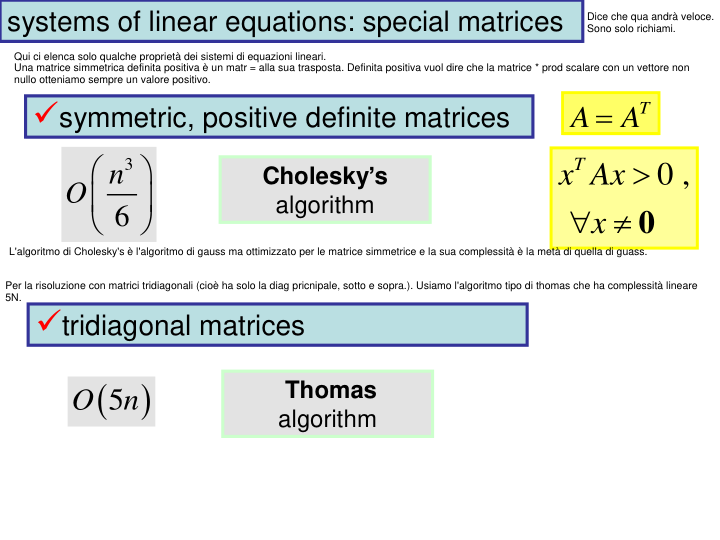
Lezione 2A

Qui ci elenca solo qualche proprietà dei sistemi di equazioni lineari.  
Una matrice simmetrica definita positiva è un matr = alla sua trasposta. Definita positiva vuol dire che la matrice \* prod scalare con un vettore non nullo otteniamo sempre un valore positivo.

Dice che qua andrà veloce.  
Sono solo richiami.

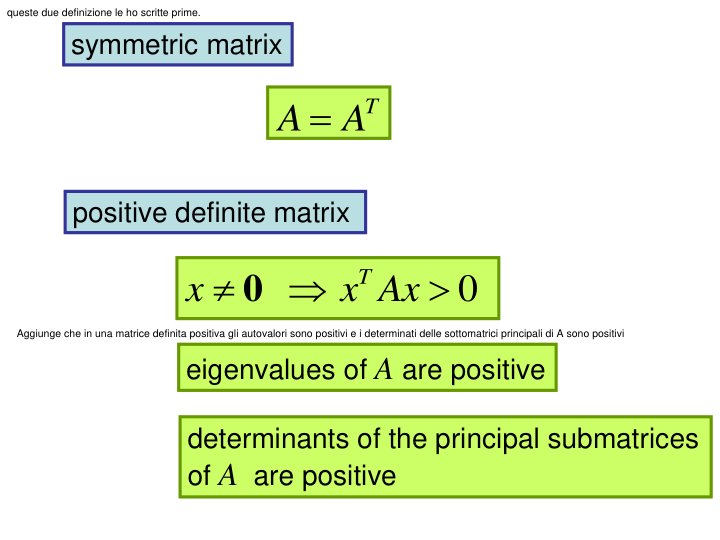
L'algoritmo di Cholesky's è l'algoritmo di gauss ma ottimizzato per le matrice simmetrice e la sua complessità è la metà di quella di guass.

Per la risoluzione con matrici tridiagonali (cioè ha solo la diag pricnipale, sotto e sopra.). Usiamo l'algoritmo tipo di thomas che ha complessità lineare 5N.

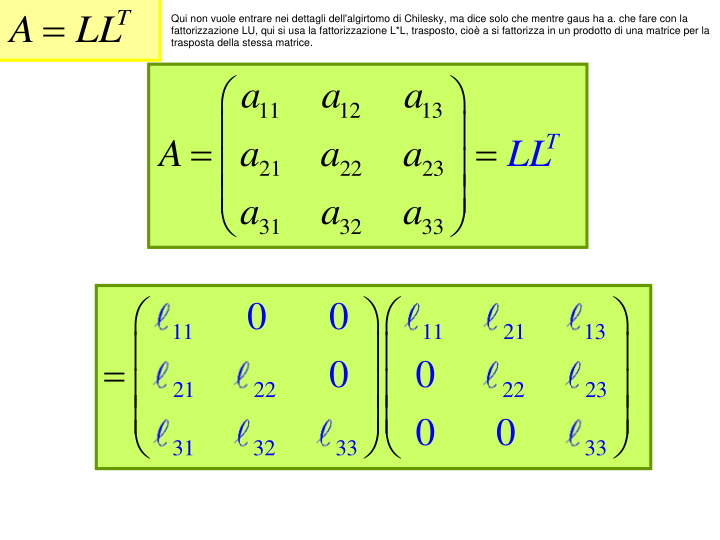


queste due definizione le ho scritte prime.

