MATLAB 第 5 次作业参考答案

几点反馈：

- 作业的参考答案来自同学们的作业，只是稍作调整，为保护隐私，隐去姓名。
- 在 WORD 里编辑难免出现笔误，建议大家交作业时有 M 文件就提交上。

第一题

Hw5_1.m

```
clear;clc
A=[4,2,3,0;
    -2,3,-1,1;
    1,3,-4,2;
    1,0,1,-1;
    3,1,3,-2];
b=[10;0;2;0;5];
x=(transpose(A)*A)\(transpose(A)*b);
fprintf("法线方程的解\n")
disp(x)

r=b-A*x;
r_norm=norm(r);
SE=r_norm^2;
RMSE=r_norm/sqrt(length(r));
fprintf("2_norm\n")
disp(r_norm)
fprintf("SE\n")
disp(SE)
fprintf("RMSE\n")
disp(RMSE)
```

执行结果如下：

命令行窗口

```
法线方程的解
    1.2739
    0.6885
    1.2124
    1.7497

2_norm
    0.8256

SE
    0.6817

RMSE
    0.3692
```

题干要求输出结果，有的同学没有按要求在命令行输出结果

第二题

Hw5_2_1.m

```
clear;clc
load('2_t.mat')
load('2_y.mat')

A=[ones(25,1),t];
b=y;
c=(transpose(A)*A)\(transpose(A)*b);
fprintf("c1 c2 分别为\n")
disp(c')
f=@(t) c(1)+c(2).*t;
u=linspace(1,25,1000);
v=f(u);
figure
plot(t,y,'o',u,v)
xlabel('t')
ylabel('y')
title('拟合效果图')
legend('raw data','linear fit')
figure
r=f(t)-y;
plot(t,r,'--','Marker','o')
xlabel('t')
ylabel('残差')
title('残差图')
```

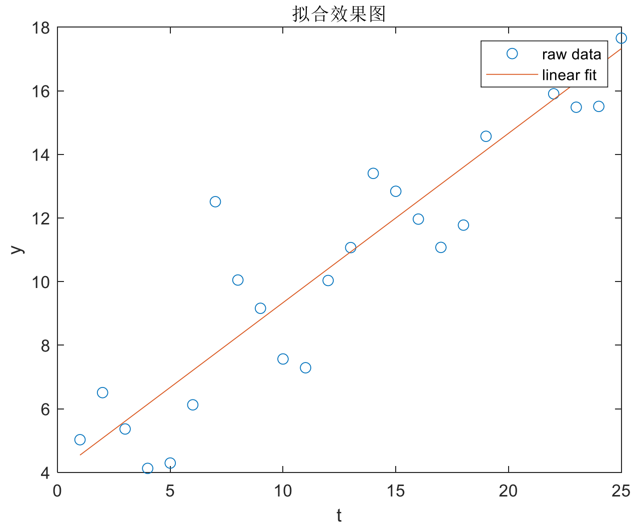
Hw5_2_2.m

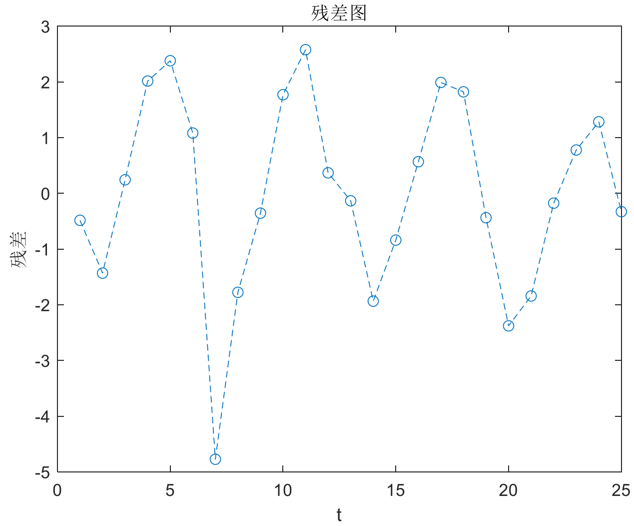
```
clear;clc
load('2_t.mat')
load('2_y.mat')

t1=t;
y1=y;
t1(7)=[];
y1(7)=[];
A1=[ones(24,1),t1,sin(t1)];
b=y1;
c=(transpose(A1)*A1)\(transpose(A1)*b);
fprintf("c1 c2 c3 分别为\n")
disp(c')
f=@(t) c(1)+c(2).*t+c(3).*sin(t);
u=linspace(1,25,1000);
v=f(u);
figure
plot(t1,y1,'o',u,v)
hold on
plot(t(7),y(7),'*')
xlabel('t')
ylabel('y')
title('拟合效果图')
hold off
legend('raw data','linear fit','离群点')
figure
r1=f(t1)-y1;
plot(t1,r1,'--','Marker','o')
hold on
plot(t(7),f(t(7))-y(7),'*')
hold off
xlabel('t')
ylabel('残差')
title('残差图')
legend('数据点','离群点')
```

