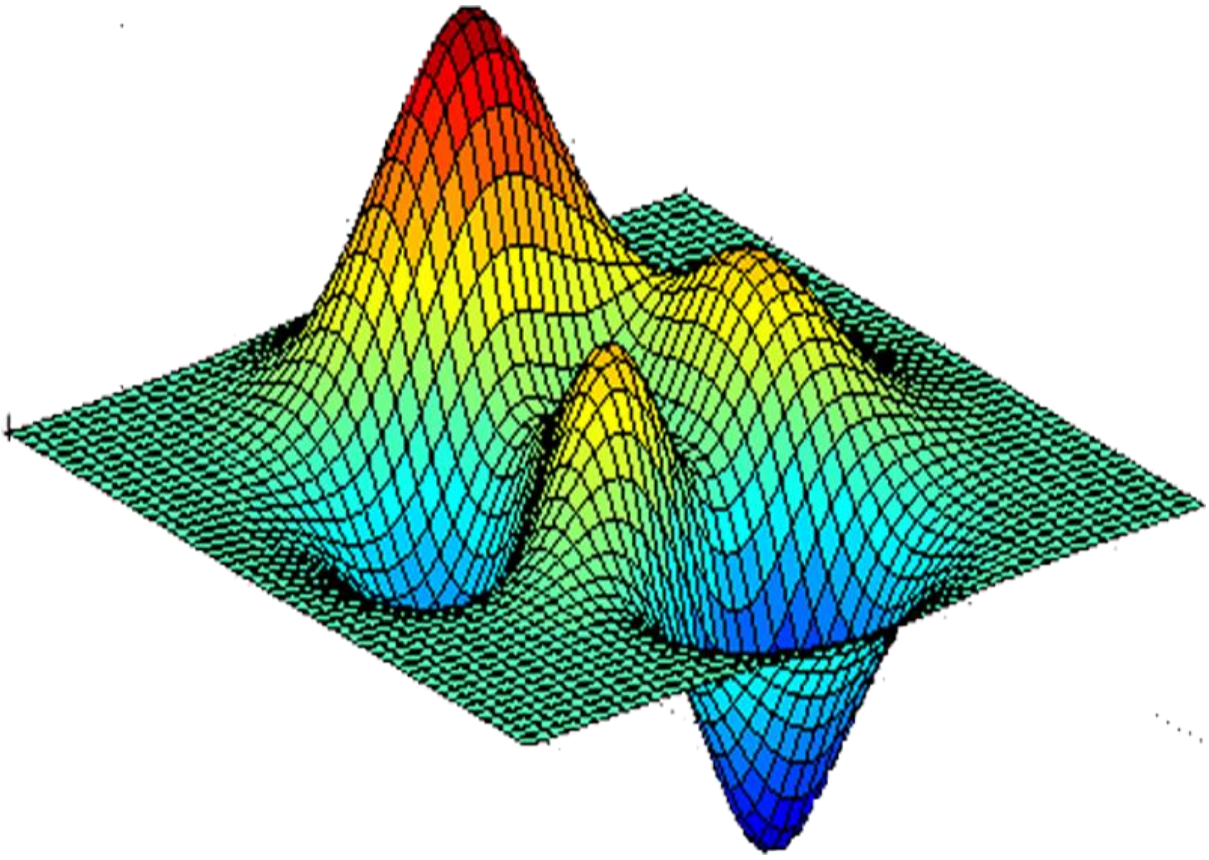
Numerical Computing

Langrange’s Polynomial and NDDIP



QASID AHMED ALEEM

COURSE TEACHER: DR. SHAHID QURESHI
BSC(HONOURS)
H-2 (MORNING)
INSTITUTE OF SPACE AND PLANETARY ASTROPHYSICS
UNIVERSITY OF KARACHI

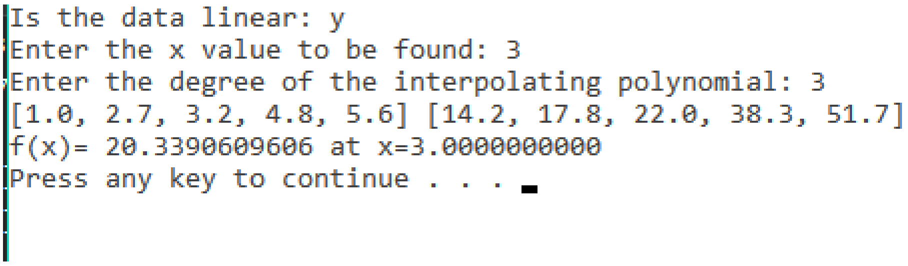
Tasks:

