**实验四 方程求根实验**

学号：19001531 姓名：陈正江

**一. 实验目的**

（1）深入理解方程求根的迭代法的设计思想，学会利用校正技术和松弛技术解决某些实际的非线性方程问题，比较这些方法解题的不同之处。

（2）熟悉 Matlab 编程环境，利用 Matlab 解决具体的方程求根问题。

**二. 实验要求**

用Matlab 软件实现根的迭代法、Newton 法、快速弦截法和弦截法，并用实例在计 算机上计算。

**三. 实验内容**

**1、实验题目**

**4-1：取=1，用迭代法求方程的根，然后用Aitken方法加速，要求计算结果有4位有效数字。**

**2、设计思想**

**（1）迭代方法的设计思想：**

构造不动点方程来求得近似根。将方程f(x)=0变换为等价形式，然后建立迭代格式。给定初值x0后，求得的迭代数列可能收敛也可能不收敛，若收敛与x\*，则它就是方程的根。

**（2）Aitken加速方法的设计思想：**

迭代：

迭代：

加速：

**3、对应程序**

1. **迭代函数代码如下：**

function f = czj4\_1\_1\_f( x )

f = (x.^3 - exp(x) + 2)/3;

1. **迭代法代码如下：**

function [ x,k ] = czj4\_1\_2\_f( f,x0,emg )

x(1) = x0;

k = 1;

x(k+1) = feval(f,x(k));

k = k+1;

x(k) = feval(f,x(k-1));

while abs(x(k)-x(k-1)) > emg

k = k+1;

x(k) = feval(f,x(k-1));

end

1. **Aitken加速方法代码如下：**

function [ x,k ] = czj4\_1\_3\_f( f,x0,emg )

x(1) = x0;

k = 1;

x(k+1) = feval(f,x(k));

k = k+1;

x(k) = feval(f,x(k-1));

while abs(x(k)-x(k-1)) > emg

k = k+1;

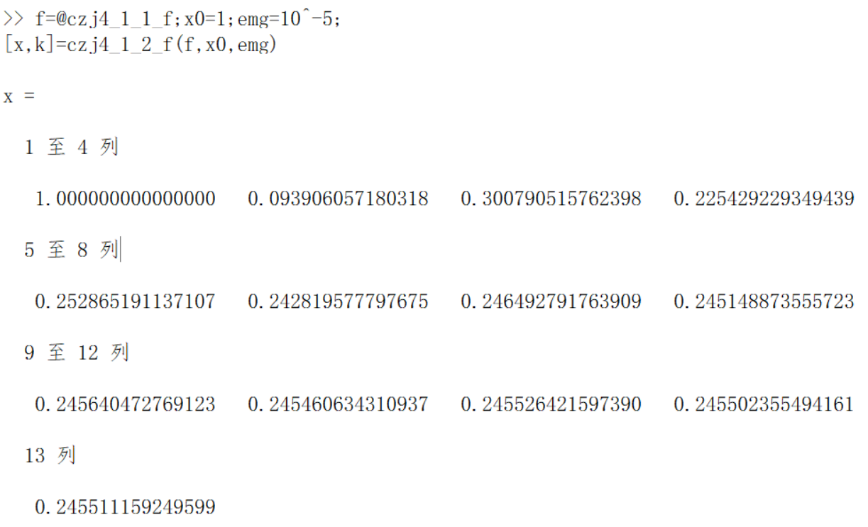
x1 = feval(f,x(k-1));

