Lista de ejercicios

1. **Normas matriciales.** Se quiere estimar la norma 2 de una matriz $\boldsymbol{A} \in \mathbb{R}^{3\times 3}$ como el máximo del valor $\|\boldsymbol{Ax}\|_2/\|\boldsymbol{x}\|_2$ entre varios vectores $\boldsymbol{x} \in \mathbb{R}^3$ no nulos generados al azar. Hacer un programa que reciba una matriz A y luego

   - genere los primeros 100 términos de la siguiente sucesión:

     $$s_1 = 0, \quad s_{k+1} = \max\left\{s_k, \frac{\|\boldsymbol{Ax}_k\|_2}{\|\boldsymbol{x}_k\|_2}\right\}$$

     donde los $x_k \in \mathbb{R}^3$ son vectores no nulos generados al azar en la bola unitaria: $B = \{\boldsymbol{x} : \|\boldsymbol{x}\|_2 \leq 1\}$.
   - Grafique la sucesión calculada, junto con el valor exacto de la norma de la matriz.

   Recordar que la norma 2 de un vector `v` puede calcularse con el comando `np.linalg.norm(v)`, mientras que la norma 2 de una matriz `A` se puede obtener con `np.linalg.norm(A,2)`. Tener en cuenta que los vectores generados al azar (comando `np.random.random`) tienen coordenadas en el intervalo $[0, 1]$ y por lo tanto abarcan sólo el primer octante en $\mathbb{R}^3$.

2. **Resolución de ecuaciones.**

   (a) Escribir un programa que implemente el método de Jacobi para la resolución de un sistema lineal $\boldsymbol{Ax} = \boldsymbol{b}$, con las siguientes condiciones:
   - que al inicio calcule el radio espectral del método y que termine si es mayor o igual a 1,
   - que finalice si el método se estaciona,
   - que finalice si se excede cierto tope de iteraciones.

   (b) Testear el programa desarrollado en el item anterior para los sistemas

   $$\begin{pmatrix} 3 & 1 & 1 \\ 2 & 6 & 1 \\ 1 & 1 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 5 \\ 9 \\ 6 \end{pmatrix}, \quad \begin{pmatrix} 5 & 7 & 6 & 5 \\ 7 & 10 & 8 & 7 \\ 6 & 8 & 10 & 9 \\ 5 & 7 & 9 & 10 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 23 \\ 32 \\ 33 \\ 31 \end{pmatrix}$$

   y analizar los resultados.

3. **Resolución de ecuaciones.**

   (a) Implementar un programa que reciba como input una función f, su derivada f', un punto inicial x_0, una tolerancia ε y un entero N y aplique el método de Newton-Raphson para buscar una raíz de f a partir de x_0. El programa debe finalizar cuando $|x_n - x_{n-1}| < \varepsilon$ o cuando llega al paso N. Si no se alcanza la convergencia luego de N pasos, imprimir un mensaje de error.

   (b) Para $f(x) = x^{15} - 2$ implementando el programa del item anterior, hallar una aproximación del cero de la función comenzando con $x_0 = 1$, usando una tolerancia de 10^{-3}.

4. **Ecuaciones diferenciales.** Escribir un programa que implemente el método de Euler explícito para aproximar numéricamente la solución $x(t)$ de la siguiente ecuación diferencial en el intervalo $[t_0, t_F]$:

$$\begin{cases} \dot{x}(t) = f(t, x(t)), \\ x(t_0) = x_0, \end{cases}$$

¿Qué parámetros debe recibir y qué información debe devolver este programa para que la aproximación obtenida pueda graficarse?

5. **Ecuaciones diferenciales.** Se quiere verificar numéricamente el orden de convergencia de los métodos de Euler y Taylor de orden 2. Para ello: resolver numéricamente el problema

$$\begin{cases} \dot{x}(t) = x(t), \\ x(t_0) = 1, \end{cases}$$

en el intervalo $[0, 1]$ con ambos métodos, tomando $h = 0.1,\ 0.05,\ 0.01,\ 0.005,\ 0.001$ y 0.0005.

Obtener la solución exacta y para cada h, calcular el error que se comete al aproximar $x(1)$: $E_N = |x(1) - x_N|$. Graficar $\log(E_N)$ en función de $\log(h)$. ¿Qué se espera ver? ¿El resultado es consistente con el esperado?

6. **Cuadrados mínimos.**

   (a) Implementar una función que reciba vectores de datos x, y y un vector de pesos positivos w y devuelva la función $f(x) = ax + b$ que minimiza el error

   $$\sum_{0 \leq i \leq n} w_i (y_i - f(x_i))^2.$$

   (b) Testear el programa para la siguiente tabla de datos

x	1	2	3	4	5
y	3	5.01	7.02	9.1	11.1
w	1	0.5	1	0.5	0.5

7. **Integración numérica.**

   (a) Sea $a = x_0, \ldots, x_n = b$ una partición regular de $[a, b]$. Implementar un programa que reciba una función f, los límites del intervalo $[a, b]$ y un parámetro n y que mediante la regla de trapecios devuelva un valor aproximado de $\int_a^b f$, partiendo $[a, b]$ en n intervalos.

   (b) Emplear el programa del item anterior para calcular

   $$\int_0^1 \frac{1}{\sqrt{1 + x^2}} dx, \quad \text{con } n = 4.$$