Higher Institute of Engineering & Technology, El-Beheira

Computer Engineering Department

Forth assignment in numerical analysis

(1- Bisection method)

(2- Newton's method)

(3- Simple iteration method)

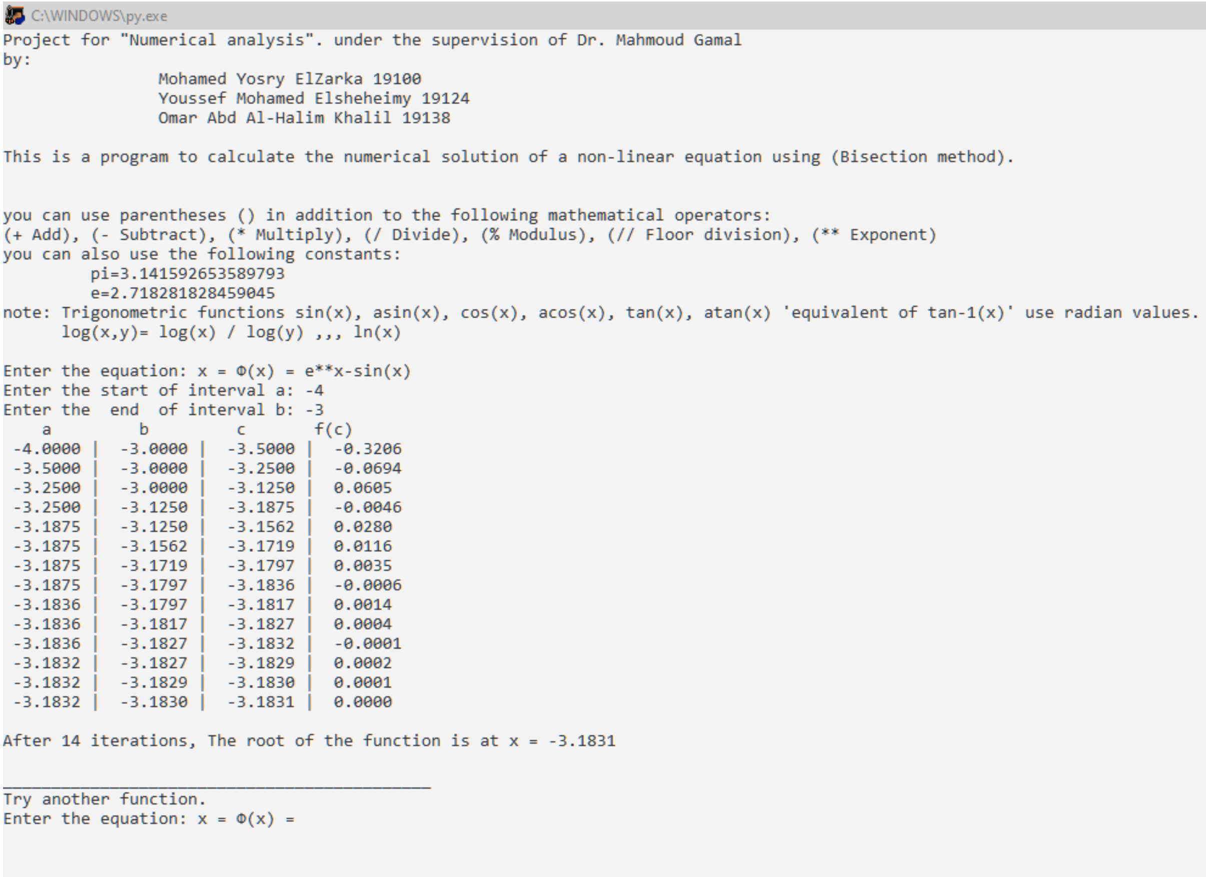
Under supervision of Dr.Mahmoud Gamal

Team	**ID**
Mohamed Yosry El-Zarka	19100
Youssef Mohamed El-Sheheimy	19124
Omar Abd Al-Halim Khalil	19138

