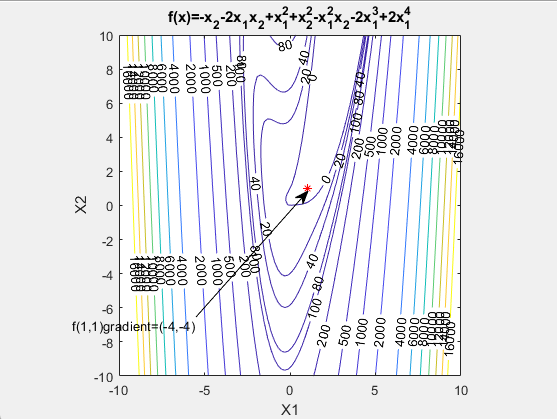
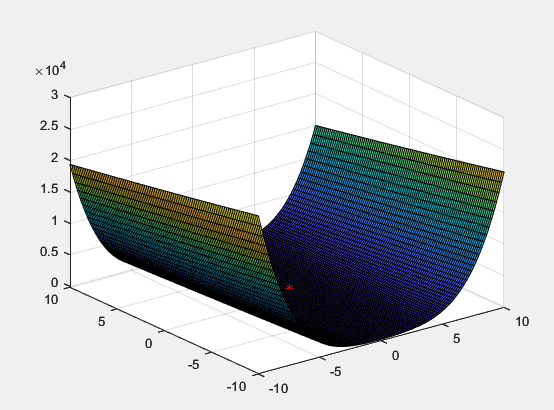
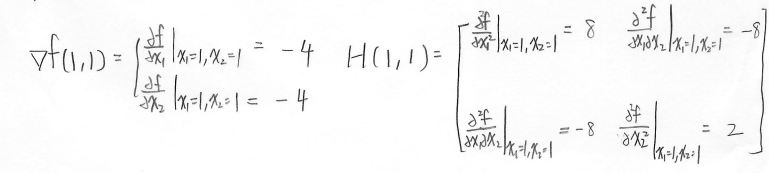
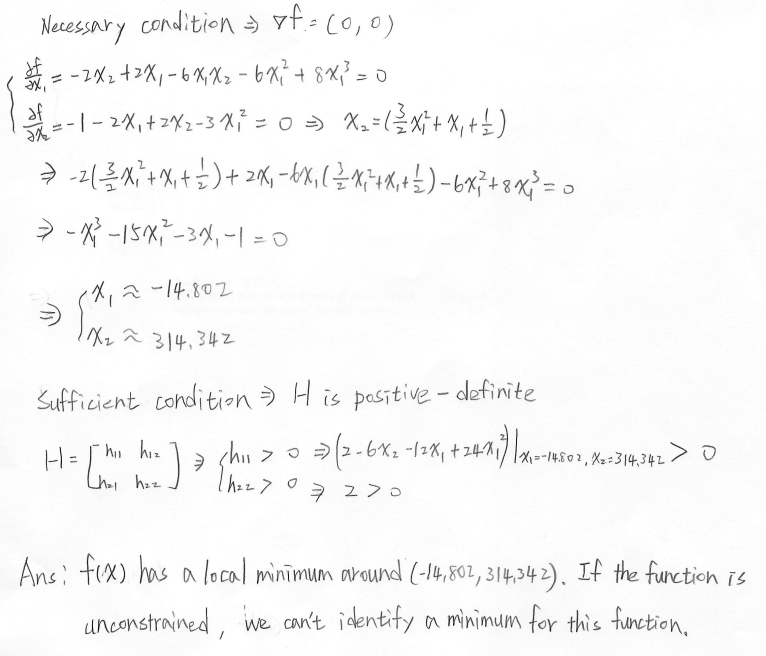
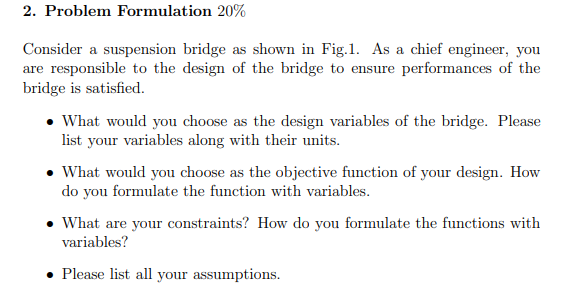
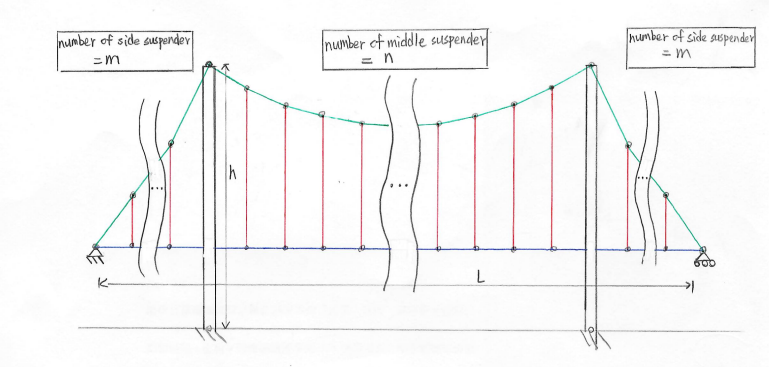
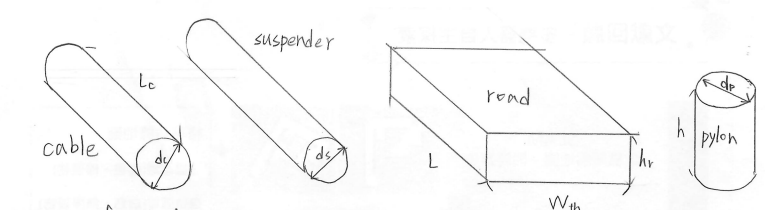
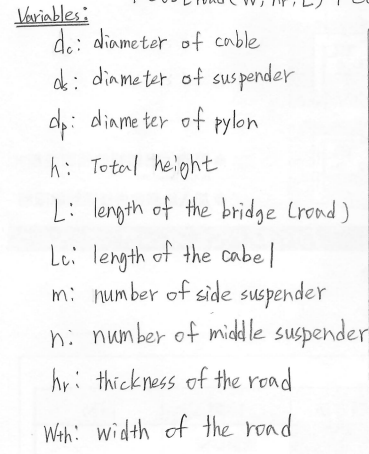
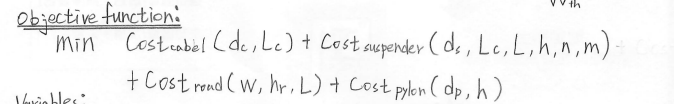
1-1

