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 p^*.
   a. $p = \pi\ , p^* = \frac{22}{7}$
   $p\ \ = 3.1415926$
   $p^* = 3.1428571$
   Error_abs = $|\ p - p^*| = |3.1415 \ldots - 3.1428 \ldots| = 1.2644 * 10^{-3}$
   Error_relativo = $\frac{|\ p-p^*|}{|p|} = 4.0249 * 10^{-4}$

   b. $p = \pi, p^* = 3.1416$
   Error_abs = $|\ p - p^*| = |3.1415 \ldots - 3.1416 \ldots| = 7.3464 * 10^{-6}$
   Error_relativo = $\frac{|\ p-p^*|}{|p|} = 2.3384 * 10^{-6}$

   c. $p\ = e, p^* = 2.718$
   Error_abs = $|\ p - p^*| = |2.718281\ - 2.718| = 2.8182 * 10^{-4}$
   Error_relativo = $\frac{|\ p-p^*|}{|p|} = 1.0367 * 10^{-4}$

   d. $p = \ \sqrt{2}, p^* = 1.414$
   Error_abs = $|\ p - p^*| = |1.4142\ - 1.414| = 2.1356 * 10^{-4}$
   Error_relativo = $\frac{|\ p-p^*|}{|p|} = 1.5101 * 10^{-4}$

2. Calcule los errores abosluto y relativo en las aproximaciones de p por p^*

   a. $p = e^{10}, p^* = 220\ 0$
   Error_abs = $|\ p - p^*| = |22026.46579 - 2200| = 26.46579$
   Error_relativo = $\frac{|\ p-p^*|}{|p|} = 1.2015 * 10^{-3}$

   b. $p = 10^{\pi}, p^* = 1400$
   Error_abs = $|\ p - p^*| = |1385.455731 - 1400| = 14.544268$
   Error_relativo = $\frac{|\ p-p^*|}{|p|} = 1.0497 * 10^{-2}$

   c. $p = 8!, p^* = 39900$
   Error_abs = $|\ p - p^*| = |40320 - 39900| = 420$
   Error_relativo = $\frac{|\ p-p^*|}{|p|} = 1.04166 * 10^{-2}$

d. $p = 9!, p^* = \sqrt{18\pi}\left(\frac{9}{e}\right)^9$
Error_abs = | $p - p^*$| = $|362880 - 359536.8728| = 3343.1272$
Error_relativo = $9.21276 * 10^{-3}$

3. Encuentre el intervalo más largo en el que se debe encontrar p^* para aproximarse a p con error relativo máximo de p^{-4} para cada valor de p.

   a. π
   b. $\sqrt{2}$
   c. e
   d. $\sqrt[3]{7}$

4. 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,

   a. $\frac{\frac{13}{14}-\frac{5}{7}}{2e-5.4}$

$$p = \frac{0.928571 - 0.714285}{5.436563 - 5.4} = \frac{0.21428}{0.036563} = 5.8606204$$

$$p^* = \frac{0.928 - 0.714}{5.436 - 5.4} = \frac{-0.212}{0.036} = 5.944$$

Error_abs = | $p - p^*$| = $|5.86062 - 5.944| = 8.33795 * 10^{-2}$
Error_relativo = $1.422 * 10^{-2}$

   b. $-10\pi + 6e - \frac{3}{61}$

$$p = -31.41592 + 16.30969 - 0.04918 = -15.15541$$

$$p^* = -31.415 + 16.309 - 0.049 = -15.155$$
Error_abs = | $p - p^*$| = $|-15.15541 - -15.155| = 4.15893 * 10^{-4}$
Error_relativo = $2.74418 * 10^{-5}$

   c. $\left(\frac{2}{9}\right) * \left(\frac{9}{11}\right)$

$$p = 0.222222222 * 0.8181818182 = 0.18181818$$

$$p^* = 0.222 * 0.818 = 0.181$$

Error_abs = | $p - p^*$ | = $|0.18181 - 0.181| = 8.1 * 10^{-4}$
Error_relativo = $4.45520 * 10^{-3}$

d. $\frac{\sqrt{13}+\sqrt{11}}{\sqrt{13}-\sqrt{11}}$

e. $p = \frac{3.605551+3.316624}{3.605551-3.316624} = \frac{6.92217}{0.288926} = 23.95826$

f. $p^* = \frac{3.605+3.316}{3.605-3.316} = \frac{6.921}{0.289} = 23.948$
Error_abs = | $p - p^*$ | = $|23.95826 - 23.948| = 1.026 * 10^{-2}$
Error_relativo = $4.28224 * 10^{-4}$

5. Los primeros tres términos diferentes a cero de la serie de Maclaurin para la función arcotangente son: $x - \left(\frac{1}{3}\right)x^3 + \left(\frac{1}{5}\right)x^5$. Calcule los errores absoluto y relativo en las siguientes aproximaciones de π mediante el polinomio en lugar del arcotangente:

a. $4\left[\arctan\left(\frac{1}{2}\right) + \arctan\left(\frac{1}{3}\right)\right]$

