数值实验报告（四）

姓名	杜鸿宇	学号	2016141211049	提交日期	2019.4.9
实验名称	第四次数值实验			实验工具	MATLAB

实验内容

1、Page 133 E 3.3;

2、Page 133 E 3.5;

3、Page 135 E 3.11;

4、Page 135 E 3.12.

实验结果

Page 133 E 3.3

(a) 经过计算机计算得到：

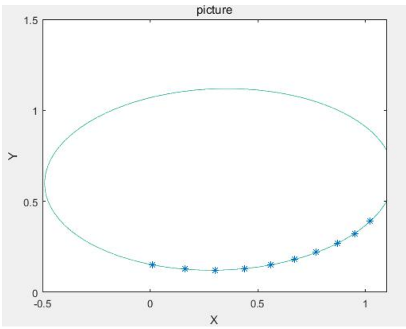
$$time(lu(10000)) \approx 18.4911s$$

(b) 实际的值为：1.4175*10^9。每次计算略有波动，但大体不会变。理论的数量级和值为：1.3333*10^9。我们得到机器值和理论值有一定的误差，但数量级不会改变。

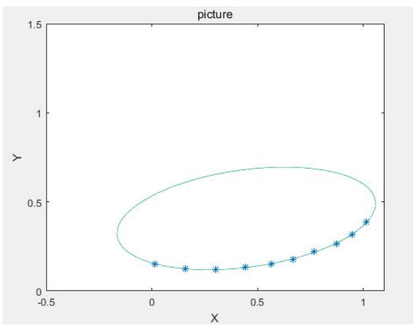
Page 133 E 3.5

(a) 参数值：a=-2.6356，b=0.1436，c=0.5514，d=3.2229，e=-0.4329

图像：



(b) 图像：



参数的新值：a=-4.5050，b=1.2432，c=0.3863，d=3.1119，e=-0.3782

	图像看出即使是微小的扰动，对轨道的变化影响也非常的大。

(c) 经过列主元变换后的 R 矩阵为：

-3.1623	-1.8183	-0.4510	-0.6514	-0.1586
0	-1.0287	-0.3659	-0.2350	-0.1106
0	0	0.1543	0.1447	0.0819
0	0	0	-0.0301	0
0	0	0	0	-0.0063

通过对其对角元的分析我们得到：

当 k=-5 时：rank=5

当 k=-4 时：rank=5

当 k=-3 时：rank=5

当 k=-2 时：rank=4

当 k=-1 时：rank=3

k=-5 时，不扰动系数值：a=-2.6356，b=0.1436，c=0.5514，d=3.2229，e=-0.4329

k=-4 时，不扰动系数值：a=-2.6356，b=0.1436，c=0.5514，d=3.2229，e=-0.4329

k=-3 时，不扰动系数值：a=-2.6356，b=0.1436，c=0.5514，d=3.2229，e=-0.4329

k=-2 时，不扰动系数值：a=0.0000，b=-1.2566，c=0.7655，d=3.2255，e=-0.4890

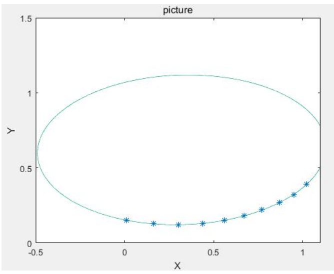
k=-1 时，不扰动系数值：a=0.0000，b=1.7688，c=0.4267，d=0.0000，e=-0.0609

k=5、4、3 扰动系数值：a=-3.9241，b=0.8927，c=0.4360，d=3.1876，e=-0.4017

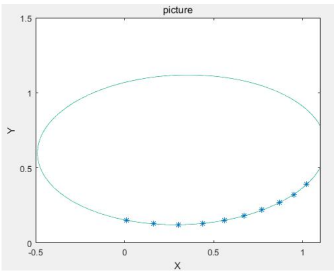
k=2 时，扰动系数值：a=0.0000，b=-1.1542，c=0.7520，d=3.1457，e=-0.4795

