Problem 1 (graded by Yiran) 50 points

(a) - 8 points

With the misfit function defined as

$$F(\boldsymbol{m}) \quad = \quad \sum_i \frac{(d_i - g_i(\boldsymbol{m}))^2}{2\sigma_d^2}$$

The posterior PDF

$$\begin{aligned} P(\boldsymbol{m}|\boldsymbol{d}) \propto P(\boldsymbol{d}|\boldsymbol{m}) &\propto \exp\left(-F(\boldsymbol{m})\right) \\ &\approx \exp\left(-F(\boldsymbol{m}_0) - \triangledown F(\boldsymbol{m}_0)^T(\boldsymbol{m}-\boldsymbol{m}_0) - \frac{1}{2}(\boldsymbol{m}-\boldsymbol{m}_0)^T\boldsymbol{H}(\boldsymbol{m}_0)(\boldsymbol{m}-\boldsymbol{m}_0)\right) \end{aligned}$$

If m_0 is the least squares solution, $\triangledown F(\boldsymbol{m}_0) = \boldsymbol{0}$

$$P(\boldsymbol{m}|\boldsymbol{d}) \quad \propto \quad \exp\left(-F(\boldsymbol{m}_0) - \frac{1}{2}(\boldsymbol{m}-\boldsymbol{m}_0)^T\boldsymbol{H}(\boldsymbol{m}_0)(\boldsymbol{m}-\boldsymbol{m}_0)\right)$$

has the form of a multivariate Gaussian distribution, and the covariance matrix is

$$\boldsymbol{V} = cov(\boldsymbol{m}) = \boldsymbol{H}(\boldsymbol{m_0})^{-1}$$

Note we want to know the covariance matrix when m_0 is the least squares solution.
In HW2, we didn't include the data error, and

$$\hat{G}_{i,k} = \frac{\partial g_i}{\partial m_k}$$

To include the data error

$$\boldsymbol{H} \approx \frac{\hat{\boldsymbol{G}}^T\hat{\boldsymbol{G}}}{\sigma_d^2}$$

So we need to modify the Hessian in HW2 by dividing by σ_d^2 when we calculate the model covariance matrix.

(See MATLAB code) Assuming $\sigma_d = 0.001$, as given in HW2P1e. The output model covariance matrix is:

```
 0.0101   -0.0123    0.0095    0.0548
-0.0123    0.0198   -0.0156   -0.0853
 0.0095   -0.0156    0.0237    0.1034
 0.0548   -0.0853    0.1034    0.4985
```

(b)- 8 points

The covariance matrix calculated here will be similar to the Monte Carlo simulation in HW2.

Diagonal elements show the variance of each parameter. The square root of the diagonal elements gives the standard deviations: $\sigma_{x_s} = 0.1006$, $\sigma_{y_s} = 0.1405$, $\sigma_{z_s} = 0.1538$, $\sigma_P = 0.7060$. The number is very close to that estimated before: $\sigma_{x_s} = 0.099670, \sigma_{y_s} = 0.137469, \sigma_{z_s} = 0.149719, \sigma_P = 0.691071$.

The off diagonal shows the covariance between parameters. We can scale the covariance matrix to the correlation matrix ($\rho_{xy} = \frac{\sigma_{xy}}{\sigma_x \sigma_y}$). The correlation matrix is:

```
 1.0000   -0.8685    0.6127    0.7721
-0.8685    1.0000   -0.7200   -0.8598
 0.6127   -0.7200    1.0000    0.9520
 0.7721   -0.8598    0.9520    1.0000
```

we see that there are strong trade-offs between model parameters. The strong negative correlation between x_s and y_s, for example, is also shown in the plot in HW2.

(c)- 8 points

Now

$$F(\boldsymbol{m}) = \frac{1}{2\sigma^2}(\boldsymbol{d} - \boldsymbol{Gm})^T(\boldsymbol{d} - \boldsymbol{Gm})$$

where σ is the data error, and

$$\begin{aligned} \boldsymbol{G} &= \begin{bmatrix} \mathbf{1} & \boldsymbol{x} \end{bmatrix} \\ \boldsymbol{m} &= [m1, m2]^T \end{aligned}$$

The Hessian is

$$\boldsymbol{H} = \frac{1}{\sigma^2}\boldsymbol{G}^T\boldsymbol{G}$$

and

$$cov(\boldsymbol{m}) = \boldsymbol{H}^{-1}$$

In this problem, the data error as stated by your friend is strongly underestimated. The plots made in HW2 suggest that the data erros are closer to $\sigma_y \approx 20$. Thus, we should calculate a model covariance matrix using this estimated data error. Your friend tells you his experimental error on y are all ≈ 0.1, but it's possible that he only tells you the instrument error. There are other sources of random errors, for example, the environmental factors, which can be large.

One could also quantitatively re-estimate the data error, with the following calculation.

Let s be our estimate of the data error, $n = 30$ the number of observations, $m = 2$ the number of model parameters, $\boldsymbol{m_{L_2}}$ the least squares solution.

