Wintersemester 2022/23
Return your written solutions either in person or by email
to vesa.kaarnioja@fu-berlin.de by Tuesday 6 December, 2022, 12:15

Please note that there are a total of 4 tasks this week!

Instructions: Download the file `week7.mat` from the course webpage. The file contains FE matrices as well as other FEM objects corresponding to a FE discretization of the computational domain $D = (0,1)^2$. The file contains the stiffness tensor `grad`, mass matrix `mass`, FE nodes `nodes`, mesh element connectivity array `element`, a vector containing indices of the interior FE nodes `interior`, element center points `centers`, the number of FE coordinates `ncoord`, and the number of FE elements `nelem`. These were generated by the `FEMdata.m` MATLAB routine. In MATLAB, you can import the data using the command `load week7.mat`. In Python, this can be achieved via

```
import numpy as np
import scipy.io
mat = scipy.io.loadmat('week7.mat')
```

The contents can be accessed via `mat['grad']`, `mat['mass']`, `mat['nodes']`, etc.

1. Let $D = (0,1)^2$, $f(\boldsymbol{x}) = x_1$, and consider the following parametric PDE problem: for all $\boldsymbol{y} \in [-1/2, 1/2]^s$, find $u(\cdot, \boldsymbol{y}) \in H_0^1(D)$ such that

$$\int_D a(\boldsymbol{x},\boldsymbol{y}) \nabla u(\boldsymbol{x},\boldsymbol{y}) \cdot \nabla v(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} = \int_D f(\boldsymbol{x}) v(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} \quad \text{for all } v \in H_0^1(D),$$

endowed with the (dimensionally-truncated) uniform and affine diffusion coefficient

$$a(\boldsymbol{x},\boldsymbol{y}) = 2 + \sum_{k=1}^{s} y_k \psi_k(\boldsymbol{x}), \quad \boldsymbol{x} \in D,\ \boldsymbol{y} \in [-1/2, 1/2]^s,$$

with stochastic fluctuations $\psi_k(\boldsymbol{x}) := k^{-2} \sin(\pi k x_1) \sin(\pi k x_2)$.

Consider the problem of approximating

$$\mathbb{E}[u(\boldsymbol{x},\cdot)] = \int_{[-1/2,1/2]^s} u(\boldsymbol{x},\boldsymbol{y})\, \mathrm{d}\boldsymbol{y}$$

using the Monte Carlo method with stochastic dimension $s = 100$. That is, for several values of n, draw $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$ from $\mathcal{U}([-1/2, 1/2]^s)$ and compute

$$\mathbb{E}[u(\boldsymbol{x},\cdot)] \approx \frac{1}{n} \sum_{i=1}^{n} u(\boldsymbol{x},\boldsymbol{y}_i).$$

To solve the PDE numerically for each $\boldsymbol{y}_i$, you can use the FEM data stored in `week7.mat`. Fix $s = 100$ and estimate the $L^2(D)$ error by using the Monte Carlo estimate corresponding to $n' \gg n$ as a reference solution. What convergence rate do you obtain?

The exercises continue on the next page!

2. Let $D = (0,1)^2$ and consider the following parametric *spectral eigenvalue* problem: for all $\boldsymbol{y} \in [-1/2,1/2]^s$, find the *smallest* eigenpair $(\lambda(\boldsymbol{y}), u(\cdot,\boldsymbol{y})) \in (\mathbb{R} \times (H_0^1(D) \setminus \{\mathbf{0}\}))$, $\|u(\cdot,\boldsymbol{y})\|_{L^2(D)} = 1$, such that

$$\int_D a(\boldsymbol{x},\boldsymbol{y}) \nabla u(\boldsymbol{x},\boldsymbol{y}) \cdot \nabla v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \lambda(\boldsymbol{y}) \int_D u(\boldsymbol{x}) v(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \quad \text{for all } v \in H_0^1(D),$$

endowed with the (dimensionally-truncated) uniform and affine diffusion coefficient

$$a(\boldsymbol{x},\boldsymbol{y}) = 2 + \sum_{k=1}^{s} y_k \psi_k(\boldsymbol{x}), \quad \boldsymbol{x} \in D,\ \boldsymbol{y} \in [-1/2,1/2]^s,$$

with stochastic fluctuations $\psi_k(\boldsymbol{x}) := k^{-2} \sin(\pi k x_1) \sin(\pi k x_2)$.

