Report on FEM Programming Homework 4

Yuejie Liu PB18010470

November 21, 2021

1 Problem

Solve

$$\begin{cases} -\Delta u = f, & x \in \Omega \\ u = 0, & on \partial\Omega \end{cases} \quad \Omega = [0,1] * [0,1] \tag{1}$$

by

$$V_h = \{v \in H_0^1(\Omega), v|_K \in P^1(K), \forall K \in \tau_h\}$$

Give an example to show the convergence.

2 Algorithm

2.1 Variations of PDE and FEM

We let $u(x,y) = \sin(2\pi x)\sin(2\pi y)$, then $f(x) = -8\pi^2 \sin(2\pi x)\sin(2\pi y)$
The solution space is

$$V = H_0^1([0,1])$$

and the weak form is

$$\begin{cases} Find\ u \in V s.t. \\ a(u,v) = F(v), \forall v \in V \end{cases} \tag{2}$$

where $a(u,v) = \int_\Omega \nabla u(x,y) \nabla v(x,y) dxdy, F(v) = \int_\Omega f(x,y)v(x,y)dxdy$
The finite element variation problem is

$$\begin{cases} Find\ u_h \in V_h(\subset V)\ s.t. \\ a(u_h,v) = F(v), \forall v \in V_h \end{cases} \tag{3}$$

2.2 The basis of V_h

We take nodal basis on each element. We assume on the i-th element $v_{ij} = a_{ij}x + b_{ij}y + c_{ij}, v_{ij}(\alpha_{ik}) = \delta_{jk}, j,k = 1,2,3, i = 1,..,N$. Then we solve the linear equations to obtain the coefficients and the expression of the basis function on each element.

However, it is necessary to notice that, as it in 1-dimension, the number of basis function is equal to the number of inner points. Take Mesh/gd0 as an example,

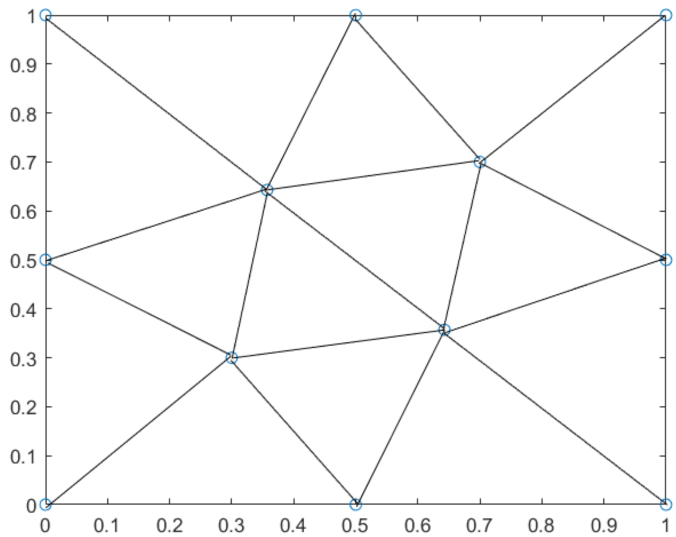


Figure 1: Mesh/gd0

the basis function are like four hills on the inner points.

2.3 Stiffness Matrix

$K = (a(\phi_i, \phi_j))_{M*M}$, where M is the number of basis function. We assemble K on each element, which means we add $(a_{mi} * a_{mj} + b_{mi} * b_{mj}) * Area(m), m = 1, ...N, i, j = 1, ..., 3$ on m-th element.

