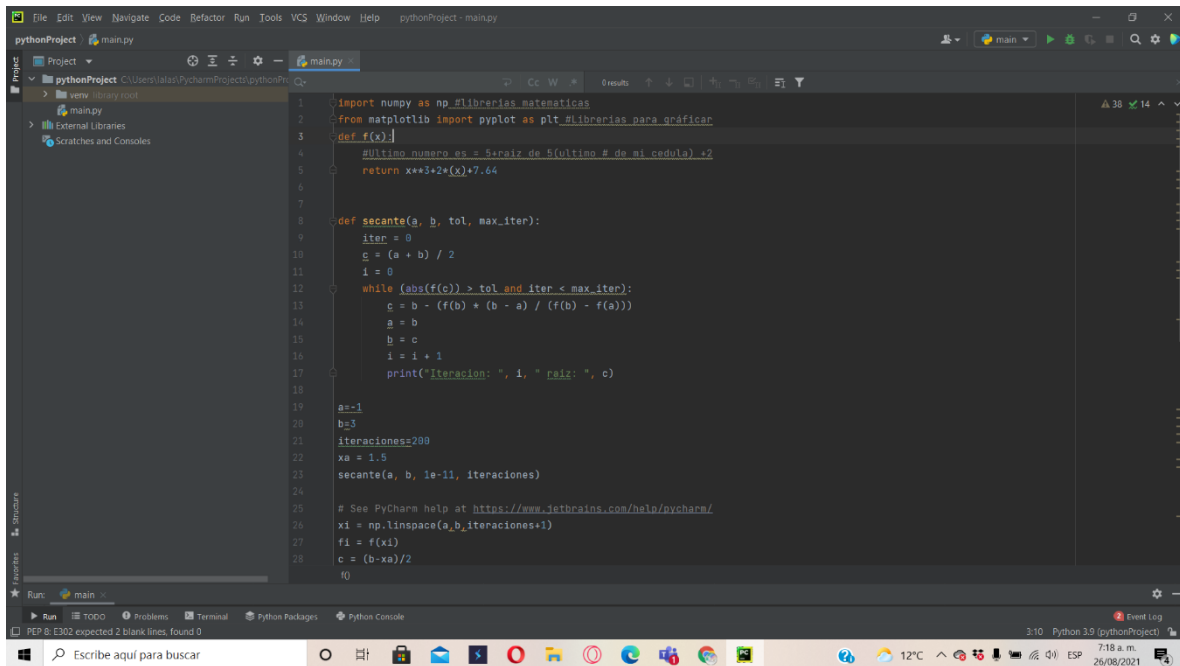


Laura Sofia Jiménez Ballén

1er parcial

1b. Método de la secante; $TOL < 10^{-11}$

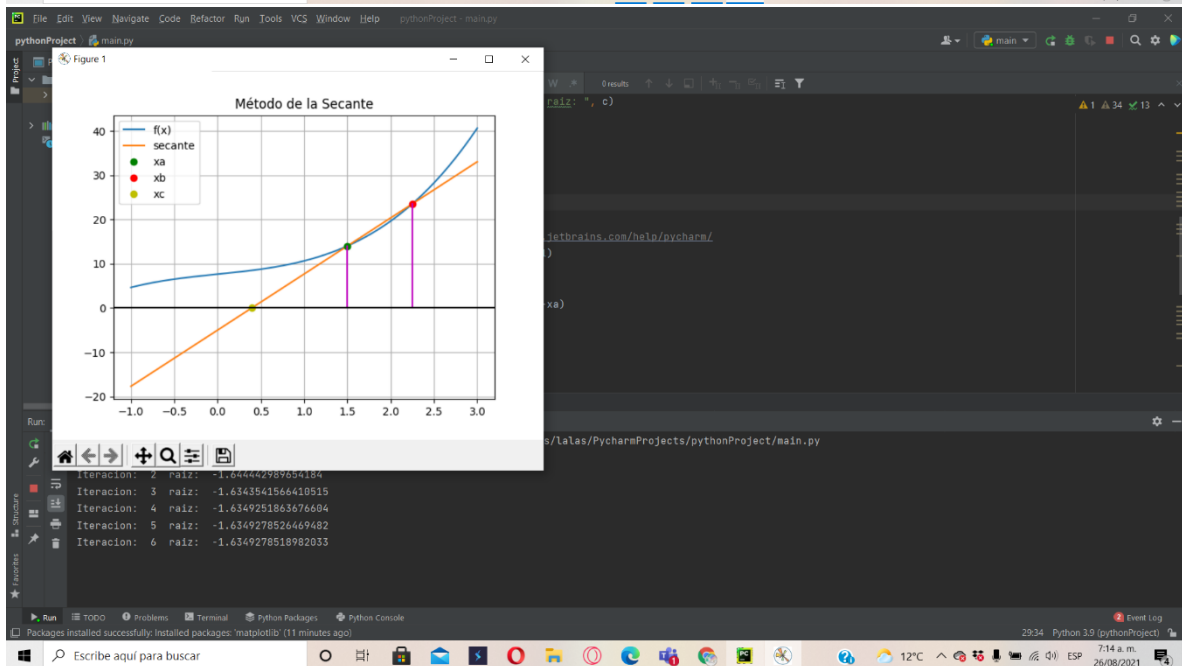
Ejercicio cuando $k = 7.64$



The screenshot shows the PyCharm IDE with a Python file named `main.py`. The code implements the secant method for finding roots of a function. The function `f(x)` is defined as `x**3 + 2*x + 7.64`. The `secante` function takes parameters `a`, `b`, `tol`, and `max_iter`. It initializes `iter = 0`, `c = (a + b) / 2`, and `i = 0`. A while loop continues as long as `abs(f(c)) > tol` and `iter < max_iter`. Inside the loop, `c` is updated to `b - (f(b) * (b - a) / (f(b) - f(a)))`, `a` is set to `b`, `b` is set to `c`, and `i` is incremented by 1. A print statement shows the iteration number and the root value `c`. The main code sets `a = -1`, `b = 1`, `iteraciones = 200`, and `xa = 1.5`, then calls `secante(a, b, 1e-11, iteraciones)`. It also generates a plot of the function `f(x)` using `np.linspace` and `plt`.

```
1 import numpy as np
2 from matplotlib import pyplot as plt
3 def f(x):
4     #ultimo numero es = 5raiz de 5(ultimo # de mi cedula) +2
5     return x**3+2*(x)+7.64
6
7
8 def secante(a, b, tol, max_iter):
9     iter = 0
10    c = (a + b) / 2
