Assignment 8: Integration, Polynomials and Functions

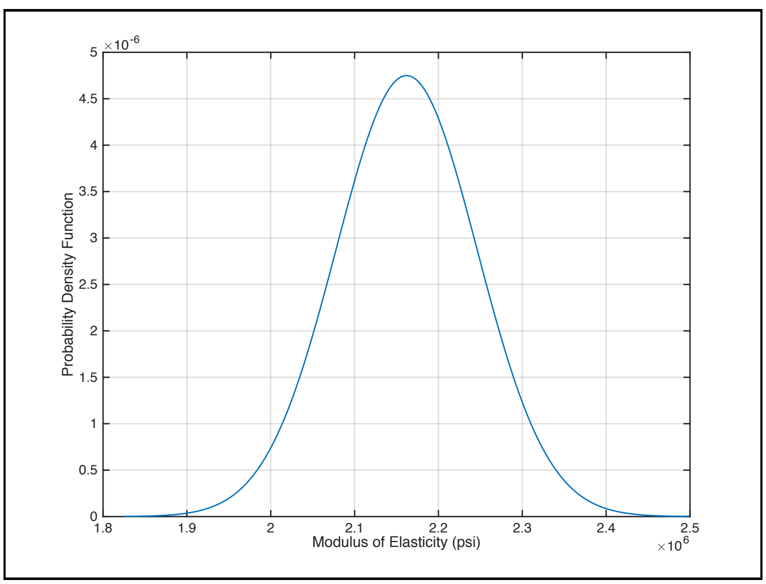
Solution

Date Due: April 16, 2015 Instructor: Trani

Problem 1

a) Task 1

```
%Problem 1, Task 1
%Programmer: Moises Bobadilla
%Date: 04/10/2015
%This script evaluates the PDF euqation using Modulus of Elasticity values
global sigma mu x

sigma = 8.4e4;
mu = 2.162e6;

x = [(mu-sigma*4):1:(mu+sigma*4)];


MOE = (1/(sigma*(2*pi).^(1/2)))*exp((-(x-mu).^2)/(2*(sigma).^2));

plot(x,MOE)
grid
xlabel('Modulus of Elasticity (psi)')
ylabel('Probability Density Function')
```



b) **Task 2**

```
%Problem 1, Task 2
%Programmer: Moises Bobadilla
%Date: 04/10/2015
%This function evaluates the PDF function, f(s)

function [fx]=PDFfunc(mu,sigma,x)

global mu sigma x

fx = (1./(sigma*sqrt(2*pi))).*exp(-((x-mu).^2)/((2*sigma).^2));
```

c) **Task 3**

```
%Problem 1, Task 3
%Programmer: Moises Bobadilla
%Date: 04/10/2015
%This script evaluates the probability that wood samples exceed a MOE of
%2e6 psi. It uses the function 'PDFfunc.m' previously created.

global sigma mu

sigma = 8.4e4;
mu = 2.162e6;

x = (mu-sigma*4):1:(mu+sigma*4); %defining values of random variable, x


%using previously created function and quad to estimate probability

Prob=quad('PDFfunc',2e6,(mu+sigma*4));

%display result on command window
msg = ['The probability that these samples meet the design criteria is ',...
    num2str(Prob*100),'%'];
disp(msg)
```

```
Command Window
New to MATLAB? See resources for Getting Started.
>> Problem1_T3
The probability that these samples meet the design criteria is 97.3076%
```

d) **Task 4**

Using the same function created for Task 2,

```
%Problem 1, Task 3
%Programmer: Moises Bobadilla
%Date: 04/10/2015
%This script evaluates the probability that wood samples exceed a MOE of
%2e6 psi. It uses the function 'PDFfunc.m' previously created.

global sigma mu

sigma = 8.4e4;
mu = 2.162e6;

%defining a specific number of samples to test
numSamples = 1000;
stepSize = ((mu+sigma*4)-(mu-sigma*4))/numSamples;


x = (mu-sigma*4):stepSize:(mu+sigma*4); %defining values of random variable, x


%using previously created function and quad to estimate probability

Prob=quad('PDFfunc',2.1e6,2.35e6);

%display result on command window
msg = ['The probability that these samples have MOE values between 2.1e6 and 2.35e6 is ',...
    num2str(Prob*100),'%'];
disp(msg)
```

```
New to MATLAB? See resources for Getting Started.
>> Problem1_T4
The probability that these samples have MOE values between 2.1e6 and 2.35e6 is 75.7164%
>>
```

