Technische Universität Berlin Winter term 2025/26
Institute of Mathematics
Prof. Dr. T. Breiten, Prof. Dr. S. May, Dr. Juliette Dubois

Numerical Mathematics II for Engineers
Tutorial 8

Topics : Matrix assembly, 1D FEM, degrees of freedom, analysis of FEM.

Discussion in the tutorials of the week 15 – 19 December 2025

Disclaimers:

- This homework is rather long, and the second exercise builds on the results of the first exercise. Don't hesitate to ask us questions!
- Exercises should be solved in **fixed groups of 3 students**.
- Hand in the solution in **one folder** labeled **hw[hw_number]_group[group_number]** and containing:
  - **One pdf** for the theoretical questions and comments on the numerical results,
  - **One python file per programming exercise**.
  - Write the group number and all names of your members **in each file**.

Exercise 8.1: Matrix assembly for 1D FEM

In this exercise, we introduce a new way to compute the Galerkin matrix. Computing integrals can become costly for very large matrices, hence the idea is to compute the integrals on a so-called *reference element* and to apply substitution for deducing the integral on each element. The computations are done in this exercise, and the corresponding implementation is done in the next exercise.

We consider the equation

$$-u''(x) + u(x) = f \quad \text{in } \Omega = (x_L, x_R) \subset \mathbb{R}, \tag{1}$$

$$u(x_L) = u(x_R) = 0. \tag{2}$$

The weak formulation corresponding to (1)-(2) reads:

Find $u \in V := H_0^1(\Omega)$ such that $a(u, v) = F(v)$ holds for all $v \in V$.

Here, the symmetric bilinear form $a\colon V \times V \to \mathbb{R}$ and the linear form $F\colon V \to \mathbb{R}$ are defined via

$$a(u, v) := \int_{x_L}^{x_R} u'(x)v'(x) + u(x)v(x)\,\mathrm{d}x, \quad F(v) := \int_{x_L}^{x_R} f(x)v(x)\,\mathrm{d}x.$$

We consider the general grid $x_L = x_0 < x_1 < \ldots < x_{N+1} = x_R$ with $N \in \mathbb{N}$, and define the subdomains $\Omega_i = (x_{i-1}, x_i)$ with length $h_i = x_i - x_{i-1}$. We use the linear finite elements in

1D introduced in Lecture 15. With $n = N$,the basis of the discrete space V_n is $(\overline{\phi}_1, ..., \overline{\phi}_n)$, with

$$\text{For } i \in \{1, \dots, n\}, \quad \overline{\phi}_i(x) = \begin{cases} \frac{x - x_{i-1}}{h_i}, & \text{if } x \in \overline{\Omega}_i, \\ \frac{x_{i+1} - x}{h_{i+1}}, & \text{if } x \in \overline{\Omega}_{i+1}, \\ 0, & \text{otherwise} \end{cases}$$

The corresponding entries of the Galerkin matrix $A_n \in \mathbb{R}^{n \times n}$ are given by

$$\begin{aligned} [A_n]_{ij} = a(\overline{\phi}_j, \overline{\phi}_i) &= \int_\Omega \overline{\phi}_i'(x)\overline{\phi}_j'(x) + \overline{\phi}_i(x)\overline{\phi}_j(x)\,\mathrm{d}x \\ &= \sum_{k=1}^{n+1} \int_{\Omega_k} \overline{\phi}_i'(x)\overline{\phi}_j'(x) + \overline{\phi}_i(x)\overline{\phi}_j(x)\,\mathrm{d}x. \end{aligned}$$

(a) For which combinations of $i, j \in \{1, \dots, n\}$ and $k \in \{1, \dots, n+1\}$ are the integrals

$$\int_{\Omega_k} \overline{\phi}_i'(x)\overline{\phi}_j'(x)\,\mathrm{d}x \quad \text{and} \quad \int_{\Omega_k} \overline{\phi}_i(x)\overline{\phi}_j(x)\,\mathrm{d}x$$

nonzero? Explain your answer.

(b) We introduce the *reference element* $\Omega_{\mathrm{ref}} = (0, 1)$. For $k = 1, \dots, n+1$, determine $a_k, b_k \in \mathbb{R}$ such that the mapping $F_k \colon \Omega_{\mathrm{ref}} \to \Omega_k$ defined via $F_k(x) := a_k x + b_k$ satisfies $F_k(0) = x_{k-1}$ and $F_k(1) = x_k$.