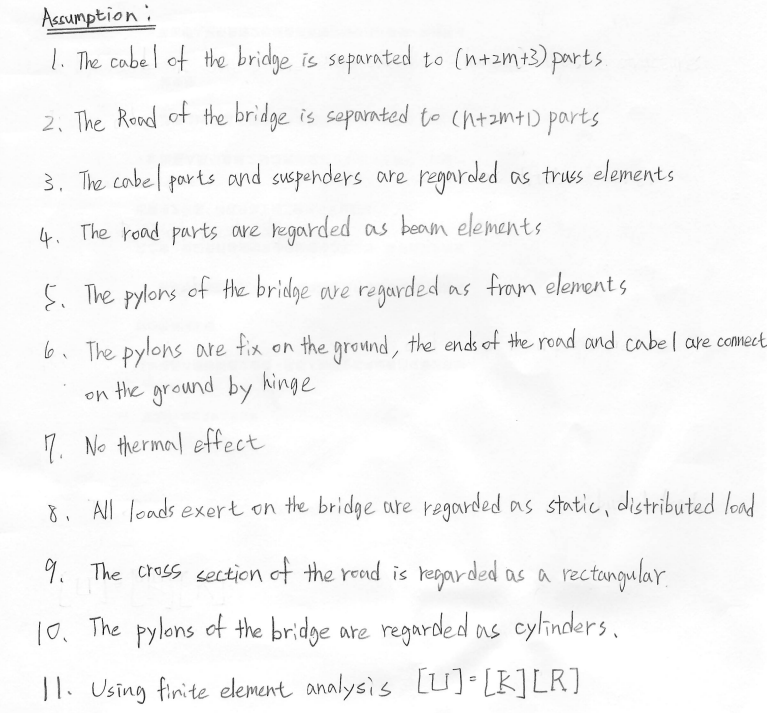
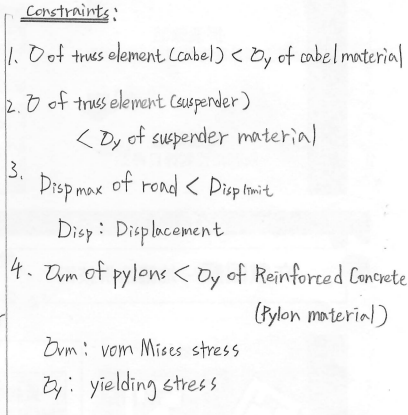
1-2

