

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: -**

1 import math

2 from sympy import \*

3 pi=3.141592653589793

4 e=2.718281828459045

5

6 print('Project for "Numerical analysis". under the supervision of Dr. Mahmoud Gamal')

7 print('by:')

8 print('\t\tMohamed Yosry ElZarka 19100')

9 print('\t\tYoussef Mohamed Elsheheimy 19124')

10 print('\t\tOmar Abd Al-Halim Khalil 19138\n')

11

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

13

14 print("""

15 you can use parentheses () in addition to the following mathematical operators:

16 (+ Add), (- Subtract), (\* Multiply), (/ Divide), (% Modulus), (// Floor division), (\*\* Exponent)

17 you can also use the following constants:

18 \t pi=3.141592653589793

19 \t e=2.718281828459045

20 note: Trigonometric functions sin(x), asin(x), cos(x), acos(x), tan(x), atan(x) 'equivalent of tan-1(x)' use radian values.

21       log(x,y)= log(x) / log(y) ,,, ln(x)

22 """)

23

24 while True:

25     equation=str(input("Enter the equation: x = Ф(x) = "))

26     a=float(eval(input("Enter the start of interval a: ")))

27     b=float(eval(input("Enter the  end  of interval b: ")))

28     x=a

29     fa=round(eval(equation),4)

30     x=b

31     fb=round(eval(equation),4)

32     if fa\*fb>=0:

33         print("\nThere is no roots in this interval, try again.")

34         print("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

35         continue

36     print("    a         b         c       f(c)")

37     i=0

38     while True:

39         i+=1

40         c=round( (a+b)/2 , 4)

41         x=c

42         fc=round(eval(equation),4)

43         x=a

44         fa=round(eval(equation),4)

45         print(" %.4f |"%a," %.4f |"%b," %.4f |"%c," %.4f "%fc)

46         if fc==0: break

47         if (fc\*fa < 0):

48             b=c

49         else:

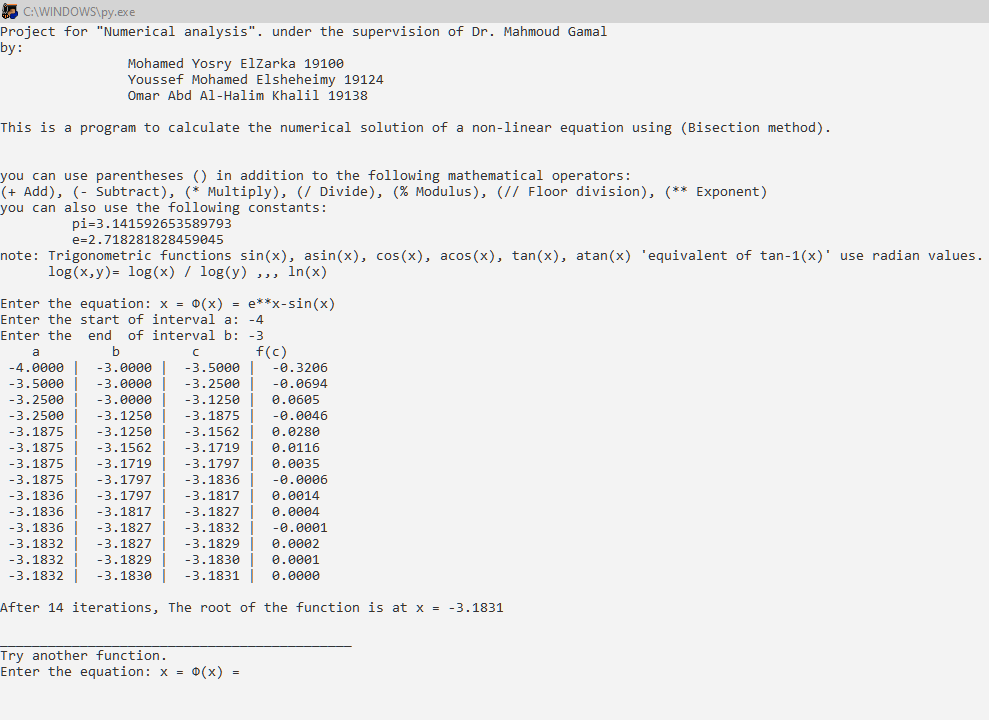
50             a=c

51     print("\nAfter",i,"iterations, The root of the function is at x =",c)

52     print("\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

53     print("Try another function.")