(c) For $i \in \{1, \dots, n\}$, we introduce the functions $\phi_1, \phi_2 \colon \Omega_{\mathrm{ref}} \to \mathbb{R}$ via $\phi_1 = \overline{\phi}_i \circ F_{i+1}$ and $\phi_2 = \overline{\phi}_i \circ F_i$. Derive explicit formulas for ϕ_1, ϕ_2 and verify that these two functions are independent of i.

(d) Explicitly compute the matrices $M \in \mathbb{R}^{2\times 2}$ and $S \in \mathbb{R}^{2\times 2}$ defined by

$$[M]_{ij} = \int_{\Omega_{\mathrm{ref}}} \phi_i(x)\phi_j(x)\mathrm{d}x \quad \text{and} \quad [S]_{ij} = \int_{\Omega_{\mathrm{ref}}} \phi_i'(x)\phi_j'(x)\mathrm{d}x.$$

respectively.

(e) Using the results of (c) and (d), compute the following integrals

$$\int_{\Omega_k} \bar{\phi}_i\bar{\phi}_j(x)\mathrm{d}x, \quad \int_{\Omega_k} \bar{\phi}_i'(x)\bar{\phi}_j'(x)\mathrm{d}x$$

for $i \in \{k-1, k\}$ and $j \in \{k-1, k\}$.

Hint: Use the fact that $\Omega_k = F_k(\Omega_{\mathrm{ref}})$ and integration by substitution.

(f) Combining the results of the previous questions, explain how you can compute the Galerkin matrix A_n from the matrices M_k, S_k and from the transformations $F_1, \dots F_{n+1}$.

Exercise 8.2: 1D FEM

The aim of this exercise is to implement the 1D FEM method computed in Exercise 8.1. We use the same PDE, weak formulation and finite elements as in the other exercise.

To implement the Galerkin matrix, we will use the following *matrix assembly algorithm* that relies on the computations of the other exercise:

1) Create the 2x2 matrices $M, S \in \mathbb{R}^{2\times 2}$ computing the integrals on the reference element Ω_{ref}.

2) Create an empty global matrix $A \in \mathbb{R}^{(n+2)\times(n+2)}$. For simplicity, we create the matrix including the boundary nodes, and we will remove them after the assembly.

3) Loop over the elements Ω_k. For each $k \in \{1, \dots n\}$:

   3.1) Compute the determinant of the Jacobian of the transformation F_k.

   3.2) Compute the local 2x2 matrices

$$M_k = \begin{pmatrix} \int_{\Omega_k} \bar{\phi}_{k-1}\bar{\phi}_{k-1}(x)\mathrm{d}x & \int_{\Omega_k} \bar{\phi}_{k-1}\bar{\phi}_k(x)\mathrm{d}x \\ \int_{\Omega_k} \bar{\phi}_{k-1}\bar{\phi}_k(x)\mathrm{d}x & \int_{\Omega_k} \bar{\phi}_k\bar{\phi}_k(x)\mathrm{d}x \end{pmatrix},$$
$$S_k = \begin{pmatrix} \int_{\Omega_k} \bar{\phi}'_{k-1}(x)\bar{\phi}'_{k-1}(x)\mathrm{d}x & \int_{\Omega_k} \bar{\phi}'_{k-1}(x)\bar{\phi}'_k(x)\mathrm{d}x \\ \int_{\Omega_k} \bar{\phi}'_k(x)\bar{\phi}'_{k-1}(x)\mathrm{d}x & \int_{\Omega_k} \bar{\phi}'_k(x)\bar{\phi}'_k(x)\mathrm{d}x \end{pmatrix}.$$

   using the reference matrix and the transformation (see the previous exercise).

   3.3) Add the results to the corresponding entries of the global matrix A.

Even if you haven't finished the exercise, you can do the implementation as we give you all the necessary formula.

The matrix assembly requires several functions defined in the separate file `femlib.py` for readability.

(a) In the file `femlib.py`, fill the functions `get_M_ref`, `get_S_ref`, `get_M_loc`, `get_S_loc`, `get_trafo`. We provide the function `get_elements` that returns the number of elements, the number of nodes, and a correspondence table `e`. This table give the correspondence between the elements and their corresponding nodes: the line `i` gives the indices of the two nodes of the `i`-th element.

