eman ta zabal zazu

Universidad del País Vasco

Euskal Herriko Unibertsitatea

Material de estudio

OCW 2018: *Utilizando Mathematica como apoyo al cálculo algebraico en los grados de Ingeniería*

Tema 2. Cálculo matricial

Equipo docente del curso

Martín Yagüe, Luis

Unzueta Inchaurbe, Aitziber

Arrospide Zabala, Eneko

García Ramiro, María Begoña

Soto Merino, Juan Carlos

Alonso González, Erik

Departamento de Matemática Aplicada

Escuela de Ingeniería de Bilbao, Edificio II-I

OCW OpenCourseWare

TEMA 2. CÁLCULO MATRICIAL

Listas

Definición

Son elementos de almacenamiento estructurado de datos. Pueden contener grupos de valores o, a su vez, otras listas; por ejemplo, una matriz se define como una lista de listas.

- **List**. Define una lista de los elementos introducidos como argumentos.

Los elementos se presentan entre llaves y separados por comas. Una lista también puede definirse de esa manera, tecleando los datos entre llaves y separados por comas.

Ejemplo. Definición de dos listas y una lista de listas (matriz).

```
lista1 = List[1, 2, 3]
```

```
{1, 2, 3}
```

```
lista2 = {4, 5, 6}
```

```
{4, 5, 6}
```

```
lista = {lista1, lista2}
```

```
{{1, 2, 3}, {4, 5, 6}}
```

Generación

Las listas son ampliamente utilizadas en *Mathematica*. Hay diversas formas de generar listas numéricas además de la mera introducción de los elementos.

- **Range**. Genera una lista de números desde el indicado como primer argumento (por defecto, 1) hasta el especificado como segundo argumento. El tercer argumento indica la separación entre elementos (por defecto, 1).

Ejemplo. Generación de listas con `Range`.

```
Range[10] (* lista de números enteros del 1 al 10 *)
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

```
Range[5, 10] (* lista de números enteros del 5 al 10 *)
```

```
{5, 6, 7, 8, 9, 10}
```

```
Range[1, 10, 1.5] (* lista de números del 1 al 10 a intervalos de 1.5 unidades *)
```

```
{1., 2.5, 4., 5.5, 7., 8.5, 10.}
```

- **Table**. Genera una lista de los valores de una *expresión* (primer argumento) en función de la variación de un determinado índice (segundo argumento: $\{i, i_{min}, i_{max}\}$).

Ejemplo. Generación de listas con `Table`.

```
l1 = Table[{0, 1, 2}, 3] (* repetición de la lista {0,1,2} tres veces *)
```

```
{{0, 1, 2}, {0, 1, 2}, {0, 1, 2}}
```

```
pot2 = Table[2^i, {i, 5}] (* cinco primeras potencias enteras de 2 *)
```

```
{2, 4, 8, 16, 32}
```

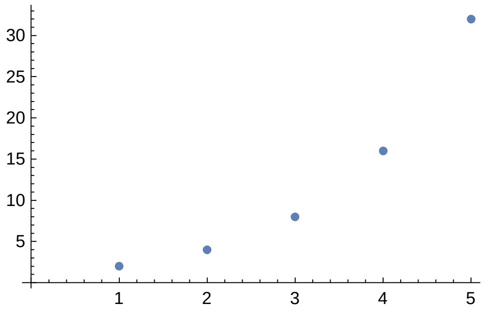
```
potpar = Table[2^j, {j, 2, 14, 2}] (* siete primeras potencias enteras pares de 2 *)
```

```
{4, 16, 64, 256, 1024, 4096, 16384}
```

Los elementos de una lista pueden representarse (**ListPlot**) como puntos del plano. Se considera como primera coordenada la posición que ocupan dentro de la lista siendo la segunda coordenada el valor del elemento.

Ejemplo. Generación de listas con `Table`.

