FEM Homework Report

Yue Wu*

October 9, 2022

1 Introduction

Make a program to solve the boundary value problem with purely Dirichlet boundary condition:

$$\begin{cases} -\Delta u = f, \quad x \in \Omega \\ u\big|_\Gamma = 0 \end{cases} \tag{1}$$

Use a triangular mesh and a piecewise linear polynomial space V_h as the finite element space. Use $f(x,y) = 2(1-x)\sin x \cos y + 2(1-y)\sin y \cos x - 2(1-x)(1-y)\sin x \sin y$, $u(x,y) = (x-1)(y-1)\sin x \sin y$, $\Omega = (0,1)^2$, $\Gamma = \partial\Omega$ to test the program, and compute the following errors:

$$\|u - u_h\|_{L^2(\Omega)}, \quad \|u - u_h\|_{H^1(\Omega)} \tag{2}$$

2 Method

2.1 The Galerkin approximation

We use a triangle mesh $\mathcal{T}_h$ to discretize the computational domain $\Omega_h = \Omega$, $\Gamma_h = \partial\Omega_h$ (since Ω is a polygon). Denote h to be the maximal arc-length of the elements as the characteristic length of the mesh. Define the piecewise-linear finite element space:

$$V_h = \left\{ v \in H^1(\Omega_h) \middle| v\big|_K \in \mathcal{P}^1(K), \forall K \in \mathcal{T}_h, v\big|_{\Gamma_h} = 0 \right\} \tag{3}$$

The Galerkin approximation to the problem using approximation space V_h is:

$$\begin{aligned} &\text{find } u_h \in V_h, \text{ such that:} \\ &\forall v_h \in V_h,\ \left(u_h', v_h'\right) = (f, v_h) \end{aligned} \tag{4}$$

2.2 The reference element

We use the right triangle as the reference element

$$\widehat{K} = \left\{(x,y) \middle| 0 \leqslant x, y \leqslant x + y \leqslant 1\right\} \tag{5}$$

with vertices $\widehat{P}_1 = (1,0)$, $\widehat{P}_2 = (0,1)$ and $\widehat{P}_3 = (0,0)$. Within this reference element, the barycentric coordinates are given by $(\lambda_1, \lambda_2, \lambda_3) = (x, y, 1-x-y)$ and the linear nodal bases are given by:

$$\widehat{N}_1(x,y) = \lambda_1(x,y) \tag{6a}$$

*School of the Gifted Young, University of Science and Technology of China, Hefei, Anhui, China. E-mail: pilotjohnwu@mail.ustc.edu.cn

$$\widehat{N}_2(x,y) = \lambda_2(x,y) \tag{6b}$$
$$\widehat{N}_3(x,y) = \lambda_3(x,y) \tag{6c}$$

Let $\widehat{\boldsymbol{N}}(x,y) = \left(\widehat{N}_1(x,y), \widehat{N}_2(x,y), \widehat{N}_3(x,y)\right)$. The reference mass matrix is

$$\widehat{\boldsymbol{M}} = \int_{\widehat{K}} \widehat{\boldsymbol{N}}(x,y)^T \widehat{\boldsymbol{N}}(x,y)\mathrm{d}x\mathrm{d}y = \begin{pmatrix} \frac{1}{12} & \frac{1}{24} & \frac{1}{24} \\ \frac{1}{24} & \frac{1}{12} & \frac{1}{24} \\ \frac{1}{24} & \frac{1}{24} & \frac{1}{12} \end{pmatrix} \tag{7}$$

The reference gradient operator (which maps nodal values to gradient values) is

$$\widehat{\boldsymbol{D}} = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & -1 \end{pmatrix} \tag{8}$$

and the differentiation relations are

$$\frac{\partial}{\partial x}\boldsymbol{N}(x,y) = \boldsymbol{N}(x,y) \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix} = \boldsymbol{N}(x,y)\widehat{\boldsymbol{D}}_1 \tag{9a}$$

$$\frac{\partial}{\partial y}\boldsymbol{N}(x,y) = \boldsymbol{N}(x,y) \begin{pmatrix} 0 & 1 & -1 \\ 0 & 1 & -1 \\ 0 & 1 & -1 \end{pmatrix} = \boldsymbol{N}(x,y)\widehat{\boldsymbol{D}}_2 \tag{9b}$$

