Numerical Methods for Partial Differential Equations
A.Y. 2022/2023

Laboratory 04

Finite Element method for the diffusion-reaction equation in 2D: convergence analysis

Exercise 1.

Let $\Omega = (0,1) \times (0,1)$, and let us consider the following Poisson problem with homogeneous Dirichlet boundary conditions:

$$\begin{cases} -\nabla \cdot (\mu\, \nabla u) + \sigma u = f & \mathbf{x} \in \Omega, \quad (1a) \\ u = 0 & \text{on } \partial\Omega, \quad (1b) \end{cases}$$

where $\mathbf{x} = (x,y)^T$, $\mu(\mathbf{x}) = 1$, $\sigma = 1$ and

$$f(\mathbf{x}) = (20\pi^2 + 1)\sin(2\pi x)\sin(4\pi y)\ .$$

The exact solution to this problem is

$$u_{\text{ex}}(x,y) = \sin(2\pi x)\sin(4\pi y)\ .$$

1.1. Write the weak formulation, the Galerkin formulation and the finite element formulation of (1).

Solution. We proceed as usual: let $V = H^1_0(\Omega)$ and $v \in V$. We multiply v to (1a) and integrate over Ω:

$$\int_\Omega -\nabla \cdot (\mu\, \nabla u)\, v d\mathbf{x} + \int_\Omega \sigma u v d\mathbf{x} = \int_\Omega f v\ ,$$

$$\int_\Omega \mu\, \nabla u \cdot \nabla v d\mathbf{x} - \underbrace{\int_{\partial\Omega} \mu(\nabla u \cdot \mathbf{n}) v d\gamma}_{=0} + \int_\Omega \sigma u v d\mathbf{x} = \int_\Omega f v d\mathbf{x}\ .$$

Let us define

$$a(u,v) = \int_\Omega \mu \nabla u \cdot \nabla v d\mathbf{x} + \int_\Omega \sigma u v d\mathbf{x}\ , F(v) \qquad = \int_\Omega f v d\mathbf{x}\ .$$

The weak formulation reads:

$$\text{find } u \in V \quad : \quad a(u, v) = F(v) \quad \forall v \in V \ . \tag{2}$$

Let us introduce a mesh in Ω, and let $V_h = X_h^r \cap V$ be the finite element space. Let φ_i be its basis functions, for $i = 1, 2, \dots, N_h$. After restricting (2) to V_h and writing the solution as a linear combination of the basis, we obtain

$$\sum_{j=1}^{N_h} U_j \left(\int_\Omega \mu \boldsymbol{\nabla} \varphi_i \cdot \boldsymbol{\nabla} \varphi_j d\mathbf{x} + \int_\Omega \sigma \varphi_i \varphi_j d\mathbf{x} \right) = F(\varphi_i) \qquad i = 1, 2, \dots, N_h \ ,$$

which can be written as a linear system:

$$\begin{gathered} A\mathbf{u} = \mathbf{f} \ , \\ A_{ij} = \int_\Omega \mu \boldsymbol{\nabla} \varphi_i \cdot \boldsymbol{\nabla} \varphi_j d\mathbf{x} + \int_\Omega \sigma \varphi_i \varphi_j d\mathbf{x} \ , \\ \mathbf{f}_i = F(\varphi_i) \ . \end{gathered}$$

1.2. Starting from the code of Laboratory 3, implement a finite element solver for problem (1). The solver should read the mesh from file (four differently refined meshes are provided as `mesh/mesh-square-*.msh`).

Solution. See the file `src/lab-04.cpp` for the implementation. An overview of the differences with respect to previous laboratory is provided below.

The major change is the introduction of the reaction term. This requires to change the assembly, adding to the local matrix the contribution of

$$\int_\Omega \sigma \varphi_i \varphi_j d\mathbf{x} \ .$$

Then, we have to import the mesh from file: see the method `Poisson2D::setup` for the implementation. The boundary of the imported mesh is labelled following the same convention as done by `deal.II` (and as seen in previous laboratory).

Finally, we need to impose the appropriate boundary conditions (homogeneous Dirichlet, in this case), and change the definition of the forcing term.

The numerical solution is shown in Figure 1.

