

Capítulo 2

Matrices. Resolución de sistemas lineales

Como ya se ha comentado, el tipo básico de dato con el que MATLAB trabaja es la matriz, incluso los escalares son considerados como matrices 1×1 , por lo que es esencial familiarizarse con esta sección.

2.1. Vectores

Los vectores se introducen escribiendo cada una de sus coordenadas entre corchetes, separadas por un espacio en blanco:

```
>> v=[1 3 pi 1/3]
```

o bien separadas por comas:

```
>> v=[1,3,pi,1/3]
```

Si separamos las componentes mediante punto y coma, obtendremos un vector columna:

```
>> w=[1;3;pi;1/3]
```

No obstante, existen formas de introducir vectores, cuando sus coordenadas guardan alguna relación entre sí:

Función	Salida
<code>[a:h:b]</code>	Vector $(a, a+h, a+2h, \dots, a+nh)$, donde n es el mayor entero tal que $a+nh \in [a, b]$ si $h > 0$ y $a+nh \in [b, a]$ si $h < 0$. En este caso, los corchetes pueden sustituirse por paréntesis o incluso eliminarse
<code>linspace(a,b,n)</code>	Vector de n componentes cuyas coordenadas son los puntos de una partición uniforme del intervalo $[a, b]$
<code>linspace(a,b)</code>	Vector de 100 componentes cuyas coordenadas son los puntos de una partición uniforme del intervalo $[a, b]$

```
>> v=[1:0.3:2]
>> v=(1:-0.4:-0.8)
```

Si se omite el incremento **h**, MATLAB toma por defecto **h=1**:

```
>> v=1:4

>> v=linspace(1,2,5)
```

Ejemplo 2.1.1 Siguiendo el ejemplo 1.3.2, supongamos ahora que queremos calcular los espacios recorridos por el móvil a velocidad $v_0 = 5$ m/s, para 5 instantes correspondientes a los 2 primeros segundos del movimiento:

```
>> t=linspace(0,2,5)
>> v0=5
>> s=v0*t
```

Obsérvese que, como cabía esperar, el resultado del producto de un escalar por un vector es el vector de las posiciones en los instantes correspondientes.

Si nos interesa conocer las posiciones en instantes de tiempo separados por 0,3 segundos

```
>> t=[0:0.3:2]
>> s=v0*t
```

Ejercicio 2.1 Créese un archivo con las ordenes siguientes:

1. Vector fila **v1** que tenga 6 números igualmente espaciados comprendidos entre el 7 y el 10.
2. Almacénese en la variable **v2** un vector que contenga a los 6 primeros números pares.
3. Almacénese en la variable **v3** un vector con componentes números naturales múltiplos de 3 y comprendidos entre 13 y 43.

Guárdese y ejecútese el programa anterior.

2.2. Matrices

Para construir una matriz, se deben tener en cuenta las siguientes reglas:

- Los elementos se introducen entre corchetes.
- Los elementos de una misma fila se introducen como un vector fila, separados por comas o por espacios en blanco.
- Para separar filas se utiliza el punto y coma o el cambio de línea:

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

Una vez definida una matriz o un vector, se puede acceder a sus elementos o submatrices con las órdenes:

Función	Salida
v(i)	Coordenada i del vector v
A(i,j)	Elemento de la matriz A que ocupa la posición i,j
A(:,j)	Columna j de la matriz A
A(:,end)	Columna última de la matriz A
A(i,:)	Fila i de la matriz A
A(end,:)	Fila última de la matriz A
A(v,w)	Submatriz de A que contiene las filas indicadas en las coordenadas de v y las columnas indicadas en w
A(i,:)=[]	Elimina la fila i de la matriz A
A(:,j)=[]	Elimina la columna j de la matriz A
	Nota: [] representa la matriz vacía(<i>empty matrix</i>)

Así, si recuperamos con el ratón en la subventana correspondiente a *Command History* el vector v y la matriz A (pinchar dos veces sobre v y A) y tecleamos:

```
>> v(2)
>> A(1,2)
```

nos devuelven respectivamente la segunda componente de v y el elemento de la primera fila, segunda columna de A.

```
>> v([1,3])
>> A([2,1],1)
```

nos devuelven respectivamente las componentes primera y tercera de v y los elementos de las filas segunda y primera, de la primera columna de A.

