MÉTODO PREDICTOR-CORRECTOR:

El código es el siguiente:

import matplotlib.pyplot as plt

import numpy as np

#definicion de puntos iniciales y final

punto\_inicial = float(input("Ingrese el punto inicial: "))

y0 = float(input("Ingrese el valor de y0: "))

punto\_final = float(input("Ingrese el punto final: "))

n = float(input("Ingrese numero divisiones: ")) #divisiones

h = (punto\_final - punto\_inicial) / n #paso

puntos = []

#funcion 1

def f1(x, y):

    return (2-3\*x-y)/(x-1)

def f2(x, y):

    return (1+4\*x\*y)/(3\*x\*\*2)

def f3(x, y):

    return (x\*y)/(x\*\*2 + y\*\*2)

def f4(x, y):

    return (x+y)/(x-y)

def iteracion(x, y, f):

    while x <= punto\_final:

        x1 = x + h

        z1 = y + h\*f(x, y)

        y1 = y + (h/2)\*(f(x, y) + f(x1, z1))

        iteracion(x1, y1, f)

        puntos.append((x1, y1))

        print(x1, y1)

        return x1, y1

def grafica(puntos):

    x = []

    y = []

    for i in puntos:

        x.append(i[0])

        y.append(i[1])

    plt.plot(x, y)

    plt.show()

iteracion(punto\_inicial, y0, f2)

grafica(puntos)

Ejercicio 1:

Carta

Descripción generada automáticamente con confianza media

Con el código de Python, la gráfica sale  
Gráfico, Gráfico de líneas

Descripción generada automáticamente

La solución real es y(6) = -8.2, y la aproximada es w100 = -8.2. Sale un error E = 0

Ejercicio 2:  
Carta

Descripción generada automáticamente con confianza mediaCon el código de Python, la gráfica sale

Gráfico, Gráfico de líneas

Descripción generada automáticamente

La solución aproximada sale: w100 = -11.4539, mientras que y(4) = -11.46. Hay un error

E = 0.0061