Finite Element for Poisson Problems

Edgar Neboit
Titouan Steyer

2025

Contents

1 Objectives

The objectives of this work are:

1. Implement a finite element $\mathbb{P}_1$ method for a Poisson problem in a bounded open space.
2. Highlight the convergence order of the method.
3. Show the impacts of the mesh (regular or random) on the method.

2 Introduction of the problem

2.1 Discretisation of the problem

The problem is this Poisson problem with Neumann boundary conditions

$$\begin{cases} -\Delta u + u = f, \text{ in } \Omega \\ \partial_n u = 0, \text{ in } \partial\Omega \end{cases} \tag{2.1}$$

With $\Omega \subset \mathbb{R}^2$ open bounded set with polygonal boundaries. The weak formulation of (2.1) is

$$\begin{cases} \text{Find } u \in H^1(\Omega), \text{ such that} \\ \forall v \in H^1(\Omega), \ \int_\Omega \nabla u \cdot \nabla v + \int_\Omega uv = \int_\Omega fv. \end{cases} \tag{2.2}$$

Let $V_h \subset H^1(\Omega)$ a subset of $H^1(\Omega)$ of dimension N and denote $(\varphi_i)_{i=1...N}$ a basis of V_h. Then the Galerkin discrete formulation of (2.2) is

$$\begin{cases} \text{Find } u_h \in V_h, \text{ such that} \\ \forall v_h \in V_h, \ \int_\Omega \nabla u_h \cdot \nabla v_h + \int_\Omega u_h v_h = \int_\Omega f v_h. \end{cases} \tag{2.3}$$

The decomposition of u_h in the basis $(\varphi_i)_{i=1...N}$ is written as

$$\forall (x,y) \in \overline{\Omega}, \ u_h(x,y) = \sum_{i=1}^{N} U_i^h \varphi_i(x,y). \tag{2.4}$$

The discrete problem (2.3) is equivalent to the following linear system

$$(\mathcal{A} + \mathcal{M}) U_h = F. \tag{2.5}$$

With $U_h = (U_i^h)_{i=1...N}$ the coordinates of u_h in $(\varphi_i)_{i=1...N}$, $\mathcal{M} = (\mathcal{M}_{ij})_{i,j=1...N} \in \mathcal{M}_N(\mathbb{R})$ such that $M_{ij} = \int_\Omega \varphi_i \varphi_j$ (the mass matrix). $\mathcal{A} = (\mathcal{A}_{ij})_{i,j=1...N} \in \mathcal{M}_N(\mathbb{R})$ such that $\mathcal{A}_{ij} = \int_\Omega \nabla\varphi_i \cdot \nabla\varphi_j$ (the stiffness matrix).
Denote $\mathcal{T} = (\mathcal{T})_{h\geq 0}$ a sequence of admissible meshes of Ω such that $\mathcal{T}_h = (T_i)_{i=1...N_T}$ is composed of N_T triangles. Let us denote $(M_i)_{i=1...N}$ the nodes.
It remained the choice of V_h and his basis.

2.2 Lagrangian finite element of order 1

The finite element $\mathbb{P}_1$ is the space

$$\mathbb{P}_1 = \{p : (x,y) \in \mathbb{R}^2 \to p(x,y) \in \mathbb{R} \mid \forall (x,y) \in \mathbb{R}^2 \ p(x,y) = ax + by + c, \ (a,b,c) \in \mathbb{R}^3\}. \tag{2.6}$$

Then, the approximation space is, in this case, V_h^1 such that

$$V_h^1 = \{v \in \mathcal{C}^0(\Omega) \mid \forall l = 1...N_T, \ v_{h|T_l} \in \mathbb{P}_1\}. \tag{2.7}$$

With this choice, the basis functions are explicitly expressed

$$\varphi_M(x) = \begin{cases} \lambda_{T,M}(x), \ x \in T, \ T \in \mathcal{T}(M) \\ 0, \text{ elsewhere} \end{cases} \tag{2.8}$$

With $\mathcal{T}(M)$ the set of triangles which share an edge with T and $\lambda_{T,M}$ the barycentric coordinates of M in the triangle T. Then $(\varphi_{M_i})_{i=1\dots N}$ is a basis of V_h^1. The goal is to implement this method and to verify the convergence order of the $\mathbb{P}_1$ which is of order one.
We can express the barycentric coordinates by

