Student Info

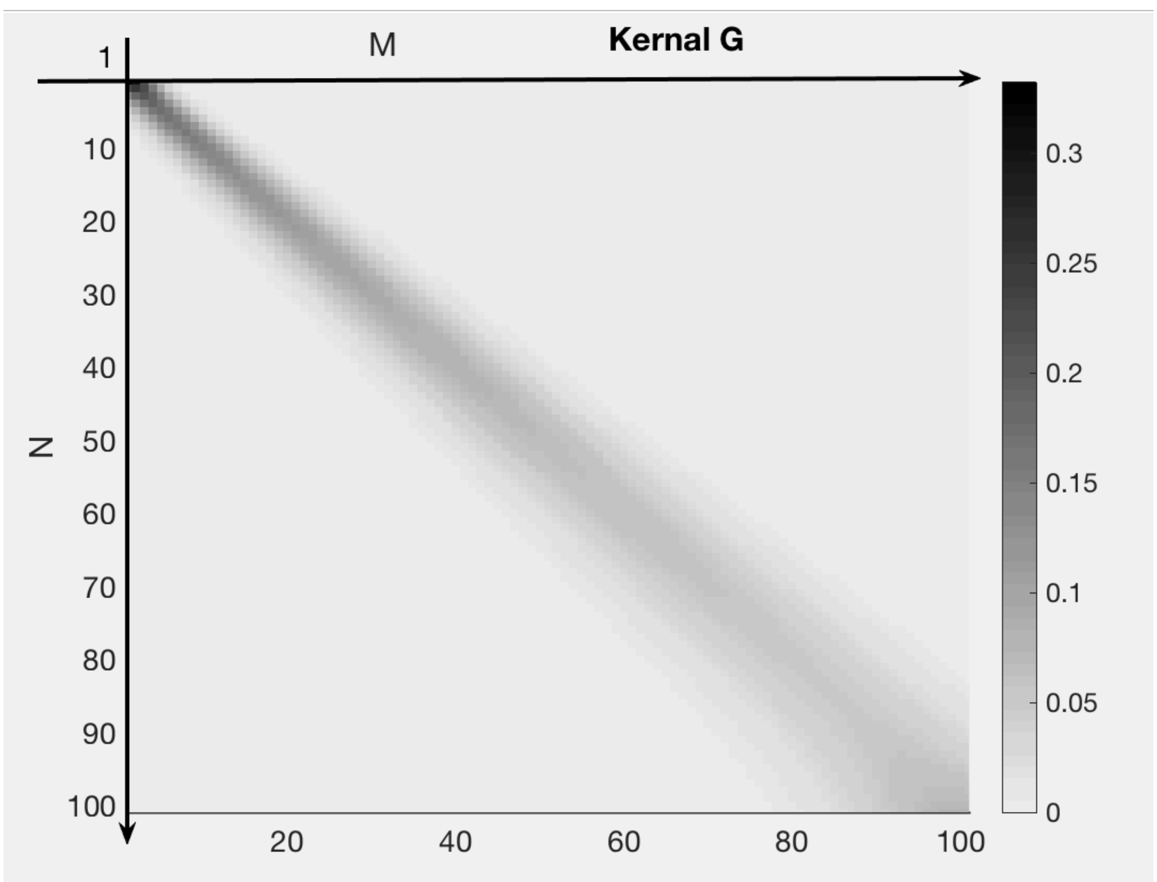
Jingyi Zhuang

jz2907@columbia.edu

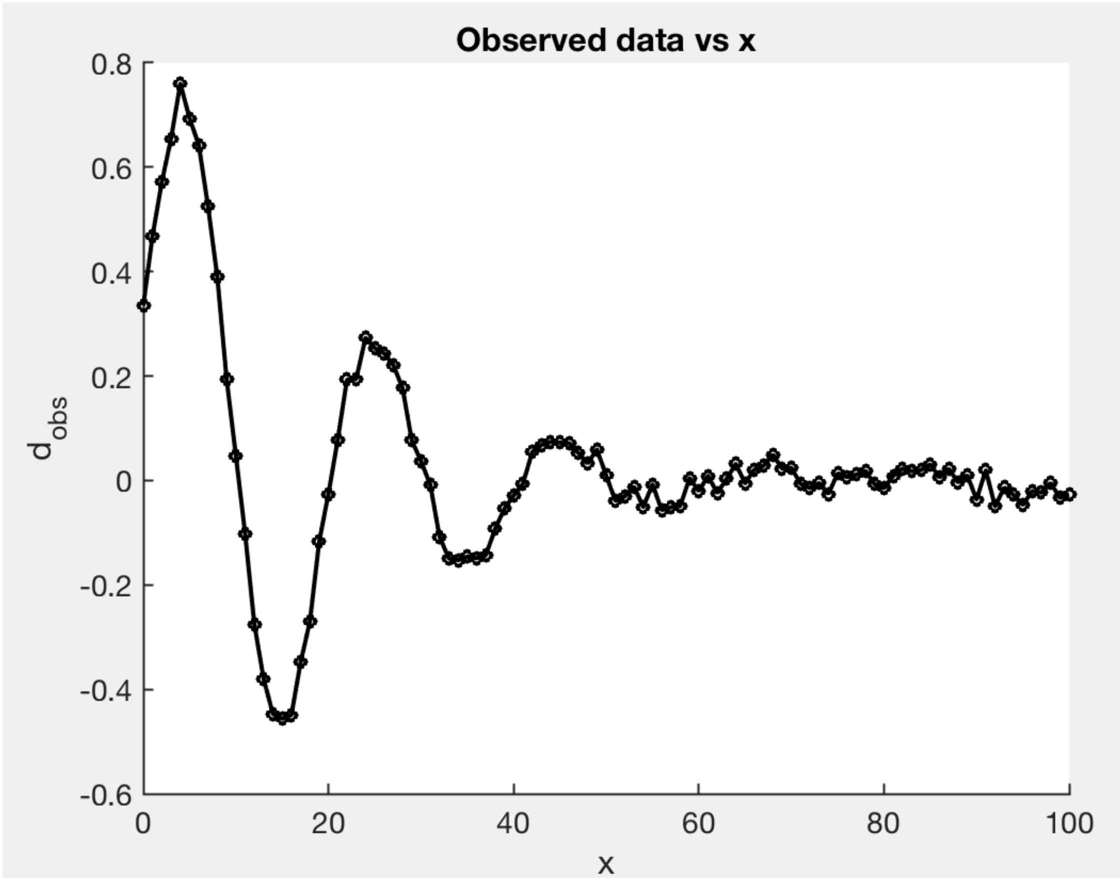
Homework 5

A.

Plot for data kernal G:



Plot for d_{obs}:



Commet:

The plot of kernal $\mathbf{G}$ shows that the data will be blurred by matrix $\mathbf{G}$ with moving the average, and blurred more when x increases.

The plot of d_{obs} shows that the data is more sin wave like at $x < 50$, with obvious fluctuation at certain amplitude. When $x > 50$, the wiggles occur due to the blur of data and noise.

Script:

```
% initiate variables

clearvars;
load('QMDA_HW_05.mat');

% A. plot data kernal G & observed data a a function of x
%    plot initialization
```

```
figure(1);
clf;
axis ij;
set(gca,'LineWidth',1,'FontSize',14);
colormap(flipud(gray)*0.93);
hold on;

%    plot kernal G
imagesc(G);
colorbar;

%    figure settings
title('Kernal G');
xlim([1 M]);
ylim([1 N]);
ylabel("N");
xlabel("M");

%    plot d obs
figure(2);
set(gca,'LineWidth',1,'FontSize',14);
hold on;

plot(x,dobs,"ko-",'LineWidth',2,'MarkerSize',5);

%    figure settings
title('Observed data vs x');
ylabel(sprintf('d_{obs}'));
xlabel("x");
```

B.

