ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA EN SISTEMAS

Nombre: Stiv Quishpe

Asignatura: Métodos Numéricos

Fecha de entrega: 10/11/2024

CONJUNTO DE EJERCICIOS 1.3

1. La serie de Maclaurin para 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 $\tan \pi/4 = 1$ para determinar el número n de términos de la serie que se necesita sumar para garantizar que $|4P_n(1) - \pi| < 10^{-3}$

$$|P_n(1) - \pi| < 10^{-3}$$

$$\arctan(1) = \frac{\pi}{4}$$

Por lo tanto,

$$\left|P_n(1) - \frac{\pi}{4}\right| < 10^{-3}$$

Como dice que el error de truncamiento de la serie se da por esta fórmula

$$|R_n(1)| \leq \frac{1}{2n+1}$$

Entonces lo que se quiere obtener es:

$$\frac{1}{2n+1} < 10^{-3}$$

$$2n + 1 > 1000$$

$$2n > 999$$

$$n > 499.5 \approx 500$$

R: Se necesitan al menos n = 500 términos para garantizar la suma.

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

$$|R_n(1)| \leq \frac{1}{2n+1} < 10^{-10}$$

$$2n + 1 > 10^{10}$$

$$n > \frac{10^{10} - 1}{2}$$

$$n > 5 \times 10^9 \approx 5{,}000{,}000{,}000$$

R: Se necesitan 5000000000 términos para obtener el grado de precisión.

2. 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 debe sumar para garantizar una aproximación π dentro de 10^{-3}.

Cálculo para $\left(\frac{1}{5}\right): \sum_{n=0}^{\infty} \frac{(-1)^n\left(\frac{1}{5}\right)^{2n+1}}{2n+1}$

Cálculo para $\left(\frac{1}{239}\right): \sum_{n=0}^{\infty} \frac{(-1)^n\left(\frac{1}{239}\right)^{2n+1}}{2n+1}$

Error de truncamiento y resolución de n para 1/5 y 1/239

- $\left|4R_n\left(\frac{1}{5}\right)\right| < 10^{-3}$
- $4 * \frac{\left(\frac{1}{5}\right)^{2n+1}}{2n+1} < 10^{-3}$
- $\frac{\left(\frac{1}{5}\right)^{2n+1}}{2n+1} < 2.5 \times 10^{-4}$
- $\left(\frac{1}{5}\right)^{2n+1} < (2n+1) * 2.5 \times 10^{-4}$
- $\left|R_n\left(\frac{1}{239}\right)\right| < 10^{-3}$

- $\frac{\left(\frac{1}{239}\right)^{2n+1}}{2n+1} < 10^{-3}$
- $\left(\frac{1}{239}\right)^{2n+1} < (2n+1) * 10^{-3}$

3.

a. ¿Cuántas multiplicaciones y sumas se requiere para determinar una suma de la forma $\sum_{i=1}^{n} \sum_{j=1}^{i} a_i b_j$?
Multiplicaciones: para cada par $(a_i b_j)$ se necesita una multiplicación. Serían n para a y m para b, dando un número total de: $n \times m$.
Sumas: Después de haber realizado las multiplicaciones, la suma requiere de los términos anteriores menos la primera suma que no necesita operación $n \times m - 1$.

b. Modifique la suma en la parte a) a un formato equivalente que reduzca el número de cálculos.
Para b_j se necesitan m-1 sumas y para a_i el mismo número que de multiplicaciones. Al juntar los dos análisis, otra variación de suma total es: $(m-1)+n$.

DISCUCIONES

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

```
import math, cmath
def root(a,b,c):
  d = b**2 - 4*a*c
  if d>0:
    x1 = (-b+ math.sqrt(d))/(2*a)
    x2 = (-b- math.sqrt(d))/(2*a)
  elif d == 0:
    x1 = -b/(2*a)
    x2 = -b/(2*a)
  else:
    x1 = (-b/(2*a)) + (cmath.sqrt(-d)/(2*a))
    x2 = (-b/(2*a)) - (cmath.sqrt(-d)/(2*a))
  return x1, x2
a = 6
b = 2
c = -2
print(root(a,b,c))
```