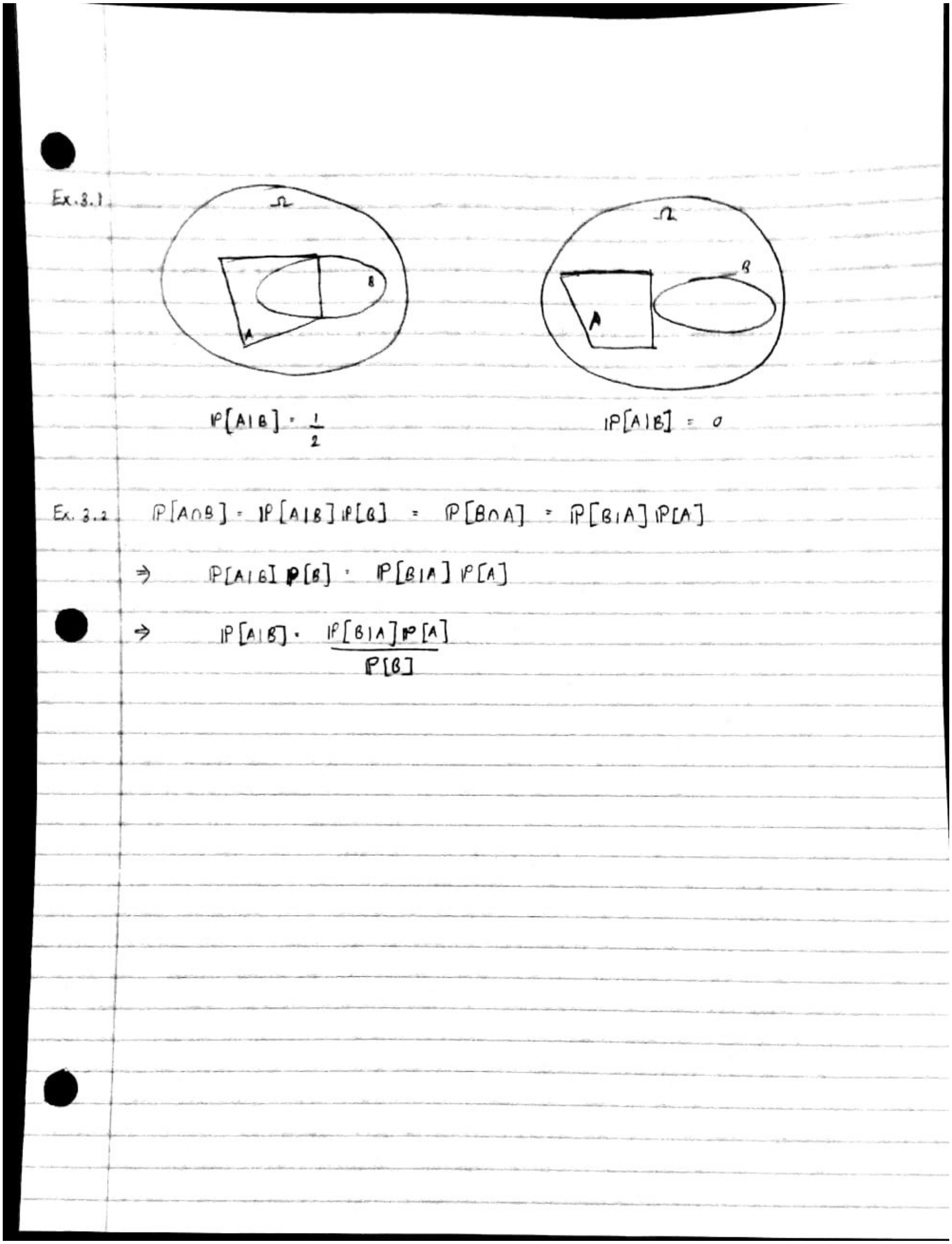
EM 397 HW #2

Group 4 (Soovadeep Bakshi, William Ruys, Wenbo Zhang, Akshay Kumar Varanasi)

Ex. 3.1 and Ex. 3.2:



Ex. 3.1

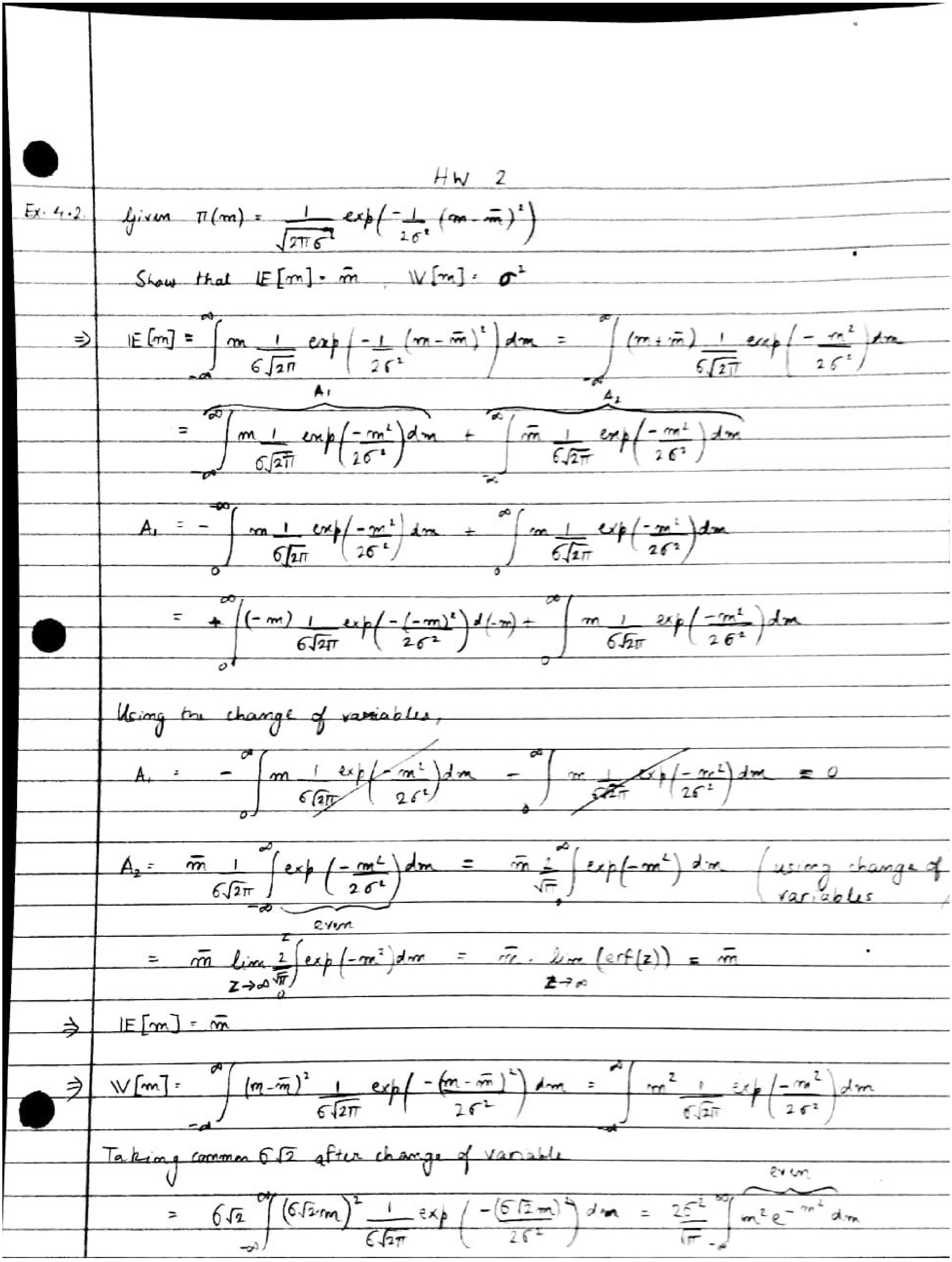
$\mathbb{P}[A|B] = \frac{1}{2}$ $\qquad$ $\mathbb{P}[A|B] = 0$

Ex. 3.2 $\mathbb{P}[A \cap B] = \mathbb{P}[A|B]\,\mathbb{P}[B] = \mathbb{P}[B \cap A] = \mathbb{P}[B|A]\,\mathbb{P}[A]$

$\Rightarrow \quad \mathbb{P}[A|B]\,\mathbb{P}[B] = \mathbb{P}[B|A]\,\mathbb{P}[A]$

$\Rightarrow \quad \mathbb{P}[A|B] = \dfrac{\mathbb{P}[B|A]\,\mathbb{P}[A]}{\mathbb{P}[B]}$

Ex. 4.2:



HW 2

Ex. 4.2. given $\pi(m) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(m-\bar{m})^2\right)$

Show that $\mathbb{E}[m] = \bar{m}$, $\mathbb{V}[m] = \sigma^2$

⇒ $\mathbb{E}[m] = \int_{-\infty}^{\infty} m \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2\sigma^2}(m-\bar{m})^2\right) dm = \int_{-\infty}^{\infty} (m+\bar{m}) \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{m^2}{2\sigma^2}\right) dm$

$$= \overbrace{\int_{-\infty}^{\infty} m \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-m^2}{2\sigma^2}\right) dm}^{A_1} + \overbrace{\int_{-\infty}^{\infty} \bar{m} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-m^2}{2\sigma^2}\right) dm}^{A_2}$$

$$A_1 = -\int_{0}^{-\infty} m \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-m^2}{2\sigma^2}\right) dm + \int_{0}^{\infty} m \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-m^2}{2\sigma^2}\right) dm$$

$$= +\int_{0}^{\infty} (-m) \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(-m)^2}{2\sigma^2}\right) d(-m) + \int_{0}^{\infty} m \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-m^2}{2\sigma^2}\right) dm$$

Using the change of variables,

$$A_1 = -\int_{0}^{\infty} m \cancel{\frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-m^2}{2\sigma^2}\right)} dm - \int_{0}^{\infty} m \cancel{\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{m^2}{2\sigma^2}\right)} dm = 0$$

$A_2 = \bar{m} \frac{1}{\sigma\sqrt{2\pi}} \underbrace{\int_{-\infty}^{\infty} \exp\left(\frac{-m^2}{2\sigma^2}\right) dm}_{\text{even}} = \bar{m} \frac{2}{\sqrt{\pi}} \int_{0}^{\infty} \exp(-m^2)\, dm$ (using change of variables

$$= \bar{m} \lim_{z\to\infty} \frac{2}{\sqrt{\pi}} \int_{0}^{z} \exp(-m^2)\, dm = \bar{m} \cdot \lim_{z\to\infty} (\text{erf}(z)) = \bar{m}$$

⇒ $\mathbb{E}[m] = \bar{m}$

⇒ $\mathbb{V}[m] = \int_{-\infty}^{\infty} (m-\bar{m})^2 \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(m-\bar{m})^2}{2\sigma^2}\right) dm = \int_{-\infty}^{\infty} m^2 \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-m^2}{2\sigma^2}\right) dm$

Taking common $\sigma\sqrt{2}$ after change of variable

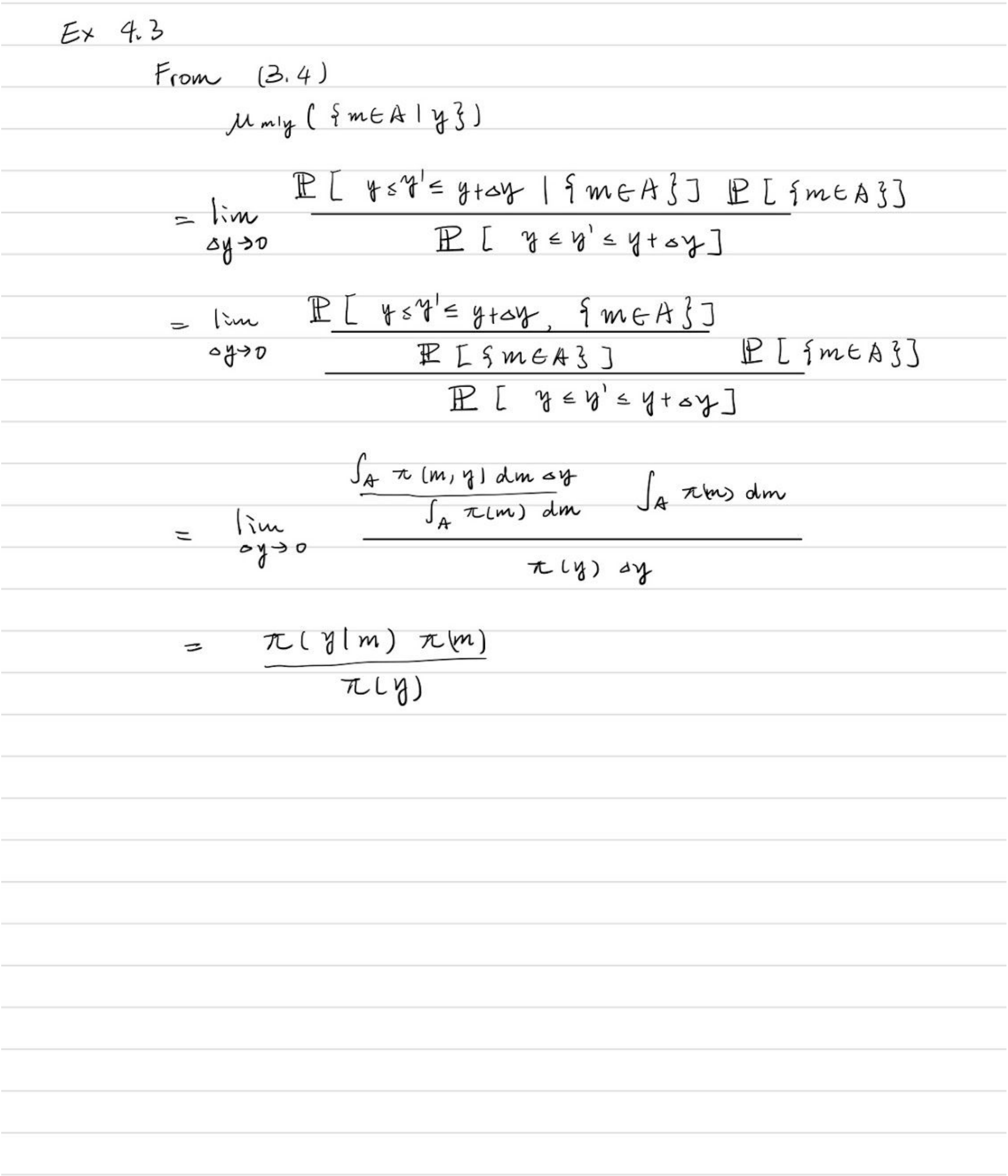
$$= \sigma\sqrt{2} \int_{-\infty}^{\infty} (\sigma\sqrt{2}m)^2 \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(\sigma\sqrt{2}m)^2}{2\sigma^2}\right) dm = \frac{2\sigma^2}{\sqrt{\pi}} \overbrace{\int_{-\infty}^{\infty} m^2 e^{-m^2} dm}^{\text{even}}$$

$$= \frac{4\sigma^2}{\sqrt{\pi}} \int_0^\infty m^2 e^{-m^2} \, dm = \frac{4\sigma^2}{\sqrt{\pi}} \int_0^\infty (\sqrt{m})^2 \left(\frac{1}{2\sqrt{m}}\right) e^{-m} \, dm$$