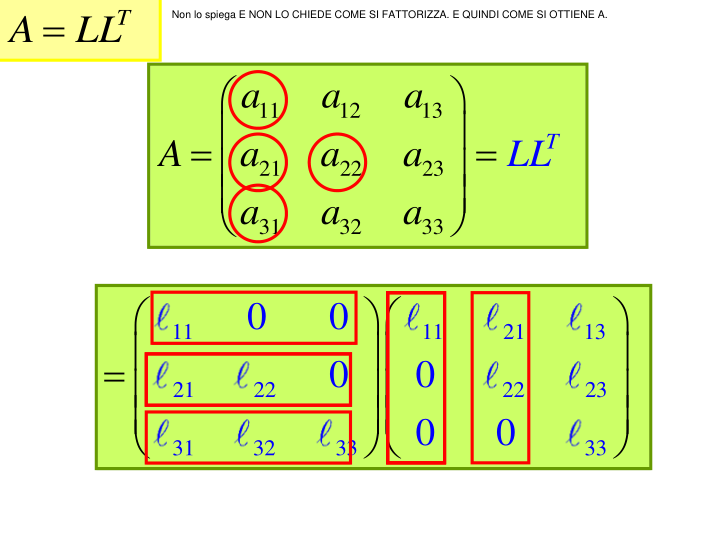
Aggiunge che in una matrice definita positiva gli autovalori sono positivi e i determinati delle sottomatrici principali di A sono positivi



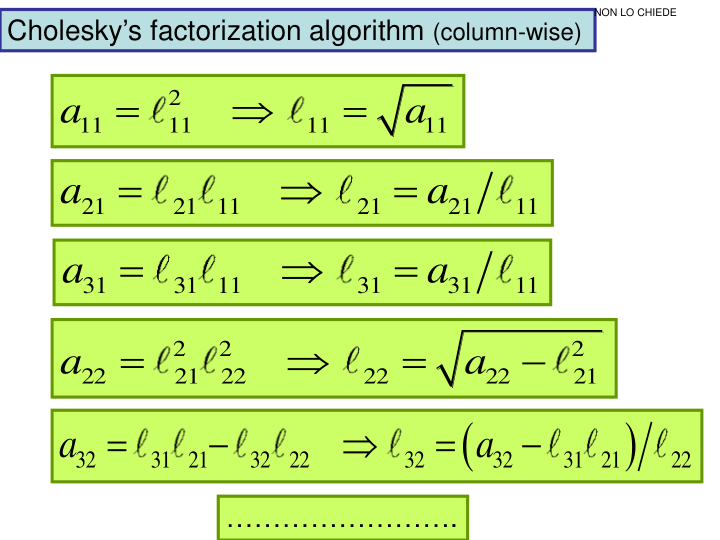
Qui non vuole entrare nei dettagli dell'algirtomo di Chilesky, ma dice solo che mentre gaus ha a. che fare con la fattorizzazione LU, qui si usa la fattorizzazione L\*L, trasposto, cioè a si fattorizza in un prodotto di una matrice per la trasposta della stessa matrice.