2.4 Numerical Integration Method and Linear Equations Solution Method

To obatin vector F, we need to compute integrals on each triangle element. First , we consider integrals on a standard triangle.

$$I = \int_K f(x,y)dxdy = \int_0^1 (\int_0^{-x+1} f(x,y)dy)dx$$

We substitute variables with $u = x, v = \frac{y}{1-x}$,

$$I = \int_0^1 \int_0^1 f(u, v(1-u))(1-u)dudv$$

Then we let $u = \frac{1+\xi}{2}, v = \frac{1+\eta}{2}$,

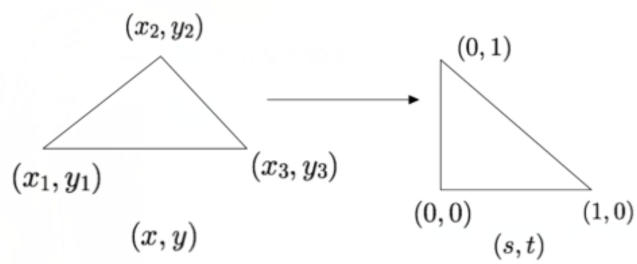
$$I = \int_{-1}^1 \int_{-1}^1 f(\frac{1+\xi}{2}, \frac{(1-\xi)(1+\eta)}{4})\frac{1-\xi}{8}d\xi d\eta$$

This is a double integral on $[-1,1]*[-1,1]$, so we use Gauss integral method to compute.

$$I_\eta = \sum_{k=1}^{3} A_k f(\frac{1+x_k}{2}, \frac{(1-x_k)(1+\eta)}{4})\frac{1-x_k}{8}$$

$$I = \sum_{m=1}^{3} A_m \sum_{k=1}^{3} A_k f(\frac{1+x_k}{2}, \frac{(1-x_k)(1+x_m)}{4})\frac{1-x_k}{8}$$

Next, we need to compute integrals on any given triangle.



$$\begin{pmatrix} x \\ y \end{pmatrix} = A \begin{pmatrix} s \\ t \end{pmatrix} + b = \begin{pmatrix} a_1 & a_2 \\ a_3 & a_4 \end{pmatrix} \begin{pmatrix} s \\ t \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ b_1 \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_3 \\ a_4 \\ b_2 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}$$

$$\int_{k_1} f(x,y)\, dxdy = \int_{k_2} f(a_1 s + a_2 t + b_1, a_3 s + a_4 t + b_2) |a_1 a_4 - a_2 a_3|\, dsdt$$

In conclusion, we have

$$I = \sum_{m=1}^{3} A_m \sum_{k=1}^{3} A_k f(a_1 \frac{1+x_k}{2} + a_2 \frac{(1-x_k)(1+x_m)}{4} + b_1,$$

$$a_3 \frac{1+x_k}{2} + a_4 \frac{(1-x_k)(1+x_m)}{4} + b_2) |a_1 a_4 - a_2 a_3| \frac{1-x_k}{8}$$

As for the linear solver, it's still $A\backslash b$.

2.5 Error Calculation of Different Norms

Since we use variable substitution method to integrate on 9 Gauss points, we also use it in error calculation, which means we take 9 Gauss point on each element and compute $L_1(m) = \int_{K_m} |u - u_h| dxdy$ and $L_\infty(m) = \max_{1<=k<=9} |u - u_h|(x_k, y_k)$. Therefore, $L_1 = \sum_{m=1}^{N} L_1(m), L_\infty = \max_{m=1}^{N} L_\infty(m)$

2.6 The Structure of Our Code

1. Compute_Uh_2D is to compute $u_h(x, y)$ for specific point (x, y)
2. ComputeError is to compute L_1, L_∞ error and the error value on each Gauss point, which stores in matrix img.
3. FindBoundary is a programm using EasyMesh-gdx.e to find the boundary of the mesh.
4. Gauss2D is to compute the Gauss integral on a triangle.
5. OneMesh contains the main calculation of one mesh. It returns the error of FEM method and shows a image of approximation solution of the function.
6. Compiling pg5_main will give us every thing we need for this programm. If we want to change the test function $u(x, y)$, we need to change it in pg5_main and OneMesh.

2.7 Improvement and Suggestions

Since we want to reuse some of the function in the future, our code has some redundant comtent in OneMesh, thus it costs 3 min to compute the results of gd4. If we have time or want to accerlerate our program, we need to adjust our storage method according to the specific program.

3 Results

NOTE:The test function we use is $u(x, y) = \sin(2\pi x)\sin(2\pi y)$.