Théorème 2.1 - Barycentric coordinates.
Let $T \in \mathcal{T}_h$ a triangle on our mesh and denotes $M_1(x_1, y_1), M_2(x_2, y_2), M_3(x_3, y_3)$ the three corners of our triangle then,

$$\begin{cases} \lambda_{M_1,T}(x,y) = \dfrac{1}{2|T|}\left(y_{23}(x-x_3) - x_{23}(y-y_3)\right), \\ \lambda_{M_2,T}(x,y) = \dfrac{1}{2|T|}\left(y_{31}(x-x_1) - x_{31}(y-y_1)\right), \\ \lambda_{M_3,T}(x,y) = \dfrac{1}{2|T|}\left(y_{12}(x-x_2) - x_{12}(y-y_2)\right), \end{cases} \tag{2.9}$$

with $y_{ij} = y_j - y_i$ the increment and $2|T| = |x_{23}y_{31} - x_{31}y_{23}|$ the area of the triangle.

Let denote $\mathcal{I}_h^1$ the interpolation operator of the space V_h^1 such that

$$\mathcal{I}_h^1 : v \in \mathcal{C}^0(\Omega) \to \sum_{i=1}^{N} v(M_i)\varphi_{M_i}. \tag{2.10}$$

One goal is to highlight the following result

Proposition 2.1 - One order convergence.

$$\forall v \in H^2(\Omega) \cap H^1(\Omega),\ \|\mathcal{I}_h^1(v) - v\|_{L^2(\Omega)} + h|\mathcal{I}_h^1(v) - v|_{H^1(\Omega)} \le ch^2|v|_{H^2(\Omega)},\ c > 0. \tag{2.11}$$

In particular, the L^2 norm decreases with a h^2 factor

$$\forall v \in H^2(\Omega) \cap H^1(\Omega),\ \|\mathcal{I}_h^1(v) - v\|_{L^2(\Omega)} \le ch^2|v|_{H^2(\Omega)},\ c > 0. \tag{2.12}$$

3 Implementation of the method

3.1 Matrix construction

We have to implement these two matrices:

1. (**Stiffness matrix**): $\mathcal{A} \in \mathcal{M}_N(\mathbb{R})$ such that $\mathcal{A}_{i,j} = \int_\Omega \nabla\varphi_i \cdot \nabla\varphi_j$.
2. (**Mass matrix**): $\mathcal{M} \in \mathcal{M}_N(\mathbb{R})$ such that $\mathcal{M}_{i,j} = \int_\Omega \varphi_i\varphi_j$.

3.1.1 Stiffness matrix

To compute the stiffness matrix on a general mesh, we need to decompose the integral over Ω as the sum of the integral over each triangle of the mesh:

$$\mathcal{A}_{i,j} = \sum_{T\in\mathcal{T}_h} \int_T \nabla\varphi_i \cdot \nabla\varphi_j. \tag{3.1}$$

Since the support of each basis functions ϕ_i are different, we can rewrite the sum as

$$\mathcal{A}_{i,j} = \sum_{T\in\mathcal{T}(M_i)\cap\mathcal{T}(M_j)} \int_T \nabla\varphi_i \cdot \nabla\varphi_j. \tag{3.2}$$

At first, we compute $\mathcal{A}^T$ the elementary stiffness matrix on $T = [M_1, M_2, M_3]$, such that

$$\mathcal{A}_{i,j}^T = \int_T \nabla\varphi_i \cdot \nabla\varphi_j. \tag{3.3}$$

Then, by using a connectivity table: a table that links the triangle with their nodes. Let denoted this table by **connect**, then the table **connect** is a $N_T \times 3$ table as for every line of this table, there is 3 integer which represents the position of the nodes in the global table with all the nodes.

3.1.2 Mass matrix

The same approach can be used for the mass matrix; we start by calculating the local mass matrix on a triangle T such that

$$\mathcal{M}^T_{i,j} = \int_T \varphi_i \varphi_j. \tag{3.4}$$

And then, we assemble the global mass matrix with the same process.

3.2 Implementation of the right hand side

We now consider the construction of the discrete right-hand side vector $F \in \mathbb{R}^N$, with components:

$$F_i = \int_\Omega f(x,y)\,\varphi_i(x,y)\,dx, \quad \text{for } i = 1,\dots,N,$$

where $(\varphi_i)_{i=1}^N$ denotes the $\mathbb{P}_1$ basis functions of V_h.

