516+0"'(7(8+- 3 +%"9'

!"#$%&$'()('*+,-.(/''"&'01+2(2&3 041&$'2

:-;"&6-(8'"$9'(<'*='*'2
:>%?(@'"0'11-(A&;'16-

!"#$%&$' "(&)*+"*, $&"' -&./$*0#1./$+$*2
3."(/.$4*+"*1$*3) '#5&$/.6(

74&"*&" '$*4"*#581./$*8$9)*:./"(/.$;
3%"$&.<"*3) ' ')(4*=>?@3?A0*BCD

Grado en Ingeniería Civil

G1954: Álgebra y Geometría

Variables simbólicas

Rodrigo García Manzanas (rodrigo.manzanas@unican.es)

Ruth Carballo Fidalgo (ruth.carballo@unican.es)

Uso de variables simbólicas en MATLAB

MATLAB permite trabajar con variables simbólicas, es decir, variables genéricas que, en principio, no tienen porqué tomar un valor concreto (piensa en las incógnitas de cualquier ecuación)

```
syms x  % defino la variable 'x' como simbólica
eq = x^2 + 5  % ecuación que depende de 'x'
```

eq = $x^2 + 5$

Por depender de la variable simbólica *x, eq* será una expresión simbólica. Podemos comprobar de qué tipo es una variable dada con el comando *class*:

```
class(x)  % 'x' es una variable simbólica
```

```
ans =
'sym'
```

```
class(eq)  % 'eq' es una expresión simbólica
```

```
ans =
'sym'
```

Podemos obligar a que una variable simbólica tome un determinado valor con el comando *subs*:

```
subs(eq, x, 3)  % forzamos a que 'x' tome el valor 3 en la expresión eq
```

ans = 14

Se pueden definir varias variables simbólicas a la vez:

```
syms x y
subs(2*x + y, [x, y], [1 6])  % forzamos a que 'x' ('y') tome el valor 1 (6)
```

ans = 8

```
% en la expresión '2*x+y'
```

Se puede especificar el tipo exacto de variable simbólica con la que queremos trabajar:

```
syms x integer  % forzamos que 'x' sólo pueda tomar valores enteros
syms x real  % forzamos que 'x' sólo pueda tomar valores reales
```

Por último, MATLAB también permite resolver sistemas de ecuaciones. De momento consideraremos sólo ecuaciones sencillas de una incógnita (la igualdad en una expresión simbólica se denota con el símbolo "=="):

```
syms x
sol = solve(7 + x == 2, x)  % devuelve el valor de 'x' que cumple la ecuación '7+x=2'
```

sol = -5

Fíjate que la variable *sol* es simbólica. Para convertirla a tipo numérico podemos utilizar el comando *double:*

```
class(sol)  % variable de tipo simbólico
```

```
ans = 
'sym'
```

```
class(double(sol))  % variable de tipo numérico
```

```
ans = 
'double'
```

Por defecto, el comando *solve* resuelve la ecuación que le pasemos igualada a 0. Las siguientes dos expresiones son equivalentes en MATLAB:

```
sol1 = solve(6 + 2*x == 0, x)  % devuelve el valor de 'x' que cumple la ecuación '6+2x=
```

sol1 = -3

```
sol2 = solve(6 + 2*x, x)  % devuelve el valor de 'x' que cumple la ecuación '6+2x=0'
```

sol2 = -3

Nota importante: Las funciones *rref* y *rank* pueden dar lugar a resultados incorrectos cuando actúan sobre matrices simbólicas, por lo que restringiremos su uso únicamente a matrices numéricas. Por contra, la función *det* puede utilizarse sin riesgo en cualquier caso, tanto con matrices numéricas como con matrices simbólicas.

Ejercicios propuestos

Ejercicio 1:

Determina para qué valores del parámetro a será ortogonal la matriz A:

$$A = \begin{bmatrix} 0.8 & a & 0 \\ -a & 0.8 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

```
syms a real  % defino la variable simbolica 'a' como real
A = [0.8 a 0; -a 0.8 0; 0 0 1];  % defino la matriz A

% Por definción, una matriz ortogonal ha de cumplir A*A' = I. Con solve
% podríamos resolver directamente esa ecuación matricial
% formado, en el que nuestra incógnita será 'a'
sol = solve(A*A' == eye(3), a)
```

sol =

$$\begin{pmatrix} -\frac{3}{5} \\ \frac{3}{5} \end{pmatrix}$$

```
class(sol)  % la variable sol es una estructura de MATLAB
```

ans =
'sym'

```
% Por tanto, la matriz A será ortogonal en los siguientes casos:
% Caso 1: a=-3/5
% Caso 2: a=-3/5

% Comprobación de que nuestro resultado es correcto:
% Caso 1:
subs(A, a, 3/5)*subs(A', a, 3/5)
```

ans =

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

```
% Caso 2:
subs(A, a, -3/5)*subs(A', a, -3/5)
```

ans =

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

Ejercicio 2:

Determina el rango de A en función del parámetro a:

$$A = \begin{bmatrix} 2 & 2a & 5 & 3 \\ 7 & -2 & 4 & -3 \\ 9 & 7 & 1 & -a \\ 1 & 7 & 0 & -2 \end{bmatrix}$$

```
syms a real  % defino 'a' como variable simbólica
A = [2 2*a 5 3; 7 -2 4 -3; 9 7 1 -a; 1 7 0 -2];  % matriz A
a = double(solve(det(A) == 0)) % rg(A) < 4 cuando 'a' = {-20.8292, 11.2042},
```

```
a = 2×1
  -20.8292
   11.2042
```

```
% rg(A) = 4 para cualquier otro valor de a
rank(double(subs(A, a(1)))) % comprobación para a = -20.8292
```

```
ans = 3
```

```
rank(double(subs(A, a(2)))) % comprobación para a = 11.2042
```

```
ans = 3
```

```
rank(double(subs(A, 0))) % comprobación para a = 0 (por ejemplo)
```

```
ans = 4
```

Ejercicio 3:

Sea la matriz $A = \begin{bmatrix} 1 & -1 & 2 \\ a & 0 & 1 \\ 2 & a & -2 \end{bmatrix}$. Determina si es o no regular (es decir, invertible), en función del parámetro a.

```
syms a real  % variable simbólica 'a'
A = [1 -1 2; a 0 1; 2 a -2]  % matriz A
```

A =

$$\begin{pmatrix} 1 & -1 & 2 \\ a & 0 & 1 \\ 2 & a & -2 \end{pmatrix}$$

```
det(A)  % expresión simbólica del determinante de A
```

ans = $2\,a^2 - 3\,a - 2$

```
solve(det(A) == 0, a)  % A será regular cuando su determinante sea distinto de 0,
```

ans =

$$\begin{pmatrix} -\frac{1}{2} \\ 2 \end{pmatrix}$$

```
% lo que ocurrirá para cualquier 'a' distinto de -0.5 y 2
```