Date.　　Page.

实验项目三

2017/10/26 徐[illegible]

一、理论基础

从Ritz方法出发：考虑两点边值问题 $\begin{cases} -\frac{d}{dx}\left(p\frac{du}{dx}\right)+qu=f & a<x<b \\ u(a)=0,\ u(b)=0 \end{cases}$

将[a,b]离散，$I_i[x_{i-1},x_i]$，$a=x_0<\cdots<x_n=b$，构造试探函数 $u_n(x)|_{I_i}$，$i=1\cdots n$，
对 $u(x)$ 进行区间上的线性插值，有 $u_n(x)|_{I_i}=u_{i-1}\left(\frac{x-x_i}{x_{i-1}-x_i}\right)+u_i\left(\frac{x-x_{i-1}}{x_i-x_{i-1}}\right)$
考虑 $J(u_n)=\frac{1}{2}a(u_n,u_n)-f(u_n)=\frac{1}{2}\sum_{i=1}^{n}\int_{x_{i-1}}^{x_i}[p(u_n')^2+qu_n^2]dx-\sum_{i=1}^{n}\int_{x_{i-1}}^{x_i}fu_n dx$
作变换标准化，令 $\xi=\frac{x-x_{i-1}}{h_i}$，代入 $J(u_n)$，利用多元函数极值必要条件 $\frac{\partial J}{\partial u_i}=0$
得到关于 u_i 的方程组，解出 u_i。

从Galerkin方法出发：构造节点对应的基函数 φ_i，有 $\varphi_1\cdots\varphi_n$ 线性无关，
作为 V 的一组基，对 $u_n\in V_h$：$u_n=\sum_{i=1}^{n}u_i\varphi_i(x)$，其中 $u_i=u_n(x_i)=u(x_i)$
形成有限元方程，等价变分问题，有 $\sum_{i=1}^{n}a(\varphi_i,\varphi_j)u_i=\sum_{i=1}^{n}\int_{x_{i-1}}^{x_i}f\varphi_j dx$
借助仿射变换，得到关于 u_i 的方程组，解出 u_i。

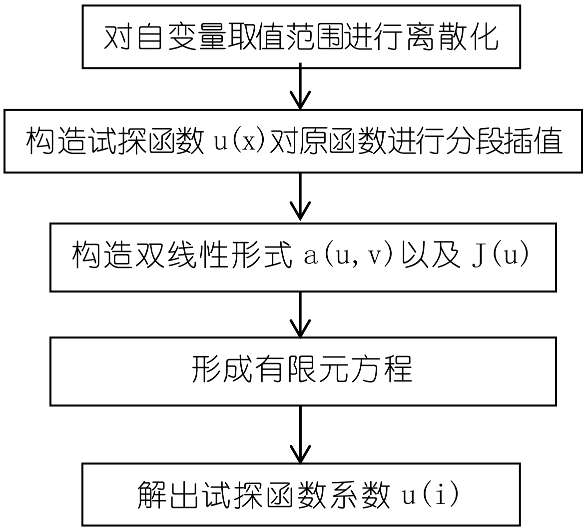
二、算法结构：见附图。

三、程序代码：见附件。

四、结果分析：图表见附件。线性元法精度高，理论基础较为完备，且当数据量较大时，求解效率有待提高。

算法结构

考虑微分方程: $\begin{cases} \dfrac{-d(p\dfrac{du}{dx})}{dx} + qu = f, a < x < b \\ u(a) = 0, u'(b) = 0 \text{ or 其他条件} \end{cases}$



