Matlab代码实现如下：

m=10;

n=4;

b=rand(m,1);

U =orth(rand(m,m));%生成一组标准正交基u

V =orth(rand(n,n));%生成另一组标准正交基v

sig=zeros(m,n);

sig(2,2)=10;%

sig(3,3)=5;

sig(4,4)=1;%最小的奇异值是1

error1=zeros(1,((7-1)/0.001+1));

error2=zeros(1,((7-1)/0.001+1));

maxsigma=zeros(1,((7-1)/0.001+1));%初始化

%%

count=1;

for k= 1:0.001:7

sig(1,1)=10^k;%最大的奇异值从1增加到10^8，按照指数的方式增长

A = U\*sig\*V';

maxsigma(1,count)=10^k;

%直接用matlab内部自带的方法计算伪逆，并认为这个解相对精准

x=pinv(A)\*b;

%方法一,直接解法方程normal equation(Cholesky)

R=chol(A'\*A);

x1=R\(R'\(A'\*b));

%[L,U]=lu(A'\*A);

%x1=U\(L\(A'\*b));

%x1=(A'\*A)\(A'\*b);

error1(1,count)=norm(x1-x);

%方法二，通过Householder方法进行QR分解

%[Q,R]=qr(A);

A2=A;

Q\_inv=eye(m)

for j =1:n %使用householder方法

v=A2(j:m,j);

len=norm(v);

if v(1,1)>=0%此处相当于对v与一个长度为||v||且只有第一个分量不为零的向量作差

v(1,1)=v(1,1)+len;%避免舍入误差

else

v(1,1)=v(1,1)-len;%避免舍入误差

end

H=eye(m-j+1)-2\*(v\*v')./(v'\*v);

A2(j:m,j:n)=H\*A2(j:m,j:n);%Householder反射子只需要作用于schur补即可

H2=[eye(j-1),zeros(j-1,m-j+1);zeros(m-j+1,j-1),H];%分块矩阵，左上角为eye，右下角为H，其他为零

Q\_inv=H2\*Q\_inv;

end

R=A2;

x2=R\(Q\_inv\*b);

%计算x2-x1的范数norm

error2(1,count)=norm(x2-x);

count=count+1;

end

figure(1);

scatter(maxsigma,error1,20,"black","filled");

xlabel("A的最大奇异值(最小奇异值为1)");

ylabel("||x1-x||");

title("对A'\*A进行cholesky分解，再通过法方程求解x1")

figure(2)

scatter(maxsigma,error2,20,"blue","filled");

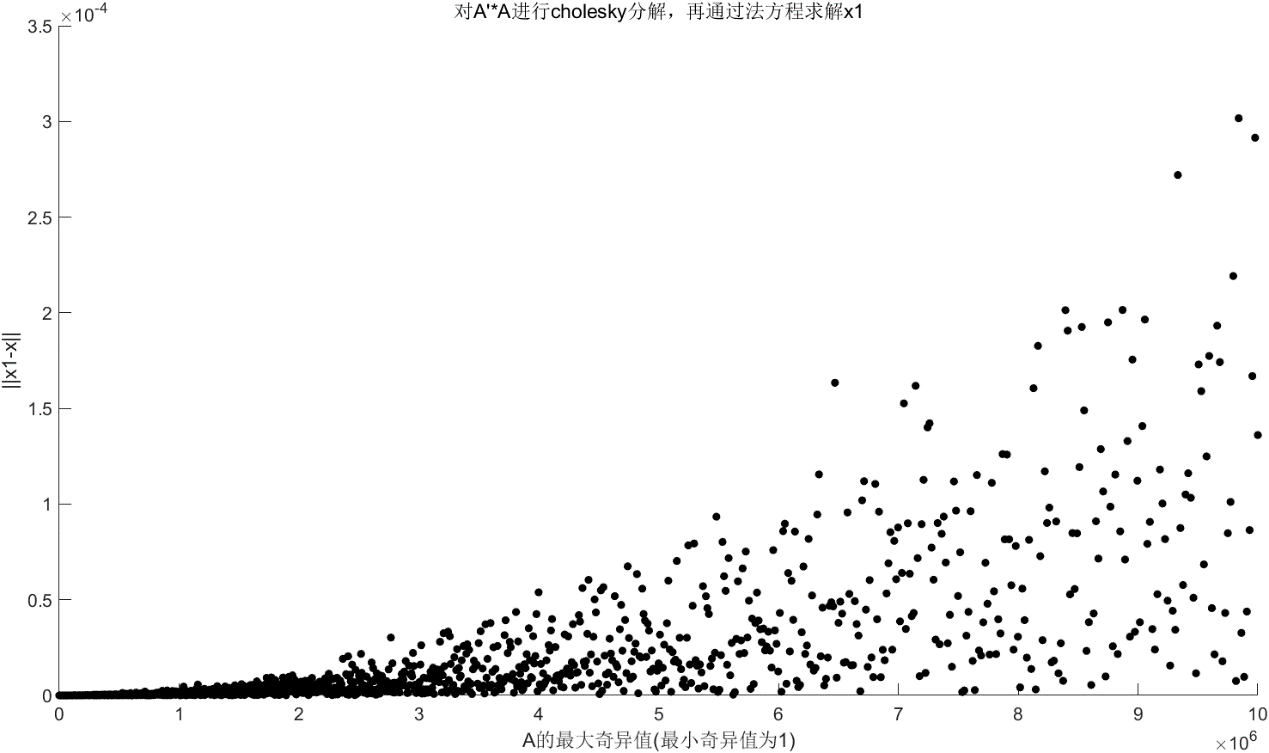
xlabel("A的最大奇异值(最小奇异值为1)");

ylabel("||x2-x||");