k=1 时，扰动系数值：a=0.0000，b=1.7543，c=0.4344，d=0.0000，e=-0.0637

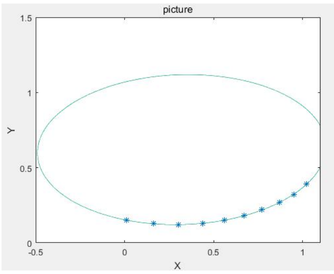
k=-5 时的不扰动图像：



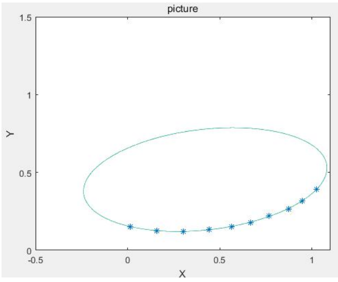
k=-4 时的不扰动图像：



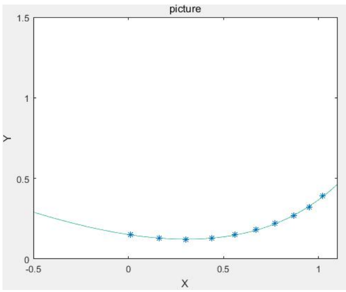
k=-3 时的不扰动图像：



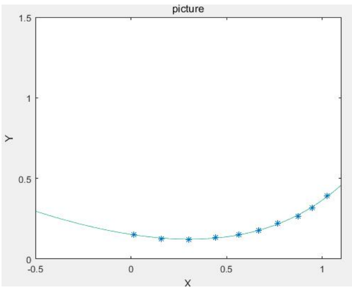
秩不变时候的扰动图像：



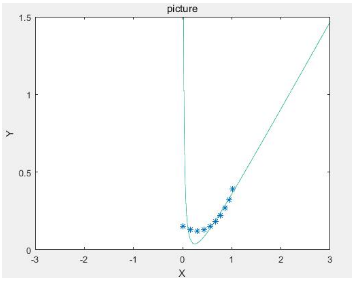
k=-2 时的不扰动图像：



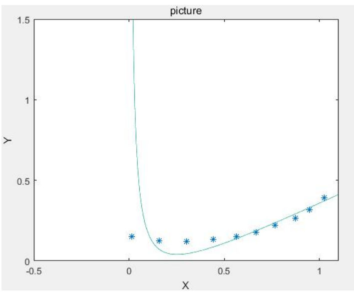
k=-2 时的扰动图像：



k=-1 时的不扰动图像：



k=-1 时的扰动图像：



结论：rank=4 的时候最好，因为这时候的扰动性最小，扰动过后的图与原来的图几乎没有什么区别。若考虑行星的轨道椭圆，则满秩的时候最好，因为这时候的拟合是一个椭圆，满足自然规律。

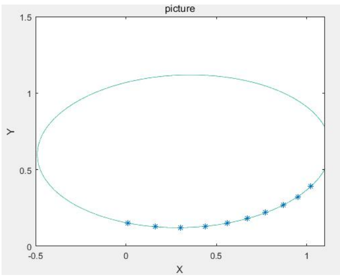
(d) 奇异值矩阵为：

$$\begin{bmatrix} 3.7860 & 0 & 0 & 0 & 0 \\ 0 & 0.9449 & 0 & 0 & 0 \\ 0 & 0 & 0.2089 & 0 & 0 \\ 0 & 0 & 0 & 0.0230 & 0 \\ 0 & 0 & 0 & 0 & 0.0055 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(e) 当 k=5 的时候：