Recuperando previamente t, con el fin de que se vea mejor como actúa la orden

```
>> t(1:2:7)
>> A(:,2)
```

nos devuelven respectivamente las componentes impares de t y los elementos de la segunda columna de A.

Haciendo uso de estas órdenes pueden introducirse matrices y vectores elemento a elemento. Al asignarle un valor a una posición, se construye la matriz o vector de menor tamaño que contiene los elementos introducidos y hace ceros los no asignados.

```
>> B(5)=3
>> B(2,4)=5
```

O bien se puede utilizar para modificar posiciones de una matriz predefinida:

```
>> A=eye(3) % Matriz identidad 3 x 3
>> A(1,2)=5
```

También puede eliminarse filas y columnas de matrices dadas.

```
>> A=eye(5) % Matriz identidad 5 x 5
>> A(3,4)=10, A(2,5)=1
>> B=A
>> B(:,2)=[]
```

La matriz B coincide con la matriz obtenida de eliminar la columna 2 de A.

Pueden definirse ciertas matrices con las siguientes órdenes:

Función	Salida
<code>ones(n)</code>	Matriz cuadrada $n \times n$ de unos.
<code>ones(m,n)</code>	Matriz $m \times n$ de unos.
<code>zeros(n)</code>	Matriz cuadrada $n \times n$ de ceros.
<code>zeros(m,n)</code>	Matriz $m \times n$ de ceros.
<code>eye(n)</code>	Matriz identidad $n \times n$.
<code>eye(m,n)</code>	Matriz $m \times n$ con unos en la diagonal principal y el resto ceros.
<code>rand(n)</code>	Matriz cuadrada $n \times n$ de números aleatorios entre 0 y 1 y distribución uniforme.
<code>rand(m,n)</code>	Matriz $m \times n$ de números aleatorios entre 0 y 1 y distribución uniforme.
<code>randn(n)</code>	Matriz cuadrada $n \times n$ de números aleatorios, distribución N(0,1).
<code>randn(m,n)</code>	Matriz $m \times n$ de números aleatorios, distribución N(0,1).

Ejemplo 2.2.1 Constrúyase una matriz A, de orden 5×5 , de unos. Cambia todos los unos de la fila tres, por el número 20.

```
>> A=ones(5)
>> A(3,:)=20*A(3,:)
```

Ejemplo 2.2.2 Constrúyase una matriz A, de orden 5×7 , de números aleatorios entre 0 y 1 y distribución uniforme. Cámbiese la última fila por un vector cuyas coordenadas sean los puntos de una distribución uniforme del intervalo $[5, 7]$.

```
>> A=rand(5,7)
>> v=linspace(5,7,7)
>> A(end,:)=v
```

2.2.1. Definición de matrices por bloques

Dadas dos matrices A y B con el mismo número de filas, se puede definir una matriz C formada por todas las columnas de A y de B:

```
>> A=rand(3), B=eye(3,2), C=[A B]
```

Análogamente, se puede definir una matriz a partir de otras dos con el mismo número de columnas:

```
>> A=randn(2,3), B=ones(3), C=[A;B]
```

Estas dos posibilidades pueden combinarse para formar matrices definidas por bloques:

```
>> A=[eye(3) ones(3,3);1:6;zeros(2) ones(2,1) eye(2,3)]
```

Nota: En lo sucesivo cada ejercicio, se realizará, en un archivo nuevo y se guardará con un nombre que haga referencia al número del ejercicio que estamos realizando.

Ejercicio 2.2 Créese un archivo nuevo y constrúyase una matriz de orden 3, con todos los elementos iguales a 1, salvo los de la diagonal, que deben tomar el valor 20.

Ejercicio 2.3 Ábrase un archivo nuevo y constrúyase una matriz A de números aleatorios entre 0 y 1 y distribución uniforme de orden 4. Modifíquese la primera fila de la matriz anterior de forma que todos los elementos de la primera fila valgan 4 y $a_{44} = 32$. Guárdese y ejecútase el programa.

