False position Method

In this chapter, we solve the following equation using the **Bisection Method**:

$$f(x) = x^3 - 10x^2 + 5 = 0 \tag{1}$$

Step 1:assuming g(x) from f(x)

First, we assume the non linear f(x) as linear g(x).

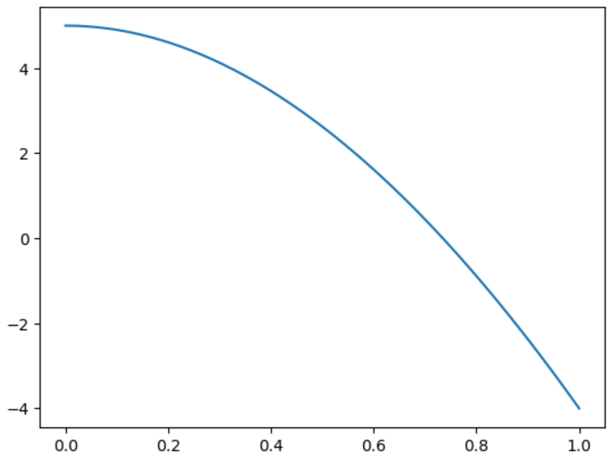

Figure 1: Plot of $f(x) = x^3 - 10x^2 + 5$

Then take slope of g(x) in two ways using a,b and varying point c (between a,b) and find relation for c in terms of a,b then find c taking f(c)=0

$$g'(x) = \frac{f(b) - f(a)}{a - b} = \frac{f(b) - f(c)}{b - c} \tag{2}$$

$$c = b - \frac{(b - a)f(b)}{f(b) - f(a)} \tag{3}$$

Step 2: Manual Calculation

Using the Bisection Method, we iteratively compute the midpoints and narrow the interval until we find the root.

Initial Calculations:

- For $x = 0.6,\ f(x) = 0.6^3 - 10(0.6)^2 + 5 = 1.616$
- For $x = 0.8,\ f(x) = 0.8^3 - 10(0.8)^2 + 5 = -0.888$

Iteration Results:

newpoint (c)	$f(c)$	**Interval**
0.729073	0.072056	$(0.729073, 0.8)$
0.734396	0.002701	$(0.734396, 0.8)$
0.734595	0.000100	$(0.734595, 0.8)$
0.7346035	0.000	—

Table 1: Manual Calculation of the False position Method

Step 3: Approximate Root

From the table above, the root of the equation is approximately:

$$\boxed{x = 0.7346035}$$

We've successfully found the root using the False position Method.

Let's see the python implementation

```

def f(x):
return x**3 -10*x**2 +5

import matplotlib.pyplot as plt
import numpy as np
x=np.linspace(0,1,100)
y=[]

for i in range(len(x)):
    y.append(f(x[i]))

plt.plot(x,y)


tolerance=0.0001
x3=[]
x2=0.8
x1=0.6
x3_=(x2+x1)/2
x3.append(x3_)
i=0


```

```

while abs(f(x3_))>0 :
    x3_=(x2+x1)/2
    x3.append(x3_)
    if f(x3_)*f(x1)>0 :
        x1=x3_

    elif f(x3_)*f(x1)<0 :
        x2=x3_
    i+=1

print(x3_)
```

Answer got from python code : 0.7346035077893033