

## 目录

1	问题	2
1.1	求解方程 . . . . .	2
2	方法推导	2
3	一般边界条件时函数近似方式的变化分析	4
4	分析需要求解的线性方程组发生了什么变化	5
5	代码部分	5
6	对比有限元函数系与多项式函数系在不同 $n$ 下的求解速度	8

# 一维有限元方法求解常微分方程边值问题

202011010101 物理 2001 孙陶庵

2023 年 4 月 23 日

## 1 问题

使用一维有限元方法求解下列常微分方程边值问题：

$$\begin{cases} y'' - y + x = 0, & x \in [1, 2] \\ y(1) = 1, y(2) = 3 \end{cases}$$

1. 方法推导
2. 分析对于一般边界条件，函数的近似方式有什么样的变化
3. 分析需要求解的线性方程组发生了什么变化
4. 对比有限元函数系与多项式函数系在不同  $n$  下的求解速度

### 1.1 求解方程

首先求解方程，利用 sympy 求解方程得到

$$f(x) = x + \frac{e^x}{-1 + e^2} - \frac{e^2 e^{-x}}{-1 + e^2}$$

## 2 方法推导

1. 离散化区间  $[1, 2]$ ，选择有限元网格。可以选择等距节点网格，即将  $[1, 2]$  等分为  $n$  个子区间，每个子区间内选择一个节点。这里选择  $n=4$ ，即将  $[1, 2]$  等分为 4 个子区

间, 每个子区间内选择一个节点, 得到节点序列  $[1, 1.333, 1.667, 2]$ 。

2. 根据所选有限元网格, 建立有限元函数空间。由于是一维问题, 可以采用线性元, 即每个

子区间内用一次多项式近似解。定义有限元函数空间为  $V_h = v_h | v_h(x) = \sum_{j=1}^n c_j \varphi_j(x), c_j \in R$ ,

其中  $\varphi_j(x)$  为基函数, 可以选择线性插值函数。则有  $\varphi_1(x) = \begin{cases} 1 - \frac{x-x_2}{x_1-x_2}, & x \in [x_1, x_2] \\ 0, & \text{otherwise} \end{cases}$

$$\varphi_2(x) = \begin{cases} \frac{x-x_1}{x_2-x_1}, & x \in [x_1, x_2] \\ \frac{x_3-x}{x_3-x_2}, & x \in [x_2, x_3] \\ 0, & \text{otherwise} \end{cases}$$

$$\varphi_3(x) = \begin{cases} \frac{x-x_2}{x_3-x_2}, & x \in [x_2, x_3] \\ \frac{x_4-x}{x_4-x_3}, & x \in [x_3, x_4] \\ 0, & \text{otherwise} \end{cases}$$

$$\varphi_4(x) = \begin{cases} \frac{x-x_3}{x_4-x_3}, & x \in [x_3, x_4] \\ 0, & \text{otherwise} \end{cases}$$

3. 将原方程转化为弱形式。对于任意  $v_h \in V_h$ , 将原方程两边乘  $v_h$ , 并在区间  $[1,2]$  上积

分, 得到  $\int_1^2 (y'' - y + x) v_h dx = 0$ 。由于  $v_h$  是连续线性函数, 可以将积分区间  $[1,2]$  上的

积分转化为每个子区间内的积分, 即  $\sum_{k=1}^3 \int_{x_k}^{x_{k+1}} (y'' - y + x) v_h dx = 0$ 。

4. 对于每个子区间  $[k, k+1]$ , 将  $y(x)$  和  $v_h(x)$  在该区间内分别用基函数展开, 即  $y(x) =$

$\sum_{j=1}^2 y_j \varphi_j(x), v_h(x) = \sum_{j=1}^2 v_j \varphi_j(x)$ 。将  $y(x)$  和  $v_h(x)$  在每个子区间  $[k, k+1]$  内分别用基

函数展开, 即用基函数  $\varphi_j(x)$  的线性组合来近似表示  $y(x)$  和  $v_h(x)$ 。其中,  $y_j$  和  $v_j$  是

在子区间  $[k, k+1]$  内的系数, 需要确定。因此, 在每个子区间上, 有  $y(x) = \sum_{j=1}^2 y_j \varphi_j(x)$