(using change of variables $\underset{\substack{\downarrow \\ \text{new } m}}{x^2} = m$)

$$= \frac{4\sigma^2}{\sqrt{\pi}} \int_0^\infty \frac{1}{2} m^{\left(\frac{1}{2}\right)} e^{-m} \, dm = \frac{4\sigma^2}{\sqrt{\pi}} \cdot \frac{1}{2} \cdot \Gamma\left(\frac{3}{2}\right) = \sigma^2$$

Ex. 4.3:



Ex 4.3

From (3.4)

$$\mu_{m|y}(\{m \in A \mid y\})$$

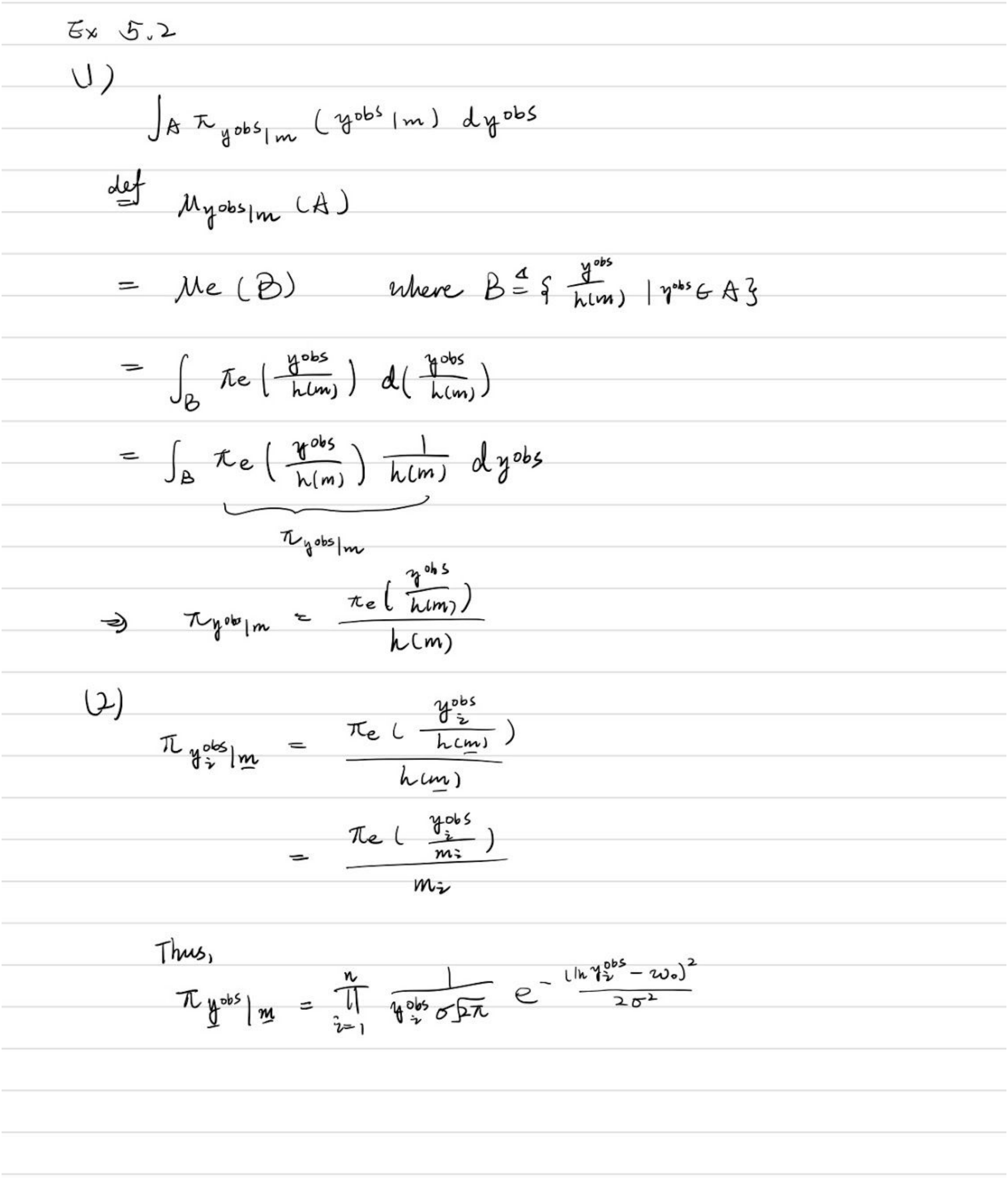
$$= \lim_{\Delta y \to 0} \frac{\mathbb{P}[\, y \le y' \le y + \Delta y \mid \{m \in A\}] \; \mathbb{P}[\{m \in A\}]}{\mathbb{P}[\, y \le y' \le y + \Delta y]}$$

$$= \lim_{\Delta y \to 0} \frac{\dfrac{\mathbb{P}[\, y \le y' \le y + \Delta y ,\ \{m \in A\}]}{\mathbb{P}[\{m \in A\}]} \; \mathbb{P}[\{m \in A\}]}{\mathbb{P}[\, y \le y' \le y + \Delta y]}$$

$$= \lim_{\Delta y \to 0} \frac{\dfrac{\int_A \pi(m, y)\, dm\, \Delta y}{\int_A \pi(m)\, dm} \; \int_A \pi(m)\, dm}{\pi(y)\, \Delta y}$$

$$= \frac{\pi(y \mid m)\, \pi(m)}{\pi(y)}$$

Ex. 5.2:



Ex 5.2

(1)

$$\int_A \pi_{y^{obs}|m}(y^{obs}|m)\, dy^{obs}$$

$$\overset{def}{=} \mu_{y^{obs}|m}(A)$$

$$= \mu_e(B) \qquad \text{where } B \overset{\Delta}{=} \{ \frac{y^{obs}}{h(m)} \mid y^{obs} \in A \}$$

$$= \int_B \pi_e\left(\frac{y^{obs}}{h(m)}\right) d\left(\frac{y^{obs}}{h(m)}\right)$$

$$= \int_B \underbrace{\pi_e\left(\frac{y^{obs}}{h(m)}\right) \frac{1}{h(m)}}_{\pi_{y^{obs}|m}} dy^{obs}$$

$$\Rightarrow \pi_{y^{obs}|m} = \frac{\pi_e\left(\frac{y^{obs}}{h(m)}\right)}{h(m)}$$

(2)

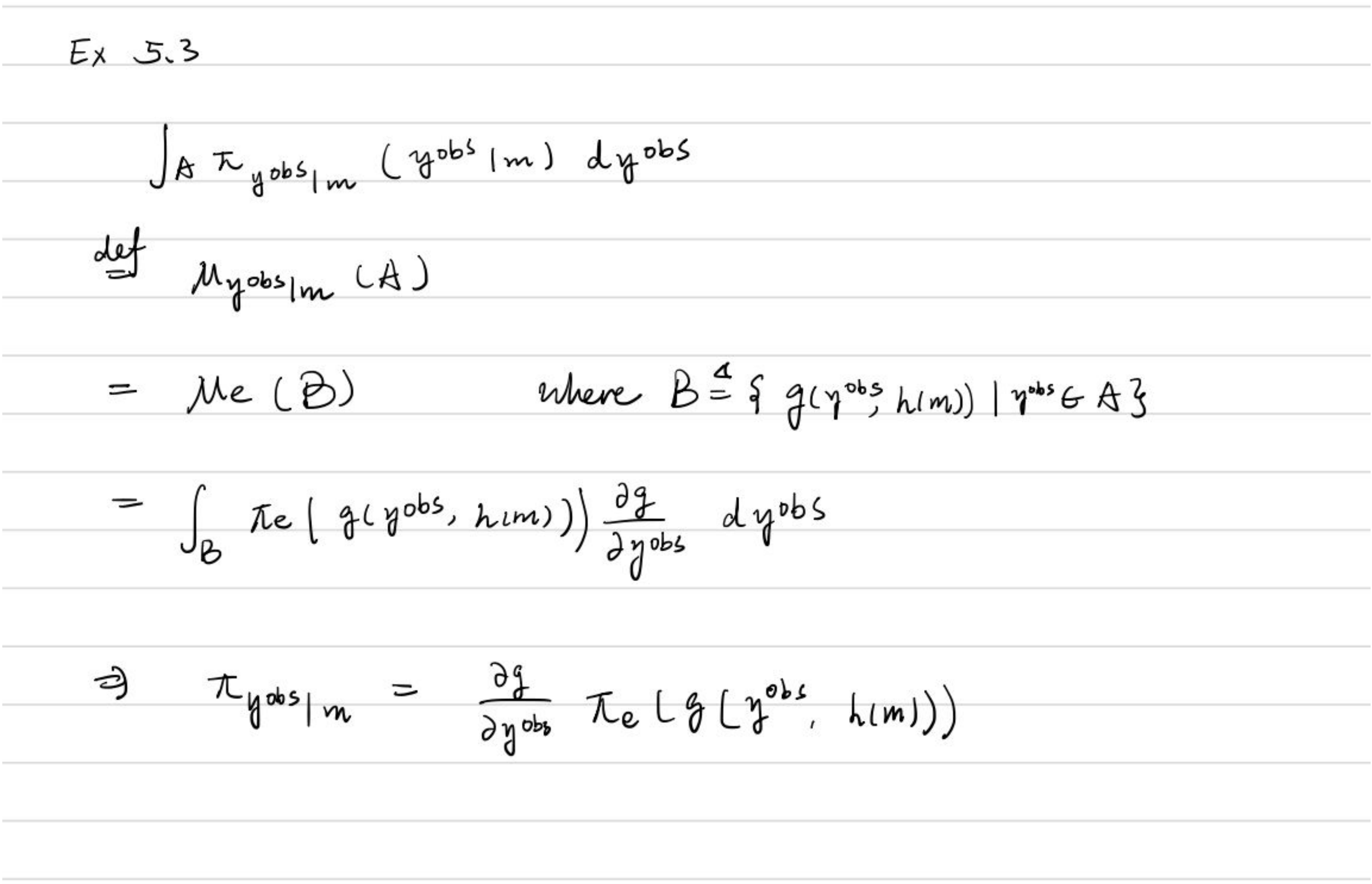
$$\pi_{y_i^{obs}|\underline{m}} = \frac{\pi_e\left(\frac{y_i^{obs}}{h(\underline{m})}\right)}{h(\underline{m})}$$

$$= \frac{\pi_e\left(\frac{y_i^{obs}}{m_i}\right)}{m_i}$$

Thus,

$$\pi_{\underline{y}^{obs}|\underline{m}} = \prod_{i=1}^{n} \frac{1}{y_i^{obs}\sigma\sqrt{2\pi}} e^{-\frac{(\ln y_i^{obs} - w_0)^2}{2\sigma^2}}$$

Ex. 5.3:



Ex 5.3

$$\int_A \pi_{y^{obs}|m} (y^{obs} | m) \, d y^{obs}$$

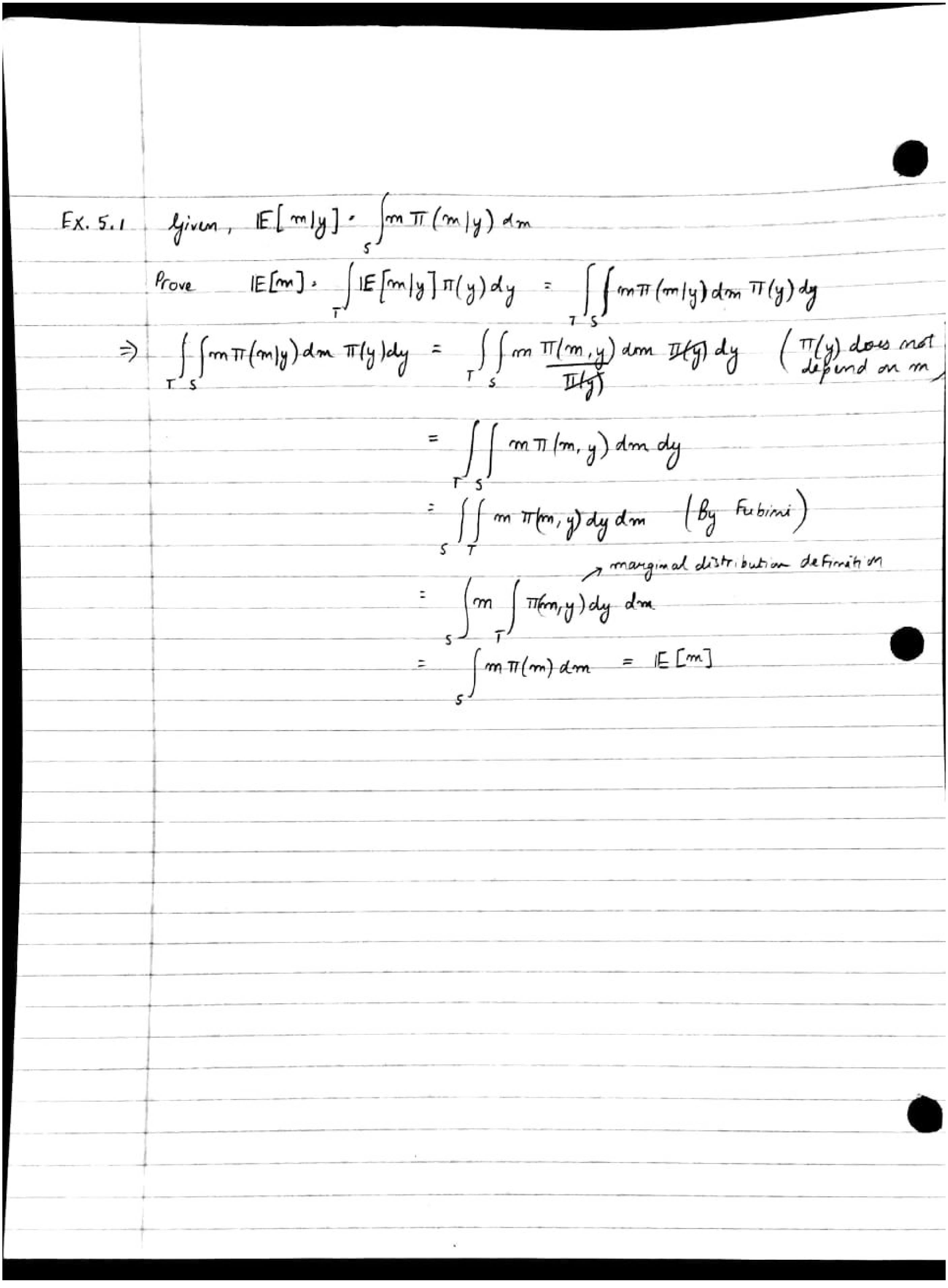
$$\overset{def}{=} \mu_{y^{obs}|m} (A)$$

$$= \mu_e (B) \qquad \text{where } B \overset{\Delta}{=} \{ g(y^{obs}, h(m)) \mid y^{obs} \in A \}$$

$$= \int_B \pi_e (g(y^{obs}, h(m))) \frac{\partial g}{\partial y^{obs}} \, d y^{obs}$$

$$\Rightarrow \quad \pi_{y^{obs}|m} = \frac{\partial g}{\partial y^{obs}} \pi_e (g (y^{obs}, h(m)))$$

