LABORATORY 4: NUMERICAL ERRORS

Computación a Gran Escala (Curso 2022/2023)

Autores:
Miguel García González
Belén Vivas García

Máster:
Máster Universitario en Ingeniería Informática
(MUII)

03/12/2022

Miguel García González – miguel.garcia02@estudiante.uam.es
Belén Vivas García – belen.vivas@estudiante.uam.es

Índice

Índice de figuras

Miguel García González – miguel.garcia02@estudiante.uam.es
Belén Vivas García – belen.vivas@estudiante.uam.es

Ejercicio 1

En este ejercicio analizaremos los resultados de la serie infinita:

$$S = \sum_{i=1}^{\infty} \frac{1}{i^4} = \frac{\pi^4}{90}$$

cuando se suman los términos de forma ascendente (de i=1 hasta infinito) y de forma descendente (de infinito hasta i=1).

Hemos escrito un programa en Python llamado `sumSeries.py`, donde comparamos ambas funciones y obtenemos diferentes gráficas y resultados.

Ejecutamos el programa, obteniendo:

```
belen@belen-linux:~/Documents/informatica/master/cge/p4/exercise_1$ python3.7 sumSeries.py
1000000 first terms in ascending order: 1.08232323371086103236
1000000 first terms in descending order: 1.08232323371113814403
Real result: 1.08232323371113792199

The descending order result is closer to the real one

n value from which different values are obtained: 13
```

Figura 1: Resultado sumSeries.py

En primer lugar, evaluamos los resultados de ambas series con los 1.000.000 primeros términos (n=1.000.000), ya que es un valor suficientemente grande para aproximarse hacia infinito. Vemos que los resultados empiezan a diferir en el doceavo dígito decimal, y que el resultado de la serie en orden descendente está más cerca del valor real de la serie, calculado como $\pi^4/90$.

Los tres resultados tienen una diferencia muy pequeña entre sí, pero suficiente para considerar que las sumas no se están realizando del todo correctamente en ambas series, especialmente en la ascendente.

Esto se debe al error de redondeo de la coma flotante: el sistema de coma flotante tiene un número limitado de dígitos de precisión por lo que, al realizar una operación que involucre un número grande y otro bastante más pequeño, no habrá precisión suficiente para representar el resultado, y este se redondeará al resultado más próximo, es decir, el número grande. Por eso podemos decir que los números muy pequeños "no se tienen en cuenta" para la suma.

En la serie ascendente, se empiezan sumando cantidades más grandes y se terminan sumando cantidades más pequeñas:

$$\frac{1}{1^4} + \frac{1}{2^4} + \frac{1}{3^4} + \frac{1}{4^4} + \cdots + \frac{1}{1000^4} + \cdots$$

Y en la descendente se comienzan sumando cantidades muy pequeñas y se termina por las más grandes:

$$\ldots + \frac{1}{1000^4} + \cdots + \frac{1}{4^4} + \frac{1}{3^4} + \frac{1}{2^4} + \frac{1}{1^4}$$

Miguel García González – miguel.garcia02@estudiante.uam.es
Belén Vivas García – belen.vivas@estudiante.uam.es

En la serie ascendente, al irse sumando primero cantidades más grandes, cuando se llega a un punto en la serie en el que las cantidades ya son demasiado pequeñas para que se pueda representar el resultado, este se terminará redondeando siempre a lo que ya llevemos sumado.

Sin embargo, en la serie descendente, al empezar primero con cantidades pequeñas, estas sí se irán sumando al principio, formando resultados un poco más grandes que sí tendrán impacto al sumarse con los números más grandes y contribuirán al resultado final.