Task 1: Langrange's Polynomial

Code:

```
x=[3.2,2.7,1.0,4.8,5.6]
y=[22.0,17.8,14.2,38.3,51.7]

#x=[1.2,1.6,2.1,2.9]
#y=[1.46,2.52,4.4,8.8]
z=0
q=input("Is the data linear: ")
quest=float(input("Enter the x value to be found: "))
n=int(input("Enter the degree of the interpolating polynomial: "))

if q=="y" or q=="Y" or q=="yes":
    x.sort()
    y.sort()
print(x,y)
for i in range(0,len(x)-1):
    if quest > x[i] and quest <x[i+1]:
        m=(x[i] +x[i+1])/2

        if (quest-m)<0:
            x.pop()
            y.pop()

            for i in range(0,n+1):
                p=1

                for j in range(0,n+1):
                    if j!=i:
                        p *= (quest-x[j])/(x[i]-x[j])

                z+=y[i]*p

        else:
            x.reverse()
            y.reverse()
            x.pop()
            y.pop()
            x.reverse()
            y.reverse()

            for i in range(0,n+1):
                p=1


                for j in range(0,n+1):
                    if j!=i:
                        p *= (quest-x[j])/(x[i]-x[j])

                z+=y[i]*p
print("f(x)={: 12.10f} at x={:12.10f}".format(z,quest))
```

Result:



```
Is the data linear: y
Enter the x value to be found: 3
Enter the degree of the interpolating polynomial: 3
[1.0, 2.7, 3.2, 4.8, 5.6] [14.2, 17.8, 22.0, 38.3, 51.7]
f(x)= 20.3390609606 at x=3.0000000000
Press any key to continue . . .
```

Excel work:

x_val	**x**	**y**	**f0*(x_val-x[j]) \(x[i]-x[j])**	**f1*(x_val-x[j]) \(x[i]-x[j])**	**f2*(x_val-x[j]) \(x[i]-x[j])**	**f3*(x_val-x[j]) \(x[i]-x[j])**	**Ans**
3	1	14.2	-0.107909	7.1798319	13.500000	-0.359962406	20.211961
	2.7	17.8					
	3.2	22					
	4.8	38.3					
	5.6	51.7					

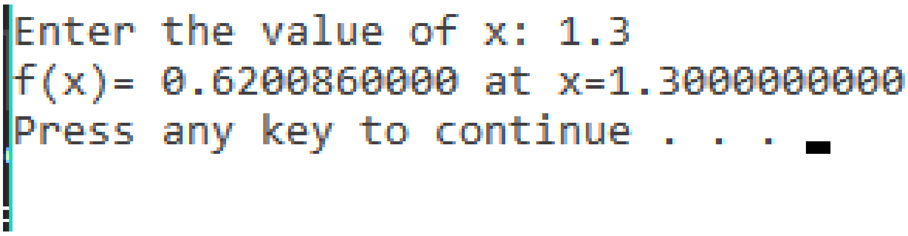
Task 2: NDDIP

Code:

```python
import numpy as np

def coef(x, y):
    #x : array of data points
    #y : array of f(x)

    n = len(x)
    a = []
    for i in range(n):
        a.append(y[i])

    for j in range(1, n):

        for i in range(n-1, j-1, -1):
            a[i] = float(a[i]-a[i-1])/float(x[i]-x[i-j])

    return np.array(a) # return an array of coefficient
```

```python

def Eval(a, x, r):



    n = len( a ) - 1
    temp = a[n]
    for i in range( n - 1, -1, -1 ):
        temp = temp * ( r - x[i] ) + a[i]
    return temp # return the y_value interpolation


x=[1.0,1.3,1.6,1.9,2.2]
y=[0.7661977,0.6200860,0.4554022,0.2818186,0.1103623]
quest=float(input("Enter the value of x: "))

print("f(x)={: 12.10f} at x={:12.10f}".format(Eval(coef(x,y),x,quest)\
,quest))
```

Result:



```
Enter the value of x: 1.3
f(x)= 0.6200860000 at x=1.3000000000
Press any key to continue . . .
```