Numerical Partial Differential Equation
Peking University
Fall 2020
Yongli Peng

Homework 11

1 Problem setting

In this experiment we want to solve the Poisson equation with mixed boundary condition using the finite element method. To be precise, the problem can be written as:

$$\begin{cases} -\Delta u = f, & \\ u = u_0, & x \in \partial\Omega_0, \\ \dfrac{\partial u}{\partial \nu} + \beta u = g, & x \in \partial\Omega_1, \end{cases}$$

where we consider simply the rectangle $\Omega = (0,1) \times (0,1)$ with $\partial\Omega_0 = \{0\} \times [0,1]$ and $\partial\Omega_1 = \partial\Omega - \partial\Omega_0$. Here we assume $f \in L^2(\Omega), g \in L^2(\partial\Omega_1), \beta \in C(\partial\Omega_1)$ and $b \geq \delta > 0$. We use the C^0 type (1) rectangles and type (1) triangles as finite elements.

The question requires us chose two different sets of (f, g) to solve the problem. We expect to choose one derived from the known real solution u^*, so we can use a priori error estimate to evalute the performance. Finally we will plot the $log - log$ figure to see its convergence rate.
To be precise we use first $u(x,y) = \frac{1}{4}[(x-\frac{1}{2})^2 + (y-\frac{1}{2})^2] - \frac{1}{8}$ as the real solution and derive $f(x,y) = -1, u_0(y) = \frac{1}{4}(y-1)y$. Moreover we chose $\beta = 4$ and obtain

$$g(x,y) = \begin{cases} (x-\dfrac{1}{2})^2, & x \in [0,1], y \in \{0,1\}, \\ (y-\dfrac{1}{2})^2, & x = 1, y \in [0,1]. \end{cases}$$

For the second setting we let consider a simper scenario where the real solution is $u(x,y) = x(1-x)y(1-y)$ so its value is zero on $\partial\Omega$. Then we can compute that $f(x,y) = 2x(1-x) + 2y(1-y), u_0(y) = 0$. We assume $\beta = 0$ with (here the choice of β does not matter since $u = 0$ on $\partial\Omega$)

$$g(x,y) = \begin{cases} -x(1-x), & x \in [0,1], y \in \{0,1\}, \\ -y(1-y), & x = 1, y \in [0,1]. \end{cases}$$

2 Variation problem and its discretization

To use the finite element method (FEM), we shall first formulate the variational form of the original problem as well as its discretization. First we choose $v \in C^\infty(\Omega)$ as the test

function and we get $\int_\Omega(-\Delta u)vdx = \int_\Omega fvdx$. Use the integration by parts and Green's formula we get

$$\begin{aligned}\int_\Omega(-\Delta u)vdx &= \int_\Omega \nabla\cdot(-v\nabla u)dx + \int_\Omega \nabla u\cdot\nabla v dx\\ &= \int_{\partial\Omega} -v\frac{\partial u}{\partial\nu}ds + \int_\Omega \nabla u\cdot\nabla v dx\\ &= \int_{\partial\Omega_0} -v\frac{\partial u}{\partial\nu}ds + \int_{\partial\Omega_1} v(\beta u - g)ds + \int_\Omega \nabla u\cdot\nabla v dx.\end{aligned}$$

To cope with the first term in the last equation we may expect $v = 0$ on $\partial\Omega_0$. So the test function space can be extended to $H^1(\Omega_1)\cap H_0^1(\Omega_0)$.
We can now define the bilinear form $a(u,v) = \int_{\partial\Omega_1}\beta uvds + \int_\Omega \nabla u\cdot\nabla v dx$ and the linear functional $f(v) = \int_\Omega fvdx + \int_{\partial\Omega_1} gvds$. The variational problem can be formualted as

$$\begin{aligned}&\text{Find } u\in V_1 = \{u\in H^1(\Omega) : u = u_0 \text{ on } \partial\Omega_0\}\\ &\text{s.t. } a(u,v) = f(v) \qquad \forall v\in V_2 = H^1(\Omega_1)\cap H_0^1(\Omega_0).\end{aligned}$$

For the convenience of the later analysis, we shall always transform the inhomongoeneous Dirichlet boundary condition into the homogenoeous one by setting $\tilde f(v) = f(v) - a(\tilde u_0, v)$, where $\tilde u_0|_{\partial\Omega_0} = u_0$ is its extension. Then the variational problem can be written as

$$\begin{aligned}&\text{Find } u\in H^1(\Omega_1)\cap H_0^1(\Omega_0)\\ &\text{s.t. } a(u,v) = \tilde f(v) \qquad \forall v\in H^1(\Omega_1)\cap H_0^1(\Omega_0).\end{aligned}$$

