南昌航空大学实验报告

二○二二年4月20日

课程名称： 计算方法 实验名称： 非线性方程的求解（一）

班级： 190841 姓名： 周亚诺 同组人：

指导教师评定： 签名：

试验目的

1. 掌握逐步搜索法求解非线性方程根区间
2. 掌握二分法求解线性方程根的近似解
3. 掌握迭代法求解非线性方程的解

实验环境

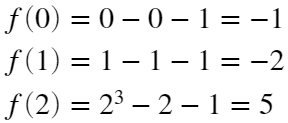
装载有Mac OS 11.6.5的计算机；

开源的Octave集成环境。

实验内容

# 用逐步搜索法求方程的一个有根区间，要求有根区间范围不得超过0.1

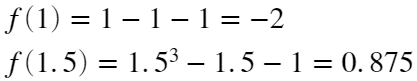
从开始以步长为 1 向右搜寻：



说明区间为一个有根区间。

区间范围超过，继续缩小区间。

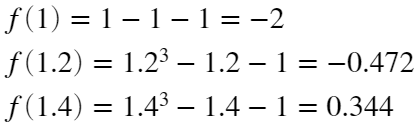
从开始以步长为 0.5 向右搜寻：



说明区间为一个有根区间。

区间范围超过，继续缩小区间。

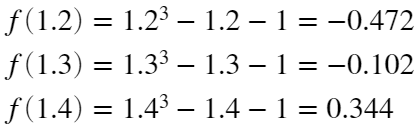
从开始以步长为 0.2 向右搜寻：



说明区间为一个有根区间。

区间范围超过，继续缩小区间。

从开始以步长为 0.1 向右搜寻：



说明区间为一个有根区间。

符合区间范围要求。

则此题的一个解为：区间

可以定义一种逐步搜索法的函数，如结尾处所示。

则可由如下脚本解决该问题，虽然效率不及上述计算过程。

clear;

f = inline('x^3-x-1','x');

[lower, higher] = stepwise(f, 0, 10, 0.1)