Assume first that $f = 1$. Since f belongs to V_h (it can be written as a constant combination of the basis functions), we can use the mass matrix M to compute F. More precisely, as

$$f(x,y) = \sum_{j=1}^N \varphi_j(x,y),$$

we obtain:

$$F_i = \int_\Omega \varphi_i(x,y)\,dx = \sum_{j=1}^N \int_\Omega \varphi_i(x,y)\varphi_j(x,y)\,dx = (M \cdot \mathbf{1})_i,$$

where $\mathbf{1} \in \mathbb{R}^N$ is the vector of ones. Thus, the entire right-hand side vector satisfies:

$$\boxed{F = M \cdot \mathbf{1}}.$$

In the general case where $f \in C^0(\Omega)$, we approximate f by its $\mathbb{P}_1$ Lagrange interpolant:

$$I_h^{(1)} f(x,y) = \sum_{j=1}^N f(M_j)\,\varphi_j(x,y),$$

where $f(M_j)$ denotes the evaluation of f at node M_j. Then, by linearity, we approximate the right-hand side as:

$$F_i \approx \int_\Omega I_h^{(1)} f(x,y)\,\varphi_i(x,y)\,dx = \sum_{j=1}^N f(M_j) \int_\Omega \varphi_i(x,y)\varphi_j(x,y)\,dx = (M \cdot f_h)_i,$$

with $f_h = (f(M_1),\dots,f(M_N))^T$ the vector of nodal values of f. Hence, we write:

$$\boxed{F \approx M \cdot f_h}.$$

This formulation will be used for assembling the right-hand side in the implementation.

4 Numerical simulation

In order to verify the convergence order of our method, we will compute an example with an exact solution:

$$u : (x,y) \to \cos(\pi x)\cos(\pi y). \tag{4.1}$$

The right-hand side is

$$f : (x,y) \to (2\pi^2 + 1)\cos(\pi x)\cos(\pi y). \tag{4.2}$$

To study the convergence order of our method in this special case, we follow the following procedure:

1. We solve the problem numerically using the finite element method on a sequence of increasingly refined meshes, obtaining in each case an approximate solution u_h.

2. For each mesh, we compute the error between the interpolation of u and the numerical approximation u_h, typically in the L^2 norm:
$$\|I_h(u) - u_h\|_{L^2} = \left(\int_\Omega |I_h(u)(x) - u_h(x)|^2 \, dx \right)^{1/2}.$$
We use the following result
$$\|I_h(u) - u_h\|_{L^2} = \sqrt{(I_h(u) - u_h)^T M (I_h(u) - u_h)}.$$
3. We then plot the error $\|I_h(u) - u_h\|_{L^2}$ against the mesh size h on a logarithmic scale: that is, we plot $\log(\|I_h(u) - u_h\|_{L^2})$ versus $\log(h)$.
4. The slope of the resulting line on the log-log plot corresponds to the convergence order of the method.

If the method has convergence order p, we expect the error to decrease like a power of h:
$$\|u - u_h\|_{L^2} \propto h^p.$$
Thus, in a logarithmic scale, the curve should approximately be a straight line with slope p.

4.1 Mesh size versus number of elements

Figure 1 shows the evolution of the mesh size h with respect to the number of triangles in the mesh. As expected for structured Cartesian meshes, the mesh size decreases as the number of elements increases.

This plot highlights that, beyond a certain number of triangles, the reduction in mesh size becomes less significant. Therefore, excessively increasing the number of elements does not always bring a clear benefit and should be weighed against the associated computational cost.

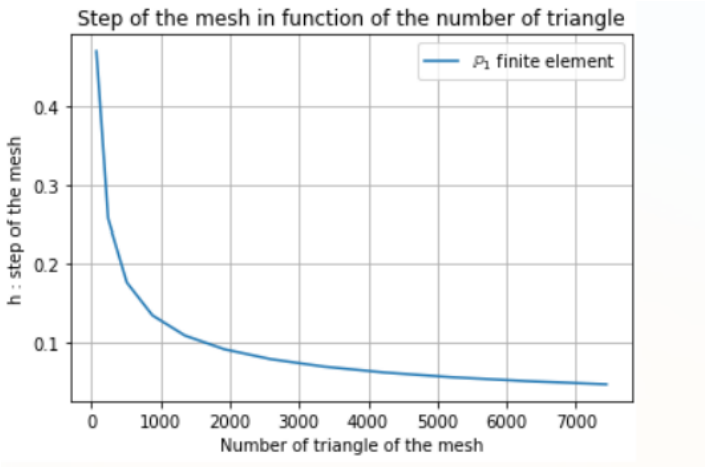


Figure 1: Mesh size h as a function of the number of triangles (structured mesh).

