Programación 2021

Guía 6: Numpy: vectores y matrices. Ajuste de funciones.

16 de Setiembre 2021

Antes de comenzar los problemas genere un nuevo directorio `guia6` donde trabajará y guardará todos los programas y archivos que se producirán en este práctico.

Problema 1: Implemente un programa que reciba un vector con posiciones en el espacio $\mathbb{R}^3$ y las refleje respecto del plano $y - z$.

Problema 2: Genere una matriz aleatoria con densidad de probabilidad normal de media 0 y desviación estandard 5 que contenga (100,2) componentes. Encuentre las distancias entre los pares de puntos y guardelas en un array. Determine la distancia máxima.

Problema 3: Genere una matriz que represente una grilla en el plano para $0 < x < 1$ y $0 < y < 1$ con una resolución de 0.1 en y y de 0.05 en x.

Problema 4: Implementar un programa que realice operaciones con matrices:

(a) Realizar una función que dada una matriz de n por n, responda si la matriz es simétrica o no (Salida True/False).

(b) Realizar una función que verifique si una matriz de dimensión n es ortogonal, es decir si el producto de ella por su traspuesta es la matriz identidad.

(c) Realizar una función que determine la norma de Frobenius (raiz cuadrada de la suma de los cuadrados de las componentes i.e. $\|\mathbf{A}\| = \sqrt{\sum_i \sum_j A_{i,j}^2}$ de una matriz dada.

(d) Realizar una función que determine la distancia de Frobenius entre dos matrices: $\|\mathbf{A}, \mathbf{B}\| = \sqrt{\sum_i \sum_j (A_{i,j} - B_{i,j})^2}$

Problema 5: Se puede rotar un punto alrededor del eje z mediante la siguiente transformación

$$\boldsymbol{r}' = \begin{pmatrix} cos(\theta) & -sin(\theta) & 0 \\ sin(\theta & cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \end{pmatrix} \tag{1}$$

donde $\boldsymbol{r}'$ es el vector (matrix columna) cuyas componentes rotadas son (x', y', z'). Implementar un programa que dado un vector $\boldsymbol{r}$ con sus componentes en el espacio, de obtenga el vector rotado alrededor del eje z en un ángulo θ.

Por otro lado, la traslación de la posición en la dirección z se puede implementar como

$$\boldsymbol{r}' = \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ D \end{pmatrix} \tag{2}$$

Asumiendo que $\theta = \omega t$(movimiento circular uniforme) y que $D = Vt$(MRU) implementar un programa que obtenga la trayectoria de la particula en el espacio.

Problema 6: Desarrolle una función cuyos argumentos de entrada sea un array de N pares de variables (x, y) y que determine la ordenada y pendiente de la regresión lineal utilizando las fórmulas vistas en el teórico [Problema de desarrollo se recomienda NO usar funciones existentes en las librerías ni googlear].

1. Desarrollo de la función.
2. Genere un conjunto de $N = 100$ datos datos por $v(t) = v_0 - gt + 0{,}2\epsilon$ con $v_0 = 4{,}5m/s$, $g = 9{,}81m/s^2$ y que evalue la velocidad cada $0{,}25s$ ϵ es un número aleatorio que representa el error observacional de media 0 y varianza 1 (np.random.normal).
3. Utilize la función de regresión lineal con los datos generados y estime el valor de v_0 y g.
4. Compare el error obtenido en la estimación con el caso de $N = 10$ y con el caso de $N = 1000$.

Problema 7: Genere 20 valores aleatorios de la ordenada y con una distribución uniforme entre 0 y 2 usando np.random.uniform asigne estos a un intervalo de x entre 0,1 asumiendo los puntos estan equidistribuidos en x.

1. Utilice los esplines cúbicos para generar 100 puntos de la función en el intervalo 0,1. Use: scipy.interpolate.splrep o scipy.interpolate.CubicSpline
2. Realice el mismo procedimiento con la interpolación lineal entre los puntos originales. En este caso desarrolle su propio interpolador.

Problema 8: Encuentre la línea recta que ajusta a los siguientes datos por cuadrados mínimos. Use la función np.polyfit

t	x
0	3.076
0.5	2.810
1.0	2.588
1.5	2.297
2.0	1.981
2.5	1.912
3.0	1.653
3.5	1.478
4.0	1.399
4.5	1.018
5.0	0.794

Problema 9: Determine los parametros a y b tal que satisface $f(x) = ae^{bx}$ ajustan los datos a través de cuadrados mínimos. Use la función np.polyfit

t	x
1.2	7.5
2.8	16.1
4.3	38.9
5.4	67.0
6.8	146.6
7.9	266.2

Problema 10: Use el método de bisección para hallar la menor solución positiva de la ecuación $2x = \tan(x)$. ¿Cuántos pasos serán necesarios para garantizar que el error sea menor a 10^{-3}?

Problema 11: Sea $f(x) = (x+2)(x+1)^2x(x-1)^3(x-2)$.¿A cuál raíz de f converge el método de bisección en los siguientes intervalos?
(a) $[-1{,}5, 2{,}5]$, (b) $[-0{,}5, 2{,}4]$, (c) $[-0{,}5, 3]$, (d) $[-3, -0{,}5]$

Problema 12: Método de Newton-Raphson y de bisección-secante.

1. Desarrolle una función que determine la raiz de una función usando el método de Newton-Raphson, con argumentos de entrada, un valor inicial, la función, la derivada y la tolerancia.
2. Desarrolle una función bisección-secante determine la raiz de una función usando el método de biseccion adaptado para qeu en lugar de tomar el punto del medio del intervalo se elija la raiz de la recta que pasa por los puntos $(a, f(a))$ y $(b, f(b))$.
3. La difracción Fraunhofer producida por una rendija tiene el máximo principal en el origen x=0, y los máximos secundarios son las raíces de la ecuación trascendente $\tan x - x = 0$. Determine el valor de x que cumple que $\tan x - x = 0$, utilizando el método de Newton-Raphson con valor inicial $x_0 = 4$ y $x_0 = 4{,}6$ analice y saque conclusiones.
4. Use el metodo de bisección usando de intervalo inicial 4 y 5.
5. Compare la velocidad de convergencia con el metodo de bisección, de bisección-secante y de Newton-Raphson.

Problema 13: Considere el problema de fichas de las damas en un tablero. Respecto del borde del tablero las fichas tienen una ubicación i, j donde i indica la distancia horizontal a la que esta la ficha respecto del borde superior izquierdo del tablero y j indica la distancia en la dirección vertical (de arriba hacia abajo). Cada ficha posee un número entero que identifica a la misma.

1. Implemente una función que busque una ficha que posee un número particular dentro del conjunto, e indique como respuesta en que lugar del tablero se encuentra dicha ficha. Usando bucles (Sin funciones numpy).
2. Realice el item anterior con funciones numpy.
3. Realice una función distancia (en unidades de ubicaciones) que ingresen dos números de fichas y salga la distancia a la que se encuentran las fichas.

Problema 14: Implementar un programa que dada una malla de enteros $M_{i,j}$ de nxn, obtenga la suma de los segundos vecinos de cualquier elemento i,j. Los segundos vecinos son los que se encuentran en las diagonales del cuadrado que se genera alrededor del punto de interés.

Problema 15: Realice un programa interactivo que permita jugar al tateti.