执行 Hw5_2_1.m，得到如下结果：

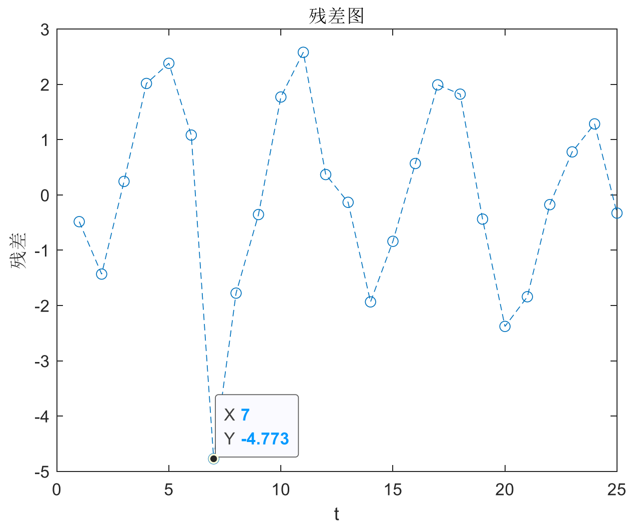
命令行窗口

```
c1  c2分别为
      4.0127      0.5326
```





离群点如图所示，为$t = 7, y = -4.773$



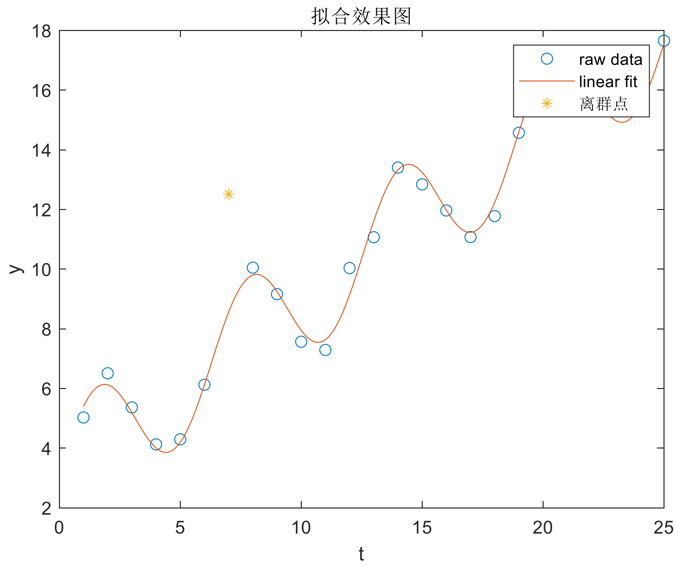
去除离群点，观察到峰值剩余点的分布与正弦函数 $y=\sin\omega t$ 相近，并且，相邻峰值之间的距离约为 6， $6\approx 6.28=2\pi$ ，猜测正弦函数的周期为 2π ，因此取 $\omega=1$。因此，取