Ex. 5.1:



Ex. 5.1 Given, $\mathbb{E}[m|y] = \int_S m\,\pi(m|y)\,dm$

Prove $\mathbb{E}[m] = \int_T \mathbb{E}[m|y]\,\pi(y)\,dy = \int_T\int_S m\,\pi(m|y)\,dm\,\pi(y)\,dy$

$\Rightarrow \int_T\int_S m\,\pi(m|y)\,dm\,\pi(y)\,dy = \int_T\int_S m\,\frac{\pi(m,y)}{\cancel{\pi(y)}}\,dm\,\cancel{\pi(y)}\,dy$ ($\pi(y)$ does not depend on m)

$= \int_T\int_S m\,\pi(m,y)\,dm\,dy$

$= \int_S\int_T m\,\pi(m,y)\,dy\,dm$ (By Fubini)

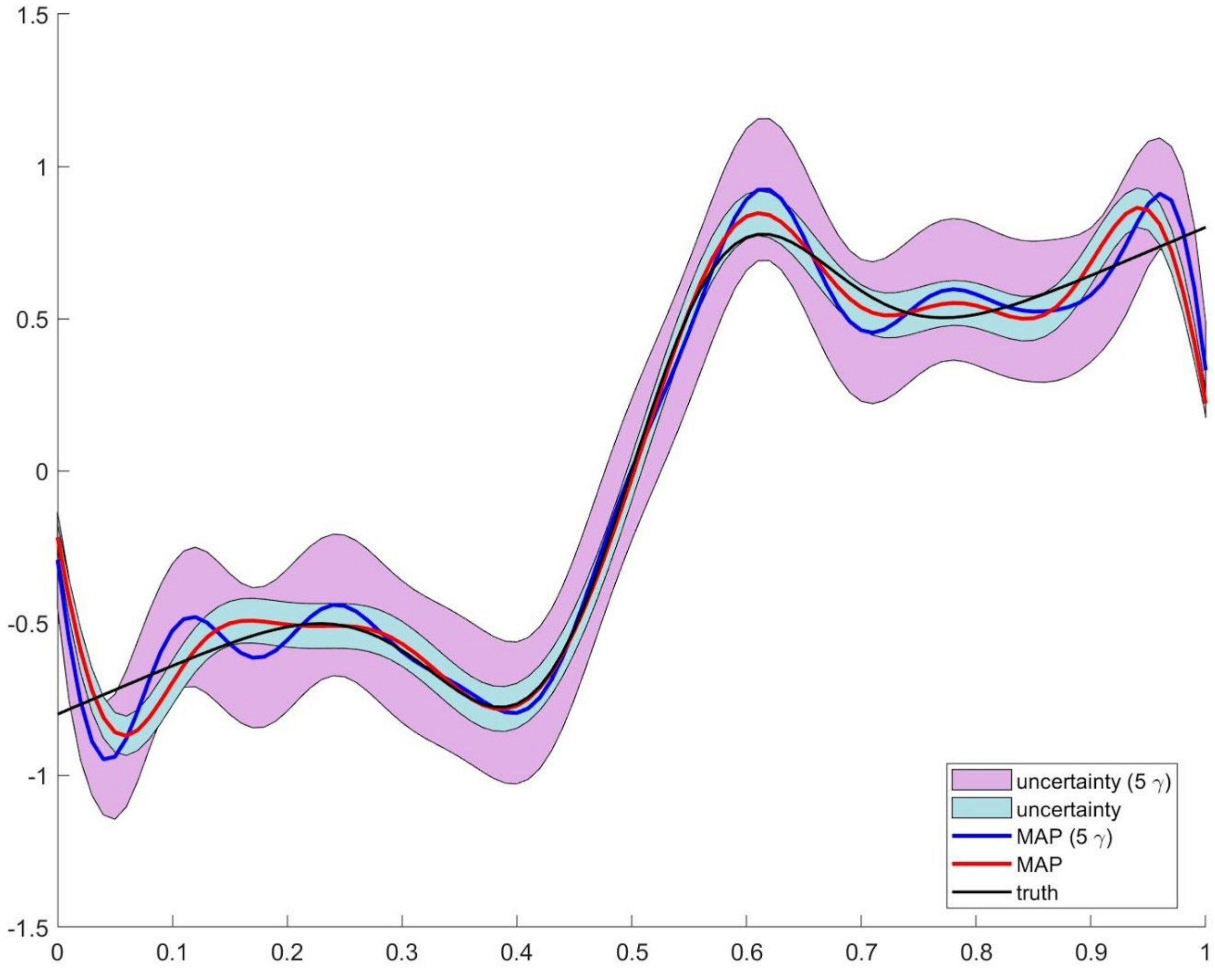
marginal distribution definition

$= \int_S m \int_T \pi(m,y)\,dy\,dm$

$= \int_S m\,\pi(m)\,dm = \mathbb{E}[m]$

Ex. 7.4:

To make the prior less strong, the parameter □ can be changed. Higher values of □ lead to more contribution from the data misfit, as is shown in the plot below.



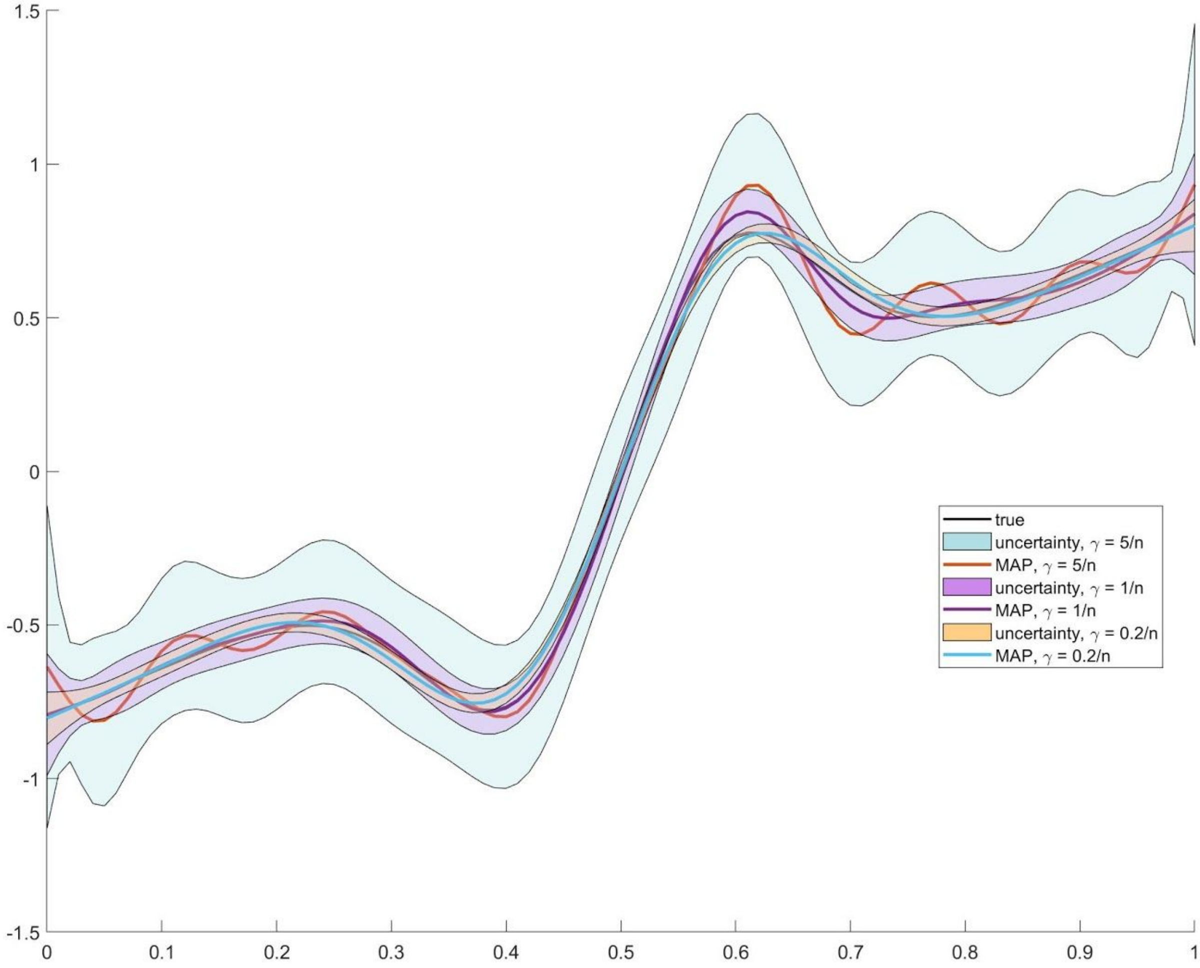
At higher values of □ (chosen to be 5□), the boundary conditions are better satisfied than the default case. However, it is clear that the uncertainty increases, as seen above. This is because the variance is proportional to the value of □, given the fact that the posterior distribution is also Gaussian.

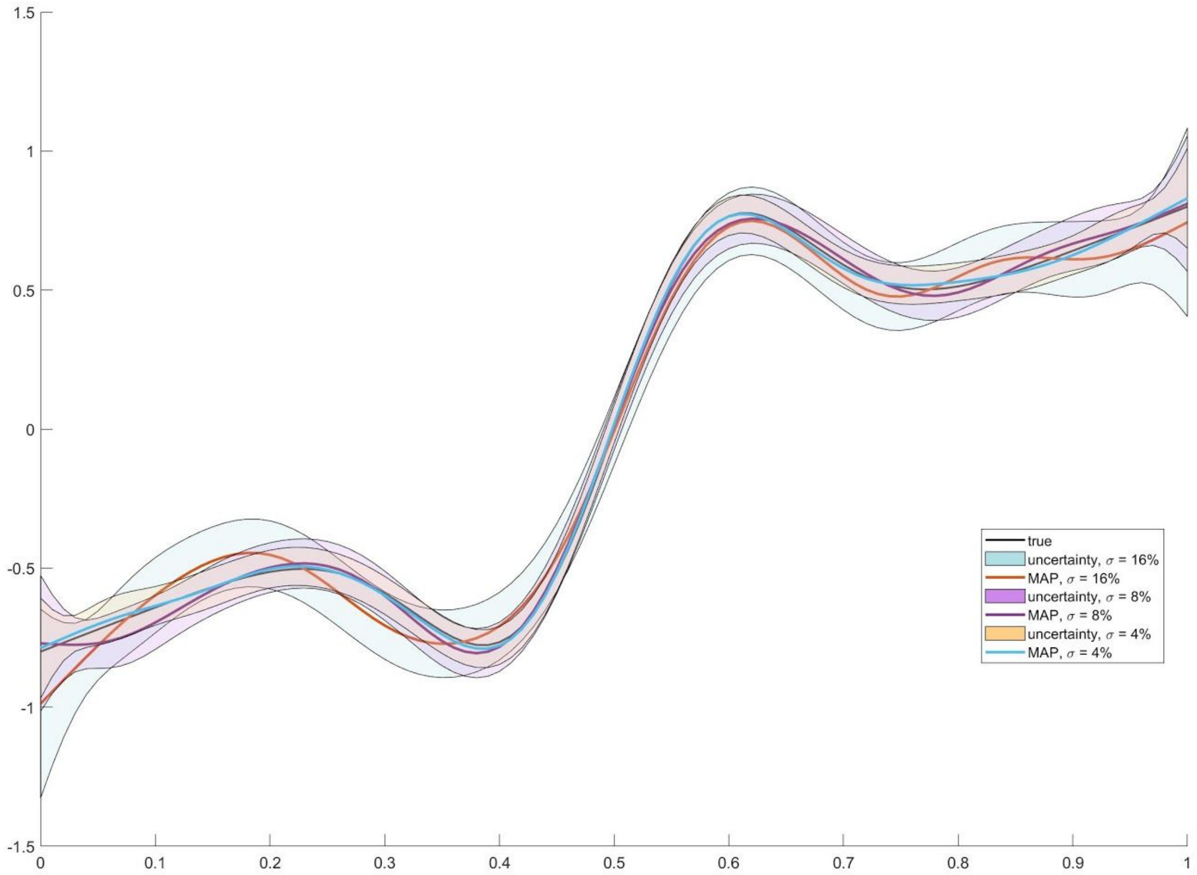
Code:

```
clear all
close all

% explore the posterior with smooth priors and without hyper-parameters.

rand('state',20);
randn('state',18);

% Mesh
n = 100;
s = linspace(0,1,n+1)'; t = s;

% Prior flag
PriorFlag = 1; % 1: L_D
               % 2: L_A
% discretize the deblurring kernel
beta = 0.05;
a = 1/sqrt(2*pi*beta^2)*exp(-0.5*(1/beta^2)*t.^2);
A = 1/n*toeplitz(a);

% Truth
xtrue = 10*(t-0.5).*exp(-0.5*1e2*(t-0.5).^2) -0.8 + 1.6*t;

%%------------additive noise-----------
noise = 5;       % Noise level in percentages of the max. of noiseless signal
y0 = A*xtrue;    % Noiseless signal
sigma = max(abs(y0))*noise/100;                % STD of the additive noise
y = y0 + sigma*randn(n+1,1);


%%------------Prior construction----------
% standard deviation of the innovation
gamma = 1/(1*n);

% Construct the L_D matrix
if PriorFlag == 1
  L = diag(ones(n+1,1)) - diag(0.5*ones(n,1),1) - diag(0.5*ones(n,1),-1);
elseif PriorFlag == 2
  L_D = diag(ones(n+1,1)) - diag(0.5*ones(n,1),1) - diag(0.5*ones(n,1),-1);
  % you should never do this, but we do it anyway for convenience
  L_Dinv = inv(L_D);
  Dev = sqrt(gamma^2 * diag(L_Dinv * L_Dinv'));

  delta = gamma./ Dev(floor(n/2));
  L = L_D;
  L(1,:) = 0; L(1,1) = delta;
  L(end,:) = 0; L(end,end) = delta;
else
  error('not supported')
end

% Calculating the MAP estimate and posterior variances, by least squares
xmean = [(1/sigma)*A;1/gamma*L]\[(1/sigma)*y;zeros(n+1,1)];
Gamma_post = inv((1/sigma^2)*A'*A + 1/gamma^2*L'*L);

% Plotting the MAP estimate and the 2*STD envelope

% Defining different shades of blue for plotting
 shades = [176 224 230;
             135 206 235;
             135 206 255;
             126 192 238;
             108 166 205];
   shades = 1/255*shades;

STD = sqrt(diag(Gamma_post));
xhigh = xmean + 2*STD;
xlow = xmean - 2*STD;

%% New gamma
gamma = 5/(1*n);

% Construct the L_D matrix
if PriorFlag == 1
  L = diag(ones(n+1,1)) - diag(0.5*ones(n,1),1) - diag(0.5*ones(n,1),-1);
elseif PriorFlag == 2
  L_D = diag(ones(n+1,1)) - diag(0.5*ones(n,1),1) - diag(0.5*ones(n,1),-1);
  % you should never do this, but we do it anyway for convenience
  L_Dinv = inv(L_D);
  Dev = sqrt(gamma^2 * diag(L_Dinv * L_Dinv'));

  delta = gamma./ Dev(floor(n/2));
  L = L_D;
  L(1,:) = 0; L(1,1) = delta;
  L(end,:) = 0; L(end,end) = delta;
else
  error('not supported')
end

% Calculating the MAP estimate and posterior variances, by least squares
xmean2 = [(1/sigma)*A;1/gamma*L]\[(1/sigma)*y;zeros(n+1,1)];
Gamma_post2 = inv((1/sigma^2)*A'*A + 1/gamma^2*L'*L);

% Plotting the MAP estimate and the 2*STD envelope

% Defining different shades of blue for plotting
 shades2 = [224 176 230;
             206 135 235;
             206 135 255;
             192 126 238;
             166 108 205];
 shades2 = 1/255*shades2;

STD = sqrt(diag(Gamma_post2));
xhigh2 = xmean2 + 2*STD;
xlow2 = xmean2 - 2*STD;


figure
axes('fontsize',12);
hold on
fill([t;t(n+1:-1:1)],[xlow2;xhigh2(n+1:-1:1)],shades2(1,:))
fill([t;t(n+1:-1:1)],[xlow;xhigh(n+1:-1:1)],shades(1,:))
plot(t,xmean2,'b-','LineWidth',2)
plot(t,xmean,'r-','LineWidth',2)
plot(t,xtrue,'k-','LineWidth',1.5)
legend('uncertainty (5 \gamma)','uncertainty','MAP (5 \gamma)','MAP','truth','location','best')
```

