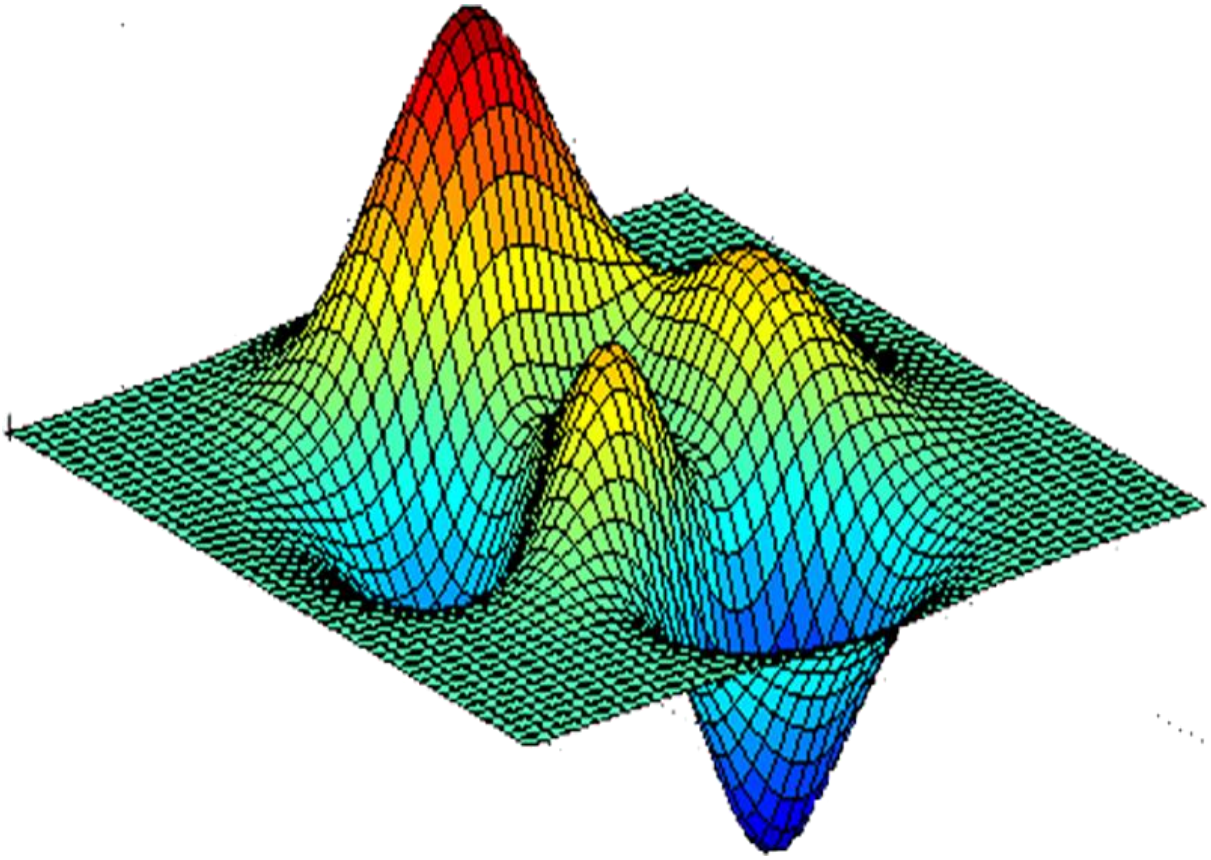
Numerical Computing

NFDIP and Bezier Curve



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:Newton's Forward Difference

Python code:

```
from math import factorial
import numpy as np
x=[0.0,0.2,0.4,0.6,0.8,1.0,1.2,1.4]
y=[3.1,4.9,6.2,7.5,8.9,11.2,10.1,7.3]
s=0
quest=float(input("Enter the value of x: "))
for i in range(0,len(x)-1):
     if quest>=x[i] and quest<=x[i+1]:
         h=x[i+1]-x[i]
         m=(x[i-1]+x[i+2])/2

         m1=(x[i]+x[i+3])/2

         if abs(quest-m) <abs(quest-m1):
             s=(quest-x[i-1])/h
             y1=[]
             for t in range(0,4):
                 y1.append(y[i-1+t])
         elif abs(quest-m) >abs(quest-m1):

             s=(quest-x[i])/h
             y1=[]
             for t in range(0,4):

                 y1.append(y[i+t])



def coef(y):
    n=len(y)
    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])
    return np.array(a)

def Eval(a,s):

    n=len(a)-1
    f=n

    temp=0


    for i in range(n,-1,-1):
        y0=1
        for j in range(0,f):

            y0=(s-j)*y0



```

```
        temp=(y0*(a[i]))/factorial(i)+temp

        f=f-1
    return temp

print("f(x) = {:12.10f} at x ={:5.2}".format(Eval(coef(y1),s),quest))
```

Output:

```
Enter the value of x: 0.75
f(x) = 8.4968750000 at x = 0.75
Press any key to continue . . .
```

Excel work:

x-val	i	x	y	▲y	▲2y	▲3y	s	2nd term	3rd term	4th term	f(x)
0.75	0	0.0	3.1				1.75	2.275	0.065625	-0.04375	8.496875
	1	0.2	4.9								
	2	0.4	6.2								
	3	0.6	7.5	1.3							
	4	0.8	8.9	1.4	0.1						
	5	1.0	11.2	2.3	0.9	0.8					
	6	1.2	10.1								
	7	1.4	7.3								

Task 2: Bezier Curve

Python code:

```
from math import factorial as fac
import matplotlib.pyplot as plt
import numpy as np
x=[]
y=[]
deg=int(input("Input the order of the curve:"))
for i in range(0,deg+1):
    xu=int(input("Enter the {:2s} cordinate:".format("x"+str(i))))
    yu=int(input("Enter the {:2s} cordinate:".format("y"+str(i))))
    y.append(yu)
    x.append(xu)
def comb(n,k):
    y=fac(n)/(fac(n-k)*fac(k))
    return y

def binom(x,y,n):
    mat=[]
    for i in range(0,n+1):

        d=comb(n,i)*(x**(n-i))*(y**i)

        mat.append(d)
    return mat

def bezier(point,n):
    bez=[]
```

```
    m=0
    for i in np.linspace(0.0,1.0,num=100):

        m=0
        for r in range(0,n+1):
            k=binom(1-i,i,n)


            m+=k[r]*point[r]

        bez.append(m)


    return bez

b1=(bezier(x,deg))
b2=(bezier(y,deg))


plt.plot(b1,b2,'-')
plt.gca().invert_yaxis()
plt.grid(1)
plt.show()

```

Output:

```
Input the order of the curve:3
Enter the x0 cordinate:120
Enter the y0 cordinate:160
Enter the x1 cordinate:35
Enter the y1 cordinate:200
Enter the x2 cordinate:220
Enter the y2 cordinate:260
Enter the x3 cordinate:220
Enter the y3 cordinate:40
```