

## 2.1 Generación y manejo de vectores y matrices.

### ▪ Generación de vectores de forma directa

En el lenguaje M se va a trabajar habitualmente con los corchetes ([ ]) para la generación de vectores y matrices. Su uso implica un ensamblado de los elementos que aparecen en su interior para formar una construcción más compleja. Hacemos notar que no tiene el significado matemático similar a los paréntesis.

Para definir un vector no hace falta establecer de antemano su tamaño (de hecho, éste cambia de forma dinámica cuando es preciso). Simplemente, se disponen los valores de los elementos que lo van a componer entre corchetes, separados por espacios o una coma, en el caso de un vector fila, o por el carácter punto y coma (;) o pulsaciones intro, en el caso de un vector columna.

Al teclear

```
>>b=[1 2 3 4 5]
```

o bien

```
>>b=[1,2,3,4,5]
```

se genera el vector fila b:      1 2 3 4 5,

que aparecerá como tal en la ventana *Workspace*.

Mientras que:

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

o bien

```
>>c=[1
```

```
2
```

```
3]
```

genera el vector columna c:

```
1
```

```
2
```

```
3
```

### ▪ Generación rápida de vectores. Operador (:)

Se van a analizar a continuación otras formas de generación de vectores que no necesitan de la escritura explícita de todos sus elementos. Implícitamente se trabajará con el operador (:) que sirve para crear sucesiones numéricas, por ejemplo:

1:4, es equivalente a 1,2,3,4

```
variable=[vin:vfin]
```

Define el vector cuyos primer y último elemento son los especificados por `vin` y `vfin`, estando los componentes intermedios separados por una unidad. Está permitido no utilizar los corchetes o sustituirlos por paréntesis.

```
variable=[vin:incr:vfin]
```

Define el vector cuyos primer y último elemento son los especificados por `vin` y `vfin`, estando los componentes intermedios separados por `incr`. Está permitido no utilizar los corchetes o sustituirlos por paréntesis.

```
variable=linspace (x1,x2,n)
```

Genera un vector con `n` valores igualmente espaciados entre `x1` y `x2`.

Ejemplos:

```
>>v=[1:10]

v=

    1  2  3  4  5  6  7  8  9 10

>>v=1:10

v=

    1  2  3  4  5  6  7  8  9 10

>>v=(1:10)

v=

    1  2  3  4  5  6  7  8  9 10

>>v=[1:2:10]

v=

    1  3  5  7  9

>> v=linspace(1,10,7)

v =

    1.0000    2.5000    4.0000    5.5000    7.0000    8.5000   10.0000
```

## ▪ Extracción de elementos de vectores

Para extraer uno o varios elementos de un vector se debe indicar después del nombre de éste y entre paréntesis, los índices correspondientes a las posiciones a extraer. En lenguaje M la numeración de los índices comienza en 1. Se puede utilizar la partícula `end` para indicar el último elemento.

Sea el vector  $x$ :

`x(n)` devuelve el componente  $n$ -ésimo del vector  $x$ .

`x(end)` devuelve el último elemento del vector  $x$ .

`x(a:b)` devuelve los componentes del vector  $x$  situados entre el  $a$ -ésimo y el  $b$ -ésimo respectivamente ( $a < b$ ).

`x(a:p:b)` devuelve los elementos del vector  $x$  situados entre el  $a$ -ésimo y el  $b$ -ésimo respectivamente ( $a < b$ ), pero separados en  $p$  unidades.

`x(b:-p:a)` devuelve los componentes del vector  $x$  situados entre el  $b$ -ésimo y el  $a$ -ésimo respectivamente ( $a < b$ ), pero separados en  $p$  unidades,

Ejemplos:

```
>>v=[2:2:20]

v=

     2     4     6     8    10    12    14    16    18    20

>>v(7)

    14

>>a=v(1:5)

a=

     2     4     6     8    10

>>b=v(2:3:10)

b=

     4    10    16

>>c=v(10:-3:2)

c=

    20    14     8
```

### ▪ Generación de matrices de forma directa

Se seguirá el mismo proceso estudiado en vectores. Los elementos se escribirán por filas, separando una fila de la siguiente por el carácter punto y coma (;) o pulsaciones intro. Los elementos de cada fila se separan por espacios o una coma.

Por ejemplo, el siguiente comando define una matriz **A** de dimensión (3x3):

```
A=[1 2 3; 4 5 6; 7 8 9]
```

La respuesta del programa es:

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

Sin embargo, aunque las matrices haya que introducirlas por teclado ordenadas por filas, MATLAB las memoriza de forma lineal (a modo de vector), ordenando los elementos por columnas. Por ejemplo, la matriz **A** del caso anterior tendría la siguiente disposición en memoria:

1	4	7	2	5	8	3	6	9
---	---	---	---	---	---	---	---	---

### ▪ Generación rápida de matrices. Operador (:)

Al igual que en el caso de vectores, se puede generar los elementos de las filas sin tener que escribirlos uno a uno. Ejemplo:

```
>> A=[1:5;5:-1:1;linspace(0,11,5)]
```

A =

```
1.0000    2.0000    3.0000    4.0000    5.0000
5.0000    4.0000    3.0000    2.0000    1.0000
0         2.7500    5.5000    8.2500   11.0000
```