Then, from equation (2.63) in Aster

$$s = \sqrt{\frac{1}{n-m}||\boldsymbol{d} - \boldsymbol{Gm_{L2}}||_2^2} \approx 26.3$$

(see MATLAB code) Using s an estimate for data error, the output covariance matrix is:

```
23.7090   -0.1283
-0.1283    0.0249
```

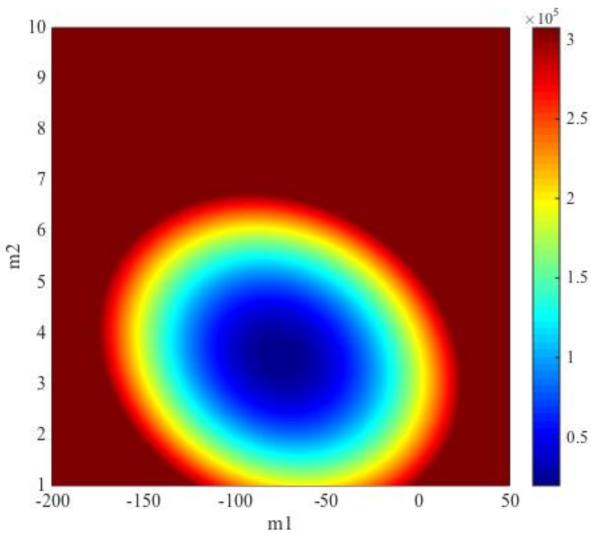
(*) Result using given data error $\sigma = 0.1$
(Not a correct answer, just for your reference)
(see MATLAB code) The output covariance matrix is:

```
1.0e-03 *
 0.3429   -0.0019
-0.0019    0.0004
```

(d)- 8 points

The model covariance is calculated from the distribution $P(\boldsymbol{m}|\boldsymbol{d})$. The square root of the eigenvalues and eigenvectors of the model covariance matrix measure the shape (length and direction of the semiaxes) of the isocontour of $P(\boldsymbol{m}|\boldsymbol{d})$. The shape of the isocontour of $P(\boldsymbol{m}|\boldsymbol{d}) \propto P(\boldsymbol{d}|\boldsymbol{m}) \propto \exp(-F(\boldsymbol{m}))$ is scaled to the isocontour of the error ellipse $F(\boldsymbol{m})$, which we plotted in HW2.

The eigenvectors of the covariance matrix are: $\begin{bmatrix} -0.0054 & -1.0000 \end{bmatrix}$ and $\begin{bmatrix} -1.0000 & 0.0054 \end{bmatrix}$, correspond to an angle of 89.7^o and 179.7^o measured clockwise from x axis - almost a horizontal ellipse; the ratio of the length of the semiaxis is $\sqrt{\lambda_1}/\sqrt{\lambda_2} = 0.0319$. We see that it is true for the error ellipse as shown below (notice the difference in the scale of x and y axis). The negative correlation between m_1 and m_2 is also clear from the off-diagonal element of the model covariance matrix, the analysis on the eigenvectors, and the error ellipse.



(e)- 10 points

First calculate the misfit $F(\boldsymbol{m})$ for the full parameter space.

From Bayes' theorem with a uniform prior,

$$\begin{aligned} P(\boldsymbol{m}|\boldsymbol{d}) &\propto P(\boldsymbol{d}|\boldsymbol{m}) \\ &\propto \exp(-F(\boldsymbol{m})) \end{aligned}$$

Normalize the pdf so its integral over the model space equals 1.

$$P(\boldsymbol{m}|\boldsymbol{d}) = \frac{\exp(-F(\boldsymbol{m}))}{\int \exp(F(\boldsymbol{m}))d\boldsymbol{m}}$$

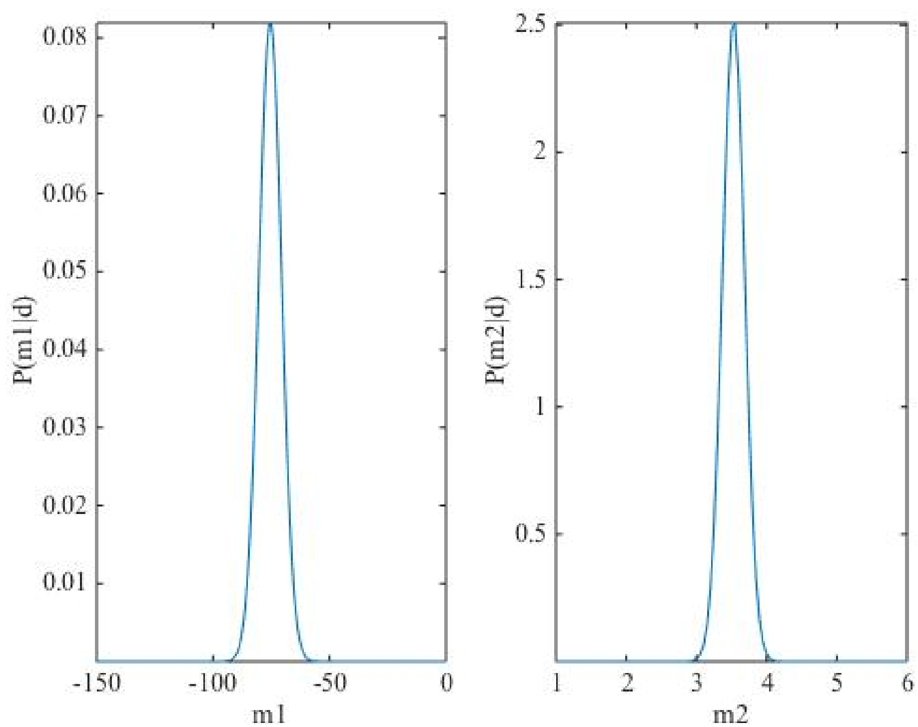
that is

$$P(m_1, m_2|\boldsymbol{d}) = \frac{\exp(-F(m_1, m_2))}{\int\int P(m_1, m_2|d)dm_1dm_2}$$

