

Vectores and matrices

Oier Lopez de Lacalle y Patxi Ramírez

Languages and Information Systems

2021/2022



GASTEIZKO
INGENIARITZAKO
UNIBERTSITATE ESKOLA
ESCUELA UNIVERSITARIA
DE INGENIERÍA
DE VITORIA-GASTEIZ

Contenido

- 1 **Vectores**
- 2 **Matrices**
- 3 **Operaciones Aritméticas**
- 4 **Vectores y matrices en las funciones**

Contenido

- 1 **Vectores**
- 2 Matrices
- 3 Operaciones Aritméticas
- 4 Vectores y matrices en las funciones

Vectores



Vector

Un **Vector** se usa para almacenar una lista de valores, todos los cuales son del mismo tipo. Un vector puede ser un **vector de fila** o un **vector de columna**.

	1	2	3	4	5
rowVector	5	8	-9	78	13

Vector Fila

colVector

1	5
2	8
3	-9
4	78
5	13

Vector Columna

Vectores fila: creación (I)

La forma más sencilla de crear un vector fila es poner los valores que lo componen entre corchetes.

Sintaxis

```
variable = [lista de valores]
```

Ejemplo

```
rowList=[5 8 -9, 78,13]
```

Note

Los valores de un vector fila pueden estar separados por espacios o comas.

Vectores fila: creación (II)

El operador de dos puntos `:` se puede utilizar para generar vectores cuyos elementos están espaciados regularmente (p.j., `[1 2 3 4]`).

Sintaxis

Dos alternativas

- `start : end`
- `start : step : end`

Ejemplo

```
numberList = 1:8  
numberList = 1:4:8  
numberList = 4:-1:1  
numberList = 9:-3:1  
numberList = 9:3:5  
numberList = 0:0.1:100
```

Vectores fila: creación (III)

La función **linspace** crea un vector espaciado linealmente. Para ello, esta función requiere el rango inclusivo de valores y el número de elementos (N). Si se omite N , el vector creado tendrá 100 elementos.

Sintaxis

Dos alternativas

- **linspace(BASE, LIMIT)**
- **linspace(BASE, LIMIT, N)**

Ejemplo

```
numList=linspace(1,10)
numList=linspace(0,10,51)
numList=linspace(4,-1)
numList=linspace(4,-1,6)
```

Vectores Columna

Creación

La forma más fácil de crear un **vector de columna** es poner explícitamente los valores entre corchetes, separados por punto y coma (;).

Sintaxis

variable = [Value₁; ...]

Ejemplo

numList=[5; 8; -9; 78; 13]

Contenido

- 1 Vectores
- 2 Matrices**
- 3 Operaciones Aritméticas
- 4 Vectores y matrices en las funciones

Matriz



Matriz

La **matriz** es un conjunto de datos del mismo tipo, organizados en filas y columnas.

numMat	1	2	3	4	5
1	5	8	-9	78	13
2	13	18	9	-8	33
3	55	0	-9	-8	-1

Matriz

Matrices: creación (I)

La forma más sencilla de crear una matriz es especificar los valores entre corchetes. Los valores deben organizarse en filas.

Sintaxis

```
varname = [row values; ...]
```

Ejemplo

```
mat=[5 8 -9; 78 13 -21]
```

Importante

Todas las filas deben tener el mismo número de elementos. Las filas están separadas por el operador ; o **ENTER** (una nueva línea).

Matrices: creación (II)

Las funciones que se describen a continuación también se utilizan para crear nuevas matrices.

Función	Descripción
zeros	Crea una matriz de ceros
ones	Crea una matriz de unos
eye	Crea una matriz de identidad
rand	Crea una matriz de números reales aleatorios
randi	Crea una matriz de números enteros aleatorios

Acceder a elementos de la matriz

Acceso a los elementos

Para acceder los elementos de una matriz, se necesitan dos índices, uno para determinar filas y otro para columnas.

`numMat(1,2)`

<code>numMat</code>	1	2	3	4	5
1	5	8	-9	78	13
2	13	18	9	-8	33
3	55	0	-9	-8	-1

`numMat([1 2],[2 3])`

<code>numMat</code>	1	2	3	4	5
1	5	8	-9	78	13
2	13	18	9	-8	33
3	55	0	-9	-8	-1

Prueba `numMat([1 2],[2 4])` y `numMat([1 2],[2:4])`

Importante

Tanto los índices de fila como de columna deben ser valores válidos, con respecto al tamaño de la matriz.

Acceso a filas y columnas de la matriz

Acceso a filas y columnas

Se puede acceder a filas o columnas completas de una matriz utilizando el operador `:`.

