Homework 2
Numerical methods for random partial differential equations: hierarchical approximation and machine learning approaches

May 12, 2023

IMPORTANT: Please refer to the course syllabus and the course outline for what an acceptable level on a handed in homework is.

1 QMC under exact sampling

1.1 Objective

Hands-on experience on quasi-Monte Carlo sampling methods under the idealised assumptions of exact sampling. Tasks include the numerical verification and computable confidence regions. The idea is to connect theory and practical aspects and experiment with the methods.

1.2 Error estimation and generating quasi-Monte Carlo points from the Normal distribution

Due to the deterministic nature of the low-discrepancy sequences used in the quasi-Monte Carlo integration we do not have a direct way of estimating the error. This problem can be overcome by Randomization of Quasi-Monte Carlo techniques. See Tuffin [2004].

To generate quasi-Monte Carlo points from the Normal distribution you can use the inverse transform technique, see Ökten and Göncü [2011]. That is, generate quasi-Monte Carlo points in $[0,1]^d$ and evaluate the inverse CDF, of the Multivariate Normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, on the generated points.

1.3 Problem

As in Homework 1, we are interested in calculating the following quantity

$$g = \int_{[0,1]^N} f(\mathbf{x})\, d\mathbf{x}$$

for some given function f. We try different cases, which mainly differ by the degree of regularity of the function f. In this exercise f is known and the problem is simply integrating the function in a high dimension N.

1.4 Model problems

We look at different examples of f for some real constants $\{c_n, w_n\}_{n=1}^N$ taken from

1. Oscillatory: $f(\mathbf{x}) = \cos\left(2\pi w_1 + \sum_{n=1}^N c_n x_n\right)$, with $c_n = 9/N$, $w_1 = \frac{1}{2}$.
   Exact solution:

   $$\int_{[0,1]^N} f(\mathbf{x})\, d\mathbf{x} = \Re\left(e^{i2\pi w_1} \prod_{n=1}^N \frac{1}{ic_n}(e^{ic_n} - 1)\right)$$

   (here i denote the imaginary unit and $\Re(z)$ the real part of $z \in \mathbb{C}$)

   $$\int f(\mathbf{x})\, d\mathbf{x} = \begin{cases} (-1)^N \sin\left(2\pi w_1 + \sum_{i=1}^N c_i x_i\right) \prod_{i=1}^N c_i^{-1} & N \text{ is odd} \\ (-1)^N \cos\left(2\pi w_1 + \sum_{i=1}^N c_i x_i\right) \prod_{i=1}^N c_i^{-1} & N \text{ is even} \end{cases}$$

2. Product peak: $f(\mathbf{x}) = \prod_{n=1}^N \left(c_n^{-2} + (x_n - w_n)^2\right)^{-1}$, with $c_n = 7.25/N$ and $w_n = \frac{1}{2}$.
   Exact solution:

   $$\int_{[0,1]^N} f(\mathbf{x})\, d\mathbf{x} = \prod_{n=1}^N c_n \left(\arctan(c_n(1 - w_n)) + \arctan(c_n w_n)\right)$$

3. Corner peak: $f(\mathbf{x}) = \left(1 + \sum_{n=1}^N c_n x_n\right)^{(-N+1)}$
   with $c_n = 1.85/N$.

4. Gaussian: $f(\mathbf{x}) = \exp\left(-\sum_{n=1}^N c_n^2 (x_n - w_n)^2\right)$, with $c_n = 7.03/N$ and $w_n = \frac{1}{2}$.
   Exact solution:

   $$\int_{[0,1]^N} f(\mathbf{x})\, d\mathbf{x} = \prod_{n=1}^N \frac{\sqrt{\pi}}{2c_n} \left(\operatorname{erf}(c_n(1 - w_n)) + \operatorname{erf}(c_n w_n)\right)$$

5. Continuous: $f(\mathbf{x}) = \exp\left(-\sum_{n=1}^{N} c_n|x_n - w_n|\right)$,with $c_n = 2.04/N$ and $w_i = \frac{1}{2}$.
   Exact solution:

$$\int_{[0,1]^N} f(\mathbf{x})\, d\mathbf{x} = \prod_{n=1}^{N} \frac{1}{c_n}\left(2 - e^{-c_n w_n} - e^{-c_n(1-w_n)}\right)$$