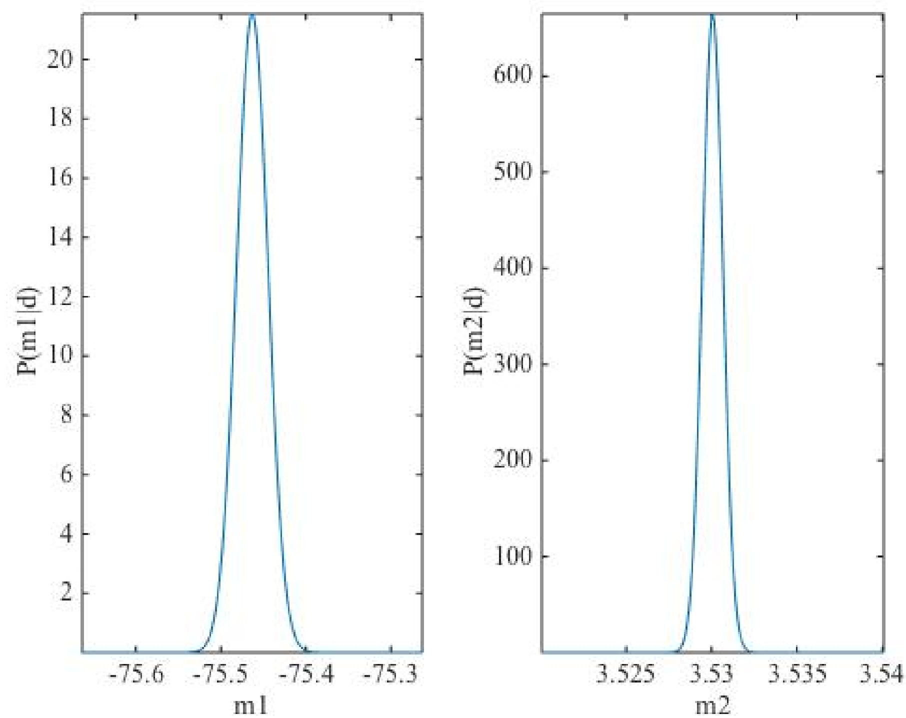
The marginals are:

$$\begin{aligned} P(m_1|d) &= \int P(m_1, m_2|d)dm_2 \\ P(m_2|d) &= \int P(m_1, m_2|d)dm_1 \end{aligned}$$

We can approximate the integral with midpoint numerical integration: $\int f(x)dx = \sum_i f(x_i)\triangle x$. Using the estimated data error $s = 26.3$, we have



(*) Result using given data error $\sigma = 0.1$
(Not a correct answer, just for your reference)



(f)- 8 points

By definition

$$\sigma_{m_i}^2 = E[m_i^2] - (E[m_i])^2 \quad = \quad \int p(m_i|d)m_i^2 dm_i - \left(\int p(m_i|d)m_i dm_i \right)$$

(see MATLAB code)

Using the estimated data error $s = 26.3$, we get $\sigma_{m1} = 4.8692$, $\sigma_{m2} = 0.1577$, which are very close to square root of the diagonal elements of the covariance matrix calculated in (c).

(*) Result using given data error $\sigma = 0.1$

(Not a correct answer, just for your reference)

We get $\sigma_{m1} = 0.0185$, $\sigma_{m2} = 0.0006$, which are very close to square root of the diagonal elements of the covariance matrix calculated in (c).

