Damian Franco

CS 375 – 001

Homework #8:

1) For number 1, we were given a data set, a basis function and where to evaluate the interpolating function at. The basis function that is given to us is {1, cos *x*, sin *x*} which is a trigonometric polynomial function. When I say what was given, I noticed that the way to approach this problem was with a monomial approach. The first step I did was I solved for the main *V* matrix in the Vandermonde system setup. The following phi equations were applied accordingly $\phi_1 = 1$, $\phi_2 = \cos x$, and lastly $\phi_3 = \sin x$. The findings from that were then used to find the *c* values which was the following $c_1 = 1$, $c_2 = 3$, and $c_3 = 2$. Next, I plugged the *c* values into the interpolation equation. Lastly, I plugged the *x* value pi/4 into the interpolation equation and solved which resulted in ***1.7071067812***. My work is shown below for this problem.

1) $D_3 = \{(0,4), (\pi/2, -1), (\pi, -2)\}$

$B_3 = \{1, \cos x, \sin x\}$ $\quad \phi_1 = 1, \quad \phi_2 = \cos x \quad \phi_3 = \sin x$

$x = \pi/4$

$$\begin{bmatrix} \phi_1(0) & \phi_2(0) & \phi_3(0) \\ \phi_1(\pi/2) & \phi_2(\pi/2) & \phi_3(\pi/2) \\ \phi_1(\pi) & \phi_2(\pi) & \phi_3(\pi) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 4 \\ -1 \\ -2 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & \cos(0) & \sin(0) \\ 1 & \cos(\pi/2) & \sin(\pi/2) \\ 1 & \cos(\pi) & \sin(\pi) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 4 \\ -1 \\ -2 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 4 \\ -1 \\ -2 \end{bmatrix}$$

$$= \left[\begin{array}{ccc|c} 1 & 1 & 0 & 4 \\ 1 & 0 & 1 & -1 \\ 1 & -1 & 0 & -2 \end{array}\right] \longrightarrow \left[\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & -2 \end{array}\right]$$

$c_1 = 1$

$c_2 = 3$

$c_3 = -2$

$$\begin{aligned}
&= c_1 \phi_1(x) + c_2 \phi_2(x) + c_3 \phi(x) \quad \boxed{x = \pi/4} \\
&= \phi_1(x) + 3\phi_2(x) - 2\phi_3(x) \\
&= 1 + 3\cos(x) - 2\sin(x) \\
&= 1 + 3\cos(\pi/4) - 2\sin(\pi/4) \\
&= 1 + 3(\sqrt{2}/2) - 2(\sqrt{2}/2) \\
&= 1 + 3\sqrt{2}/2 - 2\sqrt{2}/2 \\
&= 1 + \sqrt{2}/2 \\
&\boxed{\approx 1.7071067812}
\end{aligned}$$

2) Number 2 asked us to consider the given data set and find the polynomial which interpolates the data by using the monomial, Newton, and Lagrange approaches. You can find all the work for each part below in the screen shots of my work. The following approaches are indicated as such: (I) monomial (II) Newton (III) Lagrange. Below is also a brief explanation of each approach (a) as well as part (b).

   a) For part (a), I will briefly run through how I used each approach to achieve the polynomial. For the **(I)** monomial approach, I used the same technique as number 1 on this homework. The basis for the monomial approach in this case should be $\{1, x, x^2\}$ which leads to the phi functions: $\phi_1 = 1$, $\phi_2 = x$, and lastly $\phi_3 = x^2$. I then plugged them in and found the V matrix in the Vandermonde system setup and solved each phi function. After, I solved the linear system for the c_1, c_2 and c_3. I found that $c_1 = 1$, $c_2 = -5$, and $c_3 = 2$. I then plugged them into the interpolation equation and got the following equation $\boldsymbol{1 - 5x + 2x^2}$**.** I knew that this equation was the equation I would have to find for each approach. For the **(II)** Newton approach, I based my work off the example shown in the narration for the Newton approach. The basis for this approach should be $\{1, (x - x_1), (x - x_1)*(x - x_2)\}$. Next, I found the dived differences of each point and ended up coming out with the numbers $f\,(x_1) = 8$, $f\,[x_1, x_2] = -3$, and $f\,[x_1, x_2, x_3] = 2$. I then plugged those into the $p(x)$ function and combined like terms and ended up getting $\boldsymbol{1 - 5x + 2x^2}$ which is the same answer as the monomial approach. Lastly, I then did the **(III)** Lagrange approach to the problem and the first step of that is to find the l functions and I found the following l's: $l_1 = 1/2*(x + 1)$, $l_2 = -1/3*(x + 1)*(x - 3)$, and finally $l_3 = 1/4*(x + 1)*(x - 2)$. I then plugged those into the $p(x)$ equation and simplified by combining like terms and ended up getting the following equation: $2x^2 - 5x + 1$, which is equal to $\boldsymbol{1 - 5x + 2x^2}$. All work for this section is shown below.

   b) All three polynomials all equal $\boldsymbol{1 - 5x + 2x^2}$, and that is shown in my work below. All three approaches should equal if I did the work correctly and they all do in this case.