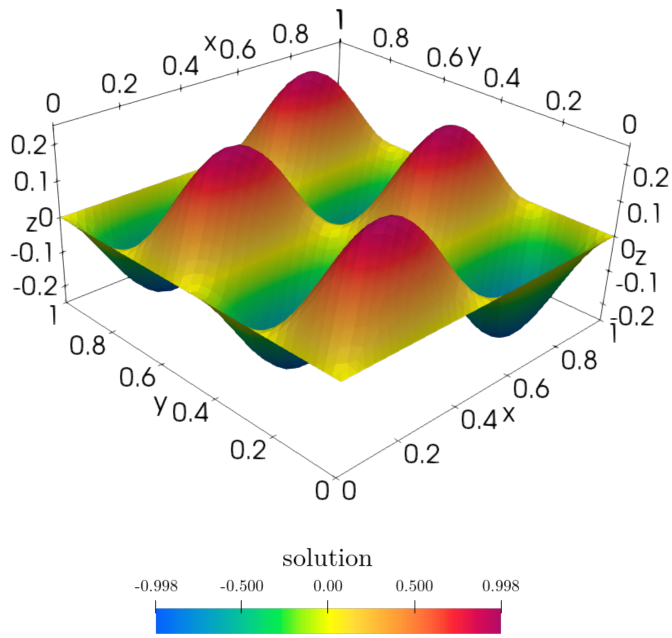


Figure 1: Numerical solution to (1), computed using linear finite elements on mesh `mesh-square-40.msh`.

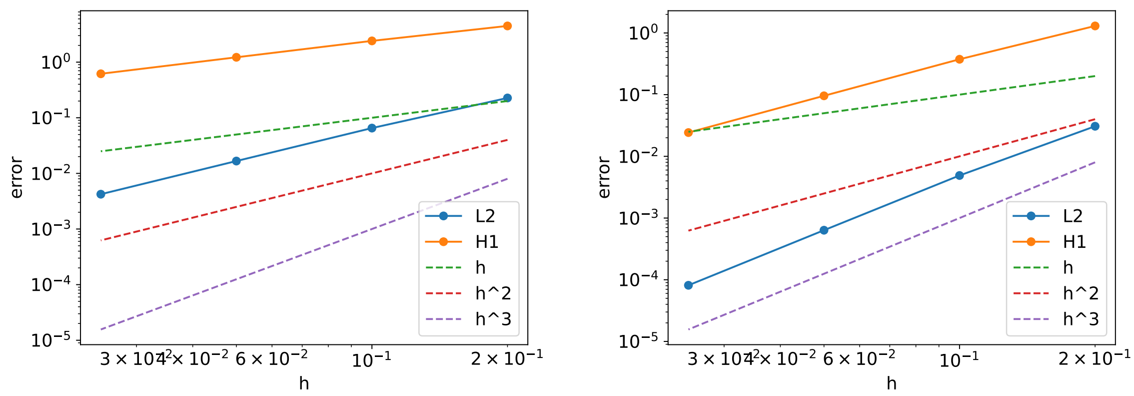
1.3. Using the four meshes provided, study the convergence of the solver for polynomials of degree $r = 1$ and of degree $r = 2$. Plot the error in the L^2 and H^1 norms against h, knowing that for every mesh file `mesh/mesh-square-`N`.msh`, the mesh size equals $h = 1/N$.

Solution. See the file `src/lab-04.cpp` for the implementation.

To study the convergence, we need to define a class `ExactSolution` to represent u_{ex}. Moreover, we need to define a method `Poisson2D::compute_error` to compute the error. The latter has the same implementation as seen in Laboratory 2, with two significant differences:

- the quadrature formula is defined using the class `QGaussSimplex<dim>`, so that it works for triangles;
- we need to provide an extra argument to the function `integrate_difference` that explicitly describes the mapping ϕ_c from the reference element to the physical elements. This mapping is constructed using the class `MappingFE`.

Then, the `main` function is modified to perform the convergence analysis and produce a CSV file `convergence.csv`, whose content can then be plotted to analyze the convergence results. See Figure 2 for the result. For both $r = 1$ and $r = 2$, we observe the expected convergence orders in both the L^2 and H^1 norms.



(a) (b)

Figure 2: Error in the L^2 and H^1 norms as a function of h for $r = 1$ (a) and $r = 2$ (b).