4.2 Validation of the Finite Element Method

4.2.1 Convergence in the L^2 norm

We compute the L^2 error between the interpolated exact solution $I_h(u)$ and the finite element solution u_h using the mass matrix M:
$$\|I_h(u) - u_h\|_{L^2}^2 \approx (I_h(u) - u_h)^T M (I_h(u) - u_h).$$

Figure 2 shows the L^2 error plotted against the mesh size h on a log-log scale. The observed slope is approximately 1.93, which confirms the expected convergence rate of order 2 in the L^2 norm for $\mathbb{P}_1$ finite elements.

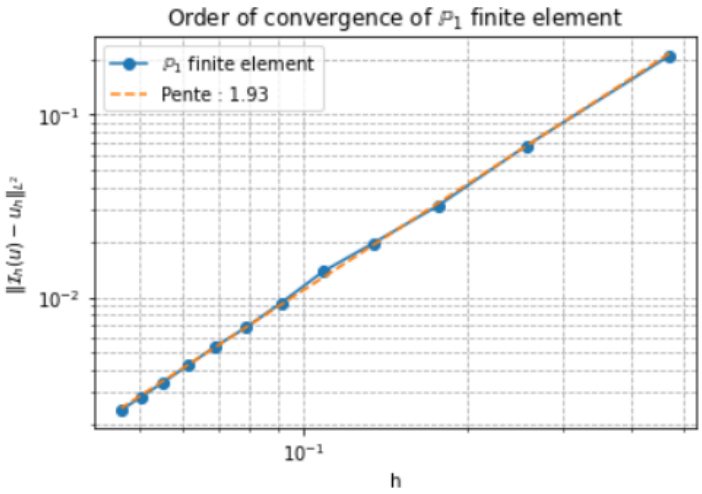


Figure 2: Convergence of the L^2 error with mesh refinement (structured mesh).

For comparison, we also plotted the L^2 error using an unstructured (irregular) mesh. The resulting slope is lower (approximately 1.35), which illustrates the impact of mesh quality on the convergence rate. This behavior is expected, since irregular meshes typically lead to reduced accuracy if not carefully controlled.

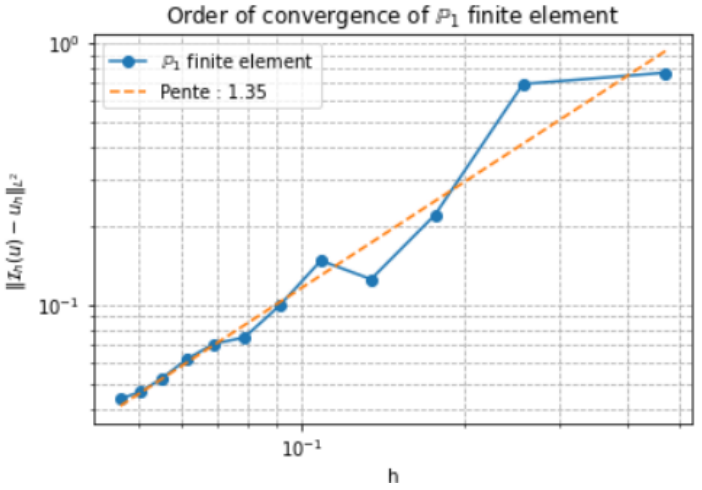


Figure 3: L^2 error convergence using an irregular mesh.

4.2.2 Convergence in the H^1 norm

We also compute the H^1 error, measuring the difference between the gradients of the interpolated exact solution and the finite element solution. Using the stiffness matrix R, we have:

$$\|\nabla I_h(u) - \nabla u_h\|_{L^2}^2 \approx (I_h(u) - u_h)^T R(I_h(u) - u_h).$$

The results are presented in Figure 4, where the error is again plotted versus the mesh size h in logarithmic scale. The estimated slope is close to 1.01, which matches the theoretical convergence rate of order 1 in the H^1 norm.

