Preliminary numerical results on Multilevel Monte Carlo for elliptic PDEs with random coefficients

Aretha Teckentrup and Robert Scheichl

May 18, 2010

1 Results on the 1D model problem

1.1 The model problem

We consider the boundary value problem

$$\begin{aligned} -\frac{\mathrm{d}}{\mathrm{d}x}\Big(k(x,\omega)\frac{\mathrm{d}u}{\mathrm{d}x}(x,\omega)\Big) &= 0, \\ u|_{x=0} = 1, \quad u|_{x=1} &= 0. \end{aligned} \tag{1}$$

for $x \in D = [0,1]$, where the coefficient k depends on the spatial variable x, as well as the outcome $\omega \in \Omega$ of a random event. This makes our solution u implicitly dependent on ω as well, hence we write $u(x,\omega)$.

For the moment, we take k to be a spatially correlated lognormal random field. More precisely, let $\log(k(x,\omega))$ be a normal random field with mean 0 and covariance function

$$C(x_1,x_2) = \sigma^2 \exp(-\frac{|x_1-x_2|}{\lambda}),$$

where σ^2 denotes the variance and λ the correlation length of $\log k$. For each given outcome ω, k is simply a (deterministic) function of the spatial variable x, and so fixing ω transforms the above boundary value problem into a deterministic one. We know how to solve deterministic problems of this type numerically, and so we will make use of this fact to get an approximate solution to our original (stochastic) boundary value problem.

We start by employing a standard Monte Carlo method to approximate statistics of the solution $u(x,\omega)$. In other words, we repeat the following two steps a large number of times.

- First generate a sample from the random field $k(x,\omega)$, denoted by $k^{(i)}(x)$.
- Then use the finite element method to solve the deterministic problem

$$\begin{aligned} -\frac{\mathrm{d}}{\mathrm{d}x}\left(k^{(i)}(x)\frac{\mathrm{d}u^{(i)}}{\mathrm{d}x}(x)\right) &= 0, \\ u^{(i)}(1) = 1, \quad u^{(i)}(0) &= 0. \end{aligned}$$

  to get an approximate solution $u_h^{(i)}(x)$, dependent on the mesh size h of our finite element approximation.

We then use these samples to calculate estimates of the quantities we are interested in.

1.2 The finite element implementation

We choose to work with piecewise linear finite elements, and set up a uniform mesh of size h on our spatial domain $D = [0, 1]$. m will denote the total number of nodes in this mesh, including the boundary nodes at 0 and 1. For the calculation of the stiffness matrix and load vector, we will use the value of $k^{(i)}$ at the midpoint of each element.

Note that theoretically, the average of k over each element j,

$$k_{h,j}^{(i)} := \frac{1}{h} \int_{(j-1)h}^{jh} k^{(i)}(x)dx,$$

is required, but as usual we employ a quadrature rule of high enough order to approximate this integral without any detrimental effect on the rate of convergence of the FE method. Choosing the midpoint rule is sufficient.

1.3 Sampling from the random field coefficient

To produce the samples of k we are going to use the Karhunen-Loeve (KL) expansion. This is an expansion of a random field $h(x, \omega)$ in terms of a countable set of uncorrelated, zero mean random variables $\{\xi_n\}_{n\in\mathbb{N}}$, given explicitly by

$$h(x, \omega) = \mathbb{E}\left[h(x)\right] + \sum_{n=0}^{\infty} \sqrt{\theta_n}\xi_n(\omega)f_n(x), \tag{2}$$

where $\{\theta_n\}_{n\in\mathbb{N}}$ is a sequence of real numbers and $\{f_n\}_{n\in\mathbb{N}}$ is a sequence of deterministic functions related to the autocovariance function of $h(x, \omega)$. More precisely, given the autocorrelation function $r(x_1, x_2)$ of $h(x, \omega)$, $\{\theta_n\}_{n\in\mathbb{N}}$ and $\{f_n\}_{n\in\mathbb{N}}$ are given as the solutions of the integral operator eigenproblem

$$\int_D r(x_1, x_2)f(x_2)dx_2 = \theta f(x_1). \tag{3}$$

In the case of the exponential correlation $r(x_1, x_2) = \exp(-\frac{|x_1-x_2|}{\lambda})$, the above integral eigenproblem can be solved analytically to get

$$\begin{aligned} \theta_n &= \frac{2\lambda}{\lambda^2\omega_n^2 + 1}, \qquad n \in \mathbb{N} \\ f_n(x) &= A\big(\sin(\omega_n x) + \lambda\omega_n \cos(\omega_n x)\big), \qquad n \in \mathbb{N} \end{aligned} \tag{4}$$

where $\{\omega_n\}_{n\in\mathbb{N}}$ are the (real) solutions of the transcendental equation

$$\tan(\omega) = \frac{2\lambda\omega}{\lambda^2\omega^2 + 1},$$

and the constant A is the normalisation constant arising from the normalisation

$$\int_D f_n^2 dx = 1$$

(following e.g. [Ghanem, Spanos, 1991]). Note that to get the eigenvalues of the autocovariance function $C(x_1, x_2)$ defined above, we simply need to multiply the eigenvalues $\{\theta_n\}_{n\in\mathbb{N}}$ by σ^2.

We hence have an explicit formulation of the KL expansion for the Gaussian field $\log(k)$. We will make use of this to produce random samples of the field $k(x, \omega)$, by first taking samples of the field $\log(k(x, \omega))$ and then transforming them. Note that since uncorrelated normal random

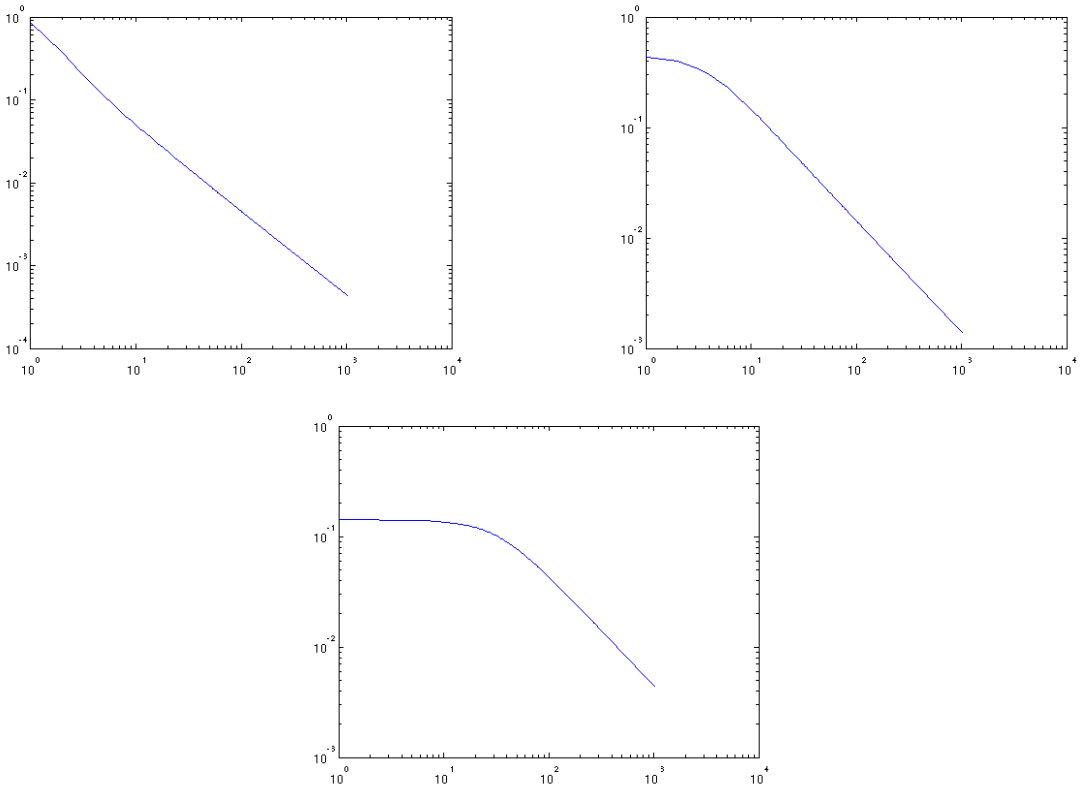

Figure 1: Plot of the square root of the largest eigenvalues of (3), for different correlation lengths: $\lambda = 1$ (top left), $\lambda = 0.1$ (top right), $\lambda = 0.01$ (bottom)

variables are independent, the $\{\xi_n\}_{n\in\mathbb{N}}$ in the KL expansion for $\log(k(x,\omega))$ are i.i.d. standard normal variables, which are straightforward to sample from.

Another issue is the fact that the KL expansion is of course infinite. In practice we need to truncate the KL expansion after a finite number of terms, say after m_{KL} terms. In order to decide how many terms we should include in the expansion, we make use of the following observations:

- The integral operator in (3) is self-adjoint, non-negative and compact, which implies that it has a countable sequence of real, non-negative eigenvalues tending to 0, and that its set of eigenfunctions is orthogonal and complete. Furthermore, we have that the sum of all eigenvalues is finite, with
$$\sum_{n=1}^{\infty} \theta_n = |D| Var(\log(k)),$$
where $|D|$ is the length of the spatial domain D. So for our particular model problem we hence have that the eigenvalues sum up to σ^2 (see e.g. [Powell, Elman, 2007]).

