

1. Matrices en Matlab

Para introducir una matriz en Matlab se procede de la forma siguiente. Si por ejemplo tenemos la matriz

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

se introduce como:

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

```
A =  
    1    2    3    4  
    5    6    7    8
```

O bien,

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

Observemos que unas matrices especiales son los vectores, de esta forma, el vector fila $v = (1.0, 1.1, 1.2, 1.3, \dots, 1.9, 2.0)$, se escribe en Matlab como

```
>>v=[1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0]
```

2. Operaciones y comandos para Matrices

Hemos visto cómo se introducen las matrices en Matlab. Veamos un ejemplo para introducir algunos de los comandos básicos:

Ejemplo 1 Operaciones Elementales

Definimos dos matrices:

```
>>A=[2 1;3 2]  
A =  
    2    1  
    3    2
```

```
>>B=[3 4;-1 5]  
B =  
    3    4  
   -1    5
```

- Para **sumar** las dos matrices:

```
>>A+B  
ans =  
    5    5  
    2    7
```

- Para **multiplicar** una matriz por un **escalar**:

```
>>3*A
ans =
     6     3
     9     6
```

- **Producto** de matrices:

```
>>C=A*B
C =
     5    13
     7    22
```

Siempre que los tamaños de las matrices sean los adecuados. Para saber cuál es el **tamaño** de una **matriz** con la que estamos trabajando,

```
>>size(A)
ans =
     2     2
```

Que quiere decir, evidentemente, 2 filas y 2 columnas.

- Para calcular la matriz **transpuesta**:

```
>>A'
ans =
     2     3
     1     2
```

Ejercicio 1 Utilizando las matrices definidas en el ejemplo anterior, comprobar que $(AB)^t = B^t A^t$. (A^t es la transpuesta de A).

Ejemplo 2 Operaciones término a término: `.*` `./` `.^`

Matlab tiene tres operaciones, que las llamaremos operaciones con punto, que permiten

- i) multiplicar matrices término a término: `.*`
- ii) dividir matrices término a término: `./`
- ii) elevar los términos de una matriz a una cierta potencia: `.^`

Si v es el vector definido en la Sección 2, explorar qué hace la orden

```
>>v.^2
```

Por otra parte, si A y B son las matrices definidas anteriormente, explorar qué hacen las órdenes

```
>>A.*B
>>A./B
```

Estas operaciones con punto son esenciales en el cálculo numérico y se utilizan para representar funciones numéricamente.

Ejemplo 3 Matrices especiales con Matlab

- Para generar la matriz **identidad** cuadrada,

```
>>eye(3)
ans =
     1     0     0
     0     1     0
     0     0     1
```

¿Por qué habrán elegido el nombre `eye`?

- Una matriz 3×2 llena de unos,
`>>ones(3,2)`
- Y si queremos que esté llena de ceros,
`>>zeros(3,2)`
- Para generar una matriz con números **aleatorios** uniformemente distribuidos entre 0 y 1,
`>>rand(3,2)`

Si se usa el comando `randn` los números aleatorios son normalmente distribuidos, siguiendo la Normal Estandar $N(0,1)$.

Ejemplo 4 Rango, Inversa y Determinante

- Definimos la matriz,

```
>>X=[2 3 4; 1 -1 0]
X =
     2     3     4
     1    -1     0
```

Para calcular su **rango**,

```
>>rank(X)
ans =
     2
```

- Supongamos que tenemos definida la siguiente matriz,

```
H =
     8     1     6
     3     5     7
     4     9     2
```

Para calcular su **inversa**,

```
>>inv(H)
ans =
    0.1472   -0.1444    0.0639
   -0.0611    0.0222    0.1056
   -0.0194    0.1889   -0.1028
```

Y si queremos ver el resultado en forma racional,

```
>>format rational
>>inv(H)
ans =
    53/360   -13/90     23/360
   -11/180     1/45     19/180
    -7/360    17/90    -37/360
```

(Para ver todas las opciones del comando `format` hacer `help format`)

- Para calcular el determinante de la matriz anterior H,

```
>>det(H)
ans =
   -360
```

Ejercicio 2 Generar una matriz cualquiera, por ejemplo 25×25 , y calcular su inversa, su rango y su determinante. (¡No imprimirla!) ¿Qué ocurre con el determinante de la matriz y el de su inversa?

Ejemplo 5 Los comandos especiales `rref` y `rrefmovie`

- El comando `rref` produce la forma reducida escalonada por filas de una matriz usando la eliminación de Gauss-Jordan, es decir, haciendo ceros por debajo y por encima de la diagonal principal sin mover las columnas.

