Conjunto de ejercicios 1

Boris Garcés

Tabla de Contenidos

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

1. Calcule los errores absolutos y relativos en las aproximaciones de p por p^*

a. $p = \pi, p^* = 22/7$

b. $p = \pi, p^* = 3.1416$

c. $p = e, p^* = 2.718$

d. $p = \sqrt{2}, p^* = 1.414$

a)

p= π

$p^* = 22/7$

```python
import math
Error_absoluto= abs(math.pi-(22/7))
Error_relativo= abs(Error_absoluto/math.pi)
print(f'El error absoluto es:  {Error_absoluto}')
print(f'El error relativo es: {Error_relativo}')
```

```
El error absoluto es:  0.0012644892673496777
El error relativo es: 0.0004024994347707008
```

b)

p= π

$p^* = 3.1416$

```python
Error_absoluto= abs(math.pi-3.1416)
Error_relativo= abs(Error_absoluto/math.pi)
print(f'El error absoluto es:  {Error_absoluto}')
print(f'El error relativo es: {Error_relativo}')
```

```
El error absoluto es:  7.346410206832132e-06
El error relativo es: 2.3384349967961744e-06
```

c)

p=e

p^*=2.718

```python
Error_absoluto= abs(math.e-2.718)
Error_relativo= abs(Error_absoluto/math.e)
print(f'El error absoluto es:  {Error_absoluto}')
print(f'El error relativo es: {Error_relativo}')
```

```
El error absoluto es:  0.0002818284590451192
El error relativo es: 0.00010367889601972718
```

d)

p= $\sqrt{2}$

p^*=1.414

```python
Error_absoluto= abs(2**(1/2)-1.414)
Error_relativo= abs(Error_absoluto/(2**(1/2)))
print(f'El error absoluto es:  {Error_absoluto}')
print(f'El error relativo es: {Error_relativo}')
```

```
El error absoluto es:  0.00021356237309522186
El error relativo es: 0.00015101140222192286
```

a. $p = e^{10}, p^* = 22000$

c. $p = 8!, p^* = 39900$

2. Calcule los errores absoluto y relativo enlas aproximaciones de p por p^*

a)

p= e^{10}

p^{*}=22000

```
Error_absoluto= abs(math.e**10-22000)
Error_relativo= abs(Error_absoluto/math.e**10)
print(f'El error absoluto es:  {Error_absoluto}')
print(f'El error relativo es: {Error_relativo}')
```

```
El error absoluto es:  26.465794806703343
El error relativo es: 0.0012015452253326688
```

b)

p=10^{π}

p^{*}=1400

```
Error_absoluto= abs(10**math.pi-1400)
Error_relativo=abs(Error_absoluto/10**math.pi)
print(f'El error absoluto es:  {Error_absoluto}')
print(f'El error relativo es: {Error_relativo}')
```

```
El error absoluto es:  14.544268632989315
El error relativo es: 0.010497822704619136
```

c)

p= 8!

p^{*}=39900

```
Error_absoluto= abs(math.factorial(8)-39900)
Error_relativo=abs(Error_absoluto/math.factorial(8))
print(f'El error absoluto es:  {Error_absoluto}')
print(f'El error relativo es: {Error_relativo}')
```

```
El error absoluto es:  420
El error relativo es: 0.010416666666666666
```

d)

p= 9!

p^{*}=$\sqrt{18*\pi}\cdot(\frac{9}{e})^9$

```
Error_absoluto= abs(math.factorial(9)-((18*math.pi)**(1/2)*(9/math.e)**9))
Error_relativo=abs(Error_absoluto/math.factorial(9))
print(f'El error absoluto es:  {Error_absoluto}')
print(f'El error relativo es: {Error_relativo}')
```

```
El error absoluto es:  3343.1271580516477
El error relativo es: 0.009212762230080598
```

3. Encuentre el intervalo más largo en el que se debe encontrar p^* para aproximarse a p con error relativo máximo de 10^{-4} para cada valor de p.

   a. π
   c. $\sqrt{2}$

Para resolver esta serie de problemas primero haremos cumplir la condición que indica que el error relativo debe ser máximo 10^{-4}