Output:

```
The standard deviation of the data error \sigma_e = 0.0141605
```

Script:

```
% B.

mest = (G'*G)\(G'*dobs);
e = dobs - G * mest;
E = e'*e;
sig2e = E/(N-1);
sige = sqrt(sig2e);

fprintf('the standard deviation of the data error \\sigma_e =
%g\n',sigd);
```

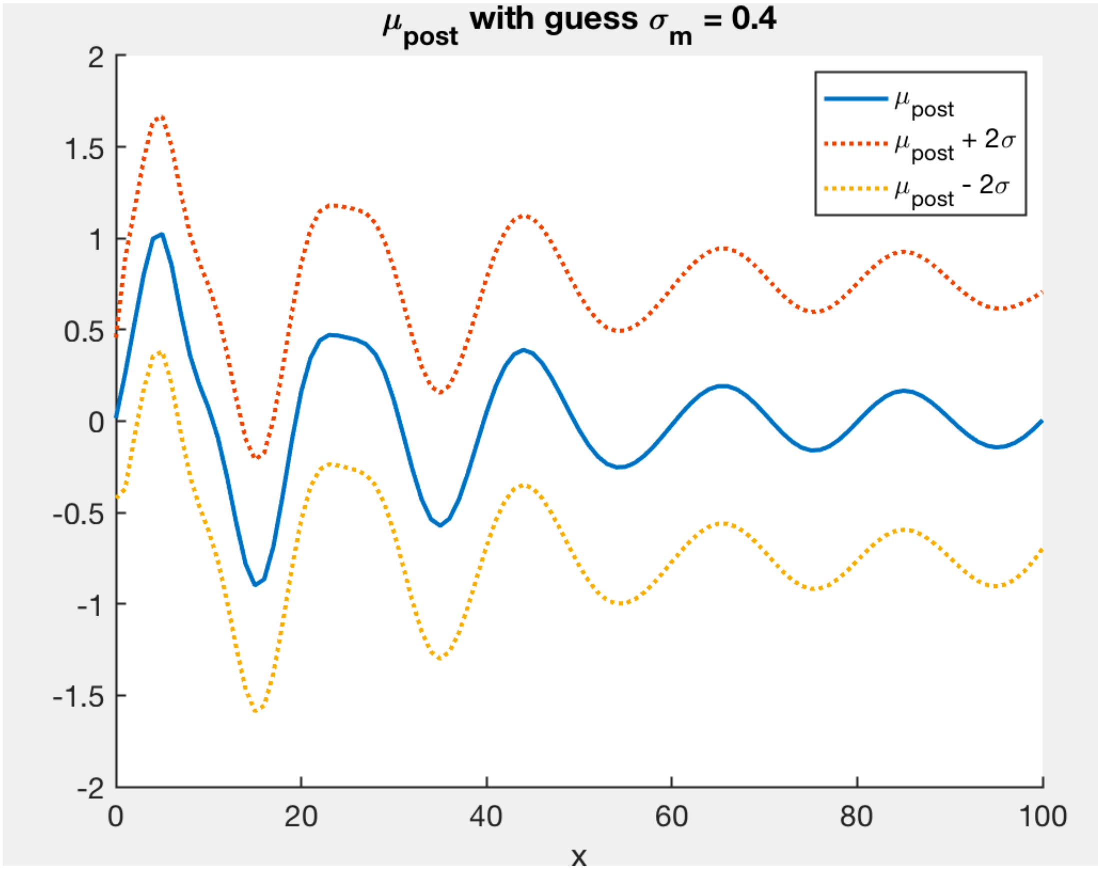
C.

In generalized least squares,

$$\mu_{\mathbf{post}} = \mu_{\mathbf{prior}} + (\mathbf{\Sigma}_{\mathbf{prior}}^{-1} + \mathbf{G}^T\mathbf{\Sigma}_{\mathbf{e}}^{-1}\mathbf{G})^{-1}\mathbf{G}^T\mathbf{\Sigma}_{\mathbf{e}}^{-1}(\mathbf{d}_{\mathbf{obs}} - \mathbf{G}\mu_{\mathbf{prior}}),$$
$$\mathbf{\Sigma}_{\mathbf{post}} = (\mathbf{\Sigma}_{\mathbf{prior}}^{-1} + \mathbf{G}^T\mathbf{\Sigma}_{\mathbf{e}}^{-1}\mathbf{G})^{-1}$$

