TP 3 : Assembly algorithm using Python

Let Ω be an open bounded subset of $\mathbb{R}^2$ with polygonal boundaries. Consider the following problem with Neumann boundary conditions : Find $u \in H^1(\Omega)$ such that

$$\begin{cases} -\Delta u + u &= f \quad \text{in } \Omega, \\ \partial_n u &= 0 \quad \text{on } \partial\Omega \end{cases}$$

1. Write the continuous variational formulation of this problem.

Discrete problem We want to implement the discrete scheme using P_1 Lagrange Finite Element in Python. Let $\mathcal{T}_h$ be an admissible mesh of Ω and V_h the approximation space of $H^1(\Omega)$. We denote by $(T_\ell)_{\ell=1\ldots N_T}$ the triangles of this mesh $\mathcal{T}_h$, $(M_i)_{i=1\ldots N}$ the nodes of this mesh and $(\varphi_i)_{i=1\ldots N}$ th basis elements of V_h.

2. Write the discrete variational formulation solved by the discrete solution $u_h \in V_h$.
3. By construction, this solution is expressed as

$$u_h(x,y) = \sum_{i=1\ldots N} u_h(M_i)\varphi_i(x,y) \quad \forall (x,y) \in \overline{\Omega}.$$

   Express the discrete problem as an equivalent linear system

$$(\mathcal{A} + \mathcal{M})U = F,$$

   with $U = (u_h(M_i))_{i=1\ldots N} \in \mathbb{R}^N$ the unknown vector, $\mathcal{M}$ the mass matrix and $\mathcal{A}$ the stiffness matrix.

Assembly We want to solve this problem on a square. In the given code, we have already implement a structured mesh on this square. It only remains to implement the two matrices $\mathcal{M}$ and $\mathcal{A}$. To do so, we recall the assembly algorithm

$\mathcal{M} = 0$, $\mathcal{A} = 0$

For all $\ell = 1, N_T$

Get nodes coordinates of T_ℓ

Compute local matrices $\mathcal{A}^{T_\ell}$

For $i = 1, 3$

$I = \text{local} \to \text{global}(\ell, i)$

For $j = 1, 3$

$J = \text{local} \to \text{global}(\ell, j)$

$\mathcal{M}_{IJ} += \mathcal{M}_{ij}^{T_\ell},\ \mathcal{A}_{IJ} += \mathcal{A}_{ij}^{T_\ell}$

End j

End i

End ℓ

4. What is the name of the connectivity class in the code ? How do you obtain the coordinate of a node numbered i of your mesh ?
5. Complete the assembly algorithm in the code.

Right hand side It now remains to implement the right hand side F.

6. First, we assume that $f = 1$. Observing that $f \in V_h$, write the right hand side F using the mass matrix $\mathcal{M}$.
7. Recall the definition of F for a general data $f \in \mathcal{C}^0(\Omega)$. Approximating f by $\mathcal{I}_h^{(1)} f$ (its interpolation using P_1 Lagrange FE), give an approximation of the right hand side using the mass matrix. We will assume that replacing f by $\mathcal{I}_h^{(1)} f$ with the P_1-FE does not change the result.

Validation We want to verify that our code gives a "good" solution u_h. To do so, we solve our continuous problem with the solution $u(x, y) = \cos(\pi x)\cos(\pi y)$ for all $(x, y) \in \overline{\Omega}$ on the unit square.

8. Compute the corresponding right hand side f.
9. Compute the error term $||\mathcal{I}_h^{(1)}(u) - u_h||_{\mathbb{L}^2(\Omega)}$ using the mass matrix $\mathcal{M}$. Plot the quantity $\log(||\mathcal{I}_h^{(1)}(u) - u_h||_{\mathbb{L}^2(\Omega)}/||u||_{L^2(\Omega)})$ with respect to $\log 1/h$ with h the mesh size.
10. Compute the error term $||\nabla\mathcal{I}_h^{(1)}(u) - \nabla u_h||_{\mathbb{L}^2(\Omega)}$ using the stiffness matrix $\mathcal{A}$. Plot the quantity $\log(||\nabla\mathcal{I}_h^{(1)}(u) - \nabla u_h||_{\mathbb{L}^2(\Omega)}/||\nabla u||_{L^2(\Omega)})$ with respect to $\log 1/h$.