MATLAB code for (a)-(b)

```
function hw2_1
sigma = 0.001;
% std_mc = [0.099670 0.137469 0.149719 0.691071]';
%%%hw 2 problem 1d
xi = [0 11 15 6 -7 3]';
yi = [0 0 6 13 10 -7]';
ui = [0.103 0.162  0.065  0.036 0.025 0.169]';
M0 = [8 -5 10 30]';  %initial guess, xs ys zs P

[Ms, mcov] = nonlinear_solver(xi,yi,ui,M0,sigma);

% model covariance matrix
disp('model covariance matrix:');
disp(mcov);
disp('standard deviation');
std_m = sqrt(diag(mcov))

% calculate correlation matrix
```

```
s = diag(std_m);
disp('coefficient matrix:');
mcoef = s^(-1)*mcov*s^(-1)
end

function [M, mcov]=nonlinear_solver(x,y,ui,Minit,sigma)

M = Minit;

misfit_old = 0;

misfit = 0;

for ii = 1:1:1000

    misfit = compute_misfit(x,y,M,ui);


    [Grad,Hess] = compute_gradient_approx_hess(x,y,M,misfit);

    deltaM= -(Hess)\Grad';

    M=M + deltaM;

    if (norm(misfit - misfit_old)<1e-7)
        break;
    end

    misfit_old = misfit;
end

mcov = inv(Hess/sigma^2);

%disp(['Number of iterations:',num2str(ii)]);
end

function [ misfit ] = compute_misfit( x,y,M,ui )

xs = M(1);
ys = M(2);
zs = M(3);
p = M(4);

misfit  = ui - p*zs./((x - xs).^2 + (y - ys).^2 + zs^2).^(3/2);

end

function [ Grad, Hess] = compute_gradient_approx_hess( x,y,M,misfit )

xs = M(1);
ys = M(2);
zs = M(3);
p = M(4);

eta = ((x - xs).^2 + (y - ys).^2 + zs^2);

dx = x-xs;
dy = y-ys;

Ghat(:,1) = (3.*p.*zs.*(dx))./((eta).^(5/2));
Ghat(:,2) = (3.*p.*zs.*(dy))./((eta).^(5/2));
Ghat(:,3) = p./(eta).^(3/2) - (3*p.*zs.^2)./(eta).^(5/2);
Ghat(:,4) = zs./(eta).^(3/2);

Grad = -(misfit')* Ghat;

Hess = (Ghat')*Ghat;
%this is the apprximated Hessian;
end
```

MATLAB code for (c)-(f)

```
function hw2_2s
% load data
load ge118_hw2.mat
sigma = 0.1;   % data error
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% least squares
disp('*** start least squares ***');
for i = 1:2
    if i == 1
        disp('(1) use given data error = 0.1');
        data_err = sigma;
    elseif i == 2
        disp('(2) use estimated data error');
        % estimate the data error
        s = sqrt( (misfit(x,y,m1_ls,m2_ls,2)/(length(x)-2) ));
        disp('estimate data error');
        disp(s);
        data_err = s;
    end

    [m1_ls,m2_ls,mcov] = least_square(x,y,data_err);

    disp('model covariance matrix:');
```

```matlab
    disp(mcov);
    %[V,D] = eig(mcov)
    disp('standard deviation of m1 and m2:');
    disp(sqrt(diag(mcov)));
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% grid search
disp('*** start monte carlo ***');

for i = 1:2
    if i == 1
        disp('(1) use given data error = 0.1');
        % narrow the searching range
        % so dm can be small enough to show the marginal pdf
        m1 = (m1_ls - 0.2):0.0005:(m1_ls + 0.2);
        m2 = (m2_ls - 0.01):0.0001:(m2_ls + 0.01);
        data_err = sigma;
    elseif i == 2
        disp('(2) use estimated data error');
        data_err = s;
        m1 = -200:0.1:50;
        m2 = 1:0.05:6;
    end
    dm1 = m1(2) - m1(1);
    dm2 = m2(2) - m2(1);

    % L2 norm
    [m1_l2,m2_l2,err_l2,err_all_l2] = grid_search(x,y,m1,m2,2);

    % joint pdf
    p_m1m2 = exp( -(err_all_l2-min(err_all_l2(:)))/(2*data_err^2) );
    p_m1m2 = p_m1m2/(sum(p_m1m2(:))*dm1*dm2);

    % marginal
    p_m1 = dm2 * sum(p_m1m2,2);
    p_m2 = dm1 * sum(p_m1m2,1);

    m1 = m1(:);
    m2 = m2(:);
    p_m1 = p_m1(:);
    p_m2 = p_m2(:);

    figure;
    subplot(121);
    plot(m1,p_m1);
    xlabel('m1'); ylabel('P(m1|d)');axis tight;
    subplot(122)
    plot(m2,p_m2);
    xlabel('m2'); ylabel('P(m2|d)');axis tight;

    % \sigma m1 m2
    sigma_m1 = sqrt( dm1 * sum(p_m1 .* m1.^2) - (dm1 * sum(p_m1 .* m1)).^2 );
    sigma_m2 = sqrt( dm2 * sum(p_m2 .* m2.^2) - (dm2 * sum(p_m2 .* m2)).^2 );
    disp('standard deviation of m1 and m2:');
    disp([sigma_m1 sigma_m2]);
end
end

% LEAST SQUARE SOLUTION
function [m1, m2, mcov] = least_square(xdata,ydata,sigma)
G = [ones(length(xdata),1) xdata(:)];
tmp = (G'*G)^(-1) * G' * ydata;
m1 = tmp(1);
m2 = tmp(2);
mcov = inv(G'*G/sigma^2);
end

% GRID SEARCH
function [m1_best,m2_best,err_best,err] = grid_search(xdata,ydata,m1,m2,flag)
% error over the model space
err = zeros(length(m1),length(m2));
for i = 1:length(m1)
    for j = 1:length(m2)
        err(i,j) = misfit(xdata,ydata,m1(i),m2(j),flag);
    end
end

% find the optimum solution
[err_best,id] = min(err(:));
[I,J] = ind2sub(size(err),id);
m1_best = m1(I);
m2_best = m2(J);
end

% CALCULATE THE MISFIT
function err = misfit(xdata,ydata,m1,m2,flag)
ypred = m1 + m2 * xdata;
if flag == 1        % L1 norm
    err = sum(abs(ypred-ydata));
elseif flag == 2    % L2 norm
    err = sum((ypred-ydata).^2);
end
end
```

Problem 2 (graded by Kangchen) - 50 points

(a) - 5 points

It is not reasonable to assume that all model parameters have constant priors (-∞ to ∞). For $x_s$and y_s, we should know that it cannot be too far from where the volcano is. For z_s which we defined as depth, it must be positive. For P which is related to pressure, we know that it must be positive too.