6. Discontinuous: $f(\mathbf{x}) = \begin{cases} 0 & \text{if } x_1 > w_1 \text{ or } x_2 > w_2 \\ \exp\left(\sum_{n=1}^{N} c_n x_n\right) & \text{otherwise} \end{cases}$, with $c_i = 4.3/N$, $w_1 = \frac{\pi}{4}$ and $w_2 = \frac{\pi}{5}$.
   Exact solution:

$$\int_{[0,1]^N} f(\mathbf{x})\, d\mathbf{x} = \frac{1}{\prod_{n=1}^{N} c_n}(e^{c_1 w_1} - 1)(e^{c_2 w_2} - 1)\prod_{n=3}^{N}(e^{c_n} - 1)$$

1.5 Numerical tasks

1. Use a quasi-Monte Carlo method, based on the Sobol sequence, to estimate these integrals. You can use the function `i4_sobol_generate.m` to generate sequences. This package has been taken from John Burkardt's home page.
   `https://people.sc.fsu.edu/~jburkardt/m_src/sobol/sobol.html`,

   Syntax: `R = i4_sobol_generate (N,M,0)` generates a matrix R(N,M) corresponding to M vectors of dimension N. Average the error over S sequences, where for each sequence, use `rand()` to add a random shift to the quasi-Monte Carlo points and `mod()` to limit these locations to [0,1].

   See also `https://people.sc.fsu.edu/~jburkardt/py_src/sobol/sobol.html` for a python version, or `https://pypi.org/project/qmcpy/`.

2. Estimate the resulting error using Randomized quasi-Monte Carlo and the Central Limit Theorem. Plot the exact error and error estimates versus the total number of samples of the S sequences. Estimate the convergence rate. Do your computations at least for dimension $N = 2$ and $N = 20$.

3. Consider the approximation of the following integral, for values of $K = 3$ and $K = 6$

$$\frac{1}{2\pi}\int_{\mathbb{R}^2} \max\left(e^{x_1} + e^{x_2} - K, 0\right) e^{\frac{-(x_1^2+x_2^2)}{2}} dx_1 dx_2$$

   Use quasi-Monte Carlo combined with importance sampling based on the shift-dilation technique (same as used in Homework1), and compare it to the standard quasi-Monte Carlo estimator. Use the Central Limit Theorem to estimate and control the statistical error.

4. Comment on your findings in relation to the results you obtained in Homework 1.

2 QMC for PDEs using approximate sampling

2.1 Objective

This exercise gives hands-on experience on quasi-Monte Carlo sampling for approximate (discretized) sampling. It also makes the student work on sampling Gaussian random fields and thus experimenting with Karhunen-Loève and Fourier expansions.

2.2 Problem

Next we look at the solution of a one dimensional boundary value problem

$$-(a(x,\omega)u'(x,\omega))' = \underbrace{4\pi^2 \cos(2\pi x)}_{=:f(x)}, \text{ for } x \in (0,1)$$

with $u(0,\cdot) = u(1,\cdot) = 0$. In this case, we are interested in computing the expected value of the following QoI

$$Q(u(\omega)) = \int_0^1 u(x,\omega)dx.$$

We use an $I+1$ uniform grid $0 = x_0 < x_1 < \ldots < x_I = 1$ on $[0,1]$ with uniform spacing $h = x_i - x_{i-1} = 1/I$, $i = 1, \ldots, I$. Using this grid we can build a piecewise linear FEM approximation,

$$u_h(x,\omega) = \sum_{i=1}^{I-1} \mathbf{u}_{h,i}(\omega)\varphi_i(x),$$

yielding a tridiagonal linear system for the nodal values, $A(\omega)\mathbf{u}_h(\omega) = F$, with

$$\begin{aligned}
A_{i,i-1}(\omega) &= -\frac{a(x_{i-1/2},\omega)}{h^2} \\
A_{i,i}(\omega) &= \frac{a(x_{i-1/2},\omega) + a(x_{i+1/2},\omega)}{h^2} \\
A_{i,i+1}(\omega) &= -\frac{a(x_{i+1/2},\omega)}{h^2}
\end{aligned}$$

and

$$F_i = f(x_i).$$

Here we used the notation $x_{i+1/2} = \frac{x_i + x_{i+1}}{2}$ and $x_{i-1/2} = \frac{x_i + x_{i-1}}{2}$.

