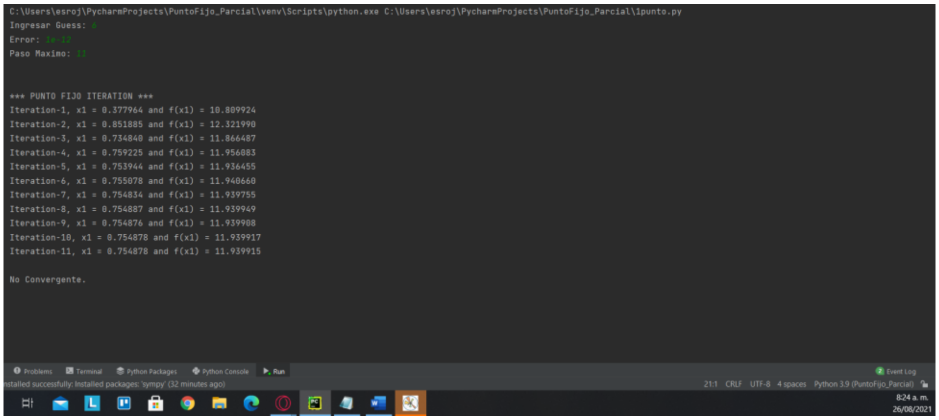
Parcial Análisis numérico

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

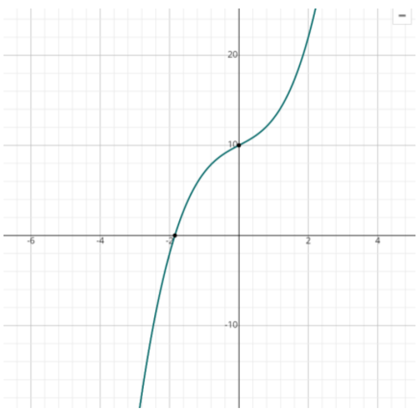
```python
import math
import numpy as np
import matplotlib.pyplot as plt

def f(x):
    return (x * x * x) + (2*x) + 10


# Se reescribe f(x)=0 a x = g(x)
def g(x):
    return 1 / math.sqrt(1 + x)


# Implementando Punto fijo
def fixedPointIteration(x0, e, N):
    print('\n\n*** PUNTO FIJO ITERATION ***')
    step = 1
    flag = 1
    condition = True
    while condition:
        x1 = g(x0)
        print('Iteration-%d, x1 = %0.6f and f(x1) = %0.6f' % (step, x1, f(x1)))
        x0 = x1

        step = step + 1

        if step > N:
            flag = 0
            break

        condition = abs(f(x1)) > e

    if flag == 1:
        print('\nRequired root is: %0.8f' % x1)
    else:
        print('\nNo Convergente.')


# Input Section
x0 = input('Ingresar Guess: ')
e = input('Error: ')
N = input('Paso Maximo: ')

# Convertiendo x0 y e to float
x0 = float(x0)
e = float(e)

# Convertiendo N a integer
N = int(N)

fixedPointIteration(x0, e, N)

x = np.linspace(0, 1.5, 100)
plt.plot(x, f(x))
plt.plot(x,f(x),label='f(x)')
plt.grid()
plt.show()
```

Salidas

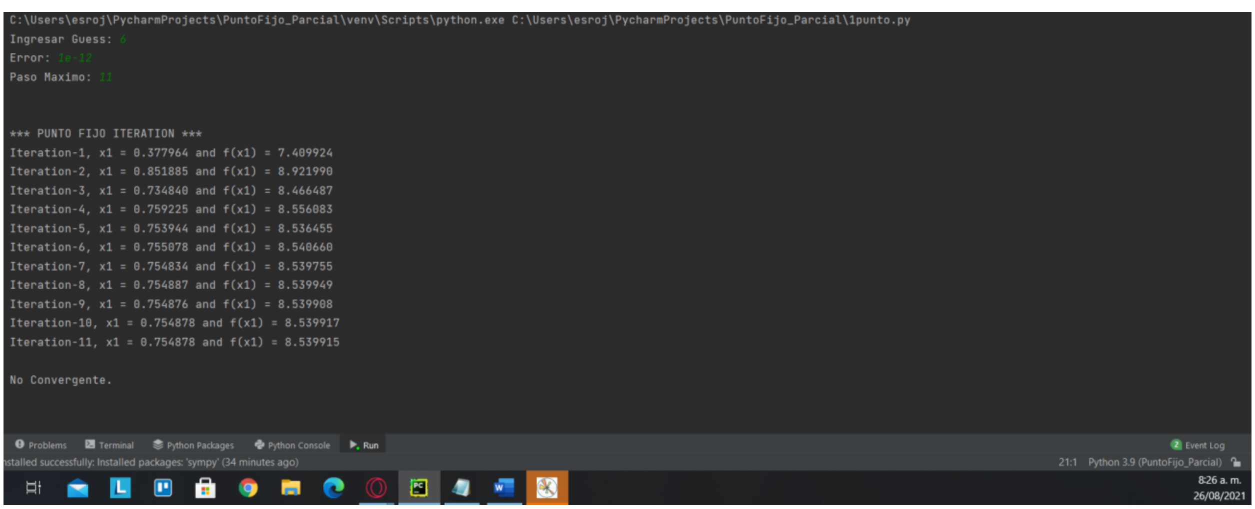
Con k+Raiz(k+2)



