**NC Lab # 06**

**Sabaun Arif CID: 110025**

**SID: 10829**

**Task # 01**

**Code:**

#cos(x)-1.3(x)-0

import math as m

import sympy as s

def f(x):

return m.cos(x)-1.3\*x

def mullersmethod(x0,x1,x2):

for i in range (4):

d1=f(x0)-f(x2)

d2=f(x1)-f(x2)

c=f(x1)

h1=x0-x2

h2=x1-x2

a=((h1\*d2)-(h2\*d1))/((h1\*h2)\*(x1-x0))

b=((h1\*\*2)\*d2-(h2\*\*2)\*d1)/((x0-x1)\*(h2\*h1))

if(b>0):

xr=x1-(2\*c)/(b+m.sqrt(b\*\*2-4\*a\*c))

else:

xr=x1-(2\*c)/(b-m.sqrt(b\*\*2-4\*a\*c))

if(x1>xr):

x2=x1

x1=xr

tolr=abs(x2-x1)

print("Iterations: ",i, "\t root : ",xr, "\t\t Tolerance : ",tolr)

else:

x0=x1

x1=xr

tolr=abs(x1-x0)

print("Iterations: ",i, "\t root : ",xr, "\t\t Tolerance : ",tolr)

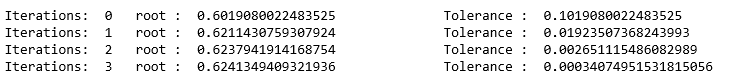
x0=0

x1=0.5

x2=1

mullersmethod(x0,x1,x2)

**Output:**

****

**Task # 02**

**Code:**

#xcos(x)-2x^2+3x-1=0

import math as m

import sympy as s

def f(x):

return x\*m.cos(x)-2\*x\*\*2+3\*x-1

def mullersmethod(x0,x1,x2):

for i in range (8):

d1=f(x0)-f(x2)

d2=f(x1)-f(x2)

c=f(x1)

h1=x0-x2

h2=x1-x2

a=((h1\*d2)-(h2\*d1))/((h1\*h2)\*(x1-x0))

b=((h1\*\*2)\*d2-(h2\*\*2)\*d1)/((x0-x1)\*(h2\*h1))

if(b>0):

xr=x1-(2\*c)/(b+m.sqrt(b\*\*2-4\*a\*c))

else:

xr=x1-(2\*c)/(b-m.sqrt(b\*\*2-4\*a\*c))

if(x1>xr):

x2=x1

x1=xr

tolr=abs(x2-x1)

print("Iterations: ",i, "\t root : ",xr, "\t\t Tolerance : ",tolr)

else:

x0=x1

x1=xr

tolr=abs(x1-x0)

print("Iterations: ",i, "\t root : ",xr, "\t\t Tolerance : ",tolr)

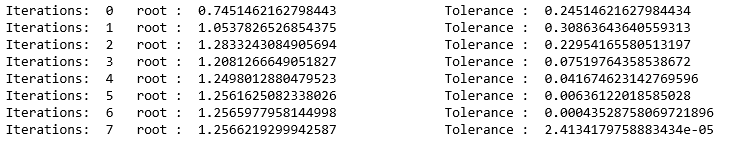
x0=0

x1=0.5

x2=1

mullersmethod(x0,x1,x2)

**Output:**

****