```
ListPlot[pot2, PlotStyle → PointSize[Medium]]
```



Obtención de los elementos de una lista

Diversas funciones permiten acceder a una lista para obtener los elementos almacenados.

Se recomienda asignar las listas a variables.

- **Part**. Recupera la parte especificada de la lista indicada como primer argumento. De forma equivalente puede escribirse el nombre de la lista (variable a la que se asignó) y la parte deseada encerrada en un doble corchete.
- **First**. Obtención del primer elemento de una lista.
- **Last**. Obtención del último elemento de una lista.

Ejemplo. Obtención de elementos de las listas creadas en el ejemplo anterior.

```
(* primer elemento de la lista l1 de dos formas diferentes *)
```

```
First[l1]
```

```
{0, 1, 2}
```

```
l1[[1]]
```

```
{0, 1, 2}
```

```
(* cuarto elemento de la lista pot2 de dos formas diferentes *)
```

```
Part[pot2, 4]
```

```
16
```

```
pot2[[4]]
```

```
16
```

```
potpar[[2 ;; 4]] (* elementos del segundo al cuarto de la lista potpar *)
```

```
{16, 64, 256}
```

```
l1[[1, 3]] (* tercer elemento de la primera lista almacenada en l1 (ver ejemplo anterior) *)
```

```
2
```

```
(* último elemento de la lista l1 de dos formas diferentes *)
```

```
Last[l1]
```

```
{0, 1, 2}
```

```
l1[[Length[l1]]]
```

```
{0, 1, 2}
```

- **Length**. Indica el número de elementos de una lista.

Matrices $m\times n$

Definición y conceptos básicos

Una matriz se define como una lista de listas.

Entonces, una matriz:

- es una lista
- sus elementos (filas) son sublistas de la lista y se separan mediante comas
- los elementos de cada fila (listas) se introducen entre llaves separados por comas
- la salida se presenta como una lista en la que cada fila es, también, otra lista

Puede obtenerse la representación de una matriz en su forma estándar con la función **MatrixForm**. Resulta útil para comprobar la correcta introducción de datos pero su salida es una imagen por lo que no se puede utilizar en cálculos posteriores.

Los elementos de una matriz se pueden extraer como en cualquier lista:

- elemento de la fila i y columna j de una matriz : **nombre_matriz[[i,j]]**
- obtención de la fila i–ésima de una matriz : **nombre_matriz[[i]]**

También, puede usarse la función **Part**.

Ejemplo. Definición de la matriz: $A=\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & -6 \\ 7 & 8 & -9 \end{pmatrix} \in \mathbb{M}_{3\times 3}$

```
A = {{1, 2, 3}, {4, 5, -6}, {7, 8, -9}}
```

```
{{1, 2, 3}, {4, 5, -6}, {7, 8, -9}}
```

```
MatrixForm[A] (* salida en forma matricial *)
```

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & -6 \\ 7 & 8 & -9 \end{pmatrix}$$

A[[2, 3]] (* obtención del tercer elemento de la segunda fila: a_{23}=-6 *)

```
-6
```

```
A[[2]] (* obtención de la segunda fila *)
```

```
{4, 5, -6}
```

Casos particulares:

- Matriz columna. Se interpreta como un vector y se introduce como una lista.
- Matriz fila. Es válida la definición general teniendo en cuenta que sólo hay una fila.
- Matriz identidad. La función **IdentityMatrix** genera una matriz identidad del orden especificado en su argumento.
- Matriz diagonal. La función **DiagonalMatrix** genera una matriz diagonal con los n elementos contenidos en una lista que constituye su argumento.

Ejemplo. Definición de matrices fila y columna.

```
fila = {{1, 2, 3}}
```

```
{{1, 2, 3}}
```

```
MatrixForm[fila]
```

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

```
col = {1, 2, 3}
```

```
{1, 2, 3}
```

```
MatrixForm[col]
```

$$\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

Ejemplo. Definición de matrices identidad y diagonal.