- Taking these facts into consideration, the number of terms to include in the KL expansion is usually based on the speed of the decay of the eigenvalues $\{\theta_n\}_{n\in\mathbb{N}}$. Since σ^2 scales all the eigenvalues by the same amount, the decay of the eigenvalues is determined by λ. Some plots are shown in Figure 1. Note the very clearly visible plateau in the pre-asymptotic region whose width is inversely proportional to λ.

  Hence, the larger the value of the correlation length λ, the faster the eigenvalues will

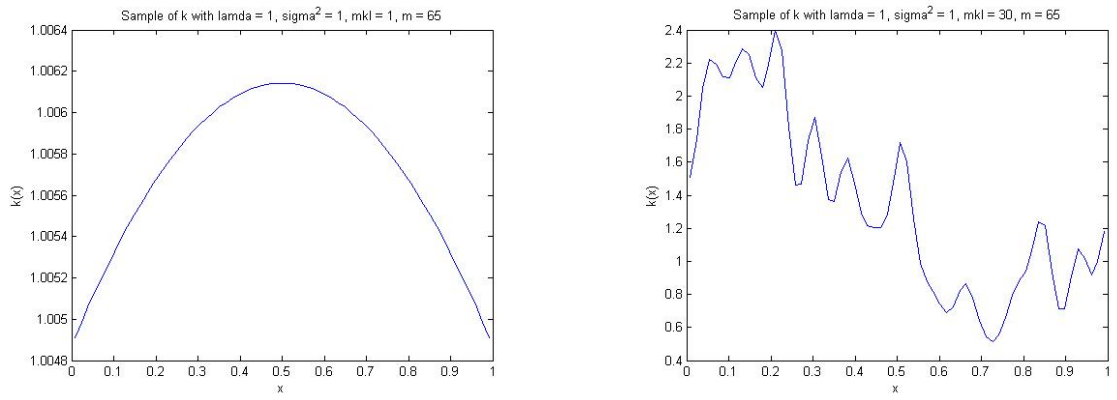

Figure 2: Approximations of samples of k with $\sigma^2 = 1$, $\lambda = 1$ and $m = 65$ truncating the KL–expansion after $m_{\text{KL}} = 1$ (left) and $m_{\text{KL}} = 30$ (right) terms.

decay. For large correlation lengths $O(1)$, the first few eigenvalues will be sufficient, while for smaller correlation lengths, many of the eigenvalues will be of approximately the same size, and can hence not be neglected.

– Considering the shapes of the eigenfunctions, this can be interpreted logically. The eigenfunctions $\{f_n\}_{n\in\mathbb{N}}$ have an increasing number of oscillations over the interval $[0, 1]$, with the first eigenfunction being close to constant on the whole domain. Hence, when the correlation length is very large, the value of k at one point in the domain is strongly correlated with the value of k at any other point in the domain, and k will have approximately the same value on the whole domain. Hence, including only the first one or two eigenvalues, and hence eigenfunctions, should be enough. If the correlation length is very small on the other hand, we expect to see greater fluctuations in the values k takes over the domain, and so we need to include the more oscillatory eigenfunctions, and hence later eigenvalues, to get a good representation of $k(x, \omega)$. Two examples of samples of k are shown in Figure 2, where only m_{KL} has been varied.

1.4 Investigating the Monte Carlo RMSE

Two statistics of the solution $u(x, \omega)$ that we might be interested in are

1. the expected value of the pressure at the centre, $\bar{u}^* := \mathbb{E}\left[u(\frac{1}{2})\right]$, and
2. the expected value of the outflow, $\bar{q} := \mathbb{E}\left[\left(k\frac{\mathrm{d}u}{\mathrm{d}x}\right)|_{x=1}\right]$.

In practice, we will only have the finite element solution u_h and we will use (for the moment) the Monte Carlo (MC) method to evaluate expected values, i.e. we compute the two quantities of interest as

$$\mathbb{E}\left[u_h(1/2)\right] \approx \bar{u}_h^* := \frac{1}{N}\sum_{i=1}^{N} u_h^{(i)}(1/2),$$

$$\mathbb{E}\left[q_h\right] \approx \bar{q}_h := \frac{1}{N}\sum_{i=1}^{N} k^{(i)}\frac{\mathrm{d}}{\mathrm{d}x}\left(u_h^{(i)}\right)\Big|_{x=1}.$$

As a measure of the error resulting from using the MC method to calculate these expectations, we shall use its root–mean–square–error, RMSE for short. To compute this, we first compute an

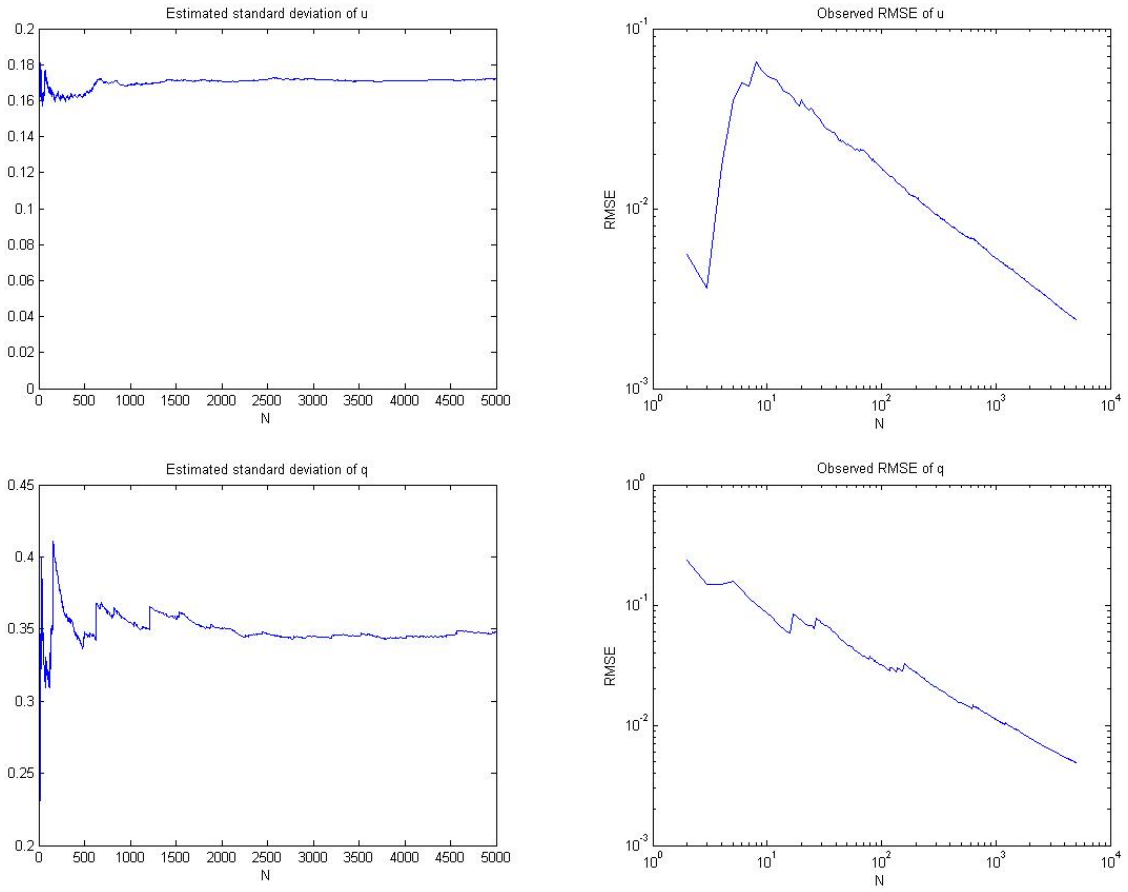

Figure 3: Plots of the Monte Carlo error.

unbiased estimate of the sample variance and then scale this appropriately to get the variance of the sample mean. So for example for the midpoint pressure, the variance of one sample can be estimated by

$$\mathrm{Var}(u_h^{(i)}) \approx \frac{1}{N-1}\left(\sum_{i=1}^{N}(u_h^{(i)})^2 - N(\bar{u}_h)^2\right),$$

and then the RMS error of the MC method is taken to be the standard deviation of the sample mean,

$$e_h^N := \sqrt{\mathrm{Var}(\bar{u}_h)} = \sqrt{\frac{1}{N}\mathrm{Var}(u_h^{(i)})} = \sqrt{\mathrm{Var}(u_h^{(i)})}\, N^{-1/2}$$

Since the estimator we are using for the sample variance is unbiased and consistent, we have that asymptotically, the MC error e_h^N decays with $O(N^{-1/2})$. In Figure 3 we can see a plot of the estimates of the standard deviation of the sample mean and the MC error as we increase the number of samples N, in the case $\lambda = 0.1, \sigma^2 = 1, m_{\mathrm{KL}} = 115$ and $h = \frac{1}{64}$. We can see that there is a 'settling down phase', or *burn in*, which precedes the predicted asymptotic stage.

