# Bisection Method and the Newton Raphson Method

IRUNGU NELSON MUNGAI SCT211–0002/2021

March 2023

## 1 Newton Raphason Method

The following equation is solved by using the Newton Raphason method

$$x^5 + 5 * x^4 - 2$$

The following is the python implementation timed using timeit

```
import math
from timeit import timeit
def f(x):
    return x**5+5*x**4-2
def df(x):
    return 5*x**4+20*x**3
x0=1
tolerance=1e-6
max_iterations=1000
for i in range(max_iterations):
    fx=f(x0)
    dfx=df(x0)
    x1=x0-(fx/dfx)
    print(f'no of iteration {i}')
    if abs(f(x1))<tolerance:
        print(f'Root of the function is:{x1}')
        print(f'Time taken to run the program:')
        print(+timeit())
        break
    else:
        x0=x1
import matplotlib.pyplot as plt
import numpy as np
x=np.array(range(-20,20))
print(x)
y=x**5+5*x**4-2
```

```
print(y)
plt.plot(x,y)
```

## 2 Bisection Method

Solving the equation $x^5 + 5 * x^4 - 2$ using the Bisection Method
The following is the code in python

```
def bisection():
    a=0
    b=3
    for i in range(max_iterations):
        c=(a+b)/2
        print(f'Iteration number: {i}')
        if abs(f(c))<tolerance:
            print(f'Found root at {c:.6}')
            print("Time taken for bisection method")
            print(timeit())
            break
        elif f(c)*f(b)<0:
            a=c
        else:
            b=c

bisection()

#The newton Raphason method is faster in execution than
using the bisection method in solving the same equation
```