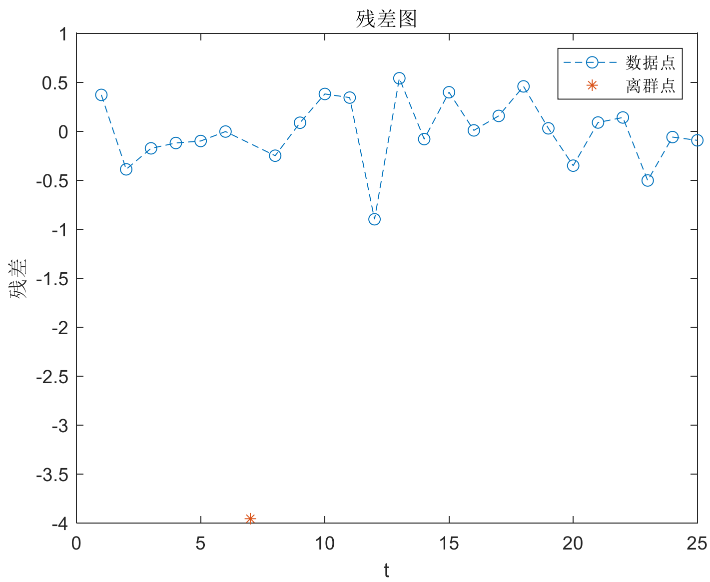
$$g(t)=\sin(t)$$

执行 Hw5_2_2.m，得到如下结果：

命令行窗口

```
c1  c2  c3分别为
      3.1536       0.5869       1.9733
```





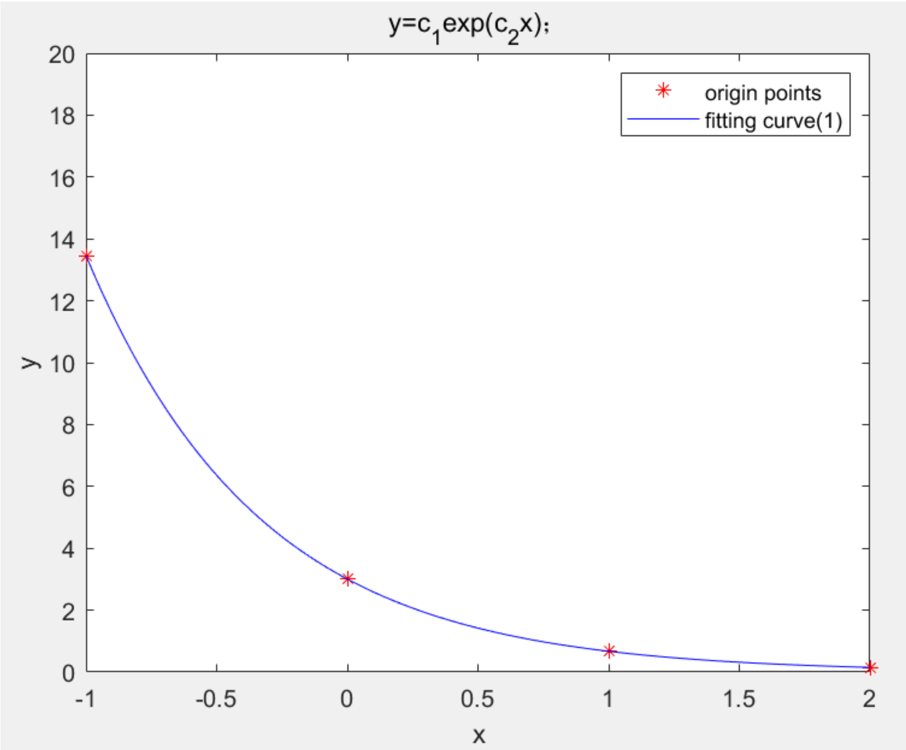
第二题有的同学没有猜对 g(t)，造成后面的拟合错误

第三题

```
hw5_3.m
clear,clc;
x=[-1;0;1;2];
y=[13.45;3.01;0.67;0.15];
n=length(x);
%(1)
A=[ones(n,1),x];
b1=log(y);
c1=(A'*A)\(A'*b1);
c1(1)=exp(c1(1));
fprintf('曲线(1)：c1=%f,c2=%f\n',c1(1),c1(2));

u=linspace(-1,2);
v1=c1(1)*exp(c1(2)*u);
figure(1);
plot(x,y,'r*',u,v1,'b-');
legend('origin points','fitting curve(1)');
xlabel 'x';ylabel 'y';
axis([-inf,inf,0,20]);
title 'y=c_1exp(c_2x)';
%(2)
B=[x,ones(n,1)];
b2=1./sqrt(y);
c2=(B'*B)\(B'*b2);
fprintf('曲线(2)：c1=%f,c2=%f\n',c2(1),c2(2));

v2=1./(c2(1)*u+c2(2)).^2;
figure(2);
plot(x,y,'r*',u,v2,'k-');
legend('origin points','fitting curve (2)');
xlabel 'x';ylabel 'y';
axis([-inf,inf,0,20]);
title 'y=1/(c_1x+c_2)^2';

%(3)
r1=c1(1)*exp(c1(2)*x)-y;
rmse1=sqrt(sum(r1.^2)/length(r1));
fprintf('RMSE1=%f\n',rmse1);
r2=1./(c2(1)*x+c2(2)).^2-y;
rmse2=sqrt(sum(r2.^2)/length(r2));
fprintf('RMSE2=%f\n',rmse2);
```

