

## Computación

Aula Virtual: <https://famaf.aulavirtual.unc.edu.ar/course/view.php?id=747>

Resguardo tutoriales: <https://www.famaf.unc.edu.ar/~moreschi/docencia/Computacion/>

### Tutorial Problema 5 de la Guía N° 4

#### Problema 5:

Sobre los gráficos del problema 5 de la guía 3 grafique las estimaciones de las raíces que se van obteniendo en cada iteración como puntos en secuencia de colores para visualizar su convergencia. Hacer para las primeras 6 iteraciones.

#### Tutorial:

Realizaremos esto de dos formas distintas.

- Guarde en el archivo p5.py las siguientes instrucciones:

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 # caso a) -----
5 # Función cuya raíz queremos hallar
6 def f(x):
7     return 3*x - np.tan(x)
8
9 a = 0.1
10 b = 1.5
11 aini = a
12 bini = b
13 xx=[]
14 yy=[]
15
16 # Tolerancia para el intervalo que contiene la raíz
17 epsilon = 1.0e-6
18 intervalo = b - a
19 n = 0
20
21 while intervalo > epsilon:
22     n += 1
23     c = a + 0.5*intervalo
24     xx.append(c)
25     yy.append(f(c))
26     if f(c)*f(a) < 0:
27         b = c
28     else:
29         a = c
30     intervalo = b - a
31
32 # Aproximación a la raíz (punto medio del intervalo final)
33 c = a + 0.5*intervalo
34
```

```

35 print ("\nLa solución de 'tan(x)=3x' es: %5.6e" % (c))
36 print ("\nEl programa realizó %d iteraciones de bisección\n" % (n))
37
38 x_f = 1.324195e+00
39
40 x = np.linspace(aini, bini, 401)
41 y = f(x)
42
43 plt.figure( figsize=(10, 7.5) )
44 plt.xlim(0, 1.5)
45 plt.ylim(-3,3)
46 plt.xlabel('x')
47 plt.ylabel('función')
48 plt.title('Función 3*x - np.tan(x) e iteraciones')
49 plt.grid()
50 plt.plot(x,y)
51 for i in range(len(xx)):
52     plt.plot(xx[i],yy[i], 'o', label=i)
53     plt.text(xx[i]+0.01,yy[i]+0.1, i)
54 plt.legend(loc="lower left")
55 plt.savefig('graficos/p5-fa.png', dpi=100)
56 plt.show()
57
58 # -----
59 # Ahora hacemos una ampliación para ver más raíces.
60 plt.figure( figsize=(10, 7.5) )
61 plt.xlim(1.321,1.326)
62 plt.ylim(-0.02,0.05)
63 plt.xlabel('x')
64 plt.ylabel('función')
65 plt.title('Función 3*x - np.tan(x) e iteraciones')
66 plt.grid()
67 plt.plot(x,y)
68 for i in range(len(xx)):
69     plt.plot(xx[i],yy[i], 'o', label=i)
70     plt.text(xx[i]+0.0001,yy[i]+0.001, i)
71 plt.legend(loc="lower left")
72 plt.savefig('graficos/p5-fa-v2.png', dpi=100)
73 plt.show()
74
75 # caso b) -----
76 # Función cuya raíz queremos hallar
77 def f(x):
78     return x**6 - 5*x**5 + x**4 - 3*x**3 + x**2 - x + 1
79
80 a = 0.
81 b = 1.
82 aini = a
83 bini = b
84 xx=[]
85 yy=[]
86
87 # Tolerancia para el intervalo que contiene la raíz
88 epsilon = 1.0e-6
89 intervalo = b - a
90 n = 0

```

