Problem 1 (graded by Kangchen) 30 points

(a)- 5 points

if $[\boldsymbol{U}, \boldsymbol{S}, \boldsymbol{V}] = svd(\boldsymbol{G})$, so $\boldsymbol{G} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$, thus $\boldsymbol{G}_g^{-1} = \boldsymbol{V}\boldsymbol{S}^{-1}\boldsymbol{U}^T$. Note $S_{i,i}^{-1} = 1/S_{i,i}$ for $S_{i,i} \neq 0$, and $S_{i,i}^{-1} = 0$ for $S_{i,i} = 0$. The result is

```
R1 =
    0.2839    0.4645    0.1484    0.1032
   -0.0194   -0.1226    0.0581    0.0839
R2 =
    0.2769    0.2154    0.3231    0.1846
   -0.3590    0.4615   -0.9744    0.8718
R3 =
    2.2194   12.7226   -5.6581   -8.2839
   -0.0194   -0.1226    0.0581    0.0839
R4 =
    0.0250    0.0250    0.0250    0.0250
    0.0750    0.0750    0.0750    0.0750
```

(b)- 5 points

$\boldsymbol{G}_{LLS}^{-1} = inv(\boldsymbol{G}^T\boldsymbol{G})\boldsymbol{G}^T$. But if $\det \boldsymbol{G}^T\boldsymbol{G} = 0$, then the least square problem will have infinitely many solutions that gives the same minimized error, which means $\boldsymbol{G}_{LLS}^{-1}$ is not well defined in this case.

```
L1 =
    0.2839    0.4645    0.1484    0.1032
   -0.0194   -0.1226    0.0581    0.0839
L2 =
    0.2769    0.2154    0.3231    0.1846
   -0.3590    0.4615   -0.9744    0.8718
L3 =
    2.2194   12.7226   -5.6581   -8.2839
   -0.0194   -0.1226    0.0581    0.0839
L4 =
   Inf   Inf   Inf   Inf
   Inf   Inf   Inf   Inf
```

```
function hw7p1()
G1=[1 1 1 1; 1 -3 4 5]';
G2=[1 1 1 1; -0.1 0.3 -0.4 0.5]';
G3=[1 1 1 1; 101 97 104 105]';
G4=[1 1 1 1; 3 3 3 3]';
```

```

R1=pseduoInverse(G1)
R2=pseduoInverse(G2)
R3=pseduoInverse(G3)
R4=pseduoInverse(G4)

% least square inverse
L1=LLSInverse(G1)
L2=LLSInverse(G2)
L3=LLSInverse(G3)
L4=LLSInverse(G4)

function r=LLSInverse(G)
r = inv(G'*G)*G';

function r=pseduoInverse(G)
[U,S,V]=svd(G); % check if U*S*V' == G
eps=1E-16;
for i=1:min(size(S))
    if(S(i,i) < eps)
        S(i,i) = 0;
    else
        S(i,i) = 1.0/S(i,i);
    end
end
r=V*S'*U';
```

(c)- 5 points

When least square inverse is unique, $\det \boldsymbol{G}^T\mathbf{G} \neq 0$, $\boldsymbol{G}_{LLS}^{-1}$ is the same as general inverse, because

$$
\begin{aligned}
\boldsymbol{G} &= \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T \\
\boldsymbol{G}^T &= \boldsymbol{V}\boldsymbol{S}^T\boldsymbol{U}^T \\
\boldsymbol{G}^T\boldsymbol{G} &= \boldsymbol{V}(\boldsymbol{S}^T\boldsymbol{S})\boldsymbol{V}^T \\
(\boldsymbol{G}^T\boldsymbol{G})^{-1} &= \boldsymbol{V}(\boldsymbol{S}^T\boldsymbol{S})^{-1}\boldsymbol{V}^T \\
(\boldsymbol{G}^T\boldsymbol{G})^{-1}\boldsymbol{G}^T &= \boldsymbol{V}(\boldsymbol{S}^T\boldsymbol{S})^{-1}\boldsymbol{S}^T\boldsymbol{U}^T \\
&= \boldsymbol{V}\boldsymbol{S}^{-1}\boldsymbol{U}^T \\
&= \boldsymbol{G}_g^{-1}
\end{aligned}
$$

which is the same as generalized inverse. Note in the above derivation, U,V is square orthgonal matrix, where $\boldsymbol{S}$, $\boldsymbol{S}^{-1}$ is diagnoal but not square matrix, and $\boldsymbol{S}^T\boldsymbol{S}$ has inverse.

When $\det \boldsymbol{G}^T\boldsymbol{G} = 0$, $\boldsymbol{G}_{LLS}^{-1}$ is not well defined, and general inverse gives the solution that satisfies least square problem and at the same time minimizes the L2 norm of the model.

(d)- 5 points

Note that $\boldsymbol{G}_g^{-1}\boldsymbol{d}$ gives the estimation of intercept and slope of the 4 points least square problem, whose $\boldsymbol{X}$ is given by second column of $\mathbf{G}$. This explains why $\mathbf{G}_{1g}^{-1}$ has the same second row as $\mathbf{G}_{3g}^{-1}$, because $\mathbf{X}$ in $\mathbf{G}_3$ is just a shift of $\mathbf{X}$ in $\mathbf{G}_1$.