(b) - 10 points

We can incorporate this information as a prior by multiplying our likelihood $P(\{d_k\}|\boldsymbol{m})$ by a prior for the model $P(\boldsymbol{m})$. In this case we will use a Gaussian distribution for model parameter p with $\mu = 35$ and $\sigma = 6$.

$$\begin{aligned} P(\boldsymbol{m}) &= e^{-\frac{1}{2}(\frac{p-\mu}{\sigma})^2} \\ P(\boldsymbol{m}|\{d_k\}) &= P(\{d_k\}|\boldsymbol{m})P(\boldsymbol{m}) \\ P(\boldsymbol{m}|\{d_k\}) &= e^{-F(\boldsymbol{m})}e^{-\frac{1}{2}(\frac{p-\mu}{\sigma})^2} \\ P(\boldsymbol{m}|\{d_k\}) &= e^{-F(\boldsymbol{m})-\frac{1}{2}(\frac{p-\mu}{\sigma})^2} \end{aligned}$$

(c.i) - 5 points

$$\begin{aligned} \int_{\phi_1}^{\phi_2} P(\phi)d\phi &= \int_{\phi_1}^{\phi_2} \frac{1}{\phi}d\phi = \ln(\phi_2) - \ln(\phi_1) \\ \int_{k\phi_1}^{k\phi_2} P(\phi)d\phi &= \int_{k\phi_1}^{k\phi_2} \frac{1}{\phi}d\phi = \ln(k\phi_2) - \ln(k\phi_1) \\ &= \ln(k) + \ln(\phi_2) - \ln(k) - \ln(\phi_1) \\ &= \ln(\phi_2) - \ln(\phi_1) \end{aligned}$$

So $P(\phi) = \frac{1}{\phi}$ satisfies the scale independent criterion.

(c.ii) - 5 points

We can again incorporate this information as a prior in our expression for $P(\boldsymbol{m}|\{d_k\})$ by multiplying it times our likelihood (just like in part b). Our expresssion becomes:

$$P(\boldsymbol{m}|\{d_k\}) = e^{-F(\boldsymbol{m})}\frac{1}{p} = e^{-F(\boldsymbol{m})-ln(p)}$$

(d) - 5 points

We want to incorporate both independent pieces of information into our prior. In this case we can multiply the two priors together. Our new expression becomes:

$$P(\boldsymbol{m}|\{d_k\}) = \frac{1}{p}e^{-F(\boldsymbol{m})}e^{-\frac{1}{2}(\frac{p-\mu}{\sigma})^2} = e^{-F_{old}(\boldsymbol{m})-\frac{1}{2}(\frac{p-\mu}{\sigma})^2-\ln(p)}$$

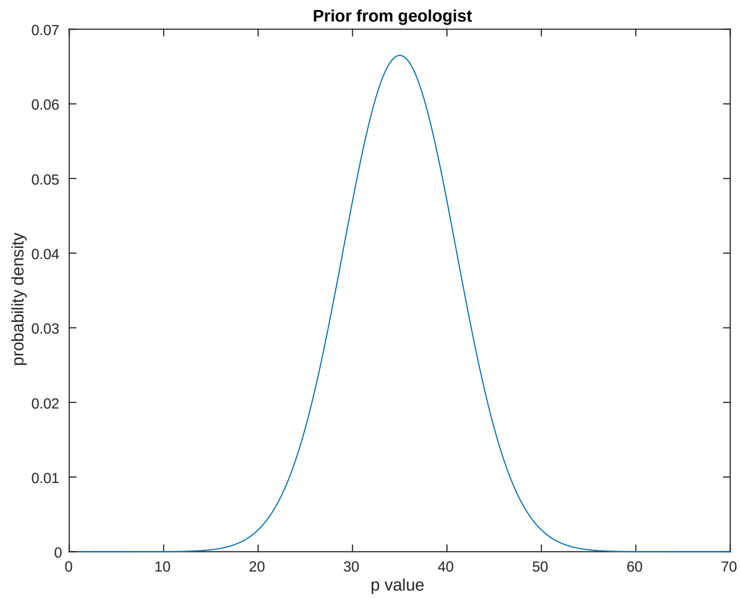


Figure 0.1: Plot of the Gaussian prior for p.

Let's plot the two priors separately and then together to see which information dominates the result. Figure 0.1 is the Gaussian prior. Figure 0.2 is the scale independent prior. Figure 0.3 is their combination. We can see that the Gaussian prior is scaled by the scale invariant prior, but still mainly retains it's shape and thus dominates the result. This is primarily because the scale invariant prior is nearly constant over the range of p that the Gaussian is significantly non-zero.