```
uni4 = IdentityMatrix[4] (* matriz identidad de orden 4 *)
```

```
{{1, 0, 0, 0}, {0, 1, 0, 0}, {0, 0, 1, 0}, {0, 0, 0, 1}}
```

```
MatrixForm[uni4]
```

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

```
diag3 = DiagonalMatrix[col] (* matriz diagonal *)
```

```
{{1, 0, 0}, {0, 2, 0}, {0, 0, 3}}
```

```
MatrixForm[diag3]
```

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

Mathematica suministra una función, **Flatten**, que permite obtener un vector de $m \cdot n$ coordenadas a partir de una matriz de orden $m \times n$ "desanidando" las listas que componen dicha matriz.

Ejemplo. Obtención de un vector de 9 coordenadas a partir de la matriz $A \in \mathbb{M}_{3\times 3}$ del ejemplo anterior.

```
v3 = Flatten[A]
```

```
{1, 2, 3, 4, 5, -6, 7, 8, -9}
```

Generación

Mathematica suministra funciones que permiten obtener matrices del orden que se establezca y cuyos elementos son generados mediante los cálculos que se especifican en una expresión que se introduce como argumento.

- **Table**. Genera una matriz de orden $m\times n$ cuyos elementos se calculan según una *expresión* indicada como primer argumento; además, también, se especifican como argumentos los índices cuya variación se usa en la generación de dichos elementos (sintaxis: $\{i, i_{min}, i_{max}\}$). El primer índice se usa en la generación de filas y el segundo en el de columnas.

Ejemplo. Definición de una matriz de orden 2×3 cuyos elementos se calculan sumando el número de la fila y el de la columna que ocupan en la matriz.

```
mat = Table[i + j, {i, 2}, {j, 3}];

m = MatrixForm[mat]
```

$$\begin{pmatrix} 2 & 3 & 4 \\ 3 & 4 & 5 \end{pmatrix}$$

- **Array**. Crea una matriz de orden $m\times n$ cuyos elementos se calculan en función de una expresión, indicada como primer argumento, que depende de los valores de los índices señalados entre llaves como segundo argumento.

Ejemplo. Definición de una matriz de orden 2×3 cuyos elementos se calculan sumando el número de la fila (*#1*) y el de la columna (*#2*) que ocupan en la matriz.

```
mat2 = Array[#1 + #2 &, {2, 3}]

{{2, 3, 4}, {3, 4, 5}}
```

Operaciones con matrices

Introducción

En este apartado se describen diferentes operaciones que pueden realizarse con matrices y las funciones necesarias para ello.

Advertencias:

- Es conveniente comprobar el orden de cada matriz para que pueda realizarse la operación correspondiente; el programa avisa con un mensaje de error cuando se intenta operar matrices cuyos órdenes son incompatibles con dicha operación.
- La salida de **MatrixForm** es considerada como una representación gráfica, no puede usarse para operar con otras matrices aunque el programa no suministra ningún mensaje de error.

Ejemplo. Errores al operar.

```
{A = {{1, 2, 3}, {4, 5, -6}, {7, 8, -9}}, mat = Table[i + j, {i, 2}, {j, 3}], m = MatrixForm[mat]};

A + mat (* suma de dos matrices de diferentes órdenes *)

... Thread: Objects of unequal length in {{1, 2, 3}, {4, 5, -6}, {7, 8, -9}} + {{2, 3, 4}, {3, 4, 5}} cannot be combined.