Por ejemplo, definimos la matriz,

```
>>A=[-1 2 -1;2 1 2;2 4 2]
```

A =

-1	2	-1
2	1	2
2	4	2

Ahora escribimos el comando aplicado a la matriz,

```
>>R=rref(A)
```

R =

1	0	1
0	1	0
0	0	0

- El comando `rrefmovie` produce exactamente el mismo resultado pero nos indica paso a paso cómo se va obteniendo la matriz resultado e incluso qué filas o columnas son despreciables (por ser linealmente dependientes de las otras), información muy útil si queremos calcular el rango de la matriz por ejemplo. Es decir, produce una especie de película (movie) de todo el proceso.

```
>>rrefmovie(A)
```

Original matrix

A =

-1	2	-1
2	1	2
2	4	2

Press any key to continue. . .

Ahora pulsamos una tecla para continuar,

swap rows 1 and 2

A =

2	1	2
-1	2	-1
2	4	2

Press any key to continue. . .

Nos indica que ha intercambiado la primera y segunda filas, pulsamos de nuevo una tecla,

pivot = A(1,1)

A =

1	1/2	1
-1	2	-1
2	4	2

Press any key to continue. . .

Ahora nos indica que va a pivotar sobre el elemento (1,1) de la matriz,

eliminate in column 1

A =

1	1/2	1
-1	2	-1
2	4	2

Press any key to continue. . .

Ahora nos está indicando que va a eliminar (hacer ceros) en la primera columna y así sucesivamente hasta obtener el mismo resultado que nos dió el comando **rref**.

Ejercicio 3 a) Calcular el rango de la matriz siguiente utilizando el comando **rref** o **rrefmovie**:

$$A = \begin{pmatrix} 16 & 2 & 3 & 13 \\ 5 & 11 & 10 & 8 \\ 9 & 7 & 6 & 12 \\ 4 & 14 & 15 & 1 \end{pmatrix}$$

b) Si una matriz H es cuadrada y no singular, es decir $\det(H) \neq 0$, ¿cuál será la matriz $R = \mathbf{rref}(H)$?

c) ¿Cómo podemos utilizar estos comandos para calcular la inversa de una matriz invertible? Aplicarlo a la matriz,

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

Para verificar el resultado se puede calcular la inversa directamente con **inv(B)**.

3. Matrices “dispersas”

Ejemplo 6 A veces usamos matrices con muchos ceros. MatLab tiene una forma de trabajar con ellas usando menos bytes con el comando **sparse**. Veámoslo con un ejemplo:

Introducimos una matriz:

```
>>A=[0 0 0 3;0 0 -1 2;3 0 0 1;0 0 0 -2];
```

- Para convertirla a matriz dispersa

```
>>s=sparse(A)
```

Si preguntamos “*whos*” vemos que *s* ocupa menos que *A*.

- Para recuperar la matriz inicial

```
>>full(s)
```

Para visualizar gráficamente la matriz:

```
>>spy(s)
```

o bien,

```
>>imagesc(s),colorbar
```

Se pueden **generar** directamente matrices “**sparse**”:

```
>>sparse(i,j,s,m,n)
```

donde: *i,j* son los subíndices de los elementos no nulos (*i,j* son vectores)

\mathbf{s} es un vector con los valores de los elementos no nulos

(\mathbf{m}, \mathbf{n}) es el tamaño de la matriz.

De modo que, en el ejemplo anterior, para generar \mathbf{s} deberíamos escribir:

```
>>i=[1 2 2 3 3 4];
```

```
>>j=[4 3 4 1 4 4];
```

```
>>s=[3 -1 2 3 1 -2];
```

```
>>m=4;n=4;
```

```
>>sparse(i,j,s,m,n)
```

Y obtenemos \mathbf{s} . Para obtener la matriz inicial `>>full(s)`

Ejercicio 4 Utilizando el comando `sparse`, generar la matriz 20×20

$$\begin{pmatrix} 0 & 1 & 0 & 0 & \dots & 0 & 0 \\ -1 & 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & -1 & 0 & 1 & \dots & 0 & 0 \\ \vdots & & & & \ddots & & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & 0 & 0 & \dots & -1 & 0 \end{pmatrix}$$

(Visualizarla para comprobar que está bien).

4. Sistemas de ecuaciones lineales

Un sistema de ecuaciones lineales,

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m \end{cases}$$

con m ecuaciones y n incógnitas se puede escribir en forma matricial,

$$\mathbf{Ax} = \mathbf{b}$$

donde,

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & & & \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}; \quad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \quad \text{y} \quad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

Vamos a ver mediante algunos ejemplos y ejercicios cómo se pueden resolver los sistemas de ecuaciones lineales utilizando algunos de los comandos de Matlab descritos anteriormente.

