EE:211

Computational Techniques in Electrical Engineering

Lab#2(II)

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:

i	x_i	$y=f(x_i)$	$D^1f(x_i)$	$D^2f(x_i)$	$D^3f(x_i)$	$D^4f(x_i)$
0	1.0	0.54030	-0.8670	-0.2270	0.15333	0.0125
1	1.1	0.45360	-0.9124	-0.1810	0.15830	0
2	1.2	0.36236	-0.9486	-0.1335	0	0
3	1.3	0.26750	-0.9753	0	0	0
4	1.4	0.16997	0	0	0	0

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:

D(:,1)	**D(:,2)**	**D(:,3)**	**D(:,4)**	**D(:,5)**
D(1,1)=0.54030	D(1,2)=-0.8670	D(1,3)=-0.2270	D(1,4)=0.15333	D(1,5)=0.0125
D(2,1)=0.45360	D(2,2)=-0.9124	D(2,3)=-0.1810	D(2,4)=0.15830	D(2,5)=0
D(3,1)=0.36236	D(3,2)=-0.9486	D(3,3)=-0.1335	D(3,4)=0	D(3,5)=0
D(4,1)=0.26750	D(4,2)=-0.9753	D(4,3)=0	D(4,4)=0	D(4,5)=0
D(5,1)=0.16997	D(5,2)=0	D(5,3)=0	D(5,4)=0	D(5,5)=0

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 enetries will be stored as:

For X as:

X(1)=1	X(2)=1.1	X(3)=1.2	X(4)=1.3	X(5)=1.4

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:

Y(1)=0.5403	Y(2)=0.45360	Y(3)=0.36236	Y(4)=0.26750	Y(5)=0.16997

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. We can now construct the Newton Polynomials of degrees 1 through 4 recursively as follows:

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

P1 =

  -0.8670    1.4073
```

And also you can go to higher polynomials like this.