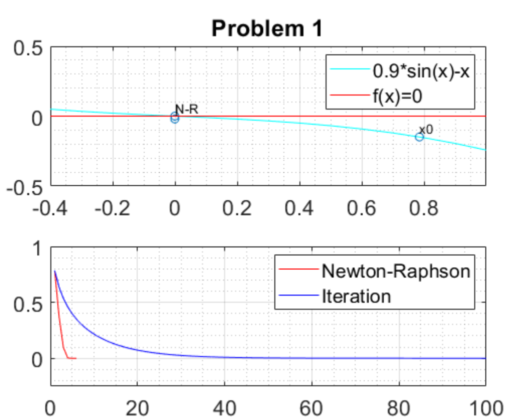
MATLAB (Problems 1-3)

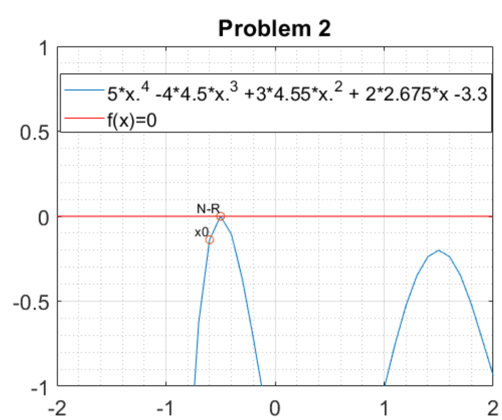
Output:

```
--------------------------------------------------------------
Oscar Martinez          Homework 5: Problems 1-3          STA 5106
--------------------------------------------------------------
-----------Problem 1----------
---Part (i): Simple Iterations---

i =

106

---Part (ii): Newton-Raphson---

x =

0.7854    0.3756    0.0963    0.0026    0.0000    0.0000


i =

6

---Part (iii): Plot---
----------Problem 2----------

m =

2


K =

0.5000


x =

Columns 1 through 9

-0.6000   -0.5530   -0.5274   -0.5139   -0.5070   -0.5035   -0.5018   -0.5009   -0.5004

Columns 10 through 18

-0.5002   -0.5001   -0.5001   -0.5000   -0.5000   -0.5000   -0.5000   -0.5000   -0.5000
```

```
i =

18


Kalt =

0.5000


Malt =

2

----------Problem 3----------

i =

7


ans =

10.0670
```

Figures:





Code:

```
clc
clear

```

```matlab
% Diary
dfile ='MATLAB_Output_OM.txt';
if exist(dfile, 'file') ; delete(dfile); end
diary(dfile)
diary on
diary MATLAB_Output_OM.txt

%Introduction
fprintf('----------------------------------------------------------------\n');
fprintf('Oscar Martinez \t Homework 5: Problems 1-3 \t STA 5106\n');
fprintf('----------------------------------------------------------------\n');

%------Problem 1:------
fprintf('------------Problem 1-----------\n');

%Part i
fprintf('---Part (i): Simple Iterations---\n');

%Define the function inline
f = inline('0.9*sin(x)-x', 'x');

x0 = pi/4;

xa(1) = x0;
gxa = f(xa(1))+xa(1);
i = 1;
while (abs(xa(i) - gxa) > 1e-6)
gxa = xa(i);
xa(i+1) = f(xa(i)) + xa(i);
i = i+1;
end
%xa(i)
i


%Part ii
fprintf('---Part (ii): Newton-Raphson---\n');

% define the function
h = inline('0.9*sin(x)-x', 'x');
x0 = pi/4;
dh = inline('0.9*cos(x)-1', 'x');