轨道参数值：a=-2.6356，b=0.1436，c=0.5514，d=3.2229，e=-0.4329

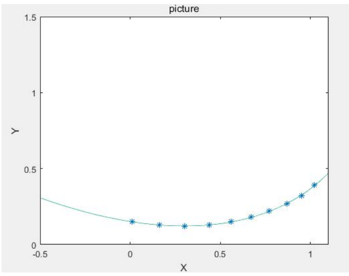
图像：



当 k=4 的时候：

轨道参数：a=-0.6528，b=-0.9451，c=0.7168，d=3.2610，e=-0.4800

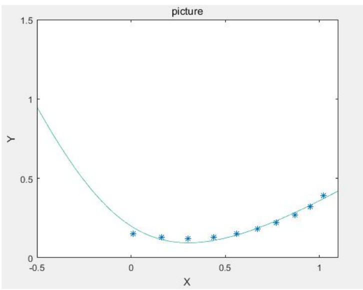
图像：



当 k=3 的时候：

轨道参数：a=0.4192，b=0.8769，c=0.5253，d=0.7561，e=-0.1674

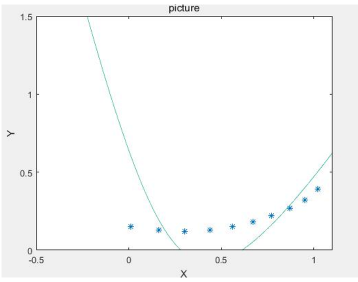
图像：



当 k=2 的时候：

轨道参数：a=0.1044，b=0.3410，c=0.8918，d=0.2025，e=-0.1718

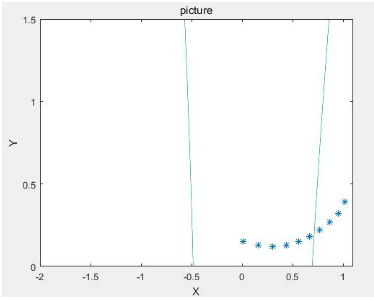
图像：



当 k=1 的时候：

轨道参数：a=0.0190，b=0.0547，c=0.2117，d=0.0737，e=0.3361

图像：



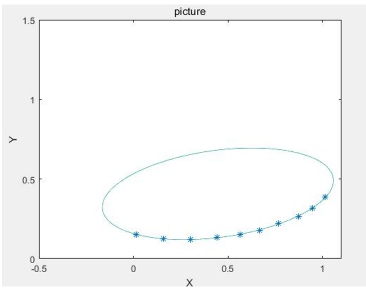
(f) 扰动之后的奇异值矩阵：

$$\begin{bmatrix} 3.7837 & 0 & 0 & 0 & 0 \\ 0 & 0.9374 & 0 & 0 & 0 \\ 0 & 0 & 0.2044 & 0 & 0 \\ 0 & 0 & 0 & 0.0235 & 0 \\ 0 & 0 & 0 & 0 & 0.0060 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

