DELHI TECHNOLOGICAL UNIVERSITY

SCIENTIFIC COMPUTING (MC-204)

PRACTICAL FILE

SUBMITTED TO:
PROF. DINESH UDAR
MR. ANKIT SHARMA

SUBMITTED BY:
NITYA MITTAL
(2K19/MC/089)

EXPERIMENT 4

AIM

Write the MATLAB program of Newton Forward and Backward interpolation table of any data set.
For example:-
x = [3, 8, 13, 18, 23, 28]
y=[7, 18, 26, 37, 48, 57]

THEORY

NEWTON FORWARD METHOD

The differences y1 – y0, y2 – y1, y3 – y2, ……, yn – yn–1 when denoted by dy0, dy1, dy2, ……, dyn–1 are respectively, called the first forward differences. Thus the first forward differences are : $\Delta Y_r = Y_{r+1} - Y_r$

Forward difference table

x	y	Δy	$\Delta^2 y$	$\Delta^3 y$	$\Delta^4 y$	$\Delta^5 y$
x_0	y_0					
		Δy_0				
x_1 $(= x_0 + h)$	y_1		$\Delta^2 y_0$			
		Δy_1		$\Delta^3 y_0$		
x_2 $(= x_0 + 2h)$	y_2		$\Delta^2 y_1$		$\Delta^4 y_0$	
		Δy_2		$\Delta^3 y_1$		$\Delta^5 y_0$
x_3 $= (x_0 + 3h)$	y_3		$\Delta^2 y_2$		$\Delta^4 y_1$	
		Δy_3		$\Delta^3 y_2$		
x_4 $= (x_0 + 4h)$	y_4		$\Delta^2 y_3$			
		Δy_4				
x_5 $= (x_0 + 5h)$	y_5					

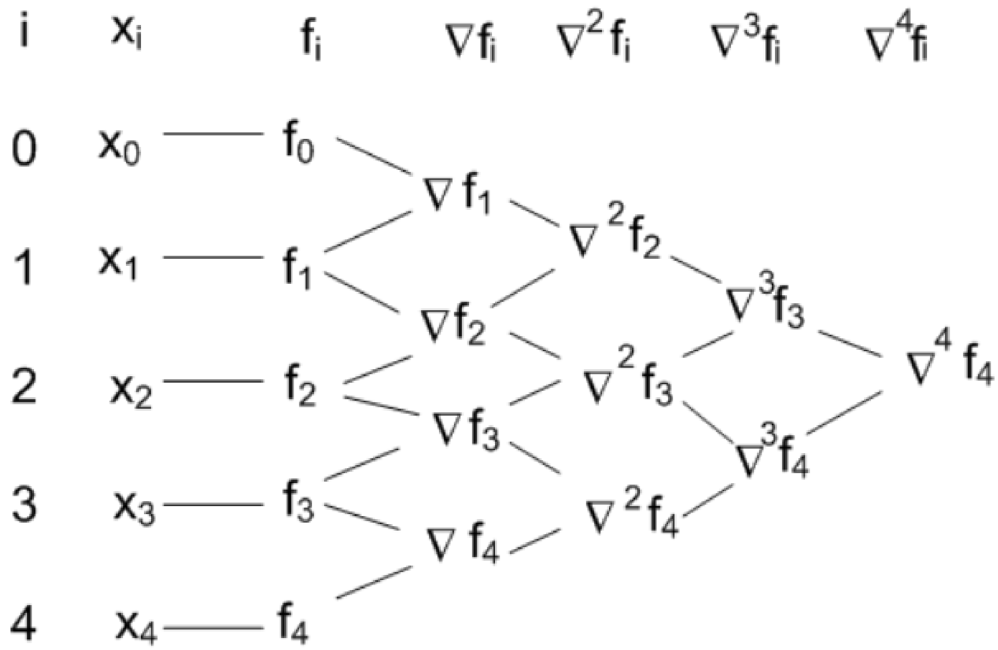
NEWTON'S GREGORY FORWARD INTERPOLATION FORMULA :

$$f(a+hu) = f(a)+u\Delta f(a)+\frac{u(u-1)}{2!}\Delta^2 f(a)+\ldots+\frac{u(u-1)(u-2)\ldots(u-n+1)}{n!}\Delta^n f(a)$$

This formula is particularly useful for interpolating the values of f(x) near the beginning of the set of values given. h is called the interval of difference and **u = (x – a) / h**, Here a is first term.

NEWTON BACKWARD METHOD

The differences y1 – y0, y2 – y1, y3 – y2, ……, yn – yn–1 when denoted by dy0, dy1, dy2, ……, dyn–1 are respectively, called the first backward differences. Thus the first backward differences are :



Newton Gregory Backward Interpolation Formula:

$$P(x_n+ph) = y_0 + p\nabla y_n + \frac{p(p+1)}{2!}\nabla^2 y_n + \frac{p(p+1)(p+2)}{3!}\nabla^3 y_n + \Lambda + \frac{p(p+1)(p+2)\ldots(p+n-1)}{n!}\nabla^n y_n$$

This formula is particularly useful for interpolating the values of f(x) near the end of the set of values given. h is called the interval of difference and **$p = (x - x_n) / h$**, Here x_n is first term.

SOURCE CODE

1. Newton Forward Method

```
clear all;
close all;
%sample data set taken
x=[3,8,13,18,23,28];
y=[7,18,26,37,48,57];
size=6;
```

```
%finding out different degrees of del for the function
delf(1,:)=y(:);
for t = 2:size+1
    for i = 1:size-t+1
        delf(t,i)= delf(t-1,i+1)-delf(t-1,i);
    end
end


%let the number be 10
%standard h=10 and num= a + hu
num=10;
h=10;
pos=0;
%finding the valu
for k = 1:size
    if (num>=x(k)) && (num<x(k+1))
        a=x(k);
        pos=k;
    end
end


u=(num-a)/h;


%finding f(x)
sum=0;
for i = 1:size
    multi=1;
    if(i>1)
        for j = 0:i-2
            multi= multi*(u-j);
        end
    end
    sum = sum + (multi*delf(i,pos)/factorial(i-1));
end


disp("The value of f(x) for x= 10 using Newton's Forward Method")
disp(sum)
```

2. Newton Backward Method

```
clear all;
close all;
%sample data set taken
x=[3,8,14,19,23,29];
y=[7,11,30,40,47,60];
size=6;


%let the number be 26
%standard h=10 and num= a + hu
num=26;
h=10;
pos=0;




%finding out different degrees of del for the function
delf(1,:)=y(:);
for t = 2:size+1
    for i = 1:size-t+1
        delf(t,i)= delf(t-1,i+1)-delf(t-1,i);
    end
end


%finding the value of a
for k = 2:size
    if (num>=x(k-1)) && (num<x(k))
        a=x(k);
        pos=k;
    end
end


%finding out different degrees of nebula for the function
nebf= zeros(size);


for i =1:size
    for j=1:size
        if (i+j==k+1)
            nebf(i)=delf(i,j);
        end
    end
end


u=(num-a)/h;
```

```
%finding f(x)
sum=0;
for i = 1:size
    multi=1;
    if(i>1)
        for j = 0:i-2
            multi= multi*(u+j);
        end
    end
    sum = sum + (multi*nebf(i)/factorial(i-1));
end


disp("The value of f(x) for x= 26 using Newton's Backward Method")
disp(sum)
```

OUTPUT

```
>> newton_forward
The value of f(x) for x= 10 using Newton's Forward Method
    19.1824

>> newton_backward
The value of f(x) for x= 26 using Newton's Backward Method
    55.6165
```