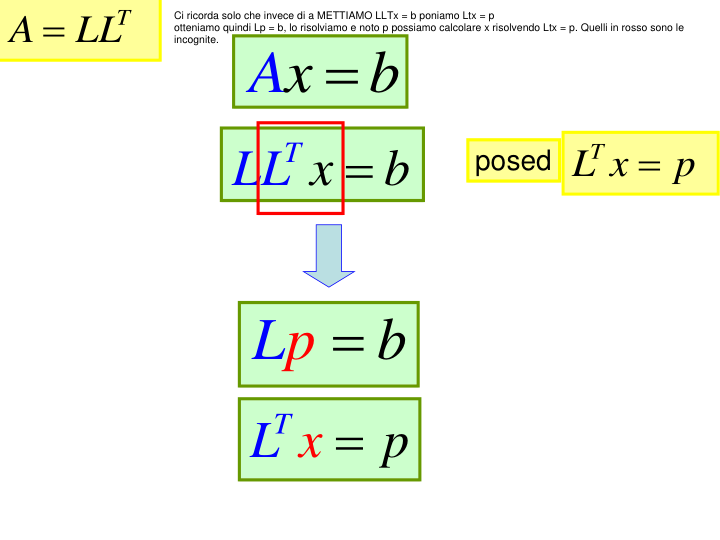
Non lo spiega E NON LO CHIEDE COME SI FATTORIZZA. E QUINDI COME SI OTTIENE A.



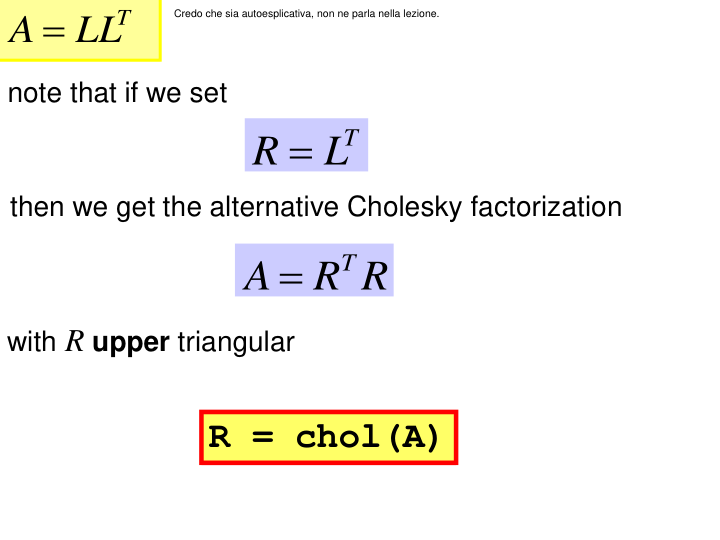
NON LO CHIEDE



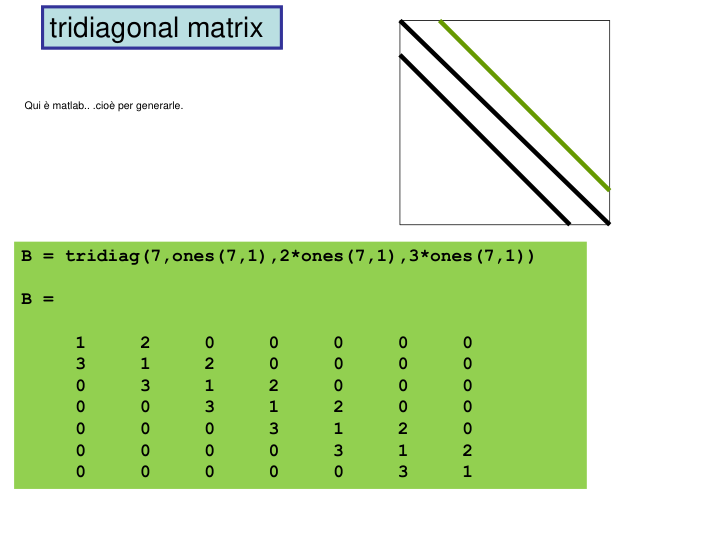
Ci ricorda solo che invece di a METTIAMO LLTx = b poniamo Ltx = p  
otteniamo quindi Lp = b, lo risolviamo e noto p possiamo calcolare x risolvendo Ltx = p. Quelli in rosso sono le incognite.