1.5 Estimating the FE–error and the 'true' solution

In addition to the error caused by the MC simulation, there is also some uncertainty in the observed values of our statistics arising from the finite element (FE) discretisation, i.e. the error resulting from approximating u by u_h. We expect this error to be decreasing as h decreases,

but due to the MC error in the observed statistics the FE error is not as easy to find as in the case of a deterministic PDE. Since it can be shown (see [Graham, Kuo et al, 2010]) that $\mathbb{E}[u_h^*] = \mathbb{E}[u^*] = 0.5$ for all h and for all possible values of parameters λ and σ^2 (i.e. there is no FE error), we will only focus on the outflow q_h as our quantity of interest.

So let q_h denote the value of the outflow on a mesh of size h, which we approximate by $\bar{q}_h$ as given above. We assume that $q_h = q^\star + Ch^\alpha + \epsilon_h$, where the ϵ_h are zero–mean, independent normal random variables with weighted variances $w_h^{-1}\sigma^2$, and $\alpha > 0$ is the associated rate of convergence of the discretisation error. We are in other words assuming that q_h is equal to the 'true' value $q^\star$, plus the FE error Ch^α, plus the Monte Carlo error ϵ_h . In Table 1 we list some computed approximations of $\bar{q}_h$ and of the RMSE for the case $\lambda = 0.1, \sigma^2 = 3$ and $m_{\mathrm{KL}} = 105$.

h	$\bar{q}_h$	Monte Carlo RMSE
1/2	1.427350	$9.1 * 10^{-4}$
1/4	0.768015	$3.2 * 10^{-4}$
1/8	0.572673	$3.6 * 10^{-4}$
1/16	0.506683	$2.9 * 10^{-4}$
1/32	0.486963	$2.7 * 10^{-4}$
1/64	0.486963	$4.6 * 10^{-4}$
1/128	0.480818	$3.3 * 10^{-4}$

Table 1: Computed approximations of the mean outflow $\bar{q}_h$ and of the RMSE for the case $\lambda = 0.1, \sigma^2 = 3$ and $m_{\mathrm{KL}} = 105$ (with a sufficiently large number of samples N so that the MC error is negligible).

From this table it can easily be seen why we cannot take $\{\epsilon_h\}$ to be a set of i.i.d normal random variables, as the observed MC error is not the same for each h. The weights w_h are determined as ratios of the obeserved MC errors.

Fitting our assumed model to the data directly requires that we know the value of α, which we do not, and so an estimate for this has to be obtained first. But this can quite easily be done using the observed data. According to our assumption, we have that $q_h = q^\star + Ch^\alpha + \epsilon_h$, and from this it follows that

$$\begin{aligned} q_h - q_{2h} &= (q^\star + Ch^\alpha + \epsilon_h) - (q^\star + C(2h)^\alpha + \epsilon_{2h}) \\ &= (Ch^\alpha - C(2h)^\alpha) + (\epsilon_h - \epsilon_{2h}) \\ &= C'h^\alpha + \epsilon'_h, \end{aligned}$$

where C' is a constant independent of h, and the ϵ'_h are again independent, zero–mean normal random variables with appropriately scaled and weighted variances. This means that the difference of two successive values q_h and q_{2h} decays at the same rate α as the actual error. Hence, fitting a generalised linear model with normal response and log link function to these differences, we obtain an estimate for the parameter α. We then use this estimate to fit the linear model specified earlier to q_h.

For the data shown in the table, the estimates are $\hat{\alpha} = 1.711932$, with a standard deviation of 0.02542, and $\hat{q}^\star = 0.480316$, with a standard deviation of 0.001011. The resulting expected value and standard deviation of the FE error are shown in Table 2. Due to the normality assumptions, the standard deviation of $q^\star - q_h$ is calculated as the square root of the sum of the variances, $\sqrt{\mathrm{Var}(q_h) + \mathrm{Var}(q^\star)}$.

We note that for the two finest meshes used, the RMSE is larger than the expected value of the FE error, and so we cannot be total sure about these values. Hence also the slight deviation from the $O(h^\alpha)$ decay.

h	$q^\star - q_h$	RMSE
1/2	0.947034	$1.36 * 10^{-3}$
1/4	0.287699	$1.06 * 10^{-3}$
1/8	0.092357	$1.07 * 10^{-3}$
1/16	0.0263673	$1.059 * 10^{-3}$
1/32	0.006647	$1.05 * 10^{-3}$
1/64	0.000793	$1.11 * 10^{-3}$
1/128	0.000502	$1.06 * 10^{-3}$

Table 2: Estimates of the FE error obtained by fitting a general linear model for $\lambda = 0.1, \sigma^2 = 3$ and $m_{\mathrm{KL}} = 105$.

For other values of the parameters λ, σ^2 and m_{KL}, we observe a very similar pattern to the one above. The rate of convergence α will play a role in the error and complexity analysis later, so the estimates $\hat{q}^\star$, $\hat{\alpha}$ and $\hat{C}$ are given for a range of parameter values in the Table 3.

λ	σ^2	m_{KL}	$\hat{\alpha}$	$\hat{C}$	$\hat{q}^\star$
0.1	1	105	1.51189	0.926874	0.745827
0.1	3	105	1.71932	3.12381	0.480316
0.3	1	75	1.81185	0.706488	0.952648
1	1	35	1.93391	1.2763888	0.368767

Table 3: Fitted general linear model parameters for a rage of values of λ, σ^2 and m_{KL}.

1.6 The Multilevel Monte Carlo method

Up until now we have been using the standard Monte Carlo approach to estimate the expected value of a quantity of interest, $\mathbb{E}[Q_h]$. We produced N independent samples of Q_h, and then used the sample mean as our estimate for the expected value. The sample mean is an unbiased and consistent estimator of the mean, but as we have seen, the variance of this estimator decreases rather slowly, with $O(N^{-1/2})$. To get a high accuracy we need to take a very large number of samples, and especially on fine grids this can quickly become computationally too expensive.

We therefore propose a new method, which hopefully will reduce the computational cost required to achieve a given accuracy by a considerable amount. The method is based on the observation that, by the linearity of expectation, we can write

$$\mathbb{E}[Q_h] = \mathbb{E}[Q_h - Q_{2h}] + \mathbb{E}[Q_{2h} - Q_{4h}] + \cdots + \mathbb{E}[Q_{2^{L-1}h} - Q_{2^L h}] + \mathbb{E}[Q_{2^L h}]. \tag{5}$$

Instead of estimating the left hand side of (5) directly, we are instead going to estimate each of the terms on the right hand side seperately and then add up. (In fact, there is nothing special about the factor 2 between two consecutive mesh widths. This could also be chosen larger, e.g. 3 or 4.) We estimate the quantities on the right hand side of (5) again using the MC method, but we hope to see a reduction in the computational cost for two reasons:

- We expect the variance of the terms of the form $\mathbb{E}[Q_h - Q_{2h}]$ to be comparatively small and tending to 0 as h tends to 0. Hence, a small number of samples are needed for the MC error of the sample mean of this quantity to be below a desired accuracy.

- We still need to estimate $\mathbb{E}[Q_{2^L h}]$, which will require a large number of samples. However, the grid on which the sampling needs to be done is coarse, with a mesh size of $H := 2^L h$,

and the cost of computing N samples on this coarse grid is much lower than computing N samples on the original grid with mesh size h.

It is worthwile to note that the FE error remains unchanged in passing to this multilevel approach; our estimator for the right hand side (RHS) of equation (5) has the same FE error than that of left hand side (LHS) .

1.6.1 Results for $\mathbb{E}[Q_H]$, i.e. on the coarsest grid

First we consider the last term used in the multilevel method, the expected value of the quantity of interest on the coarsest grid. The estimated standard deviations of the outflow q_H and of the midpoint pressure u_H^* for the case $\lambda = 0.1, \sigma^2 = 1$ and $m_{\mathrm{KL}} = 105$ are given in Table 4 .

h	s.d. of q_H	s.d. of u_H^*
1/2	0.90896	0.258496
1/4	0.526824	0.217492
1/8	0.403254	0.1893274
1/16	0.364538	0.176903
1/32	0.358122	0.1741124
1/64	0.353236	0.172533
1/128	0.354342	0.1722386

Table 4: Estimated standard deviations for the coarsest grid, based on $N = 40000$ samples.

We see that the standard deviation increases only very slightly as the mesh gets coarser, while $h < \lambda$. The growth is slightly more pronounced thereafter, but it does not grow by more than a factor 1.5 and 2.5 for q_H and for u_H^*, respectively. This means that the number of samples needed to approximate the last term on the RHS of (5) to the same accuracy as the term on the LHS is the same provided we do not coarsen beyond $h < \lambda$. If we do go beyond, it will grow by a factor 1.5^2 and 2.5^2. We will have to take this into consideration in our design of the multilevel method. For other values of the parameters λ and σ^2, we see a similar pattern as in Table 4.