Problem 2

a) **Tasks 1&2**

The values of table 2 were saved in a file named '**watwerway_depth.txt**' to be used for this problem.

```
% Problem 3, Task 1
% Programmer: Moises Bobadilla
% Date: 04/10/2015
% This script approximates a polynomial to fit the data in Problem 2

% Load the data (stored in a text file)

load waterway_depth.txt

% Data of Waterway station vs depth (2nd column)
% Column 1= station (m)
% Column 2 = depth (m)

% 0 0.00
% 5 -2.80
% 10    -4.50

HorizCoord=(waterway_depth(:,1));
VertProfile = (waterway_depth(:,2));
```

```

%SSE for each polynomal degree were estimated as follows:
%***OrderFit = estimate the value of the coeffients for each polynomial
%***OrderVals = calculate the values of polynomial using given profile
%SSE_*** = Estimates the SSE for that specific degree

%Second Order Polynomial
SecondOrderFit = polyfit(HorizCoord,VertProfile,2);
SecondOrderVals = polyval(SecondOrderFit,HorizCoord);
SSE_SecondOrder = sum((VertProfile-SecondOrderVals).^2);

%Third Order Polynomial
ThirdOrderFit = polyfit(HorizCoord,VertProfile,3);
ThirdOrderVals = polyval(ThirdOrderFit,HorizCoord);
SSE_ThirdOrder = sum((VertProfile-ThirdOrderVals).^2);

%Fourth Order Polynomial
FourthOrderFit = polyfit(HorizCoord,VertProfile,4);
FourthOrderVals = polyval(FourthOrderFit,HorizCoord);
SSE_FourthOrder = sum((VertProfile-FourthOrderVals).^2);
residuals = VertProfile-FourthOrderVals;
```