```

91
92 while intervalo > epsilon:
93     n += 1
94     c = a + 0.5*intervalo
95     xx.append(c)
96     yy.append(f(c))
97     if f(c)*f(a) < 0:
98         b = c
99     else:
100        a = c
101        intervalo = b - a
102
103 # Aproximación a la raíz (punto medio del intervalo final)
104 c = a + 0.5*intervalo
105
106 print ("\nLa solución de 'x**6 - 5*x**5 + x**4 - 3*x**3 + x**2 - x + 1' es:
107        %5.6e" % (c))
108
109 print ("\nEl programa realizó %d iteraciones de bisección\n" % (n))
110
111 x_f = 5.786958e-01
112
113
114 x = np.linspace(aini, bini, 601)
115 y = f(x)
116
117 plt.figure( figsize=(10, 7.5) )
118 plt.xlim(0, 1.0)
119 plt.ylim(-3,3)
120 plt.xlabel('x')
121 plt.ylabel('función')
122 plt.title('Función x**6 - 5*x**5 + x**4 - 3*x**3 + x**2 - x + 1 e
123        iteraciones')
124
125 plt.grid()
126 plt.plot(x,y)
127 for i in range(len(xx)):
128     plt.plot(xx[i],yy[i], 'o', label=i)
129     plt.text(xx[i]+0.01,yy[i]+0.1, i)
130 plt.legend(loc="lower left")
131 plt.savefig('graficos/p5-fb.png', dpi=100)
132 plt.show()
133
134 # -----
135 # Ahora hacemos una ampliación para ver más raíces.
136 plt.figure( figsize=(10, 7.5) )
137 plt.xlim(5.785958e-01, 5.787958e-01)
138 plt.ylim(-0.0005,0.0005)
139 plt.xlabel('x')
140 plt.ylabel('función')
141 plt.title('Función x**6 - 5*x**5 + x**4 - 3*x**3 + x**2 - x + 1 e
142        iteraciones')
143
144 plt.grid()
145 plt.plot(x,y)
146 for i in range(len(xx)):
147     plt.plot(xx[i],yy[i], 'o', label=i)
148     plt.text(xx[i]+0.000001,yy[i]+0.00005, i)
149 plt.legend(loc="lower left")
150 plt.savefig('graficos/p5-fb-v2.png', dpi=100)

```

- Desde la terminal ejecute:

```
python3 p5.py
```

e interprete el resultado.

Alternativamente ejecute:

```
python3
```

y vaya agregando uno a uno los bloques del programa.

Se deberían generar gráficos parecidos a éstos:

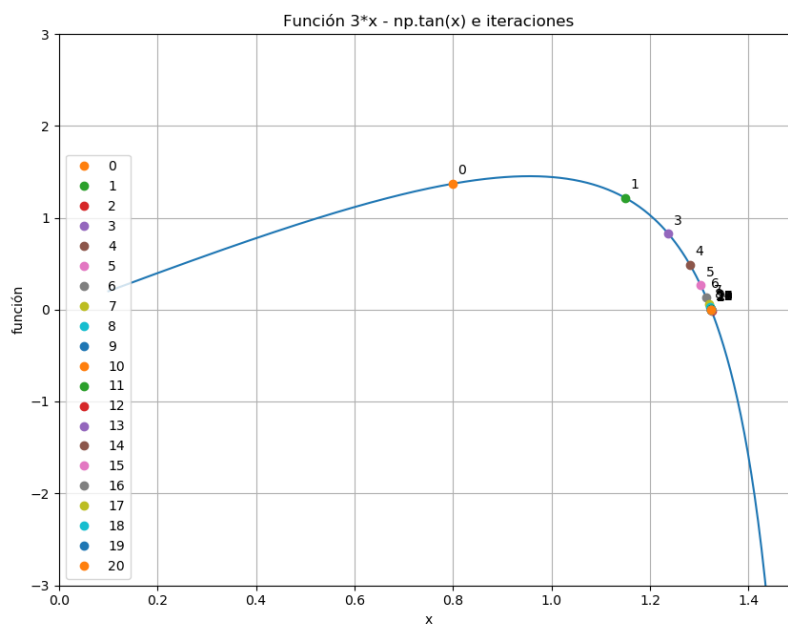


Figura 1: Primera versión  $f_a$ .

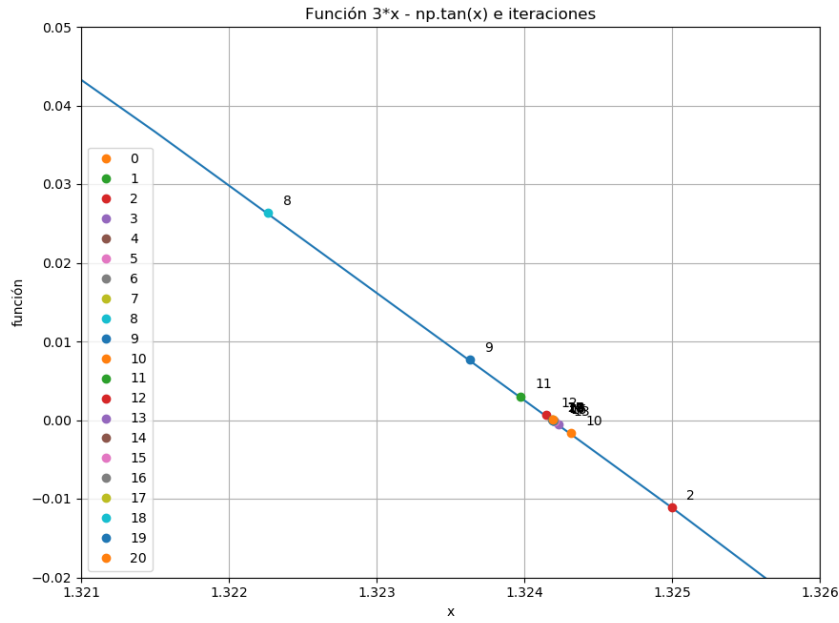


Figura 2: Ampliación  $f_a$ .

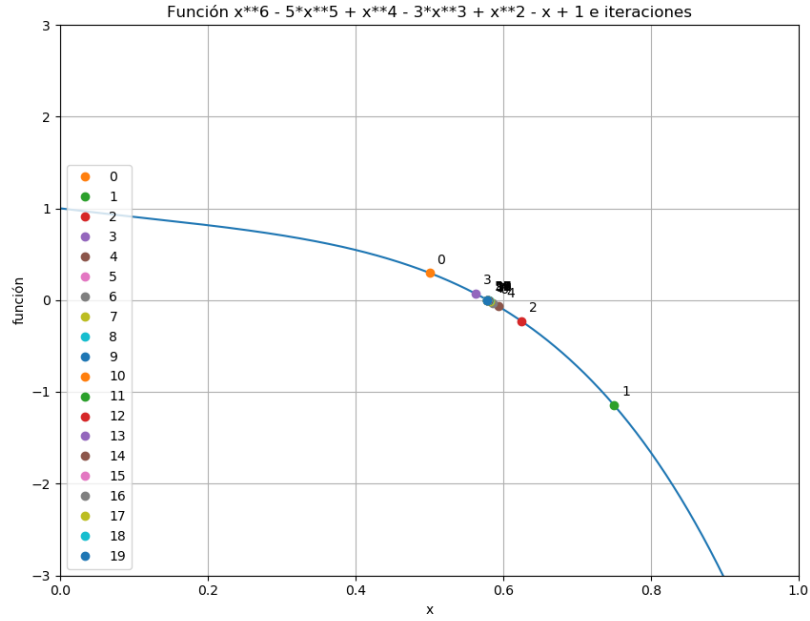


Figura 3: Primera versión  $f_b$ .

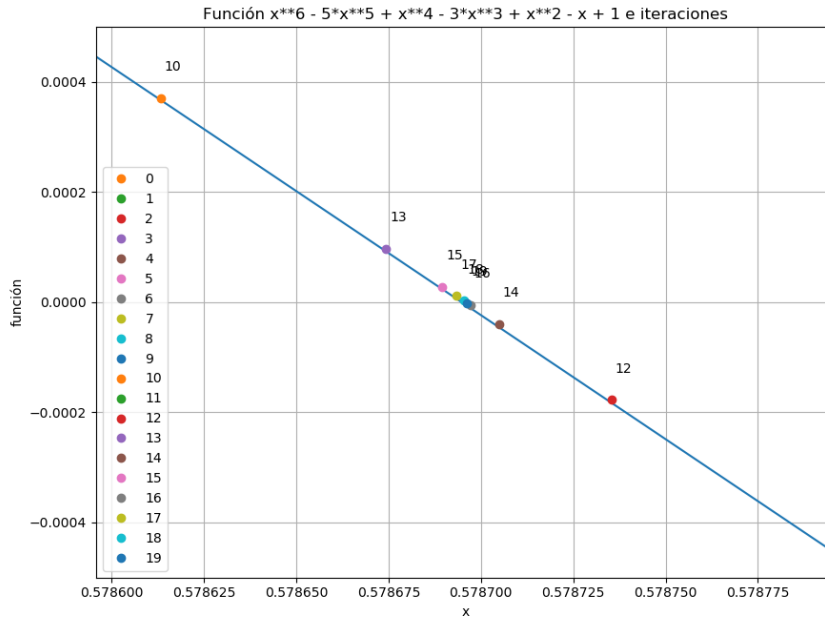


Figura 4: Ampliación  $f_b$ .