当 k=5 的时候：

轨道参数值：a=-4.5050，b=1.2432，c=0.3863，d=3.1119，e=-0.3782

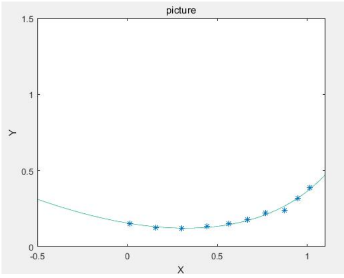
图像：



当 k=4 的时候：

轨道参数：a=-0.4879，b=-0.8564，c=0.7096，d=3.0780，e=-0.4588

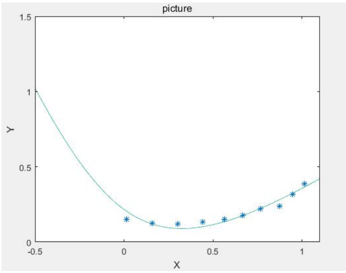
图像：



当 k=3 的时候：

轨道参数：a=0.4205，b=0.8797，c=0.5263，d=0.7614，e=-0.1692

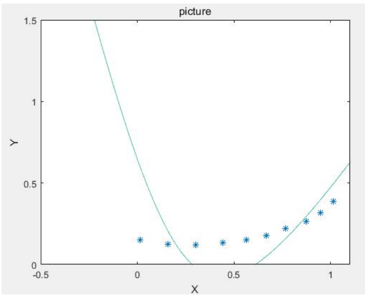
图像：



当 k=2 的时候：

轨道参数：a=0.1029，b=0.3378，c=0.8933，d=0.2012，e=-0.1729

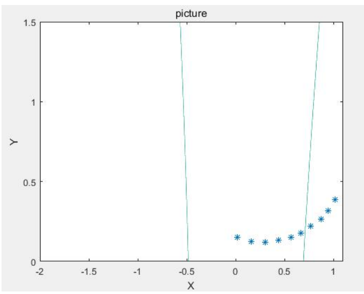
图像：



当 k=1 的时候：

轨道参数：a=0.0186，b=0.0540，c=0.2105，d=0.0729，e=0.3348

图像：



在扰动过后，轨道曲线会有所减小，且与真实点之间的误差会有所加大。主要是原来矩阵的条件数过大，所以非常敏感，原来不改动的要更好一点。

(g) 完全最小二乘得到的参数：

a=-0.1430，b=0.0201，c=0.0220，d=0.1420，e=-0.0185

Page 135 E 3.11

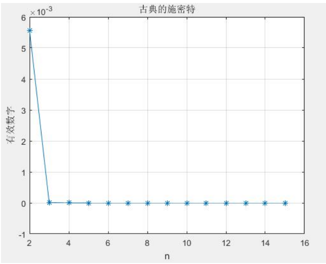
本题选取的矩阵即为书上的例题，得到的矩阵满足关系式：

$$Q * \begin{bmatrix} R \\ 0 \end{bmatrix} = A$$

所以豪斯霍德尔下的 QR 分解非常成功。

Page 135 E 3.12

a．古典格拉姆施密特：

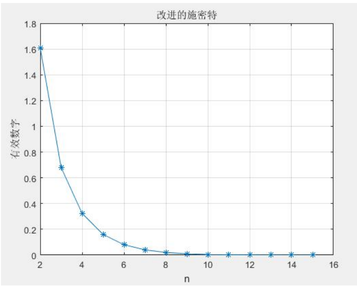


耗时：0.230s

存储量：较大

精确度：非常不好

改进格拉姆施密特：

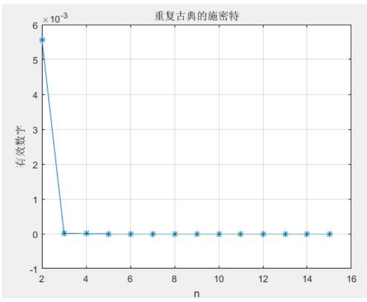


耗时：0.216s

存储量：较小

精确度：有待提高

重复的古典格拉姆施密特：

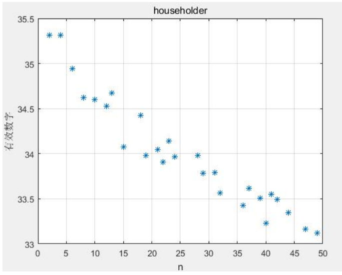


耗时：0.245s

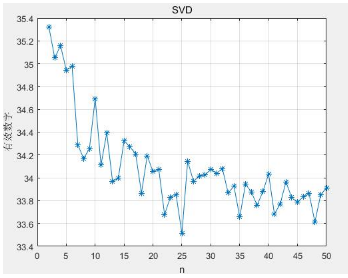
存储量：大

精确度：非常不好

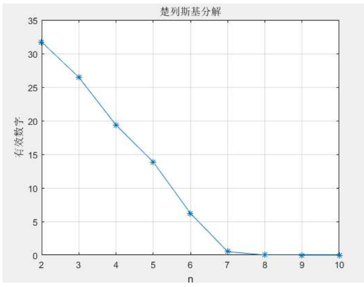
b.豪斯霍尔德方法下希尔伯特矩阵 QR 分解的正交损失图像：



c.SVD 方法下希尔伯特矩阵 QR 分解的正交损失图像：



d.楚列斯基分分解下的的希尔伯特矩阵的正交损失图像：



实验分析

Page 133 E 3.3

a.本题选择的步数是 5 步，即从 100 到 500。需要计算的时间是 LU 分解的时间，即：

$$LU(A)$$

由于题目中要求是三次拟合，所以我们得到如下的最小线性二乘单一等式：

$$[n^3, n^2, n, 1]*[a,b,c,d]=t1$$

其中 a,b,c,d 是需要求的拟合公式的系数，那么我们得到如下的线性最小二乘方程组：

$$\begin{bmatrix} 100^3 & 100^2 & 100 & 1 \\ 200^3 & 200^2 & 200 & 1 \\ 300^3 & 300^2 & 300 & 1 \\ 400^3 & 400^2 & 400 & 1 \\ 500^3 & 500^2 & 500 & 1 \end{bmatrix} * \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} t1 \\ t2 \\ t3 \\ t4 \\ t5 \end{bmatrix}$$

解出系数参数，就可以得到拟合的多项式：

$$a*n^3+b*n^2+c*n+1=f(t)$$

b.本题的思路是选取一个 1000*1000 的全是 1 的矩阵：

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \ldots & \ldots & \ldots \\ 1 & \ldots & \ldots & \ldots \\ 1 & \ldots & \ldots & 1 \end{bmatrix}_{1000*1000}$$