The integral in $Q(u_h)$ can be then computed exactly by a trapezoidal method, yielding

$$Q(u_h) = h \sum_{i=1}^{I-1} \mathbf{u}_{h,i}.$$

2.3 Models for the random coefficient a.

We look at two different models for $a(x,\omega)$, namely

2.3.1 Model 1: Piecewise constant coefficient

$$a(x,\omega) = 1 + \sigma \sum_{n=1}^{N} Y_n(\omega)\mathbb{I}_{[\hat{x}_{n-1},\hat{x}_n]}(x),$$

with equispaced nodes $\hat{x}_n = \frac{n}{N}$ for $0 \leq n \leq N$ and i.i.d. uniform random variables $Y_n \sim U([-\sqrt{3},\sqrt{3}])$. Consider different uniform mesh refinements, i.e. values of $I = N\,2^{\ell}$, $\ell \geq 0$, and different number of input random variables $N = 10$, $N = 20$ and $N = 40$. Remember to choose σ ensuring coercivity, namely $1 - \sigma\sqrt{3} > 0$.

2.3.2 Model 2: Log-Normal

$$a(x,\omega) = \exp(\kappa(x,\omega)),$$

where $\kappa(x,\omega)$ is a stationary Gaussian random field with mean zero and the Matérn covariance function

$$C(x,y) = \sigma^2 \frac{1}{\Gamma(\nu)2^{\nu-1}} \left(\sqrt{2\nu}\frac{|x-y|}{\rho}\right)^{\nu} K_\nu\left(\sqrt{2\nu}\frac{|x-y|}{\rho}\right),$$

where Γ is the gamma function and K_ν is the modified Bessel function of the second kind. We look at the following special cases of C

$$\begin{aligned}
\nu = 0.5, \quad C(x,y) &= \sigma^2 \exp\left(-\tfrac{|x-y|}{\rho}\right)\\
\nu = 1.5, \quad C(x,y) &= \sigma^2 \left(1 + \tfrac{\sqrt{3}|x-y|}{\rho}\right)\exp\left(-\tfrac{\sqrt{3}|x-y|}{\rho}\right)\\
\nu = 2.5, \quad C(x,y) &= \sigma^2 \left(1 + \tfrac{\sqrt{5}|x-y|}{\rho} + \tfrac{\sqrt{3}|x-y|^2}{\rho^2}\right)\exp\left(-\tfrac{\sqrt{5}|x-y|}{\rho}\right)\\
\nu \to \infty, \quad C(x,y) &= \sigma^2 \exp\left(-\tfrac{|x-y|^2}{2\rho^2}\right).
\end{aligned}$$

Moreover we choose $\rho = 0.1$ and $\sigma^2 = 2$.

2.4 Possible numerical approximations of Log-Normal random fields

1. Given a grid, consider multivariate Gaussian random vector X_h with mean μ_h covariance matrix Σ_h consisting of the Gaussian random field's covariance function evaluated on the grid. Let A such that $\Sigma_h = AA^T$ (e.g., via Cholesky or spectral decomposition). Sampling of X_h then via the representation $X_h = \mu_h + A\,Z$, where $Z \sim \mathcal{N}(0,I)$.

2. We expand $\kappa(x,\omega)$ using truncated Karhunen-Loève expansion with N terms

$$\kappa(x,\omega) \approx \sum_{n=1}^{N} \sqrt{\lambda_n} Y_n(\omega) e_n(x)$$

Where $\{Y_k\}$ is a set of i.i.d. standard Gaussian. To find the eigenfunctions and eigenvalues of C we solve the eigenvalue problem

$$\int_0^1 C(x,y)e_n(y)\;\mathrm{d}y = \lambda_n e_n(x) \tag{1}$$

We do this by discretizing C as a matrix by evaluating the function $C(x_{i+\frac{1}{2}}, x_{j+\frac{1}{2}})$ over the grid $\{x_{i+\frac{1}{2}}\}_{i=0}^{I-1} \times \{x_{i+\frac{1}{2}}\}_{i=0}^{I-1}$ with $N \leq I$. Then use `MATLAB`'s function `eig()`. This discretization corresponds to a piecewise constant FEM approximation to the eigenvalue problem (1). Make sure that the computed eigenfunctions have norm 1 with respect to the continuous $L^2([0,1])$ norm.