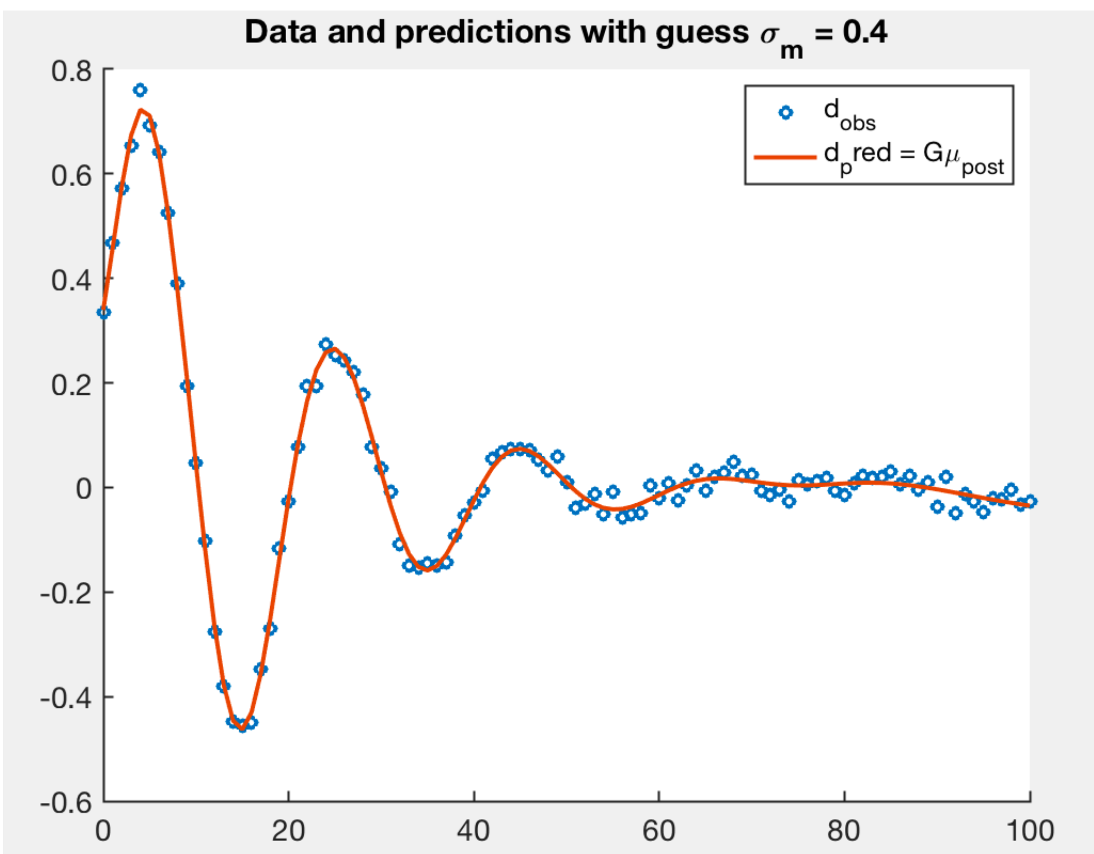
From the C. conditions we know $\mu_{\mathbf{prior}} = 0$.

C.1



μ_{post} with guess σ_m = 0.4
μ_{post}
μ_{post} + 2σ
μ_{post} - 2σ
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
20
40
60
80
100
x

C.2



Data and predictions with guess σ_m = 0.4
d_{obs}
$d_p$red = Gμ_{post}
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
0
20
40
60
80
100