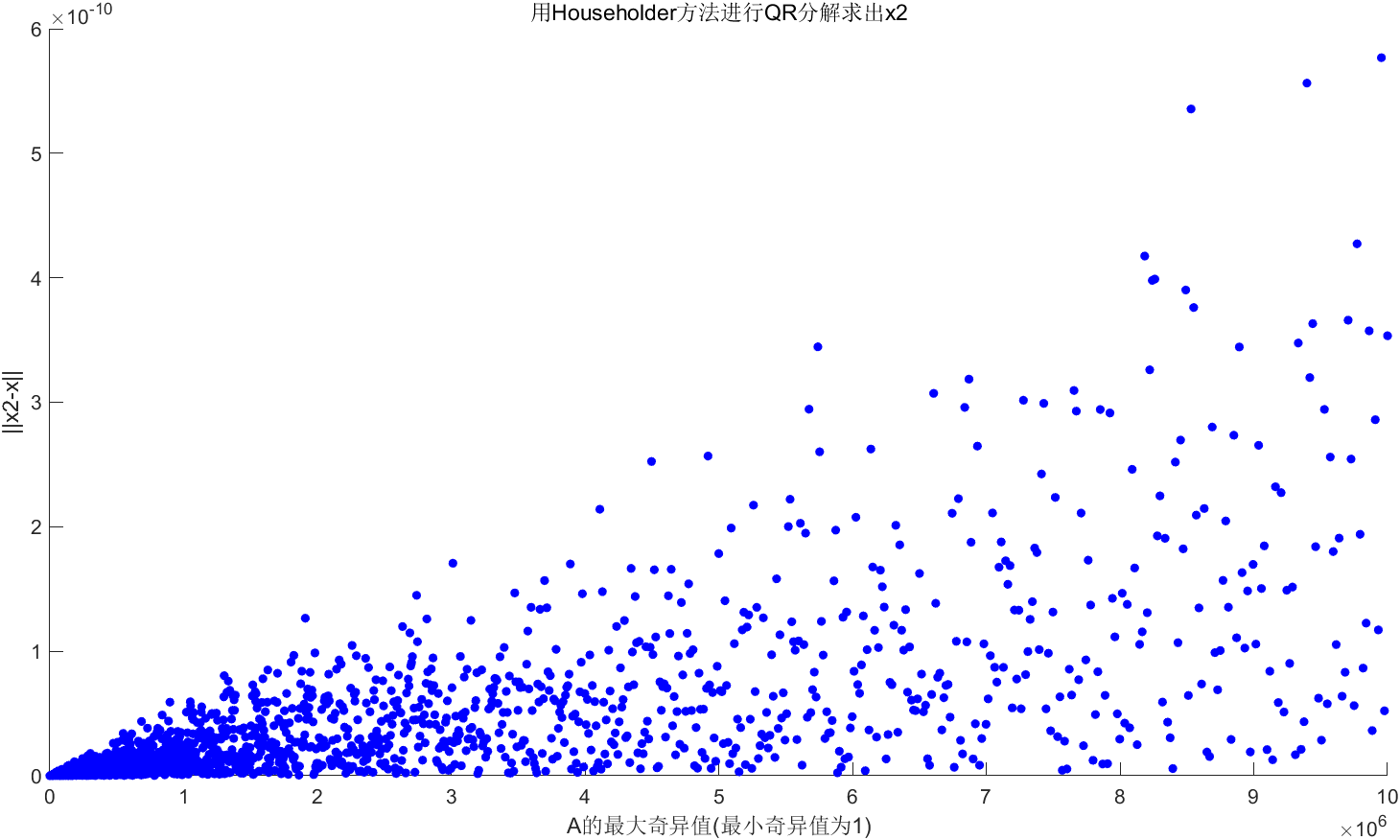
title("用Householder方法进行QR分解求出x2")

直接用matlab自带的方法计算伪逆左乘到b上，得到x，认为这个x较为准确

通过cholesky分解的方法得到x1,计算x1-x的二范数，并以此来衡量x1的误差。A的最小奇异值为1，以A的最大奇异值为横坐标，以x1-x的二范数纵坐标作图：



通过QR分解的方法得到x2,计算x2-x的二范数，并以此来衡量x2的误差。A的最小奇异值为1，以A的最大奇异值为横坐标，以x2-x的二范数纵坐标作图：



当A的最大奇异值很大时，条件数也会相应地增大。但是，QR分解的条件数增加的速度不如用法方程求解的条件数增加的速度快。从误差的数量级也可以看出，QR分解的精确度更高，法方程求解的精确度较低。