Error relativo= $\frac{|p-p^*|}{p}$

$\frac{|p-p^*|}{p} <= 10^{-4}$

$|p-p^*| <= p*10^{-4}$

$-p*10^{-4} <= p-p^* <= p*10^{-4}$

$-p-p*10^{-4} <= -p^* <= -p+p*10^{-4}$

$p+p*10^{-4} >= p^* >= p-p*10^{-4}$

a) p=π

```
# Limite superior
ls=math.pi+ (math.pi*10**-4)
#Limite inferior
li=math.pi-(math.pi*10**-4)
print(f'el límite superior es {ls}',f'el límite inferior es {li}',sep='\n')
print(f'el intervalo resultante es [{li};{ls}]')
```

```
el límite superior es 3.141906812855152
el límite inferior es 3.141278494324434
el intervalo resultante es [3.141278494324434;3.141906812855152]
```

b) p=e

```
# Limite superior
ls=math.e+ (math.e*10**-4)
#Limite inferior
li=math.e-(math.e*10**-4)
print(f'el límite superior es {ls}',f'el límite inferior es {li}',sep='\n')
print(f'el intervalo resultante es [{li};{ls}]')
```

```
el límite superior es 2.718553656641891
el límite inferior es 2.718010000276199
el intervalo resultante es [2.718010000276199;2.718553656641891]
```

c)$p = \sqrt{2}$

```
# Limite superior
ls=2**(1/2)+ (2**(1/2)*10**-4)
#Limite inferior
li=2**(1/2)-(2**(1/2)*10**-4)
print(f'el límite superior es {ls}',f'el límite inferior es {li}',sep='\n')
print(f'el intervalo resultante es [{li};{ls}]')
```

```
el límite superior es 1.4143549837293325
el límite inferior es 1.4140721410168577
el intervalo resultante es [1.4140721410168577;1.4143549837293325]
```

d)$\sqrt[3]{7}$

```
# Limite superior
ls=7**(1/3)+ (7**(1/3)*10**-4)
#Limite inferior
li=7**(1/3)-(7**(1/3)*10**-4)
print(f'el límite superior es {ls}',f'el límite inferior es {li}',sep='\n')
print(f'el intervalo resultante es [{li};{ls}]')
```

```
el límite superior es 1.9131224758906662
el límite inferior es 1.9127398896541117
el intervalo resultante es [1.9127398896541117;1.9131224758906662]
```

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. $\frac{\frac{13}{14}-\frac{5}{7}}{2e-5.4}$ b. $-10\pi + 6e - \frac{3}{61}$

c. $\left(\frac{2}{9}\right) \cdot \left(\frac{9}{11}\right)$ d. $\frac{\sqrt{13}+\sqrt{11}}{\sqrt{13}-\sqrt{11}}$

Figura 1: Ejercicio4

Para determinar los valores exactos se utilizarán los cálculos obtenidos por python, mientras que el redondeo se realizará manualmente.

a)

```
p=((13/14)-(5/7))/(2*math.e-5.4)
aprox=5.86
print(f'El valor exacto es: {p}',f'mientras que el valor de p^* es:{aprox}',sep='\n')
error_absoluto=abs(p-aprox)
error_relativo=error_absoluto/p
print(f'El error absoluto es {error_absoluto}',f'El error relativo es {error_relativo}',sep=
```

```
El valor exacto es: 5.860620417858059
mientras que el valor de p^* es:5.86
El error absoluto es 0.0006204178580588859
El error relativo es 0.00010586214663696585
```

b)

```
p=(-10*math.pi)+(6*math.e-3/61)
aprox=-15.2
print(f'El valor exacto es: {p}',f'mientras que el valor de p^* es:{aprox}',sep='\n')
error_absoluto=abs(p-aprox)
error_relativo=abs(error_absoluto/p)
print(f'El error absoluto es {error_absoluto}',f'El error relativo es {error_relativo}',sep=
```

```
El valor exacto es: -15.155415893012513
mientras que el valor de p^* es:-15.2
El error absoluto es 0.04458410698748594
El error relativo es 0.002941793699507889
```

c)

```python
p=(2/9)*(9/11)
aprox=0.182
print(f'El valor exacto es: {p}',f'mientras que el valor de p^* es:{aprox}',sep='\n')
error_absoluto=abs(p-aprox)
error_relativo=error_absoluto/p
print(f'El error absoluto es {error_absoluto}',f'El error relativo es {error_relativo}',sep=
```

```
El valor exacto es: 0.18181818181818182
mientras que el valor de p^* es:0.182
El error absoluto es 0.0001818181818181719
El error relativo es 0.0009999999999999454
```

d)

```python
p=((13)**(1/2)+(11)**(1/2))/((13)**(1/2)-(11)**(1/2))
aprox=24.0
print(f'El valor exacto es: {p}',f'mientras que el valor de p^* es:{aprox}',sep='\n')
error_absoluto=abs(p-aprox)
error_relativo=error_absoluto/p
print(f'El error absoluto es {error_absoluto}',f'El error relativo es {error_relativo}',sep=
```

```
El valor exacto es: 23.95826074310141
mientras que el valor de p^* es:24.0
El error absoluto es 0.04173925689859104
El error relativo es 0.0017421655664470355
```

