MATH521: Numerical Analysis of Partial Differential Equations
Winter 2018/19, Term 2

Due Date: Thursday, 7 February 2019
Timm Treskatis

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).

First we must find the weak formulation of the linear elasticity problem:

$$\int_\Omega (-c\Delta u + au - f)v dx = 0$$

$$-c\int_\Omega \Delta u v \,\mathrm{d}x + a\int_\Omega uv \,\mathrm{d}x - \int_\Omega fv \,\mathrm{d}x = 0$$

$$\cancelto{0}{-c\int_{\partial\Omega} \frac{\partial u}{\partial n} v \,\mathrm{d}s} + c\int_\Omega \nabla u \cdot \nabla v \,\mathrm{d}x + a\int_\Omega uv \,\mathrm{d}x = \int_\Omega fv \,\mathrm{d}x$$

Which leaves the weak formulation, given $f \in H^{-1}(\Omega)$, find $u \in H_0^1(\Omega)$ such that all test functions $v \in H_0^1(\Omega)$

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

Now if we choose a basis $(\phi_i^h)_{i=1}^N$ of $V^h \subset V$ we can then define:

$$u^h = \sum_{j=1}^{N} u_j^h \phi_j^h \qquad v^h = \sum_{j=1}^{N} v^h(x^j)\phi_j^h$$

Subbing this into the weak formulation using the Galerkin approximation

$$\begin{aligned} B(u^h, v^h) &= \langle f, v^h \rangle_{V^*,V}, \qquad \forall v^h \in V^h \\ \sum_{j=1}^{N} B(\phi_j^h, \phi_i^h) u_j^h &= \langle f, \phi_j^h \rangle_{H^{-1}, H_0^1}, \qquad \forall i \in \{1, \dots, N\} \\ \sum_{j=1}^{N} \Big(c \underbrace{\int_\Omega \nabla \phi_j^h \cdot \nabla \phi_i^h \, \mathrm{d}x}_{k_{ij}^h} + a \underbrace{\int_\Omega \phi_j^h \phi_i^h \, \mathrm{d}x}_{m_{ij}^h} \Big) u_j^h &= \sum_{j=1}^{N} \underbrace{\int_\Omega f \cdot \phi_j^h \, \mathrm{d}x}_{f_j^h}, \qquad \forall i \in \{1, \dots, N\} \end{aligned}$$

Where,

$$\sum_{j=1}^{N} k_{ij}^h = K^h \qquad \sum_{j=1}^{N} m_{ij}^h = M^h \qquad \sum_{j=1}^{N} u_j^h = \vec{u}^h \qquad \sum_{j=1}^{N} f_j^h = \vec{f}^h$$

Leaving us with:

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

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

$cK^h + aM^h$ is positive definite if either K^h or M^h is positive definite and the other is at least positive semi-definite, that is using the definition $u^T Au > 0$ and the distributive properties of matrices. It is also true that the sum of positive semi-definite matrices results in a positive definite matrix if there remains no zeros on the diagonal. It can now be shown that both K^h and M^h are both positive definite matrices resulting from the sums of positive semi-definite matrices.

$$M^h = \sum_T M_T^h \quad \text{where} \quad M_T^h = \frac{|T|}{12} \underbrace{\begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}}_{\text{PD}} \rightarrow \underbrace{\begin{bmatrix} & & & & \\ & 2 & 1 & 1 & \\ & 1 & 2 & 1 & \\ & 1 & 1 & 2 & \\ & & & & \end{bmatrix}}_{\text{PSD}}$$

Each individual M_T matrix is positive definite and becomes positive semi-definite when mapped to the corresponding vertices in the full domain, then summing all of the M_T^h matrices fills in the diagonal leaving us with a positive definite matrix. The same applies to the K_T matrices except that each individual matrix is positive semi-definite, as it is defined in 2.3.9. Now that both M^h and K^h are positive definite matrices, multiplying them by positive real constants, a and c doesn't change that fact, and so the resulting sum is also positive definite.

M^h is <u>not</u> an M-matrix because all of the entries are positive. M-matrices require that the off-diagonal entries be negative.

Note: I apologise for the wordy answer, I wasn't sure how to prove this with mathy notation (if my explanation does in fact prove it at all)

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?

Matlab has a built-in function for pretty much everything, and they are very specific in how they are to be used.

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

That I very much prefer the applied stuff compared to the analytical or theoretical. Both are usually very confusing to me, and require a lot of extra time just convincing myself of the math.