Tarea 3

Nombre:Rommel Rivera

Utilice aritmética de corte de tres dígitos para calcular las siguientes sumas. Para cada parte| ¿qué método es más preciso y por qué?

a. $\sum_{i=1}^{10} \frac{1}{i^2}$

1. Método Descendente (Grandes a Pequeños)

$p_a^* = \frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \cdots + \frac{1}{10^2}$

i	$\frac{1}{i^2}$ (Corte de 3 Dígitos)	Suma Acumulada	Suma Redondeada (Corte)
1	1.00	1.00	1.00
2	0.250	1.00+0.250=1.25	1.25
3	0.111	1.25+0.111=1.361	1.36
4	0.0625	1.36+0.0625=1.4225	1.42
5	0.0400	1.42+0.0400=1.460	1.46
6	0.0277	1.46+0.0277=1.4877	1.48
7	0.0204	1.48+0.0204=1.5004	1.50
8	0.0156	1.50+0.0156=1.5156	1.51
9	0.0123	1.51+0.0123=1.5223	1.52
10	0.0100	1.52+0.0100=1.5300	1.53

2. Método Ascendente (Pequeños a Grandes)

i	$\frac{1}{i^2}$ (Corte de 3 Dígitos)	Suma Acumulada	Suma Redondeada (Corte)
10	0.0100	0.0100	0.0100
9	0.0123	0.0100+0.0123=0.0223	0.0223
8	0.0156	0.0223+0.0156=0.0379	0.0379
7	0.0204	0.0379+0.0204=0.0583	0.0583
6	0.0277	0.0583+0.0277=0.0860	0.0860
5	0.0400	0.0860+0.0400=0.1260	0.126
4	0.0625	0.126+0.0625=0.1885	0.188
3	0.111	0.188+0.111=0.299	0.299
2	0.250	0.299+0.250=0.549	0.549
1	1.00	0.549+1.00=1.549	1.54

3. Errores y Conclusión

Método	Aproximación ($p*$)	Error Absoluto (Ea)	Error Relativo (Er)
Descendente	1.53	0.01977	0.01275
Ascendente	1.54	0.00977	0.00630

El Método Ascendente (sumar de pequeño a grande) es el más preciso. Al sumar los términos pequeños primero, la suma acumulada crece lentamente.

b. $\sum_{i=1}^{10} \frac{1}{i^3}$

1. Método Descendente (Grandes a Pequeños)

i	i31 (Corte de 3 Dígitos)	Suma Acumulada Si−1+i31	Suma Redondeada (Corte)
1	1.00	1.00	1.00
2	0.125	1.00+0.125=1.125	1.12
3	0.0370	1.12+0.0370=1.1570	1.15
4	0.0156	1.15+0.0156=1.1656	1.16
5	0.00800	1.16+0.00800=1.16800	1.16
6	0.00462	1.16+0.00462=1.16462	1.16
7	0.00291	1.16+0.00291=1.16291	1.16
8	0.00195	1.16+0.00195=1.16195	1.16
9	0.00137	1.16+0.00137=1.16137	1.16
10	0.00100	1.16+0.00100=1.16100	1.16

2. Método Ascendente (Pequeños a Grandes)

i	i31 (Corte de 3 Dígitos)	Suma Acumulada Si−1+i31	Suma Redondeada (Corte)
10	0.00100	0.00100	0.00100
9	0.00137	0.00100+0.00137=0.00237	0.00237
8	0.00195	0.00237+0.00195=0.00432	0.00432
7	0.00291	0.00432+0.00291=0.00723	0.00723
6	0.00462	0.00723+0.00462=0.01185	0.0118
5	0.00800	0.0118+0.00800=0.0198	0.0198
4	0.0156	0.0198+0.0156=0.0354	0.0354
3	0.0370	0.0354+0.0370=0.0724	0.0724
2	0.125	0.0724+0.125=0.1974	0.197
1	1.00	0.197+1.00=1.197	1.19

3. Errores y Conclusión

Método	Aproximación (p∗)	Error Absoluto (Ea)	Error Relativo (Er)
Descendente	1.16	0.04084	0.03401
Ascendente	1.19	0.01084	0.00902

Nuevamente, el Método Ascendente (sumar de pequeño a grande) es significativamente más preciso.

2. La serie de Maclaurin para la función arcotangente converge para −1<x≤1 y está dada por

$$\arctan(1) = \frac{\pi}{4} = \sum_{i=1}^{\infty} (-1)^{i+1} \frac{1}{2i-1} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots$$

a. Utilice el hecho de que tan π 4⁄=1 para determinar el número n de términos de la serie que se necesita

sumar para garantizar que |4P_n (1)−π|<10^−3

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?