{{2, 3, 4}, {3, 4, 5}} + {{1, 2, 3}, {4, 5, -6}, {7, 8, -9}}

m + mat (* suma de una matriz y su representación matricial *)
```

$$\left\{\left\{2+\begin{pmatrix} 2 & 3 & 4 \\ 3 & 4 & 5 \end{pmatrix}, 3+\begin{pmatrix} 2 & 3 & 4 \\ 3 & 4 & 5 \end{pmatrix}, 4+\begin{pmatrix} 2 & 3 & 4 \\ 3 & 4 & 5 \end{pmatrix}\right\}, \left\{3+\begin{pmatrix} 2 & 3 & 4 \\ 3 & 4 & 5 \end{pmatrix}, 4+\begin{pmatrix} 2 & 3 & 4 \\ 3 & 4 & 5 \end{pmatrix}, 5+\begin{pmatrix} 2 & 3 & 4 \\ 3 & 4 & 5 \end{pmatrix}\right\}\right\}$$

Suma, producto por escalar y producto de matrices

Operadores:

- Suma de matrices: +
- Producto de matriz por escalar: *
- Producto de matrices: .

Ejemplo. Operaciones con matrices definidas en ejemplos de los apartados anteriores.

```
A + diag3 (* suma de matrices *)
```

```
{{2, 2, 3}, {4, 7, -6}, {7, 8, -6}}
```

```
2 * uni4 (* producto de matriz por escalar *)
```

```
{{2, 0, 0, 0}, {0, 2, 0, 0}, {0, 0, 2, 0}, {0, 0, 0, 2}}
```

```
3 * A - 2 * diag3
```

```
{{1, 6, 9}, {12, 11, -18}, {21, 24, -33}}
```

```
mat2.A (* producto de matrices *)
```

```
{{42, 51, -48}, {54, 66, -60}}
```

```
A.mat2 (* producto de matrices erróneo *)
```

Dot: Tensors {{1, 2, 3}, {4, 5, −6}, {7, 8, −9}} and {{2, 3, 4}, {3, 4, 5}} have incompatible shapes.

```
{{1, 2, 3}, {4, 5, -6}, {7, 8, -9}}.{{2, 3, 4}, {3, 4, 5}}
```

Orden de una matriz

Número de filas y de columnas de una matriz.

- **Dimensions**. Devuelve una lista con las dimensiones de la matriz dada como argumento; es decir, número de filas y número de columnas.

Ejemplo. Orden de una matriz.

Dimensions[uni4] (* uni4∈$\mathbb{M}_{4\times4}$ *)

```
{4, 4}
```

Dimensions[mat] (* mat∈$\mathbb{M}_{2\times3}$ *)

```
{2, 3}
```

Potencia de una matriz

La potencia n-ésima consiste en multiplicar n veces una matriz por sí misma.

- **MatrixPower**. Calcula la potencia n-ésima de una matriz.

Ejemplo. Potencia de una matriz (se consideran matrices ya definidas).

```
MatrixPower[diag3, 3]
```

```
{{1, 0, 0}, {0, 8, 0}, {0, 0, 27}}
```

```
MatrixPower[A, 5]
```

```
{{-6228, -5652, 11772}, {8946, 8793, -15444}, {13968, 13842, -23868}}
```

```
MatrixPower[A, 5] // MatrixForm
```

$$\begin{pmatrix} -6228 & -5652 & 11772 \\ 8946 & 8793 & -15444 \\ 13968 & 13842 & -23868 \end{pmatrix}$$

Inversa de una matriz

Una matriz A cuadrada de orden n se llama regular ó no singular si existe otra matriz cuadrada del mismo orden, A^{-1}, tal que se verifica: $A \cdot A^{-1} = A^{-1} \cdot A = \mathbb{I}_n$. Siendo $\mathbb{I}_n$ la matriz identidad de orden n.

- **Inverse**. Calcula la inversa de una matriz regular. Si el argumento especificado no es una matriz regular o cuadrada el programa devuelve un mensaje de error.

Ejemplo. Cálculo de la inversa de una matriz cuadrada.

