Lab 8 实验报告

实验一

实验一用课上所给的弦截法求根。代码如下。

```
[result,~]=Exp7_7(1e-6,100,-4,-2);
% 改变后两个值的范围用来求不同的根
result(end)

function[y,k]=Exp7_7(er,n,xa,xb)
x0=xa;x1=xb;
ff=@(x) x.^3-sin(x)-12.*x+1;
y(1)=xa;y(2)=xb;
k=2;
while abs(x1-x0)>er&&k<n
    fx1=ff(x1);
    fx0=ff(x0);
    x2=x1-fx1*(x1-x0)/(fx1-fx0);
    k=k+1;
    y(k)=x2;
    x0=x1;
    x1=x2;
end
end
```

求出的三个根如下

```
ans =

   -3.4912
```

```
ans =

    0.0770
```

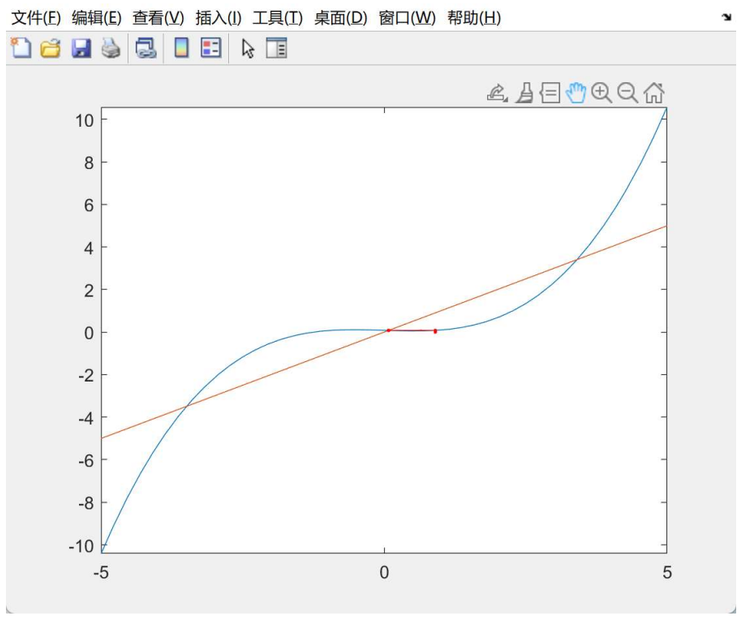
```
ans =

    3.4101
```

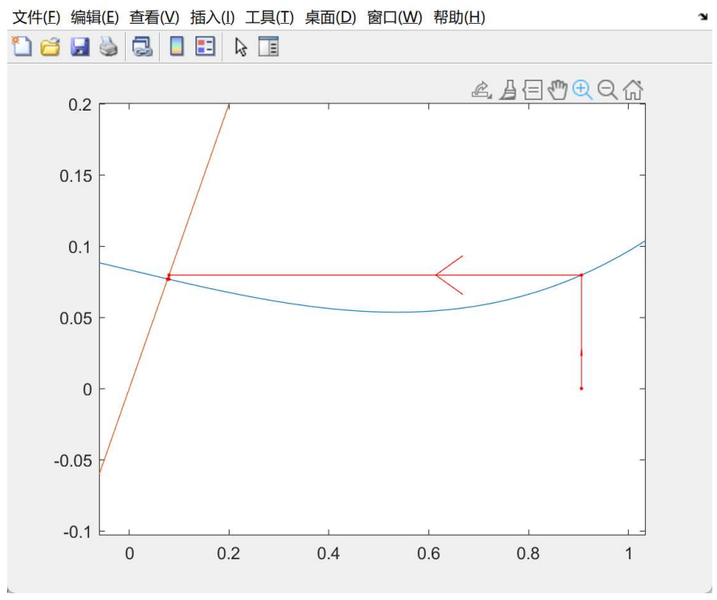
之后用不动点法画出蛛网图，设置的迭代次数为5，即显示5步。代码如下（其中的Expplot7_3函数是上课老师提供的）。