```

```
%The N-R algorithm
clear x;
x0 = pi/4;
x(1) = x0;
gx = x(1)-h(x(1))/dh(x(1));
i = 1;
while (abs(x(i) - gx) > 1e-6)
gx = x(i);
x(i+1) = x(i) - h(x(i))/dh(x(i));
i = i+1;
end
x
i

%Part iii
fprintf('---Part (iii): Plot---\n');

% plot the function
t = -1:0.1:2;
yt = f(t);
figure(1);
subplot(2,1,1);
Y=plot(t,yt, 'c-');
hold on;
XP=[pi/4 xa(106) x(5)];
YP=[f(pi/4) f(xa(11)) f(x(5))];
labels = {'x0','I','N-R'};
plot(XP,YP,'o');
text(XP,YP,labels,'VerticalAlignment','bottom','HorizontalAlignment','left')
Z=plot([-1 1], [0 0], 'r');
grid on;
grid minor;
title('Problem 1')
axis([-.4 1 -.5 .5]);
set(gca, 'fontsize', 16);
legend([Y Z],{'0.9*sin(x)-x','f(x)=0'})
subplot(2,1,2);
plot(x,'r');
grid on;
grid minor;
hold on;
plot(xa,'b');
set(gca, 'fontsize', 16);
axis([0 100 -.25 1])
```

```
legend('Newton-Raphson','Iteration')
hold off;


%------Problem 2:------
fprintf('----------Problem 2----------\n');
clear x;

%Define the function
f2 = inline('x.^5 - 4.5*x.^4 +4.55*x.^3 + 2.675*x.^2 -3.3*x -1.4375', 'x');

%Find the Roots
p=[1 -4.5 4.55 2.675 -3.3 -1.4375];
r=roots(p);

%Find the multiplicity of root rt
rt=-0.5; %Define the wanted root
[s,t]=size(r);
m=0;
for j = 1:s
if abs(rt-r(j)) < 1e-6
m=m+1;
end
end
m
K=(m-1)/m

%Begin the N-R Algorigthm
x0 = -0.6;
df2 = inline('5*x.^4 -4*4.5*x.^3 +3*4.55*x.^2 + 2*2.675*x -3.3', 'x');
clear x;
x(1) = x0;
gx = x(1)-f2(x(1))/df2(x(1));
i = 1;
while (abs(x(i) - gx) > 1e-6)
gx = x(i);
x(i+1) = x(i) - f2(x(i))/df2(x(i));
i = i+1;
end
x
i

%Alternate way to find K
%Get E
for j = 1:i-1
```

```
E(j)=abs(x(j)-x(j+1));
end
%K~Ei+1/E
for j = 1:i-2
Kalt=min(E(j+1)/E(j));
end
Kalt %K=(m-1)/m
Malt=1/(1-K) %1/(1-K)

%Plot
figure(2)
t=[-5:0.1:5];
yt=f2(t);
Y=plot(t, yt);
hold on;
Z=plot([-5 5], [0 0], 'r');
%plot(r, 0, 'go', r, f2(r), 'co', x(i), f2(x(i)), 'ro');
XP2=[-0.6 x(i)];
YP2=[f2(-0.6) f2(x(18))];
labels = {'x0','N-R'};
plot(XP2,YP2,'o');
text(XP2,YP2,labels,'VerticalAlignment','bottom','HorizontalAlignment','right')
axis([-2 2 -1 1]);
legend([Y Z],{'5*x.^4 -4*4.5*x.^3 +3*4.55*x.^2 + 2*2.675*x -3.3','f(x)=0'});
ax=gca;
ax.FontSize=16;
grid on;
grid minor;
title('Problem 2')
hold off;


%------Problem 3:------
fprintf('----------Problem 3----------\n');

clear
%Load the data
load hw5_3_data.mat
[m,n]=size(X);

theta0 = 7;


```

```
%N-R Algo
theta(1) = theta0;
%LogL=LogL + x(i) - X(j)-2*log(1+exp(x(i)-X(j)));
i = 1;
dtheta = theta(1)+.5;
while (abs(theta(i) - dtheta) > 1e-6)
dtheta = theta(i);
dLogL = sum(1-2*(exp(theta(i)-X)./(1+exp(theta(i)-X))));
ddLogL = -2*sum(exp(theta(i)-X)./(1+exp(theta(i)-X)).^2);
theta(i+1) = theta(i) - dLogL/ddLogL;
i = i+1;
end
i
theta(i)

diary off
```

Problem 3

Let $X_1, X_2, \ldots, X_n$ be independent and identically distributed samples from a logistic distribution with the probability density function

$$f(x|\theta) = \frac{\exp(\theta - x)}{(1 + \exp(\theta - x))^2}$$

Given the values of $X_1, X_2, \ldots, X_n$ in "hw5_3_data" from the blackboard website, our goal is to find the maximum likelihood estimate (MLE) of θ, using the following steps:

(a)

Derive an expression for the log likelihood function

$$l(\theta) = \sum_{i=1}^{n} \log(f(X_i|\theta)),$$

such that the MLE is given by

$$\hat{\theta} = \arg\max_{\theta} l(\theta).$$

$$\begin{aligned}
l(\theta) &= \sum_{i=1}^{n} \log(f(X_i|\theta)) \\
&= \sum_{i=1}^{n} \log\left(\frac{\exp(\theta - X_i)}{(1+\exp(\theta - X_i))^2}\right) \\
&= \sum_{i=1}^{n} \log(\exp(\theta - X_i)) - 2\log(1+\exp(\theta - X_i)) \\
&= n\theta - \sum_{i=1}^{n} (X_i - 2\log(1+\exp(\theta - X_i)))
\end{aligned}$$

(b)

Find the expression for $\dot{l}(\theta)$ and $\ddot{l}(\theta)$, the first and the second derivatives of l with respect to θ. Verify that $\ddot{l}(\theta) < 0$.

