Homework Assignment 5

Please submit the following files as indicated below: source code PDF file image file video file

Question 1 | **2 marks** | In this assignment, we consider the linear elasticity problem

$$\begin{aligned} -c\Delta u + au &= f \quad \text{in } \Omega \\ u &= g \quad \text{on } \partial\Omega \end{aligned} \tag{1}$$

on a polygonal domain Ω. The function u can be interpreted as the elongation of a rubber membrane over the x_1x_2-plane. The boundary values g prescribe the elongation on $\partial\Omega$, e.g. by means of a wire frame construction in which the membrane has been fixed. The real number $c > 0$ is the stiffness of the rubber material, $a > 0$ is a constant proportional to its mass density and the inhomogeneity f models external forces that act on the membrane.

(a) Show that under the assumption of homogeneous boundary conditions, $g = 0$, the discretisation of (1) with linear finite elements reads

$$(cK^h + aM^h)\vec{u}^h = \vec{f}^h$$

where

$$\begin{aligned} k_{ij}^h &= \int_\Omega \nabla\phi_i^h \cdot \nabla\phi_j^h \, \mathrm{d}x \\ m_{ij}^h &= \int_\Omega \phi_i^h \phi_j^h \, \mathrm{d}x \\ f_i^h &= \langle f, \phi_i^h \rangle_{H^{-1}(\Omega), H_0^1(\Omega)} \\ \phi_i^h &= \text{hat function centred at the } i\text{-th vertex} \end{aligned}$$

for $i, j = 1, \ldots, N$. N is the number of effective degrees of freedom, i.e. the number of interior grid points which are not located on the boundary $\partial\Omega$.

Note that since the domain is assumed to be a polygon, we can cover it exactly with a triangulation $\mathcal{T}^h$ such that $\Omega = \Omega^h$ (there is no mismatch on the boundary).

The weak formulation of this problem reads: find $u \in H_0^1(\Omega)$ such that for all test functions $v \in H_0^1(\Omega)$

$$c \int_\Omega \nabla u \cdot \nabla v \, \mathrm{d}x + a \int_\Omega uv \, \mathrm{d}x = \langle f, v \rangle_{H^{-1}, H_0^1}. \tag{W}$$

With the space of linear finite elements

$$V^h = \left\{ v^h \in C(\bar{\Omega}^h) \;\middle|\; v^h\big|_T \in P_1(T) \text{ for all } T \in \mathcal{T}^h, \, v^h\big|_{\partial\Omega^h} = 0 \right\}$$

the discrete problem (Galerkin equations) reads: find $u \in V^h$ such that for all test functions $v^h \in V^h$

$$c \int_\Omega \nabla u^h \cdot \nabla v^h \, \mathrm{d}x + a \int_\Omega u^h v^h \, \mathrm{d}x = \langle f, v^h \rangle_{H^{-1}, H_0^1}. \tag{G}$$

Denoting the hat-function basis of V^h by $(\phi_i^h)_{i=1,\ldots,N}$, where $\phi_i^h(p_j) = \delta_{ij}$ for all interior vertices $p_j \in \Omega^h$ ($i, j \in \{ 1, \ldots, N \}$), we write

$$u^h = \sum_{j=1}^N u_j^h \phi_j^h.$$

Since (G) is linear in v^h, it is sufficient to test with the hat functions ϕ_i^h ($i \in \{ 1, \dots, N \}$) only:

$$c \sum_{j=1}^{N} \left(\int_\Omega \nabla \phi_i^h \cdot \nabla \phi_j^h \, \mathrm{d}x \right) u_j^h + a \sum_{j=1}^{N} \left(\int_\Omega \phi_i^h \phi_j^h \, \mathrm{d}x \right) u_j^h = \langle f, \phi_i^h \rangle_{H^{-1}, H_0^1}.$$

This is the i-th row of the linear system

$$(cK^h + aM^h)\vec{u}^h = \vec{f}^h.$$

(b) Show that the matrix $cK^h + aM^h$ of this linear system is symmetric positive definite. Is M^h an M-matrix?

