学号：20002462 姓名：刘子言

实验四 方程求根实验

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 软件实现根的二分搜索、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

3-1： 用 Newton 法求方程$f(x)=x^3-3x-1=0$ 在x_0=2 附近的根，根的准确值为$x^*=1.87938524\ldots$，要求计算结果有 4 位有效数字，并绘制方程的图形进行检验。

3-2： 取x_0=1，用迭代法求方程$x^3-3x-e^x+2=0$ 的根，然后用 Aitken 方法加速，要求计算结果有 4 位有效数字。（提示：此题答案有多个，作对一个即算正确）

3-3： 分别用弦截法和快速弦截法求解方程 $f(x)=xe^x-1=0$，要求精度为 $\varepsilon=10^{-6}$，取 $x_0=0.5$，$x_1=0.6$ 作为开始值，并绘制 $f(x)=xe^x-1$ 的图形进行验证。

2. 设计思想

要求针对上述题目，详细分析每种算法的设计思想。

3. 对应程序

列出每种算法的程序。

4. 实验结果

列出相应的运行结果截图，如果要求可视化，则同时需要给出相应的图形。

四. 实验分析

对实验过程进行分析总结，对比方程求根的不同方法，指出每种方法的设计要点及应注意的事项，以及自己通过实验所获得的对方程求根问题的各种解法的理解。

（注：不要改变实验报告的结构，写清页码和题号，源程序以自己的姓名命名，如 3-1 题可命名为“zhangsan_3-1.m”，运行截图中应出现自己的姓名和题号）

题目 1、用 Newton 法求方程$f(x)=x^3-3x-1=0$在x_0=2 附近的根，根的准确值为$x^*=1.87938524\ldots$，要求计算结果有 4 位有效数字，并绘制方程的图形进行检验。

1、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：

$$x_{k+1}=x_k-\frac{f(x_k)}{f'(x_k)}$$

Newton 迭代函数：

$$\varphi(x)=x-\frac{f(x)}{f'(x)}$$

2、对应程序

（1）方程的函数：

```
function [ f,d ] = liuziyan_4_1_func( x )
f = x.^3 - 3*x - 1;
syms y;
d1 = y.^3 - 3*y - 1;
d = subs(diff(d1),y,x);   %对函数 f 求一次导数
```

（2）Newton 法的对应程序：

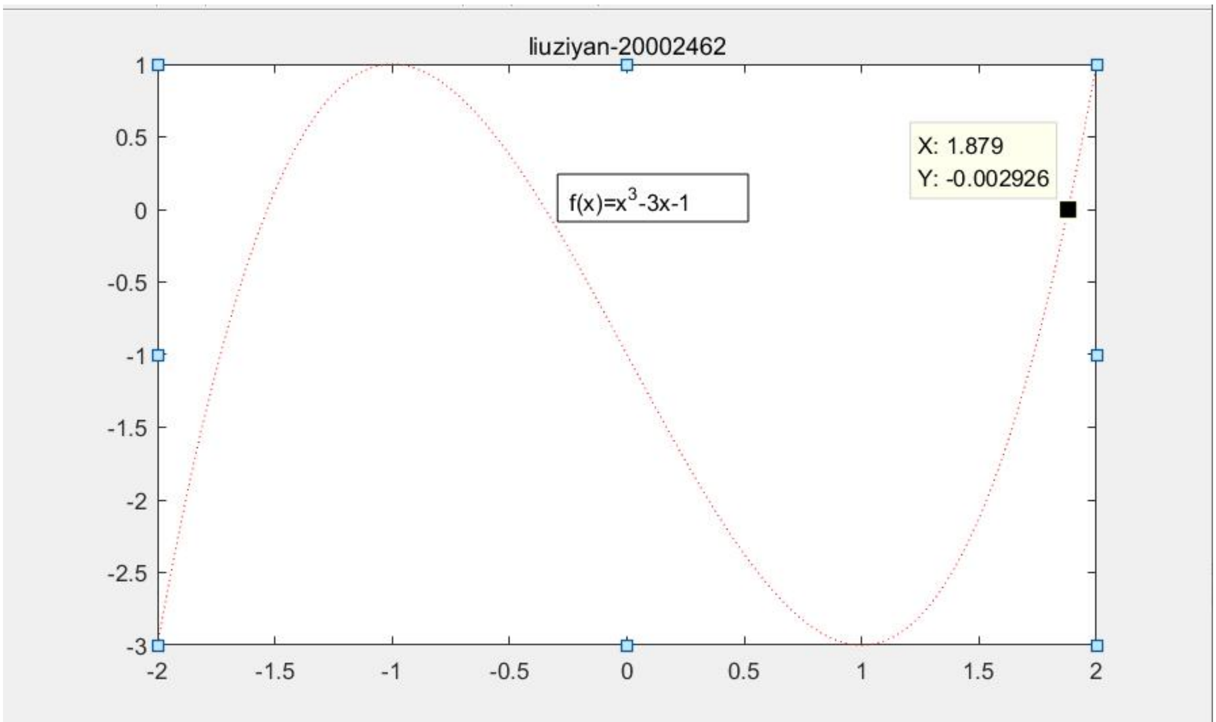
```
function [ x,k ] = liuziyan_4_1_Newton( f,x0,emg )
%   用牛顿法求解方程的根
%   f——线性方程左端函数；x0——迭代初值（此方法为局部收敛，初值选取要恰当）
%   emg——精度指标；k——迭代次数
[f1,d1] = feval(f,x0);
k = 1;
x(1) = x0;
x(2) = x(1) - f1/d1;
while abs(f1) > emg        %控制精度
    k = k+1;
    [f1,d1] = feval(f,x(k));
    x(k+1) = x(k) - f1/d1;
end
```

