第一章作业

一、问题叙述

(1) 用Newton-Raphson方法和割线法求方程 $x\tan(x)=2$ 的位于区间 $[0,\pi/2]$ 的一个根，要求相对误差限为0.01%，并画图比较两种方法的收敛速度。

(2) 用高斯消去法求解下述方程组：

$$\begin{cases} 2x_1+x_2+4x_3=16 \\ 3x_1+2x_2+x_3=10 \\ x_1+3x_2+3x_3=16 \end{cases}$$

二、Matlab 代码和运算结果

第一题 Newton 法：

代码：

```
%Using Newton-Raphson method to find solutions to a nonlinear equation
clear;close all;clc;

x2 = pi/2-0.001;            %set the max boundary
x1 = 0.001;                 %set the min boundary
tolerance = 0.0001;         %set the tolerance to end the loop

while (abs((x2-x1)/x2)>tolerance)
x1 = x2;
x2 = x1-f(x1)/df(x1);       %Newton formula
end

fprintf('%.8f\n', x2);
```

输出：

x = 1.07687401

第一题割线法：

代码：

```
%Using secant method to find solutions to a nonlinear equation
clear;close all;clc

x1 = pi/2-0.001;            %set the max boundary
x2 = 0;                     %set the min boundary
tolerance = 0.0001;         %set the tolerance to end the loop

while (abs((x2-x1)/x2)>tolerance)
    x3 = x2 - f(x2) * (x2-x1) / (f(x2)-f(x1));      %new value of 'x'

    %renew 'x1'&'x2'
    if f(x2)*f(x1) > 0
        if (abs(x2)-abs(x3)) < (abs(x1)-abs(x3))  x1 = x3;
        else  x2 = x3;
        end
    else
        if f(x3)*f(x2) > 0  x1 = x3;
        else  x2 = x3;
        end
    end
end
```

输出：

x = 1.07687399

第二题高斯消元：

部分代码：

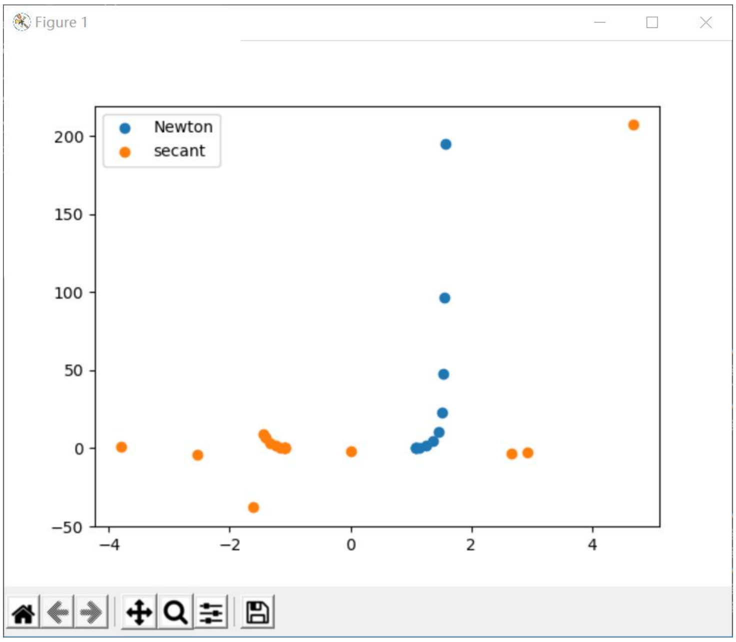
```
%Gauss elimination method
for i = n-1:-1:1
    k=0;
    for j = i+1 : n
        k = k + a(i,j)*x(j);
    end
    x(i) = (b(i)-k) / a(i,i);
end
```

输出：x = $[1\ \ 2\ \ 3]^T$

三、结果分析

第一题要求的区间范围是[0，pi/2]，但是函数在 pi/2 处无意义，因此将上限改为 pi/2-0.001，又因为在求初始的近似相对误差时 0 不能做除数，所以将下限改为 0.001。

将牛顿法和割线法可视化（割线法最后的那个点被牛顿法的挡住了），由于二者的分散程度不同，将|x|限定在 5 以内，将 f(x)限定在 200 以内，可以看到牛顿法的收敛速度更快，而割线法出现了在真值附近左右跳跃的情况。



第二题是低阶的简单矩阵，无需通过主元法、缩放法来提高精度，也显然不是病态矩阵或奇异矩阵，因此直接运算即可。