Por tanto, la serie que ofrece resultados más precisos es la **descendente.**

Por esta misma razón, la propiedad asociativa de la suma **no se cumple** al cambiar el orden de los operandos (que es en el fondo lo que estamos haciendo). Al hacer el sumatorio, los números se van evaluando de izquierda a derecha, en la ascendente:

$$(((grande + pequeño) + pequeño) + pequeño)$$

Y en la descendente:

$$(((pequeño + pequeño) + pequeño) + grande)$$

Por otro lado, hemos representado los resultados gráficamente. En las siguientes gráficas podemos ver la diferencia de los resultados de ambas series (en valor absoluto) variando el número de términos que evaluamos:

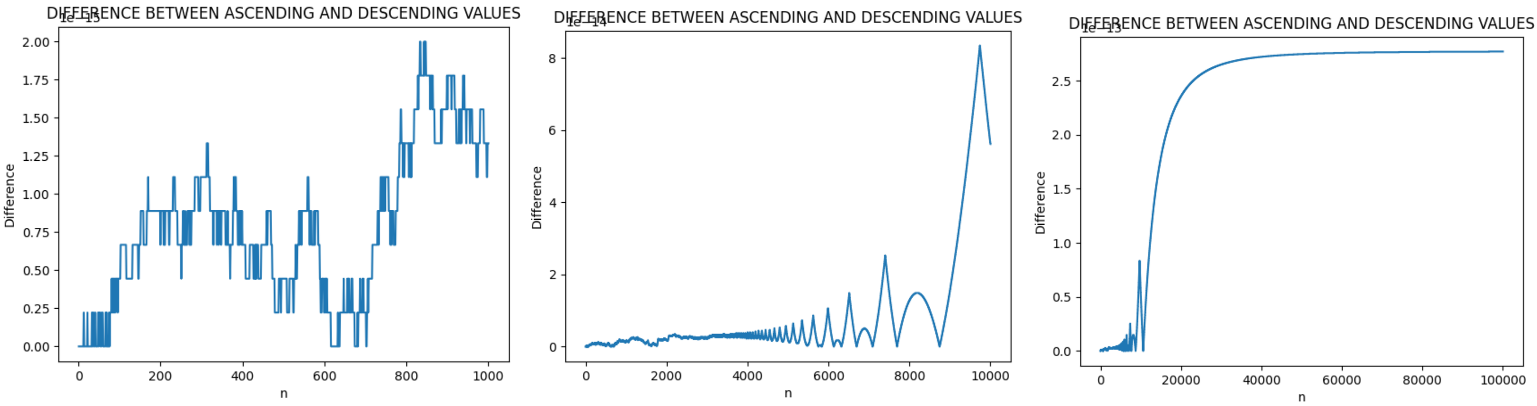

Figura 2: Gráficas diferencias resultados sumSeriesA y sumSeriesD

En la primera gráfica hemos representado hasta los 1.000 primeros términos, en la segunda hasta los 10.000 primeros y en la tercera hasta 100.000. Además, hemos visto que el valor de n a partir del cual los resultados empiezan a ser diferentes es **13**.

Hay ciertos valores de n para los cuales ambos resultados vuelven a ser iguales, y parece que esto se repite periódicamente hasta n=150.000 aproximadamente, y a partir de ahí ya se estabiliza, siendo siempre diferentes.

Esto se puede deber a que, hasta cierto valor de n, cuando esta no es muy grande, hay ciertos términos que, al hacer la división, dan un resultado algo más exacto con relativamente pocos decimales, y se tienen en cuenta para el redondeo en la serie ascendente, haciendo que en algunos casos la diferencia con la descendente sea menor o incluso 0.

Por último, hemos representado la diferencia entre cada valor y el valor real:

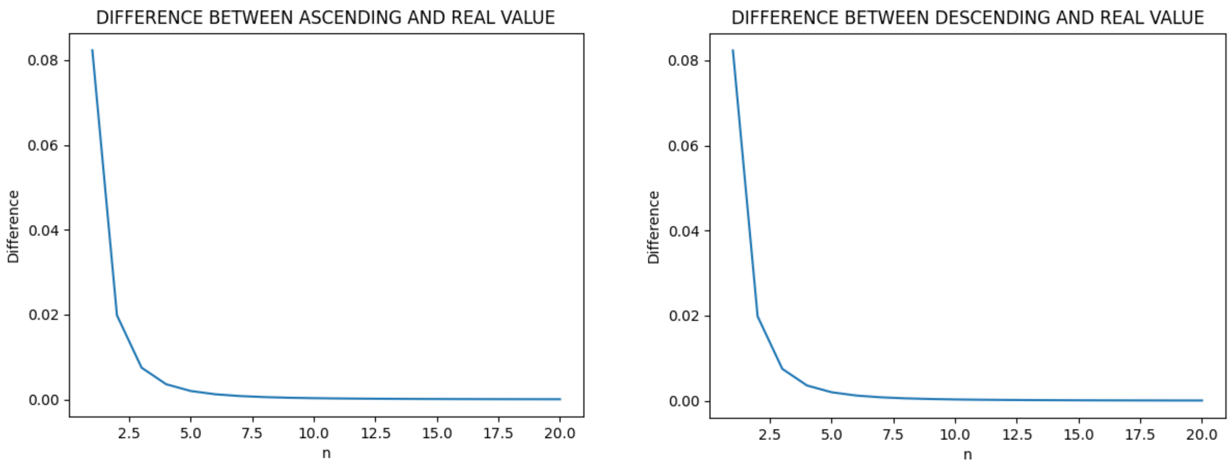

Figura 3: Diferencias entre los dos valores y el valor real

En la primera gráfica vemos cómo va evolucionando la diferencia entre el valor ascendente y el valor real según aumentamos n, y en la segunda vemos la diferencia entre el valor descendente y el real.

En ambos casos, con pocos términos, hasta 10 aproximadamente, hay una diferencia mayor, especialmente con muy pocos términos, ya que necesitamos que la serie tienda a infinito para conseguir un resultado lo suficientemente aproximado a $\pi^4/90$. A partir de 10 términos la diferencia ya es muy pequeña, teniendo que imprimir muchos decimales para notarla, como hemos visto al principio del ejercicio. Lógicamente la serie descendente al final, con un n muy grande, tendrá una diferencia menor con el valor real que la ascendente.

Ejercicio 2

En este problema tenemos tres expresiones para obtener las raíces de la ecuación cuadrática:

$$ax^2 + bx + c = 0$$

Para mostrar la equivalencia entre ellas, obtenemos en primer lugar la segunda expresión a partir de la primera, racionalizando:

$$\frac{-b + \sqrt{b^2 - 4ac}}{2a} = \left(\frac{-b + \sqrt{b^2 - 4ac}}{2a}\right)\left(\frac{-b - \sqrt{b^2 - 4ac}}{-b - \sqrt{b^2 - 4ac}}\right) = \frac{b^2 - (b^2 - 4ac)}{2a\left(-b - \sqrt{b^2 - 4ac}\right)}$$

$$= \frac{4ac}{2a(-b - \sqrt{b^2 - 4ac})} = \frac{2c}{-b - \sqrt{b^2 - 4ac}}$$

Y para el signo contrario:

$$\frac{-b - \sqrt{b^2 - 4ac}}{2a} = \left(\frac{-b - \sqrt{b^2 - 4ac}}{2a}\right)\left(\frac{-b + \sqrt{b^2 - 4ac}}{-b + \sqrt{b^2 - 4ac}}\right) = \frac{b^2 - (b^2 - 4ac)}{2a\left(-b + \sqrt{b^2 - 4ac}\right)}$$

$$= \frac{4ac}{2a(-b + \sqrt{b^2 - 4ac})} = \frac{2c}{-b + \sqrt{b^2 - 4ac}}$$

Obteniendo así la expresión final equivalente:

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = \frac{2c}{-b \pm \sqrt{b^2 - 4ac}}$$

