

Regula Falsi Method

```
In [67]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Define a function Regula Falsi (interval, tol)

Tol breaking condition

Output table store in a Dataframe

```
In [43]: def f(x):
return np.cos(x) - 1/3*x
```

```
In [60]: alst = list()
blst = list()
rootlst = list()
itrlst = list()
```

```
In [61]: def RegulaFalsi(a,b,tol):
    i=0
    while(i<200):
        Xr = (a*f(b) - b*f(a)) / (f(b) - f(a))
        if(Xr ==0 or np.abs(f(Xr)) < tol):
            break
        if (f(a) * f(Xr)) < 0:
            b = Xr
        else:
            a = Xr
        i+=1
        itrlst.append(i)
        alst.append(a)
        blst.append(b)
        rootlst.append(Xr)
    return i, Xr
```

```
In [62]: ite, root = RegulaFalsi(-3, 3, 0.001)
```

```
In [63]: print('No of iterations to find the root ', ite)
print('Root for given equation ' , root)
```

No of iterations to find the root 4
Root for given equation 0.6236870740118996

```
In [64]: rfData = {
    'Itr' : itrlst,
    'a' : alst,
    'b' : blst,
    'root' : rootlst
}
```

```
In [65]: mytable = pd.DataFrame(rfData)
```

```
In [66]: x = mvtable
```

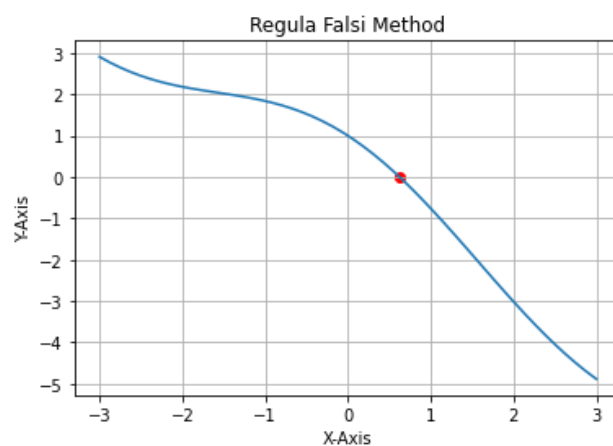
```
Out[66]:
```

	ltr	a	b	root
0	1	-0.761533	3	-0.761533
1	2	0.214639	3	0.214639
2	3	0.562570	3	0.562570
3	4	0.618359	3	0.618359

```
In [74]: x = np.linspace(-3,3)
plt.grid()

plt.title('Regula Falsi Method')
plt.xlabel('X-Axis')
plt.ylabel('Y-Axis')
plt.scatter(root, f(root), color='r');

plt.plot(x, f(x)).
```



Assembled & Compiled by : Hafiz Muhammad Waqas