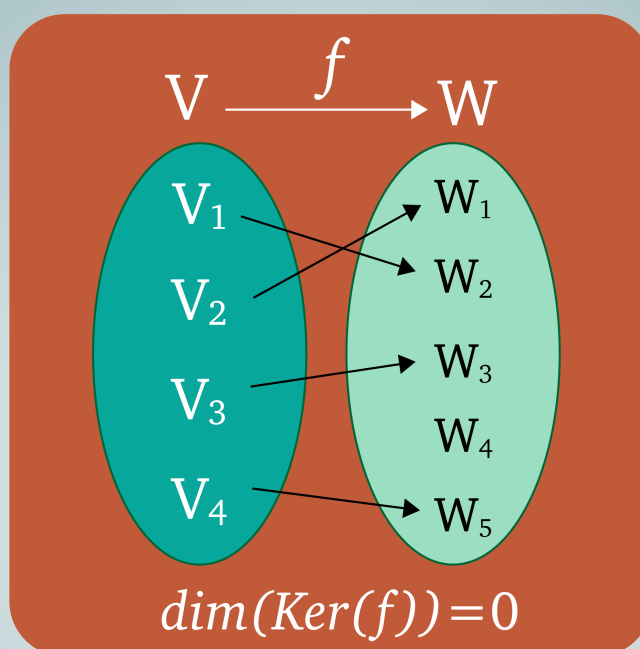


Álgebra

Práctica 3. Variables simbólicas



Rodrigo García Manzananas
Neila Campos González
Ana Casanueva Vicente

Departamento de Matemática Aplicada y
Ciencias de la Computación

Este tema se publica bajo Licencia:

[Creative Commons BY-NC-SA 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)





Grado en Ingeniería Química

G320: Álgebra

Práctica 3: Variables simbólicas

Rodrigo García Manzanos (rodrigo.manzanos@unican.es)

Neila Campos González (neila.campos@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 = x2 + 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 % 'x' sólo podría tomar valores enteros  
syms x real % 'x' podría tomar cualquier valor real
```

En esta asignatura trabajaremos siempre con variables simbólicas de tipo real.

Por último veremos el comando *solve*, que 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 real  
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 hay que 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=0'  
  
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 :

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

Ejercicio 2:

Determina el rango de A en función del parámetro a :

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

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 .