1- Bisection method Source code in python: -

```python
import math
from sympy import *
pi=3.141592653589793
e=2.718281828459045

print('Project for "Numerical analysis". under the supervision of Dr. Mahmoud Gamal')
print('by:')
print('\t\tMohamed Yosry ElZarka 19100')
print('\t\tYoussef Mohamed Elsheheimy 19124')
print('\t\tOmar Abd Al-Halim Khalil 19138\n')

print("This is a program to calculate the numerical solution of a non-linear equation using (Bisection method).\n")

print("""
you can use parentheses () in addition to the following mathematical operators:
(+ Add), (- Subtract), (* Multiply), (/ Divide), (% Modulus), (// Floor division), (** Exponent)
you can also use the following constants:
\t pi=3.141592653589793
\t e=2.718281828459045
note: Trigonometric functions sin(x), asin(x), cos(x), acos(x), tan(x), atan(x) 'equivalent of tan-1(x)' use radian values.
        log(x,y)= log(x) / log(y) ,,, ln(x)
""")

while True:
    equation=str(input("Enter the equation: x = Φ(x) = "))
    a=float(eval(input("Enter the start of interval a: ")))
    b=float(eval(input("Enter the  end  of interval b: ")))
    x=a
    fa=round(eval(equation),4)
    x=b
    fb=round(eval(equation),4)
    if fa*fb>=0:
        print("\nThere is no roots in this interval, try again.")
        print("____________________________________________")
        continue
    print("    a         b         c       f(c)")
    i=0
    while True:
        i+=1
        c=round( (a+b)/2 , 4)
        x=c
        fc=round(eval(equation),4)
        x=a
        fa=round(eval(equation),4)
        print(" %.4f |"%a," %.4f |"%b," %.4f |"%c," %.4f "%fc)
        if fc==0: break
        if (fc*fa < 0):
            b=c
        else:
            a=c
    print("\nAfter",i,"iterations, The root of the function is at x =",c)
    print("\n________________________________________")
    print("Try another function.")
```

