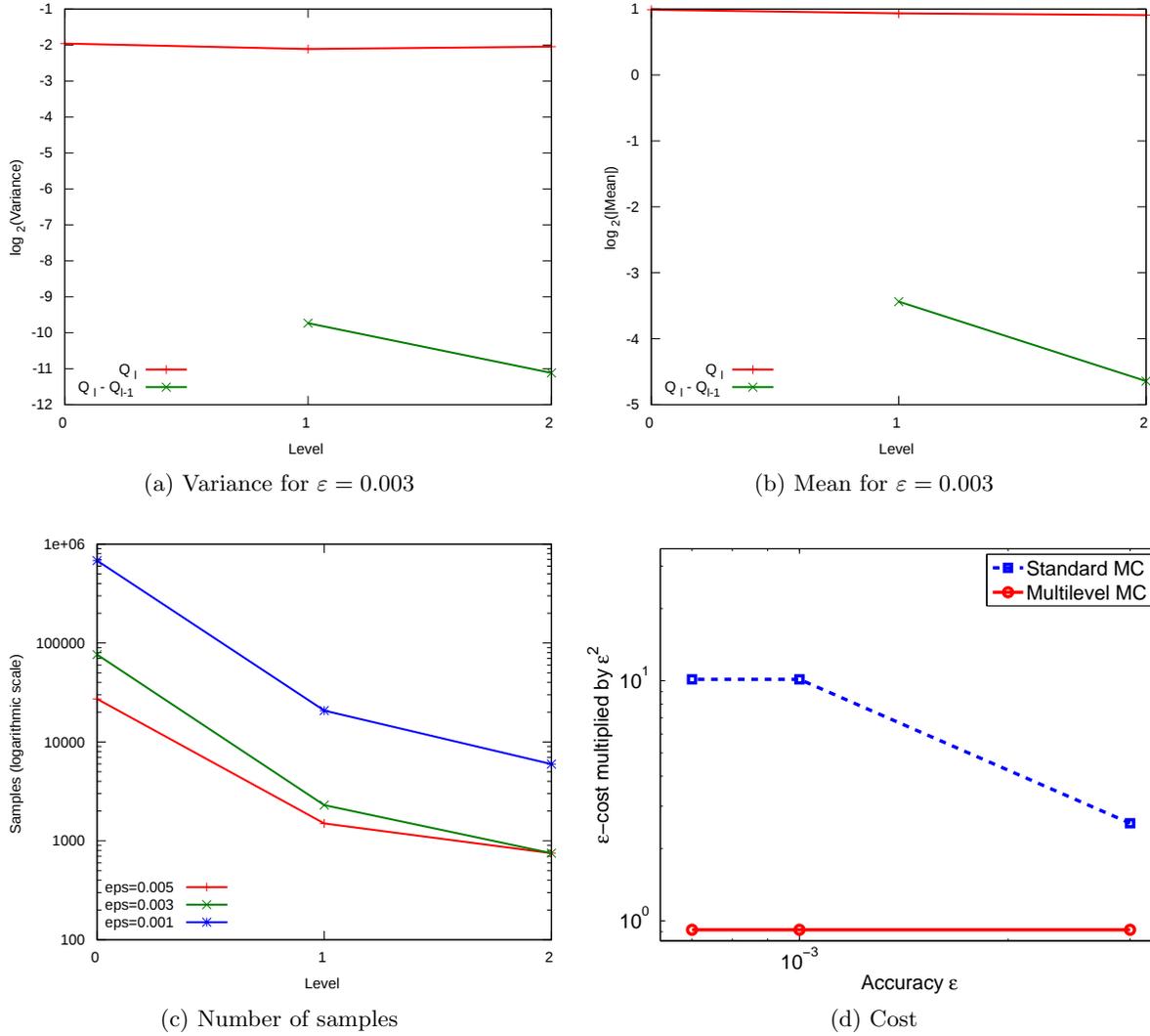


Figure 2.2.7: Variance, mean, number of samples and cost when slice 36 is the mean. Parameters: $\gamma = 1$ (optimal solver), $\lambda_1 = \lambda_2 = 100$, $\sigma^2 = 1$, $m_{\text{KL}} = 1444$.

	# dofs	# samples	Wall-clock time (hours)	$\mathbb{E}[Q_M] \approx$
# Standard MC	53361	51815	15.97	1.92043
# Multilevel MC	F: 53361 C: 13481	F: 1746 C: 65016	7.63	1.92158

Table 2.2.3: Comparison between standard MC and two-level MC when slice 36 is the mean. The fine level is the original SPE10 mesh refined once. Parameters: $\varepsilon = 0.003$, $\gamma = 1$ (optimal solver), $\lambda_1 = \lambda_2 = 100$, $\sigma^2 = 1$, $m_{\text{KL}} = 625$.