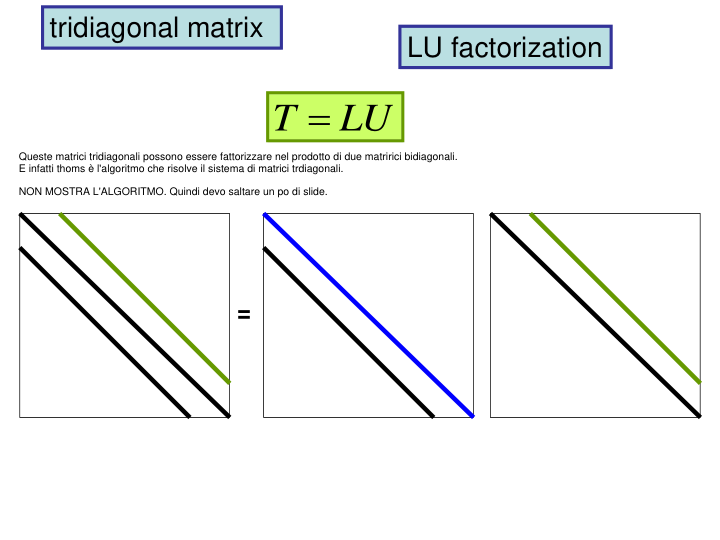
Credo che sia autoesplicativa, non ne parla nella lezione.

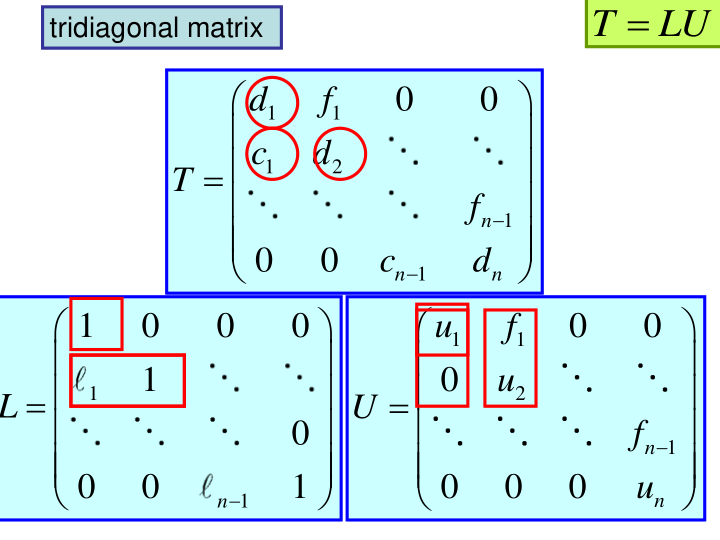


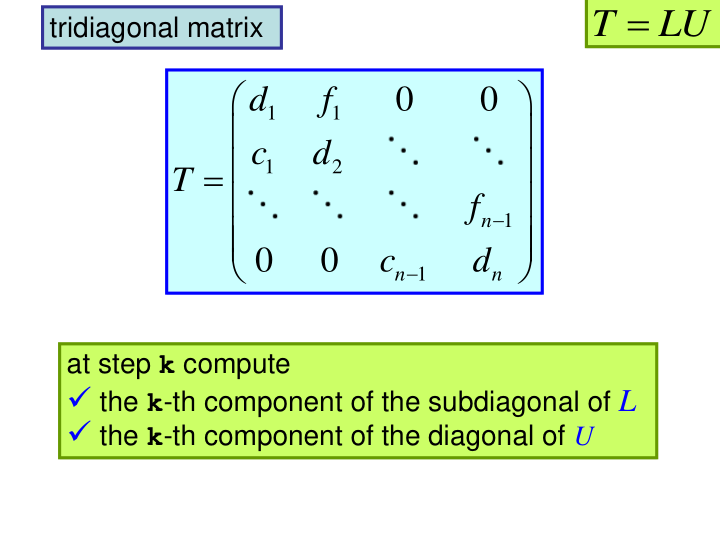
Qui è matlab.. .cioè per generarle.