Bisection method program

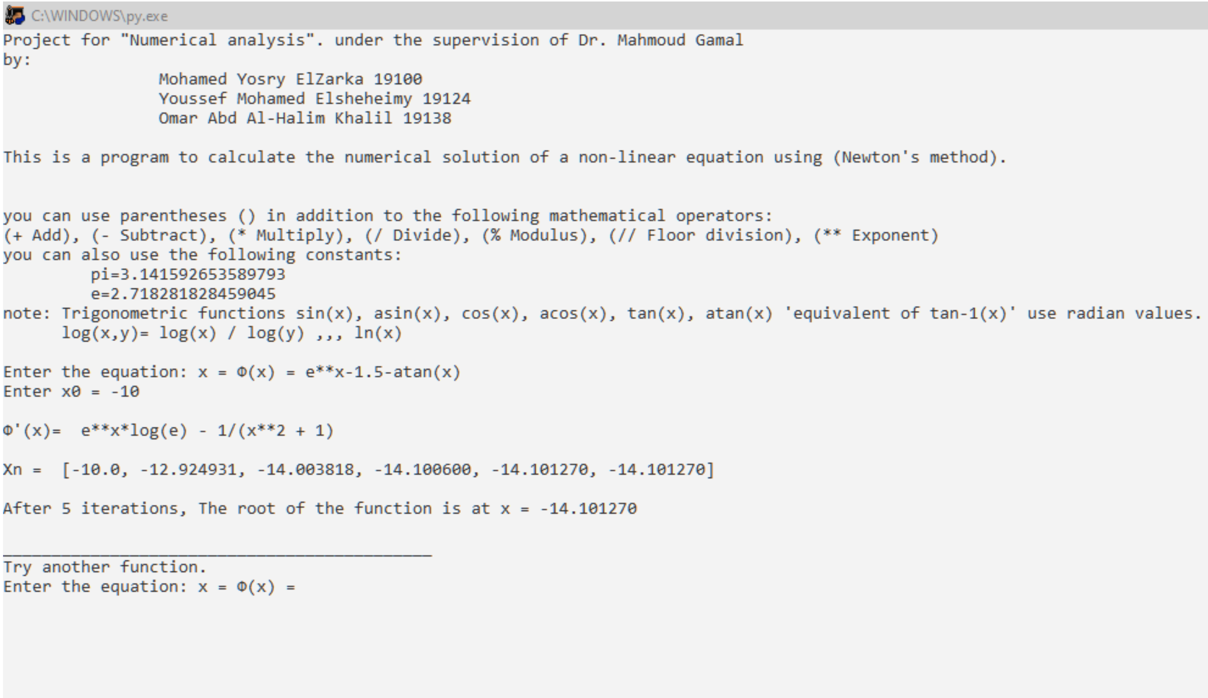


```
Project for "Numerical analysis". under the supervision of Dr. Mahmoud Gamal
by:
                Mohamed Yosry ElZarka 19100
                Youssef Mohamed Elsheheimy 19124
                Omar Abd Al-Halim Khalil 19138

This is a program to calculate the numerical solution of a non-linear equation using (Bisection method).


you can use parentheses () in addition to the following mathematical operators:
(+ Add), (- Subtract), (* Multiply), (/ Divide), (% Modulus), (// Floor division), (** Exponent)
you can also use the following constants:
         pi=3.141592653589793
         e=2.718281828459045
note: Trigonometric functions sin(x), asin(x), cos(x), acos(x), tan(x), atan(x) 'equivalent of tan-1(x)' use radian values.
      log(x,y)= log(x) / log(y) ,,, ln(x)

Enter the equation: x = Φ(x) = e**x-sin(x)
Enter the start of interval a: -4
Enter the  end  of interval b: -3
    a          b          c       f(c)
 -4.0000 |  -3.0000 |  -3.5000 |  -0.3206
 -3.5000 |  -3.0000 |  -3.2500 |  -0.0694
 -3.2500 |  -3.0000 |  -3.1250 |  0.0605
 -3.2500 |  -3.1250 |  -3.1875 |  -0.0046
 -3.1875 |  -3.1250 |  -3.1562 |  0.0280
 -3.1875 |  -3.1562 |  -3.1719 |  0.0116
 -3.1875 |  -3.1719 |  -3.1797 |  0.0035
 -3.1875 |  -3.1797 |  -3.1836 |  -0.0006
 -3.1836 |  -3.1797 |  -3.1817 |  0.0014
 -3.1836 |  -3.1817 |  -3.1827 |  0.0004
 -3.1836 |  -3.1827 |  -3.1832 |  -0.0001
 -3.1832 |  -3.1827 |  -3.1829 |  0.0002
 -3.1832 |  -3.1829 |  -3.1830 |  0.0001
 -3.1832 |  -3.1830 |  -3.1831 |  0.0000

After 14 iterations, The root of the function is at x = -3.1831


________________________________________
Try another function.
Enter the equation: x = Φ(x) =
```

2- Newton's method Source code in python: -

```python
import math
from sympy import *
pi=3.141592653589793
e=2.718281828459045

print("This is a program to calculate the numerical solution of a non-linear equation using (Newton's method).\n")

while True:
    equation=str(input("Enter the equation: x = Φ(x) = "))
    x0=float(eval(input("Enter x0 = ")))
    x0=round(x0,6)
    x = Symbol('x')
    x_list=[x0]
    equation_derivative= diff(equation,x)
    print("\nΦ'(x)= ",equation_derivative)
    del x
    x=x0
    x_dash= eval( str(equation_derivative) )

    for i in range(1,200): #maximum number of iterations is 200
        x=x_list[i-1] #update x
        xn= x-( eval(equation) / eval(str(equation_derivative)) )
        x_list.append( xn ) #calculating new x
        x_list[i]=round(x_list[i],6)
        if x_list[i]==x_list[i-1]:
            break
    print("\nXn = ",x_list)
    print("\nAfter",i,"iterations, The root of the function is at x =",x_list[i])
    print("\n____________________________________________")
    print("Try another function.")
```