Ex. 7.5:

For different values of □ and σ, the results are as shown below. For the first case, σ is kept constant at 5%, whereas for the second case, □ in taken to be 1/n.





Code:

```
clear all
close all

% explore the posterior with smooth priors and without hyper-parameters.

rand('state',20);
randn('state',18);

% Mesh
n = 100;
s = linspace(0,1,n+1)'; t = s;

% Prior flag
PriorFlag = 2; % 1: L_D
               % 2: L_A

% discretize the deblurring kernel
beta = 0.05;
a = 1/sqrt(2*pi*beta^2)*exp(-0.5*(1/beta^2)*t.^2);
A = 1/n*toeplitz(a);

% Truth
xtrue = 10*(t-0.5).*exp(-0.5*1e2*(t-0.5).^2) -0.8 + 1.6*t;

%%------------additive noise-----------
noise = 5;       % Noise level in percentages of the max. of noiseless signal
y0 = A*xtrue;    % Noiseless signal
sigma = max(abs(y0))*noise/100;                % STD of the additive noise
```

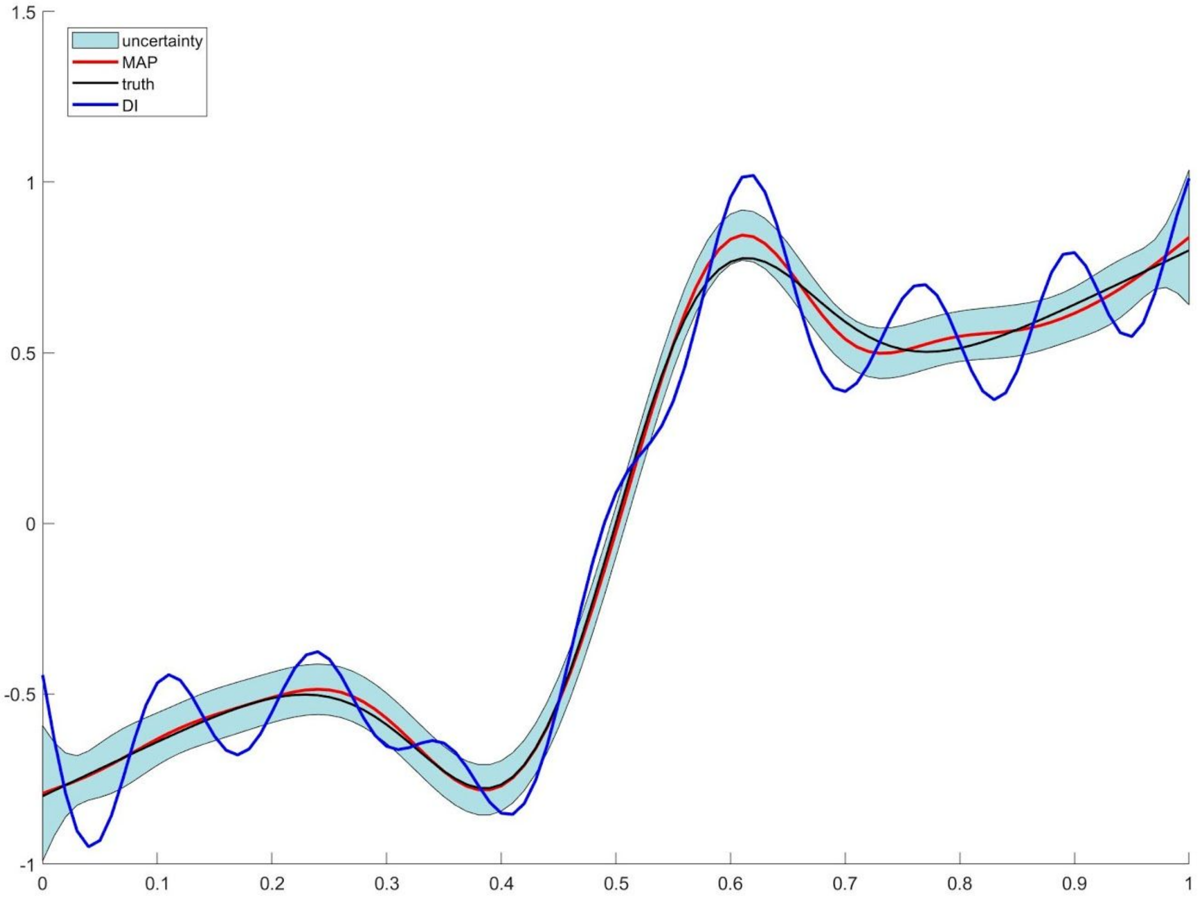
```
y = y0 + sigma*randn(n+1,1);


%%------------Prior construction----------
% standard deviation of the innovation
gamma = 5/(1*n);
shades = [176 224 230;
                 206 135 235;
                 255 206 135;
                 16 10 238;
                 10 166 205];
       shades = 1/255*shades;

figure(1);
axes('fontsize',12);
hold on
plot(t,xtrue,'k-','LineWidth',1.5)
legendInfo{1} = ['true'];

for i = 1:3

    if PriorFlag == 1
      L = diag(ones(n+1,1)) - diag(0.5*ones(n,1),1) - diag(0.5*ones(n,1),-1);
    elseif PriorFlag == 2
      L_D = diag(ones(n+1,1)) - diag(0.5*ones(n,1),1) - diag(0.5*ones(n,1),-1);
      % you should never do this, but we do it anyway for convenience
      L_Dinv = inv(L_D);
      Dev = sqrt(gamma^2 * diag(L_Dinv * L_Dinv'));

      delta = gamma./ Dev(floor(n/2));
      L = L_D;
      L(1,:) = 0; L(1,1) = delta;
      L(end,:) = 0; L(end,end) = delta;
    else
      error('not supported')
    end

    % Calculating the MAP estimate and posterior variances, by least squares
    xmean = [(1/sigma)*A;1/gamma*L]\[(1/sigma)*y;zeros(n+1,1)];
    Gamma_post = inv((1/sigma^2)*A'*A + 1/gamma^2*L'*L);

    STD = sqrt(diag(Gamma_post));
    xhigh = xmean + 2*STD;
    xlow = xmean - 2*STD;

    h = fill([t;t(n+1:-1:1)],[xlow;xhigh(n+1:-1:1)],shades(i,:));
    set(h,'facealpha',.3)
    plot(t,xmean,'LineWidth',2)
    legendInfo{2*i}=['uncertainty, \gamma = ' num2str(gamma*n) '/n'];
    legendInfo{2*i+1} = ['MAP, \gamma = ' num2str(gamma*n) '/n'];
    gamma = gamma/5;

end
legend(legendInfo,'location','best')

figure(2);
axes('fontsize',12);
hold on
noise = 16;
gamma = 1/(1*n);
% xtrue = [0;xtrue];
plot(t,xtrue,'k-','LineWidth',1.5)
legendInfo{1} = ['true'];

for i = 1:3
```

```
    y0 = A*xtrue;    % Noiseless signal
    sigma = max(abs(y0))*noise/100;                % STD of the additive noise
    y = y0 + sigma*randn(n+1,1);

    if PriorFlag == 1
      L = diag(ones(n+1,1)) - diag(0.5*ones(n,1),1) - diag(0.5*ones(n,1),-1);
    elseif PriorFlag == 2
      L_D = diag(ones(n+1,1)) - diag(0.5*ones(n,1),1) - diag(0.5*ones(n,1),-1);
      % you should never do this, but we do it anyway for convenience
      L_Dinv = inv(L_D);
      Dev = sqrt(gamma^2 * diag(L_Dinv * L_Dinv'));

      delta = gamma./ Dev(floor(n/2));
      L = L_D;
      L(1,:) = 0; L(1,1) = delta;
      L(end,:) = 0; L(end,end) = delta;
    else
      error('not supported')
    end

    % Calculating the MAP estimate and posterior variances, by least squares
    xmean = [(1/sigma)*A;1/gamma*L]\[(1/sigma)*y;zeros(n+1,1)];
    Gamma_post = inv((1/sigma^2)*A'*A + 1/gamma^2*L'*L);

    STD = sqrt(diag(Gamma_post));
    xhigh = xmean + 2*STD;
    xlow = xmean - 2*STD;

    h = fill([t;t(n+1:-1:1)],[xlow;xhigh(n+1:-1:1)],shades(i,:));
    set(h,'facealpha',.2)
    plot(t,xmean,'LineWidth',2)
    legendInfo{2*i}=['uncertainty, \sigma = ' num2str(noise) '%'];
    legendInfo{2*i+1} = ['MAP, \sigma = ' num2str(noise) '%'];
    noise = noise/2;

end
legend(legendInfo,'location','best')
```

Ex. 7.6:

For the deterministic inversion, a conjugate gradient method is used as in HW #1 with Mozorov’s discrepancy principle is used as a stopping criterion. (σ = 5%, □ = 1/n).

The results are shown below.



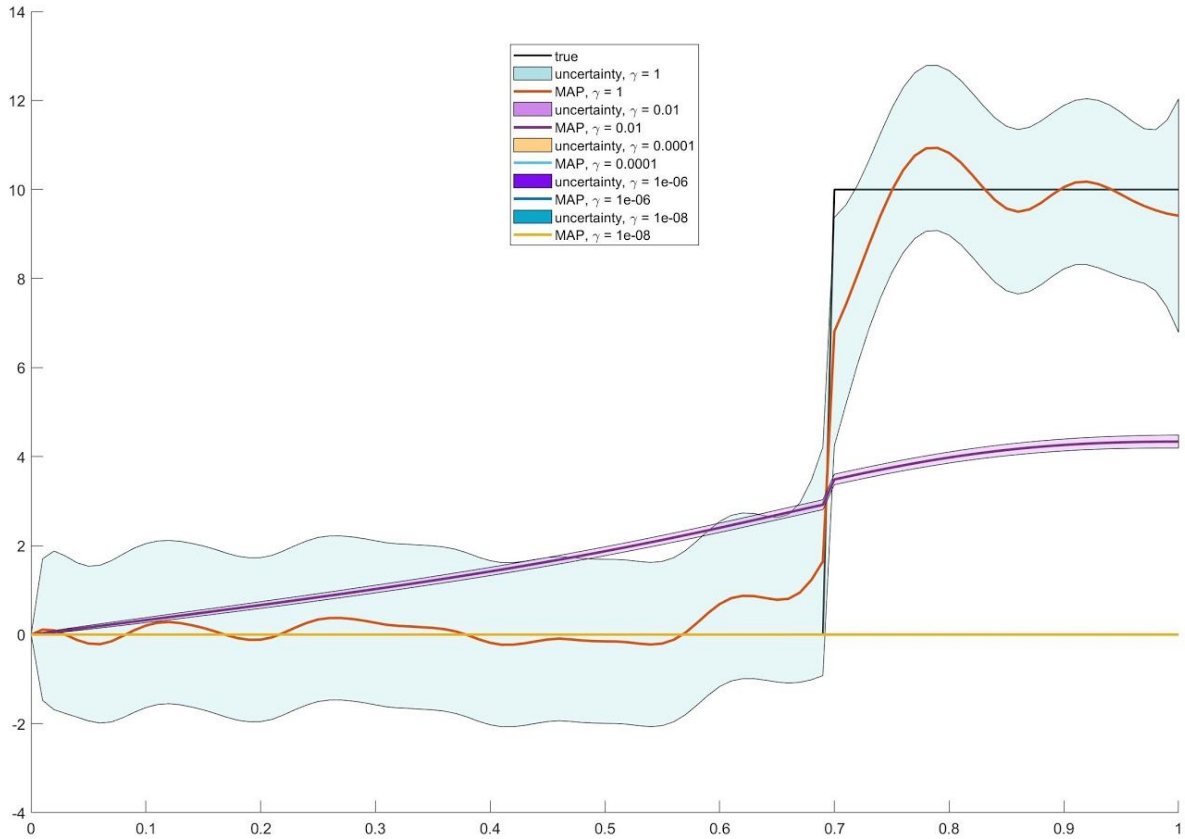
The deterministic inversion scheme cannot extract the true solution because of the noisy data.