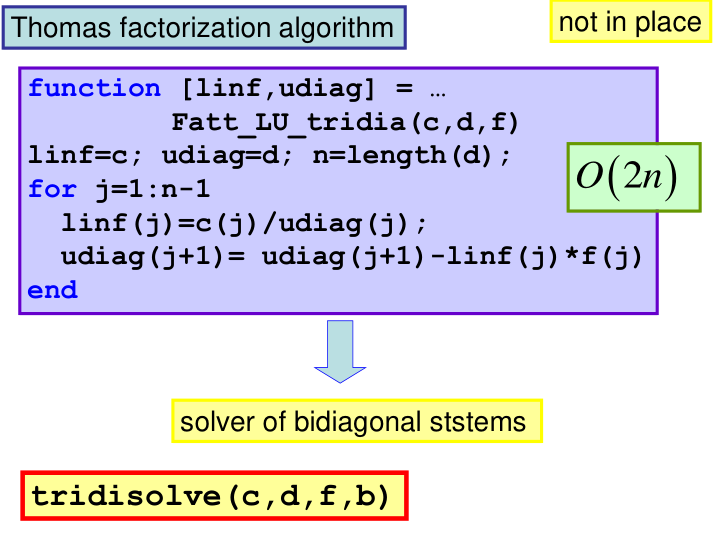


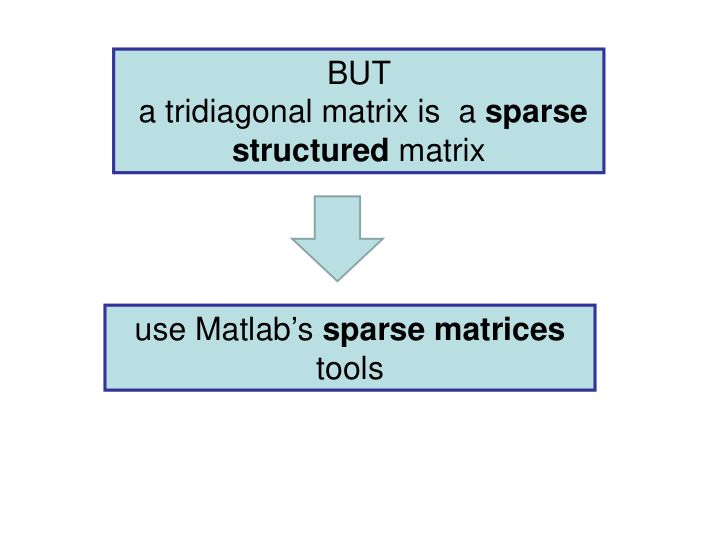
Queste matrici tridiagonali possono essere fattorizzare nel prodotto di due matririci bidiagonali.  
E infatti thoms è l'algoritmo che risolve il sistema di matrici trdiagonali.  
  
NON MOSTRA L'ALGORITMO. Quindi devo saltare un po di slide.