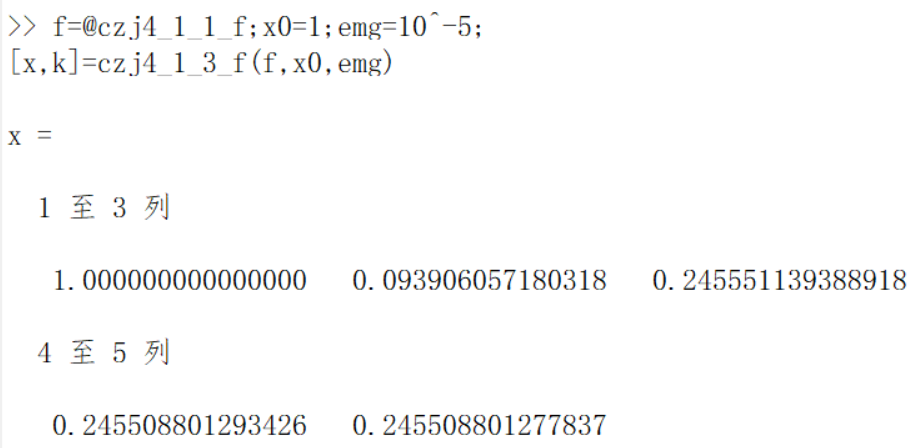
x2 = feval(f,x1);

x(k) = x2 - (x2-x1)^2/(x2-2\*x1+x(k-1));

end

**4、实验结果**

** (1) 迭代法求解：**

**(2) Aitken加速方法求解：**

**1、实验题目**

**4-2：用Newton法求方程在内的根，根的准确值为，要求误差不超过**10-4**，并绘制方程的图形进行检验。**

**2、设计思想**

Newton法的设计思想是将非线性方程组的求根逐步线性化，并且根据Newton公式推导出Newton迭代函数。

Newton公式：



Newton迭代函数：



**3、对应程序**

1. **原方程函数代码：**

function [ f,d ] = czj4\_2\_1\_f( x )

f = x.^3 + 10\*x - 20;

syms y;

d1 = y.^3 + 10\*y - 20;

d = subs(diff(d1),y,x);

1. **Newton法代码：**

function [ x,k ] = czj4\_2\_2\_f( f,x0,emg )

[f1,d1] = feval(f,x0);

k = 1;

x(1) = x0;

x(2) = x(1) - f1/d1;

while abs(f1) > emg

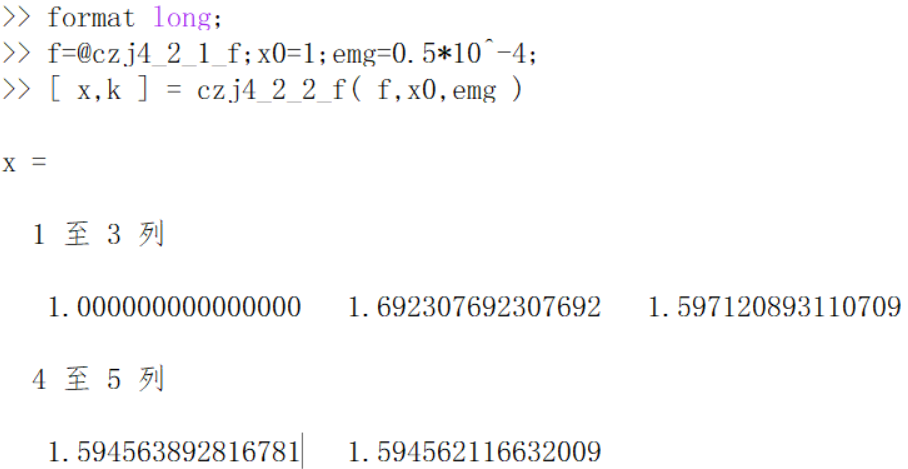
k = k+1;

[f1,d1] = feval(f,x(k));