Newton's method program

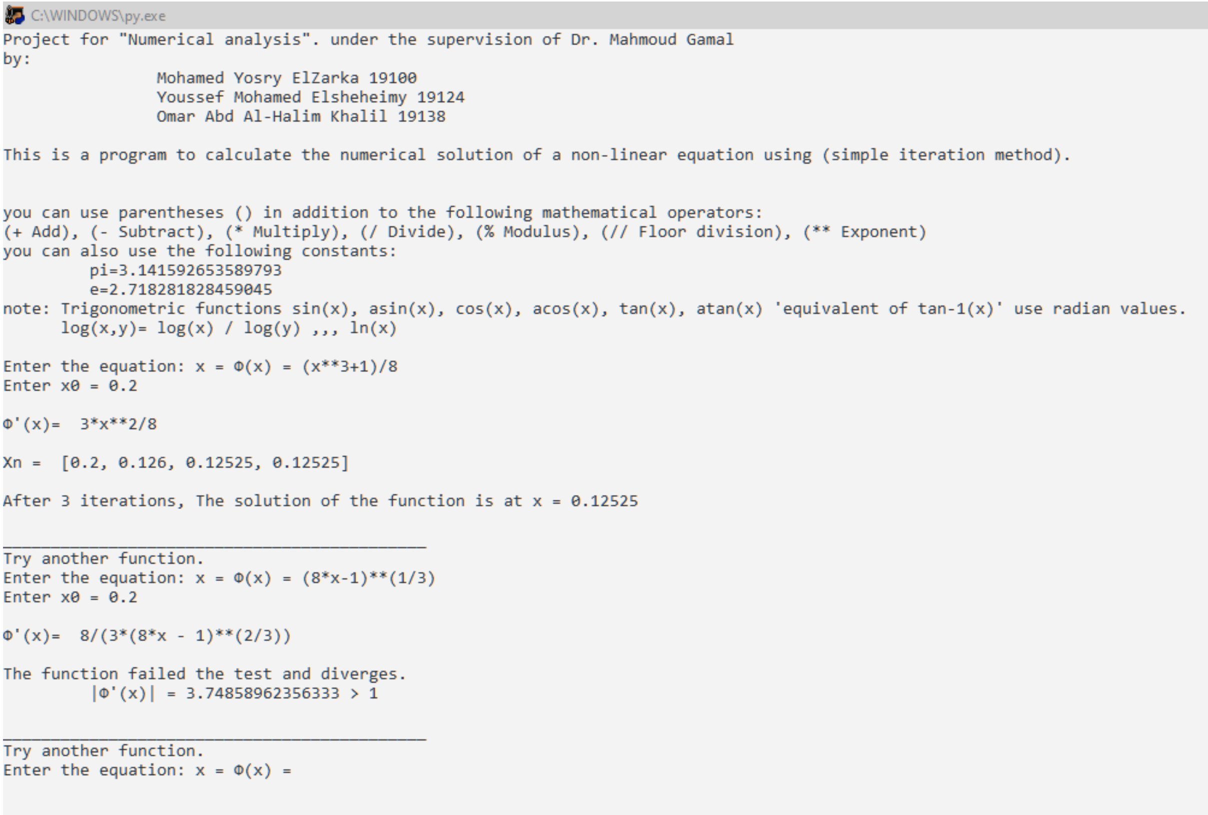


```
Project for "Numerical analysis". under the supervision of Dr. Mahmoud Gamal
by:
                Mohamed Yosry ElZarka 19100
                Youssef Mohamed Elsheheimy 19124
                Omar Abd Al-Halim Khalil 19138

This is a program to calculate the numerical solution of a non-linear equation using (Newton's method).


you can use parentheses () in addition to the following mathematical operators:
(+ Add), (- Subtract), (* Multiply), (/ Divide), (% Modulus), (// Floor division), (** Exponent)
you can also use the following constants:
         pi=3.141592653589793
         e=2.718281828459045
note: Trigonometric functions sin(x), asin(x), cos(x), acos(x), tan(x), atan(x) 'equivalent of tan-1(x)' use radian values.
      log(x,y)= log(x) / log(y) ,,, ln(x)

Enter the equation: x = Φ(x) = e**x-1.5-atan(x)
Enter x0 = -10

Φ'(x)=  e**x*log(e) - 1/(x**2 + 1)

Xn =  [-10.0, -12.924931, -14.003818, -14.100600, -14.101270, -14.101270]

After 5 iterations, The root of the function is at x = -14.101270

____________________________________________
Try another function.
Enter the equation: x = Φ(x) =
```