2.3. Operaciones con vectores y matrices

Si A y B son matrices con las dimensiones adecuadas y λ es un escalar, las operaciones habituales se efectúan con las siguientes órdenes:

Operación	Resultado
$A+B$	Suma A y B
$A-B$	Resta B de A
$A*B$	Multiplica A por B
A/B	Calcula AB^{-1}
$A \setminus B$	Calcula $A^{-1}B$
$\lambda*A$	Multiplica todos los elementos de A por λ
A^n	Eleva la matriz A al entero n
$A.'$	Calcula la traspuesta de A
A'	Calcula la traspuesta de la conjugada de A

Por ejemplo

```
>> A=randn(2,3), B=ones(3), C=A*B, C=A+C, C=B^2, C=A'
```

Además de las operaciones mencionadas, en MATLAB se definen otras operaciones a las que llamaremos **operaciones elemento a elemento**:

Operación	Resultado
$\lambda + A$	Suma a cada elemento de A el escalar λ
$A .* B$	Calcula una matriz que en la posición (i, j) contiene el producto $a_{ij}b_{ij}$ de los elementos que en A y B ocupan dicha posición
$A ./ B$	Calcula una matriz que en la posición (i, j) contiene el cociente a_{ij}/b_{ij} de los elementos que en A y B ocupan dicha posición
$a ./ B$	Calcula una matriz que en la posición (i, j) contiene el cociente a/b_{ij} , siendo a un número real fijo.
$A.^n$	Eleva cada elemento de la matriz A al entero n
$A.^B$	Calcula una matriz que en la posición (i, j) contiene $a_{ij}^{b_{ij}}$
$a.^B$	Calcula una matriz que en la posición (i, j) contiene $a^{b_{ij}}$, siendo a un número real fijo.

Por ejemplo

```
>> A=randn(2,3), B=2+ones(3)
>> C=A*B
>> C=A.*B % Da error
>> C=A.*A, C=B.^2, C=B^2, C=A.^2, C=2.^A, C=1./A
```

Ejercicio 2.4 Constrúyase un fichero que pida por pantalla un vector y devuelva otro vector cuyas coordenadas sean los cubos de las respectivas coordenadas del vector dado. Pruébese sobre el vector $v = (1, 2, 3, 4, 5)$.

2.4. Funciones predefinidas en MATLAB

2.4.1. Funciones para manipular vectores y matrices

Todos los procesos informáticos consisten básicamente en la realización de una cierta tarea que, a partir de unos *datos de entrada*, nos permite calcular unos datos de *salida*. Como veremos en la sección siguiente, aunque MATLAB permite programar funciones para realizar tareas específicas, incorpora también una gran variedad de funciones para las tareas que podríamos denominar *básicas*.

En particular, las siguientes funciones permiten obtener información sobre las matrices o vectores que tienen como argumentos de entrada:

Función	Salida
<code>size(A)</code>	Vector con las dimensiones de la matriz A
<code>size(A,1)</code>	Número de filas de la matriz A
<code>size(A,2)</code>	Número de columnas de la matriz A
<code>length(v)</code>	Número de coordenadas del vector v
<code>length(A)</code>	Mayor elemento del vector <code>size(A)</code>
<code>diag(v)</code>	Matriz cuadrada en cuya diagonal está el vector v
<code>diag(A)</code>	Vector cuyos elementos son los de la diagonal principal de A
<code>rank(A)</code>	Rango de la matriz A
<code>det(A)</code>	Determinante de la matriz A
<code>inv(A)</code>	Inversa de la matriz A cuando A es inversible
<code>trace(A)</code>	Traza de la matriz A
<code>max(A)</code>	Vector con los valores máximos de cada columna
<code>[m n]=max(A)</code>	Guarda en m el vector <code>max(A)</code> y en n un vector cuyos elementos son las filas de A que contienen los elementos de m
<code>max(v)</code>	Máximo elemento del vector v
<code>[m n]=max(v)</code>	Guarda en m el valor <code>max(v)</code> y en n la posición de m en v
<code>min(A)</code>	Vector con los valores mínimos de cada columna
<code>[m n]=min(A)</code>	Guarda en m el vector <code>min(A)</code> y en n un vector cuyos elementos son las filas de A que contienen los elementos de m
<code>min(v)</code>	Mínimo elemento del vector v
<code>[m n]=min(v)</code>	Guarda en m el valor <code>min(v)</code> y en n la posición de m en v
<code>dot(u,v)</code>	Producto escalar de los vectores u y v
<code>cross(u,v)</code>	Producto vectorial de los vectores (de tres coordenadas) u y v
<code>sum(v)</code>	Suma de las componentes del vector v
<code>prod(v)</code>	Producto de las componentes del vector v

