Mock Class Test 2022 Model Solutions

Q1

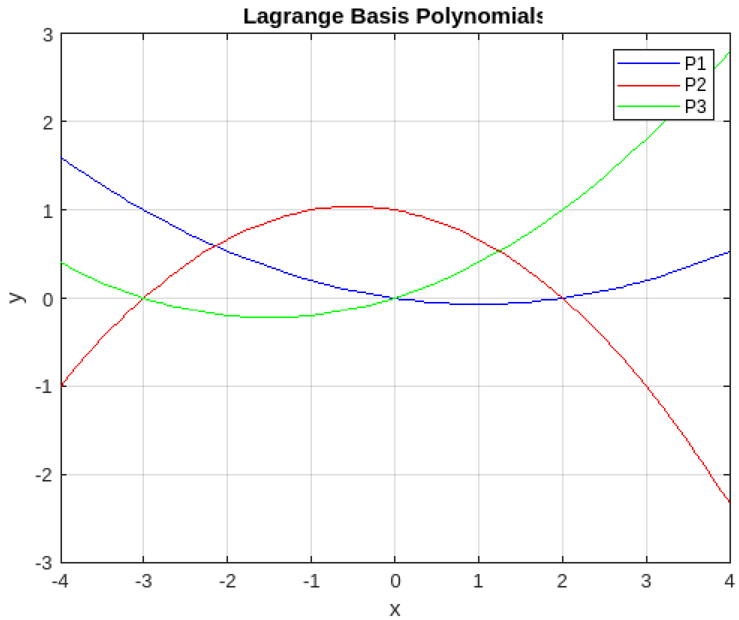
```
%This script file plots the Lagrange basis polynomials of
%the quadratic polynomial passing through points (-3,-4), (0,-4) and (2,6).
clc
clear all

%create vectors X and Y vectors for data set
X=[-3 0 2];
Y=[-4 -4 6];

%calculate the differences in x-values
a=X(1)-X(2);
b=X(2)-X(3);
c=X(3)-X(1);

P1= @(x) (x-X(2)).*(x-X(3))/(-a*c);
P2= @(x) (x-X(1)).*(x-X(3))/(-a*b);
P3= @(x) (x-X(1)).*(x-X(2))/(-b*c);

x=-4:0.1:4;
figure
plot(x,P1(x),'b',x,P2(x),'r',x,P3(x),'g')
grid on
title('Lagrange Basis Polynomials')
xlabel('x')
ylabel('y')
legend('P1','P2','P3')
```



Q2

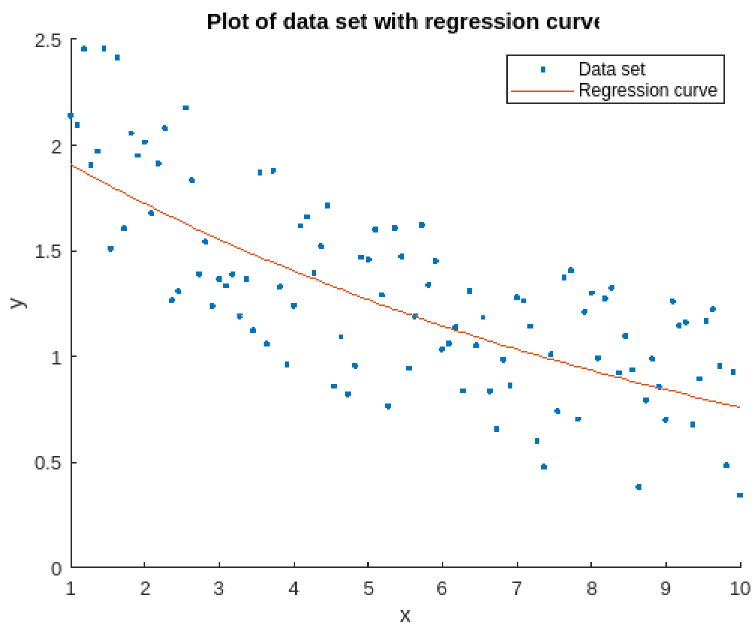
The estimation function $\widehat{y}(x) = \alpha_1 e^{\alpha_2 x}$ can be linearised by taking the transformation $\widetilde{y} = \log(\widehat{y})$. This transforms it to $\widetilde{y} = \widetilde{\alpha}_1 + \alpha_2 x$ where $\widetilde{\alpha}_1 = \log \alpha_1$. Now least squares regression can be applied to the resulting equation and the values of $\widetilde{\alpha}_1$ and α_2 can be obtained. The value of α_1 is recovered by using the equation $\alpha_1 = e^{\widetilde{\alpha}_1}$. The regression curve $\widehat{y}(x) = \alpha_1 e^{\alpha_2 x}$ is then plotted.

```
%This script file plots regression curve for the given data set
%using estimation function yhat=alpha1*exp(alpha2*x)

clc
clear all
%create data set
x=linspace(1,10,100); %create data for vector x
y=2*exp(-0.2*x)+rand(size(x)); %create data for vector y
%plot data set
figure
hold on
plot(x,y,'.')
title('Plot of data set with regression curve')
xlabel('x')
ylabel('y')

%Regression
```

```
tildey=log(y); %tildey is transformed y
A=[x',ones(size(x'))]; %create matrix A of basis functions
beta=pinv(A)*tildey';
alpha1=exp(beta(2));
alpha2=beta(1);
yhat=alpha1*exp(alpha2*x);
%plot the regression curve
plot(x,yhat)
hold off
legend('Data set','Regression curve')
```



Q3

By drawing graphs of $y = \cos x$ and $y = x$, one can see that the intersection occurs in the interval $(0, \pi/2)$. Let us assume the guess value x0 = 1.

```
f=@(x) cos(x)-x;
df=@(x) -sin(x)-1;
x0=1;
tol=0.001;
myNR_while(f,df,x0,tol)
```

```
Iteration1
r1=0.75036
Iteration2
r2=0.73911
```

The function description is given below.

```
function root = myNR_while(f,df,x0,tol)
%This function finds the root of function f by Newton-Raphson method
%using while-loop
%df is derivative of function f
%x0 is the guess value
%tol is tolerance

iter=0;

if abs(f(x0))<tol
    root=x0; % root=x0 if absolute value of error is less than tolerance
    disp(['root=',num2str(root)]); %display root
end

while abs(f(x0))>tol
    % find root if absolute value of error is greater than tolerance
    iter=iter+1;
     disp(['Iteration',num2str(iter)]);
    x0=x0-f(x0)/df(x0);
    disp(['r',num2str(iter),'=',num2str(x0)]); %display root at each iteration
end

end
```