The reference stiffness matrix is

$$\widehat{\boldsymbol{S}} = \left(\int_{\widehat{K}} \nabla N_i \cdot \nabla N_j \mathrm{d}\sigma\right)_{3\times 3} = \widehat{\boldsymbol{D}}_x^T \widehat{\boldsymbol{M}} \widehat{\boldsymbol{D}}_x + \widehat{\boldsymbol{D}}_y^T \widehat{\boldsymbol{M}} \widehat{\boldsymbol{D}}_y = \begin{pmatrix} \frac{1}{2} & 0 & -\frac{1}{2} \\ 0 & \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{2} & 1 \end{pmatrix} \tag{10}$$

2.3 The local bases and matrices

For an element $K = \Delta P_i P_j P_k$, where $P_i = (x_i, y_i)$, $P_j = (x_j, y_j)$ and $P_k = (x_k, y_k)$, the Jacobian and the local linear nodal bases are

$$\boldsymbol{J}_K = \frac{\partial\,(x,y)}{\partial(\lambda_1,\lambda_2)} = \begin{pmatrix} x_i - x_k & x_j - x_k \\ y_i - y_k & y_j - y_k \end{pmatrix} \tag{11a}$$

$$N_i(x,y) = \frac{1}{2\Delta_K} \begin{vmatrix} 1 & x & y \\ 1 & x_j & y_j \\ 1 & x_k & y_k \end{vmatrix} = \lambda_i(x,y) \tag{11b}$$

$$N_j(x,y) = \frac{1}{2\Delta_K} \begin{vmatrix} 1 & x_i & y_i \\ 1 & x & y \\ 1 & x_k & y_k \end{vmatrix} = \lambda_j(x,y) \tag{11c}$$

$$N_k(x,y) = \frac{1}{2\Delta_K} \begin{vmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x & y \end{vmatrix} = \lambda_k(x,y) \tag{11d}$$

where Δ denotes the **signed** area of the triangle, which is given by:

$$\Delta_K = \frac{1}{2}\det\left(\boldsymbol{J}_K\right) \tag{12}$$

Let $\boldsymbol{N}(x,y) = (N_1, N_2, N_3)$. The local mass matrixis:

$$\boldsymbol{M}_K = \left|\frac{\partial(x,y)}{\partial(\lambda_1,\lambda_2)}\right| \widehat{\boldsymbol{M}} = \det\left(\boldsymbol{J}_K\right)\widehat{\boldsymbol{M}} \tag{13}$$

The local gradient operator (which maps nodal values to gradient values) is:

$$\boldsymbol{D}_K = \boldsymbol{J}^{-T}\widehat{\boldsymbol{D}} = \frac{1}{2\Delta}\begin{pmatrix} y_j - y_k & x_k - x_j \\ y_k - y_i & x_i - x_k \end{pmatrix}^T \widehat{\boldsymbol{D}} = \frac{1}{2\Delta}\begin{pmatrix} y_j - y_k & y_k - y_i & y_i - y_j \\ x_k - x_j & x_i - x_k & x_j - x_i \end{pmatrix} \tag{14}$$

and the local differentiation relations are

$$\frac{\partial}{\partial x}\boldsymbol{N}(x,y) = \boldsymbol{N}(x,y)\left(\frac{\partial \lambda_1}{\partial x}\widehat{\boldsymbol{D}}_1 + \frac{\partial \lambda_2}{\partial x}\widehat{\boldsymbol{D}}_2\right) \tag{15a}$$

$$\frac{\partial}{\partial y}\boldsymbol{N}(x,y) = \boldsymbol{N}(x,y)\left(\frac{\partial \lambda_1}{\partial y}\widehat{\boldsymbol{D}}_1 + \frac{\partial \lambda_2}{\partial y}\widehat{\boldsymbol{D}}_2\right) \tag{15b}$$