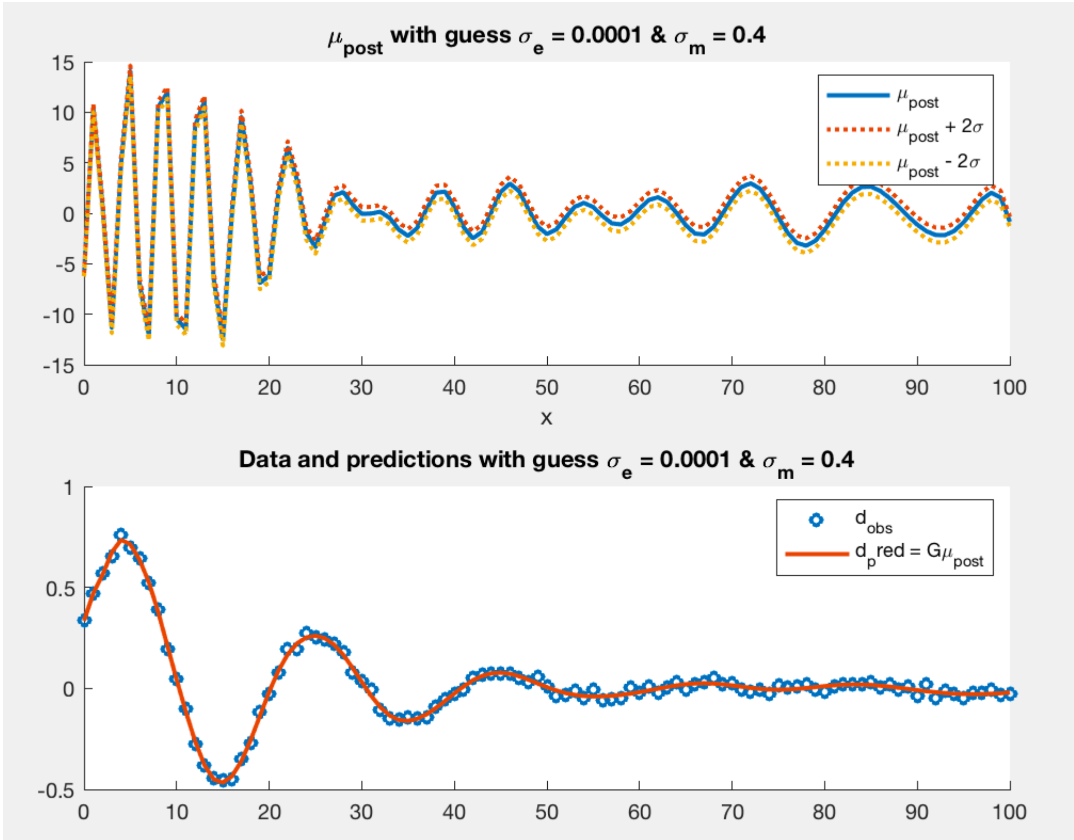
Script:

```
% C.

close all;

I = eye(N);
Cov_e = sig2_e * I;
Mu_prior = zeros(N,1);

% Guess

sig_m = 0.4;
Cov_prior = sig_m ^2 * I;

% Calculation

Mu_post = Mu_prior + inv (inv( Cov_prior) +G' * inv(Cov_e) *
G) * G' * inv(Cov_e) * (dobs - G * Mu_prior);
Cov_post = inv (inv( Cov_prior) + G' * inv(Cov_e) * G);
sig2_post = diag(Cov_post);
sig_post = sqrt(sig2_post);

% C.1 figure
figure(3);
set(gca,'LineWidth',1,'FontSize',14);
hold on;

plot(x,Mu_post,"-",'LineWidth',2,'MarkerSize',5);
plot(x,Mu_post + sig_post*2
,":",'LineWidth',2,'MarkerSize',5);
plot(x,Mu_post -
sig_post*2,":",'LineWidth',2,'MarkerSize',5);
```

```
legend({ sprintf('\\mu_{post}'),  sprintf('\\mu_{post} +
2\\sigma')  , sprintf('\\mu_{post} - 2\\sigma') });
xlabel('x');
title(sprintf('\\mu_{post} with guess \\sigma_m = %g',
sig_m));

% C.2 figure
figure(4);
set(gca,'LineWidth',1,'FontSize',14);
hold on;

plot(x,dobs,"o",'LineWidth',2,'MarkerSize',5);
plot(x,G*Mu_post, "-",'LineWidth',2 );

legend({ sprintf('d_{obs}'), sprintf('d_pred =
G\\mu_{post}')})
title(sprintf('Data and predictions with guess \\sigma_m =
%g', sig_m));
```