Ejemplo 7 Consideremos el sistema,

$$\begin{cases} 2x - y + z = 3 \\ x + y = 3 \\ y - 3z = -7 \end{cases}$$

entonces, siguiendo la notación anterior,

$$A = \begin{pmatrix} 2 & -1 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & -3 \end{pmatrix}, \quad \mathbf{x} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad \text{y} \quad \mathbf{b} = \begin{pmatrix} 3 \\ 3 \\ -7 \end{pmatrix}$$

Como se trata de un sistema con solución única, ya que el determinante de A es distinto de cero,

```
>>det(A)
ans =
    -8
```

Una forma de resolver el sistema es escribir la matriz orlada (o ampliada)

```
>>Ab=[A b]
```

y hacer `rref(Ab)` con lo que obtenemos

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

es decir, la solución es $x = 1$, $y = 2$, $z = 3$.

Otra forma de resolver el sistema consiste en despejar \mathbf{x} ,

$$\mathbf{x} = A^{-1}\mathbf{b},$$

sin más que escribir

```
>>x=inv(A)*b
x =
     1
     2
     3
```

Hay otra forma de hacerlo, utilizando lo que en Matlab se denomina como **división matricial a la izquierda**:

```
>>x=A\b
x =
     1
     2
     3
```

En este caso, el resultado es el mismo, pero es diferente la forma en la que trabaja el ordenador. En este segundo caso el método que utiliza es el de la **factorización LU**, que es una modificación de la eliminación gaussiana.

Ejercicio 5 Resolver el siguiente sistema utilizando los tres procedimientos anteriormente descritos y comprobar que sale la misma solución.

$$\begin{pmatrix} 1 & 1 & 0 & 3 \\ 2 & 1 & -1 & 1 \\ 3 & -1 & -1 & 2 \\ -1 & 2 & 3 & -1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 4 \\ 1 \\ -3 \\ 4 \end{pmatrix}$$

5. Sistemas homogéneos y su aplicación al ajuste de reacciones químicas

Un sistema de ecuaciones lineales

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m \end{cases}$$

con m ecuaciones y n incógnitas se llama **homogéneo**, si todas las constantes b_1, b_2, \dots, b_m son cero. Es decir, el sistema general homogéneo está dado por

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = 0 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = 0 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = 0 \end{cases}$$

En un sistema homogéneo, siempre existe la *solución trivial* (o solución cero):

$$x_1 = x_2 = \dots = x_n = 0$$

Por tanto, en un sistema homogéneo caben dos posibilidades:

- o bien solo existe la solución trivial;
- o bien existe un número infinito de soluciones, además de la trivial, llamadas *soluciones no triviales*.

Ejemplo 8 Un sistema homogéneo que tiene sólo la solución trivial:

$$\begin{cases} 3x_1 + 4x_2 + 6x_3 = 0 \\ 4x_1 + 5x_2 + 6x_3 = 0 \\ 3x_1 + x_2 - 2x_3 = 0 \end{cases}$$

Si escribimos la matriz ampliada

$$\begin{pmatrix} 2 & 4 & 6 & 0 \\ 4 & 5 & 6 & 0 \\ 3 & 1 & -2 & 0 \end{pmatrix},$$

y aplicamos el comando `rref`, obtendremos

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

lo que quiere decir que la única solución es la trivial,

$$x_1 = x_2 = x_3 = 0.$$

Ejemplo 9 Un sistema homogéneo con un número infinito de soluciones:

$$\begin{cases} x_1 + 2x_2 - x_3 = 0 \\ 3x_1 - 3x_2 + 2x_3 = 0 \\ -x_1 - 11x_2 + 6x_3 = 0 \end{cases}$$

Haciendo lo mismo que en el ejemplo anterior (y poniendo el resultado en formato racional, `format rational`;) obtenemos

$$\begin{pmatrix} 1 & 0 & \frac{1}{9} & 0 \\ 0 & 1 & \frac{-5}{9} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

lo que quiere decir que, son soluciones todas las ternas de números reales de la forma

$$\left(\frac{-1}{9}x_3, \frac{5}{9}x_3, x_3\right),$$

para cualquier valor de x_3 . En particular, para $x_3 = 0$, obtenemos la solución trivial; para $x_3 = 1$, obtenemos la solución

$$\left(\frac{-1}{9}, \frac{5}{9}, 1\right),$$

para $x_3 = 9\pi$, la solución

$$(-\pi, 5\pi, 9\pi).$$

Ejemplo 10 Un sistema homogéneo con más incógnitas que ecuaciones tiene un número infinito de soluciones:
Resolvemos el sistema:

$$\begin{cases} x_1 + x_2 - x_3 = 0 \\ 4x_1 - 2x_2 + 7x_3 = 0 \end{cases}$$

como en los ejemplos anteriores, escribiendo la matriz ampliada y haciendo **rref**, para obtener

$$\begin{pmatrix} 1 & 0 & \frac{5}{6} & 0 \\ 0 & 1 & \frac{-11}{6} & 0 \end{pmatrix},$$

lo cual quiere decir que son soluciones todas las ternas de números reales de la forma

$$\left(\frac{-5}{6}x_3, \frac{11}{6}x_3, x_3\right), \quad \forall x_3 \in \mathbb{R}$$

