Problem2:

Solution:

The code for the a and b is :

Function code:

function [yPlot] = largInterpo(xData,yData, xPlot)

N = length(xData);

yP = 0;

for i = 1: N

Li =1 ;

for j = 1 : N

if i ~=j

Li = (Li).\*((xPlot- xData(j))/(xData(i)- xData(j)));

end

end

yP = yP+((Li)\*yData(i));

end

yPlot = yP;

end

Script code:

clear all;

xData = linspace(-1,1,9);

yData = 1./(1+14\*xData.^2);

xPlot = linspace(-1,1,100);

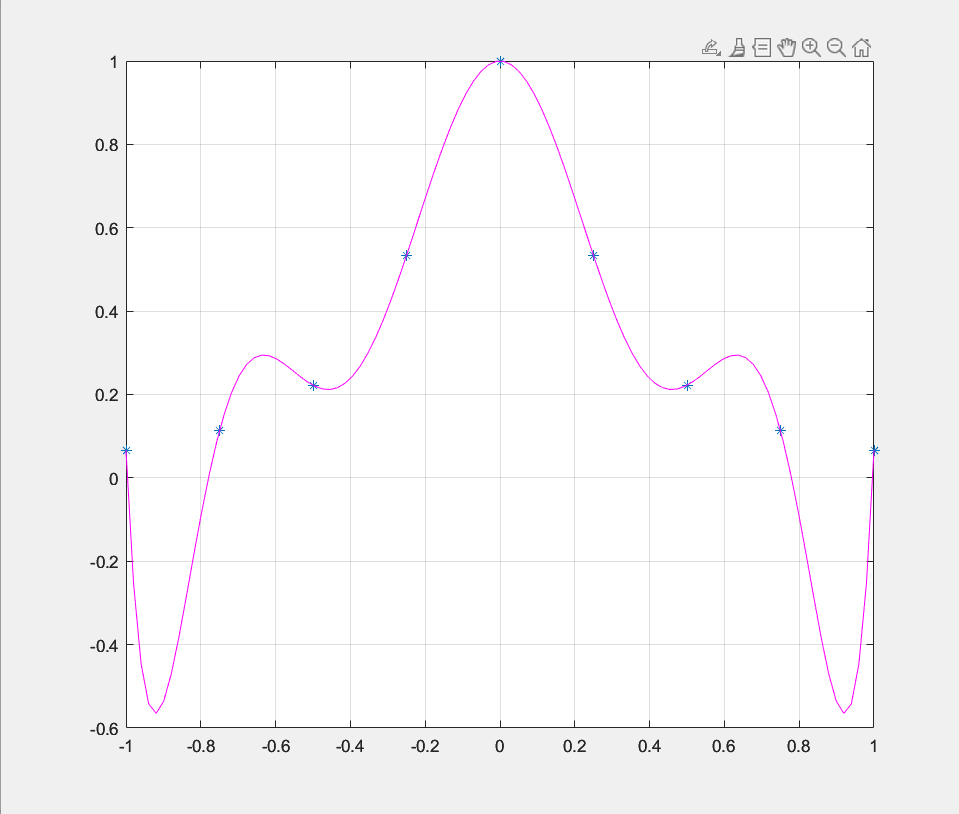
yPlot = largInterpo(xData, yData, xPlot);

%plot(xPlot, yPlot, 'm', xData, yData, "\*" );

plot( xData, yData, "\*" ,xPlot, yPlot, 'm');

grid on

Output Graph:



Again:

Solution for part c:

Here I have only made changes to the xData

Function file:

function [yPlot] = largInterpoChe(xData,yData, xPlot)

N = length(xData);

yP = 0;

for i = 1: N

Li =1 ;

for j = 1 : N

if i ~=j

Li = (Li).\*((xPlot- xData(j))/(xData(i)- xData(j)));

end

end

yP = yP+((Li)\*yData(i));

end

yPlot = yP;

end

ScriptFile:

clear all;

%xData = linspace(-1,1,10)

n =9;

k = 0:n;

xData = -cos(pi\*k./n);

yData = 1./(1+14\*xData.^2);

xPlot = linspace(-1,1,100);