3. Now, with the same covariance field as before, use a Fourier expansion of a periodic extension of it, namely

$$Cov_\kappa^{\#}(x-y) = \sum_{n=0}^{\infty} \kappa_n^2 \cos\left(\frac{n\pi(x-y)}{L_p}\right).$$

Take $L_p = 2$ for instance. As a result, the stationary random field κ admits the exact representation in $[0, L_p]$

$$\kappa(\omega, x) = \sum_{n=0}^{\infty} \kappa_n \left(y_n(\omega)\cos\left(\frac{n\pi x}{L_p}\right) + z_n(\omega)\sin\left(\frac{n\pi x}{L_p}\right)\right)$$

and

- $\mathbb{E}[y_n] = \mathbb{E}[z_n] = 0$
- $Var[y_n] = Var[z_n] = 1$
- $\{y_n, z_n\}_n$ uncorrelated.

Here take iid y_n, z_n with standard Normal distribution.

2.5 Numerical approximation of $\mathbb{E}[Q(u)]$.

For the following questions (1) and (2), use Model 1 or Model 2 for $a(x,\omega)$. For question (3), use Model 2 for $a(x,\omega)$. When working with Model 2, use the last task of 1.4 in homework 1 to deduce a relevant choice of N. Moreover, pick two different values of the parameter ν.

1. Repeat Exercise 1 for the approximation of $\mathbb{E}[Q(u_h)]$. To this end, compute it using plain quasi-Monte Carlo sampling. Estimate the statistical error using Randomized quasi-Monte Carlo and the Central Limit Theorem. Estimate the bias error using Richardson extrapolation with quasi-Monte Carlo. Choose the number of samples and the mesh size h such that the relative error is less than 10%.

2. Use a coarser discretization of the problem (say with double the grid-size h) with the same realization of $\{Y_k\}$ as a control variate for the approximation of $\mathbb{E}[Q(u_h)]$ and check the variance reduction in quasi-Monte Carlo sampling. Does it pay to use it?

3. Approximate the probability $P(Q(u) < K)$ using importance sampling (same as used in Homework1) and quasi-Monte Carlo. Compute with different values of K,

namely $K = -5$, $K = -10$, and $K = -20$. Estimate the statistical error with the Central Limit Theorem choosing the number of samples so that the relative error is less than 10%. Use the same mesh size as in question 1.

4. Compare your findings with your findings in Homework 1.

5. Using the two following hints, propose two ways to estimate the same probabilities as in Question 3 using Large deviations theory.
   i) Use the following bound

$$|Q(u)| \leq \tilde{q}_0 \exp\left(\sum_{n\geq 1} ||b_n||_\infty |Y_n|\right),$$

   where $\tilde{q}_0$ and b_n, $n \geq 1$, are to be determined.
   ii) Use the Chernoff bound

$$P(Q(u) < K) \leq \exp\left(-I_{-Q(u)}(-K)\right),$$

   where $I_{-Q(u)}(x) = \sup_{\theta>0}\{\theta x - \Lambda_{-Q(u)}(\theta)\}$ with

$$\Lambda_{-Q(u)}(\theta) = \log\left(\mathbb{E}[\exp(-\theta Q(u))]\right).$$

   For a given $h > 0$ and $\theta > 0$, approximate $\mathbb{E}[\exp(-\theta Q(u))$ by

$$\mathbb{E}[\exp(-\theta Q(u))] \approx \frac{1}{M}\sum_{i=1}^{M} \exp(-\theta Q(u_h^{(i)})),$$

   where $u_h^{(i)}$, $i = 1, \cdots, M$, are i.i.d realizations of u_h. Then, use the same samples $u_h^{(i)}$, $i = 1, 2, \cdots, M$, to approximate $I_{-Q(u)}(-K)$ numerically by repeating the same steps for several values of θ. Motivate the choice of h and M.

References

Bruno Tuffin. Randomization of quasi-monte carlo methods for error estimation: Survey and normal approximation*. *Monte Carlo Methods and Applications*, 10(3-4):617–628, 2004. doi: doi:10.1515/mcma.2004.10.3-4.617. URL `https://doi.org/10.1515/mcma.2004.10.3-4.617`.

Giray Ökten and Ahmet Göncü. Generating low-discrepancy sequences from the normal distribution: Box–muller or inverse transform? *Mathematical and Computer Modelling*, 53(5):1268–1281, 2011. ISSN 0895-7177. doi: https://doi.org/10.1016/j.mcm.2010.12.011. URL `https://www.sciencedirect.com/science/article/pii/S0895717710005935`.