Mesuga, Reymond R.
BS Physics 3-1
Computational Physics 1

```
# Importing necessary modules and package
import numpy as np
from scipy import optimize
from numpy import cos
from numpy import sin
from numpy import exp
from numpy import log
import matplotlib.pyplot as plt
```

```
# Defining the function that can solve for roots using Bisection Method
# And a function that can solve for minimum


def bisection(f,a,b,N):

    # Where f is the given equation
    # a and b is the given interval
    # N is the number of iterations

    # Part below is dedicated for solving mimimum
    xmin_local = optimize.fminbound(f,a,b)
    print("local minimum =", xmin_local)
    # Part above is dedicated for solving mimimum


    # Part below is dedicated for solving roots
    if f(a)*f(b) >= 0:
        print("Bisection method fails.")
        return None
    a_n = a
    b_n = b
    for n in range(1,N+1):
        m_n = (a_n + b_n)/2
        f_m_n = f(m_n)
        if f(a_n)*f_m_n < 0:
            a_n = a_n
            b_n = m_n
        elif f(b_n)*f_m_n < 0:
            a_n = m_n
            b_n = b_n
        elif f_m_n == 0:
            print("Found exact solution.")
            return m_n
```

```
        else:
            print("Bisection method fails.")
            return None
    return (a_n + b_n)/2
```

```
# The examples below are existing equations that has difined roots already
# make sure that you put the function inside the function used in solving
# Bisection Method, that's why we used "lambda" to inform the computer that
# we are using a function as argument of another function.
# Note Bisection Method fails if the equation has two equal roots with opposite sign.

#Example 1
approx_phi = bisection(lambda x: x**2 - x - 1,1,2,25)
print("root =",approx_phi)
```

```
local minimum = 1.0000059608609866
root = 1.618033990263939
```

```
# Example 2: From Tao Pang
approx1 = bisection(lambda x: exp(x)*log(x) -x**2,1,2,6)
print("root =",approx1)
```

```
local minimum = 1.0000059608609866
root = 1.6953125
```

```
#Example 3
approx = bisection(lambda x: 3*x + sin(x) - exp(x),0,0.5,6)
print("root =",approx)
```

```
local minimum = 4.469534883430863e-06
root = 0.36328125
```

```
# Example 4
# Bisection Method fail because function has roots of x= -2,+2
approx3 = bisection(lambda x: x**3 -12*x,0,2,10)
print("root =", approx3)
```

```
local minimum = 1.9999959949686341
Bisection method fails.
root = None
```

```
# Example 5
approx4 = bisection(lambda x: x**3 -3*x**2 +1,0,1,20)
print("root =", approx4)
```

```
local minimum = 0.9999940391390134
root = 0.6527037620544434
```