Ejercicio 2.5 Dadas las matrices $A = \begin{pmatrix} 1 & 3 & 4 & 3 \\ 2 & 1 & 10 & 6 \\ 2 & 1 & 5 & 7 \\ 1 & 2 & 10 & 3 \end{pmatrix}$ y $B = \begin{pmatrix} 11 & 0 & 6 & 3 \\ 3 & 1 & 2 & 1 \\ 21 & 0 & 5 & 0 \\ 0 & 3 & 1 & 2 \end{pmatrix}$

1. Construya una matriz C , tal que la primera fila sea 2 veces la fila primera de A , la segunda sea la suma de la segunda más dos veces la primera de A y la tercera y la cuarta coincidan con las correspondientes de la matriz A .
2. Construya una matriz cuadrada D de orden 4, tal que todos sus elementos valgan -1, salvo los de la diagonal que deben coincidir con los correspondientes de la diagonal de la matriz B .
3. Calcule $(A + B)C^{-1}$ y $C^{-1}(A - 2B)$.
4. Construya una matriz cuadrada E de orden 4, tal que sus elementos sean números racionales con numerador 2 y denominador el elemento correspondiente de A ($e_{ij} = 2/a_{ij}$). Use el formato racional.

2.4.2. Funciones matemáticas básicas

Obviamente, las funciones matemáticas habituales también están predefinidas en MATLAB, con la única particularidad de que actúan sobre vectores o matrices elemento a elemento.

MATLAB	Función	MATLAB	Función
<code>exp(x)</code>	e^x	<code>abs(x)</code>	$ x $
<code>log(x)</code>	$\ln(x)$	<code>fix(x)</code>	Redondeo hacia cero
<code>log10(x)</code>	$\log_{10}(x)$	<code>floor(x)</code>	Redondeo hacia $-\infty$
<code>log2(x)</code>	$\log_2(x)$	<code>ceil(x)</code>	Redondeo hacia $+\infty$
<code>sqrt(x)</code>	$\sqrt{x}$	<code>round(x)</code>	Redondeo hacia el entero más próximo
		<code>rem(m,n)</code>	resto de dividir m entre n

y las funciones trigonométricas:

MATLAB	Fun.	MATLAB	Fun.	MATLAB	Func.	MATLAB	Fun.
<code>sin(x)</code>	$\sin(x)$	<code>asin(x)</code>	$\arcsin(x)$	<code>sinh(x)</code>	$\sinh(x)$	<code>asinh(x)</code>	$\operatorname{arsinh}(x)$
<code>cos(x)</code>	$\cos(x)$	<code>acos(x)</code>	$\arccos(x)$	<code>cosh(x)</code>	$\cosh(x)$	<code>acosh(x)</code>	$\operatorname{arcosh}(x)$
<code>tan(x)</code>	$\tan(x)$	<code>atan(x)</code>	$\arctan(x)$	<code>tanh(x)</code>	$\tanh(x)$	<code>atanh(x)</code>	$\operatorname{artanh}(x)$
<code>cot(x)</code>	$\cot(x)$	<code>acot(x)</code>	$\operatorname{arccot}(x)$	<code>coth(x)</code>	$\coth(x)$	<code>acoth(x)</code>	$\operatorname{arcoth}(x)$
<code>sec(x)</code>	$\sec(x)$	<code>asec(x)</code>	$\operatorname{arcsec}(x)$	<code>sech(x)</code>	$\operatorname{sech}(x)$	<code>asech(x)</code>	$\operatorname{arsech}(x)$
<code>csc(x)</code>	$\csc(x)$	<code>acsc(x)</code>	$\operatorname{arccsc}(x)$	<code>csch(x)</code>	$\operatorname{csch}(x)$	<code>acsch(x)</code>	$\operatorname{acsch}(x)$

Ejercicio 2.6

1. Constrúyase con las funciones de MATLAB un vector v cuyas componentes sean los 10 primeros números naturales. Utilizando la orden `sum` sobre v , calcule: $\sum_{i=1}^{10} i$.