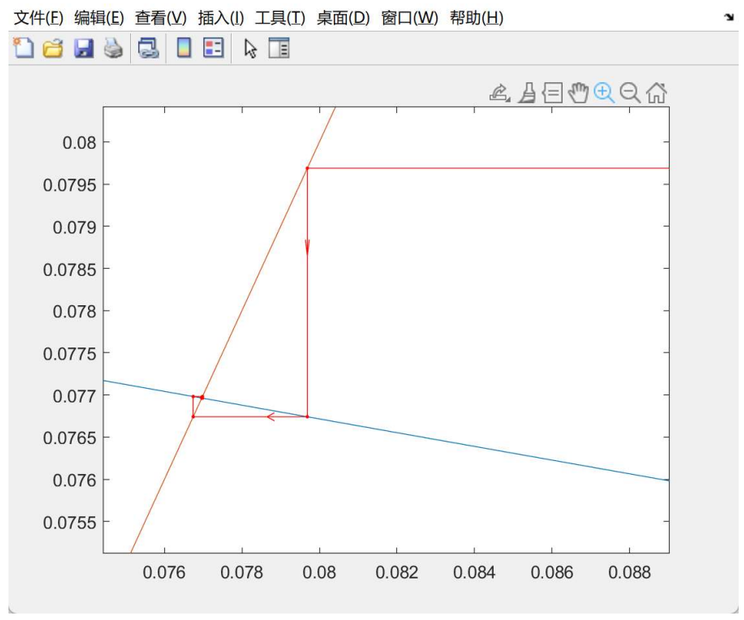
```
clf,clear,clc;
% 取消注释以使用不同函数
% func_f=@(x) nthroot((12.*x+sin(x)-1),3);
func_f=@(x) 1/12*(x^3-sin(x)+1);
```

```
fplot(func_f)
hold on
axis on
fplot(@(x) x)

x=zeros(2,1);
y=zeros(2,1);
x(2)=rand;
y(2)=func_f(x(2));
x(1)=x(2);
y(1)=0;
Expplot7_3(x,y,0.5);

for i=1:5
    x(1)=x(2);
    y(1)=y(2);
    y(2)=func_f(x(1));
    Expplot7_3(x,y,0.5)
    x(1)=x(2);
    y(1)=y(2);
    x(2)=y(2);
    Expplot7_3(x,y,0.5)
    pause(0.5)
end
```

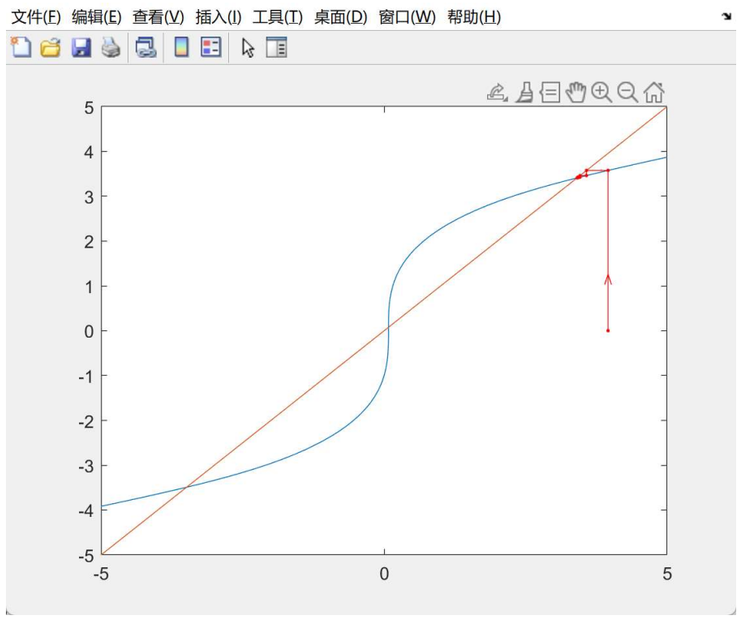
下面几张图是在0~1之间取值时的蛛网图，因为迭代的比较快，所以最后的几步迭代都是放大之后展示的，由此可见这个是收敛的。（对应根为3.4101）