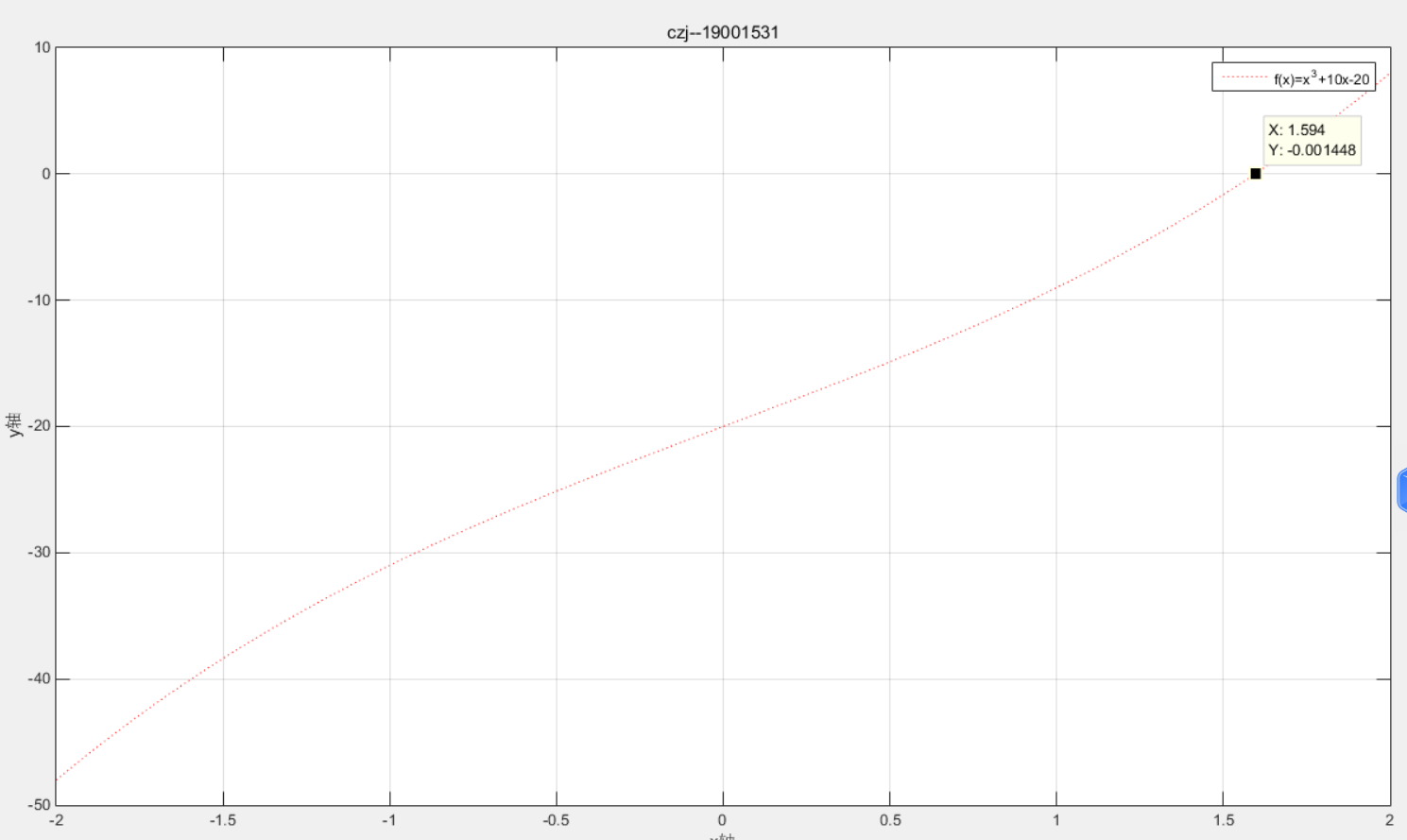
x(k+1) = x(k) - f1/d1;

end

**4、实验结果**



**绘制方程的图形如下：**



由图像可知，求得的近似解合理。

**1、实验题目**

**4-3：分别用弦截法和快速弦截法求解方程，要求精度为，取作为开始值，并绘制的图形进行验证。**

**2、设计思想**

**（1）弦截法的设计思想：**

利用插值原理和数值微分的思想，用差商代替导数，避免计算导数：



为了进一步提高速度，用新的差商代替导数，可以导出快速弦截法。

1. **快速弦截法的设计思想：**

用差商代替导数，避免计算导数：



**3、对应程序**

1. **原方程函数代码：**

function f = czj4\_3\_1\_f( x )

f = x.\*exp(x) - 1;

