Lab 7 实验报告

实验一

设计实验的思路是建立一个函数，使左端项减去右端项。正常情况下应该所有都小于0.如果在实验过程中，得出大于 0时，则记一次数。输出为0则代表所有生成的随机数都满足这个条件。

这个生成随机数的方法是：先生成两个在$[-1,1]\times[-1,1]$上的随机向量$(x_1,x_2)'$，然后使这两个向量的模长小于1即可。代码如下

```
function y=task1_func(a1,a2,b1,b2)
y1=(1+a1)^2*(1-b1^2-b2^2)/((1+b1)^2*(1-a1^2-a2^2));
y2=2*(1-a1*b1-a2*b2)*(1+a1)/((1+b1)*(1-a1^2-a2^2))-1;
y=y1-y2;
end
```

```
close,clc,clear;


count=0;
larger_than_zero=0;
while(count<=1e5)
    a=rand(2,1);
    b=rand(2,1);
    if norm(a)<1 && norm(b)<1
        count=count+1;
        if task1_func(a(1),a(2),b(1),b(2))>0
            larger_than_zero=larger_than_zero+1;
        end
    end
end
```

重复了100000次试验后，得出larger_than_zero的值为0，即证明这个等式成立。

larger_than_zero 0

实验二

（1）

首先先写出这个函数，代码如下

```
function y=task2_1_func(x)
y=3*x^2-exp(x);
end
```

第一种迭代法为二分法，代码如下，得出的结果如下（第一项为二分法算的根，第二项为该根算出的值）

```matlab
%二分法
close,clc,clear;
a1=0;
a2=1;
h=a2-a1;
while(h>=1e-5)
    a3=(a2+a1)/2;
    if(task2_1_func(a3)>0)
        a2=a3;
    elseif(task2_1_func(a3)<0)
        a1=a3;
    end
    h=h/2;
end
a3=(a1+a2)/2;

vpa(a3,10)
vpa(task2_1_func(a3),10)
```

```
ans =

0.9100074768



ans =

-0.000002847215064
```

第二种方法为迭代法，代码如下，得出的结果如下

```
%不动点法

close,clc,clear;
x=1;
while(abs(task2_1_func(x))>1e-5)
    x=exp(x)/(3*x);
end
vpa(x,10)
vpa(task2_1_func(x),10)
```

```
ans =

0.9100104878


ans =

0.000008675213143
```

第三种方法是牛顿迭代法，代码如下，得出的结果如下

```
%牛顿迭代法

close,clear,clc;
x=1;
for i=1:5
    x=x-task2_1_func(x)/(6*x-exp(x));
end

vpa(x,10)
vpa(task2_1_func(x),10)
```

```
ans =

0.9100075725


ans =

4.440892099e-16
```

(2)

设第k次迭代后的位于左边的坐标为a_k，位于右边的坐标为b_k，第k次的根的坐标为$\frac{1}{2}(a_k + b_k)$，用二分法求该方程的根，因为二分法每次都会二分，则

$$|x_k - x^*| \leq \frac{1}{2}(b_k - a_k) = ... = \frac{1}{2^{k+1}}(b - a)$$

所以，可以先设定该根在1.9 ± 0.5附近，则初始误差为1，在经过多次迭代，即不断除以2之后，误差就会降到10^{-13}以下。代码如下

```
close,clc,clear;

a1=1.4;
a2=2.4;
h=a2-a1;
while(h>1e-13)
    a3=(a2+a1)/2;
    if(task2_2_func(a3)>0)
        a2=a3;
    elseif(task2_2_func(a3)<0)
        a1=a3;
    end
    h=h/2;
end

a3=(a1+a2)/2;

vpa(a3,10)
vpa(task2_2_func(a3),15)
```

算出的结果如下

```
ans =

2.001127062


ans =

0.00000000000247251108476121
```

```
h                    5.6843e-14
```

可以看出误差控制在了10^{-13}以内了。

(3)

先写出该函数，具体思路是先生成符号变量x，构建两个向量$\vec{x}=(x,x...,x)'$以及$(1,2...,20)'$，用 $\vec{x}$ 减去第二个向量并用matlab内置函数prod使他们内部全部相乘，得出该函数。

