Report on FEM Programming Homework 1&2

Yuejie Liu PB18010470

September 30, 2021

1 Problem

1.1 Program 1

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

$$V_h = \left\{ v \in C^0([0,1]), v|_{I_j} \in P^1(I_j), v(0) = v(1) = 0 \right\}$$

Test your code with problem $u(x) = (x-1)\sin(x)$, check error with N=10,20,40,80 and order.

1.2 Program 2

We have found a basis of V_h below

$$V_h = \left\{ v \in C^0([0,1]), v|_{I_j} \in P^2(I_j), v(0) = v(1) = 0 \right\}$$

and we are asked to use it to solve the problem in program 1.

2 Algorithm

2.1 Difference Scheme of PDE and FEM

The difference scheme of PDE is

$$-\frac{u_{i+1}^n - 2u_i^n + _{i-1}^n}{h^2} = f_i, where f_i = f(x_i), i = 1, ..., N-1$$

. The difference scheme of FEM is

$$KU = F$$

where K is a (kN-1)*(kN-1) stiffness matrix (k is the degree of polynomial basis), U is a (kN-1)*1 vector of the function value on grid points, and F is a (kN-1)*1 vector of (f, ϕ_i)(ϕ_i are the basis functions)

2.2 The basis of V_h

The basis of V_h of program 1 is

$$\phi_j(x) = \begin{cases} \frac{x-x_{j-1}}{x_j-x_{j-1}}, & [x_{j-1}, x_j] \\ \frac{x_{j+1}-x}{x_{j+1}-x_j}, & [x_j, x_{j+1}] \\ 0, & elsewhere \end{cases} \quad j = 1, ..., N-1$$

The basis of V_h of program 2 is

$$\varphi_j(x) = \begin{cases} \frac{(2x-x_j-x_{j-1})(x-x_{j-1})}{h^2}, & x \in (x_{j-1}, x_j) \\ \frac{(2x-x_j-x_{j+1})(x-x_{j+1})}{h^2}, & x \in [x_j, x_{j+1}) \\ 0, & x \notin (x_{j-1}, x_{j+1}) \end{cases} \quad , \quad j = 1, \cdots, N-1.$$

$$\psi_{i+\frac{1}{2}}(x) = \begin{cases} \frac{4(x-x_i)(x_{i+1}-x)}{h^2}, & x \in (x_i, x_{i+1}) \\ 0, & x \notin (x_i, x_{i+1}) \end{cases} \quad , \quad i = 0, \cdots, N-1.$$

2.3 Stiffness Matrix

The stiffness matrix of Program 1 is $K = (k_{ij})_{(N-1)*(N-1)}$, where $k_{ij} = (\phi_j^{'}, \phi_i^{'})$. Since we use even partition $h = \frac{1}{N}$,

$$K = \frac{1}{h} \begin{bmatrix} 2 & -1 & & \\ -1 & \ddots & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix}$$

As for the stiffness matrix of Program 2, we first arrange the basis functions $\Phi_j = \varphi_j, j = 1, ..., N-1, \Phi_{i+N} = \psi_{i+\frac{1}{2}}, i = 1, ..., N-1$. Then we define $K = (k_{ij})_{(2N-1)*(2N-1)}$, where $k_{ij} = (\Phi_j^{'}, \Phi_i^{'})$, and use Simpson rule to compute the integrations(because it is accurate for degree-3 polynomials).Thus we have

$$K = \frac{1}{3h} \left\{ \begin{array}{ccccccccccc}
14 & 1 & & & & -8 & -8 & & & & \\
1 & 14 & 1 & & & & -8 & -8 & & & \\
 & \ddots & \ddots & \ddots & & & & \ddots & \ddots & & \\
 & & 1 & 14 & 1 & & & & -8 & -8 & \\
 & & & 1 & 14 & & & & & -8 & -8 \\
-8 & & & & & 16 & & & & & \\
-8 & -8 & & & & & 16 & & & & \\
 & \ddots & \ddots & & & & & \ddots & & & \\
 & & \ddots & \ddots & & & & & \ddots & & \\
 & & & -8 & -8 & & & & & \ddots & \\
 & & & & -8 & & & & & & 16
\end{array} \right\}$$

2.4 Numerical Integration Method and Linear Equations Solution Method

In program 1, we use trapezoidal rule to calculate integrations on each I_j and sum them up if the function has non-zero value on multiple I_j; while in program 2 we use Simpson rule to compute integrations since it is accurate for degree-3 polynomials. This time we have no time to write specific linear equation solution method and just use $A\backslash b$.

