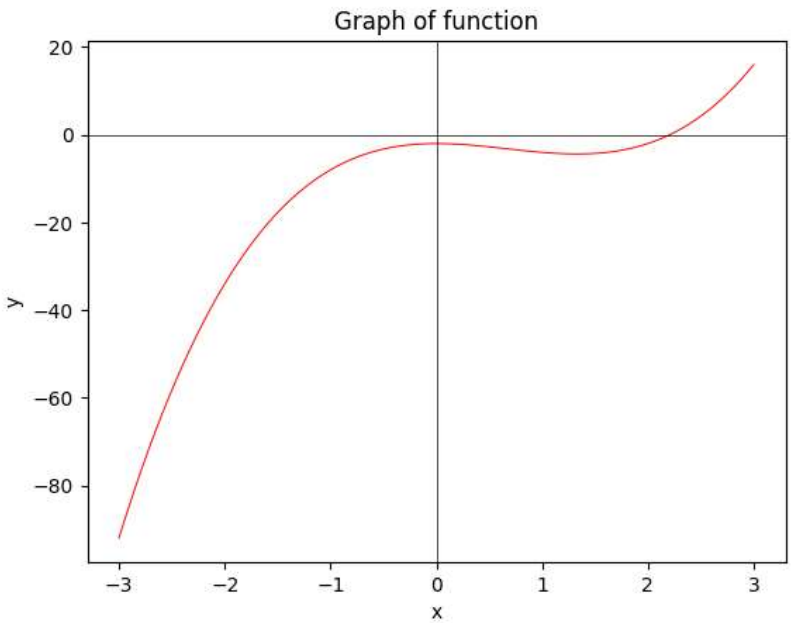
MWAURA DAVID NJUNG'E
SCT211-0002/2020
ICS 2207: SCIENTIFIC COMPUTING ASSIGNMENT

Finding the roots of a non-linear equation (Python implementation) using Bisection method and Newton Raphson method

A function f(x) is

$f(x) = 2x^3 - 4x^2 - 2$

Nature of curve



1. Bisection method

The curve has a root between $f(2)$ and $f(3)$

Initial interval (a, b) = (2, 3)

$f(a) = f(2) = -2$

$f(b) = f(3) = 16$

There exists a root between $f(2)$ and $f(3)$ since they have opposite signs

a=2; b=3; $c = \frac{a+b}{2}$

c= (2+3)/2=2.5

$f(c)$ = $f(2.5) = 4.25$ *(it is positive)*

$f(c)$ and $f(a)$ have opposite signs therefore set b=c

now a=2; b=2.5; c=(2+2.5)/2=2.25

$f(c)$=f(2.25)= 0.53125 *(it is positive)*

set b=c

now a=2; b=2.25 c=(2+2.25)/2=2.125

$f(c)$= $f(2.125) = 0.87109$ *(it is negative)*

$f(c)$ and f(a) have similar signs therefore set a=c

now a=2.125; b=2.25 c=(2.125+2.25)/2= 2.1875

$f(c)$=f(2.1875)= -0.20556 (its negative)

set a=c

now a=2.1875; b=2.25;

Continue until the desired level of accuracy is achieved.

Root=2.205569

Python implementation

```
def f(x):
    return 2*x**3-4*x**2-2

a = 2
b = 3
tolerance = 1e-6
maxiteration = 100

for i in range(maxiteration):
    c = (a + b) / 2
    if abs(f(c))<tolerance:
        print(f"Root found at x={ c:.6f}")

        break
    if f(c) * f(a) < 0:
        b = c
    else:
        a = c
```

output: `Root found at x=2.205570`

2. Newton Raphson method

$f(x) = 2x^3 - 4x^2 - 2$

$f'(x) = 6x^2 - 8x$

Choose an initial guess (x_0)

$x_0 = 2$

Evaluate $f(x_0)$ and $f'(x_0)$

$f(x_0) = f(2) = -2$

$f'(x_0) = f'(2) = 8$

The next approximation is given by:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

$x_1 = 2 - \frac{-2}{8} = 2.25$

The next approximation x_2 is given by:

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

$f(x_1) = f(2.25) = 0.53125$

$f(x_1) = f'(2.25) = 12.375$

$$x_2 = 2.25 - \frac{0.53125}{12.375} = 2.20707$$

Continue until the desired accuracy is achieved. The more the number of approximations, the more the accuracy

Root=2.205569

Python implementation

```python
def f(x):
    return 2*x**3-4*x**2-2

def df(x):
    return 6*x**2-8*x

x0 = 1
tol = 1e-6
max_iteration = 100

for i in range(max_iteration):
    x1=x0-f(x0)/df(x0)
    if abs(f(x1)) < tol:
        print(f"Root found at {x1:.6f}")
        break
    x0 = x1
else:
    print("failed to converge within maximum iterations")
```

output: `Root found at 2.205569`