第二步生成循环 i=0~15，使 $\epsilon=10^{-i}$ 即可，解出每个 i 对应的根的情况，并画图表示。代码如下

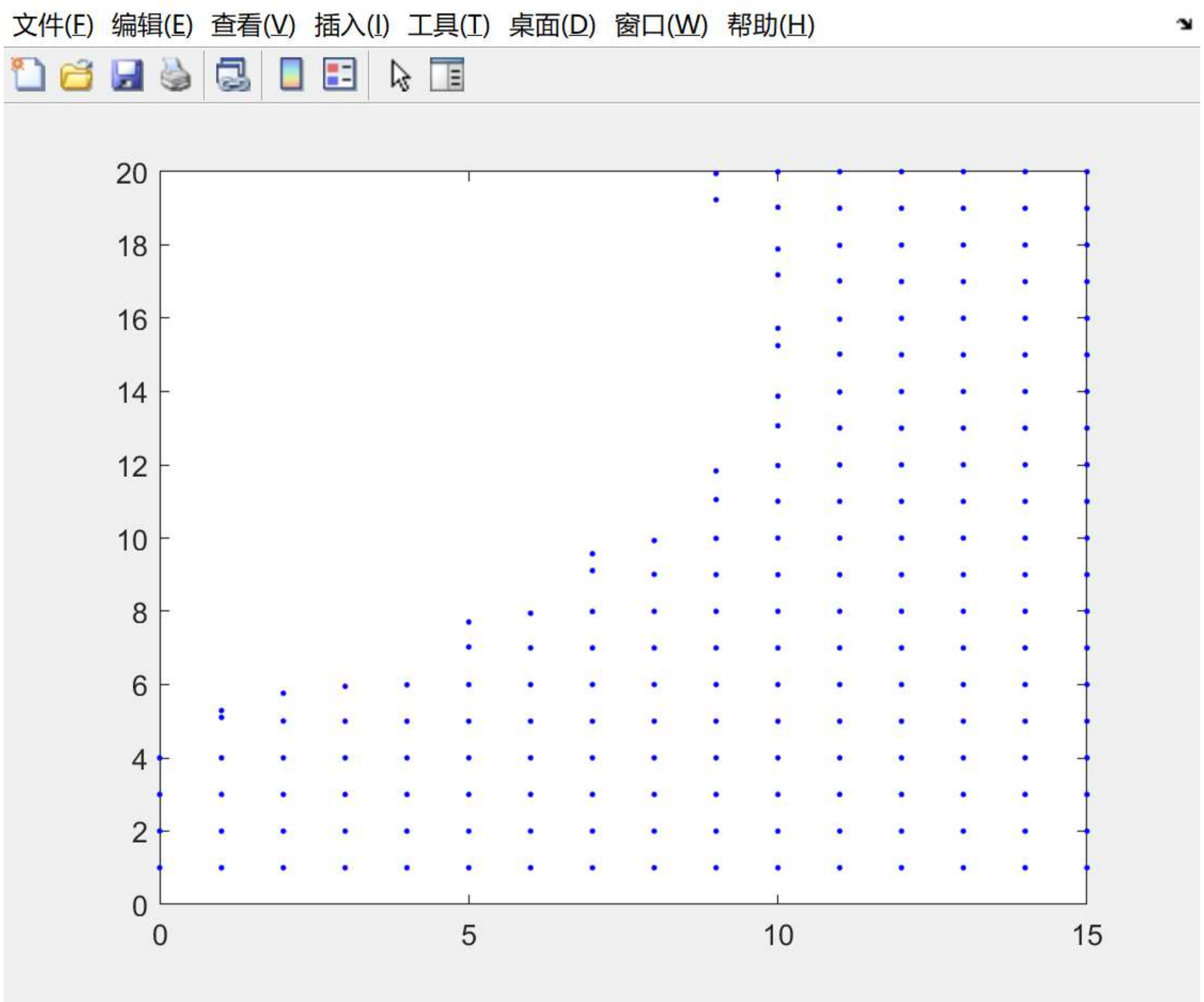
```
close,clc,clear;

x=sym("x");
y=prod(x*ones(1,20)-(1:20));



for i = 0:15
    sym_exp=y+10^(-i)*x^19;
    a=vpa(solve(sym_exp==0,x));
    plot(i*ones(length(a),1),a,'b.')
    hold on
end
```

画出的图如下


文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
20
18
16
14
12
10
8
6
4
2
0
0
5
10
15

可以看出随着 ϵ 的减小，即越来越接近 0 ，根的数量在不断增加（在 10^{-10} 的数量级时已经有 20 个根了），且根的解越接近 $(1, 2..., 20)$.