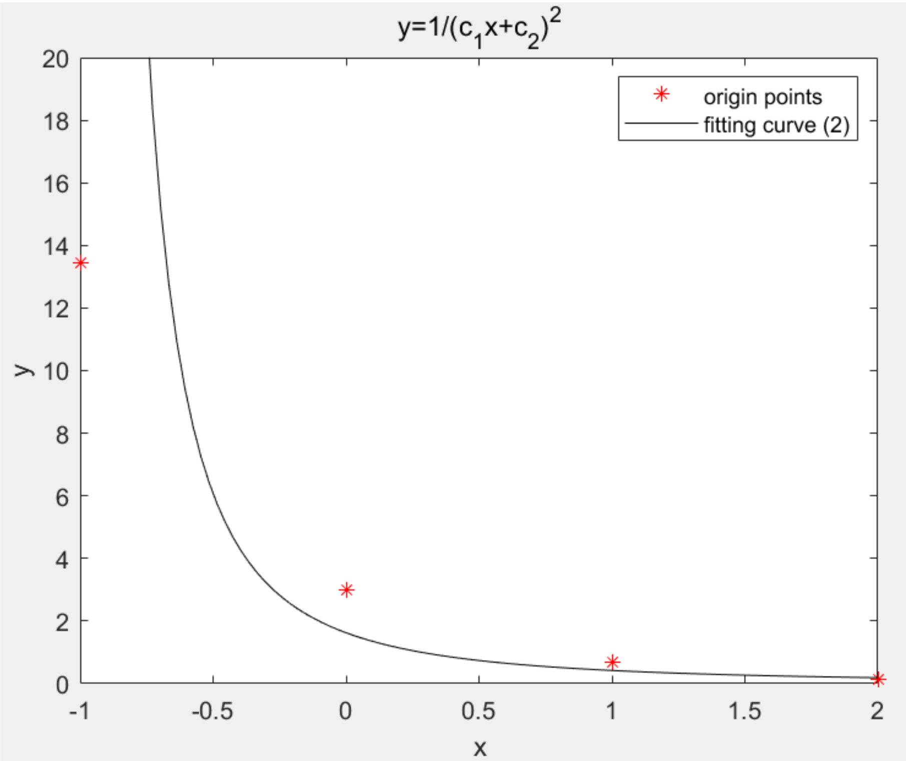
（1）

```
曲线(1)：c1=3.005266,c2=-1.499071
```



（2）

曲线(2)：c1=0.757326,c2=0.784523



（3）

RMSE1=0.003933

RMSE2=669.221864

显然第一条曲线拟合得更好

第三题是一个非线性最小二乘问题，可以用模型线性化、高斯-牛顿法、lsqnonlin、lsqcurvefit 来做，这里给的是模型线性化的解法，其他解法也很好，用的话都需要注意初值的选取，建议大家在做的时候可以多试试不同方法

第四题

hw5_4.m

```
clear,clc;
A=[3,-1,2;4,1,0;-3,2,1;1,1,5;-2,0,3];
b=[10;10;-5;15;0];
[Q,R]=QR_Houref(A);
[q,r]=qr(A);
disp('Q=');disp(Q);
disp('q=');disp(q);
disp('R=');disp(R);
disp('r=');disp(r);
x=R\(Q'*b);
fprintf(' x(1)=%f\n x(2)=%f\n x(3)=%f\n',x(1),x(2),x(3));
```

QR_Houref.m

```
function [Q,R]=QR_Houref(A)
R=A;
[m,n]=size(R);
Q=eye(m);
for i=1:n
    x=R(i:m,i);
    w=[-sign(x(1))*norm(x);zeros(m-i,1)];
    v=x-w;
    h=eye(m-i+1)-2*(v*v')/(v'*v);
    H=eye(m);
    H(i:m,i:m)=h;
    Q=Q*H;
    R=H*R;
end
end
```

函数检验：

```
Q=
   -0.4804    0.2697   -0.4057   -0.6430   -0.3442
   -0.6405   -0.5494    0.2236   -0.1427    0.4664
    0.4804   -0.6592    0.0310   -0.5023   -0.2853
   -0.1601   -0.4295   -0.6914    0.5042   -0.2400
    0.3203    0.0799   -0.5535   -0.2443    0.7246


q=
   -0.4804    0.2697   -0.4057   -0.6430   -0.3442
   -0.6405   -0.5494    0.2236   -0.1427    0.4664
    0.4804   -0.6592    0.0310   -0.5023   -0.2853
   -0.1601   -0.4295   -0.6914    0.5042   -0.2400
    0.3203    0.0799   -0.5535   -0.2443    0.7246
```

```
R=
   -6.2450    0.6405   -0.3203
   -0.0000   -2.5670   -2.0277
    0.0000    0.0000   -5.8980
    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000


r=
   -6.2450    0.6405   -0.3203
         0   -2.5670   -2.0277
         0         0   -5.8980
         0         0         0
         0         0         0
```