Por otro lado, de la última expresión:

$$q = \frac{-1}{2}\left(b + sign(b)\sqrt{b^2 - 4ac}\right)$$

Podemos obtener la equivalencia con las dos anteriores mediante sus raíces:

$$x_1 = \frac{q}{a} = \frac{\frac{-1}{2}\left(b + sign(b)\sqrt{b^2 - 4ac}\right)}{a} = \frac{-\left(b + sign(b)\sqrt{b^2 - 4ac}\right)}{2a}$$

$$= \frac{-b - sign(b)\sqrt{b^2 - 4ac}}{2a}$$

$$x_2 = \frac{c}{q} = \frac{c}{\frac{-1}{2}\left(b + sign(b)\sqrt{b^2 - 4ac}\right)} = \frac{2c}{-\left(b + sign(b)\sqrt{b^2 - 4ac}\right)}$$

$$= \frac{2c}{-b - sign(b)\sqrt{b^2 - 4ac}}$$

En cuanto a la situación planteada, el problema con las dos primeras expresiones viene cuando se tiene que realizar la resta de b con el discriminante. El sistema de coma flotante sufre una falta de precisión al restar dos cantidades muy similares y, si b^2 es bastante más grande que $|ac|$, entonces $\sqrt{b^2 - 4ac}$ y b serán números muy parecidos, y al restarlos se notará esa pérdida de precisión.

Hemos escrito un programa llamado `roots.py` donde evaluamos los resultados de las tres expresiones introduciendo un b grande y a y c pequeños.

Lo ejecutamos con a=-20, b=40300 y c=10, obteniendo:

```
belen@belen-linux:~/Documents/informatica/master/cge/p4/exercise_2$ python3.7 roots.py
Quadratic expression 1:
x1 = 2015.00024813892741804011
x2 = -0.00024813892723614118
Quadratic expression 2:
x1 = 2015.00024832530812091136
x2 = -0.00024813892725909315
Quadratic expression 3:
x1 = 2015.00024813892741804011
x2 = -0.00024813892725909315
```

Figura 4: Resultado roots.py

La expresión cuadrática 1 se corresponde con: $\frac{-b\pm\sqrt{b^2-4ac}}{2a}$, con x_1= $\frac{-b-\sqrt{b^2-4ac}}{2a}$ y x_2= $\frac{-b+\sqrt{b^2-4ac}}{2a}$

La expresión cuadrática 2 se corresponde con: $\frac{2c}{-b\pm\sqrt{b^2-4ac}}$, con x_1= $\frac{2c}{-b+\sqrt{b^2-4ac}}$ y x_2= $\frac{2c}{-b-\sqrt{b^2-4ac}}$

La expresión cuadrática 3 se corresponde con: $q = \frac{-1}{2}\left(b + sign(b)\sqrt{b^2 - 4ac}\right)$, con x_1= $\frac{q}{a}$ y x_2= $\frac{c}{q}$

En el resultado podemos ver que el valor de la raíz x_1 de la primera expresión es igual que el x_1 de la tercera, siendo x_2 distinto; y el valor de la raíz x_2 de la segunda expresión es igual que el x_2 de la tercera, siendo x_1 distinto.

El resultado correcto es el de la **expresión cuadrática 3**, por el siguiente argumento:

Sabemos que, en la expresión 1, x_2 va a dar un resultado incorrecto, ya que en el numerador se van a restar dos cantidades prácticamente iguales: -40300+40300.00993, dando lugar a la

Miguel García González – miguel.garcia02@estudiante.uam.es
Belén Vivas García – belen.vivas@estudiante.uam.es

pérdida de precisión. Sin embargo, x_1 va a dar un resultado correcto, ya que estos términos no se sustraen, sino que se suman (-40300-40300.00993).

Por el contrario, en la expresión 2, el resultado incorrecto va a ser x_1, donde se produce la resta, siendo correcto x_2, donde se suman.

La expresión 3, por otra parte, calcula las raíces eliminando este problema, determinando primero la variante que no produce cancelación, ya que las dos expresiones cuadráticas anteriores tienen signos opuestos en el radical para las mismas raíces, por lo que se puede calcular la primera raíz como en la primera expresión, y la segunda raíz como en la segunda, obteniendo así siempre el resultado correcto.