INDIAN INSTITUTE OF TECHNOLOGY GANDHINAGAR

MA 202: MATHEMATICS - IV
Semester–II, Academic Year 2022-23

Tutorial Set -3
Question - 7

By

Kush Patel
[20110131]

➔ In question 7, we are asked to estimate the value of temperature at distance equal to 3.2 and 2.7 m by nth order polynomial interpolation. I have used the Lagrange Interpolation method. For nth order polynomial interpolation, we need (n+1) data points (x_1,y_1),...., (x_{n+1},y_{n+1}) We are given input data points as shown below.

x	T (^{0}C)
0	100
2	85
4	70
6	55
8	40

- ➔ The idea behind the interpolation is to interpret a polynomial which takes on certain values at arbitrary points. Approximation for Lagrange Interpolation method, as we know, is given by:

<u>Lagrange Polynomial:</u>

$$p(x) = L_1(x)y_1 + L_2(x)y_2 + L_3(x)y_3 + \ldots L_N(x)y_N$$

$$L_k(x) = \frac{(x-x_1)(x-x_2)\ldots(x-x_{k-1})(x-x_{k+1})\ldots(x-x_N)}{(x_k-x_1)(x_k-x_2)\ldots(x_k-x_{k-1})(x_k-x_{k+1})\ldots(x_k-x_N)}$$

Numerator of $L_k(x)$ is product of all $(x - x_i)$ except for $(x - x_k)$
Denominator of $L_k(x)$ is product of all $(x_k - x_i)$ except for $(x_k - x_k)$

- ➔ Here, we need to estimate the both values for 1 to nth order polynomial interpolation. I built a function called Interpolation(N,x) that can take two inputs as previously. One is the value of n in which order of polynomial we need to estimate and the second is x(value of distance where we need to estimate the value of temperature.)

- ➔ To run this function and get the output, I ran a loop by varying the value of n(order) and gave the command to find the required values and printed it.

```matlab
for z=1:4
    a = Interpolation(z,3.2);
    disp('The value of T for x=3.2 and for order')
    disp(z)
    disp('is ')
     disp(a)
    b = Interpolation(z,2.7);
    disp('The value of T for x=2.7 and for order')
    disp(z)
    disp('is ')
    disp(b)
end

function v = Interpolation(N,t)
format('long')
x = [0,2,4,6,8];
n = size(x);
n = n(2);
T = [100,85,70,55,40];
L = zeros(1,n);
p=1;
q=1;
for i=1:N
    for j=1:N
        if i == j
            continue;
        else
            p = p * (t - x(j));
```

```matlab
p=1;
q=1;
for i=1:N
    for j=1:N
        if i == j
            continue;
        else
            p = p * (t - x(j));
            q = q * (x(i) -x(j));
        end
    end
    L(1,i) = T(i) * (p/q);
    p=1;
    q=1;
end
v=0;
for k=1:N
    v = v + L(1,k);
end
plot(x,T,'-s', 'MarkerSize',10,...
    'MarkerEdgeColor','red',...
    'MarkerFaceColor',[1 .6 .6])
end
```

```
>> T7_20110131
The value of T for x=3.2 and for order
     1


is
   100

The value of T for x=2.7 and for order
     1


is
   100

The value of T for x=3.2 and for order
     2


is
    76

The value of T for x=2.7 and for order
     2


is
  79.750000000000000

The value of T for x=3.2 and for order
     3
```

➔ We can easily see through these images that as the value of n is increasing, the resultant value becomes more accurate and goes towards the correct value. For n=1 (Linear interpolation), the value of temperature comes out 100 but for n=4, the value of temperature is 76 C(for x=3.2) and 79.75 C(for x =2.7).

➔ The points are also plotted in the graph below.

```
The value of T for x=3.2 and for order
     3

is
    76

The value of T for x=2.7 and for order
     3

is
  79.750000000000000

The value of T for x=3.2 and for order
     4

is
    76

The value of T for x=2.7 and for order
     4

is
  79.750000000000000
```