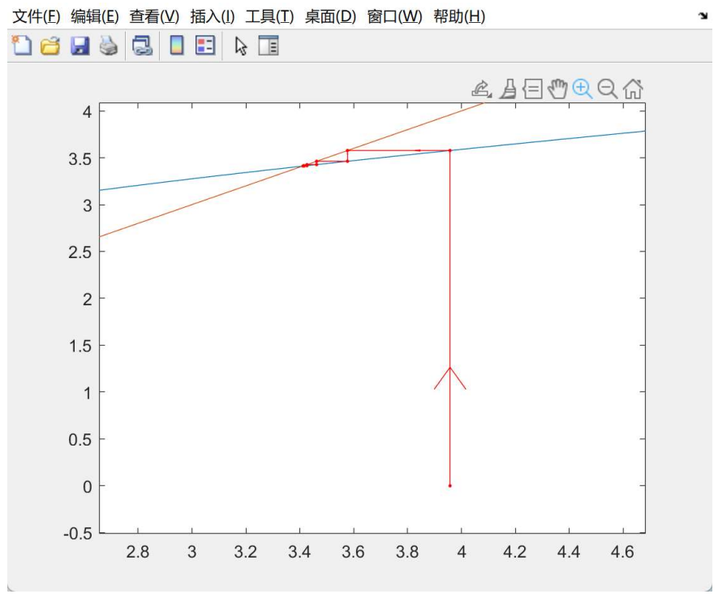


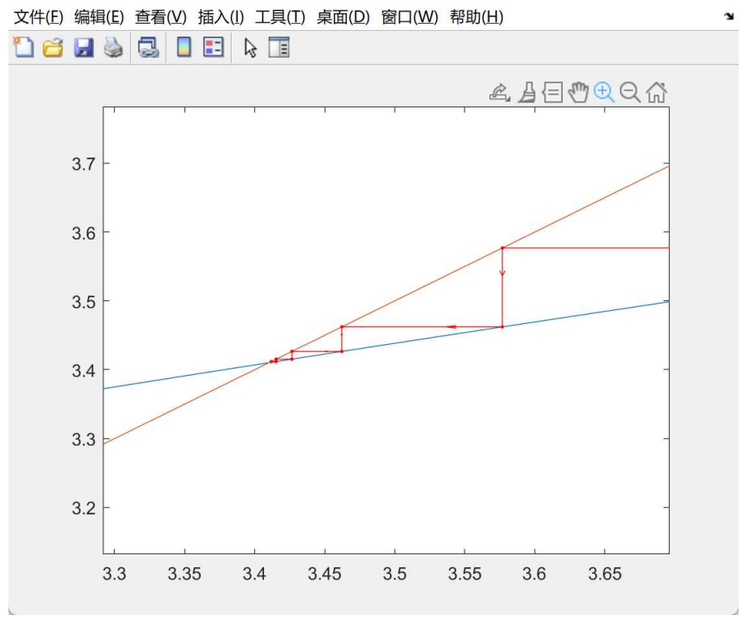




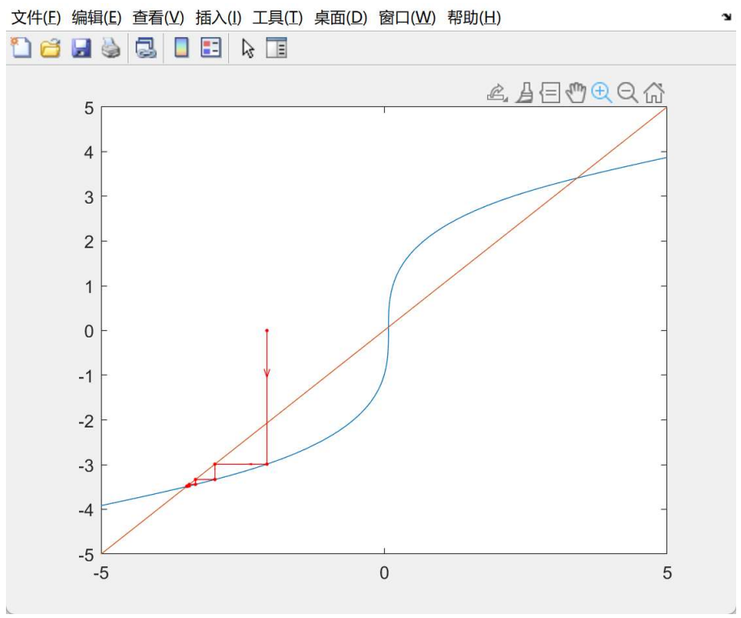
下面几张图是在3~4之间取值时的蛛网图。（对应根为（0.0770）

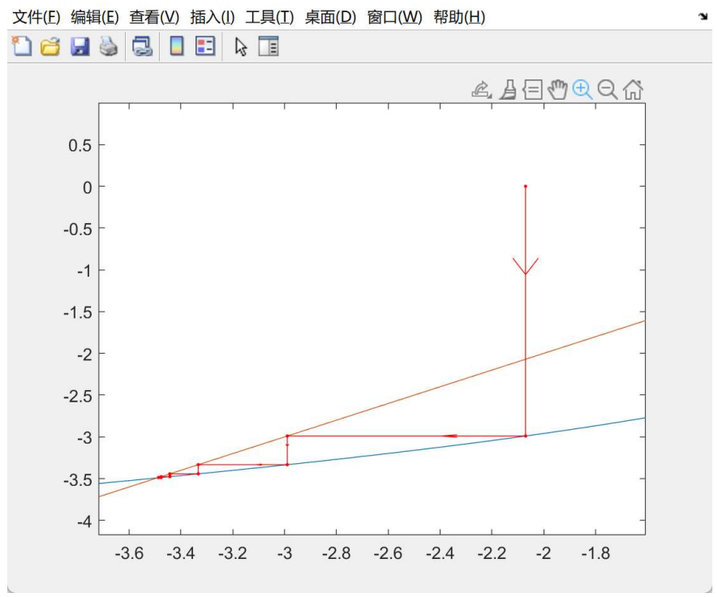


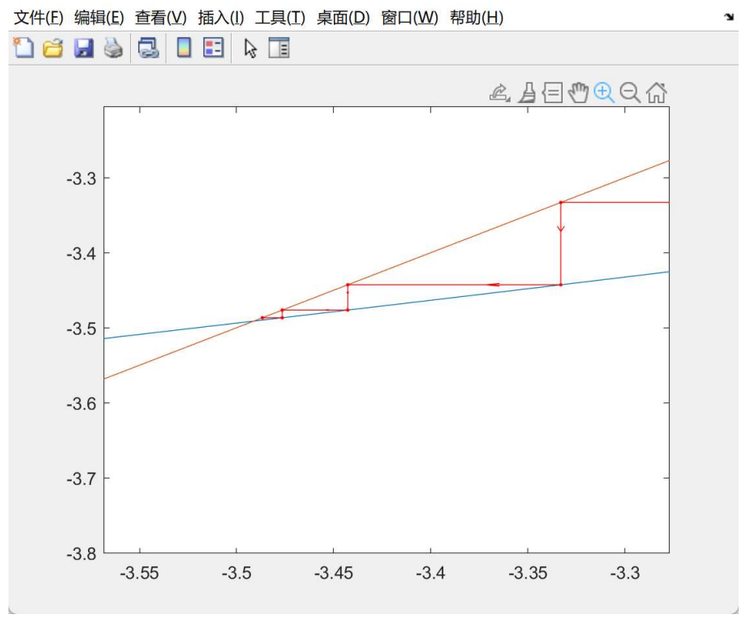




下面几张图是在-4~-2之间取值时的蛛网图。（对应的根为-3.4912）







实验二

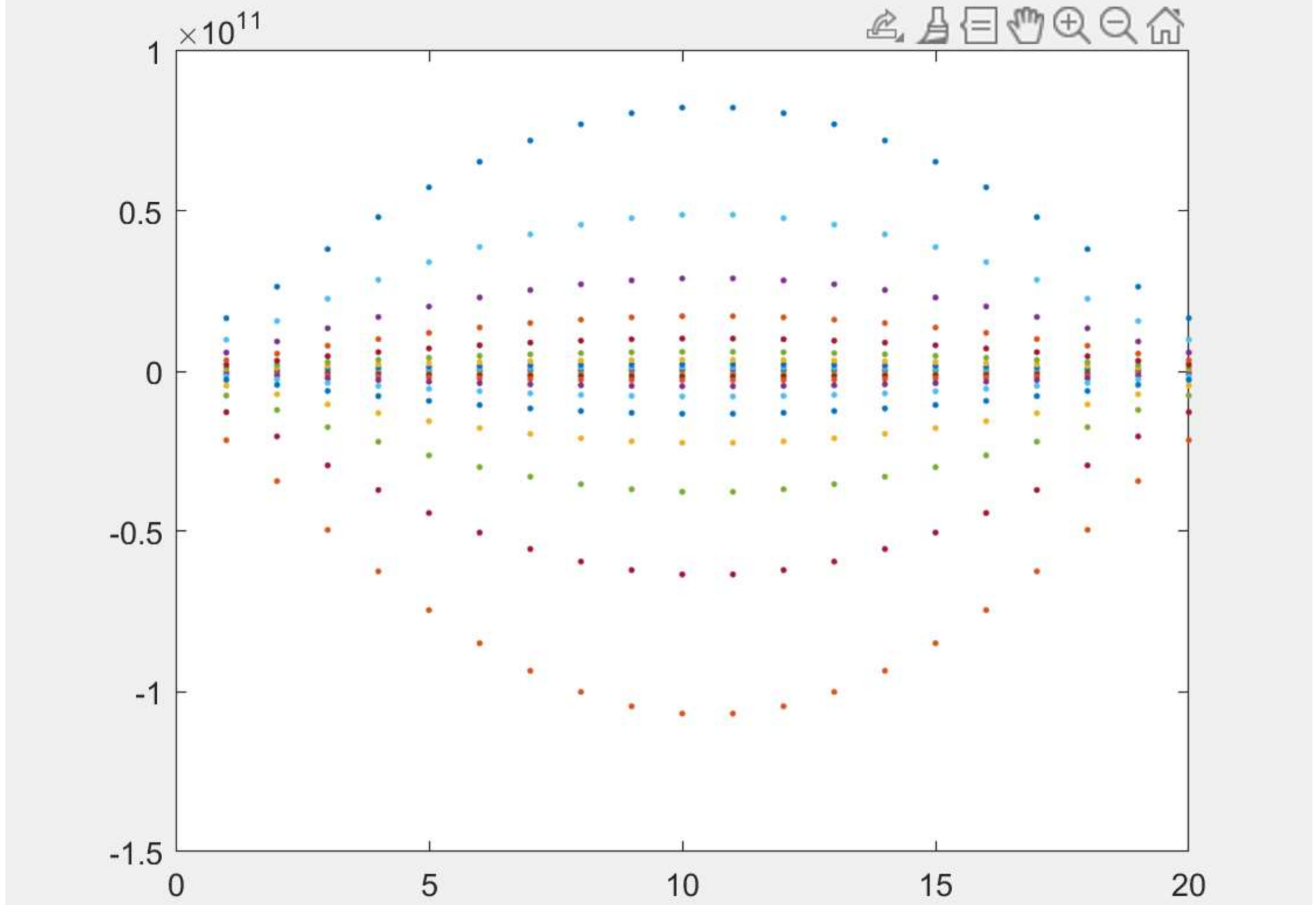
雅克比迭代

首先生成对角矩阵D，之后L+U由A-D得出，再应用雅克比迭代。代码如下

```
A=diag(ones(1,20)*3)+diag(ones(1,19),1)+diag(ones(1,19),-1)+...
    diag(ones(1,18),2)+diag(ones(1,18),-2);
b=rand(20,1);

x=zeros(20,100);
x(:,1)=rand(20,1);

D = diag(diag(A));
R = A-D;

for i = 2:100
    x(:,i)=D\(b-R*x(:,i-1));
end

plot(x,'.')
```

生成的图如下

文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)



可见雅可比迭代对于这个矩阵是不收敛的。

之后去网上查了相关的资料，查询到了雅可比矩阵的收敛条件是迭代矩阵的谱半径必须小于1，即最大特征值必须小于1.

而我们的迭代矩阵为$D^{-1}(L+U)$，算出的结果如下

```
>> eig(D\R)

ans =

   -0.7261
   -0.7232
   -0.6554
   -0.6472
   -0.5403
   -0.5346
   -0.4032
   -0.3837
   -0.2692
   -0.2027
   -0.1329
   -0.0813
   -0.0194
    0.0718
    0.3217
    0.5786
    0.8215
    1.0323
    1.1952
    1.2981
```