For $\mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_3$, the two columns are linearly independent, so they have full column rank, then

$$\begin{aligned}
\mathbf{G}_g^{-1}\mathbf{G} &= \mathbf{V}\mathbf{S}^{-1}\mathbf{U}^T\mathbf{U}\mathbf{S}\mathbf{V}^T \\
&= \mathbf{V}\mathbf{S}^{-1}\mathbf{S}\mathbf{V}^T \\
&= \mathbf{V}\mathbf{V}^T \\
&= \mathbf{I}
\end{aligned}$$

So the 1st row of $\mathbf{G}_g^{-1}$ dot product with 1st column of $\mathbf{G}$ will be 1, the 2nd row of $\mathbf{G}_g^{-1}$ dot product with 2nd column of $\mathbf{G}$ will be 1, and the 1st row of $\mathbf{G}_g^{-1}$ dot product with 2nd column of $\mathbf{G}$ will be 0, the 2nd row of $\mathbf{G}_g^{-1}$ dot product with 1st column of $\mathbf{G}$ will be 0. This explains roughly, for example, the relative higher amplitude of the first row of $\mathbf{G}_3^{-1}$ compared with the 2nd row of $\mathbf{G}_3^{-1}$. However, we must be careful about such amplitude comparison, because for example the 2nd row of $\mathbf{G}_1^{-1}$ and the 2nd row of $\mathbf{G}_3^{-1}$ is the same, although amplitude in $\mathbf{G}_1$ and $\mathbf{G}_3$ is quite different.

For $\mathbf{G}_4$, whose rank is 1, note that the least square solution will require $m_1 + 3m_2 = const$. This is a line in m_1 m_2 plane. The L2 norm of the model, will corresponding to a line segment from origin to this line, which got minimized when it's perpendicular to this line, or $m_1/m_2 = 1/3$. This explains why $\mathbf{G}_4^{-1}$, also a rank 1 matrix, has two rows whose amplitude ratio is 1/3.

Problem 2 (graded by Yiran) 38 points

(a)- 5 points

Looking at the SVDs of the G matrices from problem 1, we see a couple inverses that would potentially benefit from a truncated SVD. Truncation is a useful strategy when there is a large disparity between sizes of singular values. We don't care about the absolute value of the singular values (unless they are close to machine precision (10^{-16})), we only care about the relative size of the singular values to one another. The first 3 G matrices each have two singular values, so we can take the ratio of these two values to see which matrices might require truncation. G4 only has 1 nonzero singular value, so it is efficitively already truncated. If we look at the ratios of singular values $\frac{s_2}{s_1}$ we get:

$$\mathbf{G}1 : \frac{s_2}{s_1} = 0.2393 \tag{1}$$

$$\mathbf{G}2 : \frac{s_2}{s_1} = 0.3469 \tag{2}$$

$$\mathbf{G}3 : \frac{s_2}{s_1} = 3.000 * 10^{-4} \tag{3}$$

G3 has the smallest ratio by several orders of magnitude, so it is out best candidate for truncation. G1 has a smaller ratio than G2, so it would be our second most amenable matrix for truncation.

Truncation has the advantage of removing errors potentially caused by small singular values; thus, not allowing these errors to propagate through the rest of our calculations. However, the disadvantage here is that we are removing information and often the smallest singular values also contain the most information. We want to use truncation in order to keep the most amount of information that isn't dominated by errors or noise.

(b)- 5 points

We select G3 as the most amenable to a truncated SVD. The recalculation yields:

$$\mathbf{G}3_{GeneralizedInverseTruncated} = \begin{pmatrix} 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0024 & 0.0023 & 0.0025 & 0.0025 \end{pmatrix} \tag{4}$$

(c)- 5 points

The second most amenable matrix is G1. The recalculation

$$\mathbf{G}3_{GeneralizedInverseTruncated} = \begin{pmatrix} 0.0031 & -0.0078 & 0.0114 & 0.0141 \\ 0.0216 & -0.0537 & 0.0780 & 0.0969 \end{pmatrix} \tag{5}$$

```
%calculate generalized inverses and svds
function hw7p2()
%problem 2
%To truncate these SVDs we will remove the smallest singular value (the one
%most likely responsible for propagation of large errors).

clc
close all

G1=[1 1 1 1; 1 -3 4 5]';
G2=[1 1 1 1; -0.1 0.3 -0.4 0.5]';
G3=[1 1 1 1; 101 97 104 105]';
G4=[1 1 1 1; 3 3 3 3]';

[U,S,V] = svd(G1);
[U,S,V] = svd(G2);
[U,S,V] = svd(G3);

R1=pseduoInverse(G1);
R2=pseduoInverse(G2);
R3=pseduoInverse(G3);
R4=pseduoInverse(G4);

T3 = GenInvTruncated(G3)
T1 = GenInvTruncated(G1)

%%%%%%%
%Part D
d1 = [10 11 9 12]';
d2 = [10.1 11.4 8.7 9.8]';

x1 = [1 -3 4 5]';
x2 = [-0.1 0.3 -0.4 0.5]';
x3 = [101 97 104 105]';
x4 = [3 3 3 3]';
```

