

算法结构

对自变量取值范围进行离散化

构造试探函数u(x)对原函数进行分段插值

构造双线性形式a(u,v)以及J(u)

形成有限元方程

解出试探函数系数u(i)

代码

**1.计算a**

%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

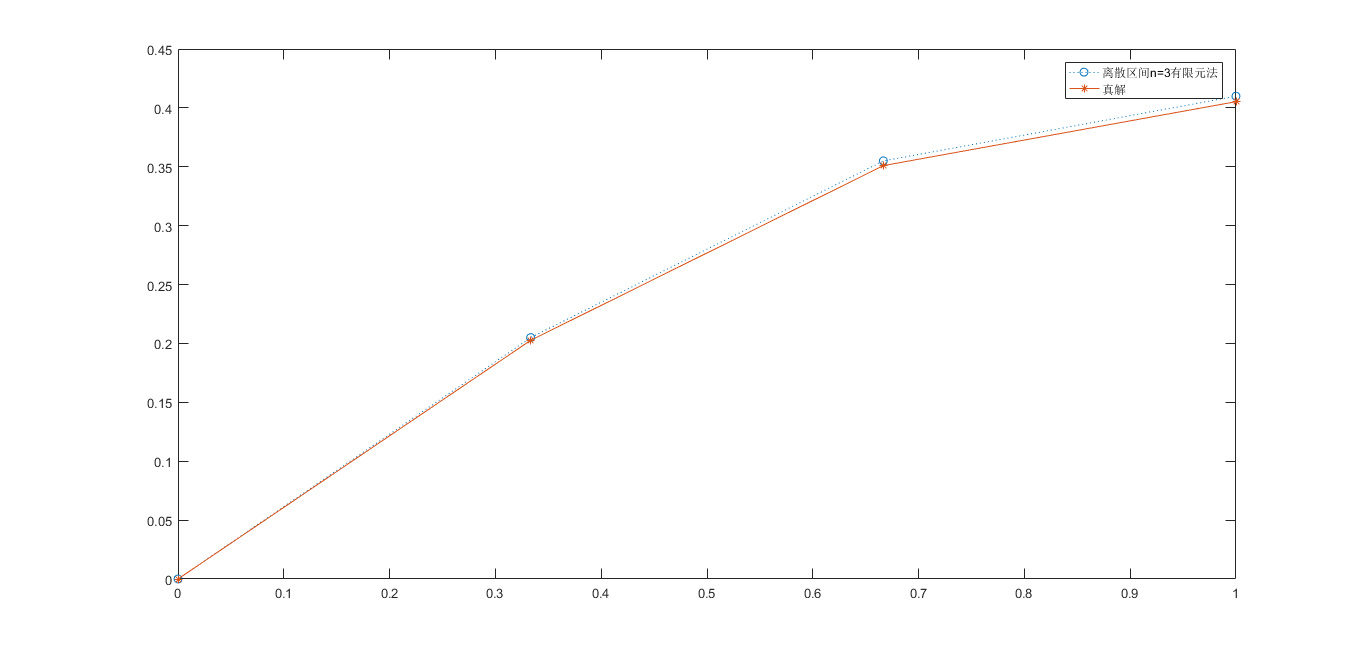
end

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 | 0.06346567 | 0.12536860 | 0.18418454 | 0.23846525 | 0.286874151 | 0.328219258 | 0.361482518 | 0.385844879 | 0.400706458 | 0.405701314 |
| 真解 | 0 | 0.06340050 | 0.12523987 | 0.18399542 | 0.23822039 | 0.286579584 | 0.327882238 | 0.361111343 | 0.385448688 | 0.400295007 | 0.405284735 |



分别作出n=3,10以及真解的图像:

