1. Boundedness/Monotonicity/Optimality Behaviors 30%

Given the following function

$$f(x) = -x_2 - 2x_1x_2 + x_1^2 + x_2^2 - 3x_1^2x_2 - 2x_1^3 + 2x_1^4$$

1. Identify the monotonicity in various regions of x_1 and x_2.
2. Plot the iso-value contour lines of the function and examine its behavior around the point $(1,1)^T$
3. Write the optimality conditions. Based on these conditions, if this function is unconstrained, Can you identify a minimum for this function?

1-1

1.

$$f(x) = -x_2 - 2x_1x_2 + x_1^2 + x_2^2 - 3x_1^2x_2 - 2x_1^3 + 2x_1^4$$

$$\Rightarrow \nabla f = \begin{cases} \dfrac{\partial f}{\partial x_1} = -2x_2 + 2x_1 - 6x_1x_2 - 6x_1^2 + 8x_1^3 \\ \dfrac{\partial f}{\partial x_2} = -1 - 2x_1 + 2x_2 - 3x_1^2 \end{cases}$$

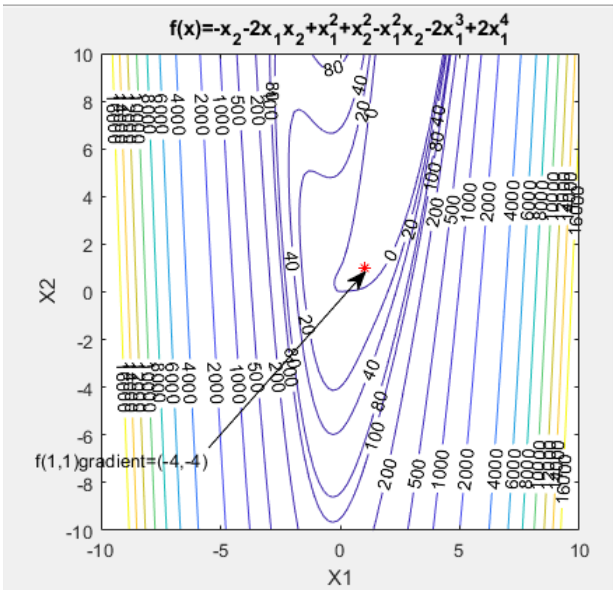
$$\Rightarrow H = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x_1^2} = 2 - 6x_2 - 12x_1 + 24x_1^2 & \dfrac{\partial^2 f}{\partial x_1 \partial x_2} = -2 - 6x_1 \\ \dfrac{\partial^2 f}{\partial x_2 \partial x_1} = -2 - 6x_1 & \dfrac{\partial^2 f}{\partial x_2^2} = 2 \end{bmatrix}$$

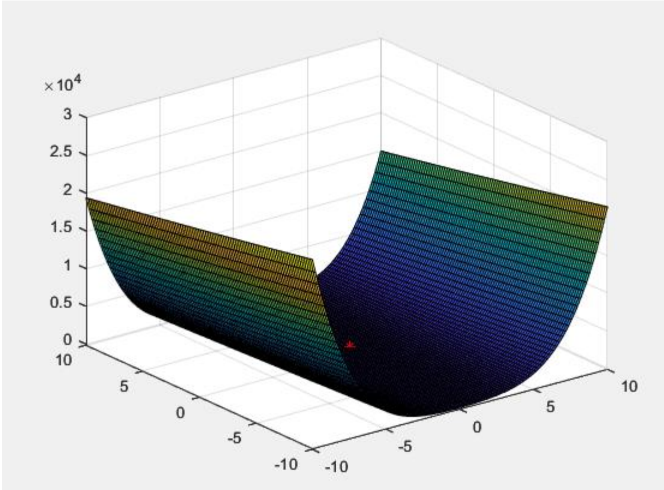
1-1

Monotanicity:

		x_1	x_2
$\dfrac{\partial f}{\partial x_1} > 0$	$\dfrac{\partial f}{\partial x_2} > 0 \Rightarrow$	+	+
	$\dfrac{\partial f}{\partial x_2} < 0 \Rightarrow$	+	−
$\dfrac{\partial f}{\partial x_1} < 0$	$\dfrac{\partial f}{\partial x_2} > 0 \Rightarrow$	−	+
	$\dfrac{\partial f}{\partial x_2} < 0 \Rightarrow$	−	−

1-2





$$\nabla f(1,1) = \begin{cases} \left.\dfrac{\partial f}{\partial x_1}\right|_{x_1=1,x_2=1} = -4 \\ \left.\dfrac{\partial f}{\partial x_2}\right|_{x_1=1,x_2=1} = -4 \end{cases} \quad H(1,1) = \begin{bmatrix} \left.\dfrac{\partial^2 f}{\partial x_1^2}\right|_{x_1=1,x_2=1} = 8 & \left.\dfrac{\partial^2 f}{\partial x_1 \partial x_2}\right|_{x_1=1,x_2=1} = -8 \\ \left.\dfrac{\partial^2 f}{\partial x_1 \partial x_2}\right|_{x_1=1,x_2=1} = -8 & \left.\dfrac{\partial f}{\partial x_2^2}\right|_{x_1=1,x_2=1} = 2 \end{bmatrix}$$

1-3

Necessary condition $\Rightarrow \nabla f = (0,0)$

$$\begin{cases} \dfrac{\partial f}{\partial x_1} = -2x_2 + 2x_1 - 6x_1x_2 - 6x_1^2 + 8x_1^3 = 0 \\ \dfrac{\partial f}{\partial x_2} = -1 - 2x_1 + 2x_2 - 3x_1^2 = 0 \Rightarrow x_2 = \left(\frac{3}{2}x_1^2 + x_1 + \frac{1}{2}\right) \end{cases}$$

$$\Rightarrow -2\left(\frac{3}{2}x_1^2 + x_1 + \frac{1}{2}\right) + 2x_1 - 6x_1\left(\frac{3}{2}x_1^2 + x_1 + \frac{1}{2}\right) - 6x_1^2 + 8x_1^3 = 0$$

$$\Rightarrow -x_1^3 - 15x_1^2 - 3x_1 - 1 = 0$$

$$\Rightarrow \begin{cases} x_1 \approx -14.802 \\ x_2 \approx 314.342 \end{cases}$$

Sufficient condition $\Rightarrow H$ is positive-definite

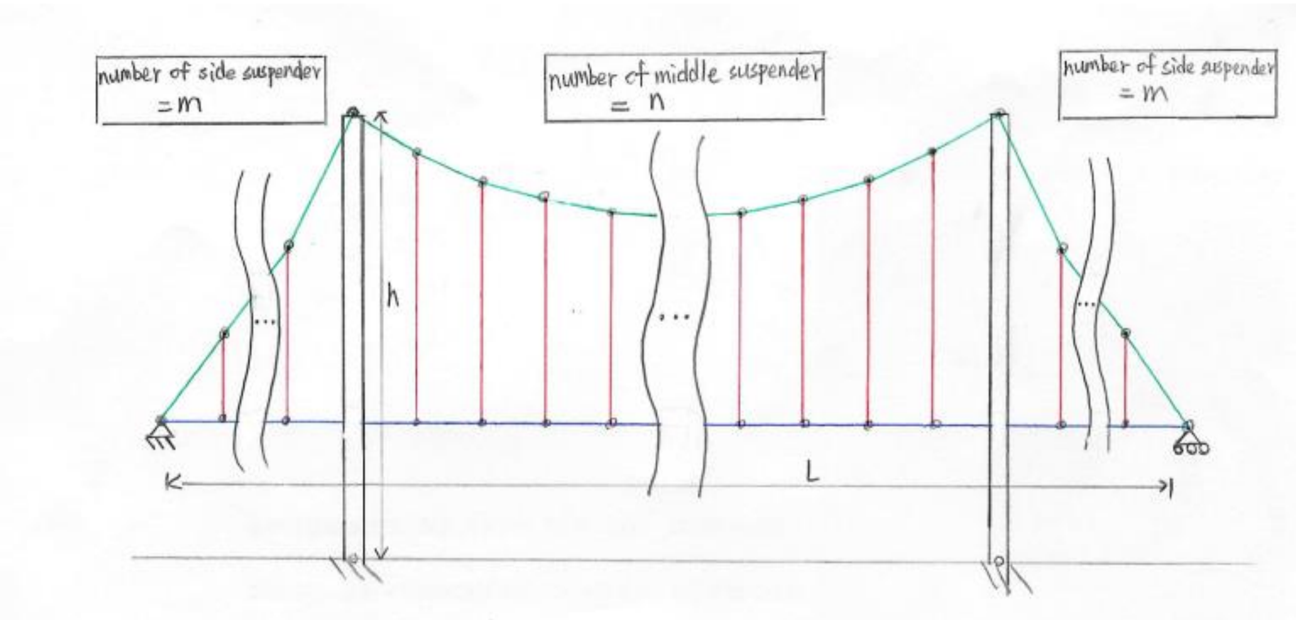
$$H = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \Rightarrow \begin{cases} h_{11} > 0 \Rightarrow \left.\left(2 - 6x_2 - 12x_1 + 24x_1^2\right)\right|_{x_1=-14.802,\, x_2=314.342} > 0 \\ h_{22} > 0 \Rightarrow 2 > 0 \end{cases}$$

