

Título

Pedro Bonilla Nadal

14 de marzo de 2018

1. Ejemplos.

Busca tres ecuaciones no lineales diferentes $f(x) = 0$, verificando $|f'(x^*)| \simeq 1$, $|f'(x^*)| \gg 1$, $|f'(x^*)| \ll 1$, respectivamente, de las que puedas obtener una solución exacta y un intervalo para cada una de ellas que contenga una única solución. Aproxima la misma en cada caso con el método de bisección fijando un error máximo y analiza la diferencia entre el número de iteraciones realizadas con cada una de las variantes $a)$, $b)$, $c)$ y $d)$ del método.

Solución.

▪ $f_1(x) = e^x - 1 \implies f_1(0) = 0, f'_1(0) = 1 \simeq 1$

Introduce el primer valor del intervalo: -1

Introduce el segundo valor del intervalo: 2

Introduzca la funcion de la que quiere hallar una raiz en el intervalo anterior:

`f(x) := np.exp(x)-1`

Criterio 1:

0.5

-0.25

0.125

-0.0625

0.03125

-0.015625

0.0078125

-0.00390625

0.001953125

-0.0009765625

0.00048828125

-0.000244140625

0.0001220703125

-6.103515625e-05

3.0517578125e-05

-1.52587890625e-05

7.62939453125e-06

-3.814697265625e-06

1.9073486328125e-06

1.9073486328125e-06

Criterio 2:

0.5
-0.25
0.125
-0.0625
0.03125
-0.015625
0.0078125
-0.00390625
0.001953125
-0.0009765625
0.00048828125
-0.000244140625
0.0001220703125
-6.103515625e-05
3.0517578125e-05
-1.52587890625e-05
7.62939453125e-06
7.62939453125e-06

Criterio 3:

0.5
-0.25
0.125
-0.0625
0.03125
-0.015625
0.0078125
-0.00390625
0.001953125
-0.0009765625
0.00048828125
-0.000244140625
0.0001220703125
-6.103515625e-05
3.0517578125e-05
-1.52587890625e-05
7.62939453125e-06
7.62939453125e-06

Criterio 4:

17
0.5
-0.25
0.125
-0.0625
0.03125
-0.015625
0.0078125
-0.00390625
0.001953125
-0.0009765625

0.00048828125
 -0.000244140625
 0.0001220703125
 -6.103515625e-05
 3.0517578125e-05
 -1.52587890625e-05
 7.62939453125e-06
 7.62939453125e-06

- $f_2(x) = e^x - e \implies f_2(1) = 0, f'_2(1) = e \gg 1$
- $f_3(x) = \ln(x) - 1 \implies f_3(e) = 0, f'_3(e) = \frac{1}{e} \ll 1$

2. Anexo-Código.

```
1 #Algoritmo de biseccion
2 #Realizadores:
3 #Pedro Bonilla , Johanna Capote y Guillermo Galindo
4 #Ejecucion:
5 # $ python3.5 Ejercicio1.py

7 #IMPORTANTE: para ejecutar este programa se requiere tener instalado "numpy"
8 #en su defecto , se puede sustituir por funciones de la libreria math
9
10 import numpy as np
11
12 Threshold = pow(10,-5)
13 Root = 0
14 counter = 0
15
16 #Lectura del intervalo
17 a=float(input('Introduce el primer valor del intervalo: '))
18 b=float(input('Introduce el segundo valor del intervalo: '))
19
20 #Los valores asignados son irrelevantes , pero necesitamos tener las variables
21   iniciadas
22 x1 = a
23 x2 = b
24
25 #Lectura de la funcion
26 funcion = input('Introduzca la funcion de la que quiere hallar una raiz en el \
27 intervalo anterior\nf(x) := ')
28 exec('f = lambda x:' + funcion)
29
30 #Signo de los extremos del intervalo
31 sgnI = np.sign(f(a))
32 sgnD = np.sign(f(b))
33
34 n = 0
35
36 #Criterio de parada 1
37 print ("Criterio 1:")
38 def halt1(x,y):
39     return True if abs(x-y) > Threshold else False
40
41 while(halt1(x1,x2)) :
42     x2=x1
43     x1 = (a+b)/2
44
45     #Reducimos el intervalo a la mitad
46     if(np.sign(f(x1)) == sgnI):
47         a = x1
48     else:
49         b = x1
50     #Descomentar la linea de abajo para mostrarlo en cada iteracion
51     #print(x1)
52
53 #Imprimimos la aproximacion final
54 print(x1)
55
56 #Criterio de parada 2
57 print ("Criterio 2:")
58 def halt2(x):
59     return True if abs(f(x)) > Threshold else False
```

```

61 while(halt2(x1)) :
    x1 = (a+b)/2
63
    #Reducimos el intervalo a la mitad
65     if(np.sign(f(x1)) == sgnI):
        a = x1
67     else:
        b = x1
69     #Descomentar la linea de abajo para mostrarlo en cada iteracion
    print(x1)
71
    #Imprimimos la aproximacion final
73 print(x1)
75
    #Criterio de parada 3
    Root=float(input('Introduce una raiz: '))
77
    print ("Criterio 3:")
79 def halt3(x):
    return True if abs(x-Root) > Threshold else False
81
    while(halt3(x1)) :
83         x1 = (a+b)/2
85
        #Reducimos el intervalo a la mitad
        if(np.sign(f(x1)) == sgnI):
87             a = x1
            else:
89                 b = x1
            #Descomentar la linea de abajo para mostrarlo en cada iteracion
91             print(x1)
93
        #Imprimimos la aproximacion final
        print(x1)
95
        #Criterio de parada 4
97         print ("Criterio 4:")
        n = int(np.log2((b-a)/Threshold)-1)
99         print(n)
        for i in range(n):
101             x1 = (a+b)/2
103
            #Reducimos el intervalo a la mitad
            if(np.sign(f(x1)) == sgnI):
105                 a = x1
                else:
107                     b = x1
                #Descomentar la linea de abajo para mostrarlo en cada iteracion
109                 print(x1)
111
        #Imprimimos la aproximacion final
        print(x1)

```

Ejercicio1.py