Programming sheet 3.

Meeting week: **28. April – 2. May 2025**

Sparse grid combination technique

Let $D \subset \mathbb{R}^2$ be a bounded and connected domain with boundary ∂D. We intend to compute the covariance function $\mathrm{Cov}[u] \in H_0^1(D) \otimes H_0^1(D)$ satisfying

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

In order to solve (1) numerically, we want to apply the sparse grid combination technique for level $J \in \mathbb{N}$ which means we should combine specific solutions of (1) in accordance with

$$\widehat{\mathbf{U}}_J = \sum_{j=1}^{J} \mathbf{u}_{j,J-j+1} - \mathbf{u}_{j-1,J-j+1}, \tag{2}$$

where $\mathbf{u}_{j_1,j_2}$ is the solution of

$$\mathbf{A}_{j_1} \mathbf{u}_{j_1,j_2} \mathbf{A}_{j_2}^\top = \mathbf{f}_{j_1,j_2}. \tag{3}$$

Since the operator under consideration is of tensor product structure, the sparse grid combination technique coincides with the Galerkin solution in the sparse grid ansatz space.

While the finite element system matrices $\mathbf{A}_{j_1}$ and $\mathbf{A}_{j_2}$ can be computed exactly, it is not obvious how to compute the right-hand side with sufficient (i.e., finest level) accuracy. To this end, we shall exploit the sparse grid interpolant of the right-hand side.

Computing the right-hand side

The sparse grid interpolant of $\mathrm{Cov}[u]$ on level J is given by

$$\widehat{\mathrm{Cov}[f]}_J(\mathbf{x},\mathbf{y}) := \sum_{j_1+j_2 \le J} \sum_{k_1 \in Y_{j_1}} \sum_{k_2 \in Y_{j_2}} c_{(j_1,j_2),(k_1,k_2)} \phi_{j_1,k_2}(\mathbf{x}) \otimes \phi_{j_2,k_2}(\mathbf{y}),$$

where $Y_j := \{k : \mathbf{x}_{j,k} \notin X_{j-1}\}$ and X_j is a set or relevant triangulation points on the level j. We can store its coefficients in the sparse block matrix $\widehat{\mathbf{C}}_J := [\mathbf{c}_{j_1,j_2}]_{j_1+j_2 \le J}$ where $\mathbf{c}_{j_1,j_2} = [c_{k_1,k_2}]_{k_1,k_2 \in Y_{j_1},Y_{j_2}}$. To obtain from this interpolant the right-hand sides required in the sparse grid combination technique, we have to compute expressions of the form

$$\mathbf{f}_{j_1,j_2} = \left[\langle \widehat{\mathrm{Cov}[f]}_J, \phi_{j_1,k_2} \otimes \phi_{j_1,k_2} \rangle\right]_{k_1,k_2}.$$

which involve rectangular finite element mass matrices. We can avoid such matrices by exploiting restrictions and prolongations. Moreover, the key for an efficient computation is the use of the identity

$$\mathrm{vec}(\mathbf{Y}) = (\mathbf{A} \otimes \mathbf{B})\,\mathrm{vec}(\mathbf{X}) \iff \mathbf{Y} = \mathbf{B}\mathbf{X}\mathbf{A}^\top$$

together with the observation that the order of execution matters in case of rectangular matrices. A detailed analysis shows that it is most efficient to proceed as follows:

When $j_1 + j_2' \le j_1' + j_2$, we compute

$$\mathbf{y} := \begin{cases} \mathbf{R}_{j_1'}^{j_1}\, \mathbf{M}_{j_1'}\, \widetilde{\mathbf{c}}_{j_1',j_2'}, & j_1 \le j_1', \\ \mathbf{M}_{j_1} \mathbf{P}_{j_1'}^{j_1}\, \widetilde{\mathbf{c}}_{j_1',j_2'}, & j_1 > j_1', \end{cases}$$

$$\mathbf{z} := \begin{cases} \mathbf{R}_{j_2'}^{j_2}\, \mathbf{M}_{j_2'}\, \mathbf{y}^\top, & j_2 \le j_2', \\ \mathbf{M}_{j_2}\, \mathbf{P}_{j_2'}^{j_2}\, \mathbf{y}^\top, & j_2 > j_2'. \end{cases} \tag{4}$$