Table 1: Accuracy test for function error in P^1 space

gd	$\|\|u - u_h\|\|_{L^1}$ error	order	$\|\|u - u_h\|\|_{L^\infty}$ error	order
gd0	1.852102341734e-01	–	6.140482671075e-01	–
gd1	1.083692416296e-01	0.936044296105	3.244540632978e-01	1.114158720258
gd2	3.445800324552e-02	1.868493732486	1.019744593885e-01	1.887440460578
gd3	8.418080124842e-03	2.018785825754	2.540512530024e-02	1.990727634811
gd4	2.163056641743e-03	1.976988733079	6.019583798315e-03	2.094941418651

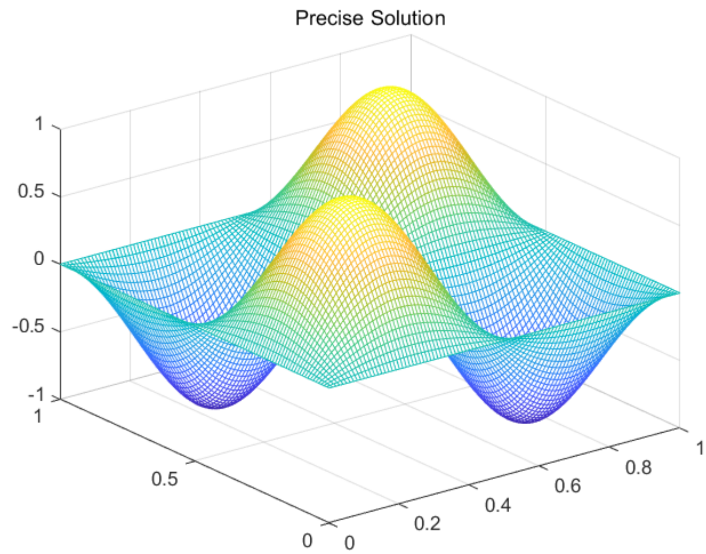


Figure 2: $u(x, y) = \sin(2\pi x)\sin(2\pi y)$

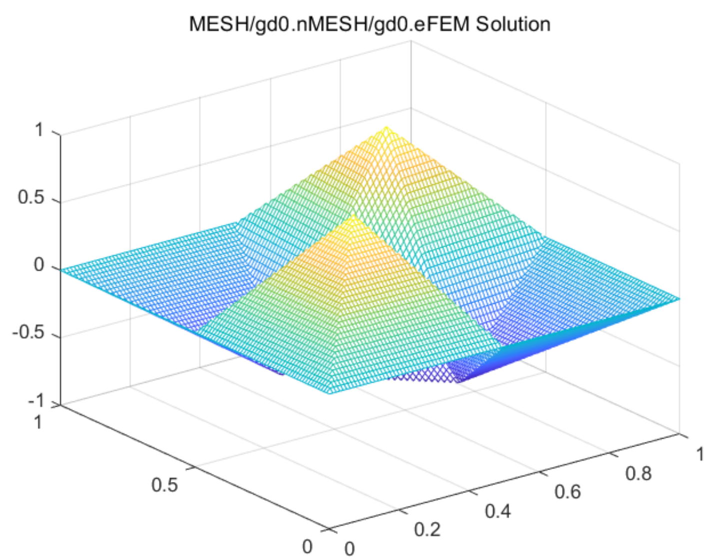


Figure 3: gd0-FEM solution

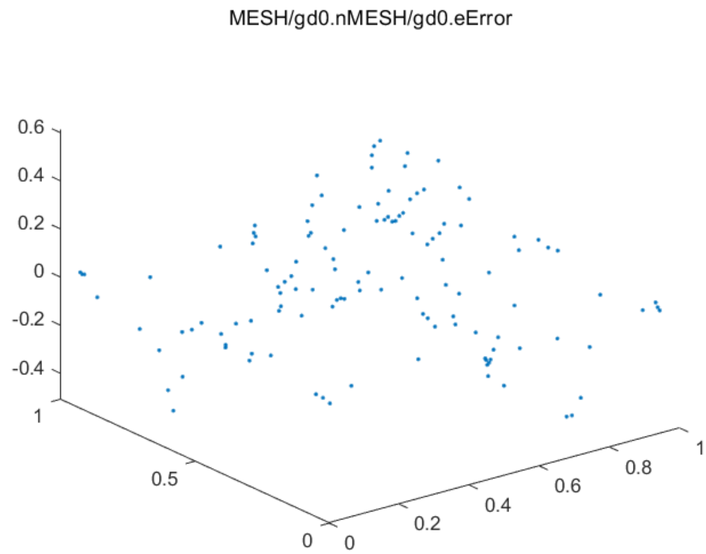


Figure 4: gd0-FEM solution error

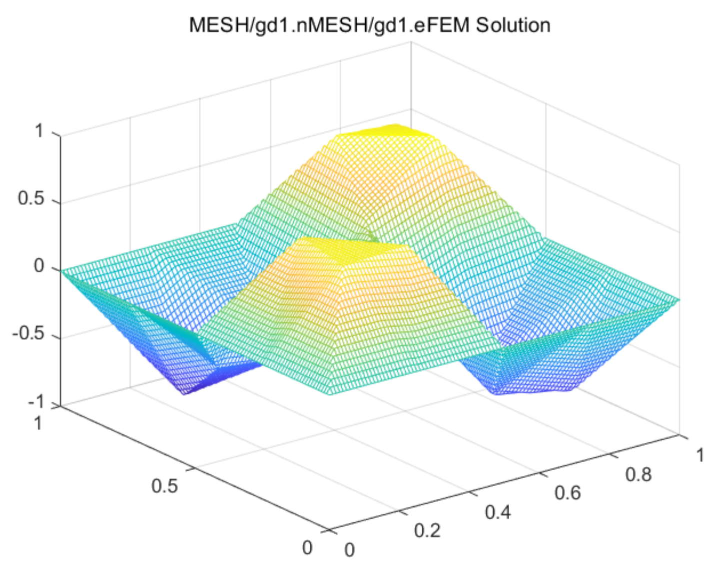


