

Parcial Análisis numérico

Punto 1d Método del punto fijo; $TOL < 10^{-1}$

```
2 import math
3 import numpy as np
4 import matplotlib.pyplot as plt
5
6 def f(x):
7     return (x * x * x) + (2*x) + 10
8
9
10 # Se reescribe f(x)=0 a x = g(x)
11 def g(x):
12     return 1 / math.sqrt(1 + x)
13
14
15 # Implementando Punto fijo
16 def fixedPointIteration(x0, e, N):
17     print('\n\n*** PUNTO FIJO ITERATION ***')
18     step = 1
19     flag = 1
20     condition = True
21     while condition:
22         x1 = g(x0)
23         print('Iteration-%d, x1 = %0.6f and f(x1) = %0.6f' % (step, x1, f(x1)))
24         x0 = x1
25
26         step = step + 1
27
28         if step > N:
29             flag = 0
30             break
31
32     condition = abs(f(x1)) > e
33
34     if flag == 1:
35         print('\nRequired root is: %0.8f' % x1)
36     else:
37         print('\nNo Convergente.')
38
39
40 # Input Section
41 x0 = input('Ingresar Guess: ')
42 e = input('Error: ')
43 N = input('Paso Maximo: ')
44
45 # Convertiendo x0 y e to float
46 x0 = float(x0)
47 e = float(e)
48
49 # Convertiendo N a integer
50 N = int(N)
51
52 fixedPointIteration(x0, e, N)
53
54 x = np.linspace(0, 1.5, 100)
55 plt.plot(x, f(x))
56 plt.plot(x, f(x), label='f(x)')
57 plt.grid()
58 plt.show()
```

Salidas

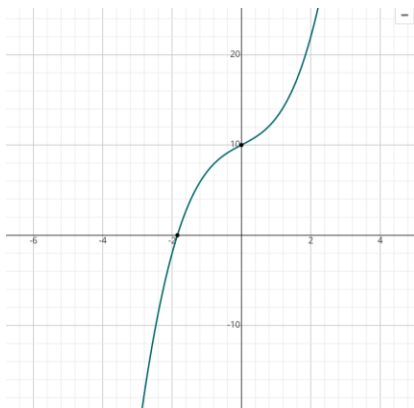
Con $k+Raiz(k+2)$

```
C:\Users\esroj\PycharmProjects\PuntoFijo_Parcial\venv\Scripts\python.exe C:\Users\esroj\PycharmProjects\PuntoFijo_Parcial\1punto.py
Ingresar Guess: 0
Error: 10
Paso Maximo: 10

*** PUNTO FIJO ITERATION ***
Iteration-1, x1 = 0.377964 and f(x1) = 10.889924
Iteration-2, x1 = 0.851885 and f(x1) = 12.321990
Iteration-3, x1 = 0.734840 and f(x1) = 11.866487
Iteration-4, x1 = 0.759225 and f(x1) = 11.956083
Iteration-5, x1 = 0.753944 and f(x1) = 11.936455
Iteration-6, x1 = 0.755078 and f(x1) = 11.940660
Iteration-7, x1 = 0.754834 and f(x1) = 11.939755
Iteration-8, x1 = 0.754887 and f(x1) = 11.939949
Iteration-9, x1 = 0.754876 and f(x1) = 11.939908
Iteration-10, x1 = 0.754878 and f(x1) = 11.939917
Iteration-11, x1 = 0.754878 and f(x1) = 11.939915

No Convergente.
```

Grafica



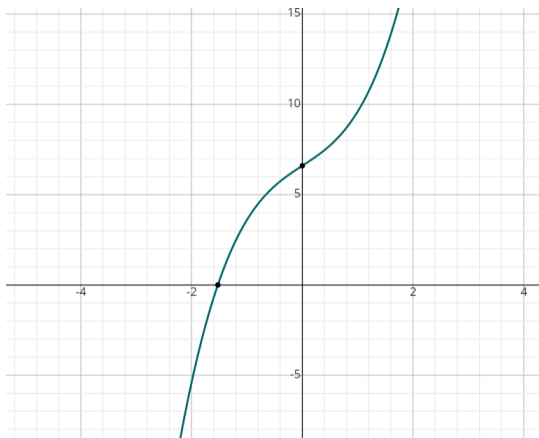
Con $k-1/3$

```
C:\Users\esroj\PycharmProjects\PuntoFijo_Parcial\venv\Scripts\python.exe C:\Users\esroj\PycharmProjects\PuntoFijo_Parcial\1punto.py
Ingresar Guess: 0
Error: 10
Paso Maximo: 10

*** PUNTO FIJO ITERATION ***
Iteration-1, x1 = 0.377964 and f(x1) = 7.409924
Iteration-2, x1 = 0.851885 and f(x1) = 8.921990
Iteration-3, x1 = 0.734840 and f(x1) = 8.466487
Iteration-4, x1 = 0.759225 and f(x1) = 8.556083
Iteration-5, x1 = 0.753944 and f(x1) = 8.536455
Iteration-6, x1 = 0.755078 and f(x1) = 8.540660
Iteration-7, x1 = 0.754834 and f(x1) = 8.539755
Iteration-8, x1 = 0.754887 and f(x1) = 8.539949
Iteration-9, x1 = 0.754876 and f(x1) = 8.539908
Iteration-10, x1 = 0.754878 and f(x1) = 8.539917
Iteration-11, x1 = 0.754878 and f(x1) = 8.539915

No Convergente.
```