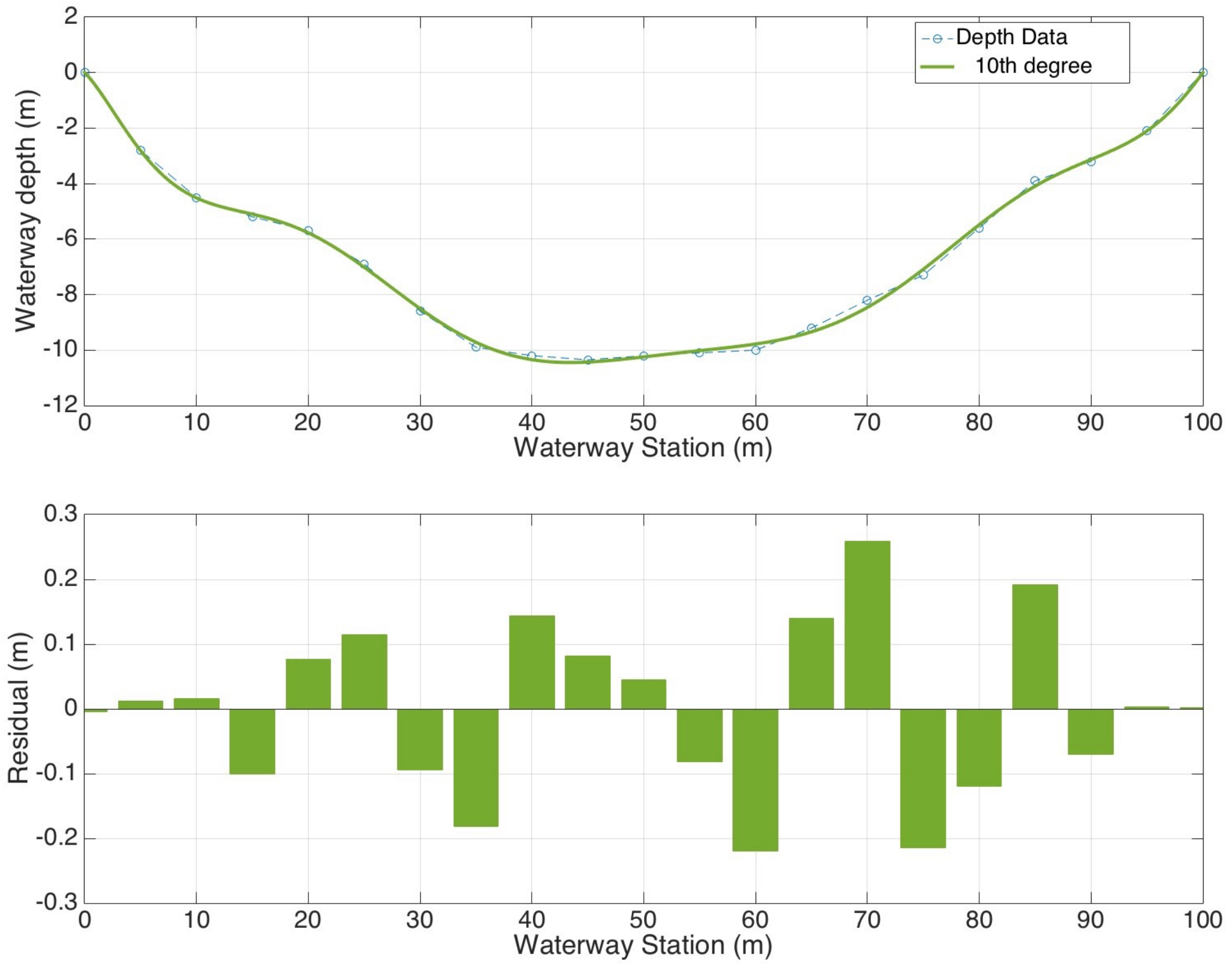


Figure : Waterway Station vs. Depth and Residuals. Tenth-Order Polynomial Approximation.

The tenth-order polynomial to fit the water depth distribution has the coefficients:

p1 = -1.0657e-15

p2 = 5.5812e-13

p3 = -1.2379e-10

p4 = 1.5122e-08

p5 = -1.1062e-06

p6 = 4.9254e-05

p7 = -0.0012894

p8 = 0.017858

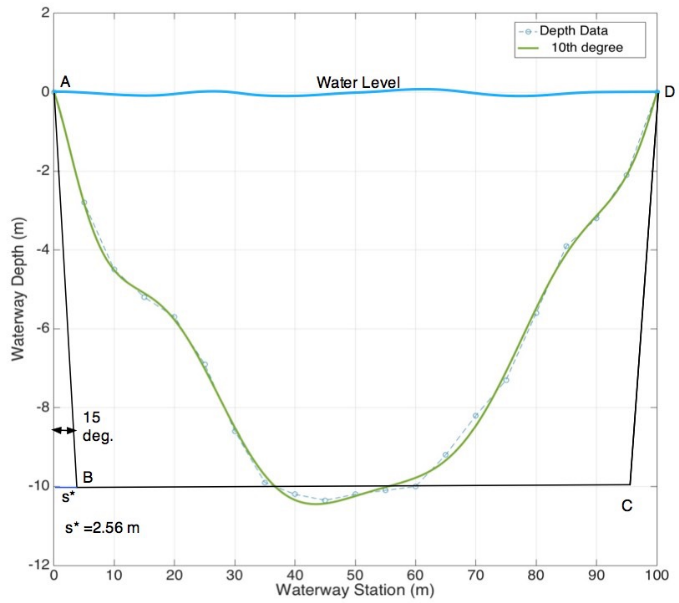
p9 = -0.093619

p10 = -0.408

p11 = 0.0039122

The approximation is very close to the actual data.

The area to be excavated using the Quad function (Newton-Cotes method) is shown the following diagram.



Area of Polygon ABCD = (100+(100-2*2.56))/2*10 = 974.4 m^2

Area of cross section = 671.9 m^2

NOTE: Between stations 36.7 and 55.5 m the channel has enough depth and does not need to be excavated.The small area between the polygon and the natural channel is 5.15 m2.