2. Utilizando la orden `sum` sobre vectores adecuados, calcule: $\sum_{i=1}^{10} i^2$, $\sum_{i=1}^{10} \frac{1}{i}$, $\sum_{i=1}^{10} \frac{1}{2^i}$. Para hacerlo, es conveniente encontrar primero cada vector utilizando las operaciones elemento a elemento sobre el vector v .

Ejercicio 2.7 Constrúyase un vector de 10 entradas que tenga un 1 en las posiciones pares y un -1 en las impares.

Ejercicio 2.8 Sea $r = 1/2$, hállese $\sum_{i=1}^n r^i$ para $n = 100$.

2.5. Sistemas lineales

Funciones y Notación

Las órdenes nuevas que necesitaremos manejar en este capítulo se recogen en el siguiente cuadro

Función	Salida
<code>rref(X)</code>	Calcula la matriz escalonada reducida de una matriz X
<code>null(A)</code>	Devuelve una base del subespacio de las soluciones de un sistema homogéneo
<code>A\b</code>	Devuelve una solución del sistema $AX = b$. Esta solución debe ser interpretada, en ocasiones, en el sentido de mínimos cuadrados

En este apartado abordaremos, a través de ejemplos, la resolución de sistemas lineales con ayuda del MatLab. Iremos analizando los distintos tipos de sistemas que nos podemos encontrar y concluiremos con la resolución de ejercicios en los que discutiremos y resolveremos completamente los sistemas propuestos.

2.5.1. Resolución de sistemas lineales con MATLAB

Ejemplo 2.5.1 (Sistema compatible determinado con matriz cuadrada) *Dado el sistema*

$$\begin{cases} x + 2y + 3z = 6 \\ x + 3y + 8z = 19 \\ 5x + 6y + 4z = 5 \end{cases}$$

discútase y resuélvase.

Solución

```
>> A1=[1 2 3; 1 3 8; 5 6 4]; b1=[6;19;5]; r=rank(A1), ra=rank([A1 b1])
r =
    3
ra =
    3
```

Se observa que el sistema es compatible determinado. Un método de resolución es:

```
>> x=A1\b1
x =
    1.0000
   -2.0000
    3.0000
```

En este caso, la instrucción `A1\b1` devuelve el valor de $(A1)^{-1} * (b1)$ que es la única solución del sistema.

Otra forma de resolver el sistema es mediante el comando **rref**:

```
>> E=rref([A1 b1])
E =
     1     0     0     1
     0     1     0    -2
     0     0     1     3
```

La orden **rref** devuelve la matriz escalonada reducida, y esto implica que el sistema inicial tiene las mismas soluciones que el sistema:

$$\begin{cases} x & = & 1 \\ y & = & -2 \\ z & = & 3 \end{cases}$$

Puesto que la matriz del nuevo sistema es la identidad la solución es la última columna de E, es decir, `E(:,4)` o bien `E(:,end)`.

Ejemplo 2.5.2 (Sistema compatible determinado con matriz no cuadrada) *Resuelva el sistema*

$$\begin{cases} x + 2y + 3z = 6 \\ x + 3y + 8z = 19 \\ 2x + 3y + z = -1 \\ 5x + 6y + 4z = 5 \end{cases}$$

Solución

```
>> A2=[1 2 3; 1 3 8; 2 3 1; 5 6 4]; b2=[6;19;-1;5];r= rank(A2), ra= rank([A2 b2])
r =
     3
ra =
     3
```

El sistema es compatible determinado, aunque la matriz no es cuadrada. Se puede resolver de las dos formas vistas en el ejemplo anterior, pero es conveniente observar que ahora la instrucción

```
>> x=A2\b2
x =
    1.0000
   -2.0000
    3.0000
```

devuelve la solución del sistema, que no coincide con $(A2)^{-1} * (b2)$, ya que la matriz no es cuadrada. También es posible utilizar la orden:

```
>> E=rref([A2 b2])
E =
     1     0     0     1
     0     1     0    -2
     0     0     1     3
     0     0     0     0
```

siendo ahora la solución E(1:3,4) (los tres primeros elementos de la última columna).

