

## Práctica 3: Introducción a Phyton y a herramientas de matemática simbólica

1. Mediante eliminación Gaussiana, hallar la inversa de la matriz

$$\begin{pmatrix} 1 & 2 & -1 \\ 0 & 3 & -2 \\ 2 & -1 & 1 \end{pmatrix}.$$

Comprobar el resultado mediante el programa *maxima*.

2. Hallar el polinomio de Taylor de orden 2 centrado en  $x=2$  de la función  $F(x) = e^{-2x} \ln(3x)$ . Verificar el resultado mediante el programa *maxima*.
3. Hallar autovalores y autovectores de la matriz

$$\begin{pmatrix} 5 & 4 \\ 1 & 2 \end{pmatrix}.$$

Rehacer el problema mediante el programa *maxima*.

4. Hallar la integral general de la ecuación diferencial lineal homogénea

$$y'' + 4y' - 5y = 0$$

(ayuda: buscar soluciones de la forma  $e^{kx}$ , con  $k$  elegido adecuadamente). Verificar la solución encontrada mediante *maxima*.

5. Obtener las raíces de los polinomios  $P(x) = 3x^3 - 4x^2 + 2x - 5$  y  $Q(x) = x^4 + 2x^2 - 6x - 8$ , mediante el uso de *Octave*. Factorizar el polinomio  $Z(x) = x^4 + 8x^3 - 3x^2 - 62x + 56$  luego de hallar sus raíces con *Octave*.
6. Verificar que la matriz

$$\begin{pmatrix} 0 & 2 & 1 & 6 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

es nilpotente y encontrar el orden de nilpotencia mediante el uso de *Octave*.

7. Las llamadas *matrices de Pauli*, están dadas por

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

$$\sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$$

$$\sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

Con la ayuda de *Octave*, verificar que satisfacen  $\{S_x, S_y\} = 0$ ,  $\{S_x, S_z\} = 0$  y  $\{S_y, S_z\} = 0$ , siendo  $\{A, B\} = AB + BA$  el operador *anticommutador*.

8. Mediante el módulo *matplotlib* de Python, graficar la función

$$z = x^2 - y^2$$

y graficar algunas de sus curvas de nivel.

9. Hacer un script de Python que lea de un archivo conteniendo una tabla de valores x-y y realice un ajuste lineal por cuadrados mínimos, encontrando y graficando la recta que mejor ajusta a los datos.
10. Mediante el módulo Tkinter de Python, escribir una interfaz gráfica de usuario con cuatro campos de entrada de texto para la introducción de las partes real e imaginaria de dos números complejos y que muestre en una etiqueta de texto la suma de ambos números al clickear cierto botón llamado 'Sumar'.
11. Crear un script de Python que resuelva y grafique la solución al problema de valores iniciales:

$$\begin{cases} \frac{dy}{dx} = xe^{-y} \\ y(0) = 1 \end{cases}$$

12. Crear un script de Python que resuelva y grafique (en el rango  $t \in [0, 100]$ ) la solución al problema de oscilaciones forzadas amortiguadas ( $\gamma$  es la constante de amortiguamiento) con una inhomogeneidad (fuerza externa) periódica de frecuencia  $\omega$ , siendo  $\omega_0$  la frecuencia natural del oscilador (se asumen todas las unidades dadas en el sistema internacional):

$$\begin{cases} \frac{d^2y}{dt^2} + \gamma \frac{dy}{dt} + \omega_0^2 y = B \sin(\omega t) \\ y(0) = 1 \\ \frac{dy}{dt}(0) = 0 \end{cases}$$

Estudiar los casos en ausencia de fuerza externa ( $B = 0$ ), con y sin amortiguamiento. Luego, en presencia de fuerza externa para  $\omega_0 \ll \omega$  y para  $\omega_0 \gg \omega$ . Que ocurre cuando  $\omega_0 \sim \omega$ ? Como referencia, poner valores de  $\omega_0$  y  $\omega$  en el orden de 1 y valores de  $\gamma$  en el orden de 0,1.