Multiplicando por 4, obtenemos la serie para π:

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

El error absoluto de la suma parcial de una serie alternante, $S_n = \sum_{i=1}^{n}(-1)^{i+1}a_i$, está acotado por el valor absoluto del primer término omitido (a_{n+1}):

$$|S - S_n| \leq a_{n+1}$$

En nuestro caso, $P_n(1)$ es la suma parcial hasta el término n, y $a_i = \frac{1}{2i-1}$.

$$|4P_n(1) - \pi| \leq 4 \cdot a_{n+1} = 4 \cdot \frac{1}{2(n+1)-1} = \frac{4}{2n+2-1} = \frac{4}{2n+1}$$

a. Determinar n para garantizar $\mathbf{|4P_n(1) - \pi| < 10^{-3}}$

$$\frac{4}{2n+1} < 10^{-3}$$

$$2n+1 > \frac{4}{10^{-3}}$$

$$2n+1 > 4000$$

$$2n > 3999$$

$$n > \frac{3999}{2}$$

$$n > 1999.5$$

Dado que n debe ser un número entero, redondeamos al siguiente entero:

$$\mathbf{n = 2000}$$

Se necesitan 2000 términos de la serie para garantizar que el error sea menor que 10^{-3}.

b. Determinar n para garantizar $\mathbf{|4P_n(1) - \pi| < 10^{-10}}$

Ahora queremos que el error esté acotado por 10^{-10}:

$$\frac{4}{2n+1} < 10^{-10}$$

Despejamos n:

$$2n+1 > \frac{4}{10^{-10}}$$

$$2n+1 > 4 \times 10^{10}$$

$$2n > 4 \times 10^{10} - 1$$

$$n > \frac{40,000,000,000 - 1}{2}$$

$$n > \frac{39,999,999,999}{2}$$

$$n > 19,999,999,999.5$$

$$\mathbf{n = 20,000,000,000}$$

Se necesitarían 20 mil millones de términos de la serie de Leibniz para obtener la precisión requerida por el lenguaje C++

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

Multiplicando por 4 para obtener π:

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

La serie de Maclaurin para $\arctan(x)$ es:

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

La serie es alternante, por lo que el error de truncamiento está acotado por el valor absoluto del primer término omitido.

$$P_n = 16\left(\sum_{i=1}^{n}(-1)^{i+1}\frac{(1/5)^{2i-1}}{2i-1}\right) - 4\left(\sum_{i=1}^{n}(-1)^{i+1}\frac{(1/239)^{2i-1}}{2i-1}\right)$$

El error absoluto de la aproximación es la suma de los errores de cada componente. Queremos que:

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

El error del término n está dado por el término $n+1$:

$$|\text{Error}| \leq \frac{x^{2(n+1)-1}}{2(n+1)-1} = \frac{x^{2n+1}}{2n+1}$$

Determinar n para el primer componente ($x = 1/5$)

El error de este componente debe ser menor que la mitad del error total, o menos. Ya que $x = 1/5$ domina el error total, nos enfocaremos en este término y asumiremos que el segundo error es despreciable.

Requerimos que:

$$16 \cdot \frac{(1/5)^{2n+1}}{2n+1} < 10^{-3}$$

n	Error Acotado: 16·2n+1(1/5)2n+1	¿Error <0.001?
1	16·3(1/5)3≈16·30.008≈0.0426	No
2	16·5(1/5)5≈16·50.00032≈0.001024	No
3	16·7(1/5)7≈16·70.0000128≈0.00002925	Sí

El número de términos necesarios para que el error de la primera parte sea menor a 10^{-3} (de hecho, mucho menor) es $n = 3$

Determinar n para el segundo componente ($x = 1/239$)

Ahora verificamos si $n = 3$ es suficiente para el segundo componente, cuyo error debe ser menor que 10^{-3}:

$$4 \cdot \frac{(1/239)^{2n+1}}{2n+1}$$

Para $n = 1$:

$$4 \cdot \frac{(1/239)^3}{3} \approx 4 \cdot \frac{0.000000073}{3} \approx 9.7 \times 10^{-8}$$

El número de términos n necesario para toda la fórmula se rige por el término que tiene el error más grande, que es $\arctan(1/5)$.

4. Compare los siguientes tres algoritmos. ¿Cuándo es correcto el algoritmo de la parte 1a?

Algoritmo a: Inicialización en Cero

- Paso 1: PRODUCT = 0
- Paso 2 (Iteración $i = 2$): PRODUCT = 0 * x_2 = 0
- Paso 2 (Iteración $i = 2$): PRODUCT = 0 * x_2 = 0

Conclusión: Una vez que PRODUCT es cero, cualquier multiplicación subsiguiente por x_i mantendrá el valor en cero. El algoritmo devuelve $\mathbf{0}$ independientemente de los valores de la lista de entrada.

Algoritmo b: Inicialización en Uno (Estándar)

- Paso 1: PRODUCT = 1
- Paso 2: Multiplica PRODUCT por cada x_i.

Conclusión: Este es el algoritmo estándar y correcto para calcular el producto de una lista. Si la lista contiene un cero, el producto final será correctamente cero.

Algoritmo c: Inicialización en Uno con Salida Anticipada (Optimizado)

- Paso 1: PRODUCT = 1
- Paso 2: Si encuentra $x_i = 0$, establece PRODUCT = 0 y termina inmediatamente (salida anticipada o early exit).

Conclusión: Este algoritmo es correcto y el más eficiente en el caso de que la lista contenga el valor cero. Permite evitar $n - i$ multiplicaciones innecesarias, ya que el producto total será cero de todos modos.

El Algoritmo (a) solo es matemáticamente correcto si, y solo si, el producto exacto de la lista de entrada es $\mathbf{0}$.

5.