Problem 1 (graded by Dunzhu) - 40 points

(a) 5 points

$m = [t_s, x_s, y_s, z_s, v]^T$ are model parameters. It's nonlinear.

(b) 10 points

using the notation in class, in this case

$$g_i(m) = t_s + \frac{\sqrt{(x_s - x_i)^2 + (y_s - y_i)^2 + (z_s - z_i)^2}}{v}$$

note x_i, y_i, z_i is some known constant, and $d_i = t_i$.

then the matrix $\hat{G}$, and the residue vector γ

$$\begin{aligned} \hat{G}_{ik} &= \frac{\partial g_i}{\partial m_k} \\ \gamma &= -G^T(d - g(m)) \end{aligned}$$

then the Hessian

$$H = G^T G - \sum_i (d_i - g_i) Q_i$$

The least square solution will be

$$m := m - H^{-1}\gamma$$

define

$$\begin{aligned} R &= \sqrt{(x_s - x_i)^2 + (y_s - y_i)^2 + (z_s - z_i)^2} \\ n_x &= (x_s - x_i)/R \\ n_y &= (y_s - y_i)/R \\ n_z &= (z_s - z_i)/R \\ e &= t_s + R/v - d_i \end{aligned}$$

note these quantity depends on i.

then

$$\begin{aligned} G(i,:) &= \begin{pmatrix} 1 \\ n_x/v \\ n_y/v \\ n_z/v \\ -\frac{R}{v^2} \end{pmatrix}^T \\ \gamma_i &= e \begin{pmatrix} 1 \\ n_x/v \\ n_y/v \\ n_z/v \\ -\frac{R}{v^2} \end{pmatrix} \end{aligned}$$

Hessian

$$H = \sum_i \begin{pmatrix} 1 & n_x/v & n_y/v & n_z/v & -R/v^2 \\ & \frac{1}{v}\left(\frac{1}{v}n_x^2 + \frac{e}{R}(1-n_x^2)\right) & \frac{1}{v}\left(\frac{1}{v}n_x n_y - \frac{e}{R}(n_x n_y)\right) & \frac{1}{v}\left(\frac{1}{v}n_x n_z - \frac{e}{R}(n_x n_z)\right) & \frac{-n_x}{v^2}\left(\frac{R}{v}+e\right) \\ & & \frac{1}{v}\left(\frac{1}{v}n_y^2 + \frac{e}{R}(1-n_y^2)\right) & \frac{1}{v}\left(\frac{1}{v}n_y n_z - \frac{e}{R}(n_y n_z)\right) & \frac{-n_y}{v^2}\left(\frac{R}{v}+e\right) \\ & & & \frac{1}{v}\left(\frac{1}{v}n_z^2 + \frac{e}{R}(1-n_z^2)\right) & \frac{-n_z}{v^2}\left(\frac{R}{v}+e\right) \\ \textit{symmetric} & & & & \frac{R}{v^3}\left(\frac{R}{v}+2e\right) \end{pmatrix}$$

inexact Hessian

$$G^T G = \sum_i \begin{pmatrix} 1 & n_x/v & n_y/v & n_z/v & -R/v^2 \\ & \frac{1}{v}\left(\frac{1}{v}n_x^2+\right) & \frac{1}{v}\left(\frac{1}{v}n_x n_y\right) & \frac{1}{v}\left(\frac{1}{v}n_x n_z\right) & \frac{-n_x}{v^2}\left(\frac{R}{v}\right) \\ & & \frac{1}{v}\left(\frac{1}{v}n_y^2+\right) & \frac{1}{v}\left(\frac{1}{v}n_y n_z\right) & \frac{-n_y}{v^2}\left(\frac{R}{v}\right) \\ & & & \frac{1}{v}\left(\frac{1}{v}n_z^2\right) & \frac{-n_z}{v^2}\left(\frac{R}{v}\right) \\ \textit{symmetric} & & & & \frac{R}{v^3}\left(\frac{R}{v}\right) \end{pmatrix}$$

which is similar to the same as exact Hessian if $e = 0$.