2.5 Error Calculation of Different Norms

We take 320 evenly distributed points in our programs to test the error between precise solution and numerical solution. And the continuous norms are substituted by discrete norm, which are

$$||u - u_h||_1 = \sum_{i=1}^{320} |u_i - u_{hi}|$$

$$||u - u_h||_\infty = \max_{1 \le i \le 320} |u_i - u_{hi}|$$

2.6 The Structure of Our Code

pg1.m is the main program of the first homework and pg2 is the second. Compute_Uh_1 and Compute_Uh_ are functions for computing the function value of finite element solution u_h, because we use different basis in each program. Other details are quite specific in the annotation of our program.

2.7 Improvment and Difficulties

At first, we wanted to try the assembly method taught in the class; however, we found it troublesome to compute the integrations and numerical solution by local basis and have not come up with a general program for any P^k space.

3 Results

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

N	$\|\|u-u_h\|\|_{L^1}$ error	order	$\|\|u-u_h\|\|_{L^\infty}$ error	order
N=10	0.417160549844	–	0.002483537948,	–
N=20	0.104766405824	1.993426551652	0.000622879852	1.995370978231
N=40	0.026449757911	1.985849756609	0.000155980997	1.997583629625
N=80	0.006565937764	2.010181537469	0.000039028623	1.998765811860

Table 2: Accuracy test for function error in P^2 space

N	$\|\|u-u_h\|\|_{L^1}$ error	order	$\|\|u-u_h\|\|_{L^\infty}$ error	order
N=10	0.001956239857	–	0.000019859879	–
N=20	0.000240489475	3.024037611603	0.000002479809	3.001555740266
N=40	0.000028617735	3.070992381969	0.000000306918	3.014302915502
N=80	0.000002865164	3.320219633045	0.000000038442	2.997092999785