11    i = 0
12    while (abs(f(c)) > tol and iter < max_iter):
13        c = b - (f(b) * (b - a) / (f(b) - f(a)))
14        a = b
15        b = c
16        i = i + 1
17        print("Iteracion: ", i, " raíz: ", c)
18
19 a=-1
20 b=1
21 iteraciones=200
22 xa = 1.5
23 secante(a, b, 1e-11, iteraciones)
24
25 # See PyCharm help at https://www.jetbrains.com/help/pycharm/
26 x1 = np.linspace(a,b,iteraciones+1)
27 f1 = f(x1)
28 c = (b-xa)/2
29
```

```
pythonProject - main.py
File Edit View Navigate Code Refactor Run Tools VCS Window Help pythonProject - main.py
Project pythonProject C:\Users\lalas\PchamProjects\pythonProject
venv library root
main.py
External Libraries
Scratches and Consoles
28 c = (b-xa)/2
29 pendiente = (f(xa+c)-f(xa))/(xa+c-xa)
30 b0 = f(xa) - pendiente*xa
31 tangente1 = pendiente*x1+b0
32
33 fxa = f(xa)
34 xb = xa + c
35 fxb = f(xb)
36
37 plt.plot(x1,fi, label='f(x)')
38 plt.plot(x1,tangente1, label='secante')
39 plt.plot(xa,f(xa), 'go', label='xa')
40 plt.plot(xa+c,f(xa+c), 'ro', label='xb')
41 plt.plot((-b0/pendiente),0,'yo', label='xc')
42
43 plt.plot([xa,xa],[0,fxa], 'm')
44 plt.plot([xb,xb],[0,fxb], 'm')
45
46 plt.axhline(0, color='k')
47 plt.title('Método de la Secante')
48 plt.legend()
49 plt.grid()
50 plt.show()
10
```