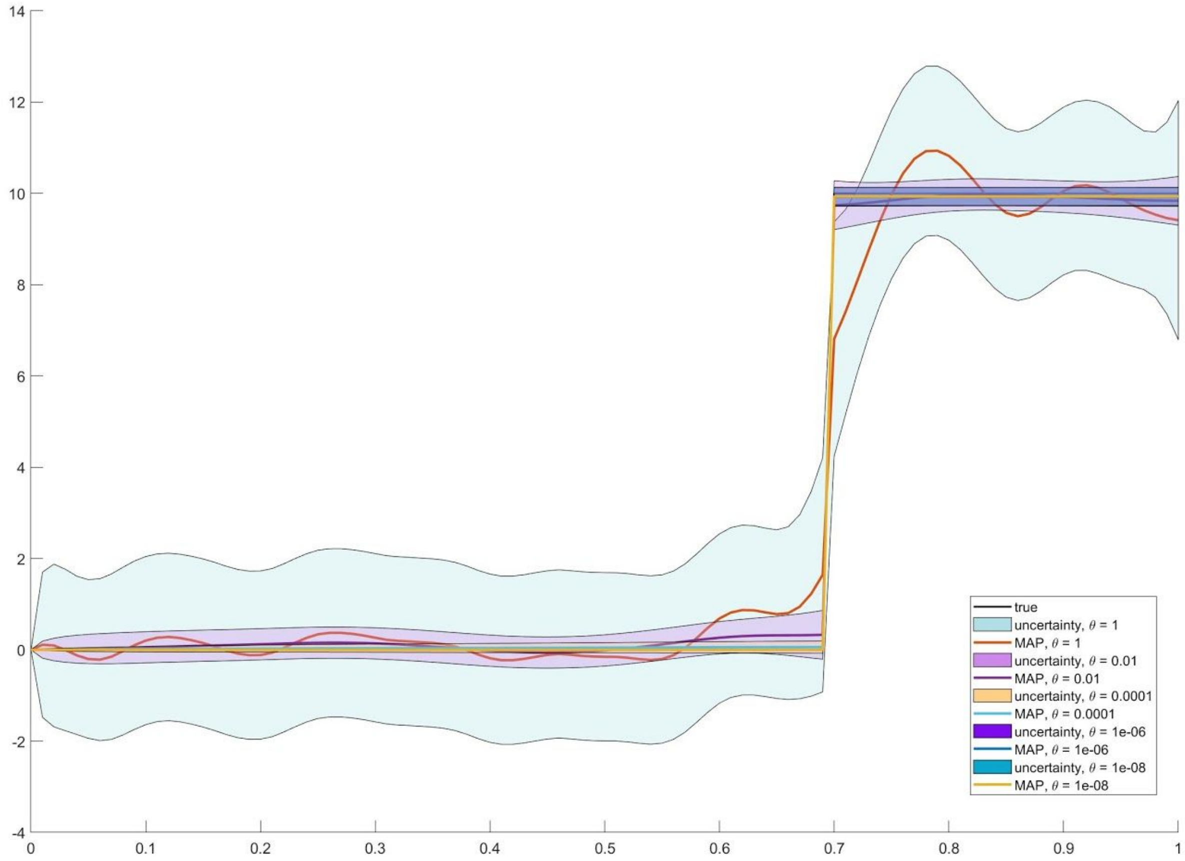
Code:

```
clear all
close all

% explore the posterior with smooth priors and without hyper-parameters.

rand('state',20);
randn('state',18);

% Mesh
n = 100;
s = linspace(0,1,n+1)'; t = s;

% Prior flag
PriorFlag = 2; % 1: L_D
               % 2: L_A
% discretize the deblurring kernel
beta = 0.05;
a = 1/sqrt(2*pi*beta^2)*exp(-0.5*(1/beta^2)*t.^2);
A = 1/n*toeplitz(a);

% Truth
xtrue = 10*(t-0.5).*exp(-0.5*1e2*(t-0.5).^2) -0.8 + 1.6*t;

%%------------additive noise-----------
noise = 5;       % Noise level in percentages of the max. of noiseless signal
y0 = A*xtrue;    % Noiseless signal
sigma = max(abs(y0))*noise/100;                 % STD of the additive noise
y = y0 + sigma*randn(n+1,1);


%%------------Prior construction----------
% standard deviation of the innovation
gamma = 1/(1*n);

% Construct the L_D matrix
if PriorFlag == 1
  L = diag(ones(n+1,1)) - diag(0.5*ones(n,1),1) - diag(0.5*ones(n,1),-1);
elseif PriorFlag == 2
  L_D = diag(ones(n+1,1)) - diag(0.5*ones(n,1),1) - diag(0.5*ones(n,1),-1);
  % you should never do this, but we do it anyway for convenience
  L_Dinv = inv(L_D);
  Dev = sqrt(gamma^2 * diag(L_Dinv * L_Dinv'));

  delta = gamma./ Dev(floor(n/2));
  L = L_D;
  L(1,:) = 0; L(1,1) = delta;
  L(end,:) = 0; L(end,end) = delta;
else
  error('not supported')
end

% Calculating the MAP estimate and posterior variances, by least squares
xmean = [(1/sigma)*A;1/gamma*L]\[(1/sigma)*y;zeros(n+1,1)];
Gamma_post = inv((1/sigma^2)*A'*A + 1/gamma^2*L'*L);

% Plotting the MAP estimate and the 2*STD envelope

% Defining different shades of blue for plotting
 shades = [176 224 230;
             135 206 235;
             135 206 255;
             126 192 238;
             108 166 205];
   shades = 1/255*shades;

STD = sqrt(diag(Gamma_post));
xhigh = xmean + 2*STD;
xlow = xmean - 2*STD;

[xObsCG,idx] = CGLS(A,y,xtrue,sigma,n);

figure
axes('fontsize',12);
hold on
fill([t;t(n+1:-1:1)],[xlow;xhigh(n+1:-1:1)],shades(1,:))
plot(t,xmean,'r-','LineWidth',2)
plot(t,xtrue,'k-','LineWidth',1.5)
plot(t,xObsCG,'b-','LineWidth',2)
legend('uncertainty','MAP','truth','DI','location','best')
```

Ex. 7.7:

With decreasing values of □ and θ, the following results can be seen. For the first case, θ is maintained at a constant value of 1. For the second case, □ = 1.





Code:

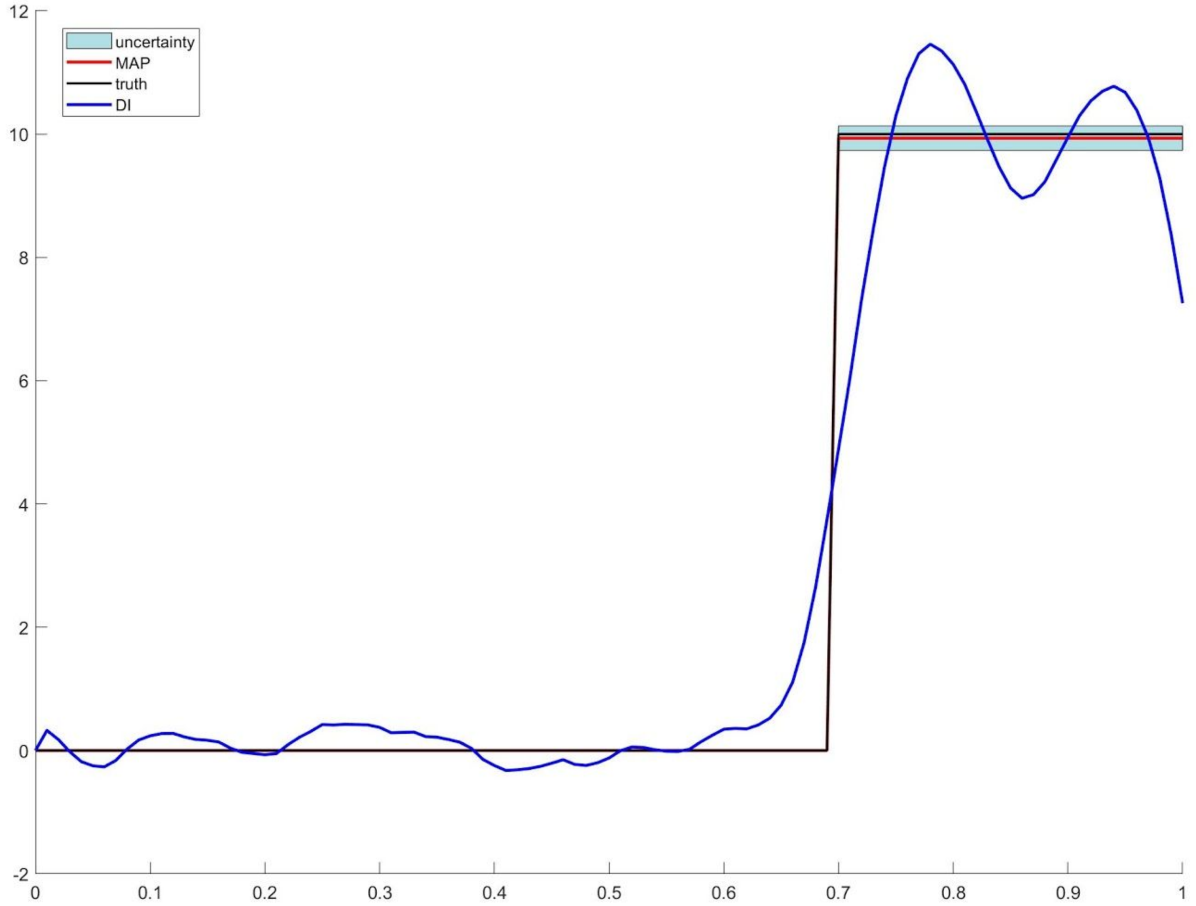
```
clear all

% explore the posterior with non-smooth prior and without hyper-parameters.

rand('state',20);
randn('state',18);

% Mesh
n = 100;
s = linspace(0,1,n+1)'; t = s(2:end);

% discretize the kernel
beta = 0.05;
a = 1/sqrt(2*pi*beta^2)*exp(-0.5*(1/beta^2)*t.^2);
A = 1/n*toeplitz(a);

% Truth
xtrue = zeros(n,1);
xtrue(70:end) = 10;

%%------------additive noise-----------
noise = 5;       % Noise level in percentages of the max. of noiseless signal
y0 = A*xtrue;    % Noiseless signal
sigma = max(abs(y0))*noise/100;                 % STD of the additive noise
y = y0 + sigma*randn(n,1);


%%------------Prior construction----------
```

```
% standard deviation of the innovation
gamma = 1;
thetaVal = 1;
t = [0;t];
shades = [176 224 230;
                 206 135 235;
                 255 206 135;
                 126 10 238;
                 10 166 205];
       shades = 1/255*shades;

figure(1);
axes('fontsize',12);
hold on
xtrue = [0;xtrue];
plot(t,xtrue,'k-','LineWidth',1.5)
legendInfo{1} = ['true'];

for i = 1:5

    L = eye(n) - diag(ones(n-1,1),-1);
    theta = thetaVal*ones(n,1);
    theta(70) = 10;
    M = diag(1./sqrt(theta));
    L = M * L;

    % Calculating the MAP estimate and posterior variances by least squares
    xmean = [(1/sigma)*A;1/gamma*L]\[(1/sigma)*y;zeros(n,1)];
    Gamma_post = inv((1/sigma^2)*A'*A + 1/gamma^2*L'*L);

    % Plotting the MAP estimate and the 2*STD envelope

    % Defining different shades of blue for plotting

    STD = sqrt(diag(Gamma_post));
    xhigh = xmean + 2*STD;
    xlow = xmean - 2*STD;
    xlow = [0;xlow];
    xhigh = [0;xhigh];
    xmean = [0;xmean];
    h = fill([t;t(n+1:-1:1)],[xlow;xhigh(n+1:-1:1)],shades(i,:));
    set(h,'facealpha',.3)
    plot(t,xmean,'LineWidth',2)
    legendInfo{2*i}=['uncertainty, \theta = ' num2str(thetaVal)];
    legendInfo{2*i+1} = ['MAP, \theta = ' num2str(thetaVal)];
    thetaVal = thetaVal/100;

end
legend(legendInfo,'location','best')

gamma = 1;
thetaVal = 1;
shades = [176 224 230;
                 206 135 235;
                 255 206 135;
                 126 10 238;
                 10 166 205];
       shades = 1/255*shades;

figure(2);
axes('fontsize',12);
hold on
plot(t,xtrue,'k-','LineWidth',1.5)
legendInfo{1} = ['true'];
```

```
for i = 1:5

    L = eye(n) - diag(ones(n-1,1),-1);
    theta = thetaVal*ones(n,1);
    theta(70) = 10;
    M = diag(1./sqrt(theta));
    L = M * L;

    % Calculating the MAP estimate and posterior variances by least squares
    xmean = [(1/sigma)*A;1/gamma*L]\[(1/sigma)*y;zeros(n,1)];
    Gamma_post = inv((1/sigma^2)*A'*A + 1/gamma^2*L'*L);

    % Plotting the MAP estimate and the 2*STD envelope

    % Defining different shades of blue for plotting

    STD = sqrt(diag(Gamma_post));
    xhigh = xmean + 2*STD;
    xlow = xmean - 2*STD;
    xlow = [0;xlow];
    xhigh = [0;xhigh];
    xmean = [0;xmean];
    h = fill([t;t(n+1:-1:1)],[xlow;xhigh(n+1:-1:1)],shades(i,:));
    set(h,'facealpha',.3)
    plot(t,xmean,'LineWidth',2)
    legendInfo{2*i}=['uncertainty, \gamma = ' num2str(gamma)];
    legendInfo{2*i+1} = ['MAP, \gamma = ' num2str(gamma)];
    gamma = gamma/100;

end
legend(legendInfo,'location','best')
```

Ex. 7.8:

Again, we use the same scheme as in Ex. 7.6 for deterministic inversion. The results are shown below for θ = 1e-08, □ = 1.