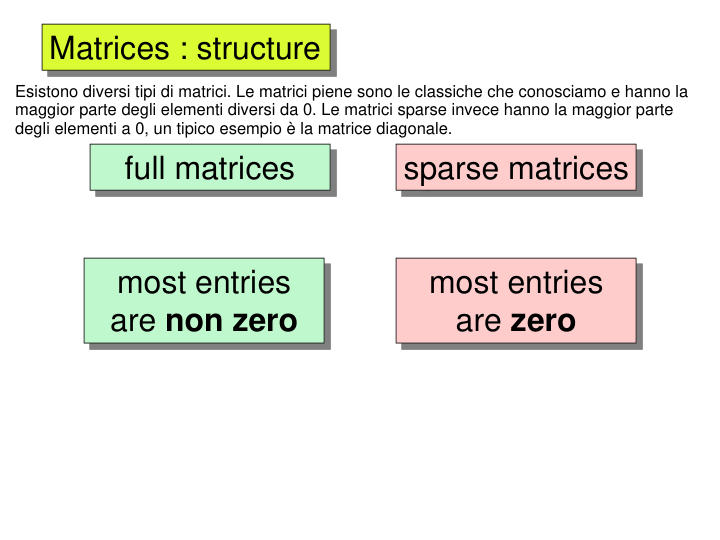






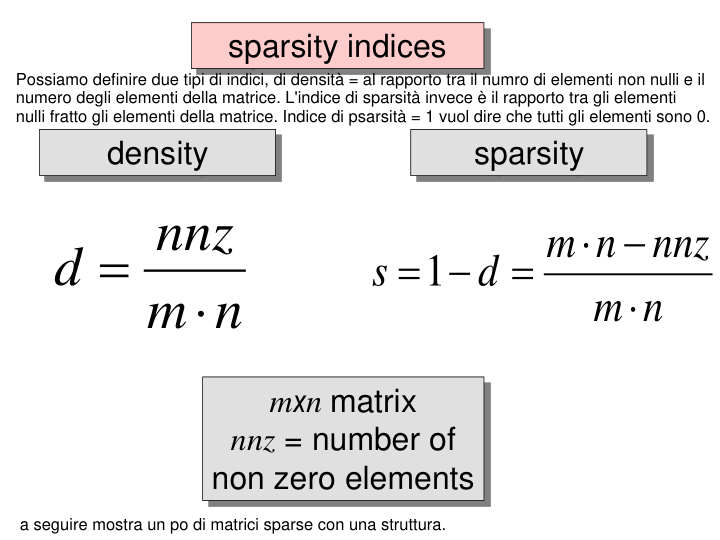


Esistono diversi tipi di matrici. Le matrici piene sono le classiche che conosciamo e hanno la maggior parte degli elementi diversi da 0. Le matrici sparse invece hanno la maggior parte degli elementi a 0, un tipico esempio è la matrice diagonale.



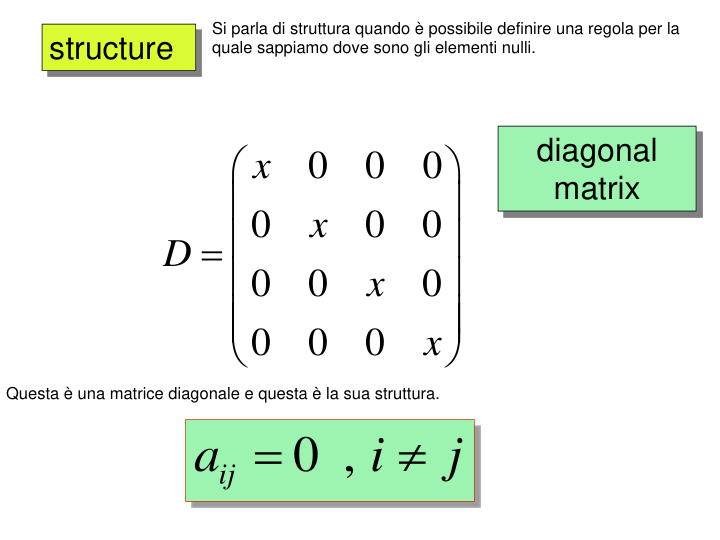
Possiamo definire due tipi di indici, di densità = al rapporto tra il numro di elementi non nulli e il numero degli elementi della matrice. L'indice di sparsità invece è il rapporto tra gli elementi nulli fratto gli elementi della matrice. Indice di psarsità = 1 vuol dire che tutti gli elementi sono 0.

a seguire mostra un po di matrici sparse con una struttura.

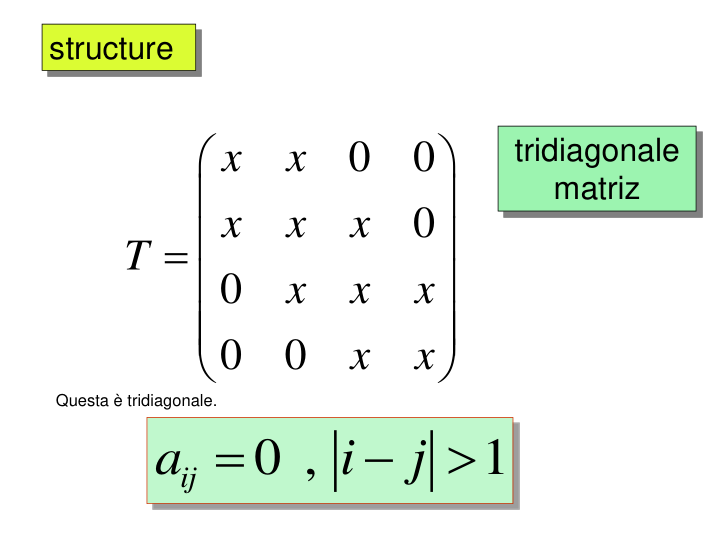


Questa è una matrice diagonale e questa è la sua struttura.