Area to be excavated = 974.4 - (671.9+5.15) = 297.4 m^2

Problem 3

```
%Problem 4, Task 1
%Programmer: Moises Bobadilla
%Date: 04/10/2015
%This script reads car data stored in cardata.txt file and plots engine
% weight vs horsepower

%loading the data
load cardata.txt

%asigning variables to each colum
weight = cardata(:,1);
t_circle = cardata(:,2);
displacement = cardata(:,3);
horsepower = cardata(:,4);
tanksize = cardata(:,5);

%plotting weight vs horsepower
figure
plot(weight,displacement,'or')
xlabel('Vehicle Weight (lb)')
ylabel('Engine Displacement (in^3)')
grid
```

```
%Vehicle Mass Estimation
%Second Order Polynomial
SecondOrderFit = polyfit(weight,displacement,2);
SecondOrderVals = polyval(SecondOrderFit,weight);
SSE_SecondOrder = sum((displacement-SecondOrderVals).^2)

%Third Order Polynomial
ThirdOrderFit = polyfit(weight,displacement,3);
ThirdOrderVals = polyval(ThirdOrderFit,weight);
SSE_ThirdOrder = sum((displacement-ThirdOrderVals).^2)

%Fourth Order Polynomial
FourthOrderFit = polyfit(weight,displacement,4);
FourthOrderVals = polyval(FourthOrderFit,weight);
SSE_FourthOrder = sum((displacement-FourthOrderVals).^2)
residuals = displacement-FourthOrderVals;

%Fifth Order Polynomial
FifthOrderFit = polyfit(weight,displacement,5);
FifthOrderVals = polyval(FifthOrderFit,weight);
SSE_FifthOrder = sum((displacement-FifthOrderVals).^2)
```

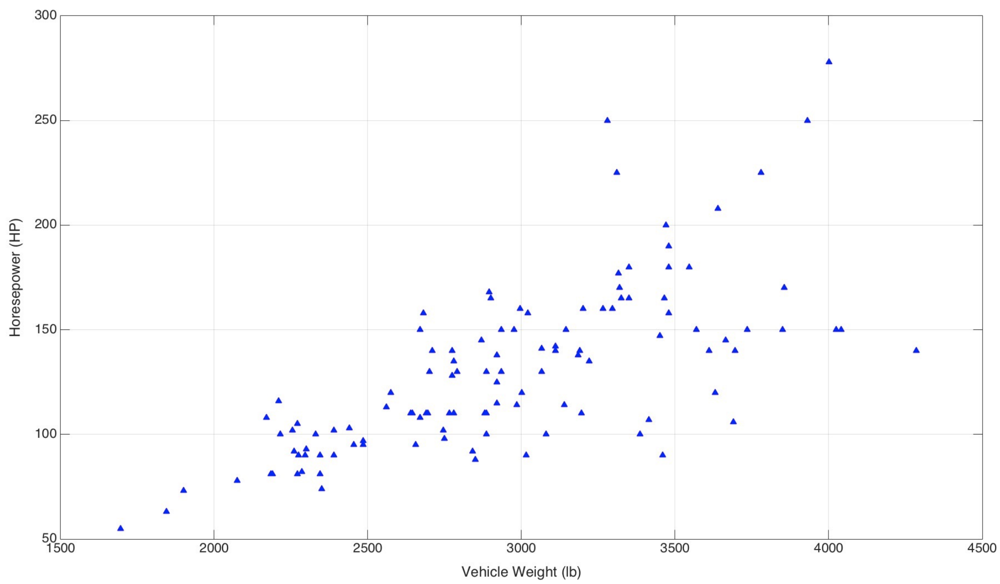
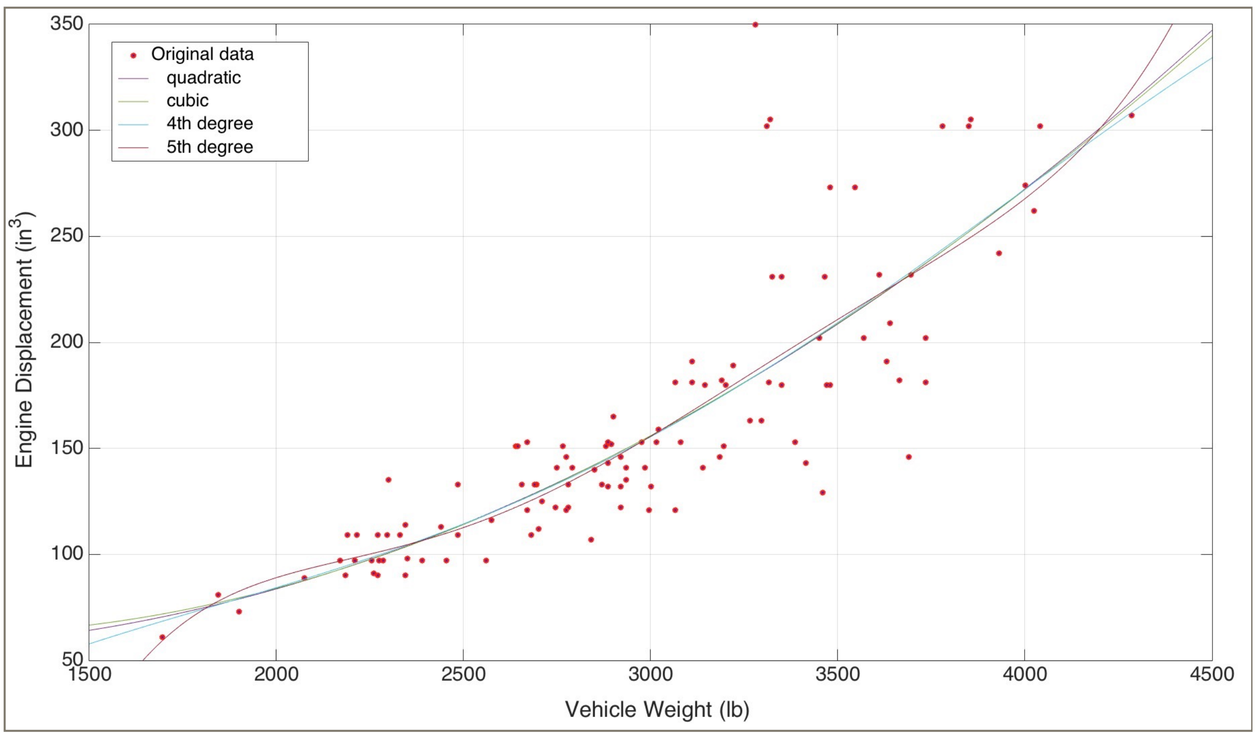


