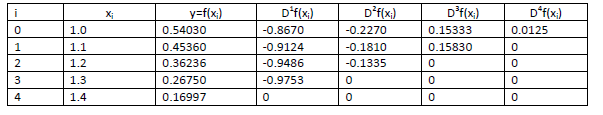
**Lab 4: Interpolation using Divided Difference and Newton’s Formula**

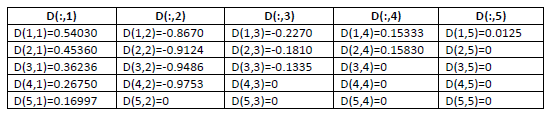
1. As a starting example we will construct the divided difference table as given in lecture slides for the following data points x=[1 1.1 1.2 1.3 1.4] and y=[0.5403 0.45360 0.36236 0.26750 0.16997]. The divided difference table for these data points is given below:



2. In order to construct the Newton polynomial in MATLAB, we would want to first construct the divided difference table. We can do this by storing the values in the rows of a 5 x 5 matrix D.

The **first column** of D, referenced in MATLAB as **D( : ,1)**, will store the function values at the interpolating points. The **second** column of D -- D(:, 2) -- will store the **first** divided differences. The **third** column of D -- D(:, 3) -- will store the **second** divided differences. The **fourth** column of D -- D(:, 4) -- will store the **third** divided differences. The **fifth** column of D -- D(:, 5) -- will store the **fourth** divided difference.

The entries in the matrix D will be:



3. Create a 5x5 matrix D initially with all zeros:

>> D = zeros(5,5);

4. Set up the vector X and Y with the x-coordinates of the interpolating values:

>> X=[1 1.1 1.2 1.3 1.4];

>> Y=[0.5403 0.45360 0.36236 0.26750 0.16997];

These entries will be stored as:

For X as:



If you run this on Matlab command window

>>X(3)

ans=1.2

>>X(1:3)

ans = 1 1.1 1.2

And for Y as:



5. Now start computing the divide differences column by column for the matrix D The first column is just the values of the function at the interpolating points, stored in Y:

» D(:,1) = Y;

6. We next work on the **second column of D** -- starting in first row ( D(1,2) ) and working down to fourth row: >> D(1,2) = (D(2,1)-D(1,1))/(X(2)-X(1));

>> D(2,2) = (D(3,1)-D(2,1))/(X(3)-X(2));

>> D(3,2) = (D(4,1)-D(3,1))/(X(4)-X(3));

>> D(4,2) = (D(5,1)-D(4,1))/(X(5)-X(4));

7. Fill the remaining column by using the following commands:

>> D(1,3) = (D(2,2)-D(1,2))/(X(3)-X(1));

>> D(2,3) = (D(3,2)-D(2,2))/(X(4)-X(2));

>> D(3,3) = (D(4,2)-D(3,2))/(X(5)-X(3));

>> D(1,4) = (D(2,3)-D(1,3))/(X(4)-X(1));

>> D(2,4) = (D(3,3)-D(2,3))/(X(5)-X(2));

>> D(1,5) = (D(2,4)-D(1,4))/(X(5)-X(1));

The final matrix D will have the following form:

>>D

D =

0.5403 -0.8670 -0.2270 0.1533 0.0125

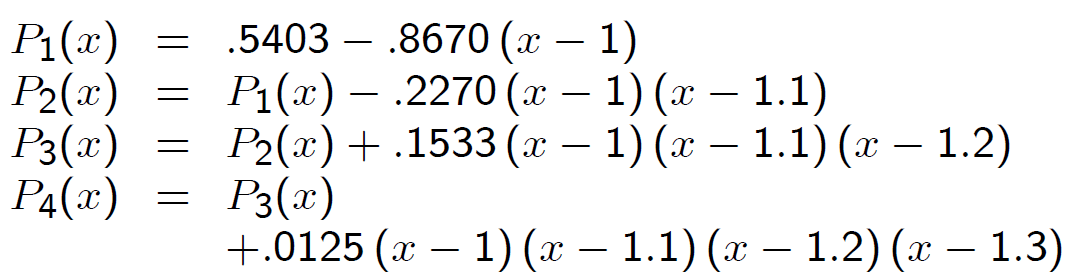
0.4536 -0.9124 -0.1810 0.1583 0

0.3624 -0.9486 -0.1335 0 0

0.2675 -0.9753 0 0 0

0.1700 0 0 0 0

8. Once the ‘D’ Matrix has been created, the Newton polynomials of degrees 1 through 4 which are:



Can be constructed recursively in Matlab as follows:

>> P1 = [0 D(1,1)] + D(1,2)\*poly(X(1))

P1 =

-0.8670 1.4073

Similarly, higher polynomials can be constructed