$p = \quad 4[0.463647609 + 0.321750] = 4[0.7853981] \approx \pi = 3.1415926535$

$p^* = \ 4[0.4645833 \quad + \ 0.321810] = 4[0.786394029] = 3.14557613168$

Error_abs = | $p - p^*$ | = $|5.86062 - 5.944| = 3.98347 * 10^{-3}$
Error_relativo = $1.267980 * 10^{-3}$

b. $16 \arctan\left(\frac{1}{5}\right) - 4 \arctan\left(\frac{1}{239}\right)$
$p = \quad 16 * 0.19739555 \quad - 4 * 0.004184076$
$p = \quad 3.1583289575 \ - \ 0.0167363 \ = 3.1415926535$

$p^* = \quad 16 * 0.1973973333 - 4 * 0.004184076$
$p^* = \quad 3.158357333 \quad - \ 0.0167363 \ = 3.141621029$

Error_abs = | $p - p^*$ | = $|3.1415926535 - 3.141621029| = 2.8375735 * 10^{-5}$
Error_relativo = $9.032277 * 10^{-6}$

Resultados por programa de Python.

6. El número e se puede definir por medio de $e = \sum_{n=0}^{10}\left(\frac{1}{n!}\right)$ Donde $n! = n(n-1)\ldots 2*1$ para $n \neq 0$ y $0! = 1$. Calcule los errores absoluto y relativo en la siguiente aproximación de e:

   a. $\sum_{n=0}^{5}\left(\frac{1}{n!}\right) = \left(\frac{1}{0!}\right) + \left(\frac{1}{1!}\right) + \left(\frac{1}{2!}\right) + \left(\frac{1}{3!}\right) + \left(\frac{1}{4!}\right) + \left(\frac{1}{5!}\right)$

   $$= \left(\frac{1}{1}\right) + \left(\frac{1}{1}\right) + \left(\frac{1}{2}\right) + \left(\frac{1}{6}\right) + \left(\frac{1}{24}\right) + \left(\frac{1}{120}\right)$$

   $$= 1 + 1 + 0.5 + 0.1666 + 0.41666 + 0.008333$$

   $$\sum_{n=0}^{5}\left(\frac{1}{n!}\right) = 2.7166$$

   $p = 2.7182818$
   $p^* = 2.7166666$
   Error_abs = $|\,p - p^*| = |2.7182818\ldots - 2.7166666\ldots| = 1.615 * 10^{-3}$
   Error_relativo = $\frac{|\,p-p^*|}{|p|} = 5.94187 * 10^{-4}$

   b. $\sum_{n=0}^{10}\left(\frac{1}{n!}\right) = \left(\frac{1}{0!}\right) + \left(\frac{1}{1!}\right) + \left(\frac{1}{2!}\right) + \left(\frac{1}{3!}\right) + \left(\frac{1}{4!}\right) + \left(\frac{1}{5!}\right) + \left(\frac{1}{6!}\right) + \left(\frac{1}{7!}\right) + \left(\frac{1}{8!}\right) + \left(\frac{1}{9!}\right) + \left(\frac{1}{10!}\right)$

   $$= \left(\frac{1}{1}\right) + \left(\frac{1}{1}\right) + \left(\frac{1}{2}\right) + \left(\frac{1}{6}\right) + \left(\frac{1}{24}\right) + \left(\frac{1}{120}\right) + \left(\frac{1}{720}\right) + \left(\frac{1}{5040}\right) + \left(\frac{1}{40320}\right) + \left(\frac{1}{362880}\right) + \left(\frac{1}{3628800}\right)$$

   $$= 1 + 1 + 0.5 + 0.1666 + 0.41666 + 0.008333 + 0.0013888 + 0.00019841 + 2.48015 * 10^{-5} + 2.7557319 * 10^{-6} + 2.7557319 * 10^{-7}$$

   $$\sum_{n=0}^{10}\left(\frac{1}{n!}\right) = 2.7182818011$$

   $p = 2.718281828$
   $p^* = 2.7182818011$
   Error_abs = $|\,p - p^*| = |2.718281828 - 2.7182818011| = 2.690 * 10^{-8}$
   Error_relativo = $\frac{|\,p-p^*|}{|p|} = 9.8959569 * 10^{-9}$

7. Suponga que dos puntos (x_0, y_0) y (x_1, y_1) se encuentran en línea recta con $y_1 \neq y_0$. Existen dos fórmulas para encontrar la intersección x de la línea:

$$x = \frac{x_0 y_1 - x_1 y_0}{y_1 - y_0} \quad y\; x = x_0 - \frac{(x_1 - x_0) y_0}{y_1 - y_0}$$

   a. Use los datos $(x_0, y_0) = (1.31, 3.24) y\ (x_1, y_1) = (1.93, 5.76)$ y la aritmética de redondeo de tres dígitos para calcular la intersección con x de ambas maneras. ¿Cuál método es mejor y por qué?

$$x = \frac{x_0 y_1 - x_1 y_0}{y_1 - y_0} = \frac{1.31 * 5.76 - 1.93 * 3.24}{5.76 - 3.24} = \frac{7.545 - 6.253}{2.52} = \frac{1.292}{2.52} = 0.512$$

$$x\ = x_0 - \frac{(x_1 - x_0)y_0}{y_1 - y_0}\ = 1.31 - \frac{(1.93 - 1.31)3.24}{5.76 - 3.24} = \frac{0.62 * 3.24}{2.52} = \frac{2.008}{2.52} = 0.796$$

La segunda fórmula de $x\ = x_0 - \frac{(x_1 - x_0)y_0}{y_1 - y_0}$ 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
```