**FOURTH ORDER**

*#RK Method (4RTH ORDER )*

**import matplotlib.pyplot as plt**

**[xo,xn,yo,h]=[0,4,1,0.5]**

**n=(xn-xo)/h**

*#f(x.y)=dy/dx (i.e derivative/slope of original; function)*

**print("RK-METHOD 4rth ORDER")**

**def f1(x,y):**

**return -2\*(x)\*\*3+12\*(x)\*\*2-20\*(x)+8.5**

*#f(x,y)=y(i.e integral of f1 which is original function)*

**def f2(x,y):**

**return (-0.5\*x\*\*4+4\*x\*\*3-10\*x\*\*2+8.5\*x+1)**

**print("xi \t yi \t k1 \t xi+h \t yi+1 \t y(true)")**

**print("==================================================================")**

**a=[]**

**b=[]**

**c=[]**

**for i in range(0,int(n)+1):**

**yi\_rk4=yo+((1/6)\*(f1(xo,yo)+2\*(f1(xo+h/2,yo+(f1(xo,yo)\*h/2)))+2\*(f1(xo+h/2,yo+(h/2)\*(f1(xo+h/2,yo+(h/2)\*(f1(xo,yo))))))+f1(xo+h,yo+h\*(f1(xo+h/2,yo+(h/2)\*(f1(xo+h/2,yo+(h/2)\*(f1(xo,yo)))))))))\*h**

**print(round(xo,4), ' \t', round(yo,4), '\t',round(f1(xo,yo),4), '\t', round(xo+h,4), '\t',round(yi\_rk4,4), '\t',round(f2(xo,yo),4))**

**a.append(round(xo,2))**

**b.append(round(yo,2))**

**c.append(f2(xo,yo))**

**xo=xo+h**

**yo=yi\_rk4**

**print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")**

**print("x=",a,"\ny(euler)=",b,"\ny(true)=",c)**

**plt.plot(a,b,label = "y(euler)=dy/dx")**

**plt.plot(a,c,label = "y(true)=y")**

**plt.title('4RTH ORDER')**

**plt.xlabel('x')**

**plt.ylabel('f(x,y)')**

**plt.legend()**

**OUTPUT**

**RK-METHOD 4rth ORDER**

**xi yi k1 xi+h yi+1 y(true)**

**==================================================================**

**0 1 8.5 0.5 3.2188 1.0**

**0.5 3.2188 1.25 1.0 3.0 3.2188**

**1.0 3.0 -1.5 1.5 2.2188 3.0**

**1.5 2.2188 -1.25 2.0 2.0 2.2188**

**2.0 2.0 0.5 2.5 2.7188 2.0**

**2.5 2.7188 2.25 3.0 4.0 2.7188**

**3.0 4.0 2.5 3.5 4.7188 4.0**

**3.5 4.7188 -0.25 4.0 3.0 4.7188**

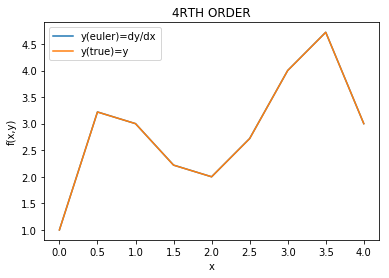
**4.0 3.0 -7.5 4.5 -3.7812 3.0**

**==================================================================**

**x= [0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0]**

**y(4RTH)= [1, 3.22, 3.0, 2.22, 2.0, 2.72, 4.0, 4.72, 3.0]**

**y(true)= [1.0, 3.21875, 3.0, 2.21875, 2.0, 2.71875, 4.0, 4.71875, 3.0]**

****