Then its discretization can be similarly formulated as (assume we have the FE partition $\mathfrak{T}_h(\Omega) = \{(K_i, P_{K_i}, \Sigma_{K_i})\}$ with A_i as the nodes):

$$\begin{aligned}&\text{Find } u\in V_h(0) = \{v\in C(\Omega) : v|_{K_i}\in P_{K_i}, v(A_i) = 0, \forall A_i\in\partial\Omega_0\}\\ &\text{s.t. } a(u,v) = \tilde f(v) \quad \forall v\in V_h(0) = \{v\in C(\Omega) : v|_{K_i}\in P_{K_i}, v(A_i) = 0, \forall A_i\in\partial\Omega_0\}.\end{aligned}$$

This subspace $V_h(0)\subset H^1(\Omega)\cap H_0^1(\Omega_0)$ and is still a linear space.
In previous exercises we have proved that this bilinear form $a(\cdot,\cdot)$ and the linear functional $f(\cdot)$ satisfies the Lax-Milgram conditions. Use the Lax-Milgram theorem for the transformed problem we can prove the weak solution to the original problem uniquely exists.

3 Construction of finite element spaces

We here explains how the grid is generated as well as the construction of finite elements $(K_i, P_{K_i}, \Sigma_{K_i})$. For simplicity we assume the elements form an affine family.

3.1 Type (1) rectangles

For type (1) rectangles we use the most simple grid. Uniformly divide the interval $[0,1]$ into N bins: $0 = x_0 < x_1 < \cdots < x_N = 1$ with $x_i = \frac{i}{N}$ and do the same things on y. Then we can give the grid $K_{ij} = (x_i, x_i + h)\times(y_j, y_j + h), (0\le i,j\le N-1)$. Use the rectangle $\hat K = (0,1)\times(0,1)$ as the reference element. The affine map $F_{ij} : \hat K\to K_{ij}$ is

$F_{ij}(\hat{x},\hat{y}) = (h\hat{x}+x_{i-1}, h\hat{y}+y_{j-1}) = \begin{bmatrix} h & 0 \\ 0 & h \end{bmatrix}\begin{bmatrix}\hat{x}\\ \hat{y}\end{bmatrix} + \begin{bmatrix} x_{i-1}\\ y_{j-1}\end{bmatrix}$. The following is a simple figure demonstrating the choice of our finite elements.

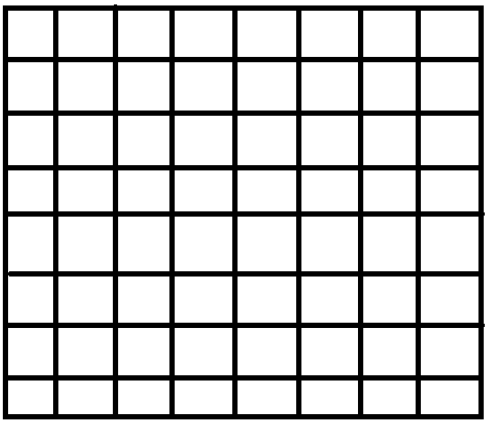

Figure 1: A simple figure illustrating our choice of mesh grids, where we divide the interval $[0,1]$ uniformly into 8 parts for both x-axis and y-axis and obtain 64 elements with 81 grid points.

For the reference element we construct the finite element space. Since $k = 1, n = 2$, we have $P_{\hat{K}} = Q_1(\hat{K}) = \{p(x) : p(x) = a_0 + a_1x + a_2y + a_3xy, a_i \in \mathbb{R}\}$ with $\dim(Q_1(\hat{K})) = 4$. Then the principal lattice is $K_1^2 = \{(0,0),(0,1),(1,0),(1,1)\}$. $\Sigma_1^2 = \{p(0,0), p(0,1), p(1,0), p(1,1)\}$ is just the principal degrees of freedom set and forms a dual basis of $Q_1(\hat{K})$. The corresponding canoncial basis in $Q_1(\hat{K})$ (or called nodal basis) can be explicitly written as: $p_1(x,y) = xy, p_2(x,y) = x(1-y), p_3(x,y) = (1-x)y, p_4(x,y) = (1-x)(1-y)$. They form a basis in $Q_1(\hat{K})$. Finally using the affine map F_{ij} we can derive the finite element $(K_{ij}, P(K_{ij}), \Sigma(K_{ij}))$ for $1 \le i,j \le N$.