(e) - 5 points

It doesn't matter in what order we do things as long as priors for analysis are not biased by our data (and they shouldn't be) and your experiment is not affected by prior information. An example of a situation in which the experiment is affected by prior is that if we have some prior information about x_s and y_s which is not the case here, we can conduct our mesasurements closer to those prior points which may increase accuracy of our inversion.

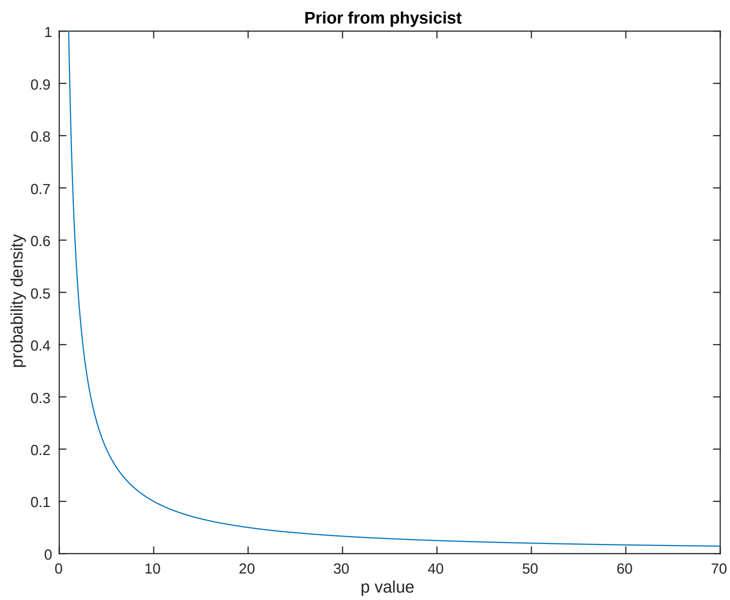


Figure 0.2: Plot of the scale invariant prior for p.

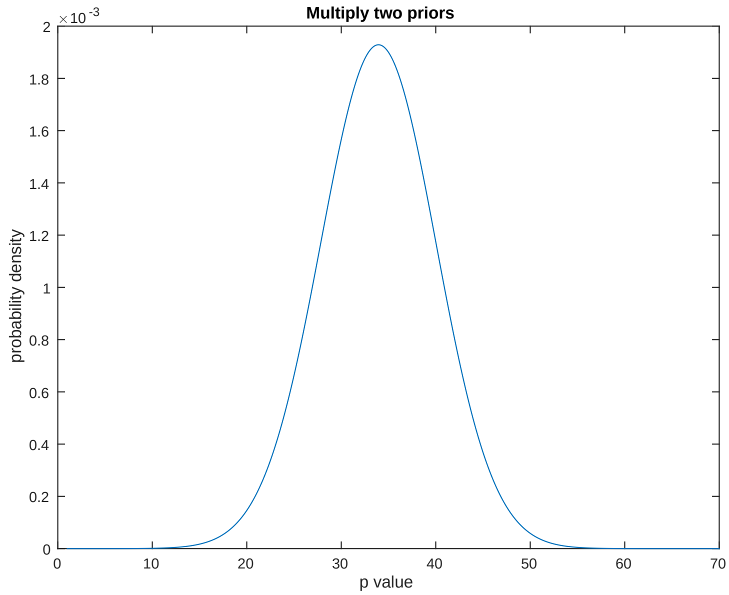


Figure 0.3: Plot of combination of the Gaussian and scale invariant priors for p.

(f) - 15 points

Previous expression: $P_{old}(\boldsymbol{m}|\{d_k\}) = e^{-F_{old}(\boldsymbol{m})}$
New expression: $P(\boldsymbol{m}|\{d_k\}) = \frac{1}{p}e^{-F_{old}(\boldsymbol{m})}e^{-\frac{1}{2}(\frac{p-\mu}{\sigma})^2}$

Previous misfit function: $F_{old}(\boldsymbol{m})$
To find the new misfit function we will need to manipulate our expression to get everything into a single exponential. Let's start with the $\frac{1}{p}$ part:

$$\frac{1}{p} = e^{(\ln(\frac{1}{p})} = e^{-\ln(p)}$$
$$P(\boldsymbol{m}|\{d_k\}) = e^{-\ln(p)}e^{-F_{old}(\boldsymbol{m})}e^{-\frac{1}{2}(\frac{p-\mu}{\sigma})^2}$$

New misfit function: $F(\boldsymbol{m}) = F_{old}(\boldsymbol{m}) + \ln(p) + \frac{1}{2}(\frac{p-35}{6})^2$

We only need to make a couple simple changes to our code from HW2. Instead of just using the L2 norm as our error function, we now need to add our extra two terms to our misfit function:
We will add a term to the last entries of γ and the Hessian to account for the new priors on p.
Add the first derivative of our new part of the misfit to γ:

$$\gamma = \gamma_{old} - \frac{1}{p} - \frac{p-\mu}{\sigma^2}$$

And the second derivative to the last entry of the Hessian:

$$\frac{\partial^2 F}{\partial p^2} = \frac{\partial^2 F_{old}}{\partial p^2} + \frac{-1}{p^2} + \frac{1}{\sigma^2}$$