接着考虑两个矩阵相乘的时间 t1，即每一行乘以每一列有 1000 次乘法，999 次加法，这样我们就得到了总的乘和加的步数：1999000000。接着用 t1/1999000000=t2。t2 是一个基本步的时间。再用库函数得到对其 LU 分解的时间 t3，用 t3/t2 就得到了计算量的主项。

而误差主要是两个原因引起的。第一个是因为在计算单个乘法和加法时作用的方法

求得的时间有误差。其次 LU 分解的时间也有误差。但都是小的误差，所以实际值和理论值不会有主项上的差别。

Page 135 E 3.11

b.扰动的估计可以用以下公式来进行衡量：

$$\frac{\|\Delta x\|_2}{\|x\|_2} \le cond(A) \bullet \frac{1}{\cos\theta} \bullet \frac{\|\Delta b\|_2}{\|b\|_2}$$

经过计算初始矩阵 A 的条件数为：

$$cond(A) = 688.4294$$

这个条件数非常的大，所以最小二乘问题的解的敏感性非常的强。所以小的变动也会引起图像大的变化。

c.当矩阵的秩不变的时候，扰动行大的原因在于原矩阵的条件数过大，及

$$cond(A) \ge 600$$

当矩阵的秩减少时，对解有影响的 A 的部分减少，这时 cond(A)就会减小，所以扰动行更小。而矩阵的秩再减小的时候，有效信息因为误差过大会被舍去，所以图像改变太大，拟合效果不好。综上所述，rank=4 的时候最佳。

e.用奇异值分解求解主要用到了以下两个公式：

$$A = U\Sigma V^T = [U_1, U_2]\begin{bmatrix}\Sigma_1 \\ 0\end{bmatrix}V^T = U_1\Sigma_1 V^T$$

$$x = V\Sigma_1^{-1}U_1^T b$$

f.我们来看扰动前后两个矩阵的条件数：

$$cond(A) = 688.4294$$

$$cond(A1) = 629.6535$$

我们发现扰动之后并没有使敏感度减小多少，相反精确度有所下降。而评判一个最小二乘的优略主要是拟合程度。所以拟合好的优先考虑。所以没有扰动的更好！

g.完全最小二乘考虑如下矩阵的 SVD 分解：

$$[A, b]$$

接着用：

$$[A,b] = U\Sigma V^T = [U_1, U_2]\begin{bmatrix}\Sigma_1 \\ 0\end{bmatrix}V^T = U_1\Sigma_1 V^T$$

$$x = V\Sigma_1^{-1}U_1^T b$$

就可以求解改进后的最小二乘值。

Page 135 E 3.11

本题的代码主要针对的就是一般的豪斯霍尔德变换，只是对 QR 分解中的 Q 进行了单独的存储，具体操作如下：

首先把 Q 存储为单位矩阵；

接着采用：Q=H*Q；

直到不需要迭代停止。

Page 135 E 3.12

a.我们先分析古典的格拉姆施密特和重复的古典格拉姆施密特方法。因为希尔伯特矩阵的条件数太大，所以希尔伯特矩阵极度地不稳定。在古典的正交方法之下，从第三列往后的正交系数非常的小，所以正交矩阵非常的奇异，所以有效值很小，及误差非常的大。

