Herramientas computacionales para la matemática

MATLAB: Arreglos

Verónica Borja Macías

Marzo 2013

- Un arreglo es una estructura que MATLAB utiliza para almacenar y manipular datos.
- Es una lista de números dispuestos en filas y/o columnas.
- Los arreglos pueden ser:
  - Unidimensionales (vectores)
  - Bidimensionales (matrices)
  - Mas de dos dimensiones (hipermatrices).

Matlab

Arreglos: Vectores

- Un arreglo unidimensional es una sucesión de números distribuidos en una fila o en una columna.
- En MATLAB un vector renglón se crea asignando sus elementos a una variable, para ello introducimos los valores deseados separados por espacios (o comas) todo ello entre corchetes [].
- Si deseamos un vector columna entonces introducimos los valores separados por punto y coma, todo ello entre corchetes [].
- Generalmente usaremos letras mayúsculas cuando nombremos a las matrices y minúsculas para vectores y escalares. Esto no es imprescindible y Matlab no lo exige, pero resulta útil.

Ejemplos

```
>> x = [5 7 -2 4 -6] % es un vector, los elementos los separamos con espacios
x =
    5    7    -2    4    -6
>> y = [2,1,3,7] % es otro vector, los elementos los separamos con comas
y =
    2    1    3    7
>> z = [0 1 2,3 4,5] % da igual separar los elementos por comas o espacios
z =
    0    1    2    3    4    5
>> w = [1 ;2; 3] % es un vector columna
w =
    1
    2
    3
```

- Para acceder a los elementos individuales de un vector lo haremos utilizando subíndices, x(n) es el n-ésimo elemento del vector x.
- Si queremos acceder al último podemos indicarlo usando end como subíndice.

Ejemplos

```
>> x = [5 7 -2 4 -6];
>> x (2) % segundo elemento del vector x
ans =
      7
>> x (end) % último elemento del vector x
ans =
      -6
```

Matlab

Arreglos: Vectores

- Un vector puede cambiar su tamaño, si tiene n elementos, basta añadir nuevos valores para las posiciones n+1, n+2 y así sucesivamente.
- Si es necesario MATLAB asigna ceros a los elementos entre el último del vector original y el añadido.

Ejemplos

```
>> x = [5 7 -2 4 -6];
>> x (6)=8 % agrega un sexto elemento al vector x con un valor de 6
ans =
     5    7    -2    4    -6    8
>> x(10)=7 % llena con ceros las posiciones 7, 8 y 9
ans =
     5    7   -2    4   -6    8    0    0    0    7
```

- También es posible añadir nuevos elementos a un vector ya existente a partir de otros vectores.

Ejemplos

```
>> x = [5 7 -2 4 -6];
>> w = [1 2 3 4];
>> z1=[x w] % une los vectores x y w para formar z1
z1=
    5    7    -2    4    -6    1    2    3    4
>> z2=[x(1) w x(3)] % une el primer elemento de x con el vector w y con el
     tercer elemento de x para formar z2
z2=
   5   1    2    3    4    -2
```

- Para acceder a un bloque de elementos a la vez, se usa la notación de dos puntos (:), así x(m:n) nos da todos los elementos desde el m-ésimo hasta el n-ésimo del vector x.

Ejemplos

```
>> x = [5 7 -2 4 -6];

>> x (2:4) % devuelve desde el segundo hasta al cuarto elemento del vector x
ans =
      7   -2   4
```

Matlab

Arreglos: Vectores

- También es posible eliminar elementos de un vector existente mediante la asignación del vacio [] al elemento o rango de elementos que se deseen eliminar.

Ejemplos

```
>> x = [1 2 3 4 5 6 7]; % eliminamos el cuarto elemento del vector x
>> x(4) = []
x =
   1    2    3    5    6    7
>> x(2:5) = [] % eliminamos los elementos desde la posición 2 hasta la 5
x =
   1    7
```

- Si introducimos un número entre el primero y el segundo también separado por dos puntos (:) se mostrarán los elementos del primero al último indicado, incrementados según el número que aparece en el centro (o decrementados si el número es negativo).

Ejemplos

```
>> x = [5 7 -2 4 -6];

>> x (1:2:5) % devuelve el primero, tercero y quinto elemento del vector x
ans =
      5    -2    -6
```

- Otra forma de obtener un conjunto concreto de elementos del vector es indicando entre corchetes [] las posiciones de los elementos que queremos obtener poniendo paréntesis fuera de los corchetes.