Grafica



Punto 2c c. $f(x)=xe^{x^2}; x_0=0; P_3(0.4$

```
import numpy as np
import sympy as sp
import matplotlib.pyplot as plt
from sympy.parsing.sympy_parser import parse_expr

X = sp.Symbol('x')

def TeoremaTaylor(f,x,Dx,t):
    taylor = []
    for i in range(len(x)):

        A = sp.sympify(f).subs(X,x[i])

        B = sp.diff(f,X).subs(X,x[i])
        B = (np.transpose(B) * Dx[i])

        C = 0.5 * np.transpose(Dx[i])

        D = sp.diff(f,X,2).subs(X,x[i] + t * Dx[i])

        C = ((C * D) * Dx[i])

        LadoDerecho = A + B + C
        taylor.append(LadoDerecho)

    return taylor

## INPUT ##
funcion = input("Ingrese una funcion univariable con 'x' (eje: x**2-4*x): " ) # x**2+6*x
```

```

30     funcion = parse_expr(funcion, locals())
31     delta = float(input("Ingrese un delta x: "))_# 0.01
32     t = float(input("Ingrese un valor para t: "))_#1
33     intervalos = input("Ingrese un intervalo (separado por coma y sin parentesis ni corchetes): ")# [-6, 1]
34     intervalos = intervalos.split(",")
35
36     for i in range(len(intervalos)):
37         intervalos[i] = float(intervalos[i])
38
39     # Ejecucion del programa
40
41     ejex = [i for i in np.arange(intervalos[0], intervalos[1], step=0.1)]
42
43     y = []
44
45     deltax = []
46
47     for i in range(len(ejex)):
48         deltax.append(delta)
49         valorY = sp.sympify(funcion).subs(X, ejex[i])
50         y.append(valorY)
51
52     # Grafico
53
54     plt.grid(True)
55     plt.plot(ejex, y, color="red", label='Exacta')
56     plt.plot(ejex, TeoremaTaylor(funcion, ejex, deltax, t), color="blue", label='Taylor')
57     plt.legend(loc="upper right")
58     plt.show()

```

EL algoritmo se realiza de forma general para poder ingresar la función

Punto 3b Punto fijo evaluado con $x - \cos x$ obtener aproximaciones precisas dentro de 10^{-5} y realice los cálculos

```

1  import math
2  import numpy as np
3  import matplotlib.pyplot as plt
4
5  def f(x):
6      return x - np.cos(x)
7
8
9  # Se reescribe f(x)=0 a x = g(x)
10 def g(x):
11     return 1 / math.sqrt(1 + x)
12
13
14 # Implementando Punto fijo
15 def fixedPointIteration(x0, e, N):
16     print('\n\n*** PUNTO FIJO ITERATION ***')
17     step = 1
18     flag = 1
19     condition = True
20     while condition:
21         x1 = g(x0)
22         print('Iteration-%d, x1 = %0.6f and f(x1) = %0.6f' % (step, x1, f(x1)))
23         x0 = x1
24
25         step = step + 1
26
27     if step > N:
28         flag = 0
29         break

```

```

31         condition = abs(f(x1)) > e
32
33     if flag == 1:
34         print('\nRequired root is: %0.8f' % x1)
35     else:
36         print('\nNo Convergente.')
37
38
39     # Input Section
40     x0 = input('Ingresar Guess: ')
41     e = input('Error: ')
42     N = input('Paso Maximo: ')
43
44     # Convertiendo x0 y e to float
45     x0 = float(x0)
46     e = float(e)
47
48     # Convertiendo N a integer
49     N = int(N)
50
51     fixedPointIteration(x0, e, N)
52
53     x = np.linspace(0, 1.5, 100)
54     plt.plot(x, f(x))
55     plt.plot(x, f(x), label='f(x)')
56     plt.grid()
57     plt.show()

```

Resultados

```

C:\Users\esroj\PycharmProjects\PuntoFijo_Parcial\venv\Scripts\python.exe C:/Users/esroj/PycharmProjects/PuntoFijo_Parcial/venv/3punto.py
Ingresar Guess: 0
Error: 1e-5
Paso Maximo: 11

*** PUNTO FIJO ITERATION ***
Iteration-1, x1 = 0.377964 and f(x1) = -0.551453
Iteration-2, x1 = 0.851885 and f(x1) = 0.193319
Iteration-3, x1 = 0.734840 and f(x1) = -0.007097
Iteration-4, x1 = 0.759225 and f(x1) = 0.033855
Iteration-5, x1 = 0.753944 and f(x1) = 0.024950
Iteration-6, x1 = 0.755078 and f(x1) = 0.026861
Iteration-7, x1 = 0.754834 and f(x1) = 0.026450
Iteration-8, x1 = 0.754887 and f(x1) = 0.026538
Iteration-9, x1 = 0.754876 and f(x1) = 0.026519
Iteration-10, x1 = 0.754878 and f(x1) = 0.026523
Iteration-11, x1 = 0.754878 and f(x1) = 0.026522

```

Podemos observar que con 11 iteraciones obtenemos la mayor precisión con este rango de error