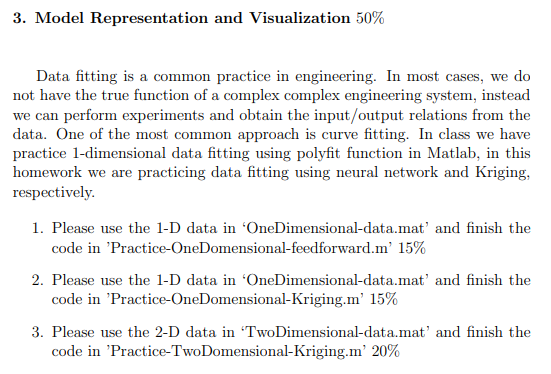


1-3









3-1

% feedforward net training and prediction demo for 1d problem

% NTU, ME, SOLab

% 2022/09/27

clc; clear; close all;

%% Step 0: Load data file

% x: 200 points between 0 and 2

% y: 200 points

load('OneDimensional\_data.mat');

x=x';

y=y';

%% Step 1: Polt the original data

figure(1);

% ----- to do -----

plot(x,y,'r.')

hold on

%% Step 2: Modeling through the all data.

% Construct a feedforward network with one hidden layer of size 10.

% ----- to do -----

net=feedforwardnet(10);

% Train the network net using the training data.

% Hint: Input will be a row vector. (1\*n matrix)

% ----- to do -----

net=train(net,x,y);

% Estimate the targets using the trained network.

% ----- to do -----

y\_net=net(x);

% Plot the estimation in the interval [0,2].

% ----- to do -----

plot(x,y\_net,'g')

hold off

%% Step 3: Estimate error (known model)

figure(2);

y\_origin = (1.7\*x.^5-6.2\*x.^4+6.3\*x.^3-2.3\*x+1.1);

% Estimate error

% ----- to do -----

err=abs(y\_net-y\_origin)./y\_origin\*100;

% Plot error with respect to x

% ----- to do -----

plot(x,err,'.')

%% Step 4: Estimate error (unknown model, leave one out)

% Leave one out: Take out the 1 sample, and model through the remaining n-1 data.

% Generate 200 models.

error=zeros(200,1);

for i = 1:size(y,2)

% Take out the ith sample.

% ----- to do -----

if i==1

y\_estimate=y(2:200);

x\_estimate=x(2:200);

elseif i==200

y\_estimate=y(1:199);

x\_estimate=x(1:199);

else

y\_estimate=[y(1:i-1) y(i+1:200)];

x\_estimate=[x(1:i-1) x(i+1:200)];

end

% Modeling through the remaining 199 data. (similar Step 2)

% ----- to do -----

net=feedforwardnet(10);

net=train(net,x\_estimate,y\_estimate);

y\_estimate\_net=net(x);

% Estimate error between model prediction and provided data

% ----- to do -----

error(i)=abs(y(i)-y\_estimate\_net(i))/y(i)\*100;

end

% Polt error with respect to each model

% ----- to do -----

figure(3)

plot(x,error,'b.')

figure(4)

histogram(error)

3-2

% kriging fitting and prediction demo for 1d problem

% NTU, ME, SOLab

% 2022/09/27

clc; clear; close all;

%% Step 0: Load data file

% x: 200 points between 0 and 2

% y: 200 points

load('OneDimensional\_data.mat');

lb = 0;

ub = 2;

%% Step 1: Polt the original data

figure(1);

% ----- to do -----

plot(x,y,'r.')

hold on

%% Step 2: Modeling through the all data.

% Fitting kriging

% Hint: parameter = f\_variogram\_fit(data x, data y, lb, ub);

% ----- to do -----

parameter=f\_variogram\_fit(x,y,lb,ub);

% Kriging prediction.

% Hint: Kriging prediction = f\_predictkrige(data x, parameter);

% ----- to do -----

[y\_krige,sigma]=f\_predictkrige(x, parameter);

% Plot the kriging average in the interval [0,2].

% ----- to do -----

plot(x,y\_krige,'g')

%% Step 3: Estimate error (known model)

% figure(2);

y\_origin = (1.7\*x.^5-6.2\*x.^4+6.3\*x.^3-2.3\*x+1.1);

plot(x,y\_origin,'b--')

legend('y data','y krige','y origin')

hold off

% Estimate error

% ----- to do -----

err=abs(y\_origin-y\_krige)./y\_origin\*100;

% Plot error with respect to x

% ----- to do -----

figure(2);

plot(x,err,'.')