### ▪ Composición de matrices

Un caso especialmente interesante es el de crear una nueva matriz componiendo como submatrices otras matrices definidas previamente. Se utiliza la misma técnica de ensamblado vista en los apartados anteriores. A modo de ejemplo, ejecútense las siguientes líneas de comandos y obsérvense los resultados obtenidos:

```

>> A=[0, 1; 1, 2];
>> B=[7, 3; 5, -8];
>> C=[A ; B]
C =
     0     1
     1     2
     7     3
     5    -8

>> D=[11; 12; 13; 14]
D =
    11
    12
    13
    14

>> X=[C, D]
X =
     0     1    11
     1     2    12
     7     3    13
     5    -8    14

```

La matriz **C** de tamaño 4x4 se forma por composición en vertical de las matrices **A** y **B**. La matriz **X** se forma por composición en horizontal de las matrices **C** y **D**. Al igual que con simples escalares, las submatrices que quieran unirse en horizontal se separan con *blancos* o *comas*, mientras que las que se ensamblen en vertical se separan entre sí con el carácter *intro* o *punto y coma*. Los tamaños de las submatrices deben de ser coherentes (igual número de filas para ensamblado horizontal e igual número de columnas para ensamblado vertical).

Se pueden realizar varias uniones en la misma sentencia colocando las parejas de corchetes adecuadamente:

```

>> X=[ [A ; B], D]
X =
     0     1    11
     1     2    12
     7     3    13
     5    -8    14

```

### ▪ Creación de submatrices

Para extraer uno o varios elementos de una matriz se procede igual que con los vectores pero indicando las posiciones de filas y columnas de los elementos a extraer. Veamos a continuación todas las posibilidades:

$A(m, n)$  define el elemento  $(m, n)$  de la matriz  $A$ .

$A(a:b, c:d)$  define la submatriz de  $A$  formada por la intersección de las filas que hay entre la  $a$ -ésima y la  $b$ -ésima y las columnas que hay entre la  $c$ -ésima y la  $d$ -ésima.

$A(a:p:b, c:q:d)$  define la submatriz de  $A$  formada por la intersección de las filas que hay entre la  $a$ -ésima y la  $b$ -ésima tomándolas de  $p$  en  $p$ , y las columnas que hay entre la  $c$ -ésima y la  $d$ -ésima tomándolas de  $q$  en  $q$ .

$A([a\ b], [c\ d])$  define la submatriz de  $A$  formada por la intersección de las filas  $a$ -ésima y  $b$ -ésima y las columnas  $c$ -ésima y  $d$ -ésima.

$A([a\ b\ c\ \dots], [e\ f\ g\ \dots])$  define la submatriz de  $A$  formada por la intersección de las filas  $a, b, c, \dots$  y las columnas  $e, f, g, \dots$ .

$A(:, c:d)$  define la submatriz de  $A$  formada por la intersección de todas las filas de  $A$  y las columnas que hay entre la  $c$  y la  $d$ .

$A(end, c:d)$  define la submatriz de  $A$  formada por la intersección de la última fila de  $A$  y las columnas que hay entre la  $c$  y la  $d$ .

$A(:, [c\ d\ e\ \dots])$  define la submatriz de  $A$  formada por la intersección de todas las filas de  $A$  y las columnas  $c, d, e, \dots$ .

$A(a:b, :)$  define la submatriz de  $A$  formada por la intersección de todas las columnas de  $A$  y las filas que hay entre la  $a$  y la  $b$ .

`A([a b c], :)` define la submatriz formada por la intersección de todas las columnas de A y las filas a,b,c.

`A(a, :)` define la fila a-ésima de la matriz A.

`A(:, b)` define la columna b-ésima de la matriz A.

`A(:)` define un vector columna cuyos elementos son las columnas de A colocadas por orden una debajo de otra.

`A(b)` define el elemento b-ésimo al colocar una columna de la matriz a continuación de la otra.

Ejemplos:

```
>>A=[1 2 3 4;5 6 7 8;9 10 11 12;13 14 15 16];
```

```
>>A(3,4)
```

```
12
```

```
>>B1=A(1:3,2:4)
```

```
2 3 4
```

```
6 7 8
```

```
10 11 12
```

```
>> B2=A([1 3],[2 4])
```

```
2 4
```

```
10 12
```

```
>>B3=A(:,2:4)
```

```
2 3 4
```

```
6 7 8
```

```
10 11 12
```

```
14 15 16
```

```
>>B4=A(:, [1 3 4])
```

```
1 3 4
```

```
5 7 8
```

```
9 11 12
```

```
13 15 16
```

```
>>B5=A(2:4,:)
```

```
5 6 7 8
```

```
9 10 11 12
```

```
13 14 15 16
```

```
>>B6=A(3,:)
```

```
9 10 11 12
```

```
>>B7=A(:,3)
```

```
3
```

```
7
```

```
11
```

```
15
```

```
>>B8=A(:) El resultado es el siguiente vector (en columna):
```

```
1 5 9 13 2 6 10 14 3 7 11 15 4 8 12 16
```

```
>>B9=A(7)
```

```
10
```