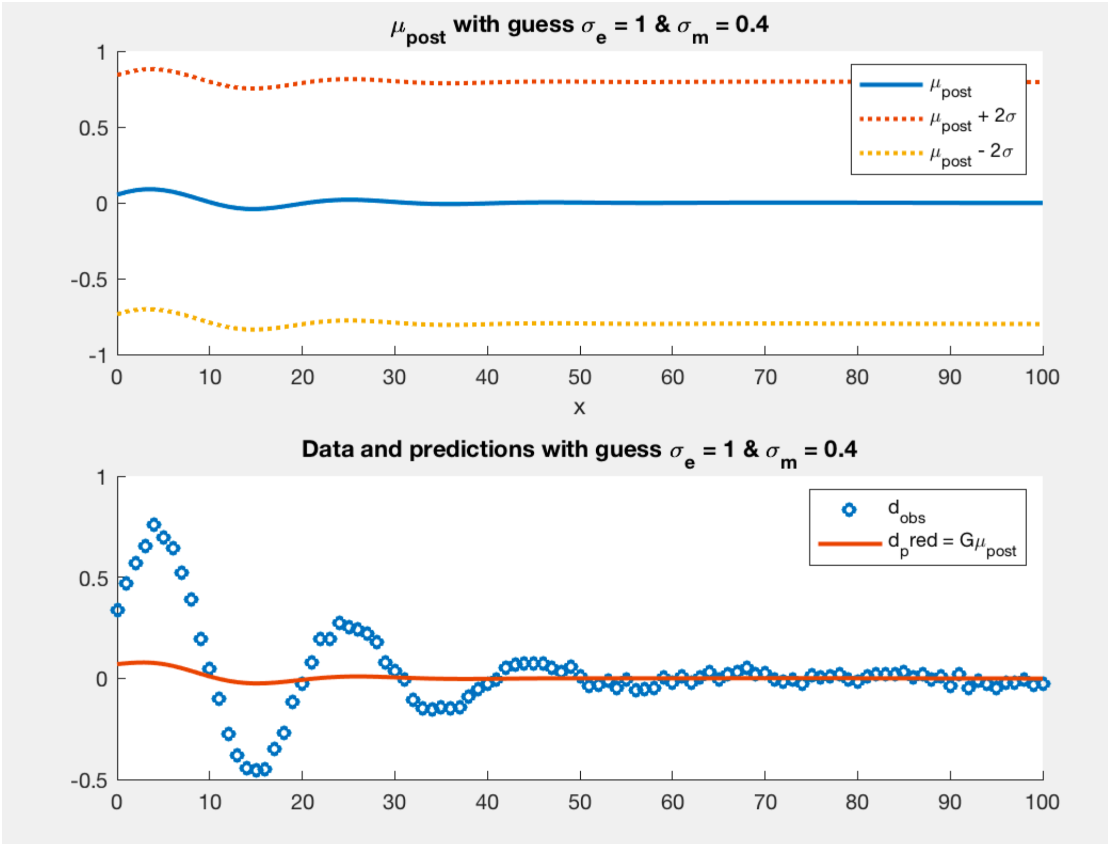
D.

σ_e becomes small:



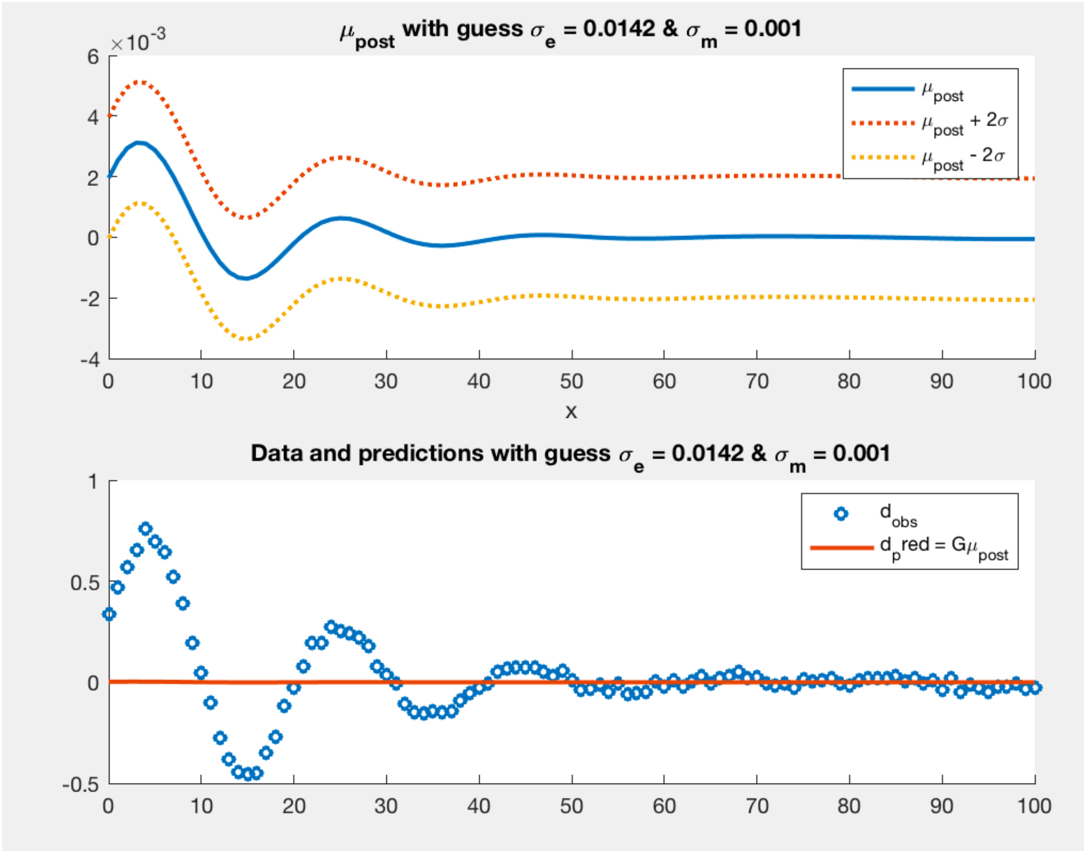
So from changing σ_e we can see: decreasing σ_e will make the fitting curve of prediction become more close to the observed data, but also makes the μ_{post}n have lots of big wiggle around $\mu_{prior} = 0$. Using a very small σ_e is just like ignoring the real noise and treating that noise as part of the true data, so there will appear incredible variation in μ_{post}.