可以看出其中的最大值已经大于1了，即这个迭代法会失效

（其中R=L+U）

Gauss-Seidel迭代

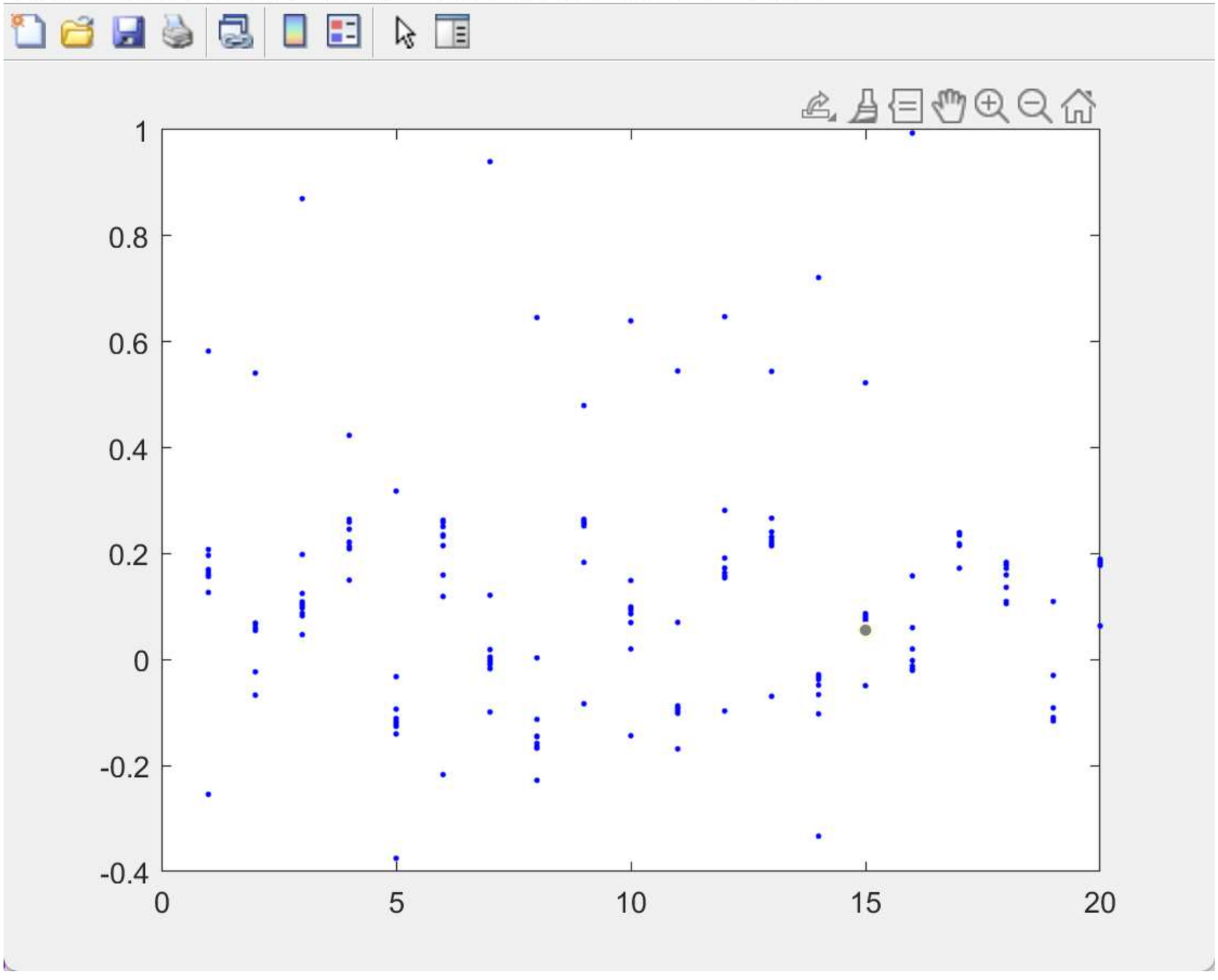
生成L和U的方法是用循环生成的，生成之后再执行迭代，代码如下

```
clear,clc;

A=diag(ones(1,20)*3)+diag(ones(1,19),1)+diag(ones(1,19),-1)+diag(ones(1,18),2)+diag(ones(1,18),-2);
b=rand(20,1);
U=zeros(20);
for i=2:20
    U(1:i-1,i)=A(1:i-1,i);
end
L=A-U;
```

```
x=zeros(20,100);
x(:,1)=rand(20,1);
for i=2:100
    x(:,i)=(L\eye(20))*(b-U*x(:,i-1));
end

norm(x(:,end)-A\b)

plot(x,'b.')
```

最后的收敛图如下图所示

文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)



可以大致看出每个分量都在趋于一点，即这个迭代法是收敛的。

为切实的验证其收敛性，计算了最终得到的 x 和 $A^{-1}b$ 的差的模值。计算结果如下

```
ans =

   2.4086e-16
```

可以看出确实是收敛的。

由上面的实验可以得出高斯塞德尔迭代的适用性应该比雅克比迭代的适用性要广。