$$\dot{l} = \frac{dl(\theta)}{d\theta} = n - 2\left[\sum_{i=1}^{n} \frac{\exp(\theta - X_i)}{1+\exp(\theta - X_i)}\right]$$

$$\ddot{l} = \frac{d^2 l(\theta)}{d\theta^2} = -2\left[\sum_{i=1}^{n} \frac{\exp(\theta - X_i)}{(1+\exp(\theta - X_i))^2}\right]$$

As $\exp(\alpha) > 0, \quad \forall \alpha \in \mathbb{R}$, the numerator and denominator of this expression are both positive. Thus since the quotient is positive and is being multiplied by a negative number, the expression must be less than 0, as needed.

(c)

See above output.

Problems 4-5

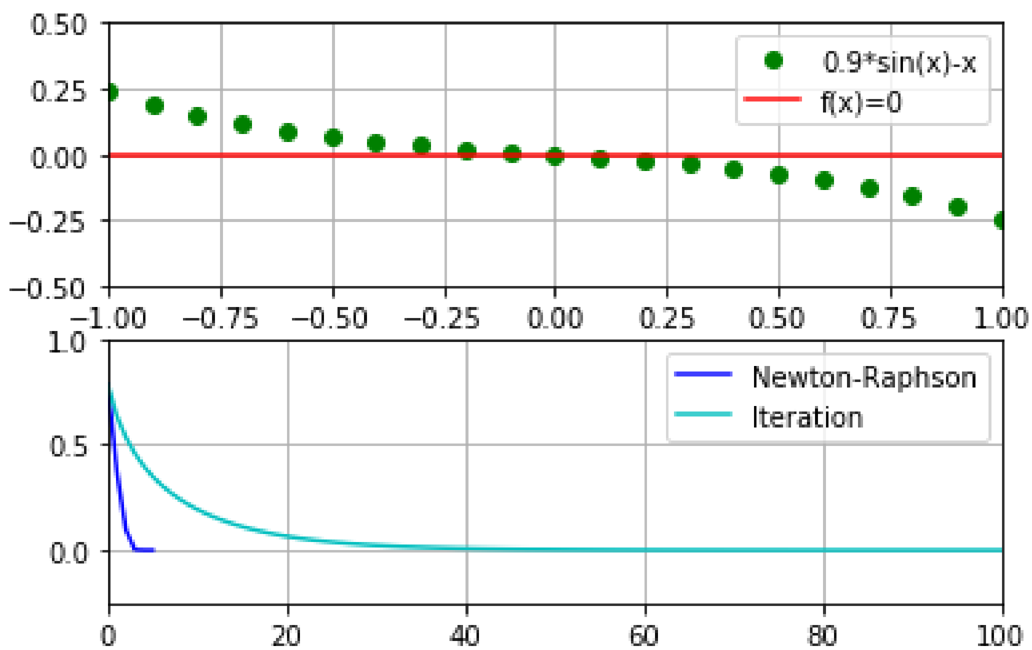
```
In [62]: #Problem 4
print("-------Problem 4-------")
from numpy import *
from matplotlib import pyplot
set_printoptions(precision=4)

# define the function
f = lambda x: 0.9*sin(x)-x
df = lambda x: 0.9*cos(x)-1

#Part i
print("---Part(i) Simple Iterations---")
```

```
x0 = pi/4
xa0 = pi/4

#Algorithm
xa = zeros(200)
xa[0]=  xa0
gxa = f(xa[0]) + xa[0]
i = 0

while abs(xa[i] - gxa) > 1e-6:
gxa = xa[i]
xa[i+1] = f(xa[i]) + xa[i]
i = i + 1

ia=i+1
print('Iterations until convergence: ',i+1)

#Part ii
print("---Part(ii) Newton-Raphson---")
# Newton-Raphson
x0 = pi/4

x = zeros(100)
x[0]=  x0
gx = x[0] - f(x[0])/df(x[0])
i = 0
while abs(x[i] - gx) > 1e-6:
gx = x[i]
x[i+1] = x[i]- f(x[i])/df(x[i])
i = i + 1

print('Iterations until convergence: ',i+1)

#Part iii
print("---Part(iii) Plot---")

# plot the function
t = arange(-1, 2, 0.1)
yt = f(t)

pyplot.title(['Problem 1'])
pyplot.subplot(2,1,1)
pyplot.plot(t, yt, 'go')
pyplot.plot((-1, 1), (0, 0), 'r')
pyplot.grid(True)
```