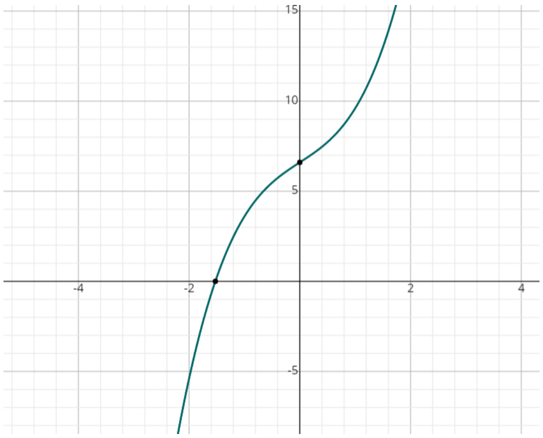
Grafica



Con k-1/3



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
```

```python
    funcion = parse_expr(funcion,locals())
    delta = float(input("Ingrese un delta x: ")) # 0.01
    t = float(input("Ingrese un valor para t: ")) #1
    intervalos = input("Ingrese un intervalo (separado por coma y sin parentesis ni corchetes): ")# [-6, 1]
    intervalos = intervalos.split(",")

    for i in range(len(intervalos)):
        intervalos[i] = float(intervalos[i])

    # Ejecucion del programa

    ejex = [i for i in np.arange(intervalos[0],intervalos[1],step=0.1)]

    y = []

    deltax = []

    for i in range(len(ejex)):
        deltax.append(delta)
        valorY = sp.sympify(funcion).subs(X, ejex[i])
        y.append(valorY)

    # Grafico

    plt.grid(True)
    plt.plot(ejex, y, color="red", label='Exacta')
    plt.plot(ejex, TeoremaTaylor(funcion, ejex, deltax, t), color="blue", label='Taylor')
    plt.legend(loc="upper right")
    plt.show()
```

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

Punto 3b Punto fijo evaluado con $x{-}cosx$ obtener aproximaciones precisas dentro de10^{-5} y realice los cálculos

```python
import math
import numpy as np
import matplotlib.pyplot as plt

def f(x):
    return x-np.cos(x)


# Se reescribe f(x)=0 a x = g(x)
def g(x):
    return 1 / math.sqrt(1 + x)


# Implementando Punto fijo
def fixedPointIteration(x0, e, N):
    print('\n\n*** PUNTO FIJO ITERATION ***')
    step = 1
    flag = 1
    condition = True
    while condition:
        x1 = g(x0)
        print('Iteration-%d, x1 = %0.6f and f(x1) = %0.6f' % (step, x1, f(x1)))
        x0 = x1

        step = step + 1

        if step > N:
            flag = 0
            break
```

```python
        condition = abs(f(x1)) > e

    if flag == 1:
        print('\nRequired root is: %0.8f' % x1)
    else:
        print('\nNo Convergente.')


# Input Section
x0 = input('Ingresar Guess: ')
e = input('Error: ')
N = input('Paso Maximo: ')

# Convertiendo x0 y e to float
x0 = float(x0)
e = float(e)

# Convertiendo N a integer
N = int(N)

fixedPointIteration(x0, e, N)

x = np.linspace(0, 1.5, 100)
plt.plot(x, f(x))
plt.plot(x,f(x),label='f(x)')
plt.grid()
plt.show()
```

Resultados

```
C:\Users\esroj\PycharmProjects\PuntoFijo_Parcial\venv\Scripts\python.exe C:/Users/esroj/PycharmProjects/PuntoFijo_Parcial/venv/3punto.py
Ingresar Guess: 6
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