Deber 01 Métodos numéricos.

**Nombre:** Luis Enrique Pérez Señalin

Conjunto de ejercicios 1

Resuelva los siguientes ejercicios, tome en cuenta que debe mostrar el desarrollo completo del ejercicio.

1. Calcule los errores absoluto y relativo en las aproximaciones de p por .

Error\_abs =

Error\_relativo =

Error\_abs =

Error\_relativo =

Error\_abs =

Error\_relativo =

Error\_abs =

Error\_relativo =

1. Calcule los errores abosluto y relativo en las aproximaciones de p por

Error\_abs =

Error\_relativo =

Error\_abs =

Error\_relativo =

Error\_abs =

Error\_relativo =

Error\_abs =

Error\_relativo =

1. Encuentre el intervalo más largo en el que se debe encontrar para aproximarse a 𝑝 con error relativo máximo de para cada valor de p.

a.

b.

c.

d.

1. Use la aritmética de redondeo de tres dígitos para realizar lo siguiente. Calcule los errores absoluto y relativo con el valor exacto determinado para por lo menos cinco dígitos,

Error\_abs =

Error\_relativo =

Error\_abs =

Error\_relativo =

Error\_abs =

Error\_relativo =

Error\_abs =

Error\_relativo =

1. Los primeros tres términos diferentes a cero de la serie de Maclaurin para la función arcotangente son: . Calcule los errores absoluto y relativo en las siguientes aproximaciones de 𝜋 mediante el polinomio en lugar del arcotangente:

Error\_abs =

Error\_relativo =

Error\_abs =

Error\_relativo =

Resultados por programa de Python.

1. El número ℯ se puede definir por medio de Donde para y . Calcule los errores absoluto y relativo en la siguiente aproximación de :

Error\_abs =

Error\_relativo =

1. =

Error\_abs =

Error\_relativo =

1. Suponga que dos puntos y se encuentran en línea recta con . Existen dos fórmulas para encontrar la intersección 𝑥 de la línea:
2. Use los datos y la aritmética de redondeo de tres dígitos para calcular la intersección con 𝑥 de ambas maneras. ¿Cuál método es mejor y por qué?

La segunda fórmula de es mejor debido a que solo tiene 1 multiplicación y una división, lo que lo hace más preciso que la formula que tiene 2 multiplicaciones y una división.

Código:  
arctanFunctions.py:

#Codigo  
import math

def funct(x):

    return x - (1/3)\*x\*\*3 + (1/5)\*x\*\*5

def arctan(x, y):

    return math.atan2(x, y)

def get\_values(list):

    values = []

    for arg in (list):

        values.append(float(arg))

    return values

#End

1arctan.py

#Codigo

import sys

import arctanFunctions

[x0, y0, x1, y1] = arctanFunctions.get\_values(sys.argv[1:])

result\_1\_1 = arctanFunctions.arctan(x0,y0)

result\_1\_2 = arctanFunctions.funct(x0/y0)

print(f"Arctan: {result\_1\_1}")

print(f"Function: {result\_1\_2}")

result\_2\_1 = arctanFunctions.arctan(x1,y1)

result\_2\_2 = arctanFunctions.funct(x1/y1)

r1 = 4\*(result\_1\_1 + result\_2\_1)

r2 = 4\*(result\_1\_2 + result\_2\_2)

print(f"Arctan: {result\_2\_1}")

print(f"Result 2: {result\_2\_2}")

print(f"1: {r1}\n2: {r2}")

#End

2arctan.py

#codigo

import sys

import arctanFunctions

[x0, y0, x1, y1] = arctanFunctions.get\_values(sys.argv[1:])

result\_1\_1 = arctanFunctions.arctan(x0,y0)

result\_1\_2 = arctanFunctions.funct(x0/y0)

print(f"Arctan: {result\_1\_1}")

print(f"Function: {result\_1\_2}")

\_2result\_1\_1 = 16\*result\_1\_1

\_2result\_1\_2 = 16\*result\_1\_2

print(f"Arctan:   16\*{\_2result\_1\_1}")

print(f"Function: 16\*{\_2result\_1\_2}")

result\_2\_1 = arctanFunctions.arctan(x1,y1)

result\_2\_2 = arctanFunctions.funct(x1/y1)

print(f"Arctan: {result\_2\_1}")

print(f"Function: {result\_2\_2}")

\_2result\_2\_1 = 4\*result\_2\_1

\_2result\_2\_2 = 4\*result\_2\_2

print(f"Arctan:   4\*{\_2result\_2\_1}")

print(f"Function: 4\*{\_2result\_2\_2}")

r1 = \_2result\_1\_1 - \_2result\_2\_1

r2 = \_2result\_1\_2 - \_2result\_2\_2

print(f"R: Arctan:   {r1}")

print(f"R. Function: {r2}")

# End

Sumatoria.py

#codigo

import sys

def factorial(n):

  if n == 0:

    return 1

  else:

    return n \* factorial(n - 1)

max\_num = int(sys.argv[1])

values = []

for n in range(max\_num+1):

    value = 1/factorial(n)

    print(value, end=" ")

    values.append(value)

print(f"\nSumatoria: {sum(values)}")

#End