1.6.2 Results for $\mathbb{E}[Q_h - Q_{2h}]$

Secondly, we study the behaviour of the terms of the form $(Q_h - Q_{2h})$ in (5). Most importantly, we want to know how these terms behave as a function of the mesh size h. So we fix the other parameters, and look at how the expected value and variance vary. As before, let u_h^* and q_h denote the expected midpoint pressure and outflow respectively, and introduce $\Delta u_h^* = u_h^* - u_{2h}^*$ and $\Delta q_h = q_h - q_{2h}$. The results for $\lambda = 0.1, \sigma^2 = 1$ and $m_{\mathrm{KL}} = 105$ are presented in Table 5, based on $N = 5000$ samples. Note that the standard deviation given is not the standard deviation of the sample mean, but rather of the quantity itself.

It is clear from the table that the expected value of these differences is always quite small, and even negligible for small values of h. The standard deviation, on the other hand, shows a very interesting trend. For both the midpoint pressure and the outflow, the standard deviation of the difference seems to clearly decrease with h. Provided h is large w.r.t. the number of KL-modes that are included in the truncated expansion, i.e. roughly $h > m_{\mathrm{KL}}^{-1}$, the decay is linear. When $h < 3\, m_{\mathrm{KL}}^{-1}$ the decay is quadratic and inbetween it jumps down by several orders of magnitude. This trend is observed for all different choices of parameters, and is illustrated even better in Figures 4 and **??**. We also note that the standard deviation before and after the

h	mean of Δu_h^*	s.d. of Δu_h^*	mean of Δq_h	s.d. of Δq_h
1/4	$O(10^{-2})$	0.27	$O(10^{-1})$	0.90
1/8	$O(10^{-3})$	0.17	$O(10^{-2})$	0.42
1/16	$O(10^{-3})$	0.10	$O(10^{-2})$	0.21
1/32	$O(10^{-3})$	0.05	$O(10^{-2})$	0.10
1/64	$O(10^{-4})$	0.02	$O(10^{-3})$	0.043
1/128	$O(10^{-5})$	0.003	$O(10^{-4})$	0.0051
1/256	$O(10^{-6})$	$7.0 * 10^{-5}$	$O(10^{-5})$	$4.7 * 10^{-5}$
1/512	$O(10^{-7})$	$2.7 * 10^{-5}$	$O(10^{-6})$	$1.4 * 10^{-5}$
1/1024	$O(10^{-7})$	$7.4 * 10^{-6}$	$O(10^{-6})$	$3.7 * 10^{-6}$
1/2048	$O(10^{-7})$	$1.8 * 10^{-6}$	$O(10^{-7})$	$7.0 * 10^{-7}$

Table 5: Estimates of expected value and standard deviation of Δu_h^* and Δq_h for $\lambda = 0.1, \sigma^2 = 1$ and $m_{\mathrm{KL}} = 105$.

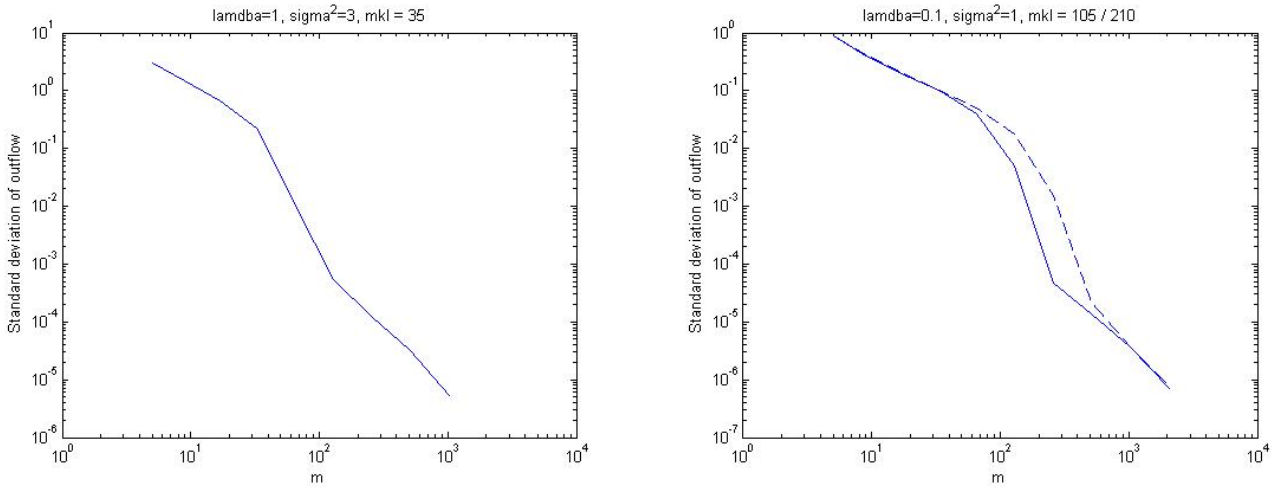

Figure 4: Plot of standard deviation of Δq_h versus number m of nodes in the mesh for $\lambda = 1, \sigma^2 = 3$ and $m_{\mathrm{KL}} = 35$ (left) and for $\lambda = 0.1, \sigma^2 = 1$ and $m_{\mathrm{KL}} = 105$ (solid line, right) and $m_{\mathrm{KL}} = 210$ (dashed line, right).

jump is the same for both values of m_{KL}, i.e. it is the same on very coarse meshes and very fine meshes.

Fixing the number of terms in our KL–expansion at a 'small' number such as 105 or 210 may seem somewhat artificial, since the random field in theory includes an infinite number of terms. By fixing m_{KL}, we are changing the problem that we are trying to solve and introducing an additional error. We need to make sure that the multilevel method is not dependent on this small, fixed choice of terms in the KL–expansion. We hence repeated the analysis of the difference terms also with larger values of m_{KL}. A graph of the result with $m_{\mathrm{KL}} = 5000$ is shown in Figure 5 (left, dashed line). We see that this choice of m_{KL} leads to a linear decrease in the standard deviation for all values of h (in the range that we investigated), i.e. it is the 'continuation' of the initial linear regime in Figure 4.

On the other hand, there is also the fact that fine scale oscillations cannot be represented by piecewise linear FEs on a coarse mesh. So including 105 or 210 modes on a mesh of size 1/8 for example, might not be necessary and seems somewhat pointless. So we tried to make the number of KL-modes that are used on each of the coarse grids dependent on the mesh size. In particular,

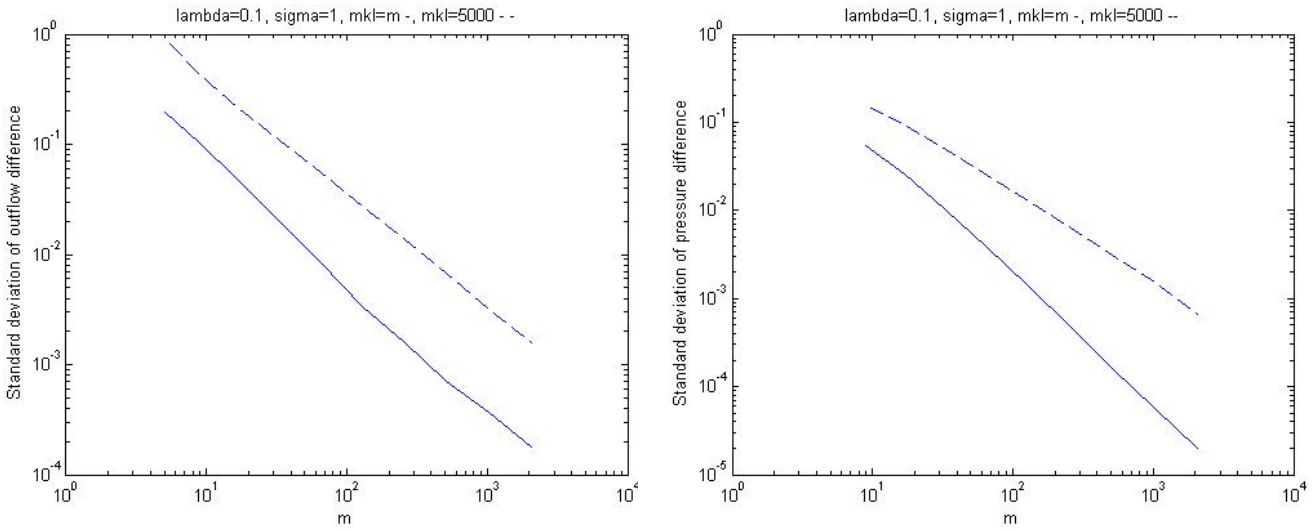

Figure 5: Plot of standard deviation of Δq_h (left) and of Δu_h^* (right) outflow versus number m of nodes in the mesh for $\lambda = 0.1$ and $\sigma^2 = 1$. The dashed line represents the case $m_{\mathrm{KL}} = 5000$ in both cases. The solid line the case $m_{\mathrm{KL}} = m$.

we studied Δu_h^* and Δq_h for the case where $m_{\mathrm{KL}} = m$ on each grid, i.e.$m_{\mathrm{KL}} = O(h^{-1})$. The results are shown in Figure 5 (solid lines). Somewhat surprisingly we see that the standard deviations of Δu_h^* and Δq_h are both significantly smaller and it seems that the decay is even of sligthly higher order in both cases, although not quite quadratic.

