



Universidad del País Vasco Euskal Herriko Unibertsitatea

Fundamentos de Informática

2. Teoría

Vectores y Matrices

Objetivos:

Aprender la sintaxis y el uso de vectores y matrices:
definición, indexación, selección de filas y columnas
Operadores para matrices
Operadores para aplicar elemento por elemento
Transposición, determinante, rango e inversa

Operaciones básicas

a) Crea un vector:

```
v=[1;2;3]
```

b) Crea una matriz-fila:

```
mf=[1 2 3]; % el punto y coma hace que no salga en pantalla  
mf % la visualizamos  
mf=[1,2,3]; % dos formas de separación
```

c) Crea una matriz:

```
m=[1 2 3; 4,5,6];  
whos % para ver qué variables hay definidas en este ámbito o espacio de trabajo  
m % visualizamos m  
t=length(1:5)  
t=size(m) % dimensiones de la matriz en t  
[mf,mc]=size(m) % en dos valores
```

d) Crea una matriz con un rango y con un incremento definido:

```
mr=[1:6]  
mi=1:0.3:5 % sin corchetes también es posible  
1:0.3:5 % el resultado se queda en ans  
ans  
1:1/3:5 % el incremento es un tercio  
x=10:-1:1  
linspace(0,20,5) % cinco elementos espaciados uniformemente entre 0 y 20
```

e) Selecciona un elemento (*indexación*):

```
m=1:0.3:5  
m(3) % tercer elemento de la variable
```

f) Selección usando varios índices:

```
i=[3 5];  
m(i) % elementos 3 y 5 de m
```

Fundamentos de Informática Tema 2

g) Selecciona un elemento, una fila o una columna:

```
m=[1 2 3; 4,5,6]
m(2,2) % elemento que vale 5
m(4) % equivalente a la anterior
mi(2,:) % segunda fila, todos los elementos de la fila
m(2,2:3) % segunda fila, elementos de la columna 2 hasta la 3
m(:,2) % segunda columna entera
m(:,end) % última columna entera
m(:,2:3) % columnas de la 2 a la 3
m(:,2:end) % columnas de la 2 a la última
m(:, [1 3]) % columnas 1 y 3
m(:,3)=[] % borrado de la columna 3
```

h) Añade un elemento al final de una variable

```
datos=[] % vacía
datos=[datos,2,4,6] % añade tres elementos en nuevas columnas
datos=[datos,1,3,5] % añade otros tres más
datos=[datos;1:6] % otra fila de seis
fila=datos(2,:) % seleccionamos una fila
lg=length(fila) % longitud de una fila
datos=[datos;1:lg] % una fila más
datos=[datos;1:length(datos(2,:))] % todo de un tirón
```

i) Operaciones de matrices:

```
A=[1 4 -3; 2 1 5; -2 5 3]
B=inv(A) % inversa de la matriz
B*A % la multiplicación de matrices da la diagonal unidad
A' % tranposición
det(A) % determinante de la matriz
rank(A) % rango de la matriz
diff(A) % matriz con las diferencias entre elementos consecutivos
% en columnas
sum([3 5 7]) % suma los elementos de las columnas de la matriz
```

j) Operaciones aplicadas a los elementos de una matriz:

```
M=[1,2,3;4,5,6;7,8,9]
2.*M % multiplica cada elemento
A.^2 % cada elemento al cuadrado
A^2 % lo mismo que A*A (multiplica la matriz consigo misma)
vec1 = 1:5;
vec2 = [3 5 8 2];
numel(vec1) % número de elementos
min(vec1) % mínimo
max(vec2) % valor máximo
sum(vec1) % suma de los elementos
prod(vec2) % producto de los elementos
cumsum(vec1) % suma acumulativa de los elementos anteriores
(1,1+2,1+2+3,1+2+3+4,1+2+3+4+5)
ans =
1 3 6 10 15
```

Fundamentos de Informática Tema 2

k) Matrices lógicas:

```
v=[5 9 4 3 8 8]
g=v>5 % g es un vector lógico (true=1, false=0)
v(g) % sólo los elementos de v que cumplen la condición
find(v > 5) % los índices de los que cumplen
v(find(v > 5)) % lo mismo que v(g)
v(v==4)=[] % borra las apariciones del valor 4
v(v<=5)=[] % borra los valores menores o iguales a 5
any(g) % verdadero, true, 1 si alguno es verdadero
any(v==8) % hay algún 8 en la matriz
all(g) & verdadero si todos son verdadero.
```

l) Generar una matriz unidad:

```
E=eye(4)
```

m) Generar matrices de ceros y de unos:

```
zeros(3,5)
zeros(4)
ones(3)
ones(3,1)
ones(2,4)
```

n) Generar una matriz mágica:

```
MM=magic(4) % help magic para saber qué es una matriz mágica
```

o) Generar una matriz aleatoria:

```
r=rand(10,10)
R=random('Uniform',-10, 10, 1, 20) % Distribución uniforme entre -10 y
10, una fila y 20 columnas
randi([-10 10],1,20) % Lo mismo en enteros
```

p) Otras formas de generar matrices:

```
[m,n]=size(A) % devuelve el número de filas y de columnas de la
matriz A. Si la matriz es cuadrada basta recoger el
primer valor de retorno
n=length(x) % calcula el número de elementos de un vector x
zeros(size(A)) % forma una matriz de ceros del mismo tamaño que una
matriz A previamente creada
ones(size(A)) % ídem con unos
A=diag(x) % forma una matriz diagonal A cuyos elementos
diagonales son los elementos de un vector existente x
x=diag(A) % forma un vector x a partir de los elementos de la
diagonal de una matriz ya existente A
triu(A) % forma una matriz triangular superior a partir de una
matriz A (no tiene por qué ser cuadrada)
tril(A) % ídem con una matriz triangular inferior
rot90(A,k) % Gira k*90 grados la matriz rectangular A en sentido
antihorario.
flipud(A) % halla la matriz simétrica de A respecto de un eje
horizontal
```

Fundamentos de Informática Tema 2

```
fliplr(A)          % halla la matriz simétrica de A respecto de un eje  
                  vertical
```

q) Generar una matriz de dos filas y Npuntos columnas con una distribución uniforme entre -R y R:

```
R=1;  
Npuntos=100; % columnas  
Datos=random('Uniform',-R, R, 2, Npuntos);  
  
% Distancia al centro de todos los puntos (cada punto es una columna)  
distancias=sqrt(Datos(1,:).^2+Datos(2,:).^2);  
  
% Filtramos y nos quedamos todos los de dentro del círculo  
DatosCirculo=Datos(:,distancias>R);  
distancias(distancias>R)=[];  
Ncirculo=length(distancias);  
  
PiEstimado=4*Ncirculo/Npuntos % calculamos  $\Pi$ 
```