LAB 1: Solution of algebraic and transcendental equation by Regula-Falsi and Newton-Raphson method

Regula-Falsi method to solve the algebraic and transcendental equation

Step 1: Express the given function in the form of $f(x)=0$

Step 2: Identify two consecutive integers *a* and *b* such that $f(a)<0$ & $f(b)>0$ (Means root lies between *a* and *b*)

Step 3: Compute $x=\dfrac{af(b)-bf(a)}{f(b)-f(a)}$

Step 4: If $f(x)<0$, take $a=x$ else if $f(x)>0$, take $b=x$ and go to step 3.

Step 5: If the difference between two consecutive values of *x* obtained is less than 0.0001, then assume the latest value *x* as root of the equation else go to step 3

Newton-Raphson method to solve the algebraic and transcendental equation near x_0

Step 1: Express the given function in the form of $f(x)=0$

Step 2: Compute $x_1=x_0-\dfrac{f(x_0)}{f'(x_0)}$

Step 3: If $x_1-x_0<0.0001\simeq 0$ then assume x_1 as root else use the above formula iteratively by taking $x_0=x_1$ until two consecutive values of *x* are almost same till fourth decimal place.

Python modules used in the lab:

```
sys.exit()      # Exit the interpreter by raising System Exit (status).

Python sys module   #The python sys module provides functions and
           variables which are used to manipulate different parts of the
           Python Runtime Environment. It lets us access system-specific
           parameters and functions. First, we have to import the sys
           module in our program before running any functions.

lambda arguments: expression   # The expression is executed and the
           result is returned.
           A lambda function is a small anonymous Function which can
           take any number of arguments, but can only have one
           expression.

diff(func, symbols)   # returns derivative of func with respect to
           symbols

x = Symbol('x') #to define single symbol
```

Program 1: Program to find the root of the Equation $x^2 - 2x - 1 = 0$, which lies between 2 and 3, by using Regula-Falsi Method

```python
import sys
f=lambda x: x**2 - 2*x - 1
error_tolerance=0.0001
max_iterations=50
a=2
b=3
if f(a)*f(b)>0:
    sys.exit("The root doesnot lies in the given range")
print("f(%0.5f)=%0.5f and f(%0.5f)=%0.5f"%(a,f(a),b,f(b)))
print("The root lies between %0.5f and %0.5f"%(a,b))
#Finding the root by Regula-Falsi iterative method
for i in range(max_iterations):
    x=(a*f(b)-b*f(a))/(f(b)-f(a))
    print(f"The approximate root obtained in iteration {i+1} is %.5f"%x)
    print("\nf(%0.5f)=%0.5f"%(x,f(x)))
    if abs(f(x))<error_tolerance:
        print("\nThe root of the equation is: %.4f"%x)
        break
    if f(x)<0:
        a=x
    else:
        b=x
    print("The root lies between %0.5f and %0.5f"%(a,b))
```

Output:

```
f(2.00000)=-1.00000 and f(3.00000)=2.00000
The root lies between 2.00000 and 3.00000
The approximate root obtained in iteration 1 is 2.33333

f(2.33333)=-0.22222
The root lies between 2.33333 and 3.00000
The approximate root obtained in iteration 2 is 2.40000

f(2.40000)=-0.04000
The root lies between 2.40000 and 3.00000
The approximate root obtained in iteration 3 is 2.41176

f(2.41176)=-0.00692
The root lies between 2.41176 and 3.00000
The approximate root obtained in iteration 4 is 2.41379

f(2.41379)=-0.00119
The root lies between 2.41379 and 3.00000
The approximate root obtained in iteration 5 is 2.41414

f(2.41414)=-0.00020
The root lies between 2.41414 and 3.00000
The approximate root obtained in iteration 6 is 2.41420
```

```
f(2.41420)=-0.00004

The root of the equation is: 2.4142
```