3.2 Type (1) triangles

For type (1) triangles we can do the similar things. We also use quite simple grid by dividing each cube obtained from the previous section into two triangles along the diagonal line orthogonal to the line $y = x$ in the plane. Then each triangle got in this way is affine equivalent to the reference element $\hat{K}$ formed by the nodes $(0,0),(1,0),(0,1)$. If we divide $[0,1]$ uniformly into n parts we can get $2n^2$ triangles in this case. A simple demonstrating this case is shown below:

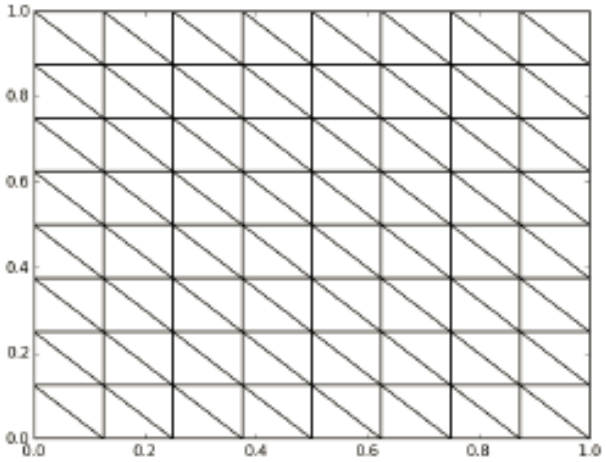

Figure 2: An example of the uniform triangulation used. Here we have triangles of side length $1/8$, giving a total of 128 total triangles with $n = 8$ parts into which $[0, 1]$ is divided.

For the reference element we can construct the finite element space explicitly. Since $k = 1, n = 2$ we have $\hat{K} = \{x = \lambda_0\vec{a}_0 + \lambda_1\vec{a}_1 + \lambda_2\vec{a}_2 : 0 \leq \lambda_i \leq 1, \lambda_0 + \lambda_1 + \lambda_2 = 1\}$ and $\vec{a}_0 = (0, 0), \vec{a}_1 = (1, 0), \vec{a}_2 = (0, 2)$. The function space $P_{\hat{K}} = P_1(\hat{K})$ contains all linear function in 2 variables (x, y) and $\dim(P_1(\hat{K})) = 3$. $p(x) \in P_1(\hat{K})$ can be written as $p(x) = a_0 + a_1x + a_2y$ for $a_i \in \mathbb{R}$. The principal lattice $K_1^2 = \{(0, 0), (1, 0), (0, 1)\}$ and the principal degrees of freedom set is $\Sigma_1^2 = \{p(0, 0), p(1, 0), p(0, 1)\}$. The corresponding nodal basis is just the barycentric coordiantes and can be written explicitly: $p_1(x, y) = 1 - x - y, p_2(x, y) = x, p_3(x, y) = y$. They form a basis in $P_1(\hat{K})$. Finally using the affine map we can derive the original finite elements. Here the affine map can be written as $F(\hat{x}, \hat{y}) = \begin{bmatrix} x_1 - x_0 & x_2 - x_1 \\ y_1 - y_0 & y_2 - y_1 \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$ if we map the vertices $(0, 0), (1, 0), (0, 1)$ to $(x_0, y_0), (x_1, y_1), (x_2, y_2)$. So we just need to specify the nodal correspondence between two triangles for the two type triangles in the above mesh generated. The following figure displays the correspondence where the same numbers on the nodes indicate this relation.

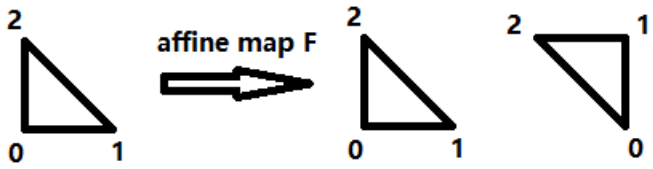

Figure 3: A figure illustrating the correspondence between nodes before and after the affine map $F : \hat{K} \to K$, where the same number indicates these two points are mapped against each other.

4 Approximate the variational problem

All the nodal basis in the previous construction form a basis for $V_h(0)$ (we shall extend each nodal basis to all possible support, and only nodal basis corresponding to the nodes in $\Omega - \partial\Omega_0$ is considered, as $u = 0$ on $\partial\Omega_0$). Each function in this basis corresponds to