经检验，编写的 QR 分解的程序是有效正确的。题（2）的求解结果如下。

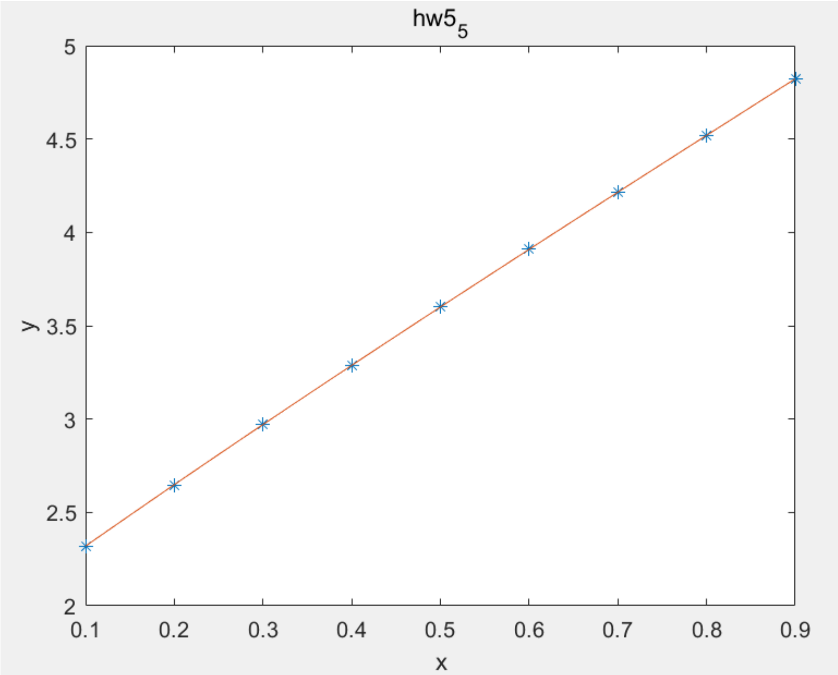
```
x(1)=2.524609
x(2)=0.661633
x(3)=2.093400
```

第五题

Hw5_5.m

```
x=(0.1:0.1:0.9)';
y=[2.3201;2.6470;2.9707;3.2885;3.6008;3.9090;4.2147;4.5191;4.8232];
f=@(c)c(1)*x+c(2)*x.^2.*exp(-c(3)*x)+c(4);
r=@(c)f(c)-y;
c0=[1 1 1 1];
plot(x,y,'*');hold on;
%use Gauss-Newton Method
Dr=@(c)[x,x.^2.*exp(-c(3)*x),-c(2)*x.^3.*exp(-c(3)*x),ones(size(x))];
c1=c0-(Dr(c0)'*Dr(c0))\(Dr(c0)'*r(c0));
TOL=1e-6;
while norm(c1-c0)>=TOL
    c0=c1;
    c1=c0-(Dr(c0)'*Dr(c0))\(Dr(c0)'*r(c0));
end
m=length(r(c1));
RESM=sqrt((r(c1)'*r(c1))/m);
u=linspace(0.1,0.9);
v=c1(1)*u+c1(2)*u.^2.*exp(-c1(3)*u)+c1(4);
plot(u,v,'-');
xlabel 'x';ylabel 'y';title 'hw5_5';
%use lsqnonlin
c2= lsqnonlin(r,c0);
fprintf(' RESM=%f\n',RESM);
fprintf('Gauss-Newton Method:c1=%f c2=%fc3=%fc4=%f\n',c1(1),c1(2),c1(3),c1(4));
fprintf('lsqnonlin:c1=%f c2=%f c3=%f c4=%f\n',c1(1),c1(2),c1(3),c1(4));
```