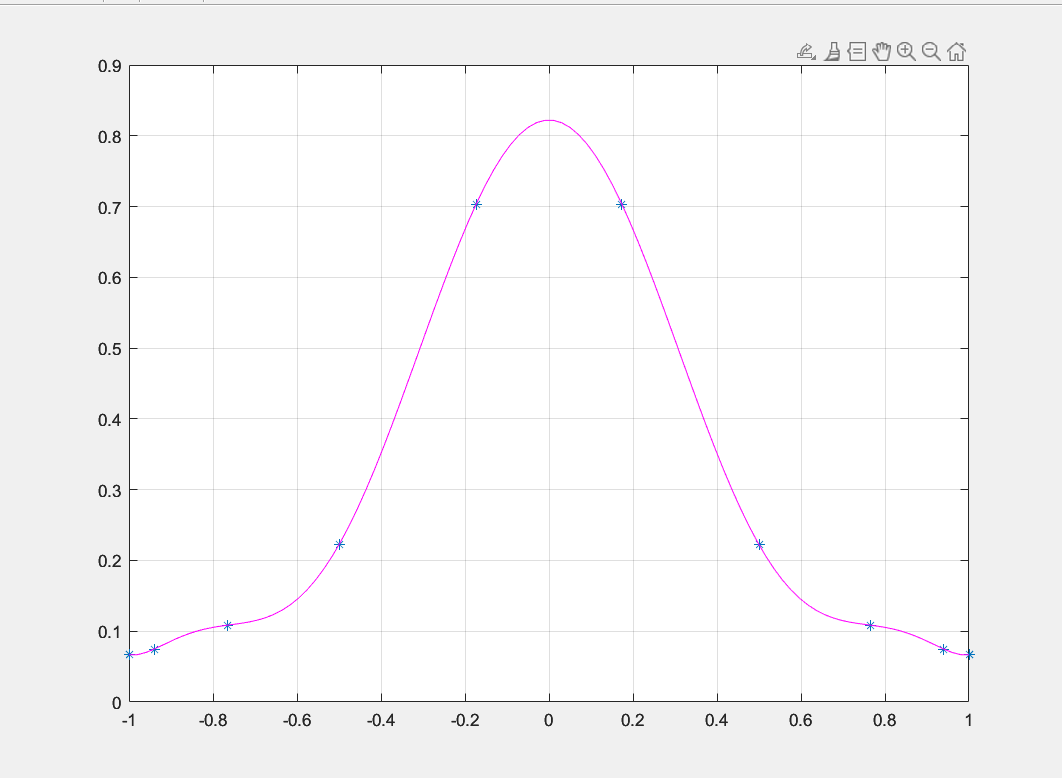
yPlot = largInterpo(xData, yData, xPlot);

%plot(xPlot, yPlot, 'm', xData, yData, "\*" );

plot( xData, yData, "\*" ,xPlot, yPlot, 'm');

grid on

Output File:



Problem 5:

I have solved by hand to find the interpolating polynomials of degrees two, three and four.

Diagram

Description automatically generatedMy code for this problem is :

fx = @(x)0.5\*x.^3 -2\*x.^2+3\*x+ 1;

p2x = @(x) -17+12.5.\*(x+2)-3.5.\*(x+2).\*(x+1);

p3x = @(x) -17+12.5.\*(x+2)-3.5.\*(x+2).\*(x+1) + 0.5.\*(x+2).\*(x+1).\*(x);

p4x = @(x) -17+12.5.\*(x+2)-3.5.\*(x+2).\*(x+1) + 0.5.\*(x+2).\*(x+1).\*(x) - 0.25.\*(x+2).\*(x+1).\*(x).\*(x-1);

xData = [-2,-1, 0,1,2];

xPlot = linspace(-2,2,100);

plot(xData, fx(xData),'\*', xPlot, p2x(xPlot), 'b', xPlot, p3x(xPlot), 'r', xPlot, p4x(xPlot), 'g',xPlot,fx(xPlot), 'm');

The plot is :

Chart

Description automatically generated