Lab 8

LAB 10: Solution of ODE of first order and first degree by Runge-Kutta 4th order method and Milne's predictor and corrector method

10.2 Runge-Kutta method

Apply the Runge Kutta method to find the solution of $dy/dx = 1 + (y/x)$ at $y(2)$ taking $h = 0.2$. Given that $y(1) = 2$.

```
from sympy import *
import numpy  as np
def RungeKutta(g,x0,h,y0,xn):

  x,y=symbols('x,y')
  f=lambdify([x,y],g)
  xt=x0+h
  Y=[y0]
  while xt<=xn:
       k1=h*f(x0,y0)
       k2=h*f(x0+h/2, y0+k1/2)
       k3=h*f(x0+h/2, y0+k2/2)
       k4=h*f(x0+h, y0+k3)
       y1=y0+(1/6)*(k1+2*k2+2*k3+k4)
       Y.append(y1)
       #print('y(%3.3f'%xt,') is %3.3f'%y1)
       x0=xt
       y0=y1
       xt=xt+h
  return np.round(Y,2)
RungeKutta('1+(y/x)',1,0.2,2,2)
```

```
array([2.  , 2.62, 3.27, 3.95, 4.66, 5.39])
```

10.3 Milne's predictor and corrector method

Apply Milne's predictor and corrector method to solve $dy/dx = x^2 + (y/2)$ at y(1.4). Given that y(1)=2, y(1.1)=2.2156, y(1.2)=2.4649, y(1.3)=2.7514. Use corrector formula thrice.

```
# Milne's method to solve first order DE
# Use corrector formula thrice
x0=1
y0=2
```

```
y1=2.2156
y2=2.4649
y3=2.7514
h=0.1
x1=x0+h
x2=x1+h
x3=x2+h
x4=x3+h
def f(x,y):
  return x**2+(y/2)

y10=f(x0, y0)
y11=f(x1,y1)
y12=f(x2,y2)
y13=f(x3,y3)
y4p=y0+(4*h/3)*(2*y11-y12+2*y13)
print('predicted value of y4 is %3.3f'%y4p)
y14=f(x4,y4p);
for i in range(1,4):
  y4=y2+(h/3)*(y14+4*y13+y12);
  print('corrected value of y4 after \t iteration  %d  is \t %3.5f\t '%
                                                    (i,y4))
  y14=f(x4,y4);
```

```
predicted value of y4 is 3.079
corrected value of y4 after          iteration  1  is          3.07940
corrected value of y4 after          iteration  2  is          3.07940
corrected value of y4 after          iteration  3  is          3.07940
```

In the next program, function will take all the inputs from the user and display the answer.

Apply Milne's predictor and corrector method to solve $dy/dx = x^2 + (y/2)$ at y(1.4). Given that y(1)=2, y(1.1)=2.2156, y(1.2)=2.4649, y(1.3)=2.7514. Use corrector formula thrice.

```
from sympy import *
def Milne(g,x0,h,y0,y1,y2,y3):
    x,y=symbols('x,y')
    f=lambdify([x,y],g)
    x1=x0+h
    x2=x1+h
    x3=x2+h
    x4=x3+h

    y10=f(x0, y0)
    y11=f(x1,y1)
    y12=f(x2,y2)
    y13=f(x3,y3)
    y4p=y0+(4*h/3)*(2*y11-y12+2*y13)
    print('predicted value of y4',y4p)
    y14=f(x4,y4p)
    for i in range(1,4):
        y4=y2+(h/3)*(y14+4*y13+y12)
        print('corrected value of y4 , iteration  %d '%i,y4)
```

```
        y14=f(x4,y4)
Milne('x**2+y/2',1,0.1,2,2.2156,2.4649,2.7514)
```

```
predicted value of y4 3.0792733333333335
corrected value of y4 , iteration  1  3.0793962222222224
corrected value of y4 , iteration  2  3.079398270370371
corrected value of y4 , iteration  3  3.079398304506173
```

```
corrected value of y4 , iteration 3 3.07939830450017
```

Apply Milne's predictor and corrector method to solve $dy/dx = x - y^2$, y(0)=2 obtain y(0.8). Take h=0.2. Use Runge-Kutta method to calculate required initial values.

```
Y=RungeKutta('x-y**2',0,0.2,0,0.8)
print('y values from Runge -Kutta method:',Y)
Milne('x-y**2',0,0.2,Y[0],Y[1],Y[2],Y[3])
```

```
y values from Runge -Kutta method: [0.    0.02 0.08 0.18 0.3 ]
predicted value of y4 0.3042133333333334
corrected value of y4 , iteration  1  0.3047636165214815
corrected value of y4 , iteration  2  0.3047412758696499
corrected value of y4 , iteration  3  0.3047421836520892
```

10.4 Exercise:

1. Find $y(0.1)$ by Runge Kutta method when $y' = x - y^2, y(0) = 1$.

   Ans: $y(0.1) = 0.91379$

2. Evaluate by Runge Kutta method : $y' = log(x + y), y(0) = 2$ at $x = 0(0.2)0.8$.

   Ans: $2.155, 2.3418, 2.557, 2.801$

3. Solve by Milnes method: $y' = x + y$, y(0)=1, h=0.1, Calculate $y(0.4)$. Calculate required initial values from Runge Kutta method.

   Ans: 1.583649219