1. **弦截法：**

function [ x,k ] = czj4\_3\_2\_f(f,x0,x1,emg)

k = 1;

y0 = feval(f,x0);

y1 = feval(f,x1);

x(k) = x1 - (x1 - x0)\*y1/(y1 - y0);

y(k) = feval(f,x(k));

k = k+1;

x(k) = x(k-1) - (x(k-1) - x0)\*y(k-1)/(y(k-1) - y0);

while abs(x(k) - x(k-1)) > emg

y(k) = feval(f,x(k));

x(k+1) = x(k) - (x(k) - x0)\*y(k)/(y(k) - y0);

k = k+1;

end

1. **快速弦截法：**

function [ x,k ] = czj4\_3\_3\_f(f,x0,x1,emg)

k = 1;

y0 = feval(f,x0);

y1 = feval(f,x1);

x(k) = x1 - (x1 - x0)\*y1/(y1 - y0);

y(k) = feval(f,x(k));

k = k+1;

x(k) = x(k-1) - (x(k-1) - x1)\*y(k-1)/(y(k-1) - y1);

while abs(x(k) - x(k-1)) > emg

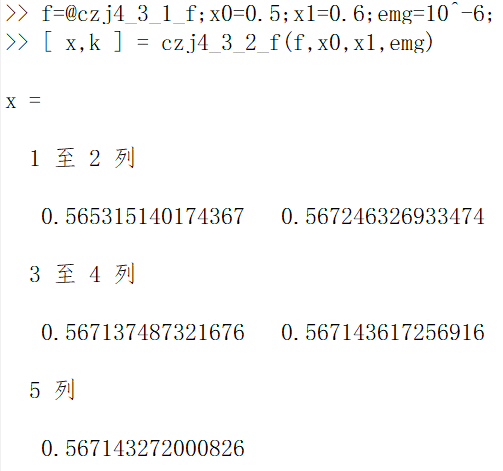
y(k) = feval(f,x(k));

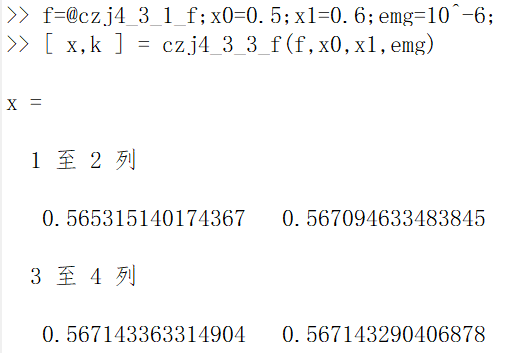
x(k+1) = x(k) - (x(k) - x(k-1))\*y(k)/(y(k) - y(k-1));

k = k+1;