Finally, we note that in all cases the standard deviation of the difference terms for the coarsest meshes can be quite large, almost as large as the standard deviation of the quantity itself. As already noted at the end of the previous subsection it might therefore not always be worthwile to include all possible levels in the multilevel method.

1.7 Implementing the multilevel method

Note that we now estimate $\mathbb{E}[Q_h]$ by

$$\mathbb{E}[Q_h] \approx \frac{1}{N_1}\sum_{i=1}^{N_1}(Q_h^{(i)} - Q_{2h}^{(i)}) + \frac{1}{N_2}\sum_{j=1}^{N_2}(Q_h^{(j)} - Q_{2h}^{(j)}) + \cdots + \frac{1}{N_L}\sum_{m=1}^{N_L} Q_H^{(m)}.$$

The accuracy of our estimate of $\mathbb{E}[Q_h]$ is determined both by the FE error and by the MC error. Hence, to get an estimate to a required accuracy, we have to make sure that both of these errors are small enough.

First, we did some initial preliminary tests where we manually chose the number of samples, based on our computed values of the variances of the different quantities. The results were good.

We then went on to write a MATLAB code which automatically chooses the number of samples we need of each term, by monitoring the variance of this. Initially, the required total variance of the estimator was spread evenly across all terms. However, since the variance of the difference terms is extremely small for small h, we found that by doing this the multilevel method overshoots the required accuracy by a substantial amount. Since we also found out that most of the 'work' seems to be spent on the coarsest level, we decided to change the program slightly, such that the required variance of the coarse level term is determined as the difference of the required total variance, and the variance already accounted for by the difference terms.

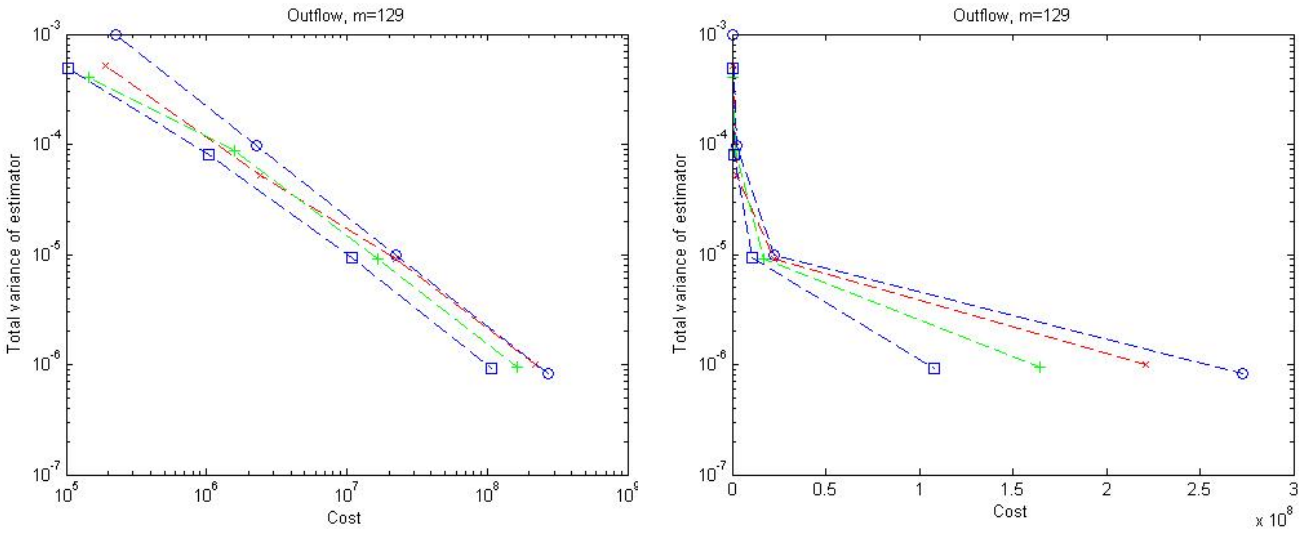

Figure 6: Plot of the total variance versus the standardised cost of the multilevel estimator and of the standard MC for $\mathbb{E}[q_h]$, for $m = 129$, $\lambda = 1$, $\sigma^2 = 1$ and $m_{\mathrm{KL}} = 5000$: Standard MC (purple $\circ$), 2-level MC (red $\times$), 3-level MC (green $+$), 4-level MC (blue $\square$); log-log scale (left) and linear-log scale (right).

The results presented in the remainder of this section refer to the 'computational cost' of the different methods. This was calculated as follows: For each sample, MATLAB has to solve a linear system $A\mathbf{U} = \mathbf{b}$ for the finite element solution. A is an $(m-1) \times (m-1)$ symmetric tridiagonal matrix, and so the solve is done using the Thomas Algorithm, which requires $8(m-1)$ operations. For standard Monte Carlo, the cost is hence $8N(m-1) = O(Nm)$. For the multilevel method, the cost of each term is calculated seperately, taking into account that we have to solve two such linear systems for the difference terms, and the total cost is then summed up. The 'standardised cost' in the graphs in the next section refers to the cost described above scaled by the constant $\frac{1}{8}$, which comes from the Thomas algorithm and is independent of m and N.

1.7.1 Including a large number of KL–modes

In the following figures (Figures 6–8) we plot the total variance of our multilevel estimator and of the standard MC estimator for $\mathbb{E}[q_h]$, as well as for $\mathbb{E}[u_h(0.75)]$ using a large number of KL–modes,i.e. $m_{\mathrm{KL}} = 5000$, in order to avoid any unexpected effects of the truncation of the KL–expansion.

In all of the three cases we see that the multilevel method outperforms standard Monte Carlo by a fair amount. However, the improvement is not drammatic. This is not surprising. The computational cost in 1D is proportional to m, and so going from a mesh h to a mesh $2h$, the cost for one sample will be reduced by factor $\frac{1}{2}$. At the same time, we require the variance of the variance of Q_{2h} in the 2–level method to be around half the variance of Q_h in the standard method, and so we cannot expect great savings. Better results might be achieved if the total variance in the 2–level method is spread unevenly accross the two terms, but we have not yet tried this out. Note that of the three cases considered here, it is the outflow for $\lambda = 1, sigma^2 = 1$ which has the largest variance, and this is also the one where the multilevel method seems to give the best results.

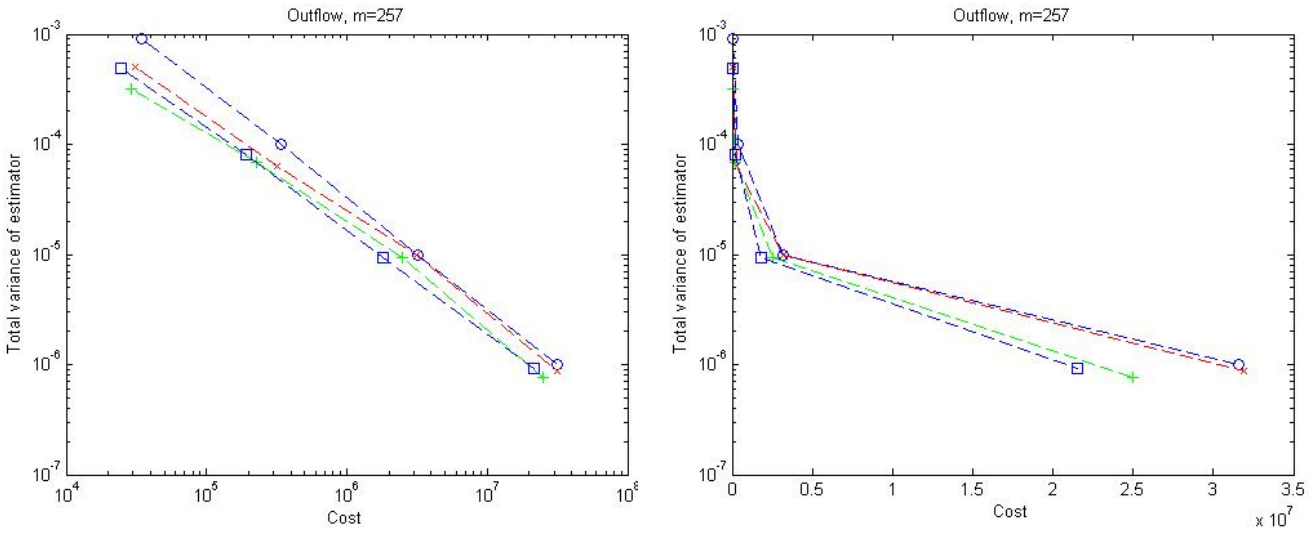

Figure 7: Same as Figure 6 but with $m = 257$ instead of $m = 129$.