```
A = {{1, 2, 3}, {4, 5, -6}, {7, 8, -9}}; (* definición de una matriz cuadrada *)
```

```
Inverse[A] // MatrixForm
```

$$\begin{pmatrix} -\frac{1}{6} & -\frac{7}{3} & \frac{3}{2} \\ \frac{1}{3} & \frac{5}{3} & -1 \\ \frac{1}{6} & -\frac{1}{3} & \frac{1}{6} \end{pmatrix}$$

```
A.Inverse[A] // MatrixForm
```

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

```
Inverse[A].A == IdentityMatrix[3]
```

```
True
```

```
mat = Table[i + j, {i, 2}, {j, 3}]; (* definición de una no matriz cuadrada *)
```

```
Inverse[mat] // MatrixForm (* el programa devuelve un mensaje de error en este caso *)
```

... Inverse: Argument {{2, 3, 4}, {3, 4, 5}} at position 1 is not a non-empty square matrix.

```
Inverse[{{2, 3, 4}, {3, 4, 5}}]
```

Determinante de una matriz

A toda matriz cuadrada A se le asocia un número real llamado determinante de la matriz y denotado $det(A) = |A|$.

- **Det**. Calcula el determinante de una matriz cuadrada. Si el argumento especificado no es una matriz cuadrada el programa devuelve un mensaje de error.

Ejemplo. Cálculo del determinante de una matriz cuadrada.

```
Det[A] (* matriz cuadrada A definida previamente *)
```

```
-18
```

```
Det[mat] (* matriz no cuadrada mat definida previamente: el programa devuelve un mensaje de error *)
```

... Det: Argument {{2, 3, 4}, {3, 4, 5}} at position 1 is not a non-empty square matrix.

```
Det[{{2, 3, 4}, {3, 4, 5}}]
```

Transposición de una matriz

Intercambio de las filas y columnas de una matriz.

- **Transpose**. Calcula la transpuesta de una matriz dada como argumento.

Ejemplo. Transpuesta de una matriz.

```
A = {{1, 2, 3}, {4, 5, -6}, {7, 8, -9}};
```

```
MatrixForm[A] (* matriz cuadrada A definida previamente *)
```

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & -6 \\ 7 & 8 & -9 \end{pmatrix}$$

```
Transpose[A] // MatrixForm
```

$$\begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & -6 & -9 \end{pmatrix}$$

Extracción de columnas

Recuperación de la i–ésima columna de una matriz.

El proceso es análogo al de extracción de una fila; en este caso, previamente, debe transponerse la matriz dada.

Ejemplo. Varias formas de extraer la columna 2 de la matriz A.

```
A = {{1, 2, 3}, {4, 5, -6}, {7, 8, -9}};
```

```
MatrixForm[A]
```

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & -6 \\ 7 & 8 & -9 \end{pmatrix}$$

```
at = Transpose[A]
```

{{1, 4, 7}, {2, 5, 8}, {3, -6, -9}}

```
MatrixForm[at]
```

$$\begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & -6 & -9 \end{pmatrix}$$

```
at[[2]]
```

{2, 5, 8}

```
Part[at, 2]
```

{2, 5, 8}

```
Part[Transpose[A], 2]
```

{2, 5, 8}

Extracción de submatrices

Obtención de una submatriz que consta de determinadas filas y columnas de una matriz dada.

Siendo **expr** el nombre de la matriz dada : **expr**[[{f_1, f_2, ..., f_n}, {c_1, c_2, ..., c_m}]].

En este caso, $\boldsymbol{f}_i$ denota la fila i-ésima y $\boldsymbol{c}_j$ la j-ésima columna de la matriz dada.

Ejemplo. Extracción de una submatriz de la matriz A.

```
MatrixForm[A] (* matriz cuadrada A definida previamente *)
```

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & -6 \\ 7 & 8 & -9 \end{pmatrix}$$

```
A[[{1, 2}, {2, 3}]] // MatrixForm (* submatriz de filas 1 y 2 y columnas 2 y 3 *)
```

$$\begin{pmatrix} 2 & 3 \\ 5 & -6 \end{pmatrix}$$

Rango de una matriz

Matriz sin parámetros

El rango de una matriz es el máximo número de filas ó columnas linealmente independientes.

- **MatrixRank**. Determina el rango de una matriz.

Ejemplo. Rango de una matriz.

```
{MatrixRank[A], MatrixRank[mat]} (* matrices A y mat definidas previamente *)
```

```
{3, 2}
```

- **RowReduce.** Devuelve la forma reducida por filas (escalonada) de la matriz dada de forma que su rango es el número de filas no nulas.

Ejemplo. Forma reducida por filas de una matriz.

```
RowReduce[Transpose[mat]] // MatrixForm
```

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

```
RowReduce[A] // MatrixForm (* matriz cuadrada A definida previamente *)
```

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

Matriz con parámetros

Para determinar el rango de una matriz en función de los valores de uno o varios parámetros se van a usar diversas funciones:

- **Minors.** Obtiene todos los menores de orden k (segundo argumento) de la matriz dada como primer argumento. Un menor es el determinante de una submatriz; por lo tanto, se dan los determinantes de todas las submatrices del orden indicado.

Ejemplo. Obtención de los menores de orden dos de una matriz.

```
Minors[A, 2] (* matriz A definida previamente *)
```

```
{{-3, -18, -27}, {-6, -30, -42}, {-3, 6, 3}}
```

- **Solve.** Resuelve, cuando es posible, el sistema de ecuaciones planteado (primer argumento, *expr*) en las variables indicadas (segundo argumento, *vars*).

Ejemplo. Resolución de ecuaciones y sistemas de ecuaciones.

```
sol1 = Solve[x^2 - 2 x - 3 == 0, x]
```

{{x → -1}, {x → 3}}

```
Solve[{x + y + z == 1, x + y - 3 z == -7, x - z == 0}, {x, y, z}]
```

{{x → 2, y → -3, z → 2}}

- **Length.** Devuelve el número de elementos de que consta el único argumento de la función.

Ejemplo. Obtención del número de elementos de una lista.

```
Length[sol1]
```

2

```
{Length[A], Length[Flatten[A]]}
```

{3, 9}

Ejemplo completo del cálculo del rango de una matriz con parámetros

Ejemplo. Hallar el rango de la matriz A en función del parámetro $m \in \mathbb{R}$:

$$A = \begin{pmatrix} 1 & 3 & 1 & 1 \\ 1 & 1 & 3 & 1 \\ 1 & m & 2m+1 & m^2 \end{pmatrix}$$

- Introducción de la matriz 3×4

```
a = {{1, 3, 1, 1}, {1, 1, 3, 1}, {1, m, 2 m + 1, m^2}};
```

- Cálculo de los cuatro menores de orden tres: $r(A) \leq 3$

```
men = Minors[a, 3]
```

$\{\{6 - 6m, 2 - 2m^2, -2 + 2m^2, -2 - 6m + 8m^2\}\}$

- Convertir la salida anterior en una lista cuyos elementos sean los menores calculados

```
l = Flatten[men]
```

$\{6 - 6m, 2 - 2m^2, -2 + 2m^2, -2 - 6m + 8m^2\}$

- Cálculo de los valores del parámetro m que anulan todos los menores

```
sol = Solve[Table[l[[i]] == 0, {i, Length[l]}], m]
```

{{m → 1}}

- Sustitución de los valores del parámetro m (uno sólo en este caso) en la matriz original y cálculo de su rango

```
a1 = a /. sol[[1]]
```

{{1, 3, 1, 1}, {1, 1, 3, 1}, {1, 1, 3, 1}}

```
MatrixRank[a1]
```

2

- Solución: $\boxed{\begin{array}{l} \text{rg}(A) = 2 \quad \text{si } m = 1 \\ \text{rg}(A) = 3 \quad \text{si } m \neq 1 \end{array}}$