Here is the full code:

```
%%%problem 2f
xi = [0 11 15 6 -7 3]';
yi = [0 0 6 13 10 -7]';
ui = [0.103 0.162  0.065  0.036 0.025 0.169]';
M0 = [8 -5 10 30]';  %initial guess
Ms = nonlinear_solver(xi,yi,ui,M0);
```

```
function [ Gamma, Hess] = compute_gradient_approx_hess( x,y,M, misfit )

xs = M(1);
ys = M(2);
zs = M(3);
p = M(4);

eta = ((x - xs).^2 + (y - ys).^2 + zs^2);

dx = x-xs;
dy = y-ys;

Ghat(:,1) = (3.*p.*zs.*(dx))./((eta).^(5/2));
Ghat(:,2) = (3.*p.*zs.*(dy))./((eta).^(5/2));
Ghat(:,3) = p./(eta).^(3/2) - (3*p.*zs.^2)./(eta).^(5/2);
Ghat(:,4) = zs./(eta).^(3/2);

sigma_P = 6;

P_mean = 35;

Gamma = -(misfit')* Ghat;
```

```
Gamma = Gamma + [0 0 0 1/p + (p-P_mean)/(sigma_P^2)];

Hess = (Ghat')*Ghat ;

Hess(4,4) = Hess(4,4) - 1/p^2 + 1/(sigma_P^2) ;

%this is the apprximated Hessian;

end
```

```
function [M]=nonlinear_solver(x,y,ui,Minit)

M = Minit;

misfit_old = 0;

misfit = 0;

for ii = 1:1:1000

misfit = compute_misfit(x,y,M,ui);


[Gamma,Hess] = compute_gradient_approx_hess(x,y,M,misfit);

deltaM= (Hess)\Gamma';

M = M - deltaM;

if (norm(misfit - misfit_old)<1e-7)
    break;
end

misfit_old = misfit;

end

disp(['Number of iterations:',num2str(ii)]);
end
```

```
function [ misfit ] = compute_misfit( x,y,M,ui )

xs = M(1);
ys = M(2);
zs = M(3);
p = M(4);

misfit = ui - p*zs./((x - xs).^2 + (y - ys).^2 + zs^2).^(3/2);

end
```

Our best fit solution is now:

$$\begin{bmatrix} x_s \\ y_s \\ z_s \\ p \end{bmatrix} = \begin{bmatrix} 8.5332 \\ -5.7685 \\ 12.1910 \\ 33.9391 \end{bmatrix}$$

Comparing with old solution, the new solution has p value closer to 35 which is given by geologist's prior.

Problem 3 (graded by Kangchen) - bonus 15 points

(a)-5 points

The posterior probability is given by

$$p(\mu|x) \propto \left(\sum_{k=1}^{N}(x_k-\mu)^2\right)^{-\frac{N-1}{2}} = \exp\left(-\frac{N-1}{2}\ln\left[\sum_{k=1}^{N}(x_k-\mu)^2\right]\right). \tag{0.1}$$

Minima of F maximize the probability density function. We have that

$$F(\mu) = \frac{N-1}{2}\ln\left(\sum_{k=1}^{N}(x_k-\mu)^2\right), \tag{0.2}$$

$$\frac{\partial F}{\partial \mu} = -(N-1)\frac{\sum_{k=1}^{N}(x_k-\mu)}{\sum_{k=1}^{N}(x_k-\mu)^2} \tag{0.3}$$

$$\frac{\partial^2 F}{\partial \mu^2} = (N-1)\frac{N\sum_{k=1}^{N}(x_k-\mu)^2 - 2(\sum_{k=1}^{N}(x_k-\mu))^2}{\left(\sum_{k=1}^{N}(x_k-\mu)^2\right)^2}. \tag{0.4}$$

Taking the derivative with respect to μ and setting it to zero yields the minimum μ_0

$$\frac{\partial F}{\partial \mu} = -(N-1)\frac{\sum_{k=1}^{N}(x_k-\mu)}{\sum_{k=1}^{N}(x_k-\mu)^2} = 0$$

Because $\sum_{k=1}^{N}(x_k-\mu)^2 > 0$, we have $\sum_{k=1}^{N}(x_k-\mu) = 0$

$$\mu_0 = \frac{1}{N}\sum_{k=1}^{N} x_k, \tag{0.5}$$

For the second derivative at the best fit solution we have

$$\frac{\partial^2 F}{\partial \mu^2}(\mu=\mu_0) = (N-1)\frac{N\sum_{k=1}^{N}(x_k-\mu_0)^2 - 2(\sum_{k=1}^{N}(x_k-\mu_0))^2}{\left(\sum_{k=1}^{N}(x_k-\mu_0)^2\right)^2} \tag{0.6}$$

$$= (N-1)\frac{N\sum_{k=1}^{N}(x_k-\mu_0)^2 - 2(\sum_{k=1}^{N}(x_k-\frac{1}{N}\sum_{k=1}^{N}x_k))^2}{\left(\sum_{k=1}^{N}(x_k-\mu_0)^2\right)^2} \tag{0.7}$$