%% Step 4: Estimate error (unknown model, leave one out)

% Leave one out: Take out the 1 sample, and model through the remaining n-1 data.

% Generate 200 models.

error=zeros(200,1);

for i = 1:size(y,1)

% Take out the ith sample.

% ----- to do -----

if i==1

y\_estimate=y(2:200);

x\_estimate=x(2:200);

elseif i==200

y\_estimate=y(1:199);

x\_estimate=x(1:199);

else

y\_estimate=[y(1:i-1) ; y(i+1:200)];

x\_estimate=[x(1:i-1) ; x(i+1:200)];

end

% Modeling through the remaining 199 data. (similar Step 2)

% ----- to do -----

parameter=f\_variogram\_fit(x\_estimate,y\_estimate,lb,ub);

[y\_est\_krige,sigma]=f\_predictkrige(x, parameter);

% Estimate error between model prediction and provided data

% ----- to do -----

error(i)=abs(y\_est\_krige(i)-y(i))/y(i)\*100;

end

% Polt error with respect to each model

% ----- to do -----

figure(3)

plot(x,error,'b.')

figure(4)

histogram(error)

3-3

% kriging fitting and prediction demo for 2d problem

% NTU, ME, SOLab

% 2022/09/27

clc; clear; close all;

%% Step 0: Load data file

% x1,x2: 21\*21 points between 0 and 2

% z: 21\*21 points

load('TwoDimensional\_data.mat');

lb = [0, 0];

ub = [2, 2];

% Reshape

x1\_flatten = reshape(x1,[21\*21 1]);

x2\_flatten = reshape(x2,[21\*21,1]);

z\_flatten = reshape(z,[21\*21,1]);

x\_data = [x1\_flatten, x2\_flatten];

z\_data = z\_flatten;

%% Step 1: Polt the original data

figure(1);

% Hint: plot3

% ----- to do -----

plot3(x1\_flatten,x2\_flatten,z\_data,'r.')

hold on

%% Step 2: Modeling through the all data.

% Fitting kriging

% Hint: parameter = f\_variogram\_fit(data x, data z, lb, ub);

% ----- to do -----

parameter = f\_variogram\_fit(x\_data, z\_data, lb, ub);

% Kriging prediction.

% Hint: Kriging prediction = f\_predictkrige(data x, parameter);

% ----- to do -----

[z\_krige,sigma]=f\_predictkrige(x\_data, parameter);

% Plot the kriging average in the interval [0,2].

% ----- to do -----

plot3(x1\_flatten,x2\_flatten,z\_krige,'g')

%% Step 3: Estimate error (known model)

% figure(2);

z\_origin = (x1\_flatten.^2-5\*x2\_flatten.^2+x1\_flatten.\*x2\_flatten-8\*x1\_flatten+9\*x2\_flatten-5);

plot3(x1\_flatten,x2\_flatten,z\_origin,'b--')

xlabel 'x1'

ylabel 'x2'

zlabel 'z'

hold off

% Estimate error

% ----- to do -----

err=abs((z\_origin-z\_krige)./z\_origin)\*100;

% Plot error with respect to x1 and x2

% ----- to do -----

figure(2);

plot3(x1\_flatten,x2\_flatten,err,'.')

%% Step 4: Estimate error (unknown model, leave one out)

% Leave one out: Take out the 1 sample, and model through the remaining n-1 data.

% Generate 21\*21 models.

error=zeros(21\*21,1);

for i = 1:size(z\_flatten,1)

% Take out the ith sample.

% ----- to do -----

if i==1

z\_estimate=z\_data(2:441,:);

x\_estimate=x\_data(2:441,:);

elseif i==441

z\_estimate=z\_data(1:440,:);

x\_estimate=x\_data(1:440,:);

else

z\_estimate=[z\_data(1:i-1,:) ; z\_data(i+1:441,:)];

x\_estimate=[x\_data(1:i-1,:) ; x\_data(i+1:441,:)];

end

% Modeling through the remaining 21\*21-1 data. (similar Step 2)

% ----- to do -----

parameter = f\_variogram\_fit(x\_estimate, z\_estimate, lb, ub);

[z\_est\_krige,sigma]=f\_predictkrige(x\_data, parameter);

% Estimate error between model prediction and provided data

% ----- to do -----

error(i)=abs((z\_est\_krige(i)-z\_data(i))/z\_data(i))\*100;

end

% Polt error with respect to each model

% ----- to do -----

figure(3)

plot3(x1\_flatten,x2\_flatten,error,'b.')

figure(4)

histogram(error)