a node in the grid generated, denoted as $\{\psi_i\}$ for $1 \le i \le T = (N-1)N$ as there're $(N-1)N$ nodes in both the type (1) rectangles and triangles that do not lie on $\partial\Omega_0$.
Then we can transform the discretization problem (the finite element problem) into linear equations: we only need to find $u = \sum_{i=1}^{T} u^i\psi_i \in V_h(0)$ such that $a(u,\psi_j) = \tilde{f}(\psi_j)$, by the linearity of $a(\cdot,\cdot)$ we have $\sum_{i=1}^{T} u^i a(\psi_i,\psi_j) = \tilde{f}(\psi_j)$. Thus we only need to solve linear equations $Ax = b$, where $A = (a_{ij})_{i,j=1}^{T}$ with $a_{ij} = a(\psi_i,\psi_j)$ and $b = (b_i)_{i=1}^{T}$ with $b_i = \tilde{f}(\psi_i)$. Next we only need to dicuss the form of A, b case by case to construct the FEM. Notice $a(\psi_i,\psi_j) \neq 0$ if and only if the nodes corresponding to ψ_i, ψ_j form a finite element in our constructions.
We use the method integration.dblquad and integration.quad in scipy package of python to compute the numerical integration in the calculation of $a(\cdot,\cdot)$ and $f(\cdot)$. Moreover we use the quadratic and linear interpolation to compute the linear functional $f(\cdot)$.

4.1 Matrix A

The computation of entries in $A = (a_{\alpha\beta})_{\alpha,\beta=1}^{N}$ can be simplifed as (if we assume F is the affine map from the reference element $\hat{K}$ to certain element K with $\psi_\alpha(x) = p_i(F^{-1}(x))$ and $\psi_\beta(x) = p_j(F^{-1}(x))$ being its two nodal basis):

$$
\begin{aligned}
a_{\alpha,\beta} = a(\psi_\alpha,\psi_\beta) &= \int_{\partial\Omega_1} \beta\psi_\alpha\psi_\beta ds + \int_\Omega \nabla\psi_\alpha\cdot\nabla\psi_\beta dx,\\
\int_\Omega \nabla\psi_\alpha\cdot\nabla\psi_\beta dx &= \sum_{m=1}^{N}\sum_{n=1}^{N}\int_{K_{mn}} \nabla\psi_\alpha\cdot\nabla\psi_\beta dx\\
&= \sum_{m=1}^{N}\sum_{n=1}^{N}\int_{\hat{K}} \nabla\psi_\alpha(F_{m,n}(\hat{x}))\cdot\nabla\psi_\beta(F_{m,n}(\hat{x}))\left|\frac{dF_{m,n}}{d\hat{x}}\right| d\hat{x}\\
&= \sum_{m=1}^{N}\sum_{n=1}^{N}\int_{\hat{K}} (J_{m,n}^{-1}\nabla p_i)\cdot(J_{m,n}^{-1}\nabla p_j)\,|J_{m,n}|\,d\hat{x},\\
&\quad \text{(where } J_{m,n} \text{ is the Jacobian of the affine map } F_{m,n})\\
\text{and } \int_{\partial\Omega_1}\beta\psi_\alpha\psi_\beta ds &= \sum_{e\in\partial\Omega_1}\int_e \beta(F_e(\hat{x}))\psi_\alpha(F_e(\hat{x}))\psi_\beta(F_e(\hat{x}))\left|\frac{dF_e}{d\hat{x}}\right| d\hat{x}\\
&= \sum_{e\in\partial\Omega_1}\int_e \beta(F_e(\hat{x}))p_i(\hat{x})p_j(\hat{x})\,|J_e|\,d\hat{x},
\end{aligned}
$$

where in the last identity we use F_e to denote the map from one edge $\hat{e} \in \hat{K}$ to another edge $e \in \partial\Omega$, which is also a restriction of the original affine map $K_{m,n}$ for some m,n and J_e is its Jacobian.
This gives a universal way to compute the stiffness matrix A for general functions and elements including rectangles and triangles.

4.2 Vector b

Then we shall specify how we calculate the right hand side, the vector b, i.e. how we cope with the linear functional $f(\cdot)$. Similarly we can give a universal way to compute this term using quadratic or linear interpolation to compute the numerical integration.

For simplicity we will use $\tilde{u}_0(x,y) = u_0(y)$ on $(x,y) \in \Omega$ as an extension. Here we have $\tilde{f}(\psi) = f(\psi) - a(\tilde{u}_0, \psi) = \int_\Omega f\psi dx + \int_{\partial\Omega_1}(g - \beta\tilde{u}_0)\psi ds - \int_\Omega \nabla\tilde{u}_0 \cdot \nabla\psi dx$ and it can be divided into three parts.

