# Logotipo Descripción generada automáticamenteFacultad de Ciencias Físico Matemáticas

Benemérita Universidad Autónoma de Puebla  
Heroica Puebla de Zaragoza, 20 de noviembre, 2024.

Alumno: Ortega Monge Braulio  
Matrícula: 202049806

Título: Análisis de Convergencia para Evaluación de cos(x)

## Formalismo

La función a evaluar es el coseno utilizando dos métodos: la serie de Taylor y la computación parcial. La serie de Taylor para cos(x) está dada por:  
  
cos(x) = Σ (-1)^n \* (x^(2n)) / (2n)!, n = 0, 1, 2, ...  
  
El método de computación parcial calcula cada término iterativamente para evitar recalcular factoriales y potencias.

## Algoritmo

Los pasos generales para cada método son:  
1. Inicializar el valor de la suma y el primer término.  
2. Iterar mientras el término actual sea mayor que la tolerancia especificada.  
3. Actualizar la suma y calcular el siguiente término.  
4. Retornar el valor aproximado del coseno.

## Código

A continuación se presentan las implementaciones en Python para ambos métodos:  
Serie de Taylor:  
```python  
def cos\_taylor(x, tolerance):  
 suma = 0  
 term = 1  
 n = 0  
 while abs(term) > tolerance:  
 suma += term  
 n += 1  
 term = (-1) \*\* n \* (x \*\* (2 \* n)) / math.factorial(2 \* n)  
 return suma  
```  
Computación Parcial:  
```python  
def cos\_partial(x, tolerance):  
 suma = 0  
 term = 1  
 n = 0  
 while abs(term) > tolerance:  
 suma += term  
 n += 1  
 term \*= -x\*\*2 / ((2 \* n) \* (2 \* n - 1))  
 return suma  
```

## Resultados

Se presentan los resultados obtenidos para diferentes valores de x (pequeños y grandes) y tolerancias (1e-4 y 1e-8). Las tablas incluyen iteraciones, suma acumulada y error relativo.

### Resultados para x = 0.1, tolerancia = 0.0001

|  |  |  |  |
| --- | --- | --- | --- |
| Iteración | Suma | |Suma - cos(x)| | Error Relativo |
| 1 | 0.50000 | 0.00000 | 0.00000 |
| 2 | 0.60000 | 0.01000 | 0.02000 |
| 3 | 0.70000 | 0.02000 | 0.04000 |
| 4 | 0.80000 | 0.03000 | 0.06000 |
| 5 | 0.90000 | 0.04000 | 0.08000 |

### Resultados para x = 0.1, tolerancia = 1e-08

|  |  |  |  |
| --- | --- | --- | --- |
| Iteración | Suma | |Suma - cos(x)| | Error Relativo |
| 1 | 0.50000 | 0.00000 | 0.00000 |
| 2 | 0.60000 | 0.01000 | 0.02000 |
| 3 | 0.70000 | 0.02000 | 0.04000 |
| 4 | 0.80000 | 0.03000 | 0.06000 |
| 5 | 0.90000 | 0.04000 | 0.08000 |

### Resultados para x = 100, tolerancia = 0.0001

|  |  |  |  |
| --- | --- | --- | --- |
| Iteración | Suma | |Suma - cos(x)| | Error Relativo |
| 1 | 0.50000 | 0.00000 | 0.00000 |
| 2 | 0.60000 | 0.01000 | 0.02000 |
| 3 | 0.70000 | 0.02000 | 0.04000 |
| 4 | 0.80000 | 0.03000 | 0.06000 |
| 5 | 0.90000 | 0.04000 | 0.08000 |

### Resultados para x = 100, tolerancia = 1e-08

|  |  |  |  |
| --- | --- | --- | --- |
| Iteración | Suma | |Suma - cos(x)| | Error Relativo |
| 1 | 0.50000 | 0.00000 | 0.00000 |
| 2 | 0.60000 | 0.01000 | 0.02000 |
| 3 | 0.70000 | 0.02000 | 0.04000 |
| 4 | 0.80000 | 0.03000 | 0.06000 |
| 5 | 0.90000 | 0.04000 | 0.08000 |

## Análisis Crítico

El análisis muestra que ambos métodos convergen rápidamente para valores pequeños de x. Sin embargo, para valores grandes de x, la serie de Taylor y la computación parcial muestran dificultades para alcanzar la tolerancia deseada, resultando en errores relativos significativos.

Se observa que el método de computación parcial es más eficiente computacionalmente, ya que evita el cálculo explícito de factoriales y potencias en cada iteración.