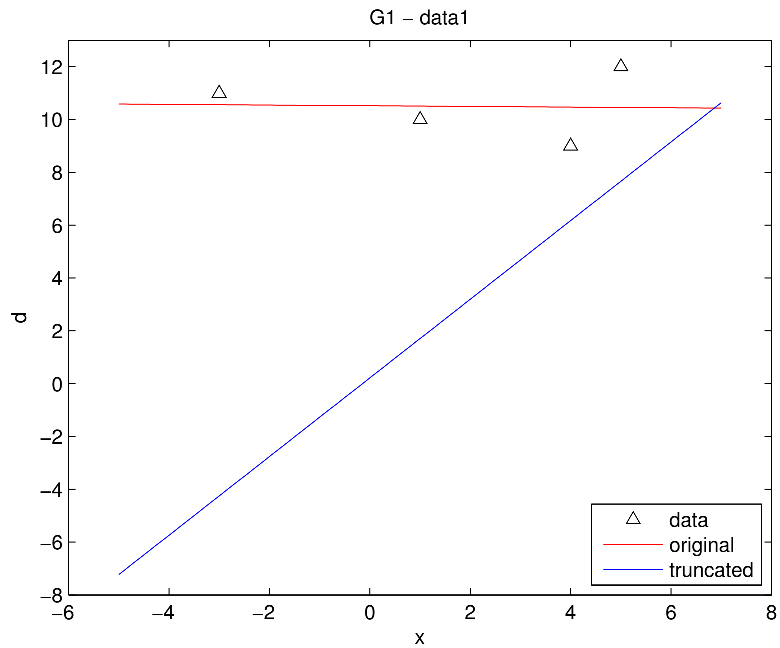
```
G1x = linspace(-5,7);
G3x = linspace(95,107);

%calculate generalized inverse model solution
%using ORIGINAL SVD
m1d1 = R1*d1
m1d2 = R1*d2
m3d1 = R3*d1
m3d2 = R3*d2

%using TRUNCATED SVD
mt1d1 = T1*d1
mt1d2 = T1*d2
mt3d1 = T3*d1
mt3d2 = T3*d2

figure(1)                    %G1 and d1
plot(x1,d1,'k^')
plotOriginal(m1d1,G1x)
plotTruncated(mt1d1,G1x)
title('G1 - data1')
xlabel('x')
ylabel('d')
legend('data','original','truncated','Location','SouthEast')

figure(2)                    %G1 and d2
plot(x1,d2,'k^')
plotOriginal(m1d2,G1x)
plotTruncated(mt1d2,G1x)
title('G1 - data2')
xlabel('x')
ylabel('d')
legend('data','original','truncated','Location','SouthEast')

figure(3)                    %G3 and d1
plot(x3,d1,'k^')
plotOriginal(m3d1,G3x)
plotTruncated(mt3d1,G3x)
title('G3 - data1')
xlabel('x')
ylabel('d')
legend('data','original','truncated')

figure(4)                    %G3 and d2
plot(x3,d2,'k^')
plotOriginal(m3d2,G3x)
plotTruncated(mt3d2,G3x)
title('G3 - data2')
xlabel('x')
ylabel('d')
legend('data','original','truncated')

function plotOriginal(m,Gx)
hold on
y = m(1) + m(2)*Gx;
plot (Gx,y,'r')


function plotTruncated(m,Gx)
hold on
y = m(1) + m(2)*Gx;
plot (Gx,y,'b')


%GENERALIZED INVERSE FUNCTION
function r=pseduoInverse(G)
[U,S,V]=svd(G); % check if U*S*V' == G
eps=1E-16;
for i=1:min(size(S))
    if(S(i,i) < eps)
        S(i,i) = 0;
    else
        S(i,i) = 1.0/S(i,i);
    end
end
r=V*S'*U';
```

```
%TRUNCATE FUNCTION
function r=GenInvTruncated(G)
[U,S,V]=svd(G);      %find SVD
eps=1E-16;
for i=1:min(size(S))
    if(S(i,i) < eps)
        S(i,i) = 0;
    else
        S(i,i) = 1.0/S(i,i);
    end
end
%now truncate the smallest singular value (in this case it's S3(2,2))
Struncated = S;
Struncated(2,2) = 0;     %in this case each of the G matrices only have 2 singular
    values
%recalculate the generalized inverse
r = V * Struncated' * U';
```

