Programming sheet 4.

Meeting week[1]: **26. – 30. May 2025**

Hierarchical tensor decomposition

In this sheet, we want to separate the variables by the hierarchical tensor decomposition to solve the high-dimensional problem.

Let $\square \subset \mathbb{R}^2$ be the unit square and let $J \in \mathbb{N}$. We consider the following four-dimensional problem: find the covariance function $\mathrm{Cov}[u] \in H_0^1(\square) \otimes H_0^1(\square)$ satisfying

$$\begin{cases} (\Delta_{\mathbf{x}} \otimes \Delta_{\mathbf{y}})\,\mathrm{Cov}[u] = \mathrm{Cov}[f] \ \text{ in } \ \square \times \square, \\ \qquad\qquad\quad \mathrm{Cov}[u] = 0 \ \text{ on } \ \partial(\square \times \square). \end{cases} \tag{1}$$

In order to solve (1) numerically, we apply the hierarchical tensor decomposition to the right-hand side $\mathrm{Cov}[f](\mathbf{x}, \mathbf{y})$, i.e.,

$$\mathrm{Cov}[f]\big((x_1, x_2), (y_1, y_2)\big) \approx \sum_{k=1}^{m} \sum_{j=1}^{r_k} \sum_{j'=1}^{r_k} v_{j,k}(x_1) \otimes w_{j,k}(x_2) \otimes v_{j',k}(y_1) \otimes w_{j',k}(y_2).$$

We can discretize of Laplace operator on $\square$ as a second-order tensor by the sum of two Kronecker products in accordance with

$$\Delta_{\mathbf{x}} \approx \mathbb{A}_{x_1} + \mathbb{A}_{x_2}, \quad \mathbb{A}_{\mathbf{x},1} := \mathbf{L}_{x_1} \otimes \mathbf{M}_{x_1}, \quad \mathbb{A}_{\mathbf{x},2} := \mathbf{M}_{x_2} \otimes \mathbf{L}_{x_2},$$

where

$$\mathbf{L}_{x_i} := \frac{1}{h^2} \begin{bmatrix} 2 & -1 & 0 & \cdots & 0 & 0 \\ -1 & 2 & -1 & \ddots & & 0 \\ 0 & -1 & 2 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & -1 & 0 \\ 0 & & \ddots & -1 & 2 & -1 \\ 0 & 0 & \cdots & 0 & -1 & 2 \end{bmatrix}, \qquad \mathbf{M}_{x_i} := \frac{1}{6} \begin{bmatrix} 4 & 1 & 0 & \cdots & 0 & 0 \\ 1 & 4 & 1 & \ddots & & 0 \\ 0 & 1 & 4 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 1 & 0 \\ 0 & & \ddots & 1 & 4 & 1 \\ 0 & 0 & \cdots & 0 & 1 & 4 \end{bmatrix}. \tag{2}$$

Thus, the tensor-product Laplacian is given as the fourth-order tensor

$$\Delta_{\mathbf{x}} \otimes \Delta_{\mathbf{y}} \approx \mathbb{A}_{\mathbf{x},1} \otimes \mathbb{A}_{\mathbf{y},1} + \mathbb{A}_{\mathbf{x},1} \otimes \mathbb{A}_{\mathbf{y},2} + \mathbb{A}_{\mathbf{x},2} \otimes \mathbb{A}_{\mathbf{y},1} + \mathbb{A}_{\mathbf{x},2} \otimes \mathbb{A}_{\mathbf{y},2}$$

and the right-hand side is given as

$$\mathbb{F} = \sum_{k=1}^{m} \sum_{j=1}^{r_k} \sum_{j'=1}^{r_k} \mathbf{M}_{x_1} V_{j,k} \otimes \mathbf{M}_{x_2} W_{j,k} \otimes \mathbf{M}_{y_1} V_{j',k} \otimes \mathbf{M}_{y_2} W_{j',k} \tag{3}$$

In order to solve the problem (1) numerically, we just need to solve the system $\mathbb{A}\mathbb{U} = \mathbb{F}$.

To work on this sheet, you need to install the *Hierarchical Tucker Matlab Toolbox*. The manual and files are found here: HTucker Toolbox.

[1] If necessary, the deadline can be extended by a week.