Program 2: Program to find the root of the Equation $x^2 - 2x - 1 = 0$ by using Regula-Falsi Method

```python
import sys
f=lambda x: x**2 - 2*x - 1
error_tolerance=0.0001
max_iterations=50
#Determine the values of a and b, between which root lies
for i in range(0,10):
    if f(i)*f(i+1)<0:
        if f(i)<0:
            a,b=i,i+1
        else:
            a,b=i+1,i
        break
    if i==10:
        sys.exit("The root doesnot lies in the given range")
print("f(%0.5f)=%0.5f and f(%0.5f)=%0.5f"%(a,f(a),b,f(b)))
print("The root lies between %0.5f and %0.5f"%(a,b))
#Finding the root by Regula-Falsi iterative method
for i in range(max_iterations):
    x=(a*f(b)-b*f(a))/(f(b)-f(a))
    print(f"The approximate root obtained in iteration {i+1} is %.5f"%x)
    print("\nf(%0.5f)=%0.5f"%(x,f(x)))
    if abs(f(x))<error_tolerance:
        print("\nThe root of the equation is: %.4f"%x)
        break
    if f(x)<0:
        a=x
    else:
        b=x
    print("The root lies between %0.5f and %0.5f"%(a,b))
```

Output:

```
f(2.00000)=-1.00000 and f(3.00000)=2.00000
The root lies between 2.00000 and 3.00000
The approximate root obtained in iteration 1 is 2.33333

f(2.33333)=-0.22222
The root lies between 2.33333 and 3.00000
The approximate root obtained in iteration 2 is 2.40000

f(2.40000)=-0.04000
The root lies between 2.40000 and 3.00000
The approximate root obtained in iteration 3 is 2.41176
```

```
f(2.41176)=-0.00692
The root lies between 2.41176 and 3.00000
The approximate root obtained in iteration 4 is 2.41379

f(2.41379)=-0.00119
The root lies between 2.41379 and 3.00000
The approximate root obtained in iteration 5 is 2.41414

f(2.41414)=-0.00020
The root lies between 2.41414 and 3.00000
The approximate root obtained in iteration 6 is 2.41420

f(2.41420)=-0.00004

The root of the equation is: 2.4142
```

Program 3: Program to find the root of the Equation $x^2-2x-1=0$ by using Newton-Raphson Method near $x_0=2$.

```python
from sympy import *
x=Symbol("x")
f=x**2-2*x-1
x0=2  #initial approximation
df=diff(f,x)
f=lambdify(x,f)
df=lambdify(x,df)
xn=x0-f(x0)/df(x0)
count=1
print(f"x{count} = %.5f"%xn)
while abs(xn-x0)>0.00001:
    x0=xn
    count+=1
    xn=x0-f(x0)/df(x0)
    print(f"x{count} = %.5f"%xn)
print(f"\nx{count}=x{count-1}, by correcting to 4 decimal places")
print("Hence the root of the equation is %.4f"%xn )
```

Output:

```
x1 = 2.50000
x2 = 2.41667
x3 = 2.41422
x4 = 2.41421

x4=x3, by correcting to 4 decimal places
Hence the root of the equation is 2.4142
```

Exercise: Write python program for the following

1. Using Regula-Falsi method find the real root of the equation

   (a) $xe^x = 2$ (b) $xe^x - \cos x = 0$ (c) $2x - \log_{10} x = 7$

   (d) $x^3 - 2x - 5 = 0$, which lies between 2 and 3

   (e) $3x - \cos x = 1$, which lies between 0 and 1.

   (f) $x^4 + x^3 - 7x^2 - x + 5 = 0$, which lies between 2 and 3.

2. Using Newton-Raphson method find the real root of the equation

   (a) $xe^x = 2$, near 1 (b) $xe^x - \cos x = 0$, near 1 (c) $2x - \log_{10} x = 7$, near 3

   (d) $x^3 - 2x - 5 = 0$, near 2 (e) $3x - \cos x = 1$, near 1.

   (f) $x^4 + x^3 - 7x^2 - x + 5 = 0$, near 2.5.