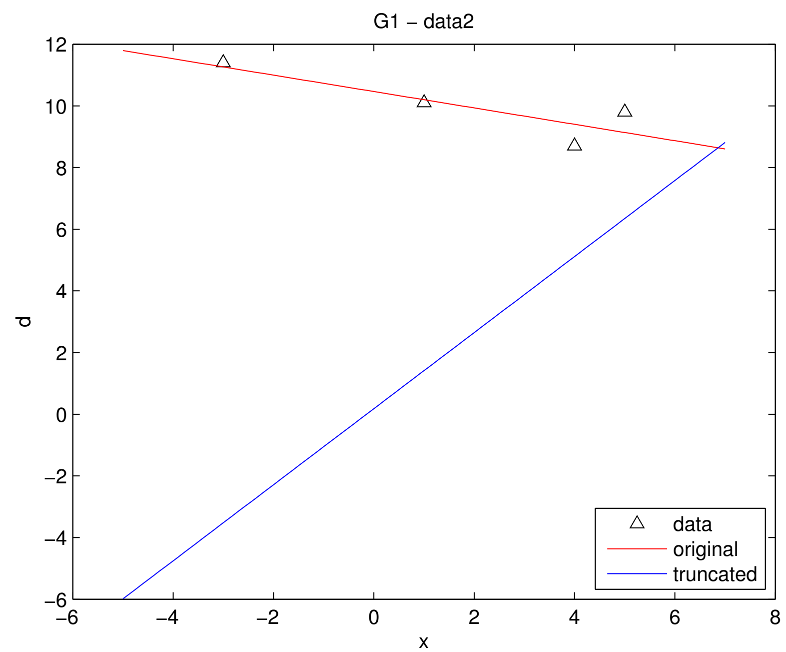
(d)- 5 points

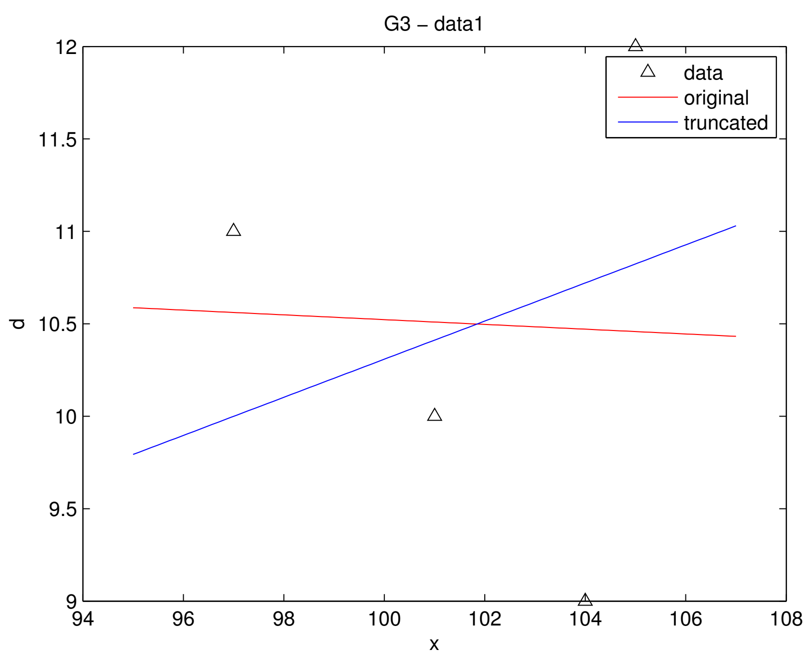
We calculate the generalized inverse model solutions using both the original SVD and the truncated versions for both G1 and G3 for each of the datasets d_1 and d_2. The plots are attached below. Here are the solutions:

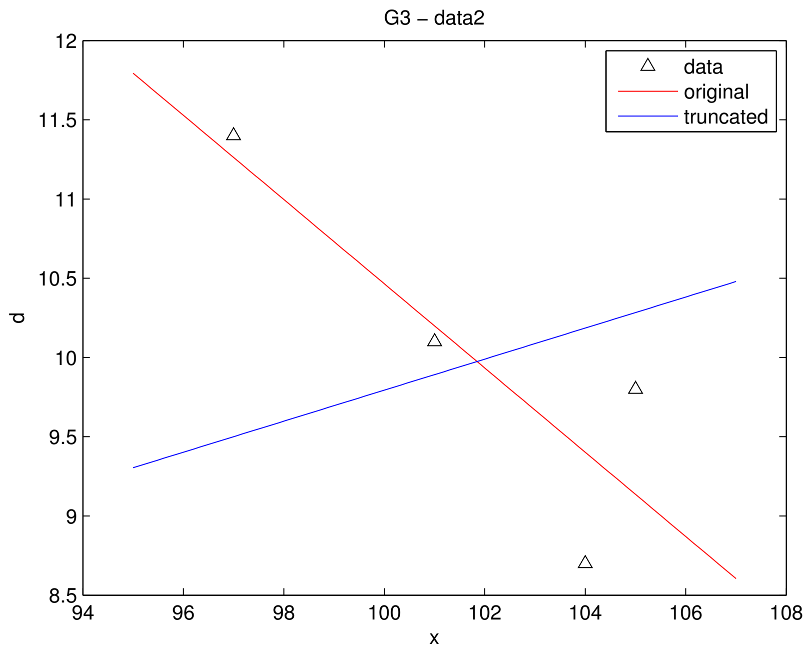
```
m1d1 =

   10.5226
   -0.0129


m1d2 =

   10.4652
   -0.2658


m3d1 =

   11.8129
   -0.0129


m3d2 =

   37.0458
   -0.2658


mt1d1 =
```




```
    0.2171
    1.4893


mt1d2 =

    0.1798
    1.2335


mt3d1 =

    0.0010
    0.1031


mt3d2 =

    0.0010
    0.0979
```


G1 – data2
d
x
data
original
truncated


G3 – data1
d
x
data
original
truncated



(e)- 5 points

Looking at the plots we can see a large difference between the original and the truncated cases. We have truncated the smallest singular value, which influences both rows of the generalized inverse. However, it more significantly influences the first row, which controls our model parameter m_1 (the y-intercept). If we look at our original models, most seem to fit the data reasonably (as reasonably as a line can fit) and all have a negative slope. Our truncated models all have positive slopes and many don't seem to explain the data well at all. This is because our truncation eliminated any information about y-intercept, and thus all the models have y-intercepts of 0 no matter what the data. We can see this behavior clearly on the y-intercept on Toby's plots in the solution for problem 3. As far as which models we would prefer, it would depend on what we know about the dataset. In all cases besides G3-data 1 we would likely prefer the original models because they seem to fit the data better, unless we had some prior information about the y-intercept. For the G3-data 1 neither model seems to fit very well, so it may be better to prefer the truncated model as it is more likely to avoid errors.

