

# Matrici sparse

Una matrice si dice **sparsa** se il numero di elementi non nulli è piccolo rispetto al numero totale degli elementi della matrice.

```
load west0479; % carica il contenuto del file
% west0479.mat in una matrice in formato sparso
W=west0479; clear west0479
whos
```

Name	Size	Bytes	Class	Attributes
W	479x479	34032	double	<b>sparse</b>

```
spy(W) % per vedere il pattern della matrice
A=full(W); % genera una mat. in formato full
% contenente gli stessi elementi di W
whos
```

Name	Size	Bytes	Class	Attributes
A	479x479	1835528	double	
W	479x479	34032	double	<b>sparse</b>

## Vantaggi della forma sparsa

- 1 Risparmio di memoria
- 2 Risparmio di cpu-time nel calcolo di prodotti matrice-vettore

**Esempio.** Costruire un vettore colonna  $x$  di dimensione  $n=\text{length}(A)$  di numeri casuali, calcolare 1000 volte i prodotti  $y=A*x$  e  $z=W*x$  e prendere i tempi di calcolo

```
tic
for i=1:1000
    y=A*x;
end
toc
tic
for i=1:1000
    z=W*x;
end
toc
```

```
Elapsed time is 0.014718 seconds.
```

```
Elapsed time is 0.002499 seconds.
```

## Generazione di una matrice in formato sparse

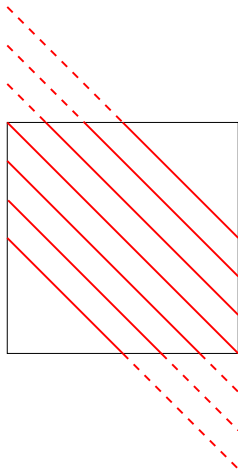
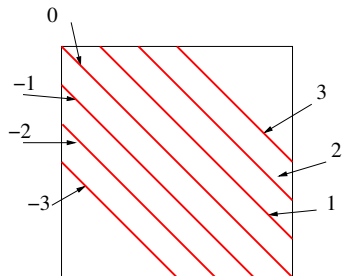
```
A=sparse(n); % genera una matrice A (n x n) nulla  
            % in formato sparse
```

**Esempio:** definire una matrice i cui elementi sulla diagonale principale sono uguali a 3, quelli in prima riga e in prima colonna (ad eccezione di  $a_{11}$ ) siano uguali a 1, mentre i restanti siano nulli.

```
n=20;  
A=3*speye(n);  
A(1,:)=1; % prima riga di A <--- tutti 1  
A(:,1)=1; % prima colonna di A <--- tutti 1  
spy(A) % visualizzo il pattern  
A % visualizzo A a video
```

# Ordinamento delle diagonali

Le diagonali di una matrice sono ordinate come nella figura qui sotto a sinistra:



## Comandi veloci per generare una matrice sparsa definita per diagonali

Sia  $A$  una matrice quadrata di dimensione 20 di elementi:

2 sulla diagonale principale,

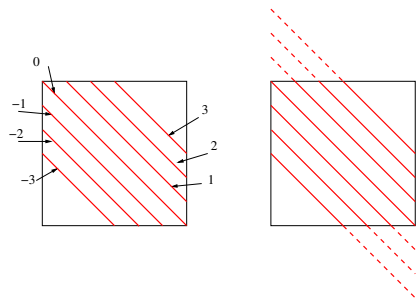
1 sulla diagonale di indice -4,

-3 sulla diagonale di indice 10.

I comandi sono:

```
e=ones(20,1); % e deve essere vettore colonna
```

```
A=spdiags([e,2*e,-3*e],[-4,0,10],20,20);
```



## Conversione sparse $\leftrightarrow$ full

È possibile convertire una matrice da formato sparso a classico e viceversa:

**sparse** converte **double array** in **double (sparse) array**

**full** converte **double (sparse) array** in **double array**

Se A1 contiene un array di tipo sparso e A un array di tipo classico (full).

```
AF1=full(A1); % copia A1 (sparse) in AF1 (full)
```

```
AS=sparse(A); % copia A (full) in AS (sparse)
```

**Operazioni con matrici sparse** Le matrici sparse sono trattate in matlab come le matrici piene, si possono fare le operazioni elementari, applicare lu, meg, \,...

Alle matrici sparse non si possono applicare i seguenti comandi:

```
cond(A)
```

```
norm(A)
```

```
eig(A) % fino a MATLAB R2021
```