MetodosNumericosT3

October 31, 2024

1 ESCUELA POLITÉCNICA NACIONAL

1.1 MÉTODOS NUMÉRICOS

1.1.1 TAREA 3

David Alejandro Puga Novoa - GR1CC - 31/10/2024

1.2 1. Conjunto de Ejercicios

1.2.1 1.2. La serie de Maclaurin para la función arcotangente converge para $-1 < x \leq 1$ y está dada por:

$$\arctan x = \lim_{n\to\infty} P_n(x) = \lim_{n\to\infty} \sum_{i=1}^{n} (-1)^{i+1} \frac{x^{2i-1}}{2i-1}$$

a) Utilice el hecho de que $\tan \frac{\pi}{4} = 1$ para determinar el número n de términos de la serie que se necesita sumar para garantizar que $|4P_n(1) - \pi| < 10^{-3}$ Entonces, dado que $\arctan 1 = \frac{\pi}{4}$ con $x = 1$, reemplazando:

$$\frac{\pi}{4} = \sum_{i=1}^{n} (-1)^{i+1} \frac{1^{2i-1}}{2i-1}$$

[4]:
```python
import math
x = math.pi / 4
print(x)
```

```
0.7853981633974483
```

El código que calcula la sumatoria para cierto número n:

[5]:
```python
sum = 0
n = 1000
for i in range(1, n + 1):
    sum += ((-1) ** (i + 1)) * (1 ** (2*i - 1)/(2*i - 1))
print(sum)
```

```
0.7851481634599485
```

Se encontró el valor $n = 1000$ para el cual, la función se acerca al valor $\frac{\pi}{4}$.

Por tanto, $P_{1000}(1) = 0.78514....$

Comprobemos si cumple la condición:

$$|4P_{22}(1) - \pi| < 10^{-3}$$

[6]:
```
total = abs(4*sum - math.pi)
print(total)
print(total < 10**(-3))
```

```
0.000999999749998981
True
```

En la función print() directamente se hizo la comprobación sobre la inecuación con los valores dados.

b) El lenguaje de programación C++ requiere que el valor de π se encuentre dentro de 10^{-10}. ¿Cuántos términos de la serie se necesitarían sumar para obtener este grado de precisión?

Respuesta: En este caso se necesitaría hacer una suma infinita de la función: $P_n(\pi)$. Y esto se debe a que la tangente de π no está definida.

1.2.2 1.3. Otra fórmula para calcular π se puede deducir a partir de la identidad: $\frac{\pi}{4} = 4\arctan\frac{1}{5} - \arctan\frac{1}{239}$. Determine el número de términos que se deben sumar para garantizar una aproximación π dentro de 10^{-3}.

Para resolver este ejercicio, despejamos el número 4 de la izquierda, quedándonos:

$$\pi = \frac{4\arctan\frac{1}{5} - \arctan\frac{1}{239}}{4}$$

Ahora, el siguiente código cumple con la suma de esta funcion con un parámetro n

[7]:
```
from numpy import arctan
sum = 0
n = 16
for i in range (1, n + 1):
    sum += (4*arctan(1 / 5) - arctan(1/239)) / 4
print(sum)
print(abs(math.pi - sum))
```

```
3.1415926535897944
1.3322676295501878e-15
```

1.2.3 1.5.

a)¿Cuántas multiplicaciones y sumas se requieren para determinar una suma de la forma:

$$\sum_{i=1}^{n}\sum_{j=1}^{i} a_i b_j$$

Respuesta: Se requieren $\sum_{i=1}^{n} i$ multiplicaciones y $\frac{n(n+1)}{2}$ sumas en total.

b) Modifique la suma en la parte a) a un formato equivalente que reduzca el número de cálculos.

Respuesta:

1. Linealidad de sumatoria: $ = {i=1}^{n} (a_i {j=1}^{i}b_j)$
2. Suma de una progresión aritmética: $\sum_{i=1}^{n} a_i \cdot \frac{i(i+1)}{2}$
3. Reagrupación: $\sum_{i=1}^{n} \frac{i(i+1)}{2} a_i$

[8]:
```python
x = [1, 2, 3] # Lista a sumar de manera inversa
sum = 0
n = len(x)
for i in range (1, n + 1):
    sum += x[(n - i)]

print(sum)
```

```
6
```

1.3 2. Discuciones

2.2 Las ecuaciones (1.2) y (1.3) en la sección 1.2 proporcionan formas alternativas para las raíces 1 y 2 de 2 + + = 0. Construya un algoritmo con entrada a, b y c y salida x_1, x_2 que calcule las raíces x_1 y x_2 (que pueden ser iguales con conjugados complejos) mediante la mejor fórmula para cada raíz.

[9]:
```python
def raices(a : float, b : float, c : float) -> tuple[float, float] | float | tuple[complex, complex]:
    discriminante = b**2 - 4*a*c

    if (discriminante == 0):
        raiz = (-b + (discriminante)**0.5) / 2*a
        return raiz

    elif (discriminante > 0):
        raiz1 = (-b + (discriminante)**0.5) / 2*a
        raiz2 = (-b - (discriminante)**0.5) / 2*a
        return raiz1, raiz2

    else:
        raiz1 = complex(-b / 2*a, (abs(discriminante))**0.5 * 100 // 2*a / 100)
        raiz2 = complex(-b / 2*a, -(abs(discriminante))**0.5 * 100 // 2*a / 100)
        return raiz1, raiz2
```

[10]:
```python
resultado = raices(1.5, -2, 1)
print('La/s raiz/ices de la ecuacion cuadratica es/son: ', resultado)
```

```
La/s raiz/ices de la ecuacion cuadratica es/son:  ((1.5+1.05j), (1.5-1.065j))
```