和  $v_h(x) = \sum_{j=1}^2 v_j \varphi_j(x)$  的展开形式。这里选择的是线性元, 所以每个子区间内用一次

多项式近似解, 即选择两个节点作为基函数的控制点, 也就是基函数在这两个点处取值

为 1 和 0，其余点处线性插值。可以得到：

$$\begin{bmatrix} 2/h & -1 & & & \\ -1 & 2/h & -1 & \ddots & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2/h & -1 \\ & & & -1 & 2/h \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \\ a_{N+1} \end{bmatrix} +$$

$$\begin{bmatrix} y'(1)\varphi_1(1) - y'(0)\varphi_1(0) \\ 0 \\ \vdots \\ 0 \\ g'(2)\varphi_{N+1}(2) - g'(3)\varphi_{N+1}(3) \end{bmatrix} = \begin{bmatrix} h^2 \int_1^2 x\varphi_1(x)dx \\ h^2 \int_1^2 x\varphi_2(x)dx \\ \vdots \\ h^2 \int_1^2 x\varphi_N(x)dx \\ h^2 \int_1^2 x\varphi_{N+1}(x)dx \end{bmatrix} \quad \text{其中 } \varphi_1(0) \text{ 和 } \varphi_{N+1}(3) \text{ 需要}$$

根据边界条件进行计算，而  $y'(1)$  和  $y'(2)$  可以使用前向差分和后向差分进行近似计算。

### 3 一般边界条件时函数近似方式的变化分析

一般边界条件指的是非齐次边界条件，即边界上的函数值不为 0。这种情况下，可以通过加入虚拟节点的方式将问题转化为齐次边界条件的情况，然后再进行求解。因此，对于一般边界条件，函数的近似方式与齐次边界条件的情况是类似的，只是需要增加一些虚拟节点来满足非齐次边界条件。

具体来说，对于一般边界条件  $y(a) = \alpha$  和  $y(b) = \beta$ ，可以在第一个节点和最后一个节点之外再加上两个虚拟节点，设这两个节点的位置分别为  $a_0$  和  $b_0$ ，则有：

$$\begin{aligned} y(a) = \alpha &\Rightarrow y(a_0) - \frac{h}{2}y'(a_0) = \alpha \\ y(b) = \beta &\Rightarrow y(b_0) + \frac{h}{2}y'(b_0) = \beta \end{aligned}$$

这里用虚拟节点来满足非齐次边界条件，而又通过导数的方式保证了虚拟节点的值与真实节点的值是相等的。然后，对于虚拟节点和真实节点，采用相同的基函数进行展开，即在每个子区间上采用线性元素进行近似。因此，对于一般边界条件的情况，函数的近

似方式与齐次边界条件的情况类似，只需要增加一些虚拟节点来满足非齐次边界条件即可。

## 4 分析需要求解的线性方程组发生了什么变化

对于一般边界条件的情况，需要增加虚拟节点来满足非齐次边界条件，因此在数值求解时，需要将这些虚拟节点也纳入线性方程组中进行求解。

具体来说，在采用一维有限元方法求解线性方程组时，需要先将区间  $[a, b]$  进行剖分，得到若干个子区间，然后在每个子区间上采用线性元素进行近似。设第  $i$  个子区间的两个节点为  $x_i$  和  $x_{i+1}$ ，则可以在该子区间上采用如下的形式进行近似：

