程序设计3

41704108 李达苇

(1)算法思想：

因为Legendre多项式在[-1,1]内有n个不同的实根，因此n次Legendre多项式可以表示成

在利用牛顿法求解出的第一个根后，利用“压缩技术”，避免重复求根。记 ,再次利用牛顿法求解的一个根，即得到的第二根。 由于 ,所以第二次迭代函数为

当我们已经求解出前k个根 后,考虑

因为 ,所以

因此第k+1次迭代函数为

利用上面的迭代函数进行迭代，可以得到第k+1个根。如此循环下去，便可以得到n个根。

算法流程：

1. 利用Legendre多项式的三项递推公式

把与表示出来

1. 给定初始值 ,计算与
2. 利用牛顿迭代法求出第一个根，即迭代 。当(给定误差)时终止迭代，此时 为所求的第一个根。
3. 循环：k =2，…，30. 迭代

当时终止迭代。 此时为所求的第k个根

（2）Matlab程序

function r=lratio(n,x)

p0=1; p0s=0; p1=x; p1s=1;

for k = 1:n-1

p2 = (2\*k+1)\*x\*p1/(k+1) - k\*p0/(k+1);

p2s = (2\*k+1)\*p1/(k+1) + (2\*k+1)\*x\*p1s/(k+1) - k\*p0s/(k+1);

p0 = p1; p1 = p2;

p0s = p1s; p1s = p2s;

maxx = abs(p2)+abs(p2s);

if maxx>1e20

d=1e-20;

elseif maxx<1e-20

d=1e20;

else

d = 1;

end

p1=p1\*d; p2=p2\*d; p1s=p1s\*d; p2s=p2s\*d;

end

r=p2/p2s;

end

function root1 = Newton1(x0,tolerance)

x = x0 - lratio(30,x0);

while abs(x-x0) > tolerance

x0 = x;

x = x0 - lratio(30,x0);

end

x1 = x;

root1 = x1;

end

function Rroots = Newton2(x0,tolerance)

xi=[Newton1(x0,tolerance)];

for k = 2:30

r = lratio(30,x0);

s = sum(1./ (x0- xi(1:k-1)));

x1 = x0 - r/(1-r\*s);

while abs(x1-x0) > tolerance

x0 = x1;

r = lratio(30,x0);

s = sum(1./ (x0- xi(1:k-1)));

x1 = x0 - r/(1-r\*s);

end

xi(k) = x1;

x0 = 1;

end

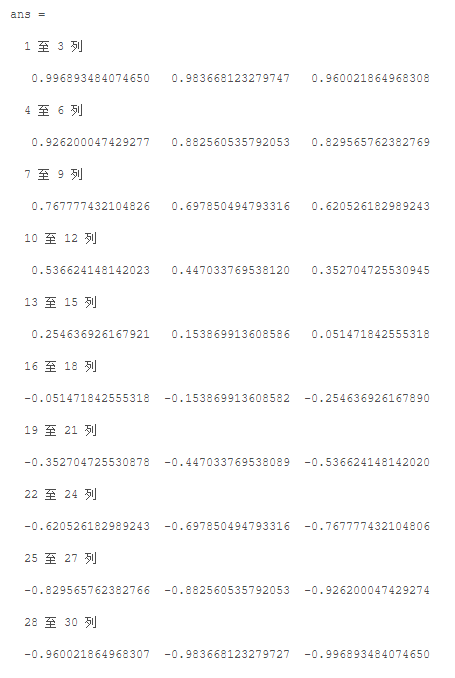
Rroots = xi;

end

Newton2(1, 0.00000005)

运行结果

在给定误差限为0.00000005的情况下



所求得的30个根与下表相对应的值的误差不超过误差限

