FEM Homework Report

Yue Wu[*]

September 15, 2022

1 Introduction

Make a program to solve the two-point boundary value problem:

$$\begin{cases} -u'' = f, & 0 < x < 1 \\ u(0) = u(1) = 0 \end{cases} \tag{1}$$

Use an equidistant mesh and a piecewise quadratic polynomial space V_h as the finite element space. Use $f(x) = -2\cos x + (x-1)\sin x,\ u(x) = (x-1)\sin x$ to test your program, and compute the following errors:

$$\|u - u_h\|_{L^2[0,1]}, \quad \|u - u_h\|_{H^1[0,1]} \tag{2}$$

2 Method

2.1 The Galerkin approximation

We use the following equidistant mesh:

$$x_j = jh,\ j = 0, 1, \cdots, N, \quad I_j = [x_{j-1}, x_j], \quad h_j = x_j - x_{j-1}, \quad h = \max_j h_j \tag{3}$$

and we use the following finite element space:

$$V_h = \left\{ v \in C[0,1] \middle| v|_{I_j} \in \mathcal{P}^2(I_j), j = 1, \cdots, N,\ v(0) = v(1) = 0 \right\} \tag{4}$$

The Galerkin approximation to the problem using approximation space V_h is given by the following:

$$\begin{aligned} &\text{find } u_h \in V_h, \text{ such that:} \\ &\forall v_h \in V_h,\ \left(u_h', v_h'\right) = (f, v_h) \end{aligned} \tag{5}$$

To approximate integrals of non-polynomial functions **in computing the right-hand-side term $\boldsymbol{f}$**, we apply the 3-point Gauss-Lobatto quadrature rule **in both cases**.

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

To approximate integrals of non-polynomial functions **in computing the errors**, we apply the 3-point Gauss-Legendre quadrature rule and the 3-point Gauss-Lobatto quadrature rule **respectively**.

Here, we give the 3-point Gauss-Legendre quadrature rule and the 3-point Gauss-Lobatto quadrature rule:

$$\int_{-1}^{1} f(x)\mathrm{d}x \approx \frac{5}{9}f(-\sqrt{\frac{3}{5}}) + \frac{8}{9}f(0) + \frac{5}{9}f(\sqrt{\frac{3}{5}}) \tag{6}$$

$$\int_{-1}^{1} f(x)\mathrm{d}x \approx \frac{1}{3}f(-1) + \frac{4}{3}f(0) + \frac{1}{3}f(1) \tag{7}$$

2.2 The global basis and matrix

The global (nodal) basis for the finite element space V_h is given by:

$$\phi_j^0(x) = \begin{cases} \frac{2}{h_j^2}(x - x_{j-\frac{1}{2}})(x - x_{j-1}), & x \in I_j \\ \frac{2}{h_{j+1}^2}(x - x_{j+\frac{1}{2}})(x - x_{j+1}), & x \in I_{j+1} \\ 0, & \text{otherwise} \end{cases}, \qquad j = 1, \cdots, N-1 \tag{8a}$$

$$\phi_j^1(x) = \begin{cases} -\frac{4}{h_j^2}(x - x_{j-1})(x - x_j), & x \in I_j \\ 0, & \text{otherwise} \end{cases}, \qquad j = 1, \cdots, N \tag{8b}$$

where we use $x_{j-\frac{1}{2}} = \frac{x_{j-1}+x_j}{2}$ to denote the element center.

We arrange these basis in the following way:

$$\phi_{2k-1}(x) = \phi_k^1(x) \qquad k = 1, \cdots, N \tag{9a}$$

$$\phi_{2k}(x) = \phi_k^0(x), \qquad k = 1, \cdots, N-1 \tag{9b}$$

thus each function $v_h \in V_h$ has the following nodal representation:

$$v_h(x) = \sum_{k=1}^{2N-1} v_h(x_{\frac{k}{2}})\phi_k(x) \tag{10}$$

In this manner, the global mass matrix is $\boldsymbol{A} = \left(\left(\phi_i', \phi_j'\right)\right)_{(2N-1)\times(2N-1)}$ with non-zero entries:

$$A_{2k-1,2k-1} = \tfrac{16}{3}h_k^{-1}, \qquad k = 1, \cdots, N \tag{11a}$$

$$A_{2k,2k} = \tfrac{7}{3}\left(h_k^{-1} + h_{k+1}^{-1}\right), \qquad k = 1, \cdots, N-1 \tag{11b}$$

$$A_{2k,2k-1} = A_{2k-1,2k} = -\tfrac{8}{3}h_k^{-1}, \qquad k = 1, \cdots, N-1 \tag{11c}$$

$$A_{2k,2k+1} = A_{2k+1,2k} = -\tfrac{8}{3}h_{k+1}^{-1}, \qquad k = 1, \cdots, N-1 \tag{11d}$$

$$A_{2k,2k+2} = A_{2k+2,2k} = \tfrac{1}{3}h_{k+1}^{-1}, \qquad k = 1, \cdots, N-2 \tag{11e}$$

2.3 The local basis and matrix

The local (nodal) shape functions within a reference element $I = [0, 1]$ are:

$$\psi^{(0)}(x) = 2(x - \frac{1}{2})(x - 1) \tag{12a}$$

$$\psi^{(1)}(x) = -4x(x - 1) \tag{12b}$$

$$\psi^{(2)}(x) = 2x(x - \frac{1}{2}) \tag{12c}$$

The local basis functions for an element I_j are:

$$\psi_j^{(k)} = \psi^{(k)}(\xi_j(x)), \quad k = 0, 1, 2, \quad j = 1, \cdots, N \tag{13}$$

with the affine transformation function

$$\xi_j(x) = \frac{x - x_{j-1}}{h_j} \tag{14}$$

The local mass matrix $\boldsymbol{K}_j \in \mathbb{R}^{(2N-1)\times(2N-1)}$ is:

$$\boldsymbol{K}_j = \text{diag}\left(\mathbf{0}_{(2j-3)}, h_j^{-1}\begin{pmatrix} \frac{7}{3} & -\frac{8}{3} & \frac{1}{3} \\ -\frac{8}{3} & \frac{16}{3} & -\frac{8}{3} \\ \frac{1}{3} & -\frac{8}{3} & \frac{7}{3} \end{pmatrix}, \mathbf{0}_{2N-2j-1}\right), \quad j = 1, \cdots, N \tag{15}$$

and the local right-hand-side $\boldsymbol{F}_j \in \mathbb{R}^{2N-1}$ is:

$$\boldsymbol{F}_j = \begin{pmatrix} \mathbf{0}_{2j-3} \\ \int_{I_j} f(x)\psi_j^0(x)\mathrm{d}x \\ \int_{I_j} f(x)\psi_j^1(x)\mathrm{d}x \\ \int_{I_j} f(x)\psi_j^2(x)\mathrm{d}x \\ \mathbf{0}_{2N-2j-1} \end{pmatrix}, \quad j = 1, \cdots, N \tag{16}$$

3 Results

We use $N = 10, 20, 40, 80$ equidistant elements for computation. By running the scripts in `main.m`, the outputs from the `matlab` program we coded are given as following:

N	L^2 error	order	H^1 error	order
10	6.7256e-06	-	5.1847e-04	-
20	8.3827e-07	3.0042	1.2971e-04	1.9990
40	1.0471e-07	3.0010	3.2433e-05	1.9997
80	1.3086e-08	3.0003	8.1085e-06	1.9999

Table 1: Table of error and order using Gauss-Legendre rule

N	L^2 error	order	H^1 error	order
10	7.2248e-07	-	8.1970e-04	-
20	4.5126e-08	4.0009	2.0508e-04	1.9989
40	2.8199e-09	4.0002	5.1280e-05	1.9997
80	1.7620e-10	4.0004	1.2821e-05	1.9999

Table 2: Table of error and order using Gauss-Lobatto rule

4 Discussion

From this experiement, we verified the theoretical 3-nd order convergence under the L^2 norm and 2-st order convergence under the H^1 norm for piecewise quadratic conforming FEM numerically. In the algorithm, we used the 3-point Gauss-Lobatto 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 is 4-th order accurate.

There is an interesting fact that from the error table resulting from different quadrature rules, we observe different convergence rates in the L^2 norm. This is a typical FEM superconvergence phenomenon. Although the convergence rates of the H^1 error are the same, we obvserve bigger error when using the Gauss-Lobatto quadrature rule. The reason may be that the 3-point Lobatto rule is less accurate that the 3-point Legendre rule.