对于改进的格拉姆施密特方法，效果好很多，除了最后一列是奇异值外，别的都非常好。在具体操作中我选择的矩阵为：

	$A(1:end-1,1:end-1)$ 排除最后一行最后一列，就得到了非常好的结果。 在时间上，三者并没有太大的差别，改进过后的要稍微好一点。 存储量上，改进的要小很多，因为它是直接往后计算，不需要对之前的重新存储，而古典的如下： $b_n = a_n - \sum_{i=1}^{n-1} \frac{<a_i, b_i>}{<b_i, b_i>} b_i$ 需要重复之前的工作，重复古典的不用多说，工作量就更大了。 e.首先我们非常清楚希尔伯特矩阵是非常不稳定的矩阵，因为： $cond(H) \le \inf$ 在这种情况下由于随着 n 的不断变大，条件数过大，矩阵 H 的正交性被进一步降低，再由于正规矩阵本身带有误差，设对 H 的扰动为 E，由公式得： $(H+E)^T(H+E)(x+\Delta x) = (H+E)^T b$ 所以随着 n 变大，E 也越大，当 n 大于等于 10 的时候，计算机的内置楚列斯基分解分解已经无法对以下公式进行分解： $A^T A = LL^T$ 另一方面受敏感性影响，从图像可以看出，“有效数字”下降非常块，这不是一个好消息。从公式出发： $-\ln(\Vert I - Q^T Q \Vert)$ 值越小，里面的越大，那么 Q 的正交性越差，所以楚列斯基分解不是特别的好。 而 SVD 和豪斯霍尔德方法就要好的多。因为它们可以从矩阵的对角元中 Q 和 M 中消去奇异值。即 $\begin{matrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 0 \end{matrix}$ 这样可以减少运行时间（出现了大流量的零，稀疏矩阵的运行速度要快很多），提高精确度。从图像也可以看出这两种分解下的“有效数字”减小速度缓慢了很多。

代码附件：

Page 133 E 3.3

(a)

```
a=[];
j=1;
for n=100:100:500
    B=rand(n);
    b=[];
    c=0;
    for i=1:10
        t1=clock;
        lu(B);
        t2=clock;
        b(1,i)=etime(t2,t1);
        c=c+b(1,i);
    end
```

```
        a(1,j)=c/10;
        j=j+1;
end
A=[];
for i=1:5
        g=3;
        for k=1:4
                A(i,k)=(i^g)*(100^g);
                g=g-1;
        end
end
Q=eye(5);
a1=A(:,1);
for i=1:4
        e=zeros(5,1);
        e(i,1)=1;
        s2=norm(a1,2);
        v=a1-s2*e;
        H=eye(5)-(2*v*v')/(v'*v);
        Q=H*Q;
        A=H*A;
        if i==4
                break;
        end
        a1=A(:,i+1);
        for j=1:i
                a1(j,1)=a1(j,1)-A(j,i+1);
        end
end
Q1=Q*a';
A1(1,:)=A(1,:);
A1(2,:)=A(2,:);
A1(3,:)=A(3,:);
A1(4,:)=A(4,:);
Q2(1:4,1)=Q1(1:4,1);
x=inv(A1)*Q2;
x3=(10000^3)*x(1,1)+(10000^2)*x(2,1)+(10000)*x(3,1)+x(4,1)
```

(b)

```
a=ones(1000);
t1=clock;
a*a;
t2=clock;
x1=etime(t2,t1);
t3=clock;
```

```
lu(a);
t4=clock;
x2=etime(t4,t3);
x2/(x1/1999000000)
```

Page 133 E 3.5:

(a)

```
x=[1.02,0.95,0.87,0.77,0.67,0.56,0.44,0.30,0.16,0.01];
y=[0.39,0.32,0.27,0.22,0.18,0.15,0.13,0.12,0.13,0.15];
A=[];
b=[];
for i=1:10
    A(i,1)=y(i)^2;
    A(i,2)=y(i)*x(i);
    A(i,3)=x(i);
    A(i,4)=y(i);
    A(i,5)=1;
    b(i,1)=x(i)^2;
end
[U,S,V]=svd(A);
x1=inv(A'*A)*A'*b;
syms X Y
ezplot(x1(1)*Y^2+x1(2)*X*Y+x1(3)*X+x1(4)*Y+x1(5)==X^2,[-0.5,1.1],[0,1.5])
title('picture');
hold on
plot(x,y,'*')
```

(b)

```
x=[1.02-0.005,0.95-0.004,0.87+0.003,0.77-0.0043,0.67-0.005,0.56+0.005,0.44+0.0011,0.30+0.0034,0.16-0.0012,0.01+0.005];
y=[0.39-0.0032,0.32-0.0041,0.27-0.0037,0.22+0.0022,0.18-0.0011,0.15+0.001,0.13+0.0029,0.12-0.0017,0.13-0.0033,0.15+0.001];
A=[];
b=[];
for i=1:10
    A(i,1)=y(i)^2;
    A(i,2)=y(i)*x(i);
    A(i,3)=x(i);
    A(i,4)=y(i);
    A(i,5)=1;
    b(i,1)=x(i)^2;
end
[U,S,V]=svd(A);
x1=inv(A'*A)*A'*b;
syms X Y
```

```
ezplot(x1(1)*Y^2+x1(2)*X*Y+x1(3)*X+x1(4)*Y+x1(5)==X^2,[-0.5,1.1],[0,1.5])
title('picture');
hold on
plot(x,y,'*')
```