For the first part $\int_\Omega f\psi_\beta dx$, if we use the finite element function space to approximate $f(x)$, i.e. $f(x) \approx \sum_\alpha f_\alpha\psi_\alpha$ (use f_α to denote the function value at the coorepsonding node of the basis ψ_α), we have

$$\int_\Omega f\psi_\beta dx \approx \int_\Omega \left(\sum_\alpha f_\alpha\psi_\alpha(x)\right)\psi_\beta(x)dx = \sum_\alpha f_\alpha \int_\Omega \psi_\alpha(x)\psi_\beta(x)dx = \sum_\alpha f_\alpha m_{\alpha,\beta}.$$

So we only need to calculate the mass matrix $M = (m_{\alpha,\beta})_{\alpha,\beta=1}^N$ with $m_{\alpha,\beta} = \int_\Omega \psi_\alpha(x)\psi_\beta(x)dx$ to compute the first integration. Similar to the previous section this computation can be simplified as $\int_\Omega \psi_\alpha(x)\psi_\beta(x)dx = \sum_K \int_K p_i(\hat{x})p_j(\hat{x})\,|J_K|\,d\hat{x}$.
For the second part $\int_{\partial\Omega_1}(g - \beta\tilde{u}_0)\psi ds$ we use the same approximation $f(x) \approx \sum_\alpha f_\alpha\psi_\alpha$ where the function f at here is replaced with $g - \beta\tilde{u}_0$. Whereas in the first part this is a quadratic interpolation of $f(x)$ in one element, in this case this can be treat as a linear interpolation on one particular edge. And we only need to calculate $m^b_{\alpha,\beta} = \int_{\partial\Omega_1}\psi_\alpha\psi_\beta ds = \sum_{e\in\partial\Omega_1}\int_e p_i(\hat{x})p_j(\hat{x})\,|J_e|\,d\hat{x}$ and $\int_{\partial\Omega_1}(g - \beta\tilde{u}_0)\psi_\beta ds \approx \sum_\alpha (g - \beta\tilde{u}_0)_\alpha m^b_{\alpha,\beta}$.
For the third part $\int_\Omega \nabla\tilde{u}_0 \cdot \nabla\psi dx$ the situation is the same, but to be more accurate we can use this approximation for $\frac{\partial}{\partial y}\tilde{u}_0$ instead of $\tilde{u}_0$ since we have $\frac{\partial}{\partial x}\tilde{u}_0 = 0$ (we assume the extension has the form $\tilde{u}_0(x,y) = u_0(y)$). So we only need to calculate $m^y_{\alpha,\beta} = \int_\Omega \psi_\alpha\frac{\partial}{\partial y}\psi_\beta dx = \sum_K \int_K (0, p_i(\hat{x}))\cdot(J_K^{-1}\nabla p_j)\,|J_K|\,d\hat{x}$ and we have $\int_\Omega \nabla\tilde{u}_0\cdot\nabla\psi_\beta dx = \int_\Omega \frac{\partial\tilde{u}_0}{\partial y}\frac{\partial\psi_\beta}{\partial y}dx \approx \sum_\alpha(\frac{\partial}{\partial y}\tilde{u}_0)_\alpha m^y_{\alpha,\beta}$.

Therefore to compute the vector b all we need are the three mass matrices with elements $m_{\alpha,\beta} = \int_\Omega \psi_\alpha(x)\psi_\beta(x)dx, m^b_{\alpha,\beta} = \int_{\partial\Omega_1}\psi_\alpha\psi_\beta ds$ and $m^y_{\alpha,\beta} = \int_\Omega \psi_\alpha\frac{\partial}{\partial y}\psi_\beta dx$.

5 Error estimation

Next we shall briefly explain how we estimate the error in using this FEM to solve the original PDE. Since we have the real solution here we shall consider the $||\cdot||_2, ||\cdot||_\infty$ and $||\cdot||_{H^1}$ norms here to evaluate the performance and draw the log-log plot to discuss the convergence rate. For the norm we will use the numerical integration to approximate it with suitable points chosen in each grid. For the $||\cdot||_2, ||\cdot||_\infty$ norm we just use the error at the grid points to approximate it. But for $||\cdot||_{H^1}$, since we only have C^0 finite elements and notice the only discontinuity of its derivative is at the edges of the grid, we just chose some special inner point to evaluate it. For the rectagle $K = [a, a+h]\times[b, b+h]$ we use the point $n_K(a + h/2, b + h/2)$ and for the triangle with vertices $K = (x_0, y_0), (x_1, y_1), (x_2, y_2)$ we use the point $n_K = (\frac{x_0+x_1+x_2}{3}, \frac{y_0+y_1+y_2}{3})$. To be precise we have

$$\begin{aligned}
||\cdot||_{0,2} &\approx (S(K)\sum_{i=1}^{N(N-1)} |u^*(A_i) - U(A_i)|^2)^{\frac{1}{2}},\\
||\cdot||_{0,\infty} &\approx \max_{1\le i\le N(N-1)} |u^*(A_i) - U(A_i)|,\\
||\cdot||_{H^1} &\approx (S(K)\sum_K |\nabla u^*(n_K) - \nabla U(n_K)|^2)^{\frac{1}{2}},
\end{aligned}$$

where we use A_i to denote the total $N(N-1)$ nodes of the grid that's not on $\partial\Omega_0$ and $S(K)$ to denote the area for a single element, which is $h^2 = 1/N^2$ for rectangles and $h^2/2 = 1/2N^2$ for triangles. Use u^*, U to denote the real solution to the original problem and the grid function obtained from the FEM.