Otherwise, we compute

$$\mathbf{y} := \begin{cases} \mathbf{R}_{j_2'}^{j_2}\, \mathbf{M}_{j_2'}\, \widetilde{\mathbf{c}}_{j_1',j_2'}^\top, & j_2 \le j_2', \\ \mathbf{M}_{j_2}\, \mathbf{P}_{j_2'}^{j_2}\, \widetilde{\mathbf{c}}_{j_1',j_2'}^\top, & j_2 > j_2', \end{cases}$$

$$\mathbf{z}^\top := \begin{cases} \mathbf{R}_{j_1'}^{j_1}\, \mathbf{M}_{j_1'}\, \mathbf{y}^\top, & j_1 \le j_1', \\ \mathbf{M}_{j_1}\, \mathbf{P}_{j_1'}^{j_1}\, \mathbf{y}^\top, & j_1 > j_1'. \end{cases} \tag{5}$$

Herein, $\mathbf{R}_{j'}^{j}$ is the restriction matrix and $\mathbf{P}_{j'}^{j}$ is the prolongation matrix for the levels j, j' while the matrices $\widetilde{\mathbf{c}}_{j_1,j_2}$ are blocks $\mathbf{c}_{j_1,j_2}$ padded to the corresponding levels.

With these block operations, we can compute all right-hand sides $\widehat{\mathbf{F}}_J = [\mathbf{f}_{j_1,j_2}]_{j_1+j_2\le J}$ required in the sparse grid combination technique in one run by using the following algorithm in $O(N_J \log^3 N_J)$ operations.

Algorithm Sparse grid matrix-vector multiplication

Input: matrices $\mathbf{M}_j$, $1 \le j \le J$ and and blockwise matrix $\widehat{\mathbf{C}}_J = [\mathbf{c}_{j_1,j_2}]_{j_1+j_2\le J}$.
Output: blockwise matrix $\widehat{\mathbf{F}}_J = [\mathbf{f}_{j_1,j_2}]_{j_1+j_2\le J}$.

```
for j1 + j2 ≤ J do
    set f_{j1,j2} := 0
    for j1' + j2' ≤ J do
        if j1 + j2' ≤ j1' + j2
            compute z by (4)
        else
            compute z by (5)
        end
        update f_{j1,j2} = f_{j1,j2} + z^⊤
    end
end
```

Exercise 1.

Implement a Matlab function

```
function fJ = sparse_tensor_product(mgmesh, mass, sparse_interpol, J),
```

which calculates $\widehat{\mathbf{F}}_J$ according to the above algorithm. Here, the sparse grid interpolation

```
sparse_interpol = compute_sparse_interpolation(kf, mgmesh, J)
```

and the finite element mass matrix

```
mass = mgsys_assemble(mgmesh).M
```

enter. The mesh `mgmes` is same as in the Programming sheet 2.

Exercise 2.

Implement a Matlab function

```
function uj1j2 = bivariate_backslash_solver(mgmesh, stiff, fJ, j1, j2)
```

that solves (3) by the backslash operation. Here, `fJ` is obtained by `sparse_tensor_product` and

```
stiff = mgsys_assemble(mgmesh).A.
```

Exercise 3.

Implement a Matlab function

```
function uJ = combi_technique(mgmesh, stiff, fJ, J),
```

which computes (2). Use `bivariate_backslash_solver` only for the levels that are needed.

Exercise 4.

To test your implementation compare it with low-rank approximation from the first exercise sheet. The L^2-error should be around zero. Consider the Gaussian covariance

$$\mathrm{Cov}_G[f](\mathbf{x},\mathbf{y}) = \mathrm{e}^{-\|\mathbf{x}-\mathbf{y}\|_2^2}.$$

Also consider the Matérn covariance function with $\nu = 1/2$ given by

$$\mathrm{Cov}_M[f](\mathbf{x},\mathbf{y}) = \mathrm{e}^{-\|\mathbf{x}-\mathbf{y}\|_2}.$$

What do you observe? In which cases do you think which method is better?