Code:

```
clear all

% explore the posterior with non-smooth prior and without hyper-parameters.

rand('state',20);
randn('state',18);

% Mesh
n = 100;
s = linspace(0,1,n+1)'; t = s(2:end);

% discretize the kernel
beta = 0.05;
a = 1/sqrt(2*pi*beta^2)*exp(-0.5*(1/beta^2)*t.^2);
A = 1/n*toeplitz(a);

% Truth
xtrue = zeros(n,1);
xtrue(70:end) = 10;
```

```
%%------------additive noise-----------
noise = 5;       % Noise level in percentages of the max. of noiseless signal
y0 = A*xtrue;    % Noiseless signal
sigma = max(abs(y0))*noise/100;                 % STD of the additive noise
y = y0 + sigma*randn(n,1);


%%------------Prior construction----------
% standard deviation of the innovation
gamma = 1;

L = eye(n) - diag(ones(n-1,1),-1);
theta = 1.e-8*ones(n,1);
theta(70) = 10;
M = diag(1./sqrt(theta));
L = M * L;

% Calculating the MAP estimate and posterior variances by least squares
xmean = [(1/sigma)*A;1/gamma*L]\[(1/sigma)*y;zeros(n,1)];
Gamma_post = inv((1/sigma^2)*A'*A + 1/gamma^2*L'*L);

% Plotting the MAP estimate and the 2*STD envelope

% Defining different shades of blue for plotting
 shades = [176 224 230;
             135 206 235;
             135 206 255;
             126 192 238;
             108 166 205];
   shades = 1/255*shades;

STD = sqrt(diag(Gamma_post));
xhigh = xmean + 2*STD;
xlow = xmean - 2*STD;

[xObsCG,idx] = CGLS(A,y,xtrue,sigma,n);

t = [0;t];
xlow = [0;xlow];
xhigh = [0;xhigh];
xmean = [0;xmean];
xtrue = [0;xtrue];
xObsCG = [0;xObsCG];


figure
axes('fontsize',12);
hold on
fill([t;t(n+1:-1:1)],[xlow;xhigh(n+1:-1:1)],shades(1,:))
plot(t,xmean,'r-','LineWidth',2)
plot(t,xtrue,'k-','LineWidth',1.5)
plot(t,xObsCG,'b-','LineWidth',2)
legend('uncertainty','MAP','truth','DI','location','best')

function [mObsCG,i] = CGLS(kernel,YObs,mActual,sigma,NObs)

mObsCG = zeros(size(mActual));
d = YObs;
r = kernel'*YObs;
p = r;
t = kernel*p;

for i = 1:10000
    if norm((kernel*mObsCG - YObs),2)^2 < sigma^2*NObs
        break;
    end
    alpha = norm(r,2)^2/norm(t,2)^2;
    mObsCG = mObsCG + alpha*p;
    d = d - alpha*t;
    rOld = r;
    r = kernel'*d;
    beta = norm(r,2)^2/norm(rOld,2)^2;
    p = r + beta*p;
    t = kernel*p;
end
```

Pi Estimation:

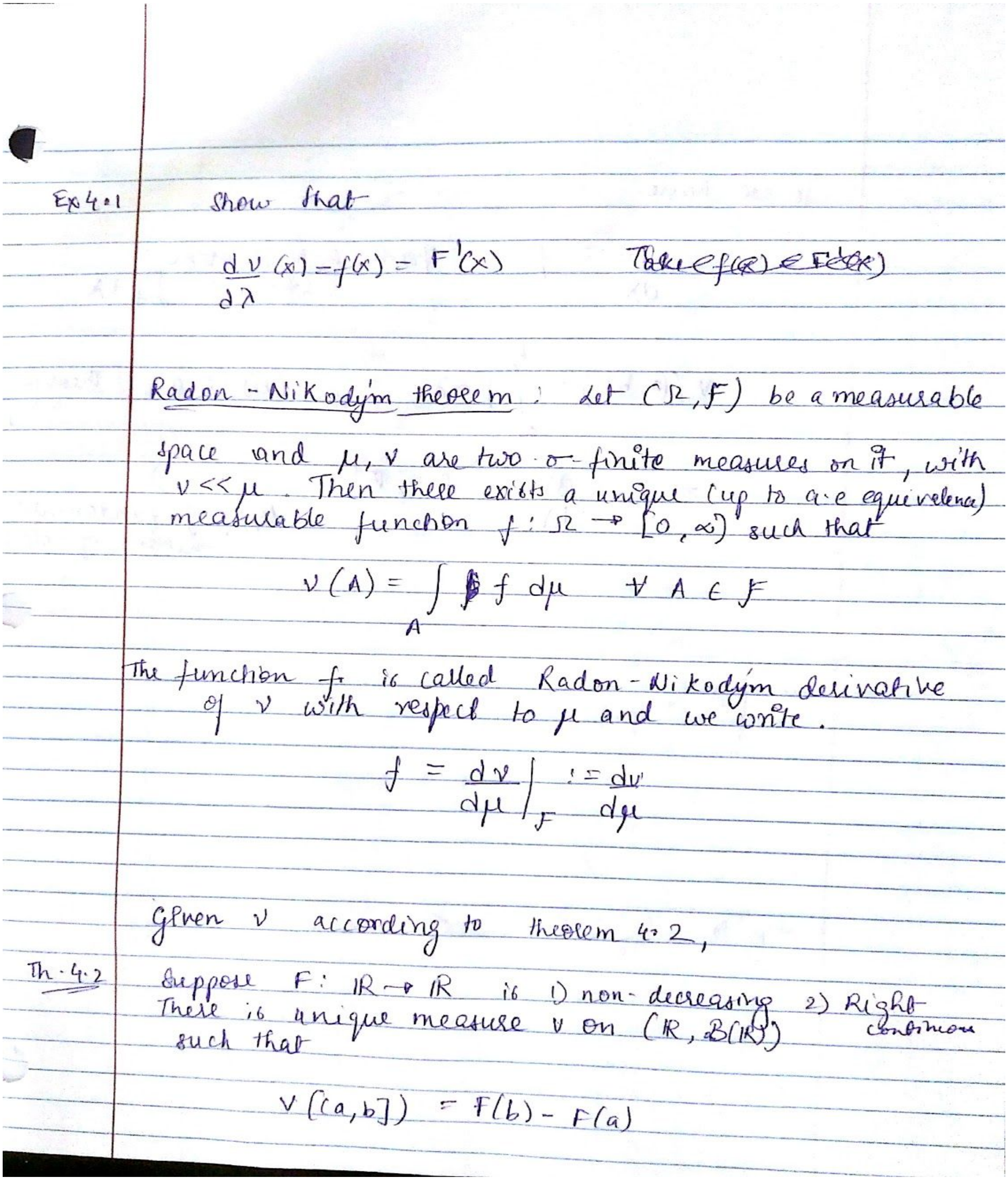
The code for estimating the value of Pi using a circle inscribed in a square is shown below, with results for six runs.

Code:

```
clear all

numPred = 100000;
piPred = 0;
count = 0;
for i = 1:numPred
    x = rand;
    y = rand;
    if (x^2 + y^2 - 1 <= 0)
        count = count + 1;
    end
end
piPred = 4*count/numPred;
disp(piPred)
```

```
>> pi_estimation
   3.14080

>> pi_estimation
   3.14828

>> pi_estimation
   3.14768

>> pi_estimation
   3.14748

>> pi_estimation
   3.13940

>> pi_estimation
   3.15384
```

Ex 4.1:



Ex 4.1 Show that

$$\frac{d\nu}{d\lambda}(x) = f(x) = F'(x) \qquad \text{Take } f(x) \in F'(x)$$

Radon - Nikodym theorem : Let $(\Omega, \mathcal{F})$ be a measurable space and μ, ν are two σ-finite measures on it, with $\nu << \mu$. Then there exists a unique (up to a.e equivalence) measurable function $f : \Omega \to [0, \infty)$ such that

$$\nu(A) = \int_A f \, d\mu \quad \forall A \in \mathcal{F}$$

The function f is called Radon - Nikodym derivative of ν with respect to μ and we write.

$$f = \frac{d\nu}{d\mu}\bigg|_{\mathcal{F}} := \frac{d\nu}{d\mu}$$

Given ν according to theorem 4.2,

Th. 4.2 Suppose $F : \mathbb{R} \to \mathbb{R}$ is 1) non-decreasing 2) Right continuous There is unique measure ν on $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$ such that

$$\nu((a, b]) = F(b) - F(a)$$

So we have

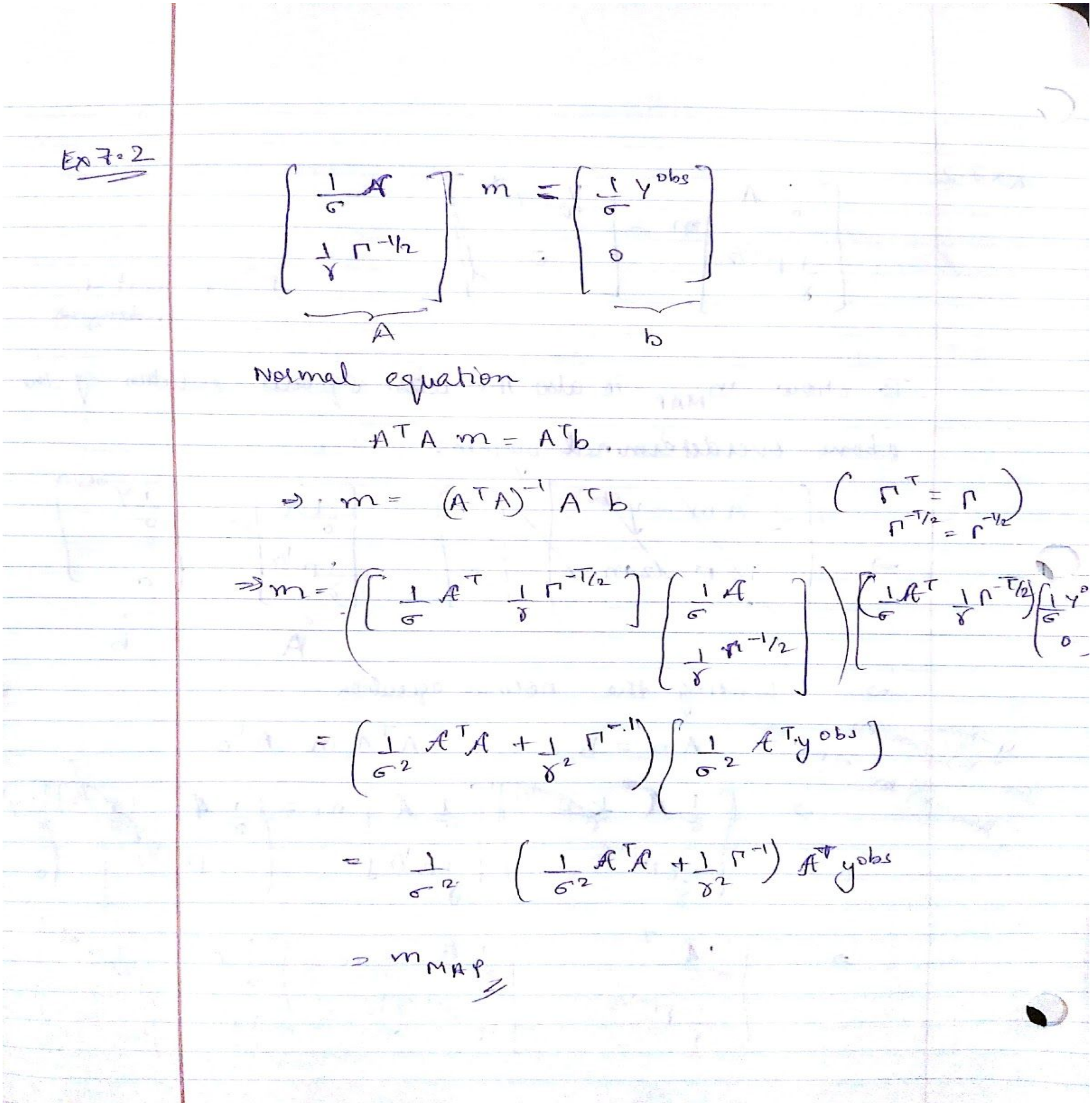
$$\frac{d\nu}{d\lambda} = f$$ from R.N Theorem where $\nu = \int f \, d\lambda$

$$\nu([a,b]) = \int_a^b f \, d\lambda = F(b) - F(a)$$ [Theorem 4.2]

$$\Rightarrow \quad \frac{d\nu}{d\lambda} = f = F'$$ (ODE)

Applying fundamental theorem of Calculus

Ex 7.2:



Ex 7.2

$$\underbrace{\begin{bmatrix} \frac{1}{\sigma} A \\ \frac{1}{\gamma} \Gamma^{-1/2} \end{bmatrix}}_{A} m = \underbrace{\begin{bmatrix} \frac{1}{\sigma} y^{obs} \\ 0 \end{bmatrix}}_{b}$$

Normal equation

$$A^T A m = A^T b.$$

$$\Rightarrow m = (A^T A)^{-1} A^T b \qquad \left(\begin{matrix} \Gamma^T = \Gamma \\ \Gamma^{-T/2} = \Gamma^{-1/2} \end{matrix} \right)$$

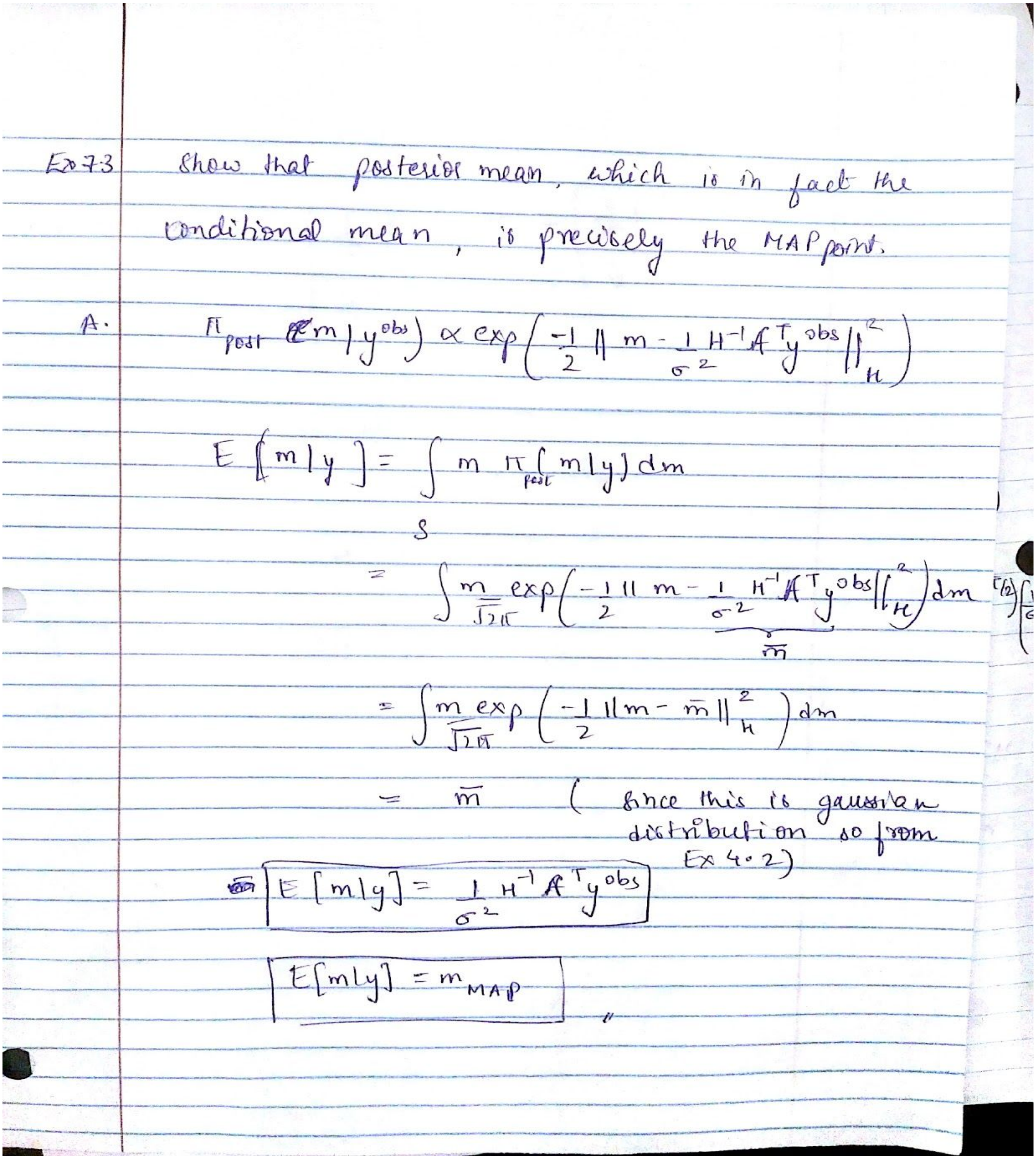
$$\Rightarrow m = \left(\begin{bmatrix} \frac{1}{\sigma} A^T & \frac{1}{\gamma} \Gamma^{-T/2} \end{bmatrix} \begin{bmatrix} \frac{1}{\sigma} A \\ \frac{1}{\gamma} \Gamma^{-1/2} \end{bmatrix} \right) \begin{bmatrix} \frac{1}{\sigma} A^T & \frac{1}{\gamma} \Gamma^{-T/2} \end{bmatrix} \begin{bmatrix} \frac{1}{\sigma} y^{o} \\ 0 \end{bmatrix}$$

$$= \left(\frac{1}{\sigma^2} A^T A + \frac{1}{\gamma^2} \Gamma^{-1} \right) \left(\frac{1}{\sigma^2} A^T y^{obs} \right)$$

$$= \frac{1}{\sigma^2} \left(\frac{1}{\sigma^2} A^T A + \frac{1}{\gamma^2} \Gamma^{-1} \right) A^T y^{obs}$$

$$= m_{MAP}$$

Ex 7.3:



Ex 7.3 Show that posterior mean, which is in fact the conditional mean, is precisely the MAP point.