(c)

不带扰动的列主元（rank=4）：

```
x=[1.02,0.95,0.87,0.77,0.67,0.56,0.44,0.30,0.16,0.01];
y=[0.39,0.32,0.27,0.22,0.18,0.15,0.13,0.12,0.13,0.15];
A=[];
b=[];
for i=1:10
    A(i,1)=y(i)^2;
    A(i,2)=y(i)*x(i);
    A(i,3)=x(i);
    A(i,4)=y(i);
    A(i,5)=1;
    b(i,1)=x(i)^2;
end
[Q,R,E]=qr(A);
b1=Q'*b;
b2=b1(1:4,1);
R=R(1:4,1:4);
x2=inv(R)*b2;
x3=[x2;0];
x1=E*x3
syms X Y
ezplot(x1(1)*Y^2+x1(2)*X*Y+x1(3)*X+x1(4)*Y+x1(5)==X^2,[-3,3],[0,1.5])
title('picture');
hold on
plot(x,y,'*')
```

(e)

```
x=[1.02,0.95,0.87,0.77,0.67,0.56,0.44,0.30,0.16,0.01];
y=[0.39,0.32,0.27,0.22,0.18,0.15,0.13,0.12,0.13,0.15];
A=[];
b=[];
for i=1:10
    A(i,1)=y(i)^2;
    A(i,2)=y(i)*x(i);
    A(i,3)=x(i);
    A(i,4)=y(i);
    A(i,5)=1;
    b(i,1)=x(i)^2;
end
[U,S,V]=svd(A);
```

```
U1=U(:,1:5);
S1=S(1:5,1:5);
S1(1,1)=1/S1(1,1);
S1(2,2)=1/S1(2,2);
S1(3,3)=1/S1(3,3);
S1(4,4)=0;
S1(5,5)=0;
x1=V*S1*U1'*b
syms X Y
ezplot(x1(1)*Y^2+x1(2)*X*Y+x1(3)*X+x1(4)*Y+x1(5)==X^2,[-0.5,1.1],[0,1.5])
title('picture');
hold on
plot(x,y,'*')
```

(f)

```
x=[1.02-0.005,0.95-0.004,0.87+0.003,0.77-0.0043,0.67-0.005,0.56+0.005,0.44+0.0011,0.30+0.0034,0.16-0.0012,0.01+0.005];
y=[0.39-0.0032,0.32-0.0041,0.27-0.0037,0.22+0.0022,0.18-0.0011,0.15+0.001,0.13+0.0029,0.12-0.0017,0.13-0.0033,0.15+0.001];
A=[];
b=[];
for i=1:10
    A(i,1)=y(i)^2;
    A(i,2)=y(i)*x(i);
    A(i,3)=x(i);
    A(i,4)=y(i);
    A(i,5)=1;
    b(i,1)=x(i)^2;
end
[U,S,V]=svd(A);
U1=U(:,1:5);
S1=S(1:5,1:5);
S1(1,1)=1/S1(1,1);
S1(2,2)=1/S1(2,2);
S1(3,3)=1/S1(3,3);
S1(4,4)=0;
S1(5,5)=0;
x1=V*S1*U1'*b
syms X Y
ezplot(x1(1)*Y^2+x1(2)*X*Y+x1(3)*X+x1(4)*Y+x1(5)==X^2,[-0.5,1.1],[0,1.5])
title('picture');
hold on
plot(x,y,'*')
```

(g)

```
x=[1.02,0.95,0.87,0.77,0.67,0.56,0.44,0.30,0.16,0.01];
```

```
y=[0.39,0.32,0.27,0.22,0.18,0.15,0.13,0.12,0.13,0.15];
A=[];
b=[];
for i=1:10
    A(i,1)=y(i)^2;
    A(i,2)=y(i)*x(i);
    A(i,3)=x(i);
    A(i,4)=y(i);
    A(i,5)=1;
    b(i,1)=x(i)^2;
end
A1=[A,b];
[U,S,V]=svd(A1);
S1=S(1:5,1:5);
S1(1,1)=1/S1(1,1);
S1(2,2)=1/S1(2,2);
S1(3,3)=1/S1(3,3);
S1(4,4)=1/S1(4,4);
S1(5,5)=1/S1(5,5);
V(1:end-1,1:5)*S1*U(:,1:5)'*b
```