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

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

   b. $16\ arctan\left(\frac{1}{5}\right) - 4\ arctan\left(\frac{1}{239}\right)$

se utilizarán los cálculos realizados a través de python para obtener el valor de las aproximaciones de π

a)

```python
tan1=1/2-(1/3)*(1/2)**3+(1/5)*(1/2)**5
tan2=1/3-(1/3)*(1/3)**3+(1/5)*(1/3)**5
aprox=4*(tan1+tan2)
print(f'El valor aproximado es: {aprox}')
print('El valor real es:', math.pi)
error_absolto=abs(aprox-math.pi)
error_relativo=error_absolto/math.pi
print(f'El error absoluto es: {error_absoluto}',f'El error relativo es: {error_relativo}',sep
```

```
El valor aproximado es: 3.1455761316872426
El valor real es: 3.141592653589793
El error absoluto es: 0.04173925689859104
El error relativo es: 0.0012679804598147663
```

b)

```python
tan1=1/5-(1/3)*(1/5)**3+(1/5)*(1/5)**5
tan2=1/239-(1/3)*(1/239)**3+(1/5)*(1/239)**5
aprox=16*(tan1+tan2)
print(f'El valor aproximado es: {aprox}')
print('El valor real es:', math.pi)
error_absolto=abs(aprox-math.pi)
error_relativo=error_absolto/math.pi
print(f'El error absoluto es: {error_absoluto}',f'El error relativo es: {error_relativo}',sep
```

```
El valor aproximado es: 3.2253025493665293
El valor real es: 3.141592653589793
El error absoluto es: 0.04173925689859104
El error relativo es: 0.026645687397149866
```

6. El número e se puede definir por medio de $e = \sum_{n=0}^{5}(1/n)$, donde $n!=n(n-1)...2.1 $ para $n \neq 0$ y 0!. Calcule los errorees absoluto y relativo en la siguiente aproximacion de e.

a. $\sum_{n=0}^{5}\left({}^{1}/_{n!}\right)$ b. $\sum_{n=0}^{10}\left({}^{1}/_{n!}\right)$

a)

```python
import math
suma=0
def factorial(n):
    if n==0:
        return 1
    else:
        factorial=1
        for i in range(1,n+1):
            factorial*=i
        return factorial
for i in range(0,6):
    suma+=(1/factorial(i))
error_absoluto= math.e-suma
print(f'El valor de error absoluto es: {error_absoluto}')
error_relativo=(error_absoluto/math.e)
print(f'El valor de error relativo es {error_relativo}')
```

```
El valor de error absoluto es: 0.0016151617923787498
El valor de error relativo es 0.0005941848175817597
```

b)

```python
import math
suma=0
def factorial(n):
    if n==0:
        return 1
    else:
        factorial=1
        for i in range(1,n+1):
            factorial*=i
        return factorial
for i in range(0,10):
    suma+=(1/factorial(i))
error_absoluto= math.e-suma
print(f'El valor de error absoluto es: {error_absoluto}')
error_relativo=(error_absoluto/math.e)
print(f'El valor de error relativo es {error_relativo}')
```

```
El valor de error absoluto es: 3.0288585284310443e-07
El valor de error relativo es 1.1142547828265698e-07
```

7. Suponga que dos puntos(x0,y0) y (x1,y1) se encuentran en línea recta con $y1! = y0$.Existen dos fórmulas para encontrar la intersección x de la línea

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

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?

```
import math
def formula1(x0,y0,x1,y1):
    punto=((x0*y1)-(x1*y0))/(y1-y0)
    return punto
def formula2(x0,y0,x1,y1):
    punto=x0-(((x1-x0)*y0)/(y1-y0))
    return punto
print(f'el valor de la formula 1 es: {formula1(1.31,3.24,1.93,5.76)}',f'el valor de la formul
print('Utilizando la formula 1:')
error_absoluto1=abs( formula1(1.31,3.24,1.93,5.76)- round(formula1(1.31,3.24,1.93,5.76),3))
print(f'El error absoluto es {error_absoluto1}')
```

```
el valor de la formula 1 es: 0.5128571428571429
el valor de la formula 2 es 0.5128571428571428
Utilizando la formula 1:
El error absoluto es 0.00014285714285711126
```

El segundo método es mejor ya que requiere de una cantidad menor de multiplicaciones y divisiones, para obtener la respuesta con un grado similar de presición