A. $\pi_{post}(m|y^{obs}) \propto \exp\left(\frac{-1}{2}\left\| m - \frac{1}{\sigma^2} H^{-1} A^T y^{obs} \right\|_H^2\right)$

$$E[m|y] = \int_S m \, \pi_{post}(m|y)\, dm$$

$$= \int \frac{m}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left\| m - \underbrace{\frac{1}{\sigma^2} H^{-1} A^T y^{obs}}_{\bar{m}} \right\|_H^2\right) dm$$

$$= \int \frac{m}{\sqrt{2\pi}} \exp\left(\frac{-1}{2}\| m - \bar{m} \|_H^2\right) dm$$

$= \bar{m}$ (since this is gaussian distribution so from Ex 4.2)

$$\boxed{E[m|y] = \frac{1}{\sigma^2} H^{-1} A^T y^{obs}}$$

$$\boxed{E[m|y] = m_{MAP}}$$ //

Ex. 6.1 - 6.4:

To save space here are both modifications to the code from 6.1 and 6.4 together

6.1 Modify `BayesianPriorElicitation.m` to take in parameters for the more general scheme.

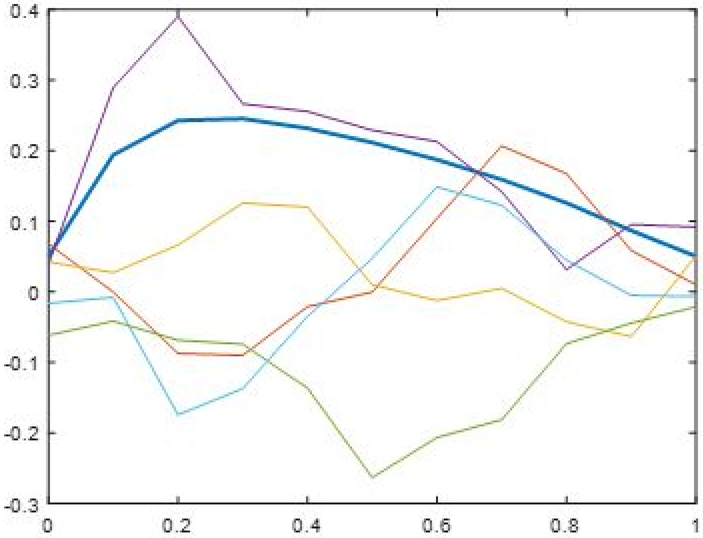
6.4 Modify `BayesianPriorElicitation.m` to take in parameters for multiple jumps

```
function BayesianPriorElicitation(lambda, t)
      rand('state',20);
      randn('state',18);
      gamma = 0.05;
      % Mesh
      n = 160;
      s = linspace(0,1,n+1)';
      % Construct the L_D matrix
      %% 6.1 Modification
      %Put lambdas in columnwise and then take the transpose to get the correct matrix
      L_D = diag(ones(n+1,1)) - lambda.*diag(ones(n,1),1) - (1-lambda).*diag(ones(n,1),-1)
      L_D = L_D'
      %Draw the Random Samples
      nv = 5;
      xn = randn(n+1,nv);
      %% 6.4 Modification
      %Scale by Theta to add jumps
      %(non-zero Dirichlet conditions can also be added and scaled through here)
      L_O = diag(t) * L_D;
      L_Oinv = inv(L_O);
      x = gamma * (L_Oinv * xn);
      Dev = sqrt(gamma^2 * diag(L_Oinv * L_Oinv'));
      figure
```

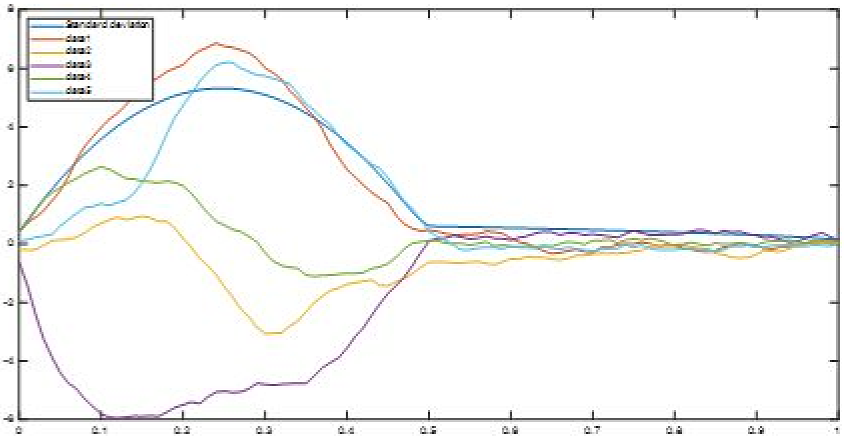
```
        axes('fontsize',12);

        plot(s,Dev,'linewidth',2)

        legend('Standard deviation','location','northwest')

        hold on

        plot(s,x)
```

6.1 Show new priors that were not presented in the text

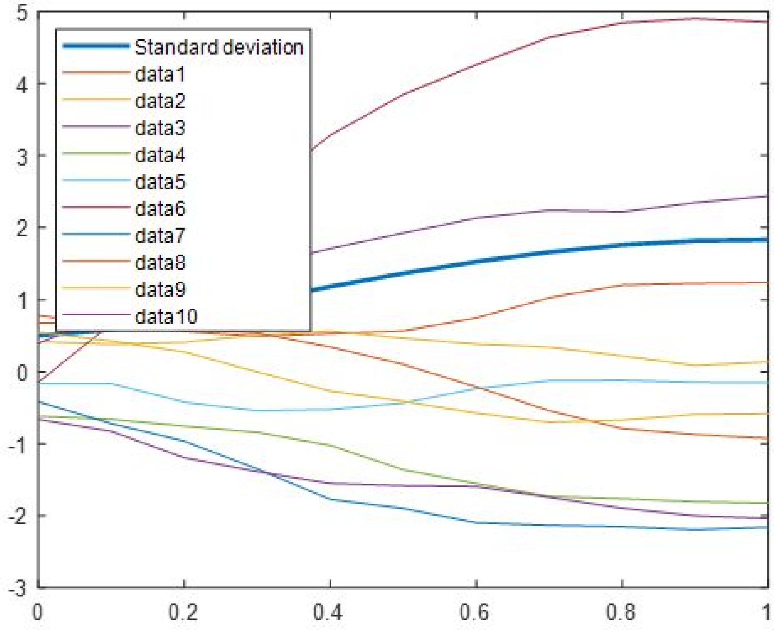
Heavily weighted to the left



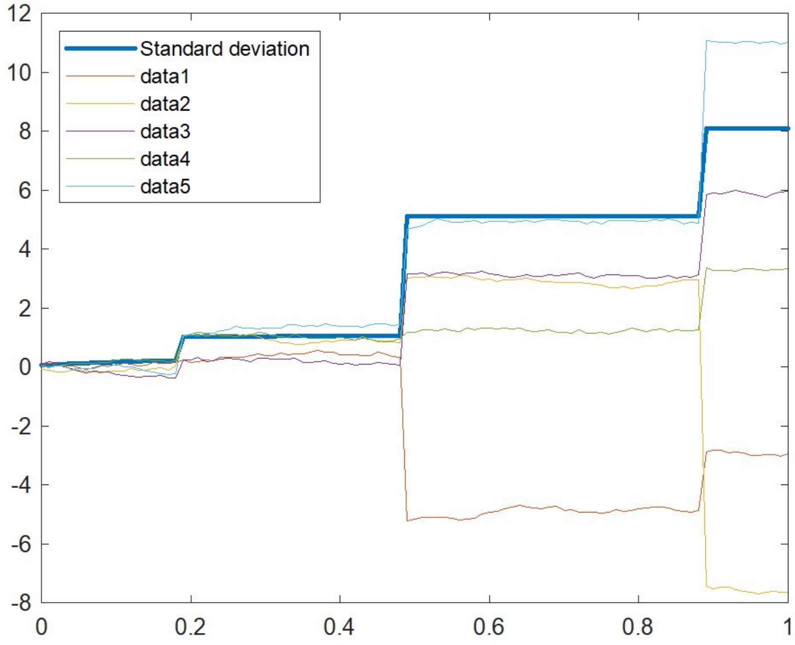
Changing Lambdas halfway through



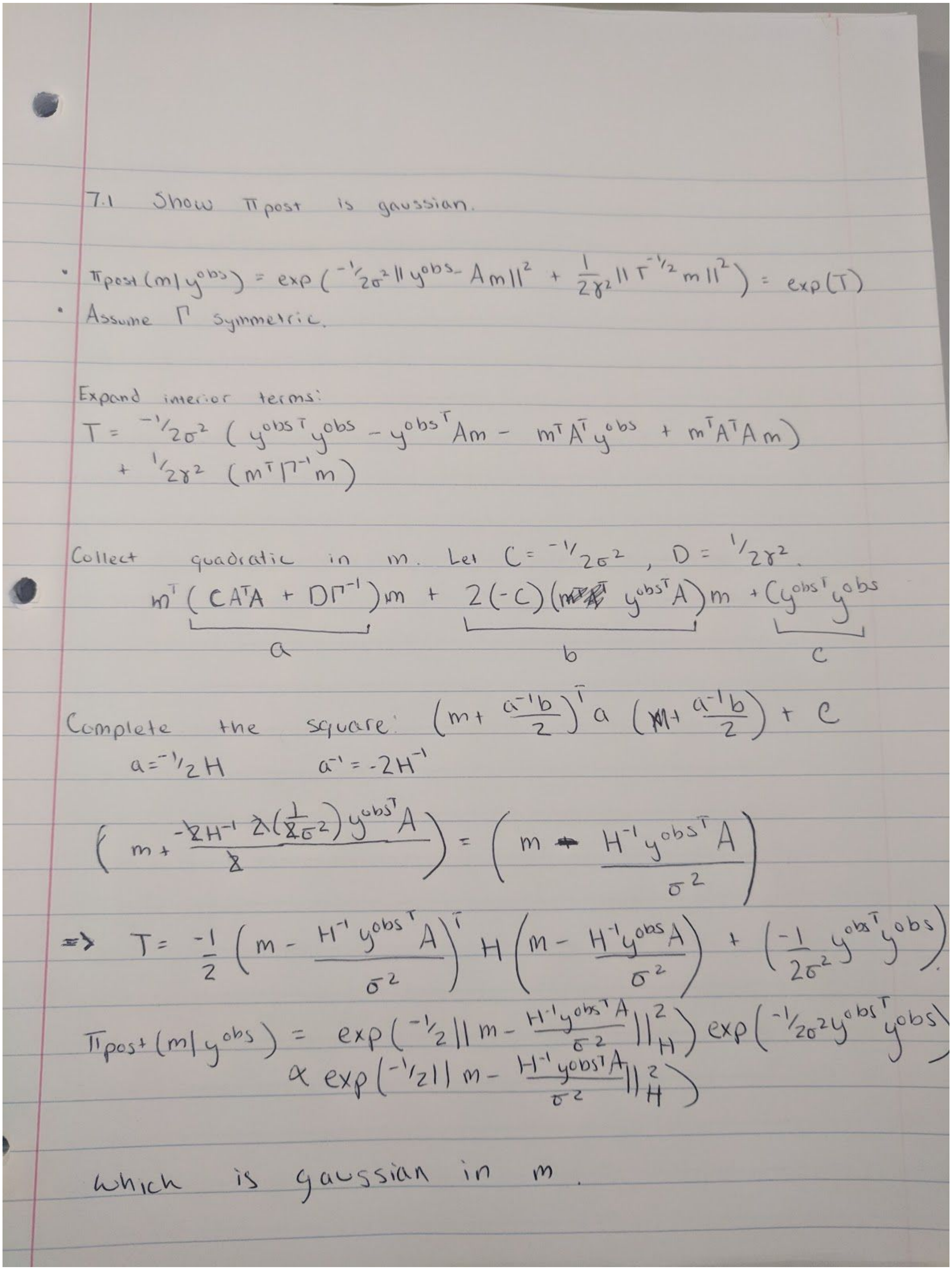
6.2 Construct a prior with a non-zero (uncertain) Dirichlet boundary condition at s=0 and a zero Neumann boundary condition at s=1.



6.3 Construct priors with 2 or more sudden jumps



Ex. 7.1:



7.1 Show π_{post} is gaussian.

- $\pi_{post}(m|y^{obs}) = \exp\left(\frac{-1}{2\sigma^2}\|y^{obs} - Am\|^2 + \frac{1}{2\gamma^2}\|\Gamma^{-1/2} m\|^2\right) = \exp(T)$
- Assume Γ symmetric.