Page 135 E 3.11:

```
A=[1,0,0;0,1,0;0,0,1;-1,1,0;-1,0,1;0,-1,1];
Q=eye(6);
a=A(:,1);
for i=1:3
    e=zeros(6,1);
    e(i,1)=1;
    s2=norm(a,2);
    v=a-s2*e;
    H=eye(6)-(2*v*v')/(v'*v);
    Q=H*Q;
    A=H*A
    if i==3
        break;
    end
    a=A(:,i+1);
    for j=1:i
        a(j,1)=0;
    end
end
```

Page 135 E 3.11:

(a)

古典格拉姆施密特：

```
b=[];
```

```
s=1;
for n=2:15
    A=hilb(n);
    for i=2:n
        for j=1:i-1
            A(:,i)=A(:,i)-(A(:,i)'*A(:,j))*A(:,j)/(A(:,j)'*A(:,j));
        end
    end
    A
    Q=-log(norm(eye(n)-A'*A));
    b(s)=Q;
    s=s+1;
end
plot([2:15],b,'-*')
grid on;
title('古典的施密特');
xlabel('n');
ylabel('有效数字');
```

改进格拉姆施米特：

```
b=[];
i=1;
for n=2:10
    A=hilb(n);
    L=chol(A'*A);
    Q=A*inv(L');
    a=eye(n)-Q'*Q;
    n=norm(a,2);
    t=-log(n);
    b(i)=t;
    i=i+1;
    clear Q
end
plot([2:10],b,'-g*')
title('楚列斯基分解');
xlabel('n');
ylabel('有效数字');
grid on;
```

重复的古典格拉姆施密特：

```
b=[];
s=1;
for n=2:15
    A=hilb(n);
    for z=1:2
        for i=2:n
```

```
            for j=1:i-1
                A(:,i)=A(:,i)-(A(:,i)'*A(:,j))*A(:,j)/(A(:,j)'*A(:,j));
            end
        end
    end
    A
    Q=-log(norm(eye(n)-A'*A));
    b(s)=Q;
    s=s+1;
end
plot([2:15],b,'-*')
grid on;
title('重复古典的施密特');
xlabel('n');
ylabel('有效数字');
```

(b)

```
v1=1;
b=[];
for n=2:1:50
    A=hilb(n);
    Q=eye(n);
    a=A(:,1);
    for i=1:n
        e=zeros(n,1);
        e(i,1)=1;
        s2=norm(a,2);
        v=a-s2*e;
        H=eye(n)-(2*v*v')/(v'*v);
        Q=H*Q;
        A=H*A;
        if i==n
            break;
        end
        a=A(:,i+1);
        for j=1:i
            a(j,1)=0;
        end
    end
    a1=eye(n)-Q'*Q;
    m=norm(a1,2);
    t=-log(m);
    b(v1)=t;
    v1=v1+1;
end
```

```
plot([2:50],b,'*')
title('householder')
xlabel('n');
ylabel('有效数字');
grid on;
```

(c)

```
b=[];
i=1;
for n=2:50
    A=hilb(n);
    [U,S,V]=svd(A);
    Q=U(:,1:n);
    a=eye(n)-Q'*Q;
    n=norm(a,2);
    t=-log(n);
    b(i)=t;
    i=i+1;
    clear Q U
end
plot([2:50],b,'-r*')
title('SVD');
xlabel('n');
ylabel('有效数字');
grid on;
```

(d)

```
b=[];
i=1;
for n=2:10
    A=hilb(n);
    L=chol(A'*A);
    Q=A*inv(L);
    a=eye(n)-Q'*Q;
    n=norm(a,2);
    t=-log(n);
    b(i)=t;
    i=i+1;
    clear Q
end
plot([2:10],b,'-g*')
title('楚列斯基分解');
xlabel('n');
ylabel('有效数字');
grid on;
```