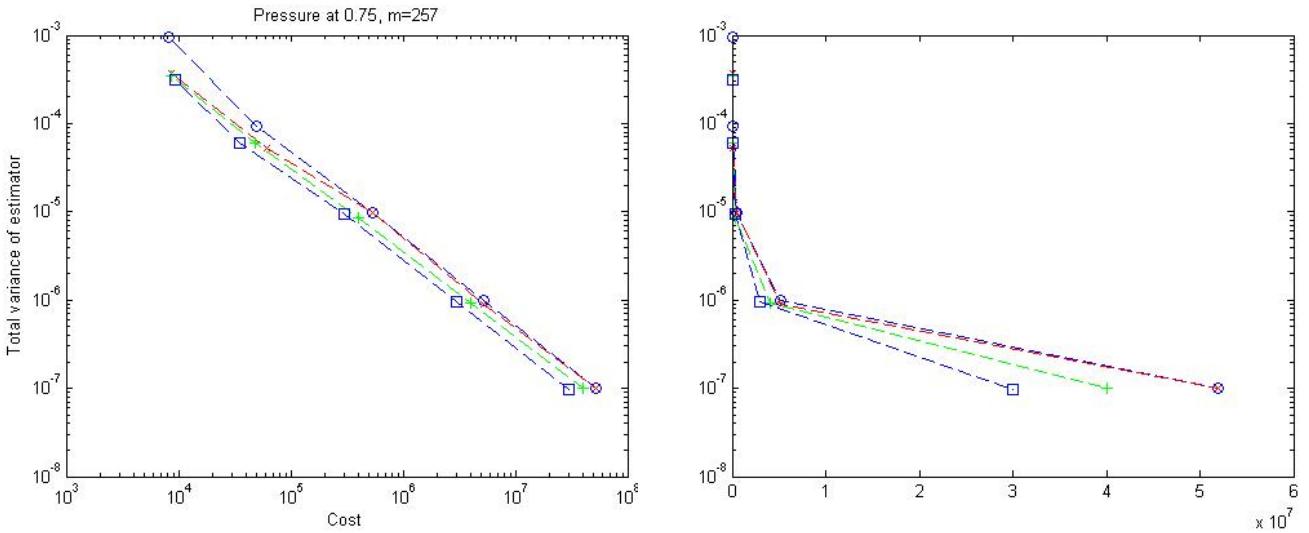

Figure 8: Same as Figure 6 but for $\mathbb{E}[u_h(0.75)]$ and $m = 257$.

1.7.2 Varying the number of KL modes according to mesh size

We saw earlier that a better decay rate for the variance of the difference terms can be achieved by reducing the number of KL modes proportionally to the number of mesh nodes. We hence expect the multilevel method to perform better and give greater savings, if we employ this strategy instead.

We consider again the case of the outflow q_h for $m = 257$, $\lambda = 0.1$, $\sigma^2 = 1$ and $m_{\mathrm{KL}} = m$. Results are shown in Figure 9. We see that the method with $m_{\mathrm{KL}} = m$ modes gives better results than that with a fixed number $m_{\mathrm{KL}} = 5000$, both in terms of standardised cost and CPU-time. Since sampling from the random field $k(x, \omega)$ and the setup of the stiffness matrix A are the computationally dominant part in the case $m_{\mathrm{KL}} = 5000$, we also performed the same test again with $m_{\mathrm{KL}} = 500$, to see whether changing m_{KL} is also of benefit in this case. It turns out it is.

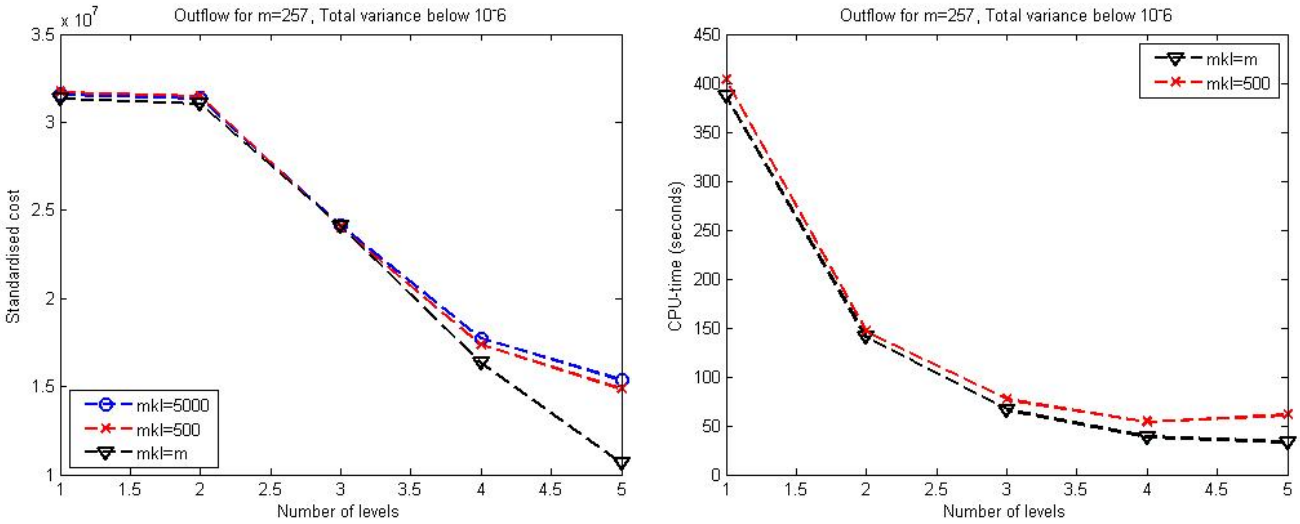

Figure 9: Plot of the standardised cost and the CPU-time versus the number of levels included in the method to obtain a total variance of the estimator less than 10^{-6}, for $m = 257, \lambda = 0.1, \sigma^2 = 1$ for various choices of m_{KL}.

2 Preliminary results on a 2D model problem

2.1 The model problem

We now consider the elliptic partial differential equation (PDE)

$$-\nabla.\,(k(\mathbf{x},\omega)\nabla u(\mathbf{x},\omega)) = 0, \qquad \mathbf{x} \in \Omega := (0,1)^2,$$

with random coefficient $k(x,\omega)$, and deterministic boundary conditions

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

We again model k as a lognormal random field, now with two–dimensional covariance function

$$C(\mathbf{x},\mathbf{y}) = \sigma^2 \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|_1}{\lambda}\right).$$

The benefit of using the norm $\|y\|_1 := |y_1| + |y_2|$ in this function is that we can again easily find analytic expressions for the eigenvalues and the eigenfunctions of the corresponding integral operator. The eigenvalues are simply the product of the eigenvalues of one–dimensional problems of the form (3) in the x_1 and in the x_2 direction, and the same is true for the eigenfunctions. In other words, an eigenfunction in 2D can be expressed as

$$f_{ij}^{2D}(\mathbf{x}) = f_i(x_1) f_j(x_2), \qquad \text{for some} \quad i,j \in \mathbb{N},$$

and the eigenvalues can be expressed as

$$\theta_{ij}^{2D} = \theta_i \theta_j, \qquad \text{for the same} \quad i,j \in \mathbb{N},$$

where (θ_i, f_i), $i \in \mathbb{N}$ are the 1D-eigenpairs in (4).

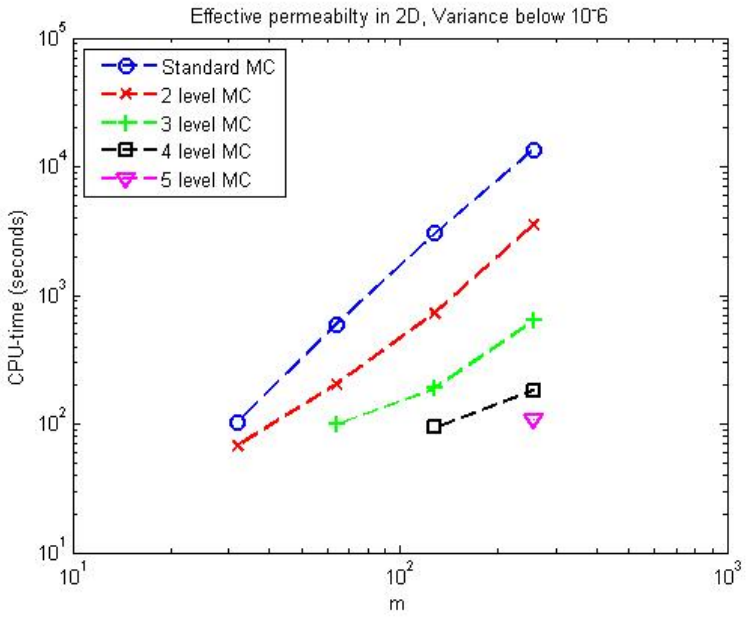

Figure 10: Plot of the CPU-time necessary to achieve a total variance of 10^{-6} for the estimator of $\mathbb{E}(k_{\text{eff}})$ versus number of discretisation points m in each coordinate direction, in the case $\lambda = 0.1,\ \sigma^2 = 1,\ m_{\text{KL}} = 500$.

2.2 Discretisation in space

At the moment we are using a cell-centred finite volume discretisation on a uniform mesh in 2D. The random field $k(\mathbf{x}, \omega)$ is sampled at the centre of the cells. The flow terms on cell faces are approximated by two-point finite differences using the harmonic average of the permeability k in the two cells adjacent to the face. This is the preferred discretisation scheme in subsurface flow in industry. Note however, that this is not essential. See Section 3.2 below for some more comments.