`numMat(1,:)`

<code>numMat</code>	1	2	3	4	5
1	5	8	-9	78	13
2	13	18	9	-8	33
3	55	0	-9	-8	-1

`numMat(:,[2 3])`

<code>numMat</code>	1	2	3	4	5
1	5	8	-9	78	13
2	13	18	9	-8	33
3	55	0	-9	-8	-1

Prueba `numMat(:,[3:end])`

Nota

El operador `end` puede usarse para referirse a la última fila o columna de la matriz.

Asignación de nuevos valores una matriz

Asignación de nuevos valores

Se pueden establecer nuevos valores para los elementos de la matriz usando el operador de **asignación**.

`numMat(1,2)=0`

numMat	1	2	3	4	5
1	5	0	-9	78	13
2	13	18	9	-8	33
3	55	0	-9	-8	-1

`numMat([1 2],[2 3])=[1 2; 3 4]`

numMat	1	2	3	4	5
1	5	1	2	78	13
2	13	3	4	-8	33
3	55	0	-9	-8	-1

Aviso

La submatriz a actualizar y la que se asignará deben tener las mismas dimensiones. Excepción, cuando se asigna un valor único, el valor se repetirá (*broadcasting*).

Dimensiones de la matriz

La función **size** puede usarse para obtener cuántas filas y columnas tiene una matriz.

Syntax

```
size(matrix)
```

Example

```
[rows,cols]=size(numMat);
```

$L = \mathbf{length}(X)$ devuelve la longitud de la dimensión de matriz más grande en X . Para los vectores, la longitud es simplemente el número de elementos. Para matrices con más dimensiones, la longitud es $\mathbf{max}(\mathbf{size}(X))$. La longitud de una matriz vacía es cero.

Sintaxis

```
length(matrix)
```

Ejemplo

```
n=length(numMat);
```


Transpuesta de un vector o matriz

Transpuesta

El operador **transponer** (') se puede usar para *voltear* un vector (y crear un vector de columna a partir de un vector de fila) o una matriz.

Sintaxis

```
vector'  
matrix'
```

Ejemplo

```
colVec=rowVec'  
transposed=matrix'
```

Contenido

- 1 Vectores
- 2 Matrices
- 3 Operaciones Aritméticas**
- 4 Vectores y matrices en las funciones

Operaciones aritméticas

- **Operaciones sobre escalares:** Se aplican a los elementos individuales de las matrices
- **Operaciones matriciales:** Se opera sobre todos los vectores o matrices. Los tamaños de los vectores o matrices deben ser compatibles.

Aviso

¡Cuidado con todas las operaciones que involucran productos, división o exponenciales!

Operaciones sobre elementos individuales

Operaciones elemento por elemento

Para aplicar a elementos individuales productos, divisiones o exponenciales

Operator	Destription
.*	Producto elemento por elemento
./	División elemento por elemento
.^	Exponencial elemento por elemento ($a_i \hat{=} b_i$)

Operaciones

Operaciones

Las operaciones basadas en la multiplicación (multiplicación, división y exponencial) de matrices requieren tamaños de matriz compatibles.

$A * B$ solo si el número de columnas en A es igual a las filas en B .
 $A^2 = A * A$ solo si A es una matriz cuadrada.

Contenido

- 1 Vectores
- 2 Matrices
- 3 Operaciones Aritméticas
- 4 Vectores y matrices en las funciones**

Uso de vectores y matrices en las function

Vectorización

Los vectores y las matrices se pueden utilizar como parámetros en muchas funciones. La función devolverá un vector o matriz del mismo tamaño, correspondiendo cada elemento al resultado de un vector de entrada o elemento de matriz.

Ejemplo: `sin(-pi:0.1:pi)`

Pruebe `x=-pi:0.5:pi; plot(x,sin(x), '*')`

Funciones

Función	Descripción
min	devuelve el elemento más pequeño del vector o matriz
max	devuelve el elemento mayor del vector o matriz
sum	sumatorio. Devuelve la suma de todos los elementos en el vector
cumsum	devuelve la suma acumulativa de los elementos de un vector
prod	productorio. Devuelve el producto de todos los elementos en el vector
cumprod	devuelve la producto acumulativo de los elementos de un vector
diff	para cada uno de los elementos, excepto el primero, muestra la diferencia con su anterior
mean	devuelve el promedio de los valores en el vector
std	devuelve la desviación estándar de los valores en el vector

Vectores and matrices

Oier Lopez de Lacalle y Patxi Ramírez

Languages and Information Systems

2021/2022



GASTEIZKO
INGENIARITZAKO
UNIBERTSITATE ESKOLA
ESCUELA UNIVERSITARIA
DE INGENIERÍA
DE VITORIA-GASTEIZ