To evaluate the convergence rate we use the loglog plot. If the error $E(h) \approx Ch^p$ with grid step size h. We shall then have

$$\log|E(h)| \approx \log|C| + p\log h,$$

so on a log-log scale the error behaves linearly with a slope equals to p, the order of the accuracy. In the next section we shall draw the curve and estimate the slope by fitting the data with a line.

6 Numerical results

We solve the Poisson problem with mixed boundary condition with the above demonstrated method with the grid size N from 2 to 20 to draw the loglog plot in L_2, L_∞, H^1 error. The following are the plots (with two functions and two finite element choices):

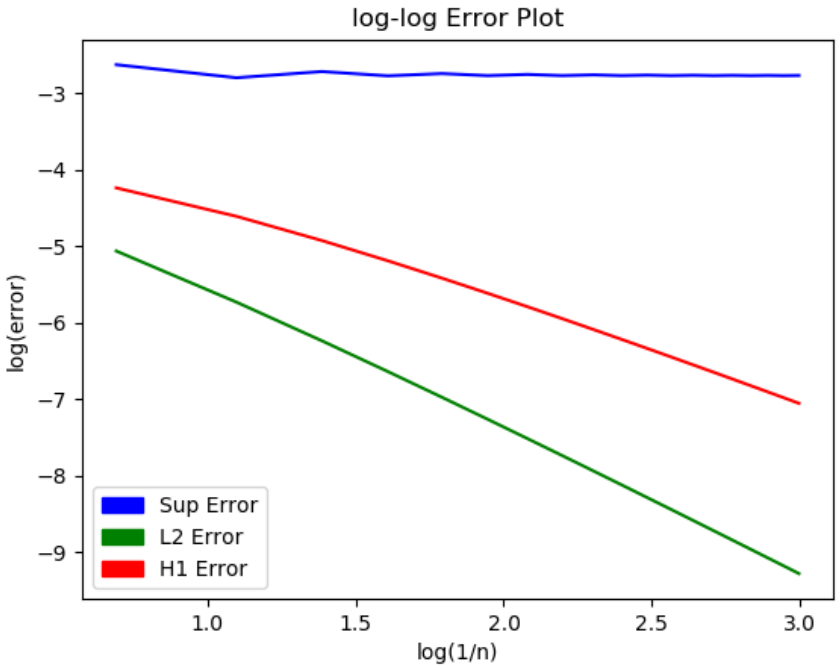

Figure 4: The log-log plot for the type (1) rectangle FEs in the first case with $u^* = \frac{1}{4}[(x-\frac{1}{2})^2 + (y-\frac{1}{2})^2] - \frac{1}{8}$, the estimated slope is 0.0302 for $||\cdot||_\infty$, 1.8571 for $||\cdot||_{0,2}$ and 1.2653 for $||\cdot||_{1,2}$.