$$= (N-1)\frac{N\sum_{k=1}^{N}(x_k-\mu_0)^2}{\left(\sum_{k=1}^{N}(x_k-\mu_0)^2\right)^2} \tag{0.8}$$

$$= \frac{N(N-1)}{\sum_{k=1}^{N}(x_k-\mu_0)^2} \tag{0.9}$$

The standard deviation σ_μ then amounts to

$$\sigma_\mu = \frac{1}{F''(\mu_0)^{1/2}} = \frac{S}{\sqrt{N}}, \tag{0.10}$$

where

$$S = \sqrt{\frac{1}{N-1}\sum_{k=1}^{N}(x_k - \mu_0)^2} \tag{0.11}$$

is the sample variance of the data.

(b)-5 points

We assume $\boldsymbol{Q} = [\boldsymbol{e}_1, \boldsymbol{e}_2]$ where $||\boldsymbol{e}_1|| = 1$,$||\boldsymbol{e}_2|| = 1$. We can write $\boldsymbol{e}_1 = (cos(\theta), sin(\theta))^T$ and $\boldsymbol{e}_2 = (cos(\phi), sin(\phi))^T$. Since $\boldsymbol{Q}$ is an orthongonal matrix, $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ are orthogonal. That means $cos(\theta)cos(\phi) + sin(\theta)sin(\phi) = cos(\theta - \phi) = 0$. So $\theta - \phi = \pi/2$ or $-\pi/2$. If we assume that $\phi = \theta + \pi/2$, then $\boldsymbol{Q} = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix}$. This can be understood as rotating a vector by θ degree counterclockwise. Another way to interpret what $\boldsymbol{Q}$ did is that $\boldsymbol{Q}^T \begin{pmatrix} x \\ y \end{pmatrix}$ gives the projection of $\begin{pmatrix} x \\ y \end{pmatrix}$ onto $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ where $\boldsymbol{Q} = (\boldsymbol{e}_1, \boldsymbol{e}_2)$.

(c)-5 points

We need to show that $\sigma_x^2 = \frac{B}{AB - C^2}$. We are dealing with an integral of the f

$$\sigma_x^2 = \frac{\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} dxdy\, x^2 \exp\left(-\frac{1}{2}x^T Ax\right)}{\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} dxdy\, \exp\left(-\frac{1}{2}x^T Ax\right)} \tag{0.12}$$

$$= \frac{\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} dxdy\, x^2 \exp\left(-\frac{1}{2}(Ax^2 + 2Cxy + By^2)\right)}{\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} dxdy\, \exp\left(-\frac{1}{2}(Ax^2 + 2Cxy + By^2)\right)} \tag{0.13}$$

$$= \frac{\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} dxdy\, x^2 \exp\left(-\frac{1}{2}(A - C^2/B)x^2 - \frac{1}{2}B(y + Cx/B)^2\right)}{\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} dxdy\, \exp\left(-\frac{1}{2}(A - C^2/B)x^2 - \frac{1}{2}B(y + Cx/B)^2\right)} \tag{0.14}$$

$$= \frac{\int_{-\infty}^{+\infty} dx\, x^2 \exp\left(-\frac{1}{2}(A - C^2/B)x^2\right) \int_{-\infty}^{+\infty} dy \exp\left(-\frac{1}{2}B(y + Cx/B)^2\right)}{\int_{-\infty}^{+\infty} dx\, \exp\left(-\frac{1}{2}(A - C^2/B)x^2\right) \int_{-\infty}^{+\infty} dy \exp\left(-\frac{1}{2}B(y + Cx/B)^2\right)} \tag{0.15}$$

$$= \frac{\int_{-\infty}^{+\infty} dx\, x^2 \exp\left(-\frac{1}{2}(A - C^2/B)x^2\right)}{\int_{-\infty}^{+\infty} dx\, \exp\left(-\frac{1}{2}(A - C^2/B)x^2\right)} \tag{0.16}$$

Note that from 0.15 to 0.16:
$\int_{-\infty}^{+\infty} dy \exp\left(-\frac{1}{2}B(y + Cx/B)^2\right) = \int_{-\infty+cx/B}^{+\infty+cx/B} dy' \exp\left(-\frac{1}{2}By'^2\right) = \int_{-\infty}^{+\infty} dy' \exp\left(-\frac{1}{2}By'^2\right)$
So that integral is not related to x. That's why we can cancel out that term from both numerator and denominator.

If we set $\sigma^2 = 1/(A - C^2/B)$, we arrive at

$$\sigma_x^2 = \frac{\int dx\, x^2 \exp\left(-\frac{1}{2}x^2/\sigma^2\right)}{\int dx\, \exp\left(-\frac{1}{2}x^2/\sigma^2\right)} \tag{0.17}$$

, which as we have seen in previous homework sets and solutions implies

$$\sigma_x^2 = \langle x^2 \rangle = \sigma^2 = \frac{1}{A - \frac{C^2}{B}} = \frac{B}{AB - C^2}. \tag{0.18}$$