

FEM Code Report 1

SA24229016 王润泽

2024 年 9 月 14 日

1 Introduction

编写程序求解以下两点边值问题 (1):

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

其中取 $f(x) = -(2 \cos x - (x - 1) \sin x)$, 已知其解析解为 $u(x) = (x - 1) \sin x$ 。

2 Method

给定双线性形式 $a(u, v)$ 和内积 (f, g) , 定义如下:

$$\begin{aligned} a(u, v) &= \int_0^1 u' v' dx, \\ (f, g) &= \int_0^1 f \cdot g dx. \end{aligned}$$

由此, 问题 (1) 转变为变分问题 (2):

寻找 $u \in \mathcal{V} = \{v \in C[0, 1], v(0) = v(1) = 0\}$, 使得对所有 $v \in \mathcal{V}$, 均有:

$$a(u, v) = (f, v). \tag{2}$$

实验中采用等距网格划分, 节点数为 $N + 2$, 在每个节点处的函数值记为 u_i , 且已知 $u_0 = 0, u_{N+1} = 0$, 网格步长为 $h = 1/N$ 。选取基函数 φ_i , 并通过这些基函数所张成的有限维线性空间 $V_h = \text{span}\{\varphi_i\}$ 进行求解。此时得到原问题的离散形式解为: $u_h = \sum_{i=1}^N u_i \varphi_i$, 其中 u_i 为待求解的系数, 而 $v_h = \sum_{i=1}^N v_i \varphi_i$ 。

此时变分问题的离散形式为问题 (3):

$$\begin{aligned} a(u_h, v_h) &= \sum_{i=1}^N \sum_{j=1}^N u_i a(\varphi_i, \varphi_j) v_j \\ (f, v_h) &= \sum_{i=1}^N f_i v_i \end{aligned} \quad (3)$$

实验中基函数定义如下:

$$\varphi_i(x) = \begin{cases} \frac{x-x_{i-1}}{x_i-x_{i-1}}, & x \in [x_{i-1}, x_i), \\ \frac{x_{i+1}-x}{x_{i+1}-x_i}, & x \in [x_i, x_{i+1}], \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

得到公式 (3) 中刚度矩阵各项:

$$a(\varphi_i, \varphi_j) = \begin{cases} -\frac{1}{h}, & |i-j| = 1 \\ \frac{2}{h}, & i = j \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

因此, 刚度矩阵为:

$$A = \frac{1}{h} \begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \cdots & 0 \\ 0 & -1 & 2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 2 \end{bmatrix} \quad (6)$$

同理, 根据公式 (4), 得到对各个基函数与 f 的内积, 即荷载向量各项为:

$$f_i = (f, \varphi_i) = \int_0^1 f \varphi_i dx = 4(hj-1) \sin^2(h/2) \sin(hj)/h - 2 \sin(h) \cos(hj) \quad (7)$$

以上, 得到变分问题的离散形式为, 对任意的 $V = (v_1, v_2, \dots, v_N)^T$, 有:

$$V^T A U = V^T F \quad (8)$$

进一步, 只要求解出 $AU = F$ 即可得到 U , 即为问题 (1) 的数值解, 其中:

$$\begin{aligned} U &= (u_1, u_2, \dots, u_N)^T, \\ F &= (f_1, f_2, \dots, f_N)^T. \end{aligned} \quad (9)$$