比较：高斯牛顿法与 lsqnonlinear 得到的结果一致



第五题有同学初值取得不好，结果误差较大；有同学 TOL 取得太弱，也会造成结果误差较大

第六题

hw5_6.m

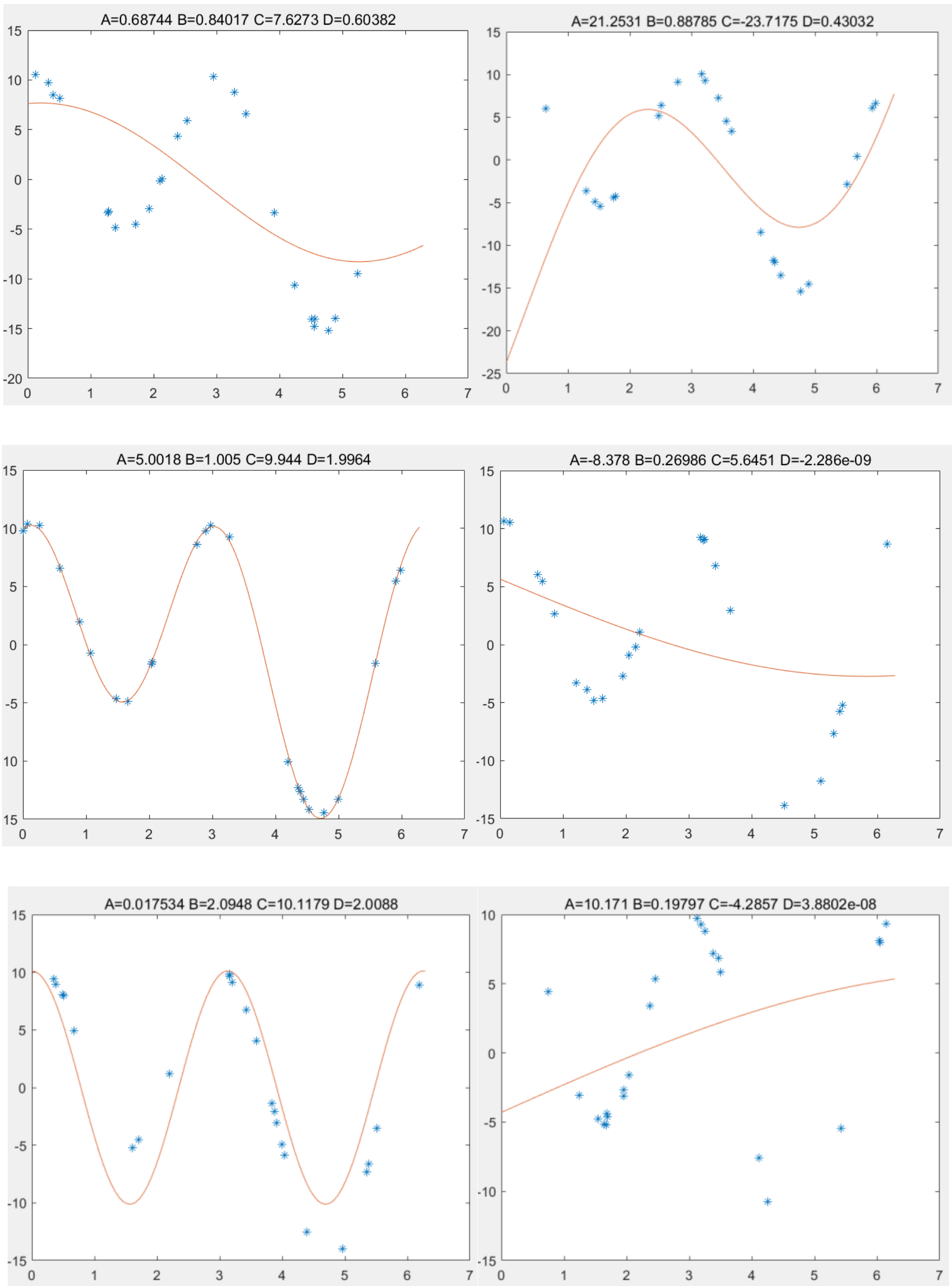
```
clc,clear;
x=rand(24,1)*2*pi;
y=5*sin(x)+10*cos(2*x)+rand(24,1)-0.5;
init=[0.5;0.5;0.5;0.5];
[A,B,C,D]=myLM(x,y,init);
fprintf('A=%f B=%f C=%f D=%f\n',A,B,C,D);
figure;
u=linspace(0,2*pi);
v=A*sin(B*u)+C*cos(D*u);
plot(x,y,'*',u,v,'-');

function [A,B,C,D]=myLM(x,y,init)
a0=init;TOL=1e-8; u=50;v=2;
r=@(a)a(1)*sin(a(2)*x)+a(3)*cos(a(4)*x)-y;
Dr=@(a)[sin(a(2)*x),a(1)*x.*cos(a(2)*x),cos(a(4)*x),-a(3)*x.*sin(a(4)*x)];
M=Dr(a0)'*Dr(a0);
a1=a0-(M+u*diag(diag(M)))\(Dr(a0)'*r(a0));
while norm(a1-a0)>=TOL
    if norm(r(a1))<norm(r(a0))
        u=u/v;
        a0=a1;
```

```
        else
                u=u*v;
        end
        a1=a0-(M+u*diag(diag(M)))\(Dr(a0)'*r(a0));
end
A=a1(1);B=a1(2);C=a1(3);D=a1(4);
end
```