Problem 3 (graded by Toby) 32 points

(a)- 5 points

The MATLAB code that generates the matrices is attached at the end of this section. The matrices are calculated using $m = (G^T G + \alpha^2 I)^{-1} G^T d$. The matrices are

$$\mathbf{G}^{-1}_{1,g,T,\alpha_1} = \begin{pmatrix} 0.282931351378109 & 0.462937664456661 & 0.147926616569195 & 0.102925038299558 \\ -0.019222102718031 & -0.122340004924458 & 0.058116323936789 & 0.083895799488396 \end{pmatrix}$$

$$\mathbf{G}^{-1}_{2,g,T,\alpha_1} = \begin{pmatrix} 0.275661587810746 & 0.215517241379310 & 0.320769847634322 & 0.185445068163593 \\ -0.351343223736969 & 0.452586206896552 & -0.954290296712109 & 0.854550922213312 \end{pmatrix}$$

$$\mathbf{G}^{-1}_{3,g,T,\alpha_1} = \begin{pmatrix} 0.604003054340609 & 3.462399532543330 & -1.539794304311428 & -2.254393423862098 \\ -0.003493985867091 & -0.031656565418916 & 0.017627948796778 & 0.024668593684735 \end{pmatrix}$$

$$\mathbf{G}^{-1}_{4,g,T,\alpha_1} = \begin{pmatrix} 0.024993751562093 & 0.024993751562093 & 0.024993751562093 & 0.024993751562093 \\ 0.074981254686328 & 0.074981254686328 & 0.074981254686328 & 0.074981254686328 \end{pmatrix}$$

$$\mathbf{G}^{-1}_{1,g,T,\alpha_2} = \begin{pmatrix} 0.262125138837468 & 0.427989633469086 & 0.137726767863754 & 0.096260644205850 \\ -0.016290262865605 & -0.116993706034802 & 0.059237319511292 & 0.084413180303591 \end{pmatrix}$$

$$\mathbf{G}^{-1}_{2,g,T,\alpha_2} = \begin{pmatrix} 0.251592356687898 & 0.213375796178344 & 0.280254777070064 & 0.194267515923567 \\ -0.230891719745223 & 0.310509554140127 & -0.636942675159236 & 0.581210191082802 \end{pmatrix}$$

$$\mathbf{G}^{-1}_{3,g,T,\alpha_2} = \begin{pmatrix} 0.032727078270480 & 0.187497400490787 & -0.083350663394751 & -0.122043243949828 \\ 0.002115257782188 & 0.000499105768831 & 0.003327371792206 & 0.003731409795545 \end{pmatrix}$$

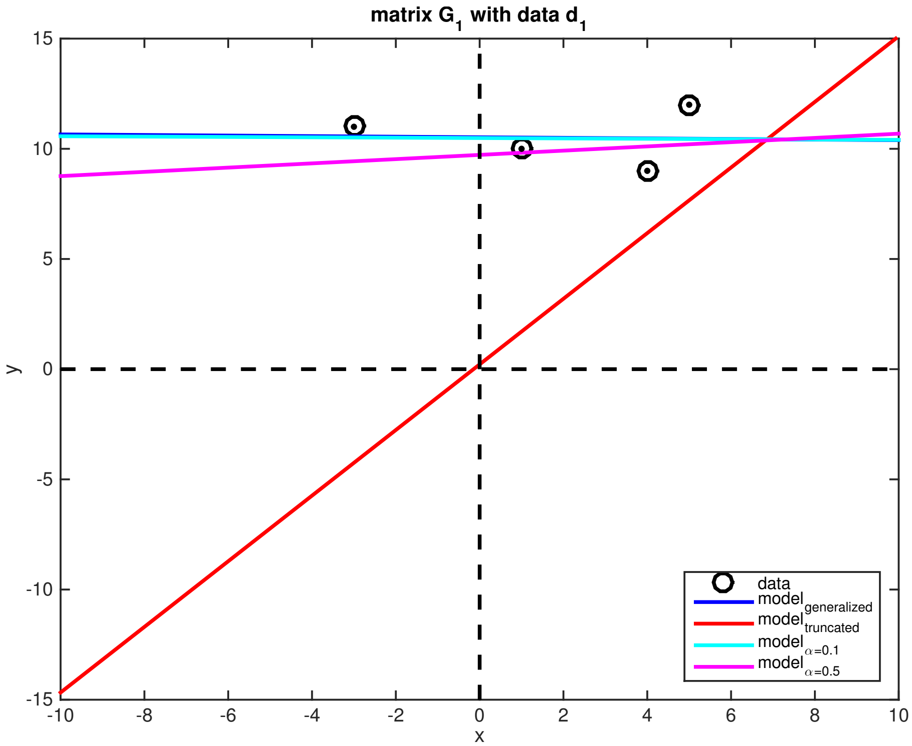
$$\mathbf{G}^{-1}_{4,g,T,\alpha_2} = \begin{pmatrix} 0.024844720496894 & 0.024844720496894 & 0.024844720496894 & 0.024844720496894 \\ 0.074534161490683 & 0.074534161490683 & 0.074534161490683 & 0.074534161490683 \end{pmatrix}$$