2.3 Implementing the Multilevel method

We implemented the multilevel method in 2D in a MATLAB code. The code again monitors the variance of each term in order to choose the number of samples we need to take, and spreads the total variance evenly across all terms, giving the coarse term extra lenience if the differences overshot their required accuracy.

As our quantity of interest, we have for the moment concentrated on the effective permeabilty (in x_1-direction), which in the case $\Omega = (0,1)^2$ and for the boundary conditions above is

$$k_{\text{eff}}(\omega) = \int_\Omega k(\mathbf{x}, \omega) \frac{\partial u}{\partial x_1}(\mathbf{x}, \omega)\, \mathrm{d}\mathbf{x}.$$

Note that it can be easily shown (see [Graham, Kuo et al, 2010]) that this is equal to the average outflow from Ω at $x_1 = 1$, and is therefore the 2D-equivalent to q_h studied above in 1D.

Results on the performance of the multilevel method in the case $\lambda = 0.1, \sigma^2 = 1$ are shown in Figures 10 and 11, where we chose $m_{\text{KL}} = 500$ terms in the KL expansion of $\log(k)$.

We have also done some of the above tests for the harder problem $\lambda = 0.05, \sigma^2 = 4$. These are shown in Figure 12. Some data points are still missing, but we see a similar trend. Note however, that we clearly observe what we already discussed in Section 1, namely that it is not

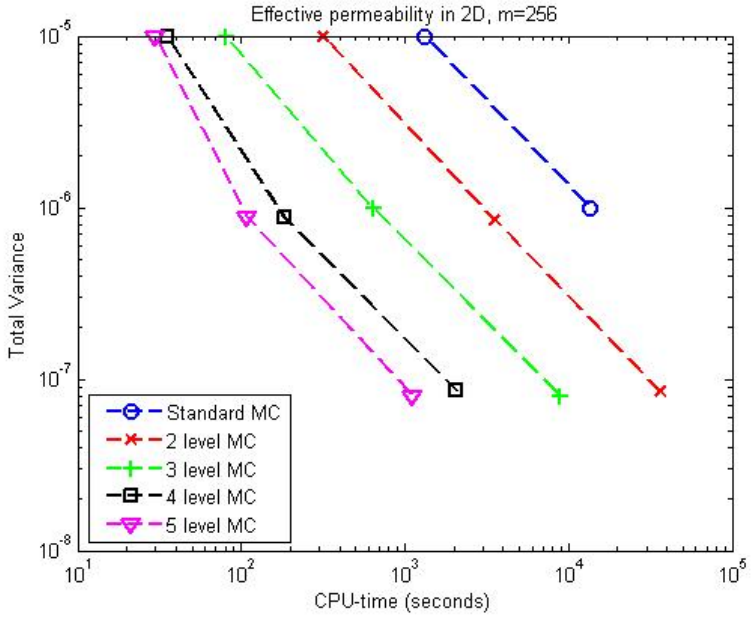

Figure 11: Plot of total variance versus CPU-time for fixed m, in the case $\lambda = 0.1,\ \sigma^2 = 1,\ m_{\mathrm{KL}} = 500$.

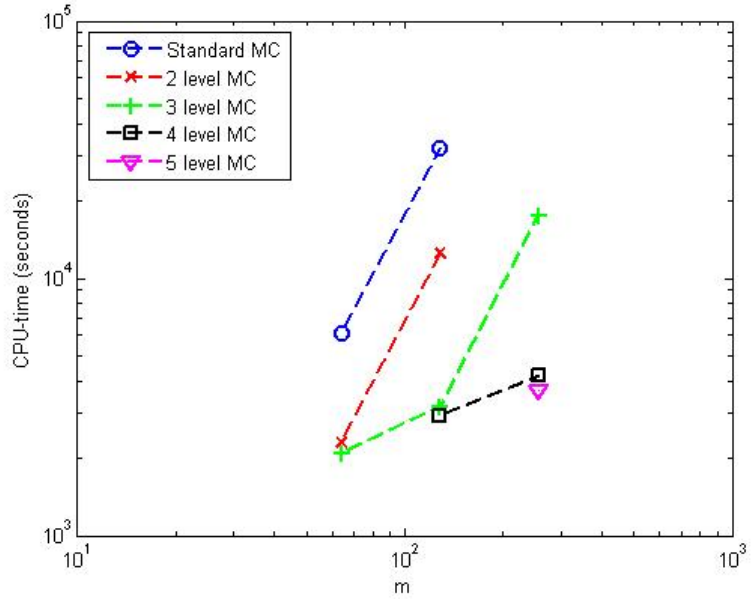

Figure 12: Plot of the CPU-time necessary to achieve a total variance of 10^{-6} versus m for a fixed maximum variance, in the case $\lambda = 0.05,\ \sigma^2 = 4,\ m_{\mathrm{KL}} = 500$.

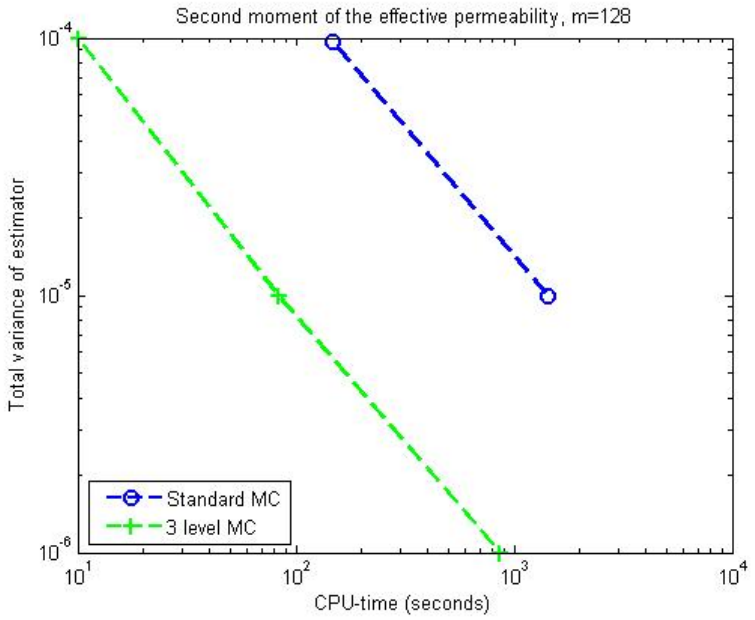

Figure 13: Plot of total variance versus CPU-time for fixed m for the second moment $\mathbb{E}(k_{\text{eff}}^2)$, in the case $\lambda = 0.1,\ \sigma^2 = 1,\ m_{\text{KL}} = 500$.

beneficial to coarsen the grid too much. In Figure 12 we see clearly that for smaller grid sizes the 2-level and the 3-level method preform equally well.

Similarly, we also checked that the excellent performance of our multilevel approach extends to higher-order moments and is not just restricted to the expected value of our quantities of interest. In Figure 13 we plot some preliminary results for the second moment, i.e. for $\mathbb{E}(k_{\text{eff}}^2)$.

3 Further Comments

In this short section we just wanted to collect some further comments regarding the model problem, the quantities of interest, the discretisation and the different sampling techniques.

3.1 Model problem and quantities of interest

Note that the model problem we are studying here and have been studying also in the joint paper with [Graham, Kuo et al, 2010] is very similar to the one outlined in Section 6 of [Giles, "Error Analysis and Variance Convergence ...", 2010] for oil reservoir homogenisation. The only difference is the choice of boundary conditions. Instead of periodic boundary conditions we choose homogeneous Neumann boundary conditions at the bottom and at the top of $[0,1]^2$ (in 3D this would simply be at the x_2 and the x_3 boundaries). In order to enforce (in our notation) $\mathbb{E}\left[-\int_\Omega \partial u/\partial x_1 \, dV\right] = 1$ (as at the end of Section 6.1 in [Giles, 2010]), it suffices then to choose $u = 1$ at $x_1 = 0$ and $u = 0$ at $x_1 = 1$, as chosen at the beginning of Section 2.

We are not sure whether one formulation is favourable over the other from a theoretical, from an engineering or from a practical point of view. The formulation in [Giles, 2010] seems more realistic in an upscaling or homogenisation type setting. However, our formulation here has some practical advantages. In particular, it suffices to compute the average over the outflow boundary ($x_1 = 1$), since it can be shown quite easily (see [Graham, Kuo, et al, 2010]) that the average over x_2 (and over x_3 in 3D) is constant as a function of x_1.

The only other quantities of interest we have studied so far (in [Graham, Kuo, et al, 2010]) are related to statistics of particle paths, computed from the velocity field $\vec{q} = k\nabla u$. Depending on the discretisation used (see below) these can be easily computed by simply computing trajectories of particles transported by the flow field in an element–by–element fashion, i.e. for each realisation of k we compute the trajectory of a particle released from a fixed point say $x^* = (0, 0.5)$ as a function of time t. Of interest are statistics (expected value, higher order moments, etc) of

- the position $y(t)$ at time t, since this can be used to study the spreading of a plume of pollutants released from a point source;
- the breakthrough time, i.e. the time it takes the particle to reach the outflow boundary ($x_1 = 1$).