Observación: Si en estos tres ejemplos hubiéramos escrito la matriz sin ampliar y hubiéramos hecho **rref**, habríamos visto las soluciones más rápidamente:

En el primer ejemplo, puesto que el determinante de la matriz del sistema no era nulo, el resultado era la matriz identidad (solución única.) En el segundo caso, aparece una fila de ceros. Y en el tercer caso, ¿qué sucede?: *lo que sucede siempre que hay un sistema homogéneo con más incógnitas que ecuaciones, que tiene un número infinito de soluciones.*

Ejercicio 6 Resolver el sistema homogéneo

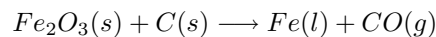
$$\begin{cases} x_1 - 2x_2 + x_3 + x_4 = 0 \\ 3x_1 + 2x_3 - 2x_4 = 0 \\ 4x_2 - x_3 - x_4 = 0 \\ 5x_1 + 3x_3 - x_4 = 0 \end{cases}$$

Ejercicio 7 Resolver el sistema

$$\begin{cases} 25x_1 - 16x_2 + 13x_3 + 33x_4 - 57x_5 = 0 \\ -16x_1 + 3x_2 + x_3 + 12x_5 = 0 \\ -8x_2 + 16x_4 - 26x_5 = 0 \end{cases}$$

Este estudio de los sistemas homogéneos se puede aplicar al *ajuste* de reacciones químicas:

Ejemplo 11 Para ajustar la reacción



que se produce cuando calentamos mineral de óxido de hierro con un exceso de carbono para obtener hierro puro, procedemos de la siguiente manera:

$$Fe : 2x_1 = x_3$$

$$O : 3x_1 = x_4$$

$$C : x_2 = x_4$$

Es decir, se trata de resolver el siguiente sistema homogéneo de 3 ecuaciones con 4 incógnitas (que, como acabamos de ver, siempre tiene solución no trivial):

$$\begin{cases} 2x_1 & - x_3 & = 0 \\ 3x_1 & & - x_4 = 0 \\ & x_2 & - x_4 = 0 \end{cases} \quad (1)$$

O lo que es lo mismo, si consideramos x_4 como un parámetro ($x_4 = \lambda$) tratamos de resolver el siguiente sistema de 3 ecuaciones con 3 incógnitas

$$\begin{cases} 2x_1 & - x_3 = 0 \\ 3x_1 & = \lambda \\ & x_2 = \lambda \end{cases}$$

obteniendo la solución

$$x_4 = \lambda, \quad x_2 = \lambda, \quad x_1 = \frac{\lambda}{3}, \quad x_3 = 2\frac{\lambda}{3}$$

De modo, que podemos decir que el sistema 1 tiene las infinitas soluciones:

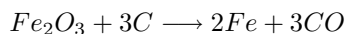
$$x_1 = \frac{\lambda}{3}, \quad x_2 = \lambda, \quad x_3 = 2\frac{\lambda}{3}, \quad x_4 = \lambda$$

para cualquier valor de $\lambda \in \mathbb{R}$.

En nuestro caso, solo nos interesa la solución de números enteros positivos x_1, x_2, x_3, x_4 que no tengan divisor común diferente de 1, es decir, para $\lambda = 3$:

$$x_1 = 1, \quad x_2 = 3, \quad x_3 = 2, \quad x_4 = 3,$$

de modo que la reacción ajustada es



Ejercicio 8 Ajustar las siguientes reacciones químicas:

1. $CO_2 + H_2O \longrightarrow C_6H_{12}O_6 + O_2$
2. $Pb(N_3)_2 + Cr(MnO_4)_2 \longrightarrow Cr_2O_3 + MnO_2 + Pb_3O_4 + NO$
3. $AgNO_3 + K_2CrO_4 \longrightarrow Ag_2CrO_4 + KNO_3$
4. $Mg + HCl \longrightarrow MgCl_2 + H_2$

Ejercicio 9 Ajustar la siguiente serie de reacciones químicas que se utilizan para producir clorato de sodio:

1. $KMnO_4 + HCl \longrightarrow KCl + MnCl_2 + H_2O + Cl_2$
2. $Cl_2 + Ca(OH)_2 \longrightarrow Ca(ClO_3)_2 + CaCl_2 + H_2O$
3. $Ca(ClO_3)_2 + Na_2SO_4 \longrightarrow CaSO_4 + NaClO_3$