   Hint: For 1D linear elements, there holds

$$M = \begin{pmatrix} 1/3 & 1/6 \\ 1/6 & 1/3 \end{pmatrix}, S = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix},$$
$$|F'_k| = h_k, \quad M_k = a_k M, \quad S_k = a_k^{-1} S,$$
$$A_{k,k} = [S_k]_{22} + [M_k]_{22} + [S_{k+1}]_{11} + [M_{k+1}]_{11}$$
$$A_{k,k+1} = [S_{k+1}]_{12} + [M_{k+1}]_{12}$$

(b) Fill the function `get_matrix_rhs` to assemble the matrix. The function should return the reduced matrix $\in \mathbb{R}^{n\times n}$, the reduced right-hand side $\in \mathbb{R}^n$ and the mesh $\in \mathbb{R}^{n+2}$. To fill the matrix and the right-hand side, use the functions defined in the previous section as described in the matrix assembly algorithm.

   You can use the same assembly algorithm for the right-hand side `b`. For a constant function f, the right-hand side is given by the formula $b_i = f|F'_k|/2 + f|F'_{k+1}|/2$.

   Hint: You can use full matrices for simplicity. Since you want to loop over the elements, your function should contain only one `for` loop.

(c) Solve for the case of a constant right-hand side $f = 1$ and with $x_L = 0, x_R = 1$. In this case, the exact solution is given by $u(x) = 1+\exp(1)-\exp(1-x)-\exp(x)/(1+\exp(1))$. Plot the approximate and the exact solutions for $n = 30$.

(d) Draw a convergence plot for 5 refinement levels.

Exercise 8.3: Degrees of freedom

We recall that the *degrees of freedom* of a FEM approximation is the dimension of the finite element space. This exercise will give you an idea of how the degrees of freedom change depending on the type of finite element and the PDE at hand.

We consider the Poisson equation with various boundary conditions and various finite elements on the unit square. Two meshes are given: a mesh with $n_e = 32$ triangular elements (see Figure 1a), and a mesh with $n_e = 16$ quadrilateral elements (see Figure 1b).

For each case, state which mesh should be used and give the number of degrees of freedom of the resulting FE approximation:

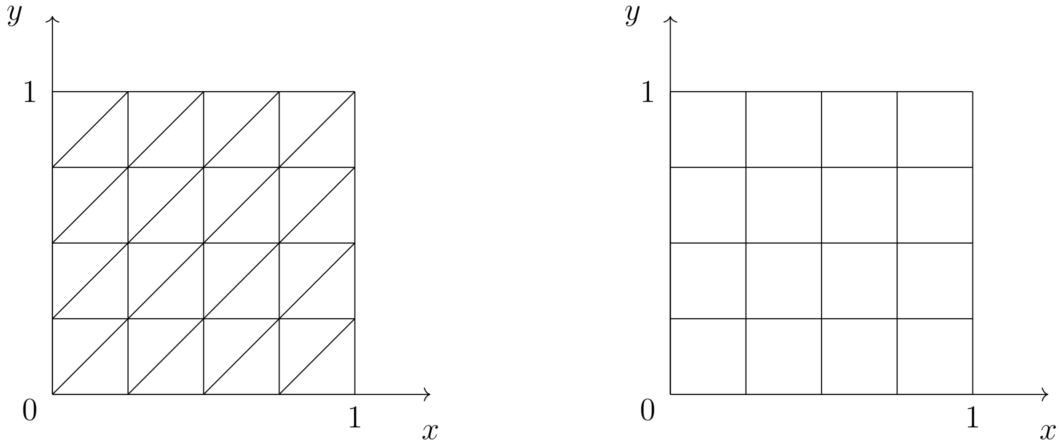
(a) Homogeneous Dirichlet boundary conditions using linear finite elements.

(b) Neumann boundary conditions using linear finite elements.

(c) Neumann boundary conditions using cubic finite elements.

(d) Homogeneous Dirichlet boundary conditions using biquadratic finite elements.

Hint: To count the degrees of freedom, drawing them on the meshes might help. We do not ask you for a general formula.



(a) Triangular mesh.

(b) Quadratic mesh.

Figure 1: The two considered meshes of the unit square.