

3. Secuencias, matrices de números aleatorios y matrices especiales

- Secuencias
- Vectores y matrices de números aleatorios:
- Matrices especiales: Identidad, Diagonal, Unos, Ceros
 - Matriz Identidad
 - Matriz diagonal
 - Matriz de unos
 - Matriz de ceros

Secuencias

La sintaxis $a:b:c$ genera la secuencia de valores que van de a a c en incrementos de magnitud b .

Si se omite el valor de b se genera una secuencia de valores de a a c de uno en uno. En general una secuencia de valores es considerada por Matlab como un vector.

```
>> U=0:10
```

```
U =
```

```
0 1 2 3 4 5 6 7 8 9 10
```

```
>> V=0:2:10
```

```
V =
```

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

```
>> W=0:0.1:0.5
```

```
W =
```

```
0.00000 0.10000 0.20000 0.30000 0.40000 0.50000
```

```
>> X=12:-3:-12
```

```
X =
```

```
12 9 6 3 0 -3 -6 -9 -12
```

Vectores y matrices de números aleatorios:

La función `rand()` crea vectores o matrices de números aleatorios distribuidos uniformemente entre 0 y 1; si se especifica `rand(n)` se crea una matriz de dimensión $n \times n$; si se especifica `rand(m,n)` se genera una matriz de dimensión $m \times n$:

```
>> rand(1)
```

```
ans = 0.38783
```

```
>> rand(2)
```

```
ans =
```

```
0.13337 0.95887  
0.93870 0.22270
```

```
>> rand(1,4)
```

```
ans =
```

```
0.51418 0.57228 0.94543 0.73161
```

```
>> rand(2,3)
```

```
ans =
```

```
0.482729 0.765492 0.448806  
0.075225 0.872439 0.782452
```

La función `randi()` es similar, pero genera números enteros:

- `randi(imax)`: genera al azar un número entero entre 1 e `imax`.
- `randi(imax, n)`: genera una matriz $n \times n$ de números enteros elegidos al azar entre 1 e `imax`.
- `randi(imax, m, n)`: genera una matriz $m \times n$ de números enteros elegidos al azar entre 1 e `imax`.
- `randi([imin imax], ...)`: Igual que los comandos anteriores, pero eligiendo los números aleatorios entre `imin` e `imax`. Donde están los puntos suspensivos podemos poner la dimensión del vector o matriz a generar.

Ejemplos:

```
>> randi(7)
```

```
ans = 7
```

```
>> randi(7,2,2)
```

```
ans =
```

```
 4  4  
 7  3
```

```
>> randi([3,9],1,10)
```

```
ans =
```

```
 6  9  9  5  8  5  6  3  3  5
```

Matrices especiales: Identidad, Diagonal, Unos, Ceros

Matriz Identidad

La matriz identidad de dimensión n se define en Matlab como `eye(n)`. La razón es que en inglés `eye` y la letra **I** (que es el símbolo utilizado habitualmente para la matriz identidad) suenan igual:

```
>> eye(3)
```

```
ans =  
  
Diagonal Matrix  
  
 1  0  0  
 0  1  0  
 0  0  1
```

Matriz diagonal

En Matlab una matriz diagonal se construye fácilmente como:

```
>> diag([1 2 3])
```

```
ans =  
  
Diagonal Matrix  
  
 1  0  0  
 0  2  0  
 0  0  3
```

Se pueden generar matrices con los términos no nulos en alguna diagonal distinta de la principal. Por ejemplo, la siguiente sintaxis genera una matriz cuadrada con los valores (1,2,3) en la quinta diagonal (**NOTA:** téngase en cuenta que para Matlab la diagonal principal es la diagonal 0):

```
>> diag([1 2 3],5)
```

```
ans =  
  
 0  0  0  0  0  1  0  0  
 0  0  0  0  0  0  2  0  
 0  0  0  0  0  0  0  3
```

```
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
```

Matriz de unos

Podemos construir una matriz de unos de dimensión $m \times n$ como `ones(m,n)`:

```
>> ones(3,2)
```

```
ans =
```

```
1 1
1 1
1 1
```

Si queremos que la matriz de unos sea cuadrada de orden n basta con `ones(n)`:

```
>> ones(3)
```

```
ans =
```

```
1 1 1
1 1 1
1 1 1
```

Matriz de ceros

Asimismo una matriz de ceros de dimensión $m \times n$ se define como `zeros(m,n)`:

```
>> zeros(2,3)
```

```
ans =
```

```
0 0 0
0 0 0
```

Si queremos una matriz cuadrada de ceros de orden n basta con `zeros(n)`:

```
>> zeros(3)
```

```
ans =
```

```
0 0 0  
0 0 0  
0 0 0
```