Tema 3: Vectores y matrices. Conceptos básicos

1. Definición

Matlab está fundamentalmente orientado al trabajo y el cálculo matricial. Veremos que las operaciones están definidas para el trabajo con este tipo de elementos. Antes de empezar a manejar y operar con ellas veamos cómo se definen.

Como en casi todos los lenguajes de programación, en Matlab las matrices y vectores son variables a las que se les puede dar nombres. Para definir una matriz no hace falta establecer de antemano su tamaño (de hecho, se puede definir un tamaño y cambiarlo posteriormente). Matlab determina el número de filas y de columnas en función del número de elementos que se introducen (o se utilizan). Las matrices se definen con los elementos entre corchetes y por filas; los elementos de una misma fila están separados por blancos o comas, mientras que las filas están separadas por pulsaciones intro o por caracteres punto y 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
```

Veamos en el Workspace como las almacena Matlab (figura 12):

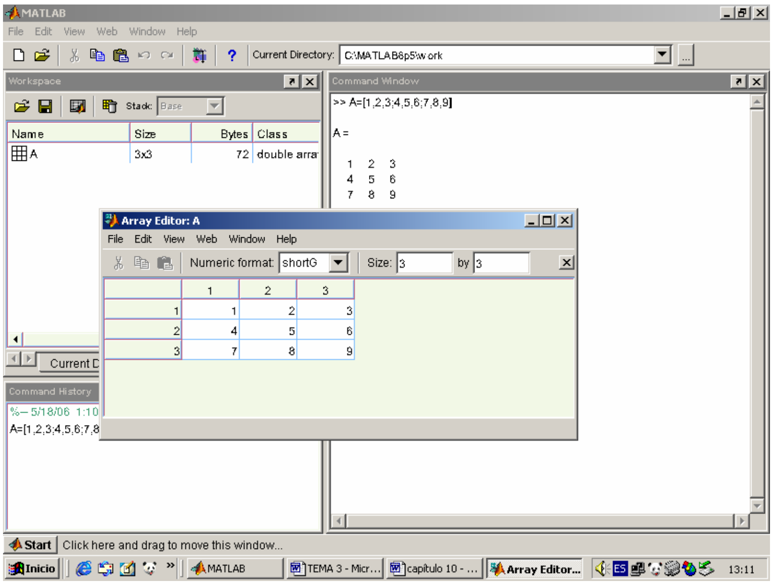

Figura 12

A partir del momento en que tenemos definidas diversas matrices, se pueden operar. Matlab puede hacer esto por medio de operadores o por medio de funciones.

2. Operaciones elementales

Operaciones básicas como suma, producto o trasposición de hacen como se muestra a continuación, permitiéndose algunas operaciones no definidas matemáticamente:

Operador suma (+)

Utilizado entre matrices de iguales dimensiones, obtiene la suma elemento a elemento. Utilizado entre una matriz y un escalar, suma el escalar a cada elemento de la matriz.

Operador resta (-)

Idéntico a la suma en su utilización.

Operador producto (*)

Utilizado entre matrices calcula el producto matricial. Las dimensiones de las matrices deben ser congruentes. Utilizado entre una matriz y un escalar, multiplica el escalar por cada elemento de la matriz.

Operador producto elemento a elemento (.*)

Se utiliza entre dos matrices de iguales dimensiones y multiplica elemento a elemento, obteniendo otra matriz de igual dimensión.

Operador potenciación (^)

Si c es un entero y A es una matriz cuadrada, A^c calcula el producto A*A*A .... *A, c veces.

Operador potenciación elemento a elemento (.^)

A.^B da como resultado una matriz cuyo elemento ij es a_{ij}^b_{ij}.

A.^c da como resultado una matriz cuyo elemento ij es a_{ij}^c.

c.^A da como resultado una matriz cuyo elemento ij es c^a_{ij}.

Operador división (/) (\)

En Matlab existe el operador división a la derecha (/) y división a la izquierda (\).

La utilización entre matrices es la siguiente:

- \ división-izquierda**:** A\B

  Si A es cuadrada A\B=inversa(A)*B. Si A no es cuadrada A\B es la solución en el sentido de mínimos cuadrados del sistema AX=B.
- / división-derecha: A/B

  Si B es cuadrada A/B=A*inversa(B). Si B no es cuadrada, A/B es la solución del sistema XB=A.

Operador división elemento a elemento (./) (.\)

A./B da como resultado una matriz cuyo elemento ij es a_{ij} /b_{ij}.

A.\B da como resultado una matriz cuyo elemento ij es b_{ij} /a_{ij}.

Operador traspuesta (‘)

A’ da como resultado la matriz transpuesta de A.

3. Operaciones por medio de funciones

En menú de la ayuda: matlab\elmat - Elementary matrices and matrix manipulation y matlab\matfun - Matrix functions - numerical linear algebra, se pueden encontrar las diversas funciones que se aplican a las matrices. Destacamos las más elementales como:

inv(A) da como resultado la matriz inversa de A.

det(A) da como resultado el determinante de A.

trace(A) da como resultado la traza de A.

rank(A) da el rango de A

Debe destacarse que la mayoría de las funciones definidas en el programa se ejecutan sobre cualquier matriz aplicándose elemento a elemento.

Por ejemplo:

```
>> Sin(A)
```

Ofrece como respuesta una matriz del mismo tamaño que A cuyos elementos son el seno del correspondiente elemento de A:

0.8415	0.9093	0.1411
-0.7568	-0.9589	-0.2794
0.6570	0.9894	0.4121

4. Otras formas de definir matrices

Algunas veces, introducir matrices por el teclado no es práctico. Veremos otras formas más potentes de generar matrices que luego, tras modificaciones nos pueden llevar a definir la que nos interesa. Los comandos a utilizar se encuentren en matlab\elmat.

Algunas de estas funciones son las siguientes:

eye(n) forma la matriz *identidad* de tamaño (nxn)

zeros(m,n) forma una matriz de *ceros* de tamaño (mxn)

zeros(n) forma una matriz de *ceros* de tamaño (nxn)

ones(n) forma una matriz de *unos* de tamaño (nxn)

ones(m,n) forma una matriz de *unos* de tamaño (mxn)

rand: este comando genera números pseudoaleatorios distribuidos uniformemente entre 0 y 1. Cada llamada proporciona un nuevo número.

rand(n): genera una matriz de números pseudoaleatorios entre 0 y 1, con distribución uniforme, de tamaño nxn.

rand(m,n): igual que en el caso anterior pero de tamaño mxn.

5. Manipulación de matrices

Con este programa es posible crear matrices a partir de una dada, extraer o cambiar elementos de una matriz y, en general, manipular de casi cualquier forma estos elementos. Veamos los diversos caminos para realizarlo:

- Crear una nueva matriz que reciba alguna característica o propiedad de la de partida
- Componer una matriz a partir de submatrices más pequeñas.
- Mediante la manipulación de los elementos de la matriz de partida.

Características de una matriz:

Destacamos algunos comandos que permiten trabajar con las características o elementos de una matriz de partida:

size (A), length(A) nos dan información sobre las características de la matriz A, en este caso dimensión de la matriz o longitud del vector.

zeros (size(A)) genera una matriz de ceros del mismo tamaño que A.

ones(size(A)) lo mismo con matriz de unos

A=diag(x) devuelve una matriz diagonal cuyos elementos de la diagonal son los del vector x.

triu(A) y **tril(A)** forman matrices triangulares superiores e inferiores a partir de la matriz A.

Composición de matrices

Crear una matriz a partir de submatrices es algo muy sencillo para el programa que sin embargo es de mucha utilidad por ejemplo a la hora de añadir datos o ampliar información.

Si contamos con la submatrices A=[1 2; 3 4] y B=[1,6], C=[A; B] genera la matriz:

1 2

3 4

1 6

Si tenemos con los vectores u=[1 2 3 4] y v=[1 6], w=[u v] devuelve:

w=[1 2 3 4 1 6]

Manipulación de elementos

Veremos como extraer, cambiar o eliminar elementos o líneas (filas o columnas de una matriz).

Dado un vector x, la ejecución de **x(i)** devuelve el elemento situado en la posición i de ese vector.

En el caso matricial **A(i,j)** devuelve el elemento situado en la fila i columna j de la matriz A.

El operador **(:)** es de gran importancia en Matlab. Puede decirse que es un operador que respeta el rango. Veamos su utilidad con algunos ejemplos:

```
>> x=1:10
x =
    1     2     3     4     5     6     7     8     9    10
>> x=[0:2:10]
x =
```

```
     0     2     4     6     8    10
```

El primer número indica el valor inicial, el segundo el paso del incremento y el tercero el elemento final. Como hemos visto si se omite el intermedio, por defecto se toma la unidad.

Nota: En general, el operador genera vectores fila. Conocido este operador, podemos decir que con él es posible definir variables vectoriales:
x= [p:q] general un vector de primer elemento p, último q y los elementos intermedios se diferencian en una unidad
x= [p:i:q] devuelve un vector de primer elemento p, último q y los elementos intermedios se diferencian en i unidades.
Destacar también el comando linspace: x=linspace(p,q,n) genera un vector de n elementos uniformemente espaciados desde p hasta q.

A esto se le puede dar gran número de aplicaciones:

Recordando que el vector x es: x=[1 2 3 4 5 6 7 8 9 10]:

```
>> x(1:4)
ans =
       1     2     3     4

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

>> A(1,:)
ans =
     1     2

>> A(:,2)
ans =
     2
     4
     6
```

Práctica 3: Matrices y vectores

1. Introducir los vectores (1 2 3 4 5) y (6 7 8 9 10) asignándoles las variables u y v respectivamente:
   a. Determinar 3u, u+v, u-v.
   b. Construir un vector cuyos elementos sean los de v incrementados 3 unidades.
   c. Determinar un vector de elementos el resultado de multiplicar cada elemento de u por el correspondiente de v.
   d. Calcular un vector de elementos los de u elevados al cubo.
   e. Calcular un vector cuyos elementos sean el resultado de elevar cada elemento de u al elemento de v correspondiente.

2. Introducir las matrices $A = \begin{pmatrix} 1 & 2 & 1 \\ 5 & 4 & 3 \\ 7 & 1 & 0 \end{pmatrix}, B = \begin{pmatrix} -2 & 5 & 7 \\ 4 & 3 & 1 \\ 8 & -1 & 3 \end{pmatrix}$
   a. Calcular A+B, AB, A^4
   b. Determinar una matriz cuyos elementos sean el resultado de multiplicar cada elemento de A por el correspondiente de B.
   c. Determinar una matriz cuyos elementos sean el resultado de dividir cada elemento de A por el correspondiente de B.

3. Determinar si es posible:
   a. La inversa de A y de B. Verificar que el producto de una matriz por su inversa es la matriz identidad.
   b. La traza de B.
   c. El determinante y el rango de A.

4. Una empresa compra los siguientes artículos:

Referencia artículo	Cantidad de artículo	Precio de la unidad (sin IVA)
100	200	190
101	150	345
102	500	69
103	49	598

   a. Introducir la tabla en mediante tres vectores: referencia, cantidad y coste.
   b. Determinar el coste total de cada producto.
   c. Construir una tabla con cada artículo y su coste total.
   d. Calcular el coste total a pagar por la empresa incluyendo un 16% de IVA

5. Introducir los vectores u=(2,3,4) y v=(3,-4,8).
   a. Determinar la suma y el producto de todos los elementos de u.
   b. Calcular el máximo y mínimo de los elementos de v, así como el lugar donde están situados.
   c. Calcular el producto escalar de u y v.
   d. Determinar el módulo de los vectores.

6. Construir los vectores cuyos elementos sean:
   a. Los números naturales comprendidos entre el 10 y el 100.
   b. (-1, -0,8, -0.6,..........., 1.6, 1.8, 2).
   c. Desde el 1 hasta el 3 igualmente espaciados y con un total de 38 elementos.

7. Dados u=(1,2,3), v=(4,5,6),
   a. Construir el vector (0,1,2,3) a partir de u.
   b. Construir el vector de elementos los de u y v

8. Construir un vector w con los cuadrados de los 15 primeros números naturales.
   a. Extraer el cuadrado de 7.
   b. Extraer los cuadrados de los elementos que van desde el 2 al 6 ambos inclusive.
   c. Extraer los cuadrados de los elementos que van desde el 7 al 13 ambos inclusive en sentido inverso
   d. Construir, a partir de w un vector con los cuadrados de 1, 3, 7,14.

9. Introducir la matiz $A = \begin{pmatrix} 2 & -8 & 3 & 4 \\ 1 & 2 & 4 & -2 \\ -1 & 3 & 6 & -9 \\ 5 & 7 & -4 & 0 \end{pmatrix}$
   a. Extraer el elemento a_{23}.
   b. Sustituir el elememto a_{22} por 100.
   c. Construir una submatriz de A formada por las filas 2, 3 y 4 y las columnas 1, 2 y 3 de A.
   d. Extraer la fila 3 de A.
   e. Extraer la columna 1 de A.

f. Construir una matriz formada por las filas 1, 2 y 3 de A.

g. Construir una matriz formada por las filas 1 y 4 de A.

10. Construir una matriz A cuadrada aleatoria de orden 3.

a. Obtener su inversa, su transpuesta y su diagonal.

b. Transformarla en una matriz triangular inferior.

c. Obtener la suma de los elementos de la primera fila, de la segunda columna y de la diagonal.

11. Estudiar, según el teorema de Rouche-Frobenius, y resolver el sistema:

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

12. Resolver el sistema $\begin{cases} x+3y+5z+t=1 \\ 2x+y+3z+t=2 \\ 4x+y+2z+2t=1 \\ 5x+y+2z+3t=5 \end{cases}$ utilizando:

a. La matriz inversa de los coeficientes si existe.

b. El operador división izquierda.