Id- 202218061

Name - Jatan sahu

ASSIGNMENT 03

1. Write a function which implements the Lagrangian interpolation method for m order polynomials. For obtaining the m order interpolating polynomial, you require the function evaluation of m+1 points.

2. Consider the function y= log 2(x) . Evaluate the function on four points x0 = 1, x1 = 4, x2 = 8 and x3 = 256 and obtain the cubic interpolation g(x) using your code. Find the value of your estimated g(x) at x = 16. Compute the error and compare it from the error obtained by Newton Divided Difference methods

```
#Importing library for operation
import math
```

```
def lagrangian():
  X=list(map(float,input("Enter values of X in list formate seprated by spaces :").split(" ")))
  Y=[math.log2(i) for i in X]
  x=float(input("Enter value for extimation :"))

  est=0
  for i in range(len(X)):
    n=1                       #Numerator
    d=1                       #Denominator
    for j in range(len(X)):
      if i!=j:
        n=(x - X[j]) * n
        d=(X[i] - X[j]) * d

    est = ((n*Y[i])/d)  + est
  #Printing answer
  print(f"\nOrginal value of log 2({x}) is {math.log2(x)}")
  print(f"\nEstimated value of log 2({x}) is :")
  return (est)
```

2. Consider the function y= log 2(x) . Evaluate the function on four points x0 = 1, x1 = 4, x2 = 8 and x3 = 256 and obtain the cubic interpolation g(x) using your code. Find the value of your estimated g(x) at x = 16. Compute the error and compare it from the error obtained by Newton Divided Difference methods

Using Langrages interpolation

```
est = lagrangian()
est
```

```
Enter values of X in list formate seprated by spaces :1 4 8 256
Enter value for extimation :16

Orginal value of log 2(16.0) is 4.0

Estimated value of log 2(16.0) is :
0.3833017077798871
```

```
#Calculating error in Langrages interpolation
ov=math.log2(16)
# error = (est - ov)
error = abs((ov - est)/ov)*100
print(f"error = {error}%")
```

```
error = 90.41745730550282%
```

Using Newton Divind difference

```
def Ninterpolation():
  X=list(map(float,input("Enter values of X in list formated seprated by spaces :").split(" ")))
  Y=[math.log2(i) for i in X]

```

```
#Checking if length of X and Y are equal or not
# while len(X)!=len(Y):
#   print("Type correct values of X and Y should be equal ")
#   X=list(map(float,input("Enter values of X in list formated seprated by spaces :").split(" ")))
#   Y=list(map(float,input("Enter values of Y in list formated seprated by spaces :").split(" ")))

x=float(input("Enter value for estimation :"))

#Making Newtons divided difference table
#Initialize m*n empty matrix
m=len(X)
n=len(X)+1
mat=[]
for i in range(m):
  a=[]
  for j in range(n):
    a.append(None)
  mat.append(a)

#Filling the values of X and Y in column 1 and 2
for i in range(m):
  mat[i][0]=X[i]
  mat[i][1]=Y[i]

#Calculating other columns
for i in range(2,n):
  for j in range(0,n-i):
    mat[j][i]=(mat[j+1][i-1] - mat[j][i-1])/(mat[i+j-1][0] - mat[j][0])

#Putting values in formula f(x) = y + (x-x0)y' +(x-x0)(x-x1)y'' +...
e=0
c=1
for i in range(1,n):
  e =  e + (mat[0][i])*c
  c =  c*(x - X[i-1])

 #printing answer
print(f"\nEstimated value of f({x}) is :")
return e

```

```
e=Ninterpolation()
e
```

```
Enter values of X in list formated seprated by spaces :1 4 8 256
Enter value for estimation :16

Estimated value of f(16.0) is :
-0.38330170777988537
```

```
#Calculating error in Newton divided diff
ov=math.log2(16)
error = abs((ov - e)/ov)*100
print(f"error = {error}%")
```

```
error = 109.58254269449714%
```

Comparing errors in Lagranges and Newton Divided diffence

As we see above errors in both methods are almost same

```
1
```

```
#
```

```
1
```

✓ 0s completed at 2:17 AM