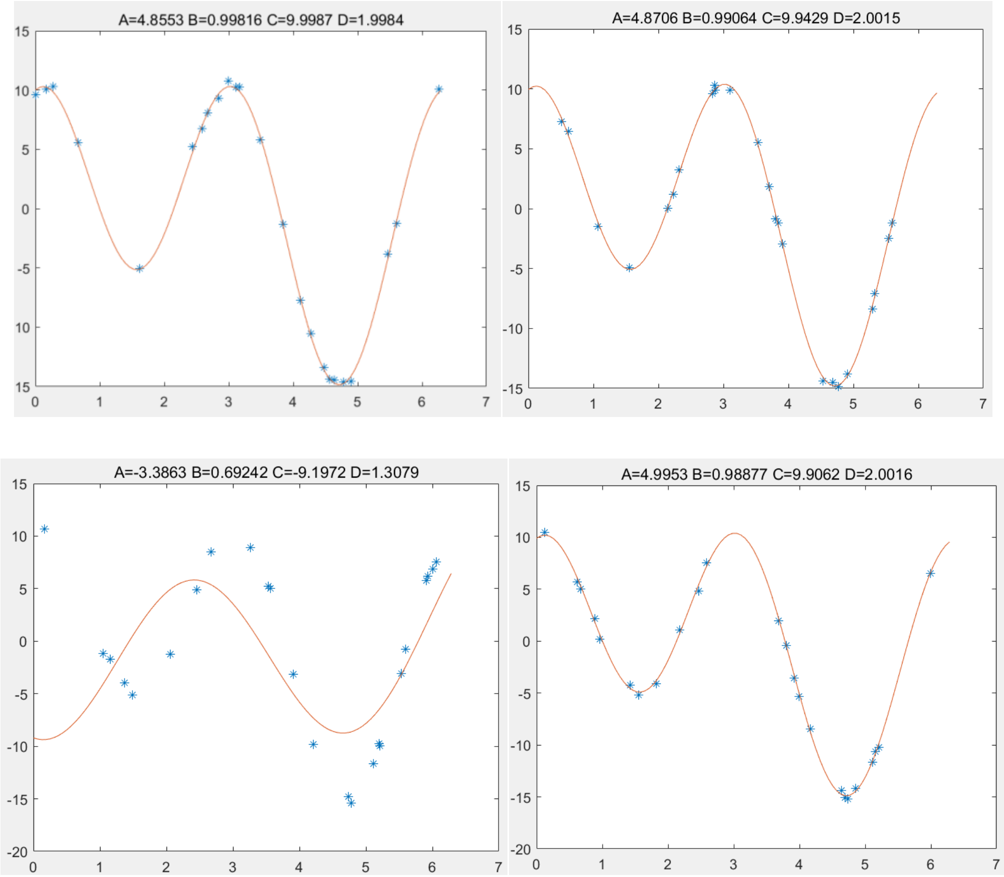
不同初值下的结果主要有以下几种情况：

（1）迭代初值中有 0 或极为接近 0 的数存在，会大概率使得计算过程中出现奇异矩阵，无法完成所要求的计算。如[1;1;1;0]时，算法失效。

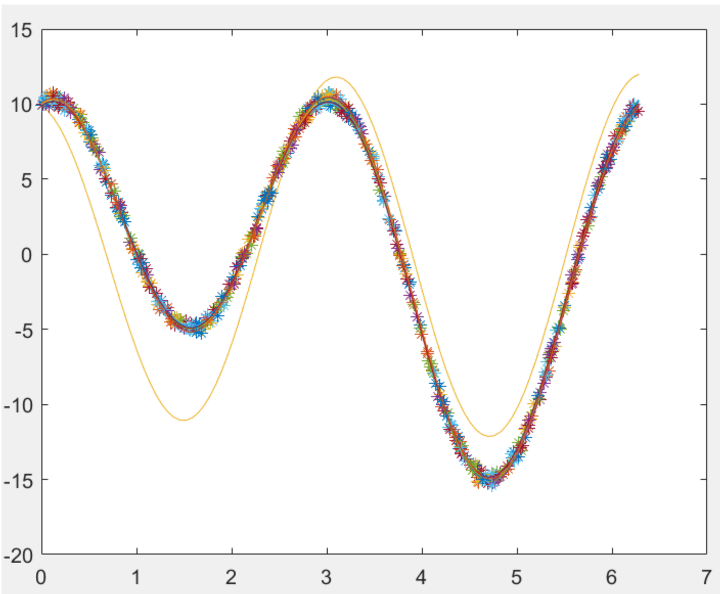
（2）当迭代初值离给定参数较远时，如[0.5;0.5;0.5;0.5]时，拟合的结果很不稳定，只有极少数情况可以拟合出接近原参数的结果，总体来说效果很差，见下图。