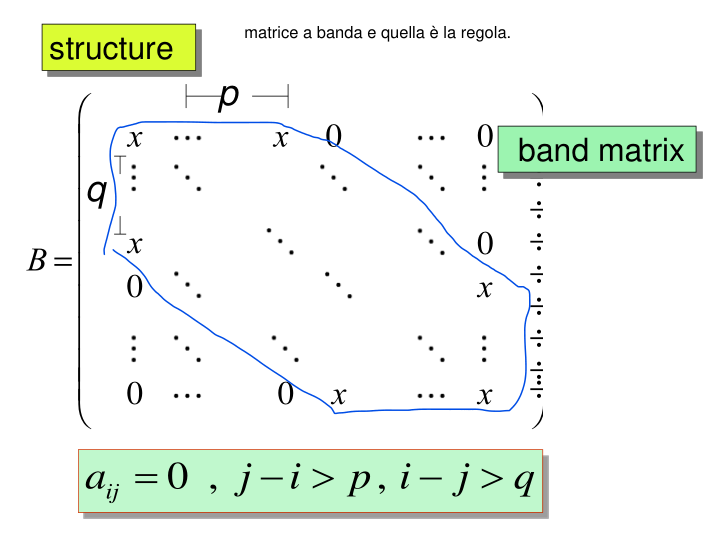
Si parla di struttura quando è possibile definire una regola per la quale sappiamo dove sono gli elementi nulli.



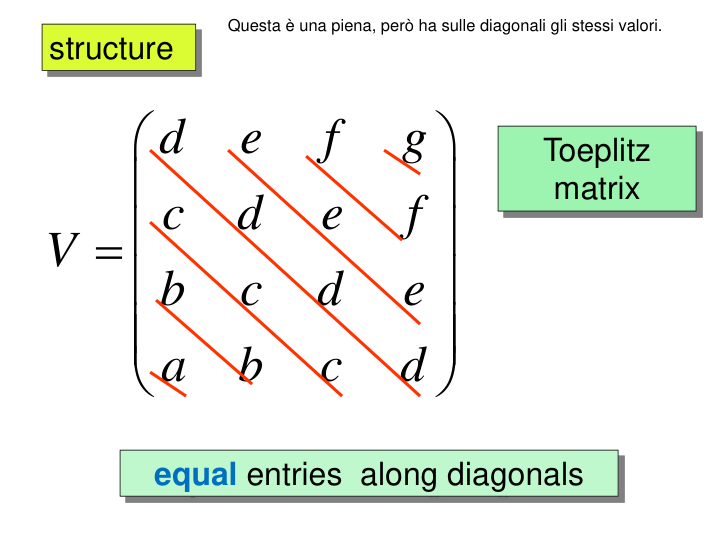
Questa è tridiagonale.



matrice a banda e quella è la regola.

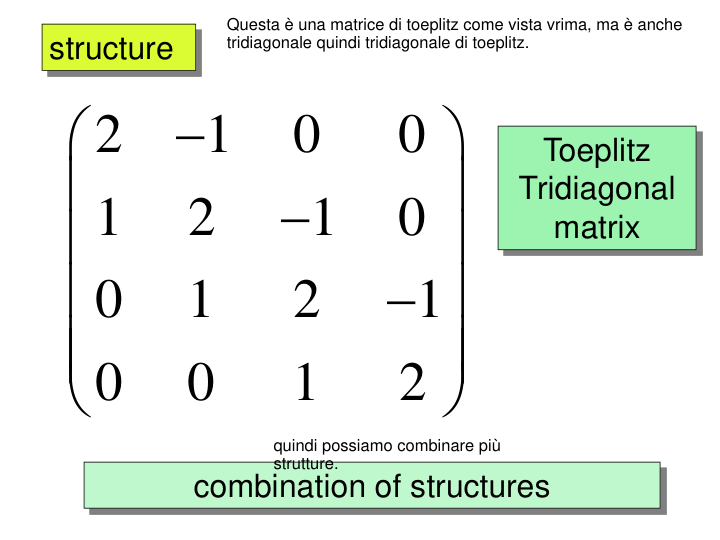


Questa è una piena, però ha sulle diagonali gli stessi valori.