(Andrew may be able to suggest some others.)

3.2 Spatial discretisations

We have used various discretisations in our projects. The improvements observed with Quasi-Monte Carlo (QMC) and with Multilevel Monte Carlo (MLMC) seem not to depend on the choice of discretisation. However, since the improvements that are possible with MLMC are crucially linked to the FE error on a given mesh, and therefore to the problem size that is necessary, it is nevertheless of importance to make a good and plausible choice.

For quantities related to the flow field, such as the effective permeability or the breakthrough time, it is usually argued that standard piecewise linear FEs are a bad choice because they are not locally mass conservative, which is why we have used mixed FEs in our work in [Cliffe, Graham, et al, 2000] and in [Graham, Kuo, et al, 2010]. It is also the reason why we have done our first test in 2D in Section 2 above using finite volume methods (which are also locally mass conservative).

A standard FE discretisation of the model problem is entirely classical and can be found in many text books. The only thing worth mentioning is that we use the midpoint rule to assemble the stiffness matrix. This has sufficiently high order so that the quadrature error is not the dominant part. In the case of highly varying coefficients (as we have them here) we'd need to do some further analysis, but we believe that it follows from the analysis in the appendix of the preprint for [Cliffe, Graham, et al, 2000].

For a short and simple description of mixed FEs for this model problem and for a fast method that reduces the computational task to solving a standard piecewise linear FE system, see [Cliffe, Graham, et al, 2000] or [Graham, Kuo, et al, 2010]. The analysis can possibly also be reduced to the standard FE case. See in particular Section 7 in [Graham, Kuo, et al, 2010] for further studies of the FE error for the quantities of interest discussed above in the case of mixed FEs.

We are not sure whether it's worth to pursue the theoretical analysis in the context of finite volume methods, but in any case, they are directly linked to mixed FEs with quadrature and thus we could also use them.

(Just so that it doesn't get lost, we have also put the little bit of analysis for the 1D problem that Rob had reported in his first note to this project as an appendix.)

3.3 Sampling from k

Finally, let us briefly discuss the different ways to sample from k. In the multilevel tests we have so far only worked with the truncated KL–expansion, and in light of the benefits that a

variable choice of KL–modes seem to have (see above), it seems like a good idea to continue this. However, truncation clearly leads to very large errors for the kind of parameter settings we are studying and when choosing large numbers of KL–modes the setup of the stiffness matrix becomes the dominant part.

Another issue with KL–expansion is that the simple explicit formula is restricted to covariance functions that are of tensor product type such as the one used in Section 2. For more realistic covariance functions in 2D and 3D it is necessary to solve the integral operator eigenproblem numerically. An efficient method to do this has recently been developed by Ernst et al (and we could obtain their code), but the cost (especially when large numbers of KL–modes are required) seems prohibitively large, but this would need to be studied in detail.

Alternatively, it would be of interest to use instead the circulant embedding techniques used and further developped in [Graham, Kuo, et al, 2010], see Section 5. Again, a C++ code exists that we could use directly to sample from k with this technique instead. The advantage here is that this sampling technique which uses FFT is exrtemely efficient.

Here is some more analysis of the 1D model problem as presented in Section 2 and analysed in Section 3 of [Giles, "Outline ideas for multilevel SPDE project", 2008].

First note that the (p.w. linear) finite element solution p_h of the model problem (on a uniform grid with mesh width h) depends on c only through the averages of c over each element, i.e. on

$$c_{h,i} := \frac{1}{h} \int_{(i-1)h}^{ih} c(x)\, dx\,, \qquad \text{for all } i = 1, \ldots, m$$

with $m = 1/h$, since the entries of the stiffness matrix A are given by

$$A_{ij} = \int_0^1 c\phi_i'\phi_j'\, dx = \begin{cases} \frac{1}{h^2}\left(\int_{(i-1)h}^{ih} c\, dx + \int_{(i)h}^{(i+1)h} c\, dx\right) \;=\; \frac{1}{h}(c_{h,i} + c_{h,i+1}), & \text{if } j = i, \\ -\frac{1}{h^2}\int_{(i-1)h}^{ih} c\, dx \;=\; -\frac{c_{h,i}}{h}, & \text{if } j = i-1. \end{cases}$$

So p_h also solves the approximate SDE with the piecewise constant coefficient function $c_h(x) = c_{h,i}$ for all $x \in ((i-1)h, ih)$ and $i = 1, \ldots, m$.

Now note secondly that the second identity in \$3.1 of [MG08] also holds true for the FE problem, i.e. if $M_h = c_h p_h'$ then

$$\int_0^1 c_h^{-1} M_h\, dx = 1$$

and hence

$$M_h = \left\{\int_0^1 c_h^{-1}\, dx\right\}^{-1} = \left\{\frac{1}{m}\sum_{i=1}^m c_{h,i}^{-1}\right\}^{-1}$$

which is just the harmonic average of the (integral) averages of c over each element.

The multilevel Monte Carlo approach is now to calculate the expected value $\mathbb{E}(M_h)$ in the following way. Since the expected value of a random variable is linear we can write

$$\mathbb{E}(M_h) = \mathbb{E}(M_h - M_{2h}) + \mathbb{E}(M_{2h} - M_{4h}) + \ldots + \mathbb{E}(M_H)$$

for some $H = 2^L h$, i.e. a sum of expected values of differences of approximations of M on two consecutive grids plus the expected value on the coarsest grid. The hope/conjecture is that the random variables $M_h - M_{2h}$ have a very small variance and are therefore easy to approximate via Monte Carlo using only very few realisations. The random variable M_H, on the other hand, is discretised on a much coarser grid and thus realisations of M_H should be easier/cheaper to obtain (even if we still might need the same number of realisations for a decent accuracy).

Before we can study the behaviour of these random variables numerically, we still need to make precise what we mean by M_{2h}. The difference lies in how we define the coefficients $\{c_{2h,i} : i = 1, \ldots, m/2\}$ on the coarse grid. There are several ways these may be defined and we highlight three of them:

1. Standard FE discretisation on the coarse grid, i.e. for all $i = 1, \ldots, m/2$:

$$c^{(1)}_{2h,i} \;:=\; \frac{1}{2h}\int_{(i-1)2h}^{i2h} c(x)\,dx \;=\; \frac{1}{2}(c_{h,2i-1} + c_{h,2i}),$$

   which is (thus) equivalent to taking the arithmetic average of the two coefficients on the fine grid corresponding to the two elements in $[(i-1)2h, i2h]$. (In multigrid this corresponds to standard FE interpolation or restriction via full-weighting.)

2. Using instead a harmonic average, i.e. for all $i = 1, \ldots, m/2$:

$$c^{(2)}_{2h,i} \;:=\; \left(\frac{1}{2}(c^{-1}_{h,2i-1} + c^{-1}_{h,2i})\right)^{-1}.$$

   (In multigrid methods this corresponds to multiscale or coefficient-dependent prolongation/restriction a la [Wan, Chan, Smith, SISC 2000], [Graham, Lechner, Scheichl, Num.Math. 2007] or [Dendy et al] (in BoxMG).)

3. The most interesting choice from a practical point is to sample $c(x)$ at the cell centre, i.e.

$$c^{(3)}_{2h,i} \;:=\; c((2i-1)h).$$

Sampling the random field can be as costly as solving the PDEs (certainly in 1D). Choices 1 and 2 are less favourable from that point of view, since even for the expected values on the coarser grids we require samples of c in all the fine grid elements. However, they may still be interesting if the solution of the PDE is the dominant part.

The 1D case is somewhat special (as already mentioned above). This is particularly the case for Choice 2 for $c_{2h,i}$, since

$$\begin{aligned}
M_h - M^{(2)}_{2h} &= \left\{\frac{1}{m}\sum_{i=1}^{m} c^{-1}_{h,i}\right\}^{-1} - \left\{\frac{2}{m}\sum_{i=1}^{m/2}(c^{(2)}_{2h,i})^{-1}\right\}^{-1} \\
&= \left\{\frac{1}{m}\sum_{i=1}^{m} c^{-1}_{h,i}\right\}^{-1} - \left\{\frac{2}{m}\sum_{i=1}^{m/2}\frac{1}{2}(c^{-1}_{h,2i-1} + c^{-1}_{h,2i})\right\}^{-1} \\
&= \left\{\frac{1}{m}\sum_{i=1}^{m} c^{-1}_{h,i}\right\}^{-1} - \left\{\frac{1}{m}\sum_{i=1}^{m} c^{-1}_{h,i}\right\}^{-1} \;=\; 0
\end{aligned}$$

Hence we have

$$\mathbb{E}(M_h) \;=\; \mathbb{E}(M^{(2)}_H)$$

where $c^{(2)}_{H,i}$ has to be recursively obtained via the formulae above. This seems to be a great result, but it is not that astonishing given the special form of the exact and the FE solution. Numerical tests have shown that this is correct.

In Choice 1, on the other hand, the terms do not cancel and we have a mix of harmonic and arithemtic averaging. There might be some possibility of exploiting results on those averages for correlated random fields, but I haven't pursued this yet.