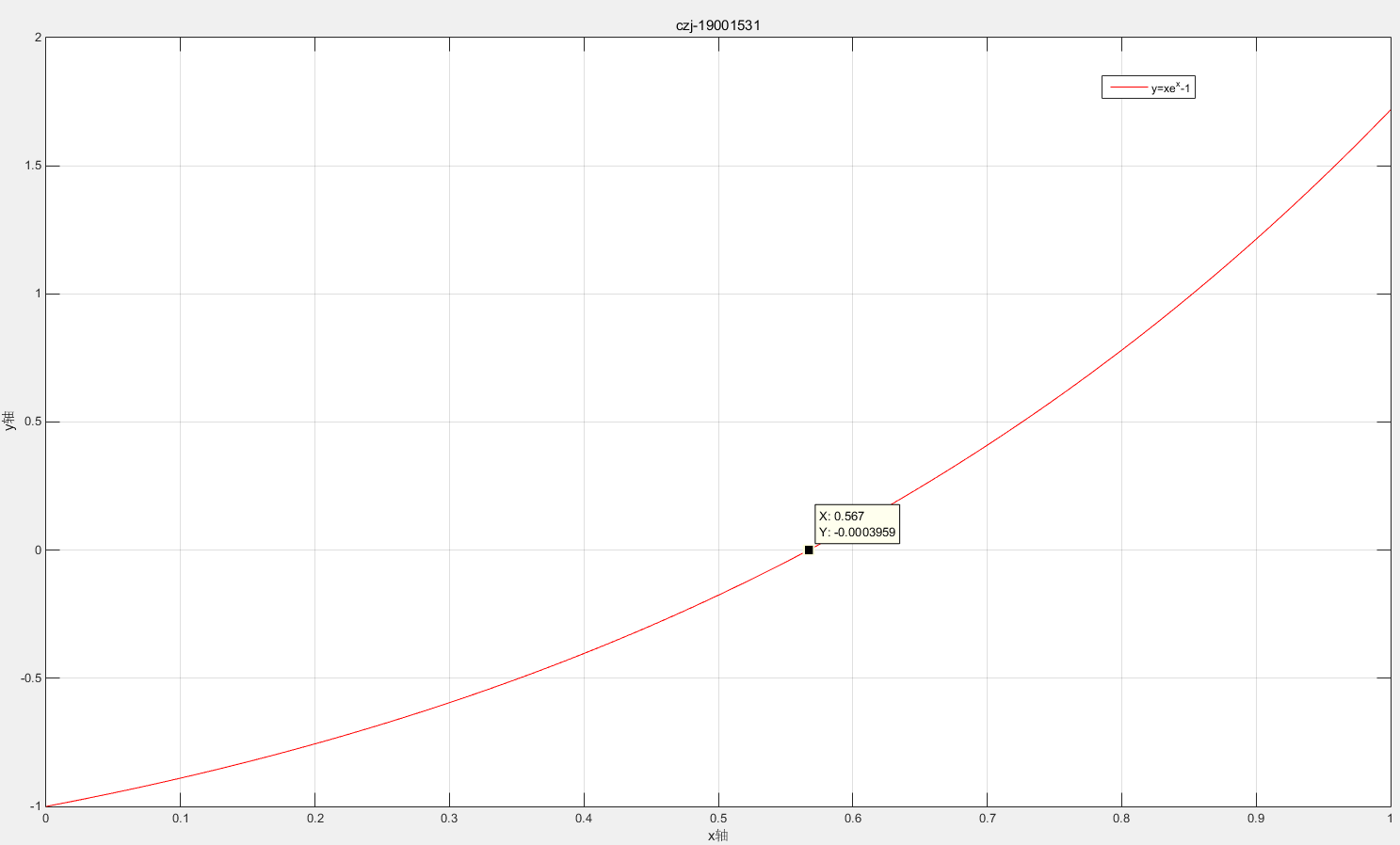
end

**4、实验结果**

**（1）弦截法求解的运行结果： （2）快速弦截法求解的运行结果：**



**绘制方程的图形如下：**

****

由图像可知，求得的近似解合理。

**四. 实验体会**

通过以上实验，我有以下总结和收获：

1. Newton法的优点就是收敛快，逻辑结构简单；缺点是如果选择的初值不恰当，迭代从一个根跳到另一个根的情形，即会导致迭代发散。
2. 迭代法有时收敛速度非常的缓慢，可能得迭代几十次才能得到在精度范围内的解；而Aitken加速迭代法收敛速度则较快。
3. 弦截法需要计算函数值，而牛顿法既需要计算函数值，又要计算导数值，所以弦截法计算强度小于牛顿法；弦截法的收敛速度稍慢于牛顿法，但是与迭代法相比要快。
4. 快速弦截法收敛速度可以与Aitken加速法相当，但是缺点是它不能够自启动，需要给定两个合适的初值。