3 Results

3.1 数值拟合效果

通过调整 $N = \{10, 20, 40, 80\}$ 的值，得到不同的数值解，与解析解进行比较，如图 1 所示。

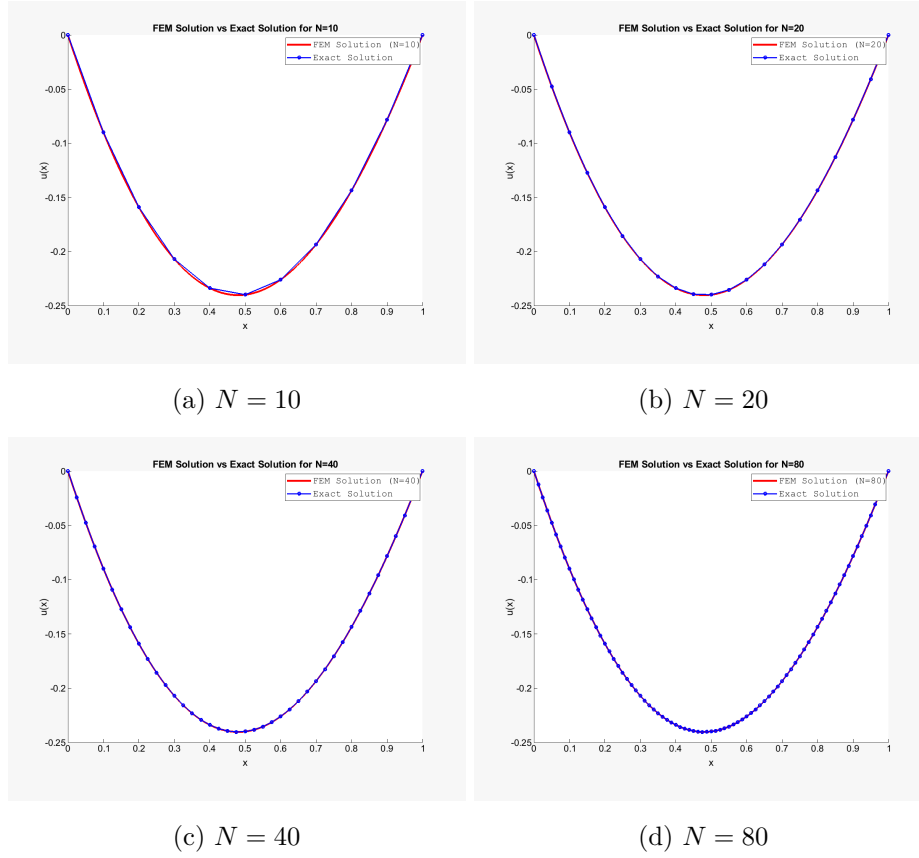


图 1: 北大天元绘制的拟合图像

3.2 误差分析

由于已知其解析解为：

$$u(x) = (x - 1) \sin x. \quad (10)$$

完成数值求解后，使用 L^2 范数和 H^1 范数计算误差，对结果进行讨论。

L^2 范数的定义为:

$$\|e\|_{L^2} = \left(\int_0^1 (u(x) - u_h(x))^2 dx \right)^{1/2}, \quad (11)$$

H^1 范数的定义为:

$$\|e\|_{H^1} = \left(\int_0^1 ((u(x) - u_h(x))^2 + (u'(x) - u'_h(x))^2) dx \right)^{1/2}. \quad (12)$$

其中, 实验中利用差分计算导数。

得到如下结果:

表 1: 误差分析

N	L^2 error	order	H^1 error	order
10	1.70×10^{-3}	-	5.39×10^{-2}	-
20	4.26×10^{-4}	2.00	2.69×10^{-2}	1.00
40	1.07×10^{-4}	2.00	1.34×10^{-2}	1.00
80	2.66×10^{-5}	2.00	6.68×10^{-3}	1.01

4 Discussion

本节着重进行误差分析, 令 $v \in V_h$ 为一分片线性函数, 满足:

$$v(x_i) = u(x_i) \quad \forall i = 1, 2, \dots, N \quad (13)$$

其中 u 是两点边值问题的古典解。

令 $e = v - u$, 那么考虑区间 $[a, b]$ 上的 C^1 函数 f , 满足 $f(a) = f(b) = 0$, 那么 $|f| < \frac{b-a}{2} \max |f'|$, 积分

$$\int_a^b |f(x)| dx < \frac{b-a}{2} \max |f'| \quad (14)$$

结合上式和微分中值定理可得, 对于 e, e' 有:

$$\begin{aligned} \sup_x |e'| &\leq C_1 h \sup_x |u''| \\ \sup_x |e| &\leq C_2 h^2 \sup_x |u''| \end{aligned} \quad (15)$$