**Bisection method program**



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

1 import math

2 from sympy import \*

3 pi=3.141592653589793

4 e=2.718281828459045

5

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

7

8 while True:

9     equation=str(input("Enter the equation: x = Ф(x) = "))

10     x0=float(eval(input("Enter x0 = ")))

11     x0=round(x0,6)

12     x = Symbol('x')

13     x\_list=[x0]

14     equation\_derivative= diff(equation,x)

15     print("\nФ'(x)= ",equation\_derivative)

16     del x

17     x=x0

18     x\_dash= eval( str(equation\_derivative) )

19

20     for i in range(1,200): #maximum number of iterations is 200

21         x=x\_list[i-1] #update x

22         xn= x-( eval(equation) / eval(str(equation\_derivative)) )

23         x\_list.append( xn ) #calculating new x

24         x\_list[i]=round(x\_list[i],6)

25         if x\_list[i]==x\_list[i-1]:

26             break

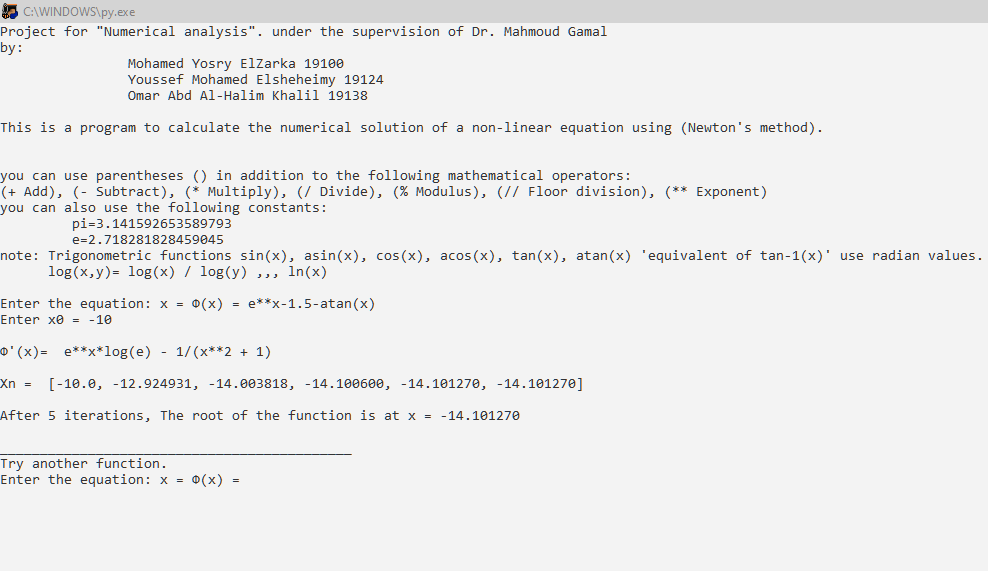
27     print("\nXn = ",x\_list)

28     print("\nAfter",i,"iterations, The root of the function is at x =",x\_list[i])

29     print("\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

30     print("Try another function.")

**Newton's method program**



**3- Simple iteration method Source code in python: -**

1 import math

2 from sympy import \*

3 pi=3.141592653589793

4 e=2.718281828459045

5

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

7

8 while True:

9     equation=str(input("Enter the equation: x = Ф(x) = "))

10     x0=float(eval(input("Enter x0 = ")))

11     x = Symbol('x')

12     x\_list=[x0]

13     equation\_derivative= diff(equation,x)

14     print("\nФ'(x)= ",equation\_derivative)

15     del x #because We will need it as a float value not as a symbol

16     x=x0

17     test= eval( str(equation\_derivative) )

18     if abs(test)>=1:

19         print("\nThe function failed the test and diverges.\n\t |Ф'(x)| = {} > 1 ".format(test))

20     else:

21         for i in range(1,200): #maximum number of iterations is 200

22             x=x\_list[i-1] #update x

23             x\_list.append( eval(equation) ) #calculating new x

24             x\_list[i]=round(x\_list[i],5)

25             if x\_list[i]==x\_list[i-1]:

26                 break

27         print("\nXn = ",x\_list)

28         print("\nAfter",i,"iterations, The solution of the function is at x =",x\_list[i])

29     print("\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_")

30     print("Try another function.")

**Simple iteration method program**