The local stiffness matrix is

$$\begin{aligned}
\boldsymbol{S}_K &= \int_K \left(\nabla_{\boldsymbol{x}}\boldsymbol{N}\right)^T \nabla_{\boldsymbol{x}}\boldsymbol{N}\mathrm{d}\boldsymbol{x} \\
&= \int_{\widehat{K}} \left(\frac{\partial(\lambda_1,\lambda_2)}{\partial(x,y)}^T \nabla_{\boldsymbol{\lambda}}\widehat{\boldsymbol{N}}\right)^T \left(\frac{\partial(\lambda_1,\lambda_2)}{\partial(x,y)}^T \nabla_{\boldsymbol{\lambda}}\widehat{\boldsymbol{N}}\right) \left|\frac{\partial(x,y)}{\partial(\lambda_1,\lambda_2)}\right| \mathrm{d}\boldsymbol{\lambda} \\
&= \left(\frac{\partial \lambda_1}{\partial x}\widehat{\boldsymbol{D}}_1 + \frac{\partial \lambda_2}{\partial x}\widehat{\boldsymbol{D}}_2\right)^T \boldsymbol{M}_K \left(\frac{\partial \lambda_1}{\partial x}\widehat{\boldsymbol{D}}_1 + \frac{\partial \lambda_2}{\partial x}\widehat{\boldsymbol{D}}_2\right) \\
&\quad + \left(\frac{\partial \lambda_1}{\partial y}\widehat{\boldsymbol{D}}_1 + \frac{\partial \lambda_2}{\partial y}\widehat{\boldsymbol{D}}_2\right)^T \boldsymbol{M}_K \left(\frac{\partial \lambda_1}{\partial y}\widehat{\boldsymbol{D}}_1 + \frac{\partial \lambda_2}{\partial y}\widehat{\boldsymbol{D}}_2\right) \\
&= \begin{pmatrix} \widehat{\boldsymbol{D}}_1^T & \widehat{\boldsymbol{D}}_2^T \end{pmatrix} \left(\left(\boldsymbol{J}_K^T\boldsymbol{J}_K\right)^{-1} \otimes \boldsymbol{M}_K\right) \begin{pmatrix} \widehat{\boldsymbol{D}}_1 \\ \widehat{\boldsymbol{D}}_2 \end{pmatrix} \\
&= \begin{pmatrix} \widehat{\boldsymbol{D}}_1^T & \widehat{\boldsymbol{D}}_2^T \end{pmatrix} \left(\left(\boldsymbol{J}_K^T\boldsymbol{J}_K\right)^{-1} \otimes \det\left(\boldsymbol{J}_K\right)\widehat{\boldsymbol{M}}\right) \begin{pmatrix} \widehat{\boldsymbol{D}}_1 \\ \widehat{\boldsymbol{D}}_2 \end{pmatrix}
\end{aligned} \tag{16}$$

To approximate integrals of non-polynomial functions, we apply the 3-point quadrature rule:

$$\int_{\widehat{K}} f(x,y)\mathrm{d}\sigma \approx \left|\widehat{K}\right| \left(\frac{1}{3}f(\frac{1}{6},\frac{1}{6}) + \frac{1}{3}f(\frac{1}{6},\frac{2}{3}) + \frac{1}{3}f(\frac{2}{3},\frac{1}{6})\right) \tag{17}$$

which has 2-order algebraic exactness.

Remark 1. *In order to make it easier to assembly the LHS matrix and RHS vector, we do not exclude boundary nodes from the total DOFs. For each j such that P_j is a boundary node, we enforce the j-th entry of the RHS vector, the j-th row and column of the LHS matrix (except the (j,j)-th entry) to be zero, and enforce the (j,j)-th entry of the LHS matrix to be 1. In this manner, the LHS matrix is kept symmetric positive definite.*

3 Results

We use the `PDE Toolbox` in `matlab` to generate the meshes in the way of repeatedly refining the coarsest mesh. By running the scripts in `main.m`, the outputs from the `matlab` program we coded are listed below.

Our initial mesh and corresponding LHS matrix sparsity pattern are shown in the following figures:

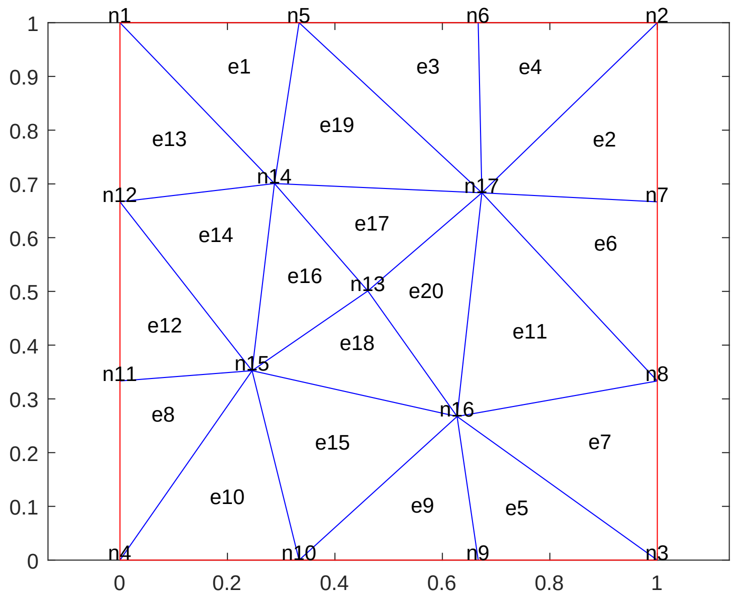

Figure 1: the coarsest mesh

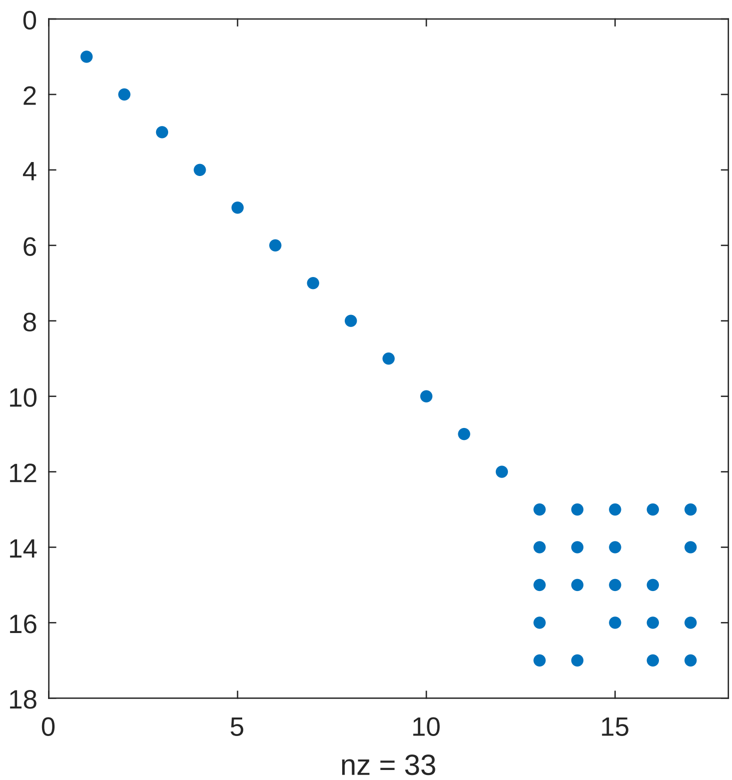

Figure 2: sparsity pattern for the coarsest mesh

The errors, meshes and sparsity pattern are shown below:

triangles	interior nodes	boundary nodes	L^2 error	order	H^1 error	order
20	5	12	5.1839e-03	-	5.3517e-02	-
80	29	24	1.4646e-03	1.8235	2.8770e-02	0.8954
320	137	48	3.8236e-04	1.9375	1.4707e-02	0.9681
1280	593	96	9.6928e-05	1.9800	7.4036e-03	0.9902