Ejemplo 2.5.3 (Sistema compatible indeterminado homogéneo) *Resuelva el sistema*

$$\begin{cases} x + 2y - z = 0 \\ 2x - y + z = 0 \\ 3x + y = 0 \end{cases}$$

Solución

```
>> A3=[1 2 -1; 2 -1 1 ; 3 1 0];r=rank(A3)
r =
     2
```

Sistema compatible indeterminado. Para obtener la solución general calculamos una base del espacio de soluciones mediante la orden

```
>> X=null(A3)
X =
    0.1690
   -0.5071
   -0.8452
```

Las soluciones del sistema son los vectores proporcionales al vector $\mathbf{X}$: observe que si un vector X verifica $AX = 0$ (es solución del sistema) entonces también se cumple que $A(cX) = 0$ para todo $c \in \mathbb{R}$ (es decir, cX es solución para todo $c \in \mathbb{R}$).

Nota: La instrucción **null** devuelve vectores de norma uno cuando opera sobre datos numéricos.

Ejemplo 2.5.4 (Sistema compatible indeterminado no homogéneo.) *Resuelva el sistema*

$$\begin{cases} x + 2y - z - t = 1 \\ 2x - y + z + t = 3 \\ 3x + y = 4 \end{cases}$$

Solución

```
>> A4=[A3 [-1;1;0]]; % Se utiliza la matriz del ejemplo anterior
b4=[1;3;4];r=rank(A4),ra=rank([A4,b4])
r =
    2
ra =
    2
```

El sistema es compatible indeterminado. Se calcula la solución general del sistema homogéneo:

```
>> Base=null(A4)
Base =
   -0.2108         0
    0.6325    0.0000
    0.5270   -0.7071
    0.5270    0.7071
```

Una solución particular del sistema completo se puede obtener con:

```
>> Xp=A4\b4
Warning: Rank deficient, rank = 2, tol = 3.3233e-015.
Xp =

    1.4000
   -0.2000
         0
         0
```

que devuelve sólo una de las infinitas soluciones y un aviso de que la matriz no tiene rango máximo.

Recuérdese que la solución general de un sistema compatible indeterminado se puede expresar como la suma de una solución particular del sistema (en nuestro caso $\mathbf{X}_p$) más una combinación lineal de los vectores que resuelven el sistema homogéneo (en nuestro caso las columnas de **Base**).

Los vectores $X=X_p+2*Base(:,1)-6*Base(:,2)$ ó $Y=X_p-0.3*Base(:,1)$ son por tanto soluciones del sistema. Si ponemos los dos escalares de la combinación lineal en un vector columna (por ejemplo $c=[5; -3]$), la mejor manera de escribir $Z=X_p+5*Base(:,1)-3*Base(:,2)$ es $Z=x+Base*c$.

Para obtener las infinitas soluciones de forma implícita se puede utilizar el comando **rref**:

```
>> G=rref([A4,b4])
G =
```

```
1.0000    0    0.2000    0.2000    1.4000
      0    1.0000   -0.6000   -0.6000   -0.2000
      0      0      0      0      0
```

es decir, que el sistema inicial es equivalente al sistema:

$$\begin{cases} x + 0,2z + 0,2t = 1,4 \\ y - 0,6z - 0,6t = -0,2 \end{cases}$$

Nota: Para calcular una solución particular, en vez de utilizar la orden $A4 \backslash b4$, es preferible trabajar con la matriz G . En ella vemos que los pivotes corresponden a las variables x e y , quedando libres z y t . Por tanto, cuando éstas últimas valgan cero, los valores de x e y , serán los que aparecen en la correspondiente fila de la última columna de G . Podemos, pues, construir una solución particular como $X_p=[G(1:2,end);0;0]$.

El ejemplo que vemos a continuación nos muestras alguno de los problemas que podemos encontrarnos si utilizamos la órdenes del tipo $A \backslash b$, sobre determinadas matrices.