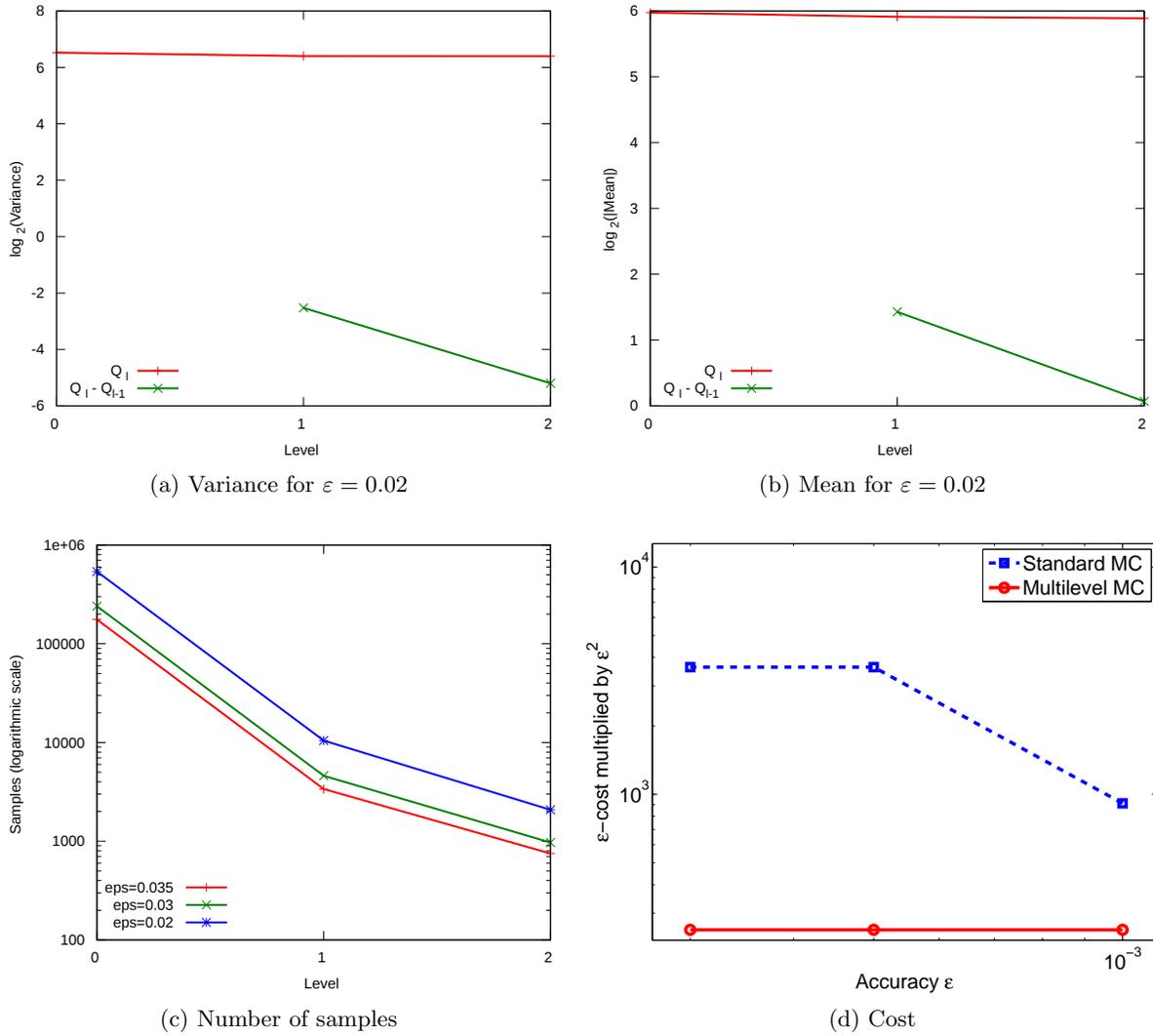
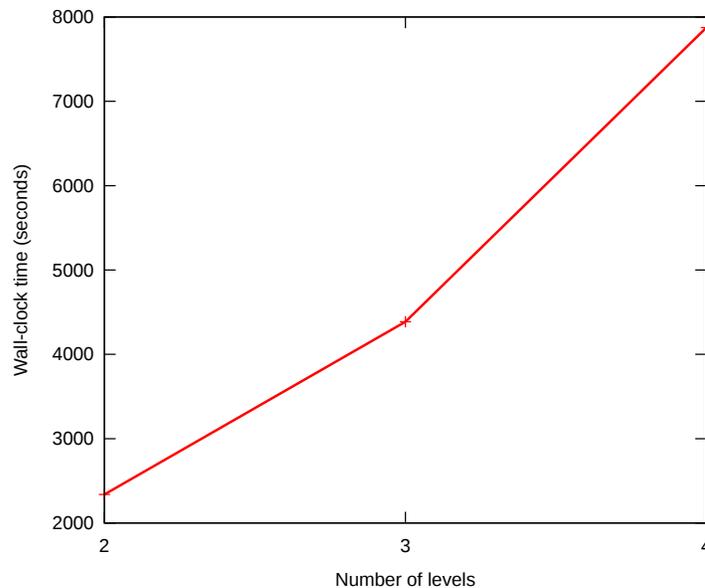