3- Simple iteration method Source code in python: -

```python
import math
from sympy import *
pi=3.141592653589793
e=2.718281828459045

print("This is a program to calculate the numerical solution of a non-linear equation using (simple iteration method).\n")

while True:
    equation=str(input("Enter the equation: x = Φ(x) = "))
     x0=float(eval(input("Enter x0 = ")))
     x = Symbol('x')
     x_list=[x0]
     equation_derivative= diff(equation,x)
     print("\nΦ'(x)= ",equation_derivative)
     del x #because We will need it as a float value not as a symbol

     x=x0
     test= eval( str(equation_derivative) )
     if abs(test)>=1:
         print("\nThe function failed the test and diverges.\n\t |Φ'(x)| = {} > 1 ".format(test))
     else:
         for i in range(1,200): #maximum number of iterations is 200
             x=x_list[i-1] #update x
             x_list.append( eval(equation) ) #calculating new x
             x_list[i]=round(x_list[i],5)
             if x_list[i]==x_list[i-1]:
                 break
         print("\nXn = ",x_list)
         print("\nAfter",i,"iterations, The solution of the function is at x =",x_list[i])
     print("\n____________________________________________")
     print("Try another function.")
```

Simple iteration method program



```
Project for "Numerical analysis". under the supervision of Dr. Mahmoud Gamal
by:
                Mohamed Yosry ElZarka 19100
                Youssef Mohamed Elsheheimy 19124
                Omar Abd Al-Halim Khalil 19138

This is a program to calculate the numerical solution of a non-linear equation using (simple iteration method).


you can use parentheses () in addition to the following mathematical operators:
(+ Add), (- Subtract), (* Multiply), (/ Divide), (% Modulus), (// Floor division), (** Exponent)
you can also use the following constants:
         pi=3.141592653589793
         e=2.718281828459045
note: Trigonometric functions sin(x), asin(x), cos(x), acos(x), tan(x), atan(x) 'equivalent of tan-1(x)' use radian values.
      log(x,y)= log(x) / log(y) ,,, ln(x)

Enter the equation: x = Φ(x) = (x**3+1)/8
Enter x0 = 0.2

Φ'(x)=  3*x**2/8

Xn =  [0.2, 0.126, 0.12525, 0.12525]

After 3 iterations, The solution of the function is at x = 0.12525


____________________________________________
Try another function.
Enter the equation: x = Φ(x) = (8*x-1)**(1/3)
Enter x0 = 0.2

Φ'(x)=  8/(3*(8*x - 1)**(2/3))

The function failed the test and diverges.
         |Φ'(x)| = 3.74858962356333 > 1


____________________________________________
Try another function.
Enter the equation: x = Φ(x) =
```