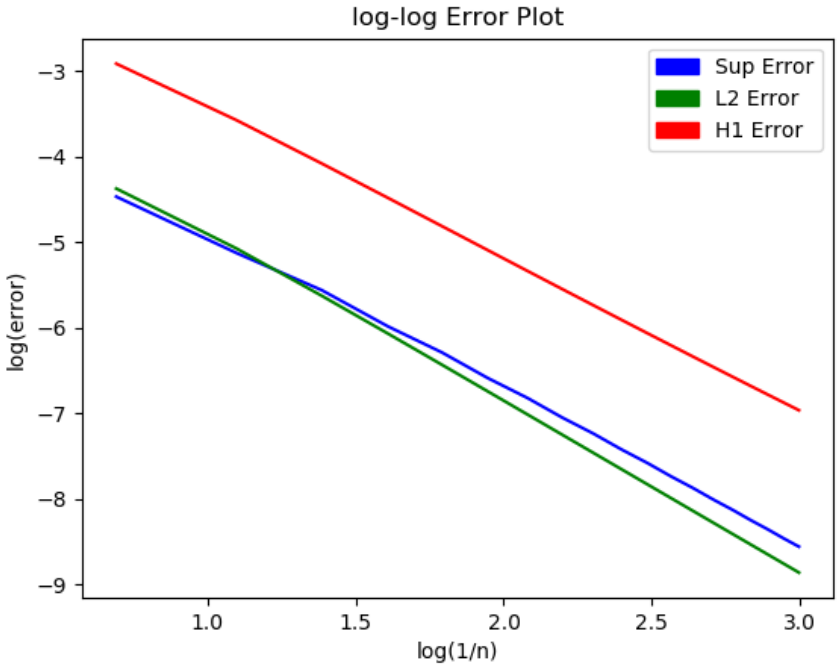

Figure 5: The log-log plot for the type (1) rectangle FEs in the second case with $u^* = x(1-x)y(1-y)$, the estimated slope is 1.8042 for $||\cdot||_\infty$, 1.9784 for $||\cdot||_{0,2}$ and 1.7788 for $||\cdot||_{1,2}$.

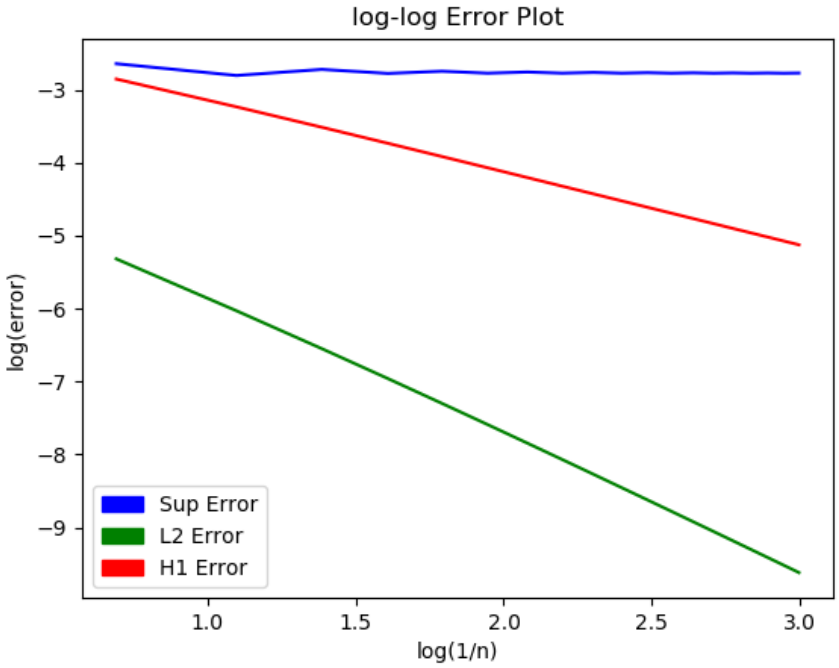

Figure 6: The log-log plot for the type (1) triangle FEs in the first case with $u^* = \frac{1}{4}[(x-\frac{1}{2})^2 + (y-\frac{1}{2})^2] - \frac{1}{8}$, the estimated slope is 0.0269 for $||\cdot||_\infty$, 1.8839 for $||\cdot||_{0,2}$ and 0.9939 for $||\cdot||_{1,2}$.

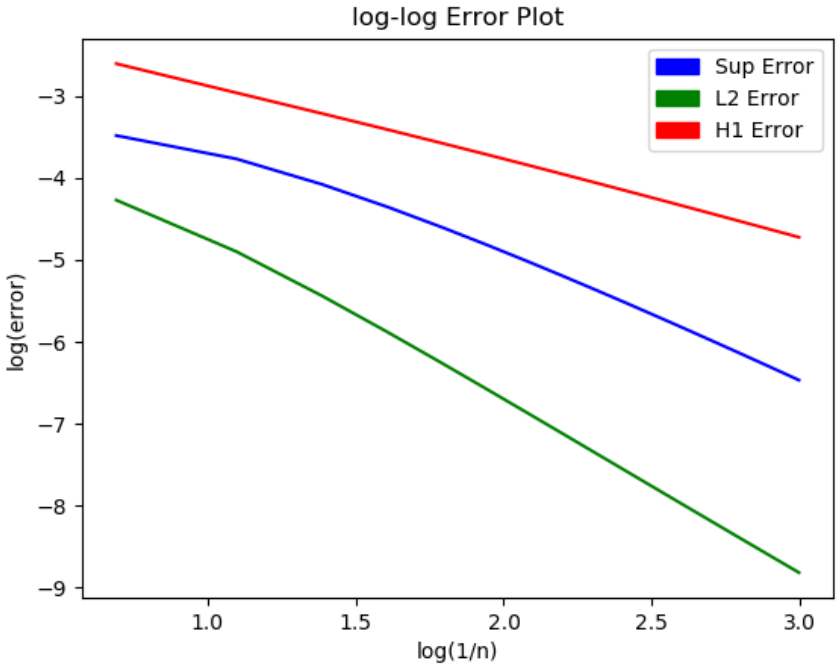

Figure 7: The log-log plot for the type (1) triangles FEs in the second case with $u^* = x(1-x)y(1-y)$, the estimated slope is 1.3757 for $||\cdot||_\infty$, 2.0335 for $||\cdot||_{0,2}$ and 0.9246 for $||\cdot||_{1,2}$.