Figure 2.2.8: Variance, mean, number of samples and cost when slice 84 is the mean. Parameters: $\gamma = 1$ (optimal solver), $\lambda_1 = \lambda_2 = 100$, $\sigma^2 = 1$, $m_{KL} = 1444$.

	# dofs	# samples (N_ℓ)	Wall-clock time (hours)	$\mathbb{E}[Q_M] \approx$
Standard MC	53361	47093	14.56	60.5969
Multilevel MC	F: 53361	F: 1500	6.47	60.6008
	C: 13481	C: 56254		

Table 2.2.4: Comparison between standard MC and two-level MC when slice 84 is the mean. The fine level is the original SPE10 mesh refined once. Parameters: $\varepsilon = 0.06$, $\gamma = 1$ (optimal solver), $\lambda_1 = \lambda_2 = 100$, $\sigma^2 = 1$, $m_{KL} = 625$. (F–fine; C–coarse)



	# dofs	N_ℓ	$\mathbb{V}[Y_\ell] \approx$	$\mathbb{V}[Y_\ell]/N_\ell \approx$	$\mathbb{E}[Y_\ell] \approx$	Wall-clock time (seconds)	$\mathbb{E}[Q_M] \approx$
2-level MC	0: 13481	0: 7888	0: 0.0933024	0: 1.18284e-05	0: 0.933343	2338	0.917998
	1: 53361	1: 750	1: 6.04015e-05	1: 8.05353e-08	1: -0.0153453		
3-level MC	0: 13481	0: 8573	0: 0.092807	0: 1.08255e-05	0: 0.926812	4386	0.909411
	1: 53361	1: 750	1: 5.86373e-05	1: 7.8183e-08	1: -0.0155542		
	2: 212321	2: 375	2: 4.57804e-05	2: 1.22081e-07	2: -0.00184678		
4-level MC	0: 13481	0: 9960	0: 0.0942711	0: 9.46497e-06	0: 0.923786	7874	0.901161
	1: 53361	1: 750	1: 6.0509e-05	1: 8.06786e-08	1: -0.0154669		
	2: 212321	2: 375	2: 5.4952e-05	2: 1.46539e-07	2: -0.00207778		
	3: 847041	3: 187	3: 2.95588e-05	3: 1.58069e-07	3: -0.00508019		

Figure 2.2.9: Results for increasing number of levels using slice 0 as mean. The coarsest level, indexed with 0, is fixed. The graph illustrates how the wall-clock time (in seconds) grows with respect to the number of levels. Parameters: $\varepsilon = 0.005$, $\gamma = 1$ (optimal solver), $\lambda_1 = \lambda_2 = 100$, $\sigma^2 = 1$, $m_{\text{KL}} = 625$.

3. CONCLUSIONS

The presented MLMC method clearly demonstrates the potential for substantially speeding-up of practically infeasible by the single level MC expensive simulation of quantities of interest that require for each sample solution of (discretized) elliptic PDE with random coefficients. The multilevel approach, considered as a variance reduction technique for Monte Carlo simulations, shows clear advantages over the stand-alone (single level) MC method in terms of number of floating point operations and, consequently, in terms of time required to ensure a prescribed bound of the error.

REFERENCES

- [1] A. Brandt, “Multiscale Scientific Computation: Review 2000.” (available at <http://www.wisdom.weizmann.ac.il/~achi/>).
- [2] M. B. Giles. Multilevel Monte Carlo path simulation. *Operations Research*, 56(3):607–617, 2008.
- [3] K. A. Cliffe, M. B. Giles, R. Scheichl, and A. L. Teckentrup. Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients. *Computing and Visualization in Science*, 14(1):3–15, 2011.

- [4] R. G. Ghanem and P. D. Spanos. *Stochastic Finite Elements: A Spectral Approach*. Civil, Mechanical and Other Engineering Series. Dover Publications, Incorporated, 2003.

CENTER FOR APPLIED SCIENTIFIC COMPUTING, LAWRENCE LIVERMORE NATIONAL LABORATORY, P.O. BOX 808, L-560, LIVERMORE, CA 94550, U.S.A.

E-mail address: delyank@gmail.com, panayot@llnl.gov