Table 1: Table of error and order

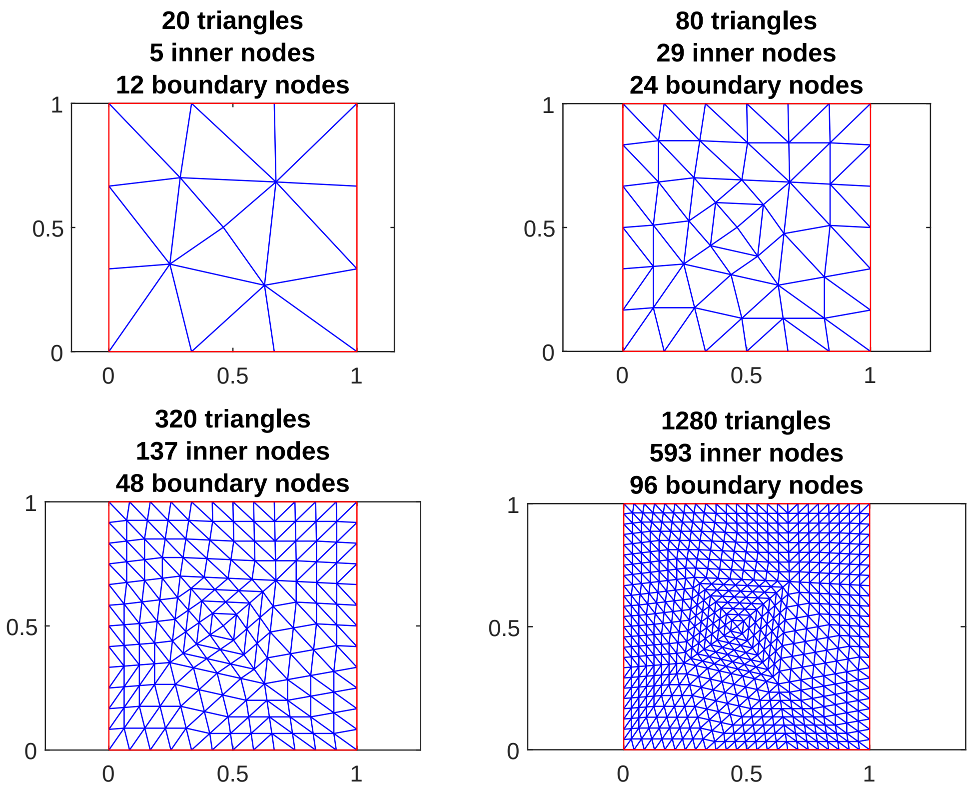

Figure 3: meshes

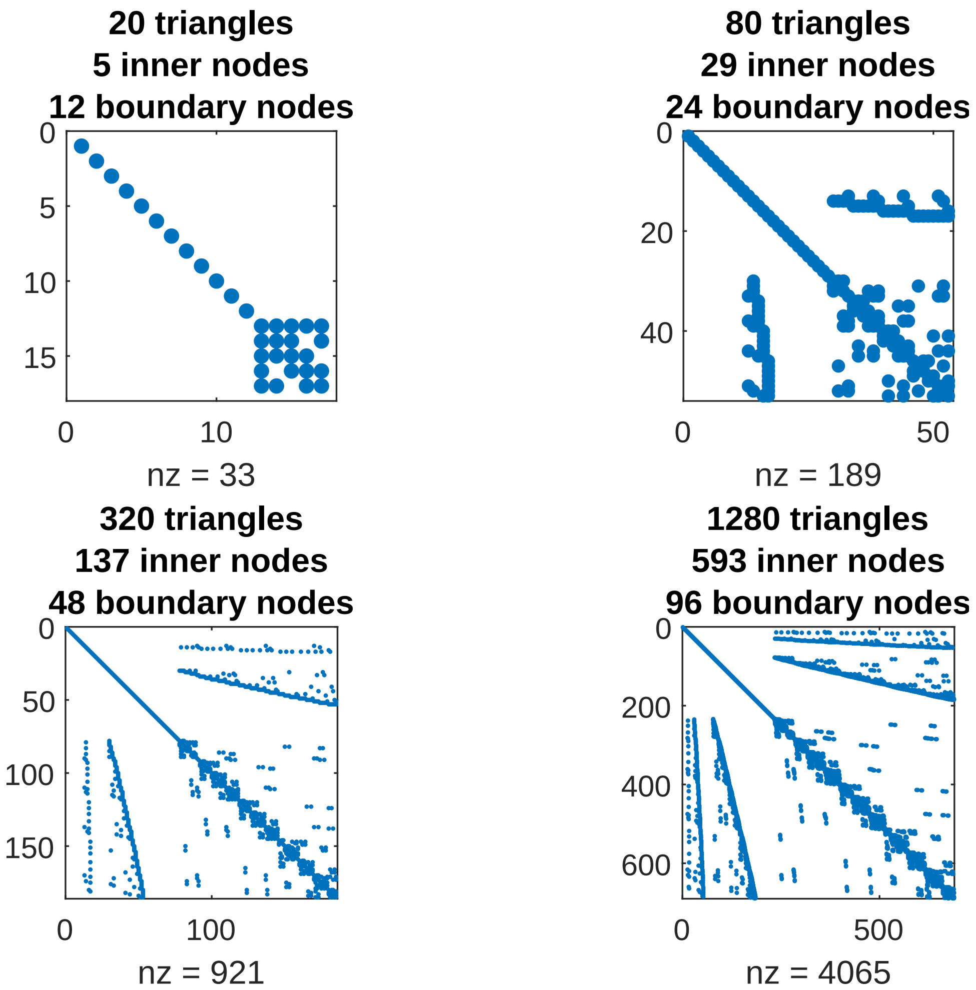

Figure 4: sparsity patterns

4 Discussion

From this experiement, we verified the theoretical 2-nd order convergence under the L^2 norm and 1-st order convergence under the H^1 norm for piecewise linear conforming FEM numerically. The numerical convergence orders shown in the table seem to approach 2 and 1 from below, respectively.

We learned from the experiement that in higher dimensions, there are not collocation methods where the collocation points can serve as high-order quadrature points for exact integration of inner-products between basis polynomials. In this program, we used the special 3-point quadrature rule to approximate integration of non-polynomial functions in the right-hand-side. This does not cause the convergence order to degenerate since the quadrature has algebraic exactness of order 2. Meanwhile, the unstructure nature of the triangular mesh makes it harder for us to assemble the left-hand-side matrix, especially in the case of Dirichlet BCs, thus we need to adapt new strategies to enforce such boundary condition. Last but not least, when testing FEM programs using unstructured grids, we have to refine the coarse mesh by a factor of 2 in order to preserve similarity between elements in most of the cases. As a result, the computational cost could be prohibitive after several refinement and our test numbers might be limited.