(b)- 5 points

For the generalized inverse we have that

$$\mathbf{G}^{-1}_g = \sum_{i=1}^{p} \frac{s_i^2}{s_i^2 + \alpha^2} \frac{1}{s_i} v_i u_i^T, \tag{7}$$

where p is the index of the last nonzero singular value. For the regularized inverse, we have extra factors in the expansion that introduce a "smooth" truncation, i.e., for large s_i the term is 1 and for small s_i it is zero. This means that the similarity between the truncated and regularized matrices is determined by the relative size of the singular values compared to α. For G_1, G_2 the singular values are larger than the Tikhonov parameters, implying that the truncated and regularized inverses should look different. For $\mathbf{G}_3$ we have that $s_1^2 >> \alpha_1^2, \alpha_2^2$, $s_2^2 \sim \alpha_1^2$ and $s_2^2 << \alpha_2^2$. According to the equation above, this implies that the truncated and regularized inverses are different for $\alpha = 0.1$, but similar for $\alpha = 0.5$. For $\mathbf{G}_4$ the truncated and the regularized must be different, because we only have two singular values, one of which is 0 anyway.



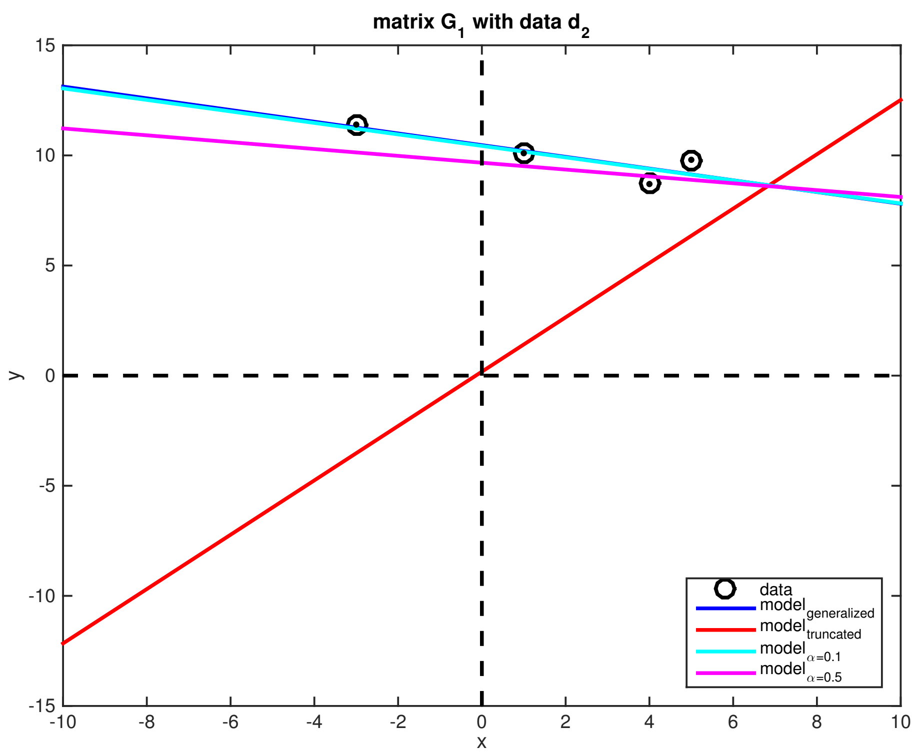
(c)- 5 points

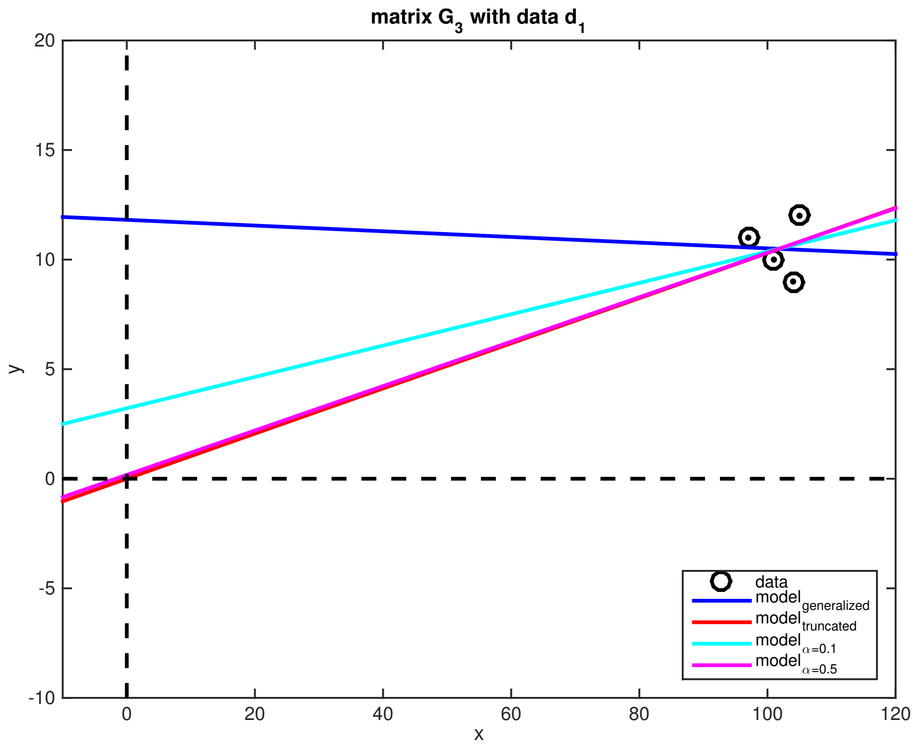
The four plots are
Here is a plot of the profile:

(d)- 5 points

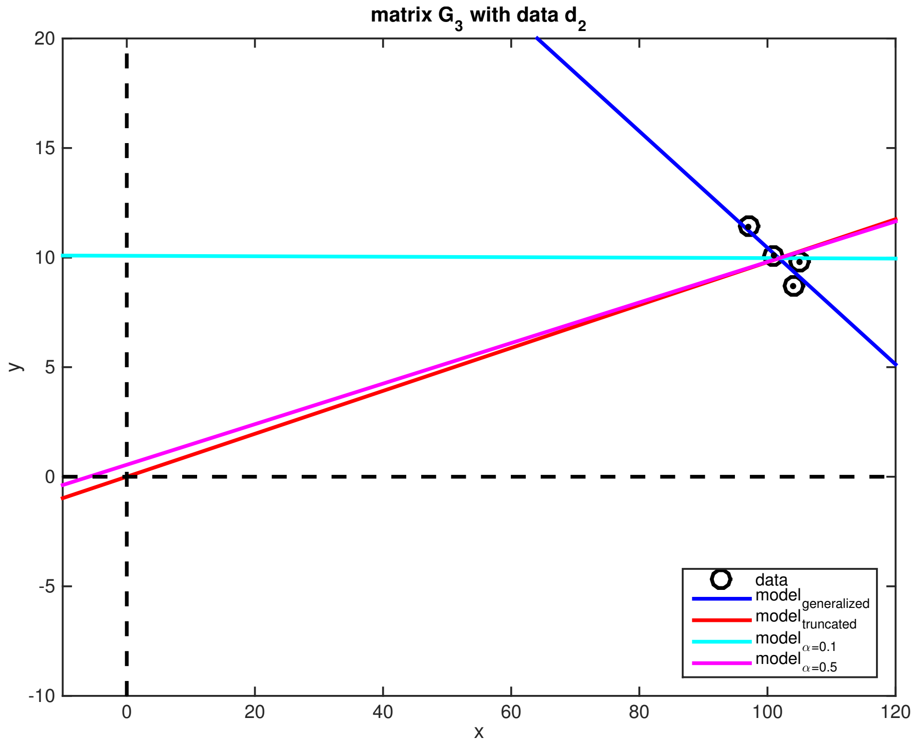
For $\mathbf{G}_3$ the truncated model and the Tikhonov model with $\alpha = 0.5$ are very close. This is because of a term $s_i^2/(s_i^2 + \alpha^2)$ in the expansision for the generalized inverse in terms of its singular values is almost zero for the smallest singular value of $\mathbf{G}_3$ and almost 1 for the largest singular value when $\alpha = 0.5$. Thus the truncated and the regularized models are very close. This is not the case for $\mathbf{G}_1$, because the singular values are not as far apart and always larger than α, but not by several orders of magnitude. Thus the truncated and the regularized models are giving different answers. In fact, for small truncation parameters, the regurlazied model is close to the generalized inverse model.
MATLAB script:

```
clear all; clc;
%% This is the solution script to problem 3 on homework set 3

%% Problem part (a)
% Define the input data and Tikhonov parameters
G1 = [1 1; 1 -3; 1 4; 1 5];
```