（3）当迭代初值离给定参数较近时，如[4.5;0.5;9.5;1.5]时，拟合的结果仍然不稳定，但拟合成功的概率提升了很多。总体来说效果较好，见下图。



（4）当频率项也即 B 和 D 与原始参数十分接近时，幅值项 A 和 C 的影响就比较小了。比如取初值为[0.5;0.9;0.5;1.9]时，拟合的结果已经较为稳定了，下面为连续 20 次的拟合结果，仅有一次拟合失败。



```
A=4.923348 B=0.996835 C=9.979624 D=1.999558
A=4.868220 B=1.000100 C=10.013383 D=1.997563
A=5.129177 B=0.997545 C=9.905032 D=2.001504
A=5.023591 B=1.007128 C=10.058595 D=1.996066
A=2.272471 B=-1.689465 C=9.882970 D=1.996275
A=5.016211 B=1.001322 C=9.980440 D=2.000342
A=4.852304 B=1.001268 C=10.012884 D=2.001864
A=4.991754 B=0.987963 C=10.035841 D=2.002313
A=4.859493 B=0.996758 C=9.936977 D=2.001698
A=4.784536 B=0.989673 C=10.106909 D=2.001520
A=5.098148 B=0.995046 C=9.944514 D=2.002194
A=5.038110 B=0.997862 C=10.022544 D=1.999773
A=4.921961 B=1.012368 C=9.817532 D=1.999572
A=5.045807 B=0.979606 C=10.054862 D=2.002775
A=5.096632 B=1.008553 C=9.950816 D=1.996578
A=5.092274 B=0.995146 C=10.064272 D=1.998373
A=5.105915 B=1.011007 C=9.916827 D=1.997359
A=5.101544 B=1.006630 C=9.978899 D=1.997601
A=5.006834 B=1.000520 C=9.997924 D=1.998361
A=5.084730 B=0.999689 C=10.061593 D=2.000938
A=4.967347 B=1.004996 C=9.883732 D=2.001130
A=4.900367 B=0.990531 C=10.076194 D=1.999377
```

综上，对于拟合该函数的迭代初值的选择，若离原始参数较远，拟合效果极不稳定；若离原始参数较近，效果较稳定。拟合结果稳定与否其实主要取决于频率项与原始参数差距是否过大。

附加题

```
function [Q,R]=QR_Schimidt(A)
[m,n]=size(A);
A(:,n+1:m)=eye(m,m-n);
for j=1:m
    y=A(:,j);
    for i=1:j-1
        R(i,j)=Q(:,i)'*A(:,j);
        y=y-R(i,j)*Q(:,i);
    end
    R(j,j)=norm(y);
    Q(:,j)=y/R(j,j);
end
R(:,n+1:m)=[];
end
```

```
hw5_7.m
clear,clc;
A=[4 8 1;0 2 -2;3 6 7];
[Q,R]=QR_Schimidt(A);
[q,r]=qr(A);
```

```
disp('Q=');disp(Q);
disp('q=');disp(q);
disp('R=');disp(R);
disp('r=');disp(r);
```

```
Q=
    0.8000         0   -0.6000
         0    1.0000         0
    0.6000         0    0.8000


q=
   -0.8000    0.0000   -0.6000
         0   -1.0000   -0.0000
   -0.6000   -0.0000    0.8000
R=
     5    10     5
     0     2    -2
     0     0     5


r=
   -5.0000  -10.0000   -5.0000
         0   -2.0000    2.0000
         0         0    5.0000
```

可以看出 QR_Schimidt 进行的消减 QR 分解结果与内置函数 qr 结果一致，符号不同，是因为 qr 在进行 Householder 变换时为了稳定性取了钝角，添加了负号。