Figure : Vehicle Weight vs. Engine Horsepower.

Table: *Vehicle Weight & Engine Displacement SSE Estimation.*

Degree	SSE
Second	1.2386E+05
Third	1.2385E+05
Fourth	1.2380E+05
Fifth	1.2314E+05

```
>> weight =  [2300 2760 3400 2230 3450];
>> [displacement,Horsepower] = DisplacementHorse(weight)

displacement =

  102.3060  132.1468  199.9086   99.3182  205.3981


Horsepower =

   93.8251  120.8854  155.9345   89.2640  158.7743

>>
```



From the table above we may conclude that the fifth order polynomial is the best solution (lowest value of SSE). However, if we examine the resulting polynomials plotted against the data (shown in the next Figure), we conclude that most of the regressions provide similar accuracy with the fifth-order approximation. In fact the fifth-order polynomial introduces some “strange" artifacts at the ends of the regression. This rules in favor of a simple quadratic (2nd order polynomial) or perhaps a 3rd order polynomial.

```matlab
%Problem 4, Task 3
%Programmer: Moises Bobadilla
%Date: 04/10/2015
%This function estimates vehicle displacement and HP given values of weight

function [displacement,HP]=DisplacementHorse(weight)

dispCoefficients = [2.01398319935638e-14,-3.04091904510251e-10,...
    1.79960687340542e-06,-0.00518856869328795,7.33345892072261,...
    -3999.46836310293];

HPCoefficients = [-1.20281703129457e-14,1.69027265394147e-10,...
    -9.35522677360560e-07,0.00254359518630748,-3.33247618071494,...
    1729.50464276513];

displacement = polyval(dispCoefficients,weight);
HP = polyval(HPCoefficients,weight);
end
```

Degree	**Sum of Square Errors (SSE)**
Second	9.0091E+04
Third	8.9397E+04
Fourth	8.887E+04
Fifth	8.8754E+04

b) <u>Task 3</u>

Call from command window using provided mass test values :