Set $L^h = cK^h + aM^h$.

- For all $i, j \in \{1, \dots, N\}$
$$l_{ij}^h = c \int_\Omega \nabla\phi_i^h \cdot \nabla\phi_j^h \,\mathrm{d}x + a \int_\Omega \phi_i^h \phi_j^h \,\mathrm{d}x = lji^h.$$
- Let $\vec{v}^h \in \mathbb{R}^N \setminus \{0\}$ be arbitrary and set $v^h = \sum_{j=1}^N v_j^h \phi_j^h$.
$$(\vec{v}^h)^\top L^h \vec{v}^h = \sum_{i,j=1}^N l_{ij}^h v_i^h v_j^h = c \int_\Omega |\nabla v^h|^2 \,\mathrm{d}x + a \int_\Omega |v^h|^2 \,\mathrm{d}x = c\|\nabla v^h\|_{L^2}^2 + a\|v^h\|_{L^2}^2 > 0.$$
- An M-matrix is a monotone Z-matrix, but M^h contains positive off-diagonal entries, so it is not a Z-matrix and hence not an M-matrix.

Question 2 | **3 marks** | Download the file `discretiseLinearElasticity.m`. We will turn this function into a finite-element solver for Problem (1) next week. Today we implement some core components.

The files `video.mat` and `kiwi.mat` contain arrays `P`, `E` and `T` which define a triangulation on a polygonal computational domain Ω^h. Note that some versions of MATLAB's plotting functions from the PDE Toolbox require extra rows in `E` and `T`. If you are not using the PDE Toolbox, then you may delete all but the first two rows of `E` and all but the first three rows of `T`, as described in the video on triangulations.

To import the variables from `video.mat` or `kiwi.mat` into a structure `msh`, you may use the `load` command. In Python, use `scipy.io.loadmat` and subtract 1 from all entries in `E` and `T` for Python's zero-based indexing.

(a) Recall that the k-th triangle in the triangulation has the vertices `T(1,k)`, `T(2,k)` and `T(3,k)` and you may look up their coordinates in the matrix `P`. For example, `P(:,T(1,k))` returns the two coordinates of the first vertex in the k-th triangle.

Complete the main function and the `assembleMass` subfunction. (Optional: Can you do it without `for` loops?)

Hint: In GNU Octave / MATLAB, the command `sparse` may be helpful. The corresponding command in Python is `scipy.sparse.csr_matrix`. Even though this is not documented, if multiple values have the same row and column indices, the `csr_matrix` command automatically sums up these values. In GNU Octave and MATLAB, this behaviour of the `sparse` command is documented.

(b) Write a script `hw5.m` to plot the triangular mesh and the sparsity pattern of the mass matrix that the function `discretiseLinearElasticity` returns (you don't have to remove the rows/columns corresponding to boundary points). Do this for both data sets `video.mat` and `kiwi.mat`. Make sure your plots are not distorted by using the `axis equal` command.

Hint: In installations of MATLAB with the PDE Toolbox, the command `pdemesh` may be helpful. In GNU Octave and MATLAB without the PDE Toolbox, the command `trimesh` may be helpful. In Python, use `plot_trisurf` with zero z-values. Note that `trimesh` and `plot_trisurf` need the transpose of `T`.

(c) Add extra commands to this script to plot your favourite function u^h on the kiwi domain and compute its L^2-norm. Constant functions are not allowed! Make sure your plots are not distorted.

Hint: The commands `pdeplot`, `trisurf` or `plot_trisurf`, respectively, may be helpful.

Your Learning Progress | **0 marks, but -1 mark if unanswered** | What is the one most important thing that you have learnt from this assignment?

What is the most substantial new insight that you have gained from this course this week? Any *aha moment*?