(c) 10 points

```
function hw2p1()
rng(0); % this generate determined random variable
xi=[0 10 15 6 -7 3]';
yi=[0 0 6 13 10 7]';
zi=[0 0 0 0 0 0]';
ti=[322.418 321.031 321.228 323.093 324.415 322.706]';

M0=do_one_ti(xi,yi,zi,ti);
ti_0=predict(xi,yi,zi,M0);


for it=1:1000
    ti_new = ti + 0.01*randn(6,1);
    % inexact Hessian
    M(:,it)=do_one_ti(xi,yi,zi,ti_new);
end
scatter(M(2,:), M(3,:), 10, M(1,:)); colorbar;
hold on;plot(M0(2),M0(3),'rp','markersize',20,'markerfacecolor','r');
print -depsc fig01.eps
plot([0 10 15 6 -7 3],[0 0 6 13 10 7],'kv');
hold off;
print -depsc fig02.eps

function M=do_one_ti(xi,yi,zi,ti)

M=[300 20 -10 10 2]'; % initial guess
for step = 1:1000
    grad=zeros(5,1);
    hess=zeros(5,5);
    for i=1:length(xi)
        [tmp_grad, tmp_hess]=get_grad_hessian(xi(i),yi(i),zi(i),ti(i),M);
        grad = grad + tmp_grad;
```

```
33          hess = hess + tmp_hess;
34      end
35      err(step) = norm(predict(xi,yi,zi,M)-ti);
36      if(step > 1 && err(step) > err(step-1))
37          break;
38      end
39          M = M- inv(hess)*grad;
40  end
41  plot(err);
42
43  function [grad,hess] = get_grad_hessian(xi,yi,zi,ti,M)
44  ts=M(1); xs=M(2); ys=M(3); zs=M(4); v=M(5);
45  R=sqrt((xs-xi)^2 + (ys-yi)^2 + (zs-zi)^2);
46  nx=(xs-xi)/R;
47  ny=(ys-yi)/R;
48  nz=(zs-zi)/R;
49  e=(ts+R/v-ti);
50
51  grad = e*[1 nx/v ny/v nz/v -R/v^2]';
52  sen = [1 nx/v ny/v nz/v -R/v^2]';
53  hess = sen*sen'; % approximate hess
54
55  function r=predict(xi,yi,zi,M)
56
57  ts=M(1); xs=M(2); ys=M(3); zs=M(4); v=M(5);
58  r=xi*0;
59  for i=1:length(xi)
60      R=sqrt((xs-xi(i))^2 + (ys-yi(i))^2 + (zs-zi(i))^2);
61      r(i)=(ts+R/v);
62  end
```

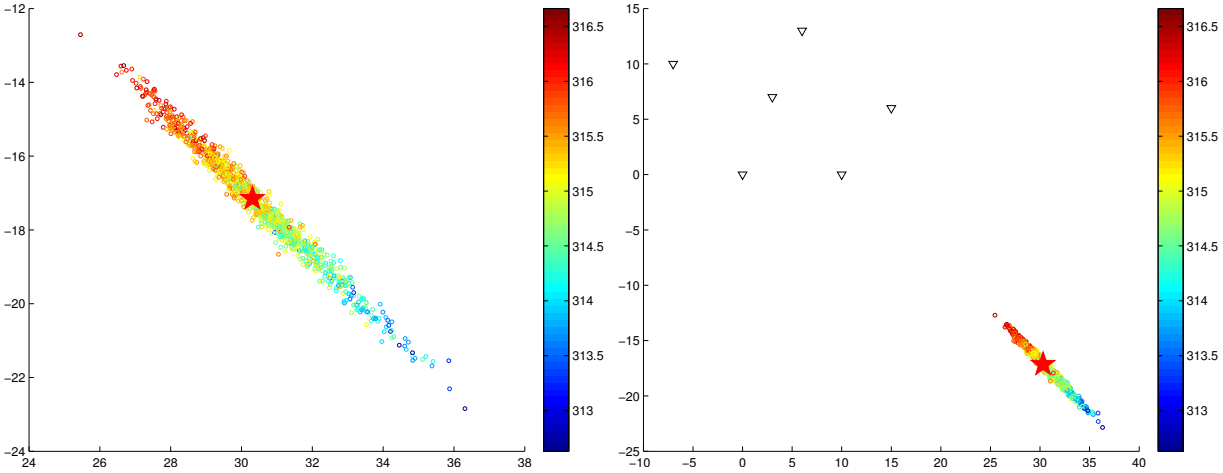

Figure 1: (e): Note the trade off between position and t_s.