Expand interior terms:

$$T = \frac{-1}{2\sigma^2}\left(y^{obs^T} y^{obs} - y^{obs^T} Am - m^T A^T y^{obs} + m^T A^T A m\right)$$
$$+ \frac{1}{2\gamma^2}\left(m^T \Gamma^{-1} m\right)$$

Collect quadratic in m. Let $C = \frac{-1}{2\sigma^2}$, $D = \frac{1}{2\gamma^2}$.

$$m^T \underbrace{\left(CA^TA + D\Gamma^{-1}\right)}_{a} m + \underbrace{2(-C)\left(\cancel{m^T A^T}\; y^{obs^T} A\right)}_{b} m + \underbrace{Cy^{obs^T} y^{obs}}_{c}$$

Complete the square: $\left(m + \frac{a^{-1}b}{2}\right)^T a \left(\cancel{m} + \frac{a^{-1}b}{2}\right) + c$

$a = -\frac{1}{2}H \qquad a^{-1} = -2H^{-1}$

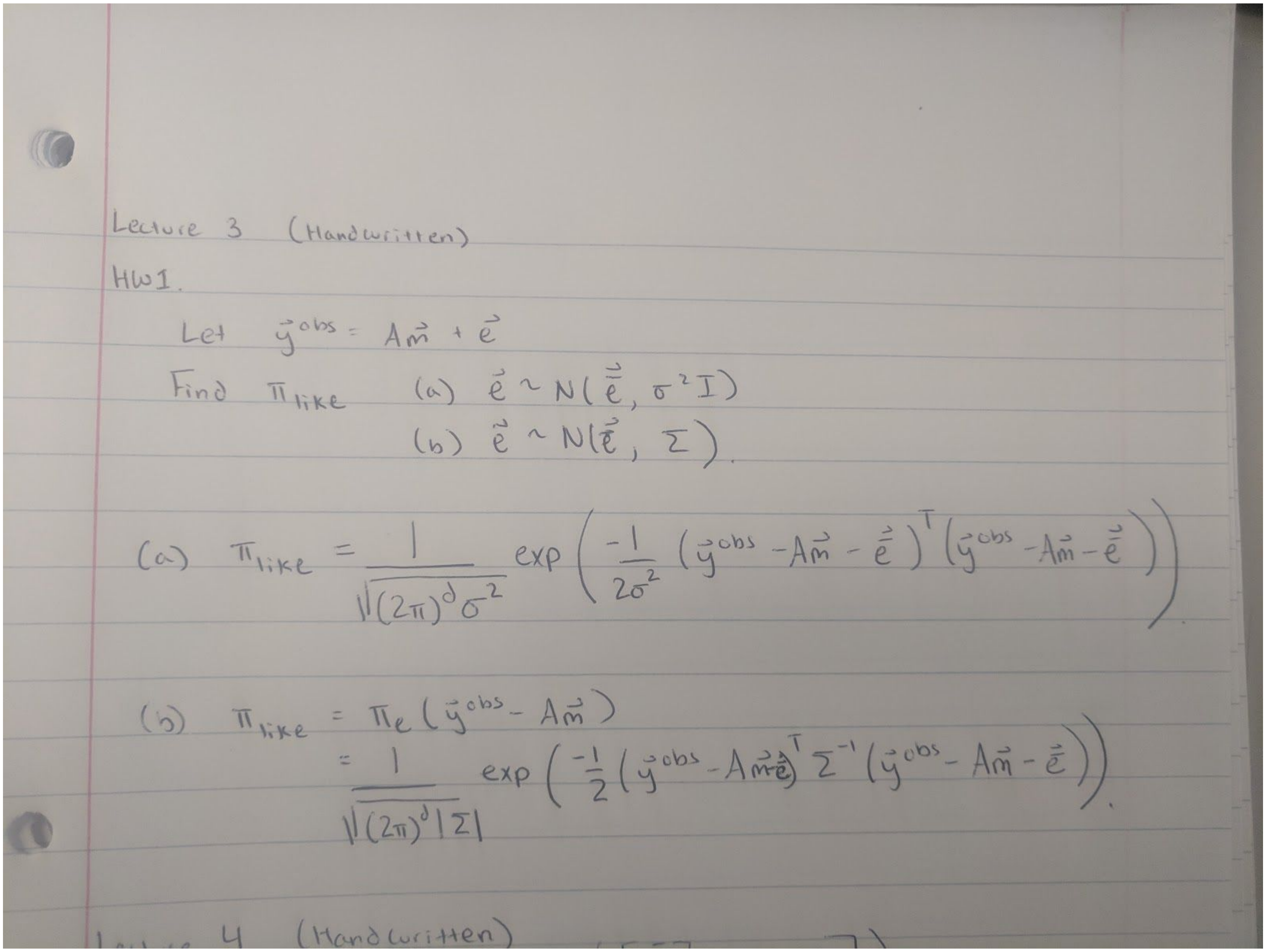
$$\left(m + \frac{-\cancel{2}H^{-1}\,\cancel{2}\left(\frac{1}{\cancel{2}\sigma^2}\right) y^{obs^T} A}{\cancel{2}}\right) = \left(m - \frac{H^{-1} y^{obs^T} A}{\sigma^2}\right)$$

$$\Rightarrow T = \frac{-1}{2}\left(m - \frac{H^{-1} y^{obs^T} A}{\sigma^2}\right)^T H \left(m - \frac{H^{-1} y^{obs} A}{\sigma^2}\right) + \left(\frac{-1}{2\sigma^2} y^{obs^T} y^{obs}\right)$$

$$\pi_{post}(m|y^{obs}) = \exp\left(-\tfrac{1}{2}\left\|m - \frac{H^{-1}y^{obs^T}A}{\sigma^2}\right\|_H^2\right)\exp\left(-\tfrac{1}{2\sigma^2} y^{obs^T} y^{obs}\right)$$
$$\propto \exp\left(-\tfrac{1}{2}\left\|m - \frac{H^{-1}y^{obs^T}A}{\sigma^2}\right\|_H^2\right)$$

which is gaussian in m.

Lecture 3 (Handwritten) Problem:



Lecture 3 (Handwritten)

HW1.

Let $\vec{y}^{obs} = A\vec{m} + \vec{e}$

Find π_{like} (a) $\vec{e} \sim N(\vec{\bar{e}}, \sigma^2 I)$

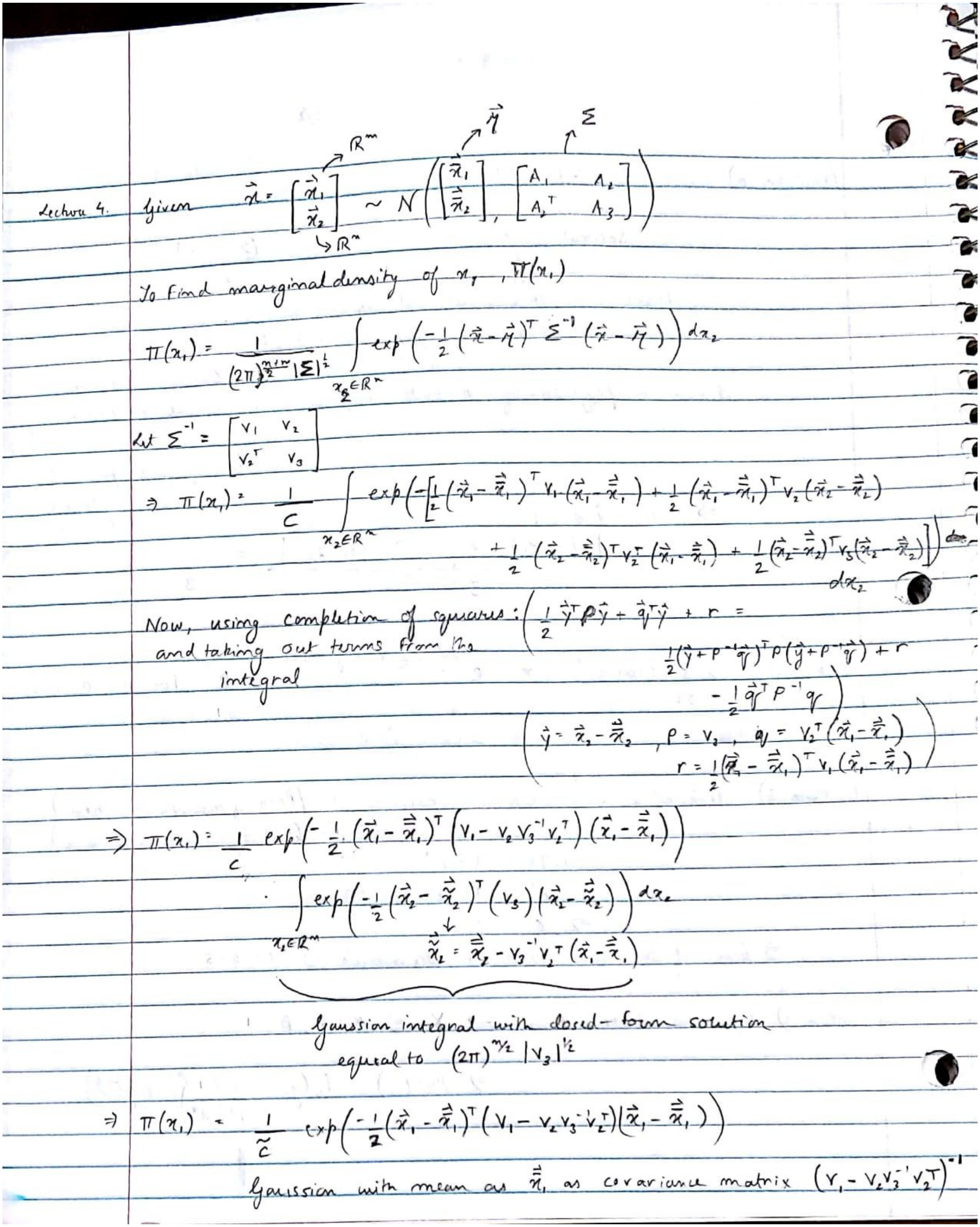
(b) $\vec{e} \sim N(\vec{\bar{e}}, \Sigma)$.

(a) $$\pi_{like} = \frac{1}{\sqrt{(2\pi)^d \sigma^2}} \exp\left(\frac{-1}{2\sigma^2} (\vec{y}^{obs} - A\vec{m} - \vec{\bar{e}})^T (\vec{y}^{obs} - A\vec{m} - \vec{\bar{e}})\right).$$

(b) $$\pi_{like} = \pi_e(\vec{y}^{obs} - A\vec{m})$$
$$= \frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left(\frac{-1}{2}(\vec{y}^{obs} - A\vec{m} - \vec{\bar{e}})^T \Sigma^{-1} (\vec{y}^{obs} - A\vec{m} - \vec{\bar{e}})\right).$$

Lecture 4 (Handwritten)

Lecture 4 (Handwritten) Problem:



Lecture 4.

given $\vec{x} = \begin{bmatrix} \overset{\mathbb{R}^m}{\vec{x}_1} \\ \underset{\mathbb{R}^n}{\vec{x}_2} \end{bmatrix} \sim N\left(\overbrace{\begin{bmatrix} \bar{\vec{x}}_1 \\ \bar{\vec{x}}_2 \end{bmatrix}}^{\vec{\eta}}, \overbrace{\begin{bmatrix} A_1 & A_2 \\ A_2^T & A_3 \end{bmatrix}}^{\Sigma} \right)$

To find marginal density of x_1, $\pi(x_1)$

$$\pi(x_1) = \frac{1}{(2\pi)^{\frac{m+n}{2}} |\Sigma|^{\frac{1}{2}}} \int_{x_2 \in \mathbb{R}^n} \exp\left(-\frac{1}{2} (\vec{x} - \vec{\eta})^T \Sigma^{-1} (\vec{x} - \vec{\eta}) \right) dx_2$$

Let $\Sigma^{-1} = \begin{bmatrix} V_1 & V_2 \\ V_2^T & V_3 \end{bmatrix}$

$$\Rightarrow \pi(x_1) = \frac{1}{C} \int_{x_2 \in \mathbb{R}^n} \exp\Big(-\Big[\frac{1}{2} (\vec{x}_1 - \bar{\vec{x}}_1)^T V_1 (\vec{x}_1 - \bar{\vec{x}}_1) + \frac{1}{2} (\vec{x}_1 - \bar{\vec{x}}_1)^T V_2 (\vec{x}_2 - \bar{\vec{x}}_2)$$
$$+ \frac{1}{2} (\vec{x}_2 - \bar{\vec{x}}_2)^T V_2^T (\vec{x}_1 - \bar{\vec{x}}_1) + \frac{1}{2} (\vec{x}_2 - \bar{\vec{x}}_2)^T V_3 (\vec{x}_2 - \bar{\vec{x}}_2) \Big] \Big) dx_2$$

Now, using completion of squares and taking out terms from the integral : $\left(\frac{1}{2} \vec{y}^T P \vec{y} + \vec{q}^T \vec{y} + r = \frac{1}{2} (\vec{y} + P^{-1} \vec{q})^T P (\vec{y} + P^{-1} \vec{q}) + r - \frac{1}{2} \vec{q}^T P^{-1} q \right)$

$\left(\vec{y} = \vec{x}_2 - \bar{\vec{x}}_2, \; P = V_3, \; q = V_2^T (\vec{x}_1 - \bar{\vec{x}}_1) \quad r = \frac{1}{2} (\vec{x}_1 - \bar{\vec{x}}_1)^T V_1 (\vec{x}_1 - \bar{\vec{x}}_1) \right)$

$$\Rightarrow \pi(x_1) = \frac{1}{c} \exp\left(-\frac{1}{2} (\vec{x}_1 - \bar{\vec{x}}_1)^T (V_1 - V_2 V_3^{-1} V_2^T) (\vec{x}_1 - \bar{\vec{x}}_1) \right)$$
$$\cdot \underbrace{\int_{x_2 \in \mathbb{R}^n} \exp\left(-\frac{1}{2} (\vec{x}_2 - \tilde{\vec{x}}_2)^T (V_3) (\vec{x}_2 - \tilde{\vec{x}}_2) \right) dx_2}_{\text{Gaussian integral with closed-form solution equal to } (2\pi)^{n/2} |V_3|^{1/2}}$$

$$\tilde{\vec{x}}_2 = \bar{\vec{x}}_2 - V_3^{-1} V_2^T (\vec{x}_1 - \bar{\vec{x}}_1)$$

$$\Rightarrow \pi(x_1) = \frac{1}{\tilde{c}} \exp\left(-\frac{1}{2} (\vec{x}_1 - \bar{\vec{x}}_1)^T (V_1 - V_2 V_3^{-1} V_2^T) (\vec{x}_1 - \bar{\vec{x}}_1) \right)$$

Gaussian with mean as $\bar{\vec{x}}_1$ as covariance matrix $(V_1 - V_2 V_3^{-1} V_2^T)^{-1}$

$$\text{But}\quad \Sigma^{-1} = \begin{bmatrix} A_1 & A_2 \\ A_2^T & A_3 \end{bmatrix} = \begin{bmatrix} \overset{\nearrow \text{ using woodbury identity}}{\left(V_1 - V_2 V_3^{-1} V_2^T\right)^{-1}} & - \\ - & \left(V_3 - V_2^T V_1^{-1} V_2\right)^{-1} \end{bmatrix}$$

$$\Rightarrow A_1 = \left(V_1 - V_2 V_3^{-1} V_2^T\right)^{-1}$$

$\Rightarrow$ $\pi(x_1)$ is a gaussian with mean $\vec{\bar{x}}_1$ and covariance matrix A_1

Similarly, $\pi(x_2)$ is a gaussian with mean $\vec{\bar{x}}_2$ and covarience matrix A_3