Figure 5: gd1-FEM solution error

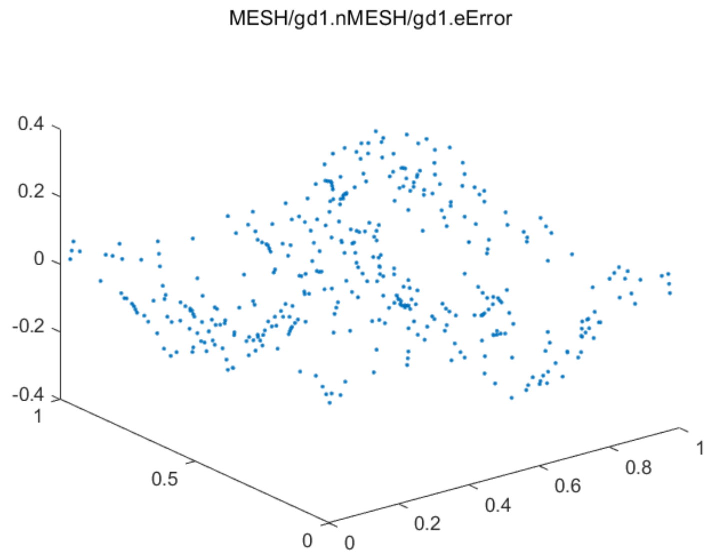


Figure 6: gd1-FEM solution error

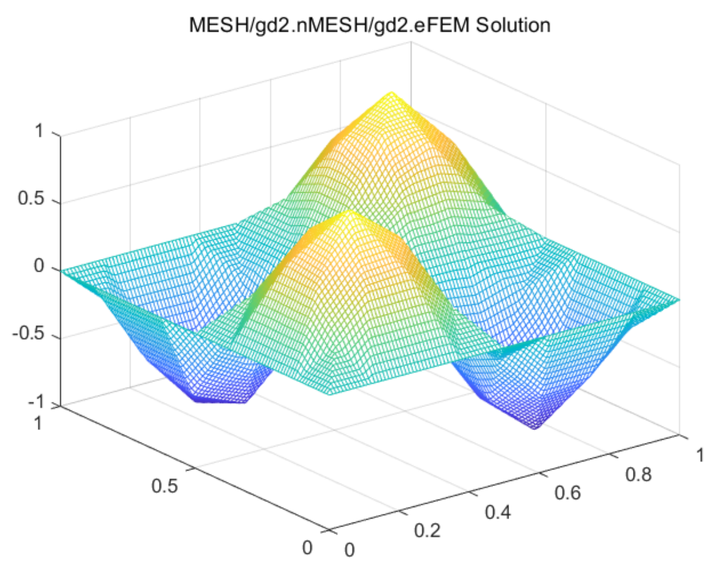


Figure 7: gd2-FEM solution

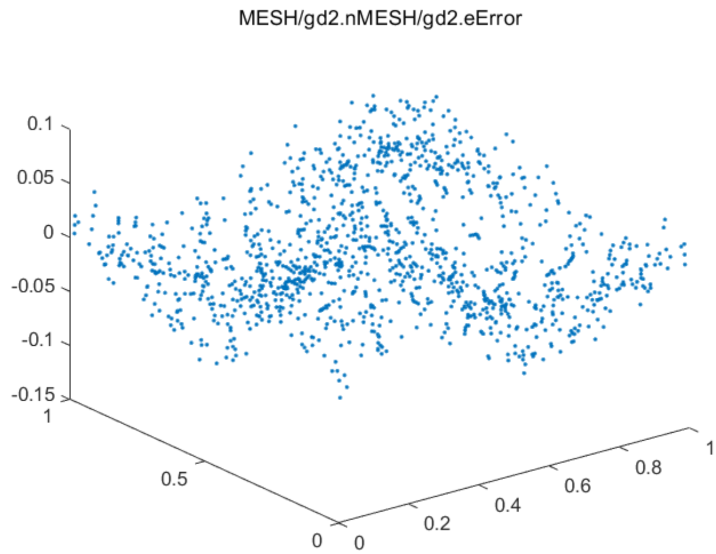


Figure 8: gd2-FEM solution error

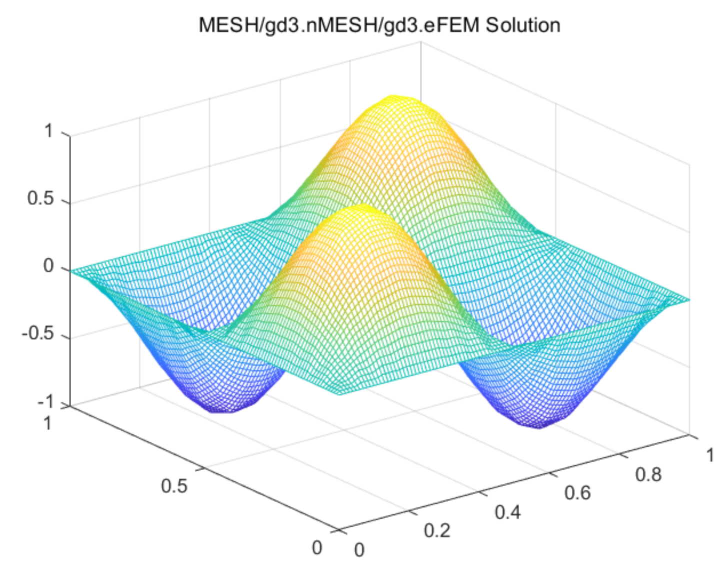


Figure 9: gd3-FEM solution

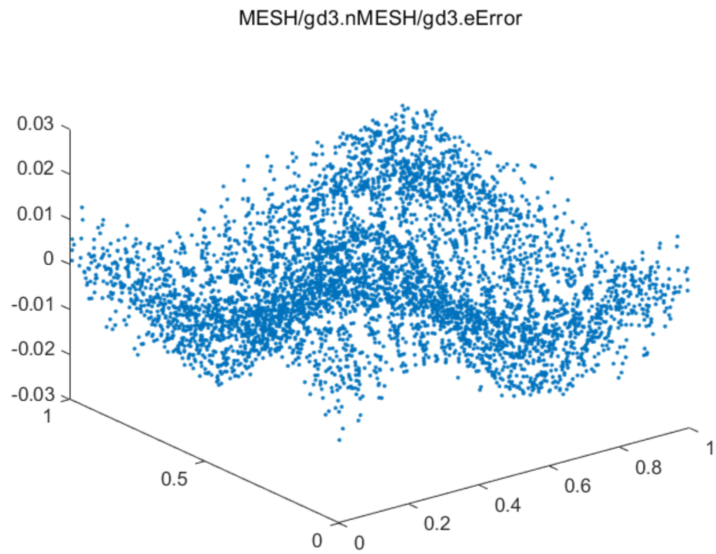