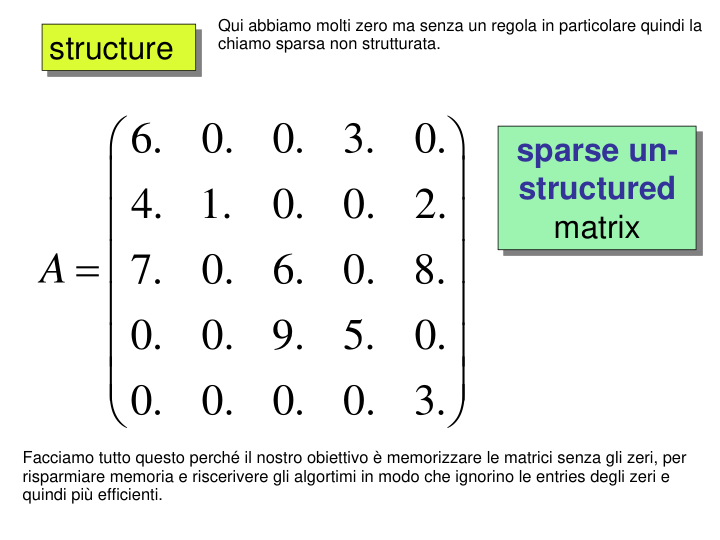
Questa è una matrice di toeplitz come vista vrima, ma è anche tridiagonale quindi tridiagonale di toeplitz.

quindi possiamo combinare più strutture.

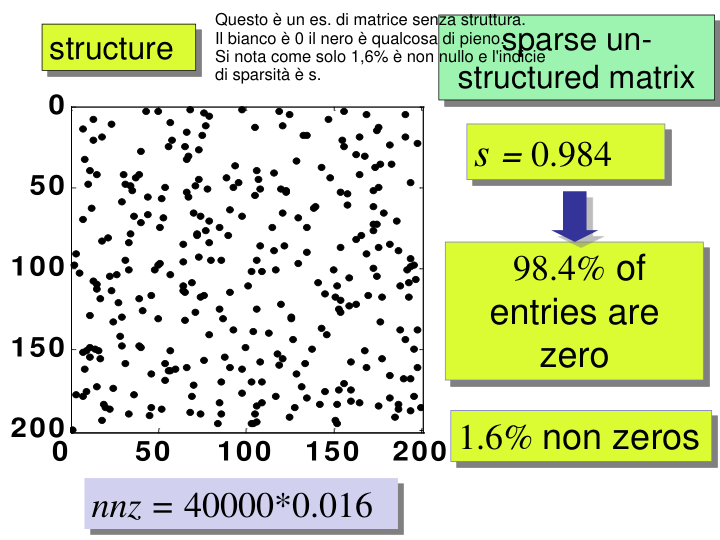


Qui abbiamo molti zero ma senza un regola in particolare quindi la chiamo sparsa non strutturata.

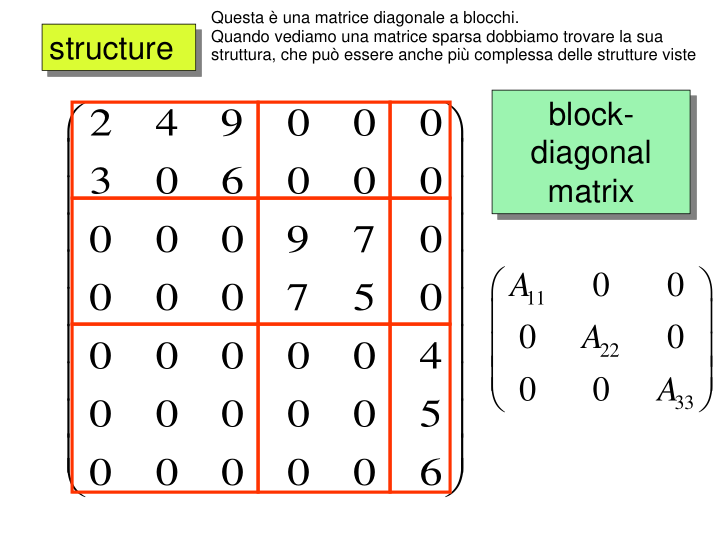
Facciamo tutto questo perché il nostro obiettivo è memorizzare le matrici senza gli zeri, per risparmiare memoria e riscerivere gli algortimi in modo che ignorino le entries degli zeri e quindi più efficienti.



Questo è un es. di matrice senza struttura.  
Il bianco è 0 il nero è qualcosa di pieno.  
Si nota come solo 1,6% è non nullo e l'indicie  
di sparsità è s.