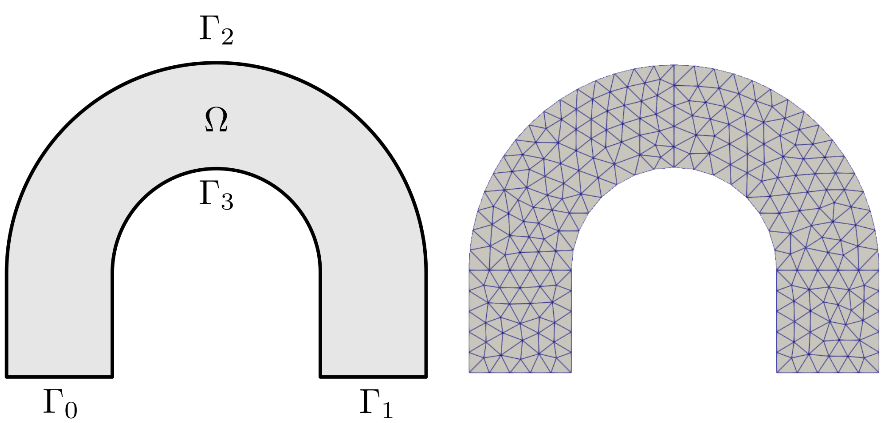


Figure 3: Domain for Exercise 2 (left), and a triangular mesh over it (right), corresponding to the file `mesh/mesh-u-5.msh`.

Exercise 2.

Let Ω be the domain depicted in Figure 3, contained in the files `mesh/mesh-u-*.msh`. The boundaries of the mesh are labelled as shown in Figure 3 (i.e. Γ_0 is labelled 0, Γ_1 is labelled 1, and so on). Consider the problem:

$$\begin{cases} -\boldsymbol{\nabla} \cdot (\mu \boldsymbol{\nabla} u) = 0 & \text{in } \Omega\,, \\ u = 0 & \text{on } \Gamma_0\,, \\ u = 1 & \text{on } \Gamma_1\,, \\ \mu \boldsymbol{\nabla} u \cdot \mathbf{n} = 0 & \text{on } \Gamma_2 \cup \Gamma_3\,. \end{cases}$$

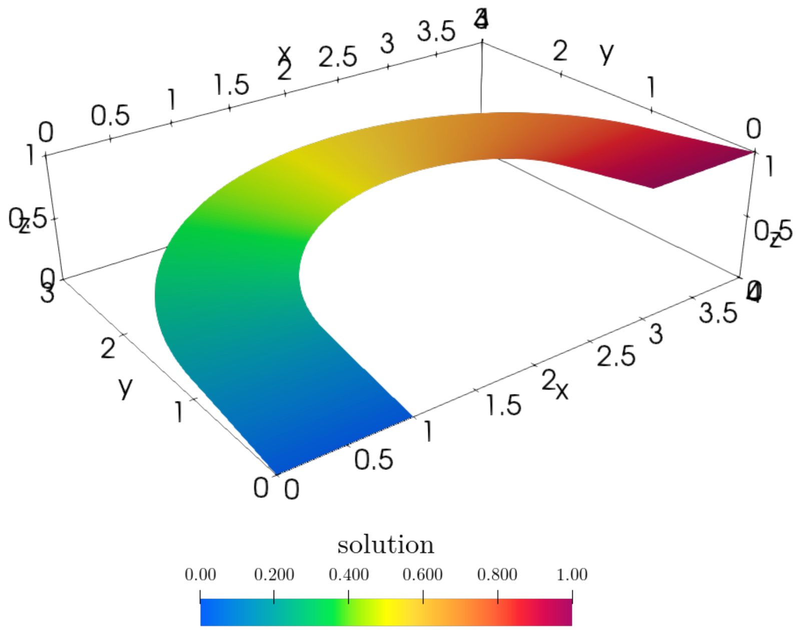


Figure 4: Numerical solution to Exercise 2, computed using linear finite elements and the mesh `mesh/mesh-u-40.msh`.

2.1. Starting from the previously implemented code, solve (2) using linear finite elements.

Solution. See the file `src/lab-04.cpp` for the implementation. The resulting solution is depicted in Figure 4.