From the theoretical analysis we know for quasi-uniform, regular C^0- type(1) Lagrange affine equivalent FEs we shall have $O(h)$ H^1 error and $O(h^2)$ L^2 error. This theortical results are roughly consistentt with our numerical results. The order of accuracy for $||\cdot||_{0,2}$ is roughly 2 and the order for $||\cdot||_{1,2}$ is roughly 1. For the $||\cdot||_{0,\infty}$, its numerical results is around 0 for the frist function and bigger than 1 for the second function. Maybe this norm is greatly affected by the property for the specfic function as it concerns with its extreme value, some local property rather than the average behavior or the global property such as $||\cdot||_{0,2}, ||\cdot||_{1,2}$ behaves.

Moreover, we can also draw the 3D plot for our approximated solution with $N = 26$. In this case the error is

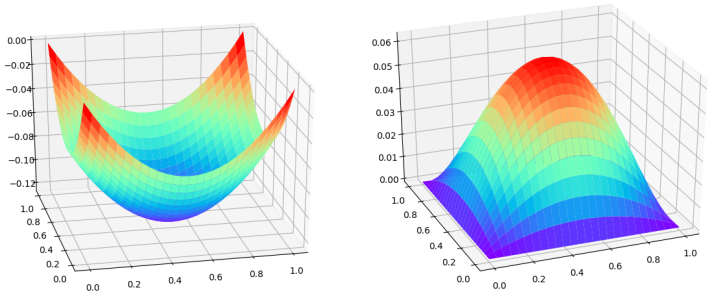

Figure 8: This is a demonstration of the 3D plot of our numerical solution to the two problem using rectangle FEs. The left one is for $u^* = \frac{1}{4}[(x-\frac{1}{2})^2 + (y-\frac{1}{2})^2] - \frac{1}{8}$ and the right one is for $u^* = x(1-x)y(1-y)$. The error for the left one and the right is 6e-4 (both) for $||\cdot||_{1,2}$, 1e-4 and 8.4e-5 for $||\cdot||_{0,2}$, 0.0626 and 1.16e-4 for $||\cdot||_{0,\infty}$

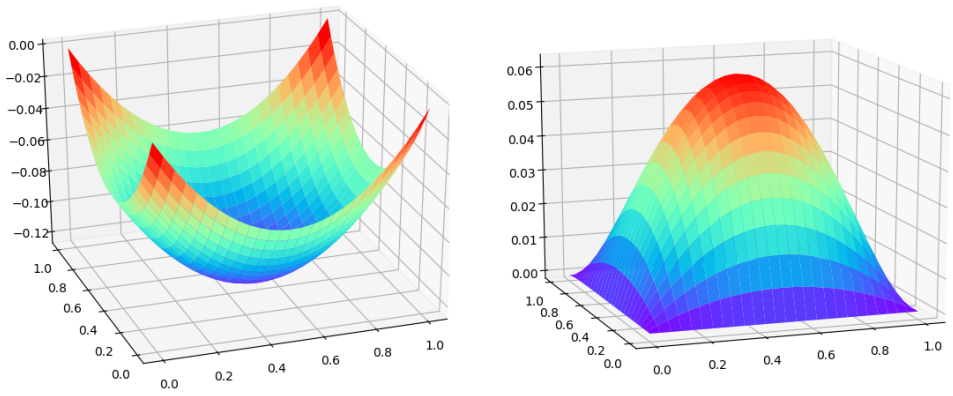

Figure 9: This is a demonstration of the 3D plot of our numerical solution to the two problem using triangle FEs. The left one is for $u^* = \frac{1}{4}[(x-\frac{1}{2})^2 + (y-\frac{1}{2})^2] - \frac{1}{8}$ and the right one is for $u^* = x(1-x)y(1-y)$. The error for the left one and the right is 6.8e-3 (both) for $||\cdot||_{1,2}$, 4e-5 and 8.6e-5 for $||\cdot||_{0,2}$, 0.0626 and 0.001 for $||\cdot||_{0,\infty}$

As we have expected in this case when $n = 26$, the function solved by FEM has roughly the shape we expect the real solution behaves. Our results can be treat as an acceptable approximation to the real solution.