Ejemplo 2.5.5 (Sistema compatible indeterminado no homogéneo.) *Resuelva el sistema*

$$\begin{cases} x + y = 1 \\ 2x + 2y = 2 \end{cases}$$

Solución

```
>> A=[1 1;2 2]; b=[1;2]; r= rank(A),ra=rank([A b])
r =
    1
ra =
    1
```

El sistema es compatible indeterminado. Se calcula la solución general del sistema homogéneo:

```
>> Base=null(A)
Base =
   -0.7071
    0.7071
```

Sin embargo , al calcular una solución particular del sistema completo con la orden $A \backslash b$ nos encontramos:

```
>> Xp=A\b
Warning: Matrix is singular to working precision.
```

```
Xp =
    NaN
    NaN
```

Por ello, es necesario trabajar con la reducida escalonada

```
>> G=rref([A,b])
G =
     1     1     1
     0     0     0
```

En ella vemos que el único pivote corresponde a la variable x , mientras que la variable y queda libre. Dándole a esta última el valor cero, el valor de x será el que vemos en la correspondiente fila de la última columna de G . Una solución particular será, por tanto, $Xp = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$.

Ejemplo 2.5.6 (Sistema incompatible.) *Resuelva el sistema*

$$\begin{cases} 2x + y = 1 \\ x - y = 0 \\ -x - y = 3 \end{cases}$$

Solución

```
>> A=[2 1;-1 -1;-1 -1]; b=[1;0;3]; r= rank(A), ra=rank([A b])
r =
     2
ra =
     3
```

y por tanto el sistema es incompatible. Sin embargo la instrucción

```
>> s=A\b
s =
    0.0714
   -0.7143
```

proporciona la *solución de mínimos cuadrados*, que debe ser interpretada en el sentido siguiente: El vector s es el que hace mínima la norma $\|AX - b\|$, es decir, $\|As - b\| \leq \|AX - b\|$ para todo $X \in \mathbb{R}^2$.

Ejercicio 2.9 *Se considera el sistema*

$$\begin{cases} 5x + 5y - 2z - 2t = 6 \\ 5x - 10y + 7z + 7t = 9 \\ 5x + z + t = 7 \\ 5y - 3z - 3t = -1 \end{cases}$$

1. Pruebe que el sistema es compatible indeterminado.
2. Calcule una solución particular del sistema.
3. Calcule un conjunto de vectores que permita determinar las soluciones del sistema homogéneo asociado.
4. Teniendo en cuenta los dos apartados anteriores, dedúzcase como se obtendrían las infinitas soluciones del sistema dado.
5. Pruebe que el sistema definido por las dos últimas ecuaciones es equivalente al sistema inicial.

Ejercicio 2.10 Se considera el sistema

$$\begin{cases} x + 2y - z + t + u = 0 \\ 3x - y + t - u = 6 \\ 6x + y + t + u = 1 \\ x - 2y + 2z - 2t = -5 \end{cases}$$

1. Pruebe que el sistema es compatible indeterminado.
2. Calcule una solución particular del sistema.
3. Calcule un conjunto de vectores que permita determinar las soluciones del sistema homogéneo asociado.
4. Teniendo en cuenta los dos apartados anteriores, dedúzcase como se obtendrían las infinitas soluciones del sistema dado.

Ejercicio 2.11 Dado el sistema:

$$\begin{cases} -3x + y + z - 7t - 10w = 10 \\ -14x + 12y + 9z + 23t - 7w = 3 \\ 3x + 7z + t - 8w = 16 \\ x + 12y - 2z - t - w = 7 \end{cases}$$

1. Pruebe que el sistema es compatible indeterminado.
2. Calcule una solución particular del sistema.
3. Calcule un conjunto de vectores que permita determinar las soluciones del sistema homogéneo asociado.
4. Teniendo en cuenta los dos apartados anteriores, dedúzcase como se obtendrían las infinitas soluciones del sistema dado.

Ejercicio 2.12 Utilizando la orden **rref** constrúyase una solución particular del sistema

$$\begin{cases} x + y + z + t = 1 \\ 2z + t = 2 \end{cases}$$