lower =

1.300000000000000

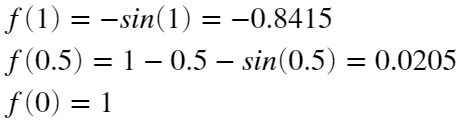
higher =

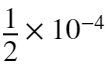
1.400000000000000

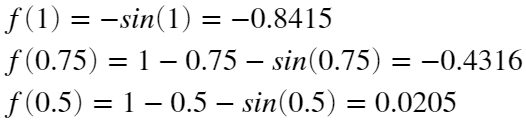
则此题的一个解为：区间

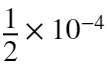
# 用二分法求解方程在区间内的一个实根，使得误差不大于

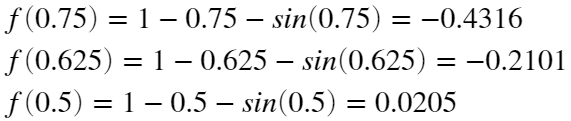
使用二分法，计算端点和中点的值：

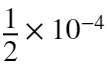


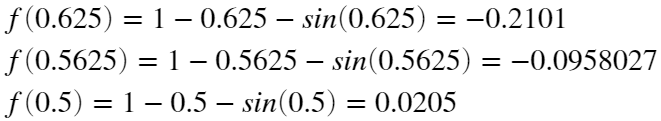
中点误差大于，继续二分：

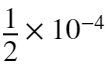


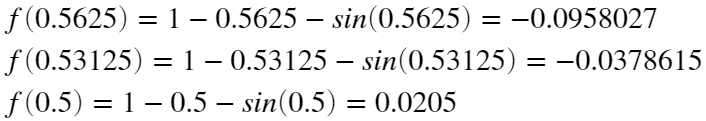
中点误差大于，继续二分：

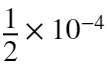


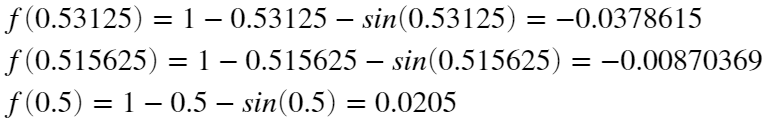
中点误差大于，继续二分：

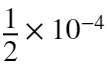


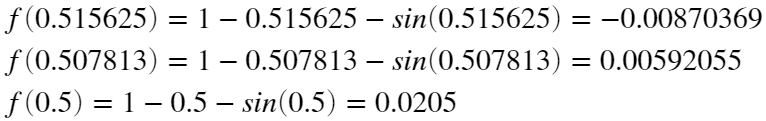
中点误差大于，继续二分：

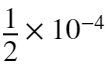


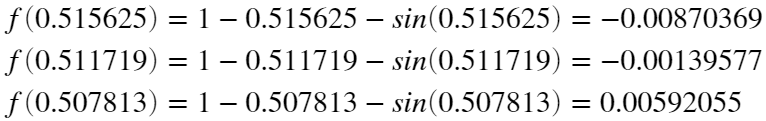
中点误差大于，继续二分：



中点误差大于，继续二分：



中点误差大于，继续二分：



计算过程过于复杂，余下省略。



可以由结尾处代码定义一种二分法求解方程的函数

则可由如下脚本解决该问题：

clear;

f = inline('1-x-sin(x)', 'x');

[res, err] = bisection(f, 0, 1, 5e-5)

res =

0.510986328125000

err =

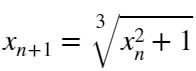
2.414986223420179e-05

则此题结果为：

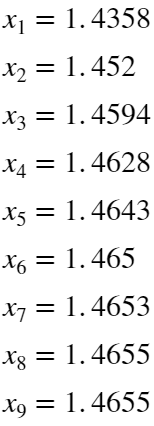
求得一个实根为，其误差为。

# 用迭代法求解方程在区间上的根，要求保留至少5位有效数字

将方程改为迭代式，即为：



带入，可得：



故该方程在区间内一个根为

可以由结尾处代码定义一种用迭代法求解方程的函数

则可由如下脚本解决该问题：

clear;

[res, err, count] = iterationSample(1.4, 1e-4)

count=1, res=1.4358

count=2, res=1.452

count=3, res=1.4594

count=4, res=1.4628

count=5, res=1.4643

count=6, res=1.465

count=7, res=1.4653

count=8, res=1.4655

count=9, res=1.4655

res = 1.4655

err = 6.5266e-05

count = 9

故该方程在区间内一个根为

# 此实时脚本中使用的函数：

function [lower, higher] = stepwise(f, lower, higher, step)

% 逐步搜索

node = lower;

while node < higher

if f(node)\*f(node+step) < 0

lower = node;

higher = node + step;

break

end

node = node + step;

end

end

function [res, err] = bisection(f, lower, higher, err)

while f(lower)\*f(higher)<0

mid = (lower+higher)/2;

if f(lower)\*f(mid) < 0

higher = mid;

elseif f(higher)\*f(mid) < 0

lower = mid;

end

if abs(f(mid)) < err

break;

end

end

err = abs(f(mid));

res = mid;

end

function [res, err, count] = iterationSample(x, errRequest)

err = 1e4;

count = 0;

while err >= errRequest

res = (x^2+1)^(1/3);

err = res - x;

x = res;

count = count + 1;

disp("count="+count+", res="+res)

end

end

实验小结

通过此次实验，掌握了使用逐步搜索法求解非线性方程根区间、使用二分法求解非线性方程根的近似解、使用迭代法求解非线性方程的解的方法，并编写了相关脚本程序，实践了求解的过程。

此次实验采用开源的Octave完成，相比于完整的MATLAB集成环境，Octave的占用空间仅为600MB，极大程度解决了电脑空间不足的问题。

在使用迭代法的时候发现，如果选取的起始点离根的位置较远，就无法在短时间内收敛，甚至有超过运算规定的运行内存而强制结束的现象。此现象有待改进。