3、实验结果

（1）牛顿法求解方程根的运行结果如下：

```
>> format long;
>> f=@liuziyan_4_1_func;x0=2;emg=0.5*10^-4;
>> [ x,k ] = liuziyan_4_1_Newton( f,x0,emg )

x =

   2.000000000000000   1.888888888888889   1.879451566951567   1.879385244836671   1.879385241571817


k =

     4
```

（2）下面绘制方程的图形进行检验：



由图像可知，求得的近似解合理。

题目 2、取x_0=1，用迭代法求方程$x^3-3x-e^x+2=0$的根，然后用 Aitken 方法加速，要求计算结果有 4 位有效数字。（提示：此题答案有多个，作对一个即算正确）

1、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$x=\varphi(x)$，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：$\bar{x}_{k+1} = \varphi(x_k)$

迭代：$\tilde{x}_{k+1} = \varphi(\bar{x}_{k+1})$

加速：$x_{k+1} = \tilde{x}_{k+1} - \dfrac{(\tilde{x}_{k+1} - \bar{x}_{k+1})^2}{\tilde{x}_{k+1} - 2\bar{x}_{k+1} + x_k}$

2、对应程序

（1）迭代函数：

```
function f = liuziyan_4_2_func( x )
f = (x.^3 - exp(x) + 2)/3;
```

（2）迭代法对应程序：

```
function [ x,k ] = liuziyan_4_2_Diedai( f,x0,emg )
% 用迭代法求解线性方程
%   f——线性方程左端函数；x0——迭代初值
%   emg——精度指标；k——迭代次数
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        %控制精度
      k = k+1;
      x(k) = feval(f,x(k-1));
end
```

（3）Aitken 加速方法对应程序：

```
function [ x,k ] = liuziyan_4_2_Aitken( f,x0,emg )
% 用 Aitken 加速法求解方程
% f——线性方程左端函数；x0——迭代初值（此方法为局部收敛，初值选取要恰当）
% emg——精度指标；k——迭代次数
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        %控制精度
      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
```

3、实验结果

（1）迭代法求解的运行结果如下：

```
>> f=@liuziyan_4_2_func;x0=1;emg=10^-5;
>> [ x,k ] = liuziyan_4_2_Diedai( f,x0,emg )

x =

  1 至 6 列

   1.000000000000000   0.093906057180318   0.300790515762398   0.225429229349439   0.252865191137107   0.242819577797675

  7 至 12 列

   0.246492791763909   0.245148873555723   0.245640472769123   0.245460634310937   0.245526421597390   0.245502355494161

  13 列

   0.245511159249599


k =

    13
```

（2）Aitken 加速方法求解的运行结果如下：

```
>> f=@liuziyan_4_2_func;x0=1;emg=10^-5;
>> [ x,k ] = liuziyan_4_2_Aitken( f,x0,emg )

x =

   1.000000000000000   0.093906057180318   0.245551139388918   0.245508801293426   0.245508801277837


k =

     5
```

题目 3、分别用弦截法和快速弦截法求解方程 $f(x) = xe^x - 1 = 0$，要求精度为 $\varepsilon = 10^{-6}$，取 $x_0 = 0.5$，$x_1 = 0.6$ 作为开始值，并绘制 $f(x) = xe^x - 1$ 的图形进行验证。

1、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 $\dfrac{f(x_k)-f(x_0)}{x_k - x_0}$ 代替导数，避免计算导数：

$$\varphi(x)=x-\frac{f(x)}{f(x)-f(x_0)}(x-x_0)$$

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。

（2）快速弦截法的设计思想：

用差商$\dfrac{f(x_k)-f(x_{k-1})}{x_k-x_{k-1}}$代替导数，避免计算导数：

$$x_{k+1}=x_k-\frac{f(x_k)}{f(x_k)-f(x_{k-1})}(x_k-x_{k-1})$$

2、对应程序

（1）方程的函数：

```
function f = liuziyan_4_3_func( x )
f = x.*exp(x) - 1;
```

（2）弦截法对应程序：

```
function [ x,k ] = liuziyan_4_3_Chord(f,x0,x1,emg)
% 用弦截法求解方程的根
% f——线性方程左端函数；x1，x2——迭代初值
% emg——精度指标；k——循环次数
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
```