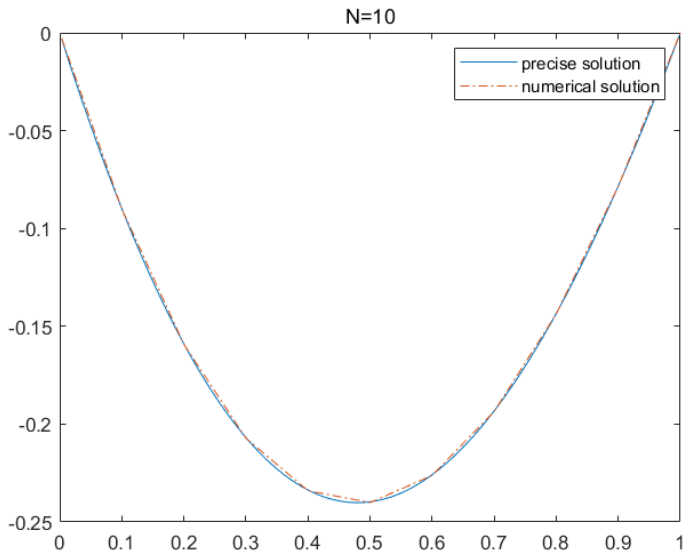

Figure 1: Program 1,N=10

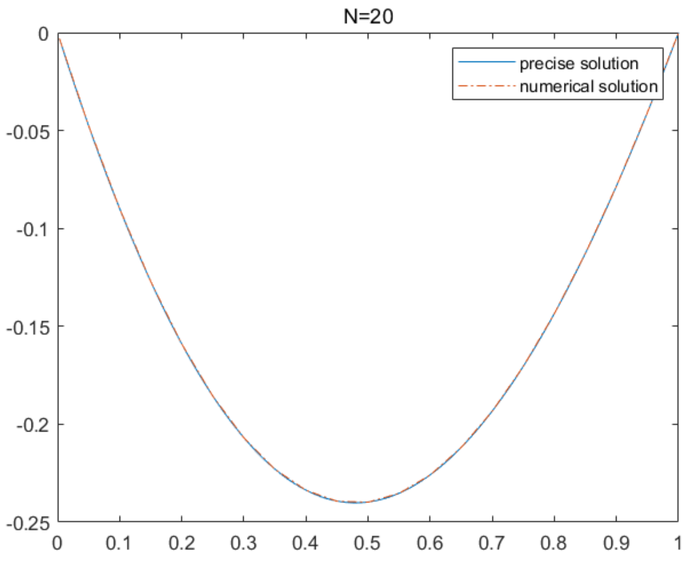

Figure 2: Program 1,N=20

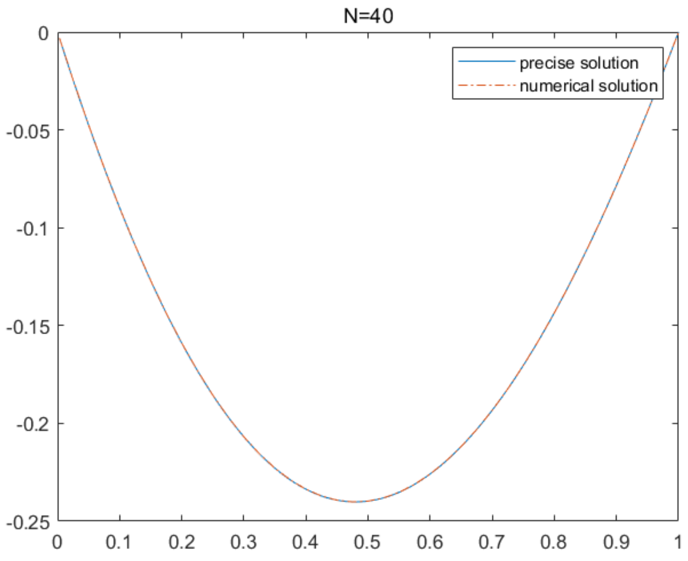

Figure 3: Program 1,N=40

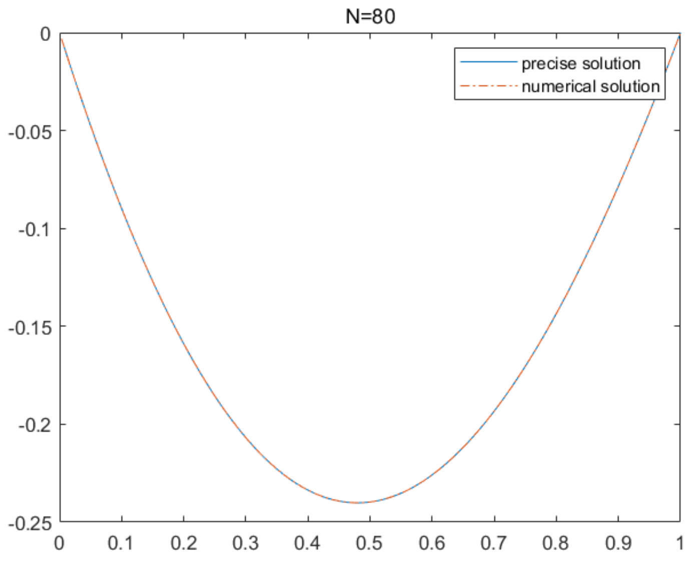

Figure 4: Program 1,N=80

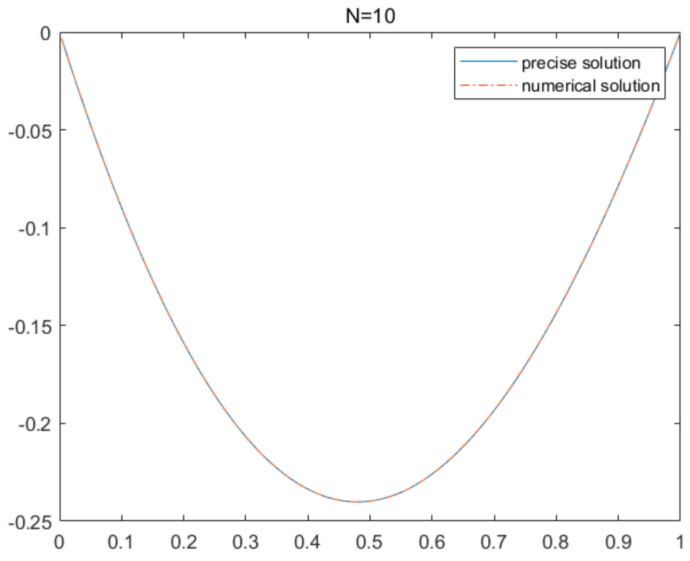

Figure 5: Program 2,N=10

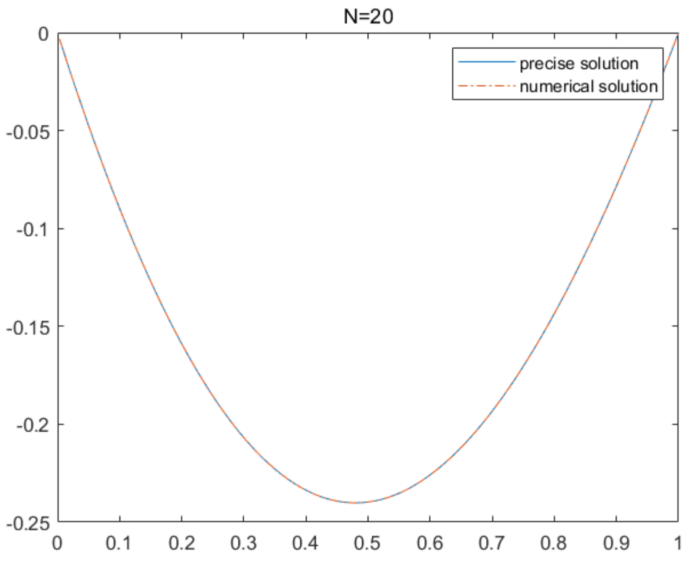

Figure 6: Program 2,N=20