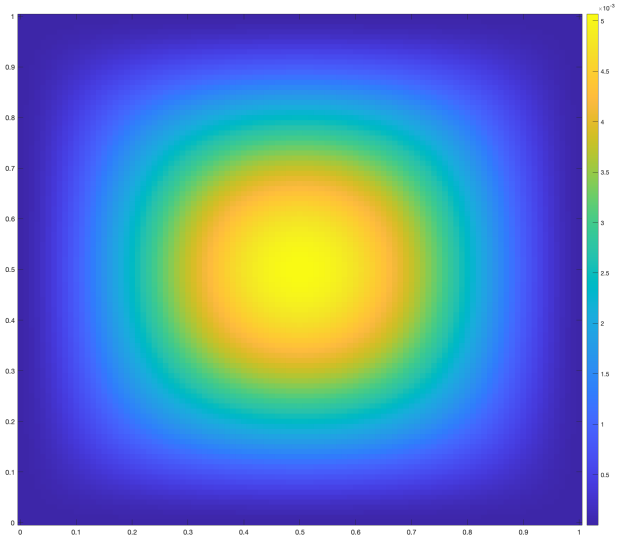

Figure 1: Variance $\mathrm{Var}[u](\mathbf{x}) = \mathrm{Cov}[u](\mathbf{x},\mathbf{x})$.

Exercise 1. To get the tensor decomposition (3) of the right-hand side, implement a Matlab function

```
function [U,V] = ht_decomp(ker, xs, tol),
```

which returns a 4d `htucker` tensor `function [Phi,Psi]`. We intend to approximate the right-hand side on a uniform product grid on □×□. It is based on the uniform grid `xs = [x1(:), x2(:)]'`, where

```
[x1, x2] = meshgrid(linspace(0, 1, n), linspace(0, 1, n))
```

with `n=101`.

We first compute the decomposition

$$\mathrm{Cov}[f](\mathbf{x},\mathbf{y}) \approx \mathbf{C}\mathbf{C}^\top = \widehat{\mathbf{V}}\mathbf{S}^2\widehat{\mathbf{V}}^\top$$

by the pivoted Cholesky decompostion with tolerance $\texttt{tol} = 10^{-6}$. To this end, compute first the singular-value decomposition $\mathbf{C}^\top\mathbf{C} = \widetilde{\mathbf{V}}\widetilde{\mathbf{S}}\widetilde{\mathbf{W}}^\top$. Then set $\widehat{\mathbf{V}} := \mathbf{C}\widetilde{\mathbf{W}}$ and matricize each column $\widehat{\mathbf{V}}_{:,k} \to \overline{\mathbf{V}}_k$ by the function `reshape` to compute its singular-value decomposition $[\mathbf{V}_k, \mathbf{S}_k, \mathbf{W}_k] = \mathtt{svd}(\overline{\mathbf{V}}_k)$. Truncate the singular-value decomposition to size r_k:

$$\mathbf{V}_k\mathbf{S}_k\mathbf{W}_k^\top \to \mathbf{V}_k^{r_k}\mathbf{S}_k^{r_k}(\mathbf{W}_k^{r_k})^\top,$$

where r_k is the largest index such that $\mathbf{S}_k(r_k, r_k) > \texttt{tol}$ for $k = 1, \dots, m$. Finally, assemble the matrices $\mathbf{V} = [\mathbf{V}_1^{r_1}\mathbf{S}_1^{r_1}, \dots, \mathbf{V}_m^{r_m}\mathbf{S}_m^{r_m}]$ and $\mathbf{W} = [\mathbf{W}_1^{r_1}, \dots, \mathbf{W}_m^{r_m}]$, which should be output of the function.

You can test your implementation as folllows:

```
ker = @(x , y) exp(-((x(1,:)-y(1,:).').^2+(x(2,:)-y(2,:).').^2));
n = 101; tol = 1e-6;
[x1 , x2] = meshgrid (linspace(0 , 1 , n), linspace(0, 1, n));
xs = [x1(:) , x2(:)]';
K = ker(xs, xs); [V,W] = ht_decomp(ker,xs,tol);
F = htensor({V, W, V, W});
Af = matricize(full(F),[1,2]);
norm(K-Af,"fro")
```

Exercise 2.

Take the same kernel and grid as before. Construct the matrices $\mathbf{L}$ and $\mathbf{M}$ as given in (2). Compute the right-hand side $\mathbb{F}$ by the function `htucker`. To solve the partial differential equation (1), use the following code:

```
A_handle = handle_lin_mat(L,M);
B_handle = handle_inv_mat(L);
opts.max_rank = 30; opts.rel_eps = tol;
opts.maxit = 30; opts.tol = 0;
opts.plot_conv = false;
[U, ~] = cg_tensor(A_handle, B_handle, F, opts)
```

Plot the variance of u. It is obtained by using the function `matricize` and then take the diagonal.