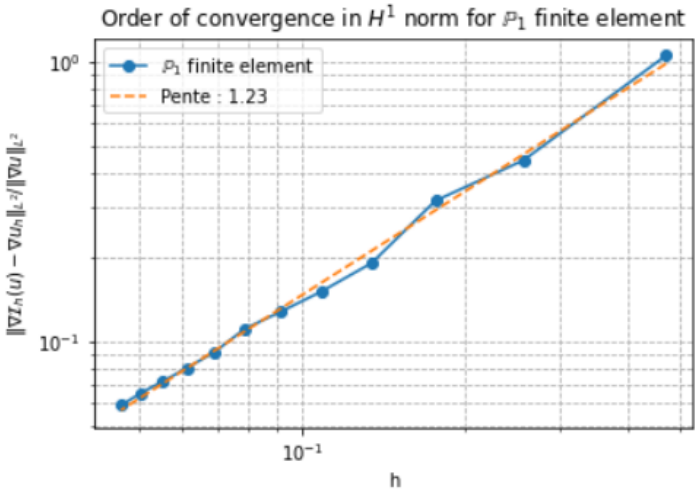


Figure 4: Convergence of the H^1 error with mesh refinement (structured mesh).

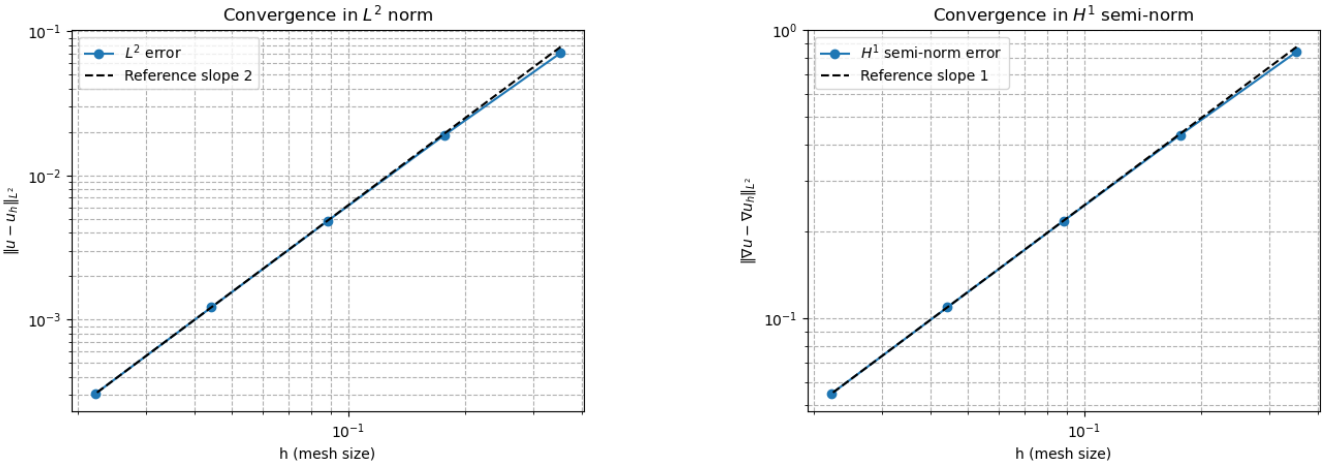
4.3 Comparison with Fenics

We want to compare with the implemented solver, Fenics, because it's a very efficient library with trust in the results. We follow the same procedure as before, we take the same exact solution u with the same right-hand side f.
As

$$u : (x, y) \to \cos(\pi x)\cos(\pi y). \tag{4.3}$$

The right-hand side is

$$f : (x, y) \to (2\pi^2 + 1)\cos(\pi x)\cos(\pi y). \tag{4.4}$$



(a) L^2-norm (b) H^1-norm

Figure 5: Strong order convergence from Fenics

We can see that with Fenics the convergence order is far more precise and easy to observe.

5 Conclusion

In this work, we successfully implemented a finite element $\mathbb{P}_1$ method to solve a Poisson problem with Neumann boundary conditions. The primary objectives were to demonstrate the convergence order of the method and to analyze the impact of different mesh types on the solution accuracy.

Through numerical simulations, we verified that the $\mathbb{P}_1$ finite element method achieves the expected convergence rates: second-order convergence in the L^2 norm and first-order convergence in the H^1 norm. These results were consistent with

theoretical predictions and were validated using both structured and unstructured meshes. The comparison with the Fenics solver further confirmed the reliability and accuracy of our implementation.

The study also highlighted the importance of mesh quality in achieving optimal convergence rates. While structured meshes provided more predictable and efficient convergence, unstructured meshes, although more flexible, required careful consideration to avoid reduced accuracy.

Overall, this work provides a solid foundation for further exploration of finite element methods, including the application to higher-order schemes and more complex boundary conditions. Future improvements could involve extending the method to three-dimensional problems and incorporating adaptive mesh refinement techniques to enhance computational efficiency and accuracy.