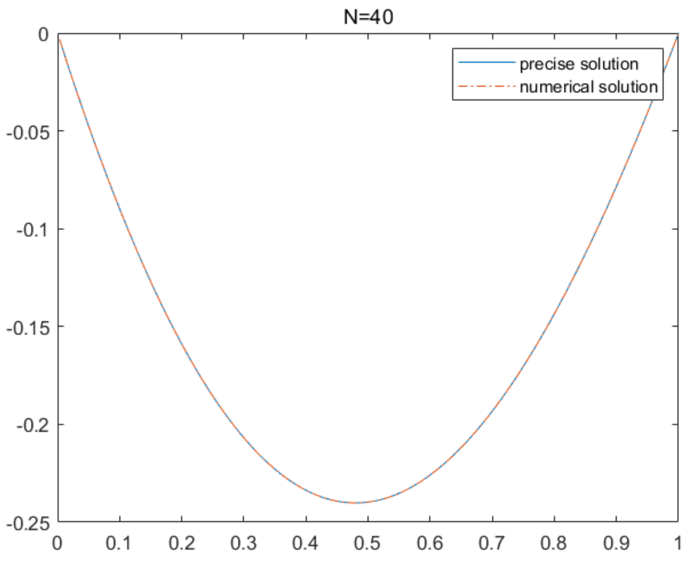

Figure 7: Program 2,N=40

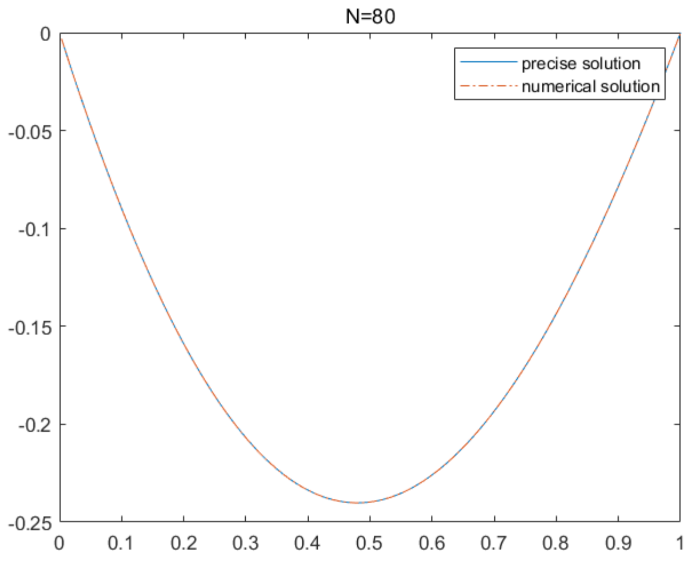

Figure 8: Program 2,N=80

4 Conclusions

The order of P^1 space is 2 and the order of P^2 space is 3, no matter $L_1 or L_\infty$ norm, which are the same as theoretical values.
Generally, P^2 space is better than P^1 space when N is relatively small. However, we cannot elimate the influence of our integration method when compute F.