– Práctica 1 –

Introducción a Python y aritmética de la máquina

1. Escribir una función Python de nombre `sumanveces` que tome como parámetros de entrada un número real a y un número natural n y devuelva la suma de a n veces, calculada con un bucle de tipo `for`. Repetir el ejercicio utilizando un bucle de tipo `while`. Aplicar el programa para $a = 0,1$ y distintos valores de n. Observar y comentar los resultados obtenidos a medida que aumenta n.

2. Escribir un script que calcule la precisión de la máquina ε para los números en coma flotante en Python, utilizando que $\varepsilon = \min\{x \in \mathcal{F} : 1 + x > 1\}$.

3. Representar gráficamente con Python la función $f(x) = x - e^{-x}$ en el intervalo $[-2, 2]$.

4. **Aproximación del número e como límite de una sucesión.** Este ejercicio se basa en que

$$\lim_{n\to\infty} \left(1 + \frac{1}{n}\right)^n = e.$$

Escribir una función Python de nombre `aproxe` que tome como entrada n y devuelva la aproximación de e dada por $\left(1 + \frac{1}{n}\right)^n$, así como el error absoluto cometido. Para calcular el valor "exacto" de e, utilizar la función `exp` del módulo `numpy`. Observar y comentar el comportamiento de la aproximación cuando aumenta n.

5. **Cálculo de una serie.** Escribir una función Python de nombre `sumaparcial` que tome como parámetro de entrada n y calcule la n-ésima suma parcial de la serie $\sum_{k=1}^{\infty} \frac{1}{\sqrt{k}}$, es decir,

$$S_n = \sum_{k=1}^{n} \frac{1}{\sqrt{k}}.$$

El programa debe mostrar en pantalla tanto el valor de n como el de la suma parcial S_n.

Observar que la serie es divergente. ¿Para qué valor de n se obtiene que la suma parcial es mayor que 50 con el programa realizado?

6. Escribir un script en Python que represente gráficamente los valores de S_n del problema anterior con respecto a n, para $n \in [1, 100]$.

7. **Cálculo de una serie y error.** Escribir una función Python que tome como entrada n y calcule la n-ésima suma parcial de la serie $\sum_{k=1}^{\infty} \frac{1}{k(k+1)}$, es decir,

$$S_n = \sum_{k=1}^{n} \frac{1}{k(k+1)}.$$

La suma exacta de la serie es 1. El programa debe mostrar en pantalla los valores de n, de la suma parcial S_n y del error $|1 - S_n|$.

Es fácil probar la igualdad

$$Sn = 1 - \frac{1}{n+1}.$$

Escribir un segundo programa para calcular la suma parcial n-ésima basada en esta última fórmula, con el mismo parámetro de entrada y las mismas impresiones en pantalla.

Probar ambos programas con $n = 10^j$, con $j = 3,\ 5$ y 7. Probar también sólo el segundo programa con $n = 10^{20}$. Comentar los resultados.

8. **Aproximación del número e mediante la serie de Taylor.** Otra forma posible de definir e es la siguiente:

$$e = \exp(1) = \sum_{k=0}^{\infty} \frac{1}{k!},$$

lo que sugiere aproximar e mediante una suma parcial de la serie, esto es,

$$e \approx \sum_{k=0}^{n} \frac{1}{k!},$$

para n suficientemente grande.

Escribir una función Python que calcule, para n dado, esta aproximación del número e. Observar el comportamiento de las aproximaciones conforme aumenta el valor de n. Comparar la calidad de las aproximaciones con las obtenidas en el ejercicio anterior en que se aproximaba e como límite de una sucesión.

9. **Aproximación de la funcion $\exp(x)$ mediante la serie de Taylor.** A partir de la identidad

$$\exp(x) = \sum_{k=0}^{\infty} \frac{x^k}{k!}, \qquad x \in \mathbb{R},$$

escribir una función Python que tome como parámetros de entrada un natural n y un número real x y calcule la aproximación de e^x dada por la n-ésima suma parcial

$$\exp(x) \approx S_n(x) = \sum_{k=0}^{n} \frac{x^k}{k!}, \qquad x \in \mathbb{R}.$$

El programa debe imprimir en pantalla el valor aproximado y el error $|\exp(x) - S_n(x)|$. Probar el programa para $x = 0, 1, 5, 10, -1, -5$ y -10. Comentar los resultados.

10. Escribir una función Python que tome como entrada una lista numérica y dé como resultado la suma de los elementos de la lista y su media aritmética. Comparar los resultados con las funciones predefinidas `sum` y `mean` del módulo `numpy`.