（3）快速弦截法对应程序：

```
function [ x,k ] = liuziyan_4_3_Fast_chord(f,x0,x1,emg)
% 用快速弦截法求解方程的根
% f——线性方程左端函数；x1，x2——迭代初值
% emg——精度指标；k——循环次数
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
```

3、实验结果

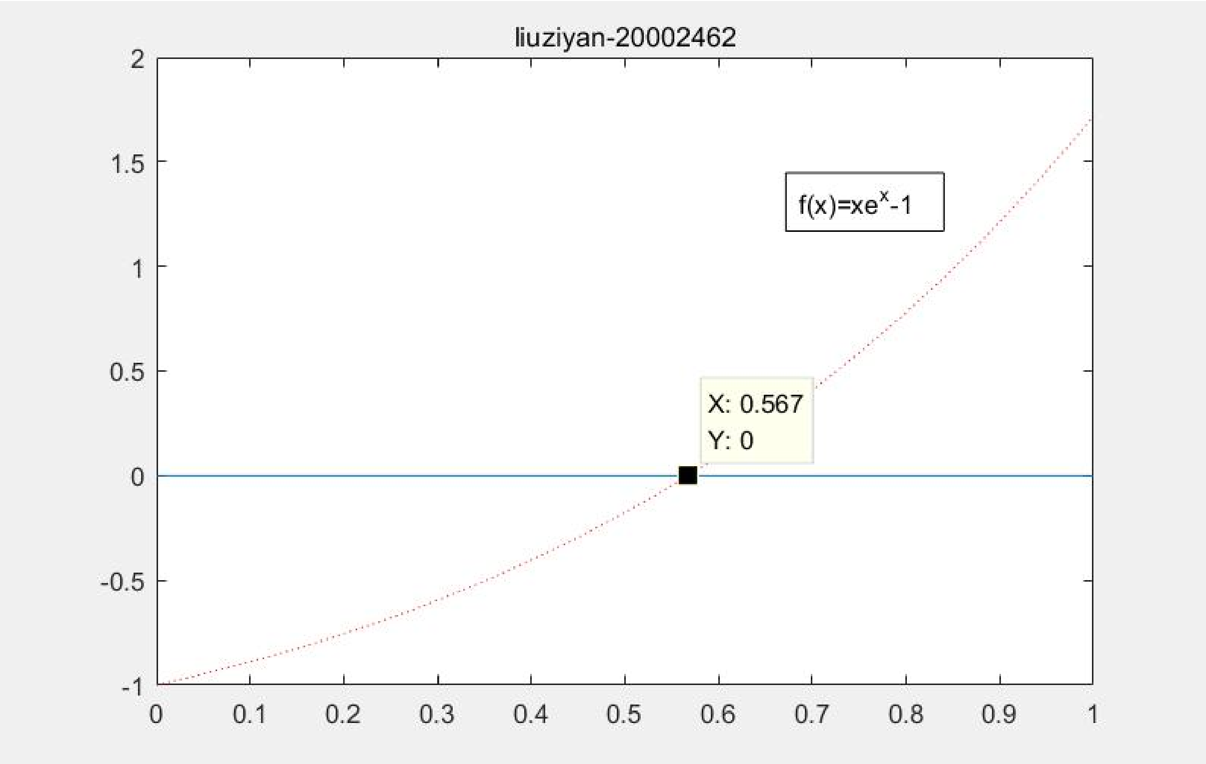
（1）弦截法求解的运行结果：

```
>> f=@liuziyan_4_3_func;x0=0.5;x1=0.6;emg=10^-6;
>> [ x,k ] = liuziyan_4_3_Chord(f,x0,x1,emg)

x =

   0.565315140174367   0.567246326933474   0.567137487321676   0.567143617256916   0.567143272000826


k =

     5
```

（2）快速弦截法求解的运行结果：

```
>> f=@liuziyan_4_3_func;x0=0.5;x1=0.6;emg=10^-6;
>> [ x,k ] = liuziyan_4_3_Fast_chord(f,x0,x1,emg)

x =

   0.565315140174367   0.567094633483845   0.567143363314904   0.567143290406878


k =

     4
```

（3）下面绘制方程的图形进行检验：



由图像可知，求得的近似解合理。

四、实验分析

对实验过程进行分析总结，对比方程求根的不同方法，指出每种方法的设计要点及应注意的事项，以及自己通过实验所获得的对方程求根问题的各种解法的理解。

答：

（1）Newton 法的优点就是收敛快，逻辑结构简单；缺点是每一步都要计算函数函数值和导数值，程序常常发生中断，且初始值只有在根的附近才能保证收敛；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。