Ejemplos

```
>> x = [5 7 -2 4 -6];

>> x ( [3 5 1] ) % devuelve el tercer, quinto y primer elemento del vector x
ans =
      -2    -6    5
```

Construcción abreviada de algunos vectores

(a:b) crea un vector que comienza en el valor a y acaba en el valor b aumentando de 1 en 1.

(a:c:b) crea un vector que comienza en el valor a y acaba en el valor b aumentando de c en c.

linspace (a,b,c) genera un vector linealmente espaciado entre los valores a y b con c elementos.

linspace (a,b) genera un vector linealmente espaciado entre los valores a y b con 100 elementos.

logspace (a,b,c) genera un vector logarítmicamente espaciado entre los valores 10^a y 10^b con c elementos.

logspace (a,b) genera un vector logarítmicamente espaciado entre los valores 10^a y 10^b con 50 elementos.

Ejercicios

1. (1:7)
2. 1:7
3. 1:3:10
4. 1:4:10
5. 1:0.1:1
6. 50:-7:1
7. linspace (2,6,3)
8. linspace (2,6,4)
9. linspace (2,10)
10. logspace (0,2,4)
11. logspace (0,25)

- Podemos conocer el tamaño de un vector v con alguna de las funciones length(v) o size(v).
- Es posible realizar operaciones entre vectores y escalares

Expresión	Operación
x + k	Suma a los elementos del vector x el escalar k
x – k	Resta a los elementos del vector x el escalar k
k*x	Multiplicación los elementos del vector x por el escalar k
x / k	División los elementos del vector x por el escalar k
k .^ x	Potenciación del escalar k a cada uno de los elementos de x
x .^ k	Potenciación los elementos del vector x a la potencia escalar k

Expresión	Operación
x + y	Suma de los vectores x e y con el mismo tamaño
x – y	Resta de los vectores x e y con el mismo tamaño
x .* y	Multiplicación elemento a elemento
x ./ y	División elemento a elemento por la derecha
x .\ y	División elemento a elemento por la izquierda
x .^ y	Potenciación elemento a elemento
x '	Transposición compleja conjugada
x .’	Transposición
cross (x,y)	producto cruz de los vectores x e y de dimensión 3
dot (x,y)	producto punto de los vectores x e y

- Si una función matemática predefinida tiene como argumento un vector, obtendremos como resultado un vector del mismo tamaño con entradas obtenidas realizando la operación especificada.

Ejemplos

```
>>v=[1 2 3]
>> sin(v)
ans =
        0.8415    0.9093    0.1411
>> log(v)
ans =
             0    0.6931    1.0986
```

- Con un poco de práctica se aprenderá como evaluar expresiones más complejas, por ejemplo, evaluar la expresión $x^2 - 2x - 3$ para valores de x entre 1 y 10, con incremento de 1, la solución es muy simple:

Ejemplos

```
>> x=1:10
>> y=x.^2-2*x-3
```

Matlab

Arreglos: Operaciones con vectores

Expresión	Operación
prod(x)	calcula el producto de los valores de x.
sum(x)	calcula la suma de los valores de x.
cumprod(x)	calcula un producto acumulado de los valores de x.
cumsum(x)	calcula la suma acumulada de los valores de x.
sort(x)	ordena los elementos del vector x.

Ejercicios

Sean a=(1 2 3), b=(-2 3 5).

1. Sume 3 a cada elemento de a y divida cada elemento de b entre 2.
2. Realiza las operaciones que se indican: a+b, a-b, a·b y a×b.
3. Divide los elementos de a entre los elementos de b
4. Un vector columna que contenga los números impares entre 1 y 1000.
5. Un vector fila que contenga los números pares entre 2 y 1000.
6. Si x=0:2:20, escribe el comando de MATLAB que eleva al cuadrado cada componente de x.
7. Verifique si x=(1,3,2) e y=(-2 2 -2) son ortogonales.

Ejercicios

8. Obtenga un vector ortogonal x=(1,3,2) e y=(2 2 -2).
9. Si x=[0,1,4,9,16,25], calcula la raíz cuadrada de cada componente de x.
10. Si x=0:.1:1, eleva cada componente de x a 2/3.
11. Si x=0:pi/2:2*pi, calcula el coseno de cada componente de x.
12. Si x=-1:.1:1, calcula el seno inverso de cada componente de x.
13. Si x=linspace(0,2*pi,1000), ¿cuál es la entrada 50 de x? ¿Cuál es la longitud de x?
14. Si k=0:100, ¿cuál es la entrada número 12 de y=0.5.^k?
15. Evaluar la expresión sen(x)/x para valores de x entre −1 y 1 con incrementos de 0.1 unidades.