σ_e becomes large:



Increasing σ_e will make the μ_{post} become very smooth, while the fitting curve becomes worse. That is because most of the fluctuations in the data are treated as noise, all of the infomation inside the data peaks are disgarded.

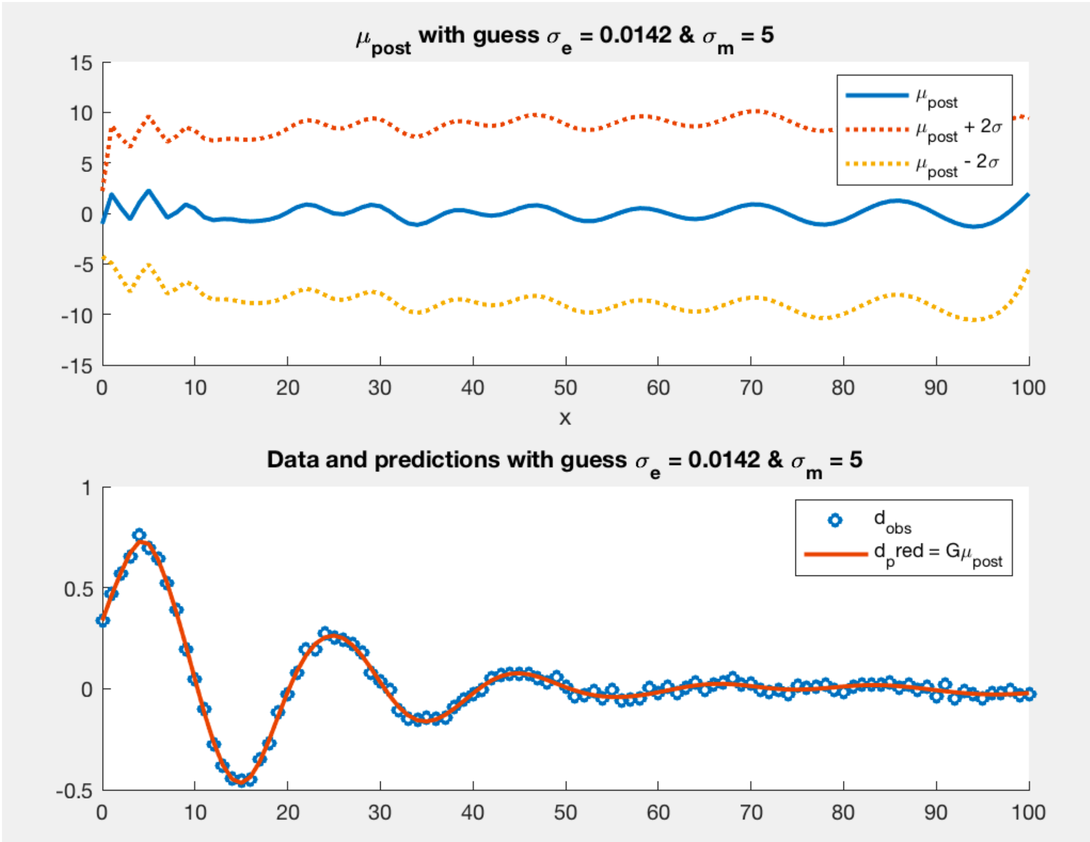
σ_m becomes small:



The predicted data is compressed to μ_{prior} as σ_m should describe the true data variation. Since σ_m has been set to be very small, the $\mathbf{m}$ is oversmoothed so the predicted data is not varying a lot.

This is a bad prediction.

σ_m becomes large:



When making σ_m large, we are allowing μ_{post} to vary a lot around $\mu_{posterior}$. It is a good fitted curve but the μ_{post} is not so reasonable, especially in $x > 50$, the noise are also amplified by have larger σ_m. Also wiggles are observed in the μ_{post} plot.

```
% initiate variables

clearvars;
load('QMDA_HW_05.mat');



close all;

% Guess

%%% INITIAL VALUES:

% sig_e ==> small
```

```
% sig_e = 0.0001;
% sig_m = 0.4;

% sig_e ==> large
% sig_e = 1;
% sig_m = 0.4;

% sig_m ==> small
% sig_e = 0.0142;
% sig_m = 0.001;

% sig_m ==> large
% sig_e = 0.0142;
% sig_m = 5;

% E. observe and get guess of sig_e, sig_m
sig_e = 0.1;
sig_m = 1.4;


% Calculation

I = eye(N);
sig2_e = sig_e ^2;
Cov_e = sig2_e * I;
Mu_prior = zeros(N,1);

Cov_prior = sig_m ^2 * I;

Mu_post = Mu_prior + inv (inv( Cov_prior) +G' * inv(Cov_e) *
G) * G' * inv(Cov_e) * (dobs - G * Mu_prior);
Cov_post = inv (inv( Cov_prior) + G' * inv(Cov_e) * G);
sig2_post = diag(Cov_post);
sig_post = sqrt(sig2_post);
```

```
% 1 figure
figure(1);
set(gca,'LineWidth',1,'FontSize',14);

subplot(2,1,1);
hold on;
plot(x,Mu_post,"-",'LineWidth',2,'MarkerSize',5);
plot(x,Mu_post + sig_post*2
,":",'LineWidth',2,'MarkerSize',5);
plot(x,Mu_post -
sig_post*2,":",'LineWidth',2,'MarkerSize',5);

legend({ sprintf('\\mu_{post}'),  sprintf('\\mu_{post} +
2\\sigma')  , sprintf('\\mu_{post} - 2\\sigma') });
xlabel('x');
title(sprintf('\\mu_{post} with guess \\sigma_e = %g &
\\sigma_m = %g', sig_e, sig_m));
hold off;

% 2 figure

subplot(2,1,2);
hold on;

plot(x,dobs,"o",'LineWidth',2,'MarkerSize',5);
plot(x,G*Mu_post, "-",'LineWidth',2 );

legend({ sprintf('d_{obs}'), sprintf('d_pred =
G\\mu_{post}')})
title(sprintf('Data and predictions with guess \\sigma_e = %g
& \\sigma_m = %g', sig_e, sig_m));
```

E.

From (D) I learned that the σ_e should be a proper value that describes how large the noise is in the dataset, if it's too small, all the noises will count into the prediction; if it's too large, all the data variation will be treated as noise and give bad predictions. And σ_m shows how the data varies within the model by controling the smoothness of $\mathbf{m}$. If it's too small, the predicted data is not varying; if it's too big, the noise will also be amplified.

Here I am choosing:

$\sigma_e = 0.1$, because I'm assuming the small variations around $x > 50$ is mainly due to noise and 0.1 seems to be a suitable value to represent the $x > 50$ region.

$\sigma_m = 1.4$ is determined due to the repeating tests.