代码

1.计算 $a(\varphi_i,\varphi_j)$和$a(f,\varphi_j)$

```
%计算 a(u,v) = 积分(u'v'+pi^2/4*uv)和 积分(f,φ)
function [k_cache,b_cache]= Finite_element_a_f(a,h,x,Interval_number)
%参数分别是 a 区间右端点,h 步长,x 端点矩阵,所需离散成的区间个数
k_cache = zeros(Interval_number,Interval_number);%缓存矩阵,存放每次计算得到的单刚矩阵
b_cache = zeros(Interval_number,1);%同上
syms t%构造插值基函数
fail_l = ( x(a) - t )/h;
fail_r = ( t - x(a-1) )/h;
if a == 2
    fail_l = 0;end %第一个小区间没有φ左
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%上述基函数构造代码思想:计算每个小区间上对应左右端点的插值基函数,循环覆盖
%比如 计算第一个小区间 x1 对应的插值基函数 fail1 时,x1 在第一个区间上对应的是
%一个区间右端点,此时 x1 对应基函数应该带入 fail_r 中
%但是在第二个小区间上计算新的 fail1 和 fail2 时,这时 x1 对第二个小区间的左端点
%此时 x1 对应的基函数应该带入 fail_l 中
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%构造系数矩阵的被积函数
a1 = diff(fail_l)*diff(fail_l) + pi^2/4*fail_l*fail_l;%左上
a2 = diff(fail_l)*diff(fail_r) + pi^2/4*fail_l*fail_r;%右上和左下
a3 = diff(fail_r)*diff(fail_r) + pi^2/4*fail_r*fail_r;%右下
```

```
if a==2 a1=0;a2=0;end %第一个小区间系数阵只有左上角有元素,但这个左上角元素是由 fail_r 和 fail_r 得到的
%构造解矩阵的被积函数
b1 = 2*sin(pi/2 * t) * fail_l;
b2 = 2*sin(pi/2 * t) * fail_r;
if a == 2 %第一个小区间系数矩阵和解矩阵的计算
    k_cache(1,1) = int(a3,t,x(a-1),x(a));
    b_cache(1) = int(b2,t,x(a-1),x(a));
else%其他小区间系数矩阵和解矩阵的计算
    k_cache(a-2,a-2) = int(a1,t,x(a-1),x(a));
    k_cache(a-2,a-1) = int(a2,t,x(a-1),x(a));
    k_cache(a-1,a-2) = k_cache(a-2,a-1);
    k_cache(a-1,a-1) = int(a3,t,x(a-1),x(a));
    b_cache(a-2) = int(b1,t,x(a-1),x(a));
    b_cache(a-1) = int(b2,t,x(a-1),x(a));
end
end
```

2.主程序

```
function result = Finite_element(Interval_number)%参数所需离散成的区间个数
h = 1/Interval_number;%步长
k = zeros(Interval_number,Interval_number);%刚度矩阵-系数矩阵
b = zeros(Interval_number,1);%刚度矩阵-解矩阵
Number_nodes = Interval_number + 1;%结点个数是所需区间个数+1
x(1) =0;%结点矩阵
for i = 2:Number_nodes
        x(i) = x(1)+ (i-1)*h;
end
for a = 2:Number_nodes %a=2 时,得到的是 a(φ1, φ1)
    [result1,result2] =  Finite_element_a(a,h,x,Interval_number);
    k = k + result1;
    b = b + result2;
end
result = k\b;% 解出 y
end
```

结果

分别取将区间离散为 3 段和 10 段时，得到的近似结果与真解

x(i)	0	1/3	2/3	1
n=3	0.00000000	0.20495111	0.35498574	0.40990222
真解	0.00000000	0.20264237	0.35098688	0.40528473

x(i)	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
n=10	0	0.063 46567	0.1253 6860	0.1841 8454	0.2384 6525	0.2868 74151	0.3282 19258	0.3614 82518	0.3858 44879	0.4007 06458	0.4057 01314
真解	0	0.063 40050	0.1252 3987	0.1839 9542	0.2382 2039	0.2865 79584	0.3278 82238	0.3611 11343	0.3854 48688	0.4002 95007	0.4052 84735

分别作出 n=3, 10 以及真解的图像：