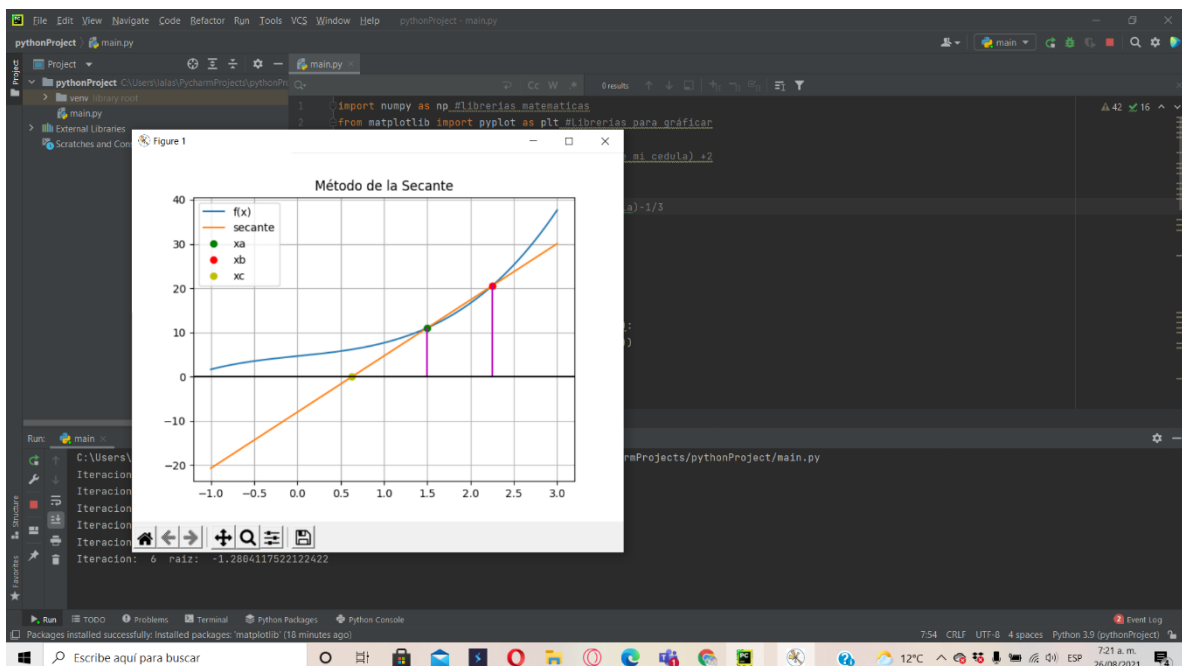


```
Run: main
C:\Users\lalas\PchamProjects\pythonProject\venv\Scripts\python.exe C:\Users\lalas\PchamProjects\pythonProject\main.py
Iteration: 1 raiz: -1.515555555555553
Iteration: 2 raiz: -1.644442989654184
Iteration: 3 raiz: -1.6343541566410515
Iteration: 4 raiz: -1.6349251863676604
Iteration: 5 raiz: -1.6349278526469482
Iteration: 6 raiz: -1.6349278518982033
Process finished with exit code 0
Packages installed successfully: Installed packages: 'matplotlib' (11 minutes ago)
```

Ejercicio cuando $k = 4.66$

```
pythonProject - main.py
1 import numpy as np
2 from matplotlib import pyplot as plt
3 def f(x):
4     # Ultimo numero es = 5*raiz de 5(ultimo # de mi cedula) +2
5     #return x**3+2*(x)+7.64
6
7     # Ultimo numero es = 5(ultimo # de mi cedula)-1/3
8     return x**3+2*(x)+4.66
9
10 def secante(a, b, tol, max_iter):
11     iter = 0
12     c = (a + b) / 2
13     i = 0
14     while (abs(f(c)) > tol and iter < max_iter):
15         c = b - (f(b) * (b - a)) / (f(b) - f(a))
16         a = b
17         b = c
18         i = i + 1
19     print("Iteracion: ", i, " raiz: ", c)
20
21 a=-1
22 b=3
23 iteraciones=200
24 xa = 1.5
25 secante(a, b, 1e-11, iteraciones)
26
27 # See PyCharm help at https://www.jetbrains.com/help/pycharm/
28 xi = np.linspace(a,b,iteraciones+1)
29
30
```

```
Run: main
C:\Users\lalas\PycharmProjects\pythonProject\venv\Scripts\python.exe C:\Users\lalas\PycharmProjects\pythonProject/main.py
Iteracion: 1 raiz: -1.1844444444444444
Iteracion: 2 raiz: -1.2557144682771
Iteracion: 3 raiz: -1.2817832726484872
Iteracion: 4 raiz: -1.28839278531322
Iteracion: 5 raiz: -1.2884117377136612
Iteracion: 6 raiz: -1.2884117522122422
Process finished with exit code 0
```



Punto 3:

c. $x^3 - 2x - 5$

```
1 import math
2 def puntofijo(fn, dfn, x, tol, max):
3     for i in range(max):
4         xnueva = x - fn(x)/dfn(x)
5         if abs(fn(xnueva)) < tol:
6             fixedPointIteration()
7     else:
8         fixedPointIteration()

Run: main
C:\Users\lalas\PycharmProjects\pythonProject1\venv\Scripts\python.exe C:/Users/lalas/PycharmProjects/pythonProject1/main.py
Tolerancia: 1e-5

Iteraciones y raices
Iteration-1, x1 = 0.408248 and f(x1) = -5.748455
Iteration-2, x1 = 0.842676 and f(x1) = -6.086965
Iteration-3, x1 = 0.736674 and f(x1) = -6.073564
Iteration-4, x1 = 0.758824 and f(x1) = -6.080706
Iteration-5, x1 = 0.754030 and f(x1) = -6.079348
Iteration-6, x1 = 0.755060 and f(x1) = -6.079649
Iteration-7, x1 = 0.754838 and f(x1) = -6.079584
Iteration-8, x1 = 0.754886 and f(x1) = -6.079598
Iteration-9, x1 = 0.754876 and f(x1) = -6.079595
Iteration-10, x1 = 0.754878 and f(x1) = -6.079596

Convergente
Intervalo:(5,10)

Process finished with exit code 0
```

Se necesitan 7 iteraciones para obtener aproximaciones precisas dentro de 10^{-5}