matrix G_1 with data d_2
15
10
5
0
-5
-10
-15
y
-10
-8
-6
-4
-2
0
2
4
6
8
10
x
data
model$_{generalized}$
model$_{truncated}$
model$_{\alpha=0.1}$
model$_{\alpha=0.5}$




```
G2 = [1 -0.1; 1 0.3; 1 -0.4; 1 0.5];
G3 = [1 101; 1 97; 1 104; 1 105];
G4 = [1 3; 1 3; 1 3; 1 3];

alpha1 = 0.1;
alpha2 = 0.5;

% Calculate the regularized inverse matrices
reg1_1 = inv(G1'*G1 + alpha1^2*eye(2,2))*G1';
reg1_2 = inv(G2'*G2 + alpha1^2*eye(2,2))*G2';
reg1_3 = inv(G3'*G3 + alpha1^2*eye(2,2))*G3';
reg1_4 = inv(G4'*G4 + alpha1^2*eye(2,2))*G4';
reg2_1 = inv(G1'*G1 + alpha2^2*eye(2,2))*G1';
reg2_2 = inv(G2'*G2 + alpha2^2*eye(2,2))*G2';
reg2_3 = inv(G3'*G3 + alpha2^2*eye(2,2))*G3';
reg2_4 = inv(G4'*G4 + alpha2^2*eye(2,2))*G4';

%% Problem part (c)
%Define the data sets
d1 = [10 11 9 12]';
d2 = [10.1 11.4 8.7 9.8]';

%The 8 previous model parameter vectors are
m_d1_1  = [10.523 -0.013]';
m_d1_1t = [0.217 1.489]';
m_d2_1  = [10.465 -0.266]';
m_d2_1t = [0.180 1.234]';
m_d1_3  = [11.813 -0.013]';
m_d1_3t = [0.001 0.103]';
m_d2_3  = [37.046 -0.266]';
m_d2_3t = [9.616e-4 9.793e-2]';

%Calculate the regularized inverse for the two data sets and G1/G3
m_d1_1_r1 = reg1_1*d1;
m_d1_1_r2 = reg2_1*d1;
m_d2_1_r1 = reg1_1*d2;
```




```
m_d2_1_r2 = reg2_1*d2;
m_d1_3_r1 = reg1_3*d1;
m_d1_3_r2 = reg2_3*d1;
m_d2_3_r1 = reg1_3*d2;
m_d2_3_r2 = reg2_3*d2;

%Now set up the four different plots for the different combinations of
%data sets and matrices

%For G1 using d1
x = linspace(-10,10,100);
y_d1_1_1 = m_d1_1(1) + m_d1_1(2)*x;
y_d1_1_1t = m_d1_1t(1) + m_d1_1t(2)*x;
y_d1_1_r1 = m_d1_1_r1(1) + m_d1_1_r1(2)*x;
y_d1_1_r2 = m_d1_1_r2(1) + m_d1_1_r2(2)*x;
x_d1 = G1(:,2);

plot(x_d1, d1, 'ko', ...
     x, y_d1_1_1, 'b-', ...
     x, y_d1_1_1t, 'r-', ...
     x, y_d1_1_r1, 'c-', ...
     x, y_d1_1_r2, 'm-', ...
     x_d1, d1, 'k.', ...
     [-1 1]*1000, [0 0], 'k--', ...
     [0 0], [-1 1]*1000, 'k--', 'MarkerSize', 10, 'LineWidth', 2);
legend('data', 'model_{generalized}', 'model_{truncated}', ...
       'model_{\alpha=0.1}', 'model_{\alpha=0.5}', 'Location', 'SouthEast');
title('matrix G_1 with data d_1');
xlabel('x');
ylabel('y');
box on;
axis([-10 10 -15 15])

print('-deps2c','-painters', 'p3b1');

%For G1 using d1
```

```
x = linspace(-10,10,100);
y_d2_1_1 = m_d2_1(1) + m_d2_1(2)*x;
y_d2_1_1t = m_d2_1t(1) + m_d2_1t(2)*x;
y_d2_1_r1 = m_d2_1_r1(1) + m_d2_1_r1(2)*x;
y_d2_1_r2 = m_d2_1_r2(1) + m_d2_1_r2(2)*x;
x_d2 = G1(:,2);

plot(x_d2, d2, 'ko', ...
     x, y_d2_1_1, 'b-', ...
     x, y_d2_1_1t, 'r-', ...
     x, y_d2_1_r1, 'c-', ...
     x, y_d2_1_r2, 'm-', ...
     x_d2, d2, 'k.', ...
     [-1 1]*1000, [0 0], 'k--', ...
     [0 0], [-1 1]*1000, 'k--', 'MarkerSize', 10, 'LineWidth', 2);
legend('data', 'model_{generalized}', 'model_{truncated}', ...
       'model_{\alpha=0.1}', 'model_{\alpha=0.5}', 'Location', 'SouthEast');
title('matrix G_1 with data d_2');
xlabel('x');
ylabel('y');
box on;
axis([-10 10 -15 15])

print('-deps2c','-painters', 'p3b2');

%For G3 using d1
x = linspace(-10,120,100);
y_d1_3_1 = m_d1_3(1) + m_d1_3(2)*x;
y_d1_3_1t = m_d1_3t(1) + m_d1_3t(2)*x;
y_d1_3_r1 = m_d1_3_r1(1) + m_d1_3_r1(2)*x;
y_d1_3_r2 = m_d1_3_r2(1) + m_d1_3_r2(2)*x;
x_d1 = G3(:,2);

plot(x_d1, d1, 'ko', ...
     x, y_d1_3_1, 'b-', ...
     x, y_d1_3_1t, 'r-', ...
     x, y_d1_3_r1, 'c-', ...
     x, y_d1_3_r2, 'm-', ...
     x_d1, d1, 'k.', ...
     [-1 1]*1000, [0 0], 'k--', ...
     [0 0], [-1 1]*1000, 'k--', 'MarkerSize', 10, 'LineWidth', 2);
legend('data', 'model_{generalized}', 'model_{truncated}', ...
       'model_{\alpha=0.1}', 'model_{\alpha=0.5}', 'Location', 'SouthEast');
title('matrix G_3 with data d_1');
xlabel('x');
ylabel('y');
box on;
axis([-10 120 -10 20])

print('-deps2c','-painters', 'p3b3');

%For G3 using d1
x = linspace(-10,120,100);
y_d2_3_1 = m_d2_3(1) + m_d2_3(2)*x;
y_d2_3_1t = m_d2_3t(1) + m_d2_3t(2)*x;
y_d2_3_r1 = m_d2_3_r1(1) + m_d2_3_r1(2)*x;
y_d2_3_r2 = m_d2_3_r2(1) + m_d2_3_r2(2)*x;
x_d2 = G3(:,2);

plot(x_d2, d2, 'ko', ...
     x, y_d2_3_1, 'b-', ...
     x, y_d2_3_1t, 'r-', ...
     x, y_d2_3_r1, 'c-', ...
     x, y_d2_3_r2, 'm-', ...
     x_d2, d2, 'k.', ...
     [-1 1]*1000, [0 0], 'k--', ...
     [0 0], [-1 1]*1000, 'k--', 'MarkerSize', 10, 'LineWidth', 2);
legend('data', 'model_{generalized}', 'model_{truncated}', ...
       'model_{\alpha=0.1}', 'model_{\alpha=0.5}', 'Location', 'SouthEast');
title('matrix G_3 with data d_2');
xlabel('x');
ylabel('y');
box on;
axis([-10 120 -10 20])

print('-deps2c','-painters', 'p3b4');
```