Práctica 0. Introducción a Matlab (I)

SIstemas de control inteligente

Jorge Revenga;Ángel

2023

Índice

[Ejercicio 1. Matrices y vectores. 2](#_Toc147315772)

[Código: 2](#_Toc147315773)

[Ejecución: 2](#_Toc147315774)

[Ejercicio 2. Matrices y vectores. 3](#_Toc147315775)

[Código: 3](#_Toc147315776)

[Ejecución: 4](#_Toc147315777)

[Ejercicio 3. Matrices y vectores. 5](#_Toc147315778)

[Código: 5](#_Toc147315779)

[Ejecución: 5](#_Toc147315780)

[Ejercicio 4. Tiempo de cómputo y representación gráfica. 5](#_Toc147315781)

[Código: 5](#_Toc147315782)

[Ejecución: 5](#_Toc147315783)

[Ejercicio 5. Representación gráfica en 3D 5](#_Toc147315784)

[Código: 5](#_Toc147315785)

[Ejecución: 5](#_Toc147315786)

[Ejercicio 6. Sistemas lineales. 5](#_Toc147315787)

[Código: 5](#_Toc147315788)

[Ejecución: 5](#_Toc147315789)

[Ejercicio 7. Polinomios. 5](#_Toc147315790)

[Código: 5](#_Toc147315791)

[Ejecución: 5](#_Toc147315792)

## Ejercicio 1. Matrices y vectores.

### Código:

% Paso 1: Crear la matriz A y el vector v

A = [1 2; 3 4; 5 6; 7 8];

v = [14; 16; 18; 20];

% Paso 2: Obtener y visualizar la matriz B concatenando A y v

B = [A v];

% Paso 3: Obtener y visualizar un vector fila concatenando las filas de B

vector\_fila = [B(1,:) B(2,:) B(3,:) B(4,:)]

% Opción 2:

vector\_fila = reshape(B', 1, []);

% Paso 4: Obtener y visualizar un vector columna concatenando las columnas de B

vector\_columna = [B(:,1); B(:,2); B(:,3)]

% Opción 2:

vector\_columna = reshape(B, [], 1);

% Visualizar los resultados

disp('Matriz B:');

disp(B);

disp('Vector fila resultante de concatenar filas de B:');

disp(vector\_fila);

disp('Vector columna resultante de concatenar columnas de B:');

disp(vector\_columna);

### Ejecución:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

## Ejercicio 2. Matrices y vectores.

### Código:

% Paso 1: Solicitar al usuario el tamaño de la matriz cuadrada

n = input('Indique el tamaño de la matriz: ');

% Paso 2: Generar una matriz aleatoria de tamaño n x n

matriz = rand(n);

% a) Mostrar la matriz generada

disp('Matriz generada:');

disp(matriz);

% b) Obtener una segunda matriz con las columnas impares de la matriz inicial

matriz\_impares = matriz(:, 1:2:end);

disp('Segunda matriz con columnas impares:');

disp(matriz\_impares);

% c) Obtener y mostrar los elementos de la diagonal de la matriz generada

diagonal = diag(matriz);

disp('Elementos de la diagonal:');

disp(diagonal);

% d) Calcular y graficar el máximo, mínimo, medio y varianza de cada fila

maximos = max(matriz, [], 2);

minimos = min(matriz, [], 2);

medios = mean(matriz, 2);

varianzas = var(matriz, 0, 2);

% Crear un gráfico de barras para los valores calculados

figure;

bar([maximos, minimos, medios, varianzas]);

title('Valores por fila');

xlabel('Número de fila');

ylabel('Valor');

legend('Máximo', 'Mínimo', 'Medio', 'Varianza');

### Ejecución:

Tabla

Descripción generada automáticamente

Gráfico, Gráfico de barras

Descripción generada automáticamente

## Ejercicio 3. Matrices y vectores.

### Código:

% Función para generar y rellenar una matriz

function Matriz = IntroducirMatriz(Dimensiones)

if numel(Dimensiones) == 1 % Si solo se proporciona un número, asumimos una matriz cuadrada

filas = Dimensiones;

cols = Dimensiones;

elseif numel(Dimensiones) == 2

filas = Dimensiones(1);

cols = Dimensiones(2);

else

error('Formato de dimensiones incorrecto. Debe ser [filas cols].');

end

Matriz = zeros(filas, cols);

for i = 1:filas

for j = 1:cols

fprintf('Ingrese el valor de la posición (%d, %d) (deje en blanco para aleatorio): ', i, j);

valor = input('','s'); % Leer como cadena de caracteres

if isempty(valor) || isempty(strtrim(valor)) % Si el usuario ingresa un valor vacío