那么:

$$\begin{aligned} \|u - v\|_2 &\leq C_1 h^2 \sup_x |u''| \\ \|u' - v'\|_2 &\leq C_2 h \sup_x |u''| \end{aligned} \quad (16)$$

因此, L^2 误差和 H^1 误差的阶数分别为 2 和 1, 和实验结果一致。

A Code

```

1 function [x, u_h] = fem_solver(N,F_load)
2     % Mesh generation
3     a = 0; b = 1; % Interval [0,1]
4     h = (b - a) / N; % Step size
5     x = linspace(a, b, N+1); % Mesh points
6     % Preallocate index and value arrays for the sparse matrix
7     I = zeros(3*N-5, 1); % Row indices
8     J = zeros(3*N-5, 1); % Column indices
9     S = zeros(3*N-5, 1); % Non-zero values
10    index = 0; % Index counter
11    % Construct the sparse matrix K
12
13    for i = 1:N-1
14        index = index + 1;
15        % Diagonal element
16        I(index) = i; J(index) = i; S(index) = 2/h;
17        if i > 1
18            % Lower diagonal element
19            index = index+1; I(index) = i; J(index) = i-1; S(index) = -1/h;
20        end
21        if i < N-1
22            % Upper diagonal element
23            index = index+1; I(index) = i; J(index) = i+1; S(index) = -1/h;
24        end
25    end
26
27    % Construct the sparse matrix K using the sparse function
28    K = sparse(I, J, S, N-1, N-1);
29    % Construct the load vector F

```

```

30     F = zeros(N-1, 1);
31     for j = 1:N-1
32         F(j) = F_load(j,h);
33     end
34     % Solve the linear system K * u_h = F
35     u_h = K \ F;
36     % Apply boundary conditions u(0) = 0 and u(1) = 0
37     u_h = [0, u_h', 0];
38 end

```

Listing 1: FEM Solver

```

1  % Main script: Call FEM solver and plot results for different N values
2  Ns = [10, 20, 40, 80]; % Different values for N
3  u_exact = @(x) (x - 1) .* sin(x); % Exact solution
4  u_exact_der = @(x) sin(x)+(x-1).*cos(x);
5  % Load function integrate f*phi
6  F_load = @(i,h) 4*(h*i - 1) * sin(h/2)^2 * sin(h*i)/h - 2*sin(h) * cos(h*i);
7
8  num = 10000;
9  delta_x = 1/num;
10 x = linspace(0,1,num);
11 u_exact_values = u_exact(x);
12 L2_error_list = zeros(1,4);
13 L2_diff_error_list = zeros(1,4);
14
15 for i = 1:length(Ns)
16     N = Ns(i);
17     % Call FEM solver to get mesh points and FEM solution
18     [x_sample, u_sample] = fem_solver(N,F_load);
19     % Calculate the exact solution at mesh points
20     u_h = interp1(x_sample,u_sample, x);
21     % Compute the L2 norm error
22     L2_error = sqrt(sum((u_h - u_exact_values).^2)*delta_x);
23     u_diff_h = diff(u_h)/delta_x;
24     u_diff_exact = diff(u_exact_values)/delta_x;
25     L2_diff_error = sqrt(sum((u_diff_h - u_diff_exact).^2)*delta_x);
26     L2_diff_error = sqrt(L2_error^2 + L2_diff_error);
27     % Print the error

```

```

28     fprintf('L2 norm error with N=%d: %e\n', N, L2_error);
29     fprintf('LH1 norm error with N=%d: %e\n', N, L2_diff_error);
30     L2_error_list(i) = L2_error;
31     L2_diff_error_list(i) = L2_diff_error;
32 end
33
34 for i = 2:length(Ns)
35     N_now = Ns(i);
36     N_old = Ns(i-1);
37     order_l2 = log(L2_error_list(i-1)/L2_error_list(i))/log(N_now/N_old);
38     order_l2_diff = log(L2_diff_error_list(i-1)/L2_diff_error_list(i))/log(
        N_now/N_old);
39     fprintf("N:%d, L2 order:%e, LH1 order:%e\n",N_now,order_l2,order_l2_diff);
40 end

```

Listing 2: Main Code