Consider the problem of approximating

$$\mathbb{E}[\lambda(\cdot)] = \int_{[-1/2,1/2]^s} \lambda(\boldsymbol{y})\,\mathrm{d}\boldsymbol{y} \quad \text{and} \quad \mathbb{E}[u(\boldsymbol{x},\cdot)] = \int_{[-1/2,1/2]^s} u(\boldsymbol{x},\boldsymbol{y})\,\mathrm{d}\boldsymbol{y}$$

using the Monte Carlo method with stochastic dimension $s = 100$. That is, for several values of n, draw $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$ from $\mathcal{U}([-1/2,1/2]^s)$ and compute

$$\mathbb{E}[\lambda(\cdot)] \approx \frac{1}{n} \sum_{i=1}^{n} \lambda(\boldsymbol{y}_i) \quad \text{and} \quad \mathbb{E}[u(\boldsymbol{x},\cdot)] \approx \frac{1}{n} \sum_{i=1}^{n} u(\boldsymbol{x},\boldsymbol{y}_i).$$

To solve the PDE numerically for each $\boldsymbol{y}_i$, you can use the FEM data stored in `week7.mat`. Fix $s = 100$ and estimate the Euclidean error of $\mathbb{E}[\lambda(\cdot)]$ and the $L^2(D)$ error of $\mathbb{E}[u(\boldsymbol{x},\cdot)]$ by using the Monte Carlo estimate corresponding to $n' \gg n$ as a reference solution. What convergence rate(s) do you obtain?

3. Repeat task 1, but instead of using a Monte Carlo sample average to compute the expected value, use instead an *off-the-shelf lattice rule*. Download the file `offtheshelf.txt` from the course webpage. The file contains an *extensible*, 100-dimensional generating vector $\boldsymbol{z} \in \mathbb{N}^{100}$. For $n = 2^k$, $k \in \{10, 11, \ldots, 20\}$, you can compute the n-point QMC point set using the formula

$$\boldsymbol{y}_i = \operatorname{mod}\left(\frac{i\boldsymbol{z}}{n}, 1\right) - 0.5, \quad i = 0, 1, \ldots, n-1.$$

The QMC estimator using this *deterministic* point set is

$$\mathbb{E}[u(\boldsymbol{x},\cdot)] \approx \frac{1}{n} \sum_{i=0}^{n-1} u(\boldsymbol{x},\boldsymbol{y}_i).$$

To solve the PDE numerically for each $\boldsymbol{y}_i$, you can use the FEM data stored in `week7.mat`. Fix $s = 100$ and estimate the $L^2(D)$ error of the QMC approximation by using a QMC estimate corresponding to $n' \gg n$ as a reference solution. What convergence rate do you obtain?

The exercises continue on the next page!

4. Repeat task 2, but instead of using a Monte Carlo sample average to compute the expected value, use instead an *off-the-shelf lattice rule*. Download the file `offtheshelf.txt` from the course webpage. The file contains an *extensible*, 100-dimensional generating vector $\boldsymbol{z} \in \mathbb{N}^{100}$. For $n = 2^k$, $k \in \{10, 11, \ldots, 20\}$, you can compute the n-point QMC point set using the formula

$$\boldsymbol{y}_i = \mathrm{mod}\left(\frac{i\boldsymbol{z}}{n}, 1\right) - 0.5, \quad i = 0, 1, \ldots, n-1.$$

The QMC estimators using this *deterministic* point set are

$$\mathbb{E}[\lambda(\cdot)] \approx \frac{1}{n}\sum_{i=0}^{n-1} \lambda(\boldsymbol{y}_i) \quad \text{and} \quad \mathbb{E}[u(\boldsymbol{x}, \cdot)] \approx \frac{1}{n}\sum_{i=0}^{n-1} u(\boldsymbol{x}, \boldsymbol{y}_i).$$

To solve the PDE numerically for each $\boldsymbol{y}_i$, you can use the FEM data stored in `week7.mat`. Fix $s = 100$ and estimate the Euclidean error of $\mathbb{E}[\lambda(\cdot)]$ and the $L^2(D)$ error of the QMC approximations by using a QMC estimate corresponding to $n' \gg n$ as a reference solution. What convergence rate(s) do you obtain?

Hints: It should be possible to complete these tasks by modifying the files `ex4.m` / `ex4.py` and `lognormal_demo.m` / `lognormal_demo2.m` available on the course page appropriately.

Python users: You will need to implement an analogue of the `UpdateStiffness.m` routine. For example, something like the following should work:

```
import numpy as np
from scipy import sparse

def UpdateStiffness(grad,a):
    n = np.sqrt(grad.shape[0]).astype(int)
    return (grad@sparse.csr_matrix(a.reshape((a.size,1)))).reshape((n,n))
```