2) $D_3 = \{(-1, 8), (2, -1), (3, 4)\}$

I) $B_3 = \{1, x, x^2\}$ $\quad \phi_1 = 1, \ \phi_2 = x, \ \phi_3 = x^2$

$$\begin{bmatrix} \phi_1(-1) & \phi_2(-1) & \phi_3(-1) \\ \phi_1(2) & \phi_2(2) & \phi_3(2) \\ \phi_1(3) & \phi_2(3) & \phi_3(3) \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 8 \\ -1 \\ 4 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & -1 & 1 \\ 1 & 2 & 4 \\ 1 & 3 & 9 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} 8 \\ -1 \\ 4 \end{bmatrix}$$

$$= \left[\begin{array}{ccc|c} 1 & -1 & 1 & 8 \\ 1 & 2 & 4 & -1 \\ 1 & 3 & 9 & 4 \end{array}\right] \longrightarrow \left[\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -5 \\ 0 & 0 & 1 & 2 \end{array}\right]$$

$c_1 = 1$
$c_2 = -5$
$c_3 = 2$

$$p(x) = c_1 \phi_1(x) + c_2 \phi_2(x) + c_3 \phi_3(x)$$
$$= 1 \phi_1(x) - 5 \phi_2(x) + 2 \phi_3(x)$$
$$= 1 - 5x + 2x^2$$

<u>Interpolation (monomial):</u>

$$\boxed{p(x) = 1 - 5x + 2x^2}$$

II) $D_3 = \{(-1, 8), (2, -1), (3, 4)\}$

$\phi_1 = 1$, $\phi_2 = (x - x_1)$, $\phi_3 = (x - x_1)(x - x_2)$

Newton's Basis

$$\begin{array}{c|c}
-1 & 8 \\
2 & -1 \\
3 & 4
\end{array}
\quad
\begin{array}{c}
\frac{-1-8}{2+1} = \frac{-9}{3} = -3 \\
\frac{4+1}{3-2} = \frac{5}{1} = 5
\end{array}
\quad
\frac{5+3}{3+1} = \frac{8}{4} = 2$$

$$\begin{aligned}
p(x) &= 8 - 3(x+1) + 2(x+1)(x-2) \\
&= 8 - 3x - 3 + 2(x^2 - x - 2) \\
&= 8 - 3x - 3 + 2x^2 - 2x - 4 \\
&= 8 - 5x + 2x^2 - 3 - 4 \\
&= 1 - 5x + 2x^2
\end{aligned}$$

$c_1 = 1$, $c_2 = -5$, $c_3 = 2$

Interpolation (Newton):

$\boxed{p(x) = 1 - 5x + 2x^2}$

III) $D_3 = \{(-1, 8), (2, -1), (3, 4)\}$

$$\underbrace{\begin{aligned} \ell_1(x) &= (x-x_2)(x-x_3) / (x_1-x_2)(x_1-x_3) \\ \ell_2(x) &= (x-x_1)(x-x_3) / (x_2-x_1)(x_2-x_3) \\ \ell_3(x) &= (x-x_1)(x-x_2) / (x_3-x_1)(x_3-x_2) \end{aligned}}_{\text{Lagrange Basis}}$$

$$\ell_1(x) = \frac{(x-2)(x-3)}{(-1-2)(-1-3)} = \frac{(x-2)(x-3)}{12} = \tfrac{1}{12}(x-2)(x-3)$$

$$\ell_2(x) = \frac{(x+1)(x-3)}{(2+1)(2-3)} = \frac{(x+1)(x-3)}{-3} = -\tfrac{1}{3}(x+1)(x-3)$$

$$\ell_3(x) = \frac{(x+1)(x-2)}{(3+1)(3-2)} = \frac{(x+1)(x-2)}{4} = \tfrac{1}{4}(x+1)(x-2)$$