(d) 10 points

The best model we get is

$$m = \begin{pmatrix} 315.147372215551 \\ 30.3124982007349 \\ -17.1481612373344 \\ 15.9867180697526 \\ 5.26992730665649 \end{pmatrix}$$

with prediction error at six stations

$$d_i - g_i = \begin{pmatrix} -0.000967665303107879 \\ -0.00257624919419186 \\ 0.00285105963382648 \\ -0.00508794038000815 \\ 0.000968216227079211 \\ 0.0048125790164022 \end{pmatrix}$$

(e) 5 points

The scatter plot of earthquake location forms an elliptical shape, pointing to stations, this indicates the trade off between x_s and y_s, in other words, the earthquake location in this case is well constrained within this azimuth with respect to station position. The color also indicates the trade off between t_s and distance, if event are near, t_s will be larger.

Problem 2 (graded by Toby) - 15 points

An example of a journal article that solves an inverse problem is *Multiscale estimation of GPS velocity fields* by Carl Tape, Pablo Muse, Mark Simons, Danan Dong and Frank Webb. It was published in *Geophysical Journal International* in 2009.

(a) - 5 points

The data are velocity measurements from GPS stations. They are obtained at discrete locations. The authors mention the as reference frame error as a possible source of noise in their GPS data.

(b) - 5 points

The model is the coefficients of a wavelet decomposition. The authors use a finite and discrete number of wavelets.

(c) - 5 points

There are no "physical laws" that determine the operator $\mathbf{G}$ but it consists of the design matrix of the wavelet expansion. The operator is nonlinear in the locations, but linear in the model parameters. The authors discuss issues associtated with uniqueness and stability and they apply a regularization procedure to their least squares inversion.

Problem 3 (graded by Stephen) - 45 points

(a) 2 points

(see Figure 1 below for plot)

(b) 3 points

m1 = [100:0.1:500]; m2 = [-9:0.01:3]; The ranges can vary slightly as long as they have reasonable values. We can get these estimates by looking at the plot and estimating slope and intercepts. However, we need a conservative enough range that we don't have parts of the interesting region get cut off our later plots.

(c) 10 points

```
function [misfit] = lineMisfit(type,x,y,m1,m2)
%type is either 1(L1) or 2(L2) norms
%x and y are the data we are using
%y = m1 + m2*x
%m1 is slope for linear fit and m2 is y-intercept


misfit = zeros(length(m1),length(m2));

if (type==1)        %do L1 misfit
    for(i=1:length(m1))
        for(j=1:length(m2))
            for(k=1:length(x))
                pointMisfit = abs(y(k) - (m1(i)+m2(j)*x(k)));
                misfit(i,j) = misfit(i,j) + pointMisfit;
            end
        end
    end
end

if (type==2)        %do L2 misfit
    for(i=1:length(m1))
        for(j=1:length(m2))
            for(k=1:length(x))
                pointMisfit = (y(k) - (m1(i)+m2(j)*x(k)))*((y(k) - (m1(i)+m2(j)*x(k))));
                misfit(i,j) = misfit(i,j) + pointMisfit;
            end
        end
    end
end
```

(d) 10 points

```
%L1 misfit
misfit = lineMisfit(1,x,y,m1,m2);

minmisfit = min(min(misfit));
maxmisfit = max(max(misfit));
```