Figure 10: gd3-FEM solution error

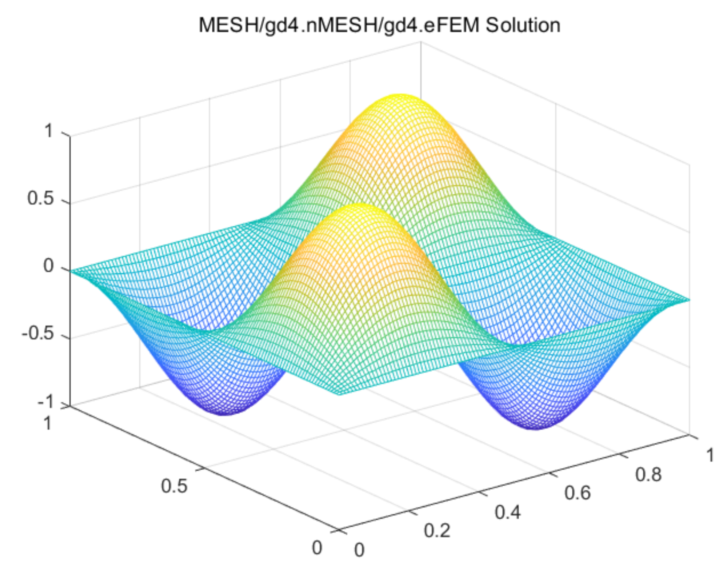


Figure 11: gd4-FEM solution

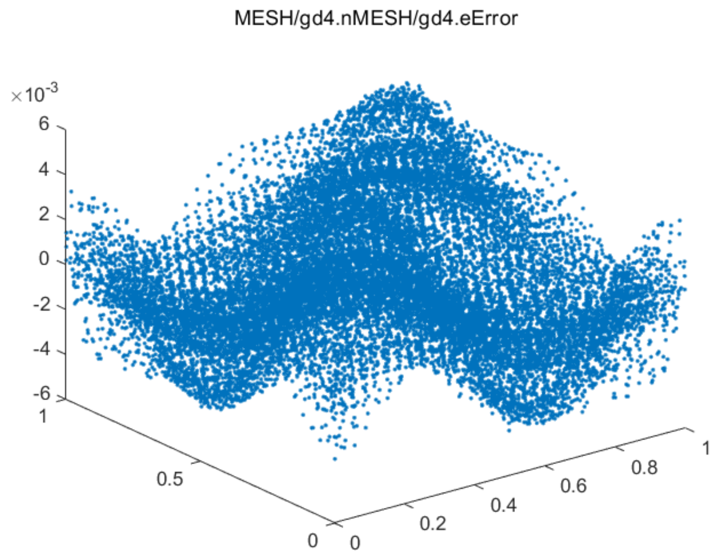


Figure 12: gd4-FEM solution error

4 Conclusions

The order is 2 according to the error table. Since the function we choose is a little complicated in$[0, 1] * [0, 1]$, the error may not seem small enough until gd4.