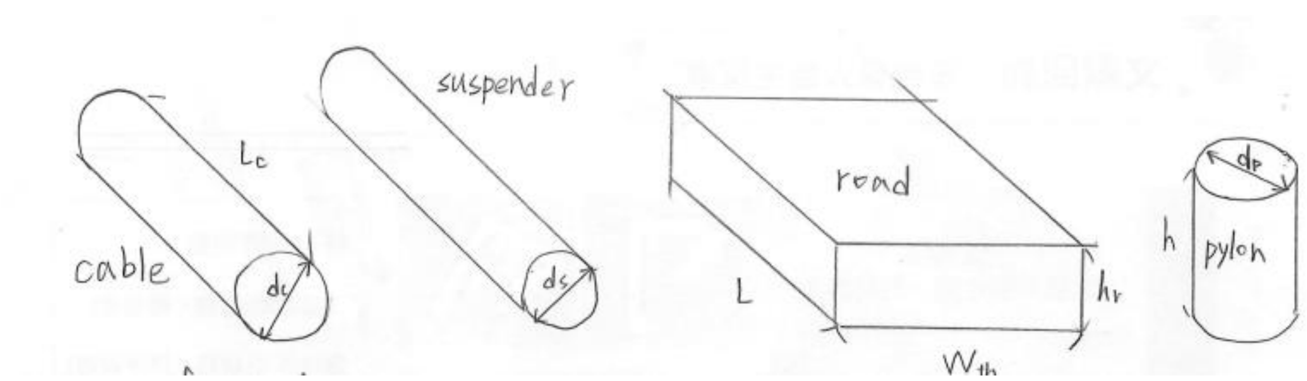
Ans: $f(x)$ has a local minimum around $(-14.802, 314.342)$. If the function is unconstrained, we can't identify a minimum for this function.

2. Problem Formulation 20%

Consider a suspension bridge as shown in Fig.1. As a chief engineer, you are responsible to the design of the bridge to ensure performances of the bridge is satisfied.

- What would you choose as the design variables of the bridge. Please list your variables along with their units.
- What would you choose as the objective function of your design. How do you formulate the function with variables.
- What are your constraints? How do you formulate the functions with variables?
- Please list all your assumptions.





Variables:

d_c: diameter of cable

d_s: diameter of suspender

d_p: diameter of pylon

h: Total height

L: length of the bridge (road)

L_c: length of the cabel

m: number of side suspender

n: number of middle suspender

h_r: thickness of the road

W_{th}: width of the road

Objective function:

$$\min \quad Cost_{cabel}(d_c, L_c) + Cost_{suspender}(d_s, L_c, L, h, n, m) + Cost_{road}(W, h_r, L) + Cost_{pylon}(d_p, h)$$

Constraints:

1. σ of truss element (cabel) < σ_y of cabel material

2. σ of truss element (suspender)

   < σ_y of suspender material

3. $Disp_{max}$ of road < $Disp_{limit}$

   Disp: Displacement

4. σ_{vm} of pylons < σ_y of Reinforced Concrete (Pylon material)

   σ_{vm}: vom Mises stress

   σ_y: yielding stress

Assumption:

1. The cabel of the bridge is separated to $(n+2m+3)$ parts
2. The Road of the bridge is separated to $(n+2m+1)$ parts
3. The cabel parts and suspenders are regarded as truss elements
4. The road parts are regarded as beam elements
5. The pylons of the bridge are regarded as fram elements
6. The pylons are fix on the ground, the ends of the road and cabel are connect on the ground by hinge
7. No thermal effect
8. All loads exert on the bridge are regarded as static, distributed load
9. The cross section of the road is regarded as a rectangular
10. The pylons of the bridge are regarded as cylinders.
11. Using finite element analysis $[U]=[K][R]$

3. Model Representation and Visualization 50%

Data fitting is a common practice in engineering. In most cases, we do not have the true function of a complex complex engineering system, instead we can perform experiments and obtain the input/output relations from the data. One of the most common approach is curve fitting. In class we have practice 1-dimensional data fitting using polyfit function in Matlab, in this homework we are practicing data fitting using neural network and Kriging, respectively.

1. Please use the 1-D data in 'OneDimensional-data.mat' and finish the code in 'Practice-OneDomensional-feedforward.m' 15%
2. Please use the 1-D data in 'OneDimensional-data.mat' and finish the code in 'Practice-OneDomensional-Kriging.m' 15%
3. Please use the 2-D data in 'TwoDimensional-data.mat' and finish the code in 'Practice-TwoDomensional-Kriging.m' 20%

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)
```