Practical End Semester Lab Exam

Jeevan Koshy

1740256

In [1]:

```
import numpy as np
import matplotlib.pyplot as plt
import math
```

1. Water is flowing in a trapezoidal channel at a rate of $Q = 20m^3/s$. The critical depth y for such a channel must satisfy the equation $0 = 1 - \frac{Q^2}{gA_c^3}B$ where, $g = 9.81m/s^2$, A_c = the cross - sectional area can be related to depth y by $B = 3 + y$ and $A_c = 3y + \frac{y^2}{2}$. Solve for the critical depth using Bisection and false position. Use initial guesses of 0.5 and 2.5 and iterate until the approximate error falls below 1% or the number of iterations exceeds 10. Discuss your results.

In [ ]:

```
g = 9.81

def fun(x)->float:
    return 1 - (Q^2/g*A^3)*B

def nextapprox(a, b)->float:
    return (a+b)/2

if __name__=="__main__":
    a=float(input("Enter lower limit: "))
    b=float(input("Enter upper limit: "))
    X=np.linspace(a-1,b+1,1000)

    print()
    print("a={0}".format(a))
    print("b={0}".format(b))
    neg=0.0
    pos=0.0
    count=0
    print("f(a)={0}\nf(b)={1}\n\n".format(round(fun(a),6),round(fun(b),6)))
    error=[]
    funlist=[]
    if fun(a)*fun(b)<0.0:
        dash = '-' * 113
        print(dash)
        print("x\t\t   a\t\t       b\t\t     Aprroximation\t\t f(approx)\tRel err")
        print(dash)
        print()
        if fun(a)<0.0:
            neg=a
            pos=b
        else:
            neg=b
            pos=a
        print("{0}\t\t{1:.6f}\t\t{2:.6f}\t\t{3:.6f}\t\t{4:.6f}\t{5:.6f}".format(count+1,rou
        while True:
            count=count+1
            if fun(neg)*fun(pos)<0.0:
                print()
                x0=nextapprox(neg, pos)
                funlist.append(fun(x0))
                if fun(x0)<0:
                    neg=round(x0, 6)
                else:
                    pos=round(x0, 6)
                x1=nextapprox(neg, pos)
                error.append(abs(pos-neg)/abs(pos+neg))
                print("{0}\t\t{1:.6f}\t\t{2:.6f}\t\t{3:.6f}\t\t{4:.6f}\t{5:.6f}".format(cou
            if(abs(pos-neg)/abs(pos+neg) < 0.00005) :
                print()
                print("Approximate root is {0}".format(round(x1,6)))
                funlist.append(fun(x0))
                break
    else:
        print()
        print("Invalid interval entered")
```

2. It is general practice in engineering and science that equations be plotted as lines and discrete data as symbols. Here are some data for concentration (c) versus time (t) for the photodegradation of aqueous bromine. These data can be described by the following function: $c = 4.8e^{-0.034t}$. Use python program to create a plot displaying both the data (using diamond - shaped, filled - red symbols) and the function (using a green , dashed line). Plot the function for t = 0 to 70 min.

t(min)	C(ppm)
10	3.4
20	2.6
30	1.6
40	1.3
50	1.0
60	0.5

In [2]:

```
t = [10,20,30,40,50,60]
C = [3.4,2.6,1.6,1.3,1.0,0.5]

plt.plot(t,C)
plt.title("Time vs Concentration")
plt.xlabel("Time")
plt.ylabel("Concentration")
plt.show()
```

Out[2]:

```
<matplotlib.text.Text at 0x1638fd71710>
```