$$\begin{aligned} p(x) &= y_1\ell_1(x) + y_2\ell_2(x) + y_3\ell_3(x) \\ &= 8\,\ell_1(x) - 1\,\ell_2(x) + 4\,\ell_3(x) \\ &= 8(\tfrac{1}{12}(x-2)(x-3)) - 1(-\tfrac{1}{3}(x+1)(x-3)) + 4(\tfrac{1}{4}(x+1)(x-2)) \\ &= 8(\tfrac{1}{12}(x^2-5x+6)) - 1(-\tfrac{1}{3}(x^2-2x-3)) + 4(\tfrac{1}{4}(x^2-x-2)) \\ &= 8(\tfrac{1}{12}x^2 - \tfrac{5}{12}x + \tfrac{1}{2}) - 1(-\tfrac{1}{3}x^2 + \tfrac{2}{3}x + 1) + 4(\tfrac{1}{4}x^2 - \tfrac{1}{4}x - \tfrac{1}{2}) \\ &= \tfrac{2}{3}x^2 - \tfrac{10}{3}x + 4 + \tfrac{1}{3}x^2 - \tfrac{2}{3}x - 1 + x^2 - x - 2 \\ &= \tfrac{2}{3}x^2 + \tfrac{1}{3}x^2 + x^2 - \tfrac{10}{3}x - \tfrac{2}{3}x - x + 4 - 1 - 2 \\ &= 2x^2 - 5x + 1 \end{aligned}$$

$c_1 = 1,\ c_2 = -5,\ c_3 = 2$

Interpolation (Langrange):

$$\boxed{p(x) = 1 - 5x + 2x^2}$$

3) For question 3, we were given two MATLAB functions *interpnewt.m* and *hornernewt.m*. The function *interpnewt.m* uses the Newton approach to find the divided differences to compute the expansion coefficients for the interpolating polynomials. The function *hornernewt.m* uses the Horner method to form a polynomial by the coefficients *c* and shifts *x*. Both will be used to solve polynomials which interpolate data sets. We were given two data sets that which was the following: *-4 + 8 * (j - 1)/(n - 1)* (uniformly space points) and *4*cos((pi*(2*j - 1))/(2*n))* (Chebyshev points). We were then tasked to plot the graph of the given equation against the set of interpolation points given with the *interpnewt.m* and *hornernewt.m* MATLAB functions. First, I plotted the first equation, which shows the *c* coefficients found and where we need to find the interpolant at and the interpolant of the coefficients. The graph seems to be reflected along the y-axis and it is very accurate with the interpolant found. The second graph gives us the interpolant of the second equations along with the data points marked by the red x. In both graphs, I observe that both interpnewt.*m* and *hornernewt.m* MATLAB functions used will return accurate interpolants when trying to find the interpolant for these functions. Numerically I do not observe any precision errors, but I do estimate that they are most likely still exist. Overall, the functions are useful to accurately represent an interpolant of a given function.

Coefficient Values:

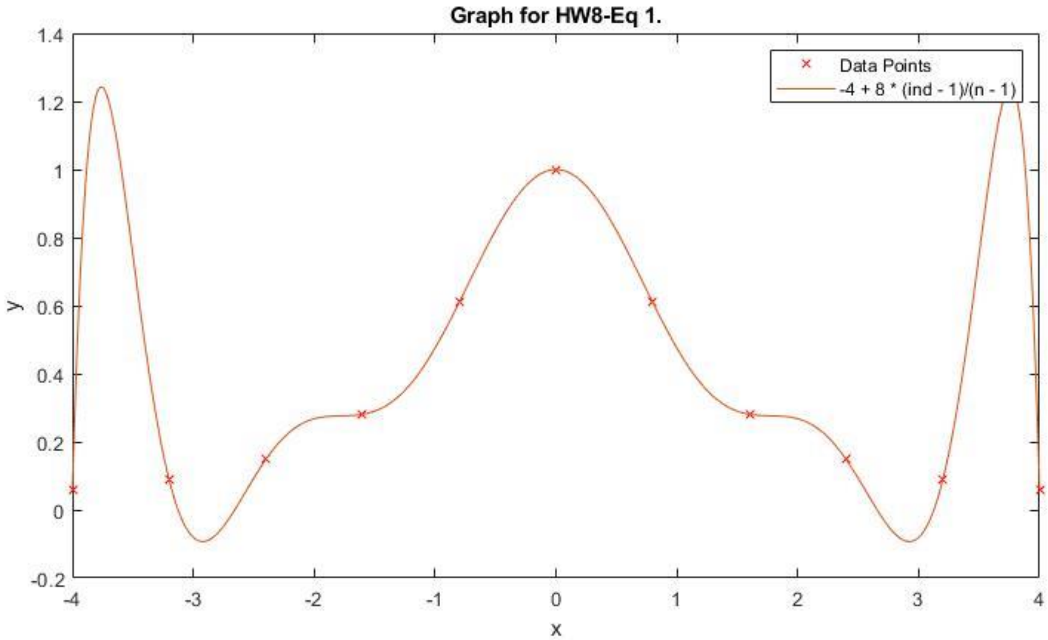
Eq. 1

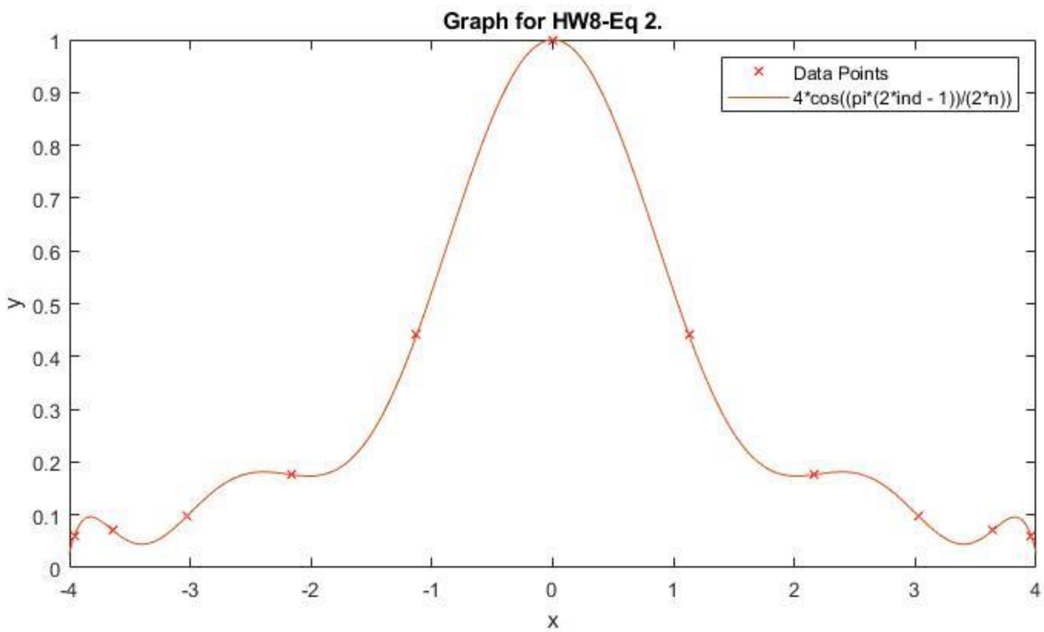
```
Command Window
>> c

c =

    0.0588
    0.0377
    0.0225
    0.0147
    0.0078
   -0.0085
   -0.0006
    0.0028
   -0.0016
    0.0005
   -0.0001

fx >>
```

Eq. 2

```
Command Window
>> HW8_3
>> c

c =

    0.0588
   -0.0940
    0.0019
   -0.0211
   -0.0010
    0.0027
    0.0003
   -0.0003
   -0.0002
   -0.0001
   -0.0000

fx >>
```

Graphs:


Graph for HW8-Eq 1.
Data Points
-4 + 8 * (ind - 1)/(n - 1)
y
x
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
-4
-3
-2
-1
0
1
2
3
4


Graph for HW8-Eq 2.
Data Points
4*cos((pi*(2*ind - 1))/(2*n))
y
x
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-4
-3
-2
-1
0
1
2
3
4

MATLAB Source Code:

HW8_3.m

```
% function HW8_3
% Sets up and solves for HW8, problem 3.
% Get the data and construct the polynomial interpolant.
n = 11;
ind = transpose([1:n]);
y = 1./(1 + x.^2);
% Equation 1.
% x = -4 + 8 * (ind - 1)/(n - 1);
% Equation 2.
x = 4*cos((pi*(2*ind - 1))/(2*n));
c = interpnewt(x, y);
plot(x, y, 'rx')
hold on
% Plot the interpolant.
z = transpose(linspace(-4, 4, 500));
p = hornernewt(c, x, z);
plot(z, p)
title('Graph for HW8-Eq 1.')
xlabel('x')
ylabel('y')
legend('Data Points', '-4 + 8 * (ind - 1)/(n - 1)')
```

hornernewt.m

```
function p = hornernewt(c,x,z)
% function p = hornernewt(c,x,z)
% Uses Horner method to evaluate in nested form a polynomial defined
% by coefficients c and shifts x. Polynomial is evaluated at z.
n = length(c); % 1 + degree of polynomial.
p = c(n);
for k = n-1:-1:1
    p = p.*(z - x(k)) + c(k);
end
```

interpnewt.m

```
function c = interpnewt(x, y)
% function c = interpnewt(x,y)
% computes coefficients c of Newton interpolant through
% (x_k, y_k), k = 1:length(x)
n = length(x);
for k = 1:n-1
    y(k+1:n) = (y(k+1:n) - y(k))./(x(k + 1:n) - x(k));
end
c = y;
```