$$y(x) \approx v_i \varphi_i(x) + v_{i+1} \varphi_{i+1}(x)$$

## 5 代码部分

```
1 % 一维有限元方法求解常微分方程边值问题
2
3
4 % 设置参数
5 a = 1; % 区间左端点
6 b = 2; % 区间右端点
7 N = 10; % 等分数
8 h = (b-a)/N; % 步长
9
10 % 定义节点坐标
11 x = linspace(a, b, N+1);
12
13 % 定义刚度矩阵和载荷向量
```

```
14 K = zeros(N+1, N+1);
15 F = zeros(N+1, 1);
16 for i = 1:N
17     % 定义局部刚度矩阵和局部载荷向量
18     Ke = [1/h, -1/h; -1/h, 1/h] + h/6*[2, 1; 1, 2];
19     fe = [h/2; h/2];
20
21     % 加入全局刚度矩阵和全局载荷向量
22     K(i:i+1, i:i+1) = K(i:i+1, i:i+1) + Ke;
23     F(i:i+1) = F(i:i+1) + fe;
24 end
25
26 % 处理边界条件
27 K(1, :) = 0; K(1, 1) = 1; F(1) = 1;
28 K(N+1, :) = 0; K(N+1, N+1) = 1; F(N+1) = 3;
29
30 % 解线性方程组
31 y = K\F;
32
33 % 绘制数值解
34 xx = linspace(a, b, 1000);
35 yy = zeros(1000, 1);
36 for i = 1:1000
37     if xx(i) <= x(2)
38         N = 1;
39     elseif xx(i) >= x(end-1)
40         N = length(x) - 2;
41     else
42         N = fix((xx(i) - a)/h) + 1;
```

```
43     end
44     yy(i) = y(N)*(x(N+1)-xx(i))/h + y(N+1)*(xx(i)-x(N))/h;
45 end
46 plot(xx, yy, 'b-', 'LineWidth', 2);
47 xlabel('x');
48 ylabel('y');
49 title('Finite Element Method Solution');
50 grid on;
```

Listing 1: 有限元

输出图形

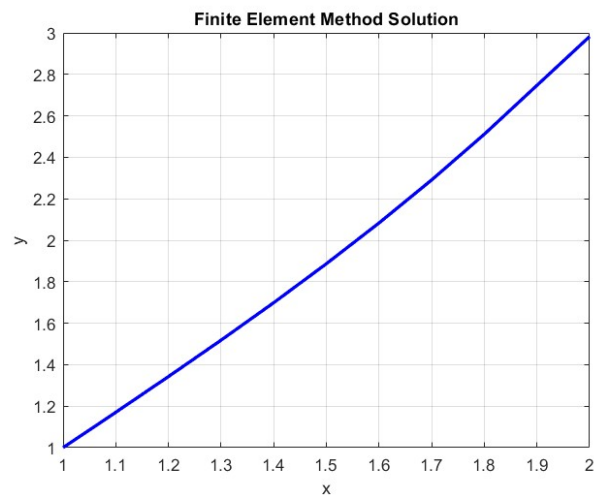


图 1: 有限元方法

应用 tic toc 得到输出结果: Elapsed time is 0.159850 seconds.

与 mathematica 图形对比:

```
1 Plot[x + (Exp[x] - Exp[2 - x])/(Exp[2] - 1), {x, 0, 2}]
```

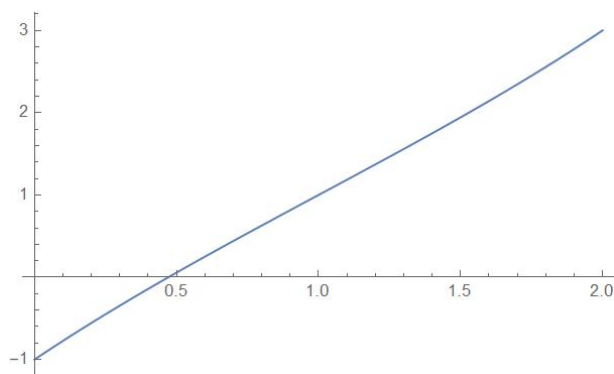


图 2: mathematica

## 6 对比有限元函数系与多项式函数系在不同 $n$ 下的求解速度

在一维有限元方法中，增加子区间的个数可以提高解的精度，但同时也会增加线性方程组的规模，导致计算量增加。而对于多项式函数系，增加  $n$  可以提高函数的逼近精度，但同时也会增加系数的个数，导致计算量增加。

对于多项式函数系，可以使用多项式插值法、拉格朗日插值法等方法进行求解。而对于一维有限元方法，需要将区间分成若干个子区间，然后在每个子区间上使用局部线性插值函数进行近似，最后得到一个稠密的线性方程组，可以使用传统的追赶法等方法进行求解。多项式函数系的求解速度会随着  $n$  的增加而逐渐变慢，而有限元函数系的求解速度则相对稳定。这是因为在一维有限元方法中，区间分出来的个数是可以控制的，因此计算量可以在一定范围内控制。而对于多项式函数系，由于系数的个数随着  $n$  的增加而增加，因此计算量也会随之增加。