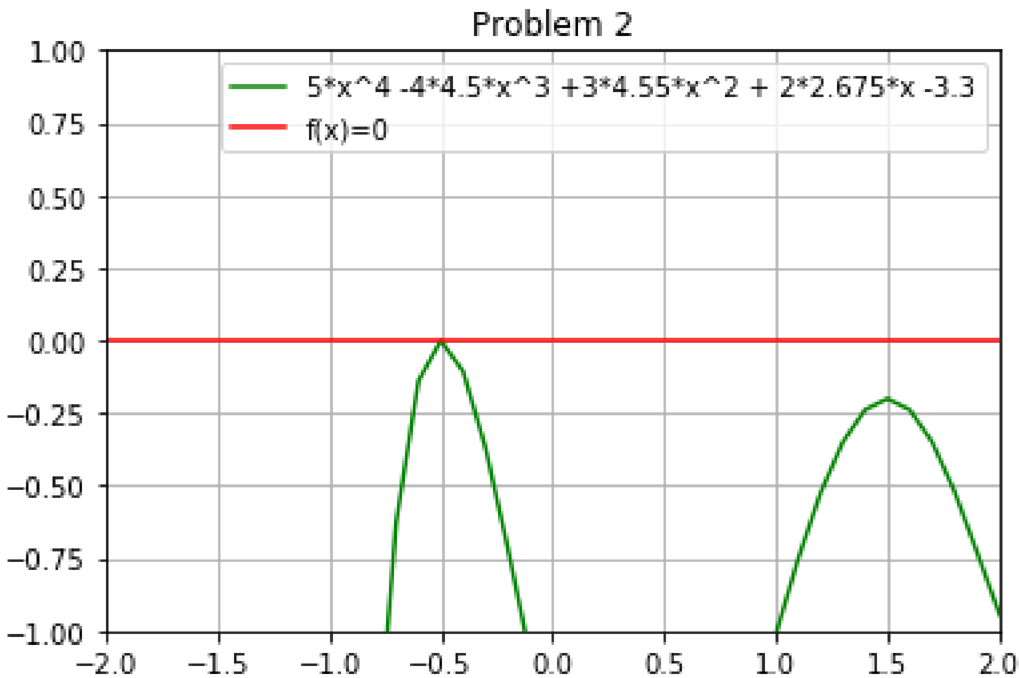
```
pyplot.axis([-1, 1, -.5, .5])
pyplot.legend(['0.9*sin(x)-x','f(x)=0'])
pyplot.subplot(2,1,2)
pyplot.plot(range(0,i+1), x[0:i+1], 'b-')
pyplot.plot(range(0,ia), xa[0:ia], 'c-')
pyplot.grid(True)
pyplot.axis([0, 100, -.25, 1])
pyplot.legend(['Newton-Raphson','Iteration'])
pyplot.show()

#Problem 5
print("-------Problem 5-------")

#Define the function
f2 = lambda x: x**5 - 4.5*x**4 +4.55*x**3 + 2.675*x**2 -3.3*x -1.4375
df2 = lambda x: 5*x**4 -4*4.5*x**3 +3*4.55*x**2 + 2*2.675*x -3.3

#Begin Algo
x0 = -0.6;
x = zeros(100)
x[0]=  x0
gx = x[0] - f2(x[0])/df2(x[0])
i = 0
while abs(x[i] - gx) > 1e-6:
gx = x[i]
x[i+1] = x[i]- f2(x[i])/df2(x[i])
i = i + 1

print('Iterations until convergence: ',i+1)

#M-K
E = zeros(i+1)
for j in range(i+1):
E[j]=abs(x[j]-x[j+1])

K = zeros(i-1)
for j in range(i-1):
K[j]=(E[j+1]/E[j])

Kmin=min(K)
print('K ~ ', Kmin)
M=1/(1-Kmin)
print('M ~ ', M)

#Plot
```

```
# plot the function
t = arange(-1, 5, 0.1)
y2t = f2(t)

pyplot.title('Problem 2')
pyplot.plot(t, y2t, 'g-')
pyplot.plot((-5, 5), (0, 0), 'r')
pyplot.grid(True)
pyplot.axis([-2, 2, -1, 1])
pyplot.legend(['5*x^4 -4*4.5*x^3 +3*4.55*x^2 + 2*2.675*x -3.3','f(x)=0'])
pyplot.show()
```

```
-------Problem 4-------
---Part(i) Simple Iterations---
Iterations until convergence:  106
---Part(ii) Newton-Raphson---
Iterations until convergence:  6
---Part(iii) Plot---
```




```
-------Problem 5-------
Iterations until convergence:  18
K ~  0.5000022589142745
M ~  2.0000090356979197
```


Problem 2
5*x^4 -4*4.5*x^3 +3*4.55*x^2 + 2*2.675*x -3.3
f(x)=0
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

```
In [ ]:
```