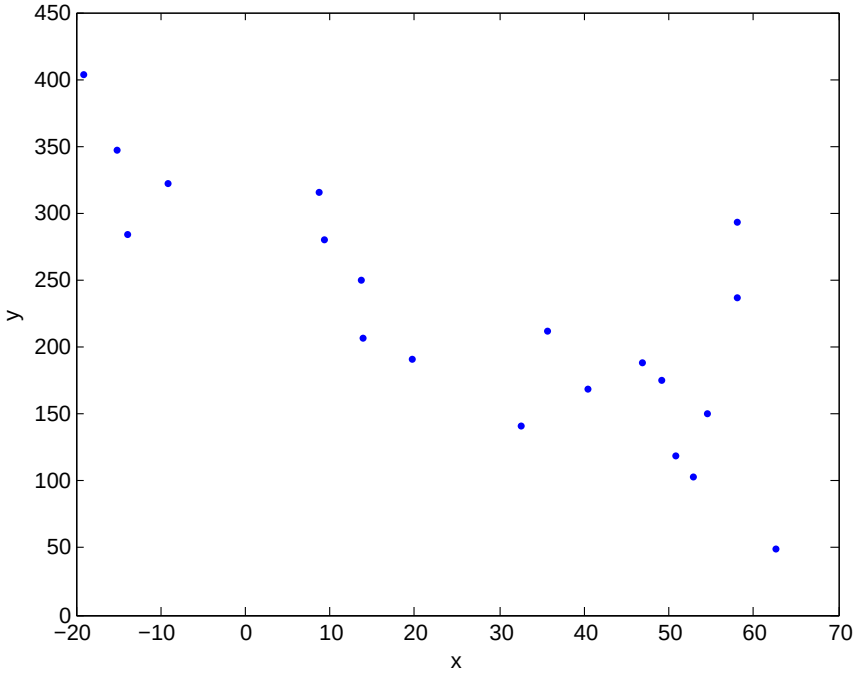

Figure 2: Data plotted for part a

```
figure;
pcolor(m2,m1,misfit)
xlabel('m2')
ylabel('m1')
shading flat
caxis([minmisfit minmisfit+0.15*(maxmisfit-minmisfit)])
hcb = colorbar;
colorTitleHandle = get(hcb,'Title');
titleString = 'L1 Misfit';
set(colorTitleHandle ,'String',titleString);


%L2 misfit
misfit2 = lineMisfit(2,x,y,m1,m2);

minmisfit2 = min(min(misfit2));
maxmisfit2 = max(max(misfit2));

figure;
pcolor(m2,m1,misfit2)
xlabel('m2')
ylabel('m1')
shading flat
caxis([minmisfit2 minmisfit2+0.15*(maxmisfit2-minmisfit2)])
hcb = colorbar;
colorTitleHandle = get(hcb,'Title');
titleString = 'L2 Misfit';
set(colorTitleHandle ,'String',titleString);
```

(see below Figures 2 and 3 for output)

(e) 5 points

```
[r1,c1]=find(misfit==min(min(misfit)));
[r2,c2]=find(misfit2==min(min(misfit2)));

L1bestm1=m1(r1)
L1bestm2=m2(c1)
L2bestm1=m1(r2)
L2bestm2=m2(c2)
```

Answers will vary slightly depending on the resolution of the discretization. For my ranges of m_1 and m_2 above I get the following values: L1: $m_1 = 297.2$, $m_2 = -2.7$ L2: $m_1 = 292$, $m_2 = -2.55$

(f) 5 points

Formula for finding the best fit for method of least squares: $m_{LLS} = (G^T G)^{-1} G^T d$

```
G = [ones(20,1) x];          %we get this from our m1+m2*x
%now plug into our formula
m = (G'*G)^-1*G'*y
```

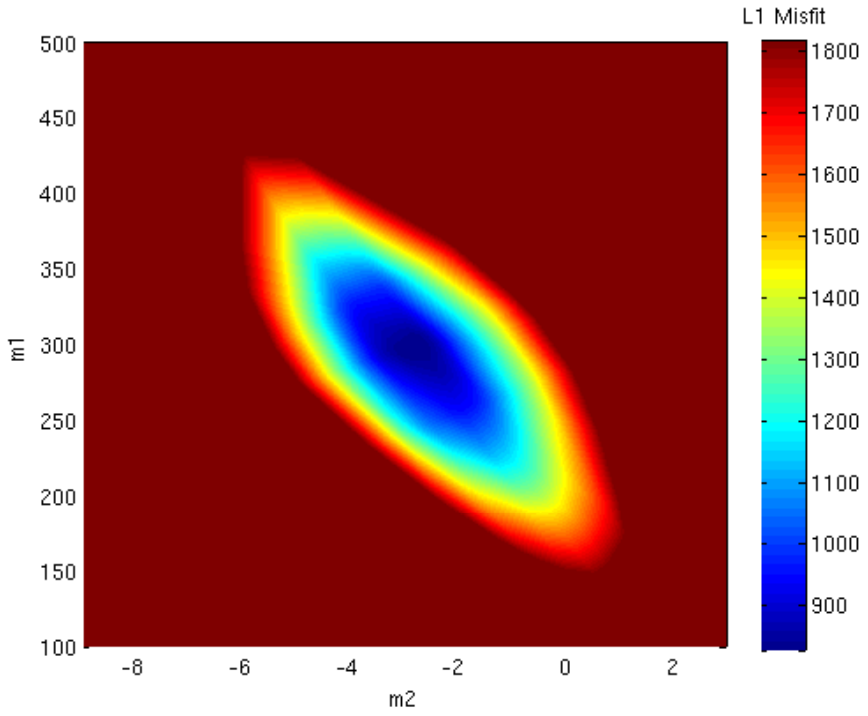

Figure 3: L1 misfits calculated for the whole relevant parameter space of m1 and m2. Lower values (blue) are a better fit with the data. We want a big enough range to see the entire interesting region.

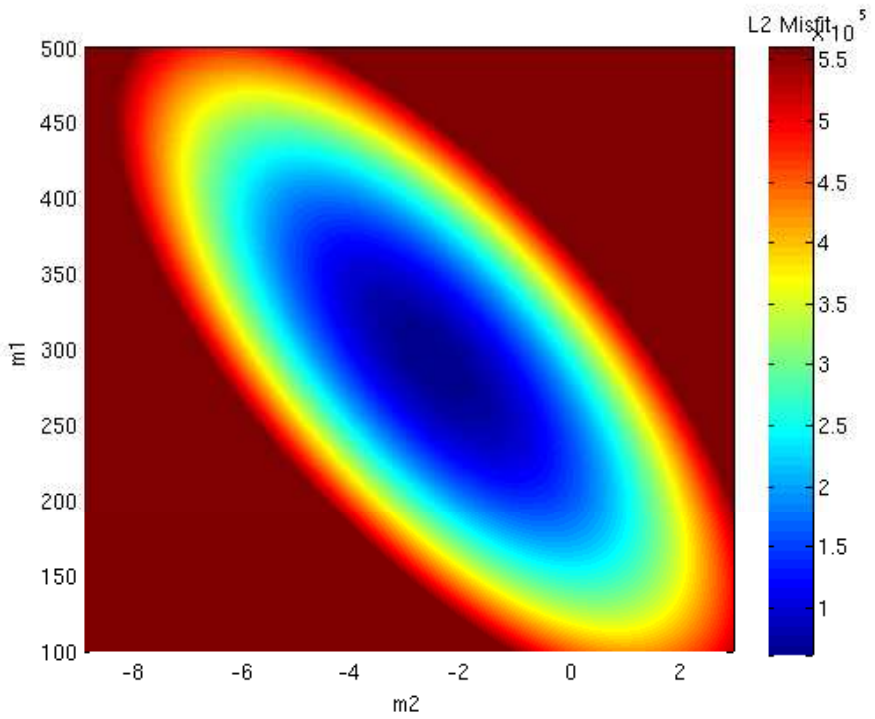

Figure 4: L2 misfits calculated for the whole relevant parameter space of m1 and m2. Lower values (blue) are a better fit with the data. This should have a big enough range to see the entire interesting region.

This should give $m_1 = 291.8387$ and $m_2 = -2.5454$ as the exact LLS solution.

(g) 5 points

The plots produced in d show a great deal of information about the whole model space that is only summarized in a standard result. The standard result shows the best fit parameters and the standard deviation, but it doesn't show any correlation between the parameters. It's not as useful because we can't see the region of good fit in a realistic model space; we can only see what the exact best fits are.

(h) 5 points

Will take any answer as correct as long as it's well defended. The L1 and L2 would be preferred because they are more descriptive of the model space and we can see what realistic model values would give low misfits. Least squares does not show this level of description. However, it is not discretized and instead gives the exact answer. It's also faster and less computationally expensive which could be important in real research questions.

Different error bars shouldn't change the answer much, but if we have some significant outliers then L1 would be preferable to L2. This is due to the fact that the L1 norm is more robust than the L2 norm because it doesn't weight outliers as much. For L1, the absolute value of the error matters, whereas for L2 we look at the square of the error. In the case of an outlier, the squared error will be much larger than the absolute value.

If the error bars are large then we can say some of the misfit can be explained by measurement error. However, in this case the error bars are small compared to the data scatter, so measurement error is not a significant source of misfit.