Matriz(i, j) = rand(); % Rellenar con valor aleatorio

else

Matriz(i, j) = str2double(valor); % Convertir entrada a número

end

end

end

end

% Paso 1: Solicitar las dimensiones de las matrices al usuario

dimensiones\_A = input('Introduce las dimensiones de la matriz A en formato [filas cols]: ');

dimensiones\_B = input('Introduce las dimensiones de la matriz B en formato [filas cols]: ');

% Paso 2: Generar las matrices A y B

A = IntroducirMatriz(dimensiones\_A);

B = IntroducirMatriz(dimensiones\_B);

% Paso 4: Realizar los cálculos y mostrar los resultados

disp('Matriz A:');

disp(A);

disp('Matriz B:');

disp(B);

disp('Transpuesta de A:');

disp(A');

disp('Transpuesta de B:');

disp(B');

if isequal(size(A), size(B)) && size(A,1) == size(A,2)

disp('Determinante de A: ');

disp(det(A));

else

disp('A no es una matriz cuadrada, no se puede calcular el determinante.');

end

if isequal(size(B), size(A)) && size(B,1) == size(B,2)

disp('Determinante de B: ');

disp(det(B));

else

disp('B no es una matriz cuadrada, no se puede calcular el determinante.');

end

disp(['Rango de A: ' num2str(rank(A))]);

disp(['Rango de B: ' num2str(rank(B))]);

disp('Producto matricial A\*B:');

if size(A,2) == size(B,1)

disp(A \* B);

else

disp('No se puede calcular el producto matricial A\*B debido a dimensiones incompatibles.');

end

disp('Producto elemento a elemento A.\*B:');

if isequal(size(A), size(B))

disp(A .\* B);

else

disp('No se puede calcular el producto elemento a elemento A.\*B debido a dimensiones incompatibles.');

end

vector\_fila = [A(1, :), B(1, :)];

disp('Vector fila obtenido concatenando la primera fila de A y B:');

disp(vector\_fila);

vector\_columna = [A(:, 1); B(:, 1)];

disp('Vector columna obtenido concatenando la primera columna de A y B:');

disp(vector\_columna);

### 

### Ejecución:

Imagen que contiene ventana, texto, parado, grande

Descripción generada automáticamente

Tabla

Descripción generada automáticamente

## Ejercicio 4. Tiempo de cómputo y representación gráfica.

### Código:

% Inicializar matrices para almacenar los tiempos de procesamiento

tiempo\_rango = [];

tiempo\_determinante = [];

% Bucle para calcular los tiempos

for n = 1:25

% Generar una matriz aleatoria de tamaño n x n

matriz = rand(n, n);

% Calcular el tiempo para el cálculo del rango

tic;

rango = rank(matriz);

tiempo\_rango(n) = toc;

% Calcular el tiempo para el cálculo del determinante

tic;

determinante = det(matriz);

tiempo\_determinante(n) = toc;

end

% Crear una figura para representar los tiempos

figure;

plot(tiempo\_rango);

hold on;

plot(tiempo\_determinante);

xlabel('Tamaño de la matriz');

ylabel('Tiempo (segundos)');

title('Tiempo de cálculo del rango y determinante en función del tamaño de la matriz');

legend('Tiempo de cálculo del rango', 'Tiempo de cálculo del determinante');

hold off;

### Ejecución:

Gráfico, Histograma

Descripción generada automáticamente

## Ejercicio 5. Representación gráfica en 3D

### Código:

% Definir el rango de valores para x y y

x = -5:0.1:5;

y = -5:0.1:5;

% Crear una malla de valores para x y y

[X, Y] = meshgrid(x, y);

% Calcular la función z en función de x y y

Z = Y .\* sin(pi \* X / 10) + 5 \* cos((X.^2 + Y.^2) / 8) + cos(X + Y) \* cos(3 \* X - Y);

figure;

subplot(2, 2, 1);

surf(X, Y, Z);

title('Superficie');

xlabel('Eje X');

ylabel('Eje Y');

subplot(2, 2, 2);

mesh(X, Y, Z);

title('Superficie en Forma de Malla');

xlabel('Eje X');

ylabel('Eje Y');

subplot(2, 2, 3);

contourf(X, Y, Z);

colorbar;

title('Contorno con Barra de Color');

xlabel('Eje X');

ylabel('Eje Y');

% Ajustar tamaño a la pantalla

set(gcf, 'Position', get(0, 'Screensize'));

### Ejecución:

Diagrama

Descripción generada automáticamente con confianza media

## Ejercicio 6. Sistemas lineales.

### Código:

% Definir las matrices A y b

A = [0 2 10 7; 2 7 7 1; 1 9 0 5; 4 0 0 6; 2 8 4 1; 10 5 0 3; 2 6 4 0; 1 1 9 3; 6 4 8 2; 0 3 0 9];

As = [90; 59; 15; 10; 80; 17; 93; 51; 41; 76];

b = [0.110 0 1 0; 0 3.260 0 1; 0.425 0 1 0; 0 3.574 0 1; 0.739 0 1 0; 0 3.888 0 1; 1.054 0 1 0; 0 4.202 0 1; 1.368 0 1 0; 0 4.516 0 1];

bs = [317; 237; 319; 239; 321; 241; 323; 243; 325; 245];

% 1. Obtener el número de condición de la matriz A

c = cond(A);

disp(['Número de condición de A: ' num2str(c)]);

% 2. Resolver los sistemas de ecuaciones A = As y b = bs

X = linsolve(A, As);

disp('Solución del sistema de ecuaciones A:');

disp(X);

Y = linsolve(b,bs);

disp('Solución del sistema de ecuaciones b (sin ruido):');

disp(Y)

% 3. Añadir ruido a la matriz b y resolver el sistema

r = normrnd(0, 1, 10, 1);

b2 = bs + r;

% 4. Resolver el sistema de ecuaciones b = bs

B = b;

Y = linsolve(B, b2);

disp('Solución del sistema de ecuaciones b (con ruido):');

disp(Y);

### Ejecución:

Imagen que contiene Texto

Descripción generada automáticamente

## Ejercicio 7. Polinomios.

### Código:

function [solucion, reales, complejas] = raices(poli\_1, poli\_2)

% Recoge los arrays con los que se crean los polinomios

p1 = poly2sym(poli\_1);

p2 = poly2sym(poli\_2);

% Solicita al usuario a cuál de los polinomios o al producto se aplicará la solución

fprintf('Elija a cuál de los polinomios o al producto desea aplicar la solución:\n');

fprintf('1. Polinomio 1\n');

fprintf('2. Polinomio 2\n');

fprintf('3. Producto de los polinomios\n');

choice = input('Ingrese el número correspondiente: ');

switch choice

case 1

% Calcular raíces del primer polinomio

solucion = roots(poli\_1);

case 2

% Calcular raíces del segundo polinomio

solucion = roots(poli\_2);

case 3

% Calcular raíces del producto de los polinomios

prod\_poli = conv(poli\_1, poli\_2);

solucion = roots(prod\_poli);

otherwise

error('Opción no válida. Debe elegir 1, 2 o 3.');

end

% Clasificar las raíces

reales = sum(isreal(solucion));

complejas = length(solucion) - reales;

% Representar las raíces en el plano complejo

figure;

hold on;

plot(real(solucion), imag(solucion), 'ro');

xlabel('Parte Real');

ylabel('Parte Imaginaria');

title('Ubicación de las raíces en el plano complejo');

hold off;

end

[solucion, reales, complejas] = raices([1 2 2], [1 3]);

fprintf('Raíces: ');

disp(solucion);

fprintf('Raíces reales: %d\n', reales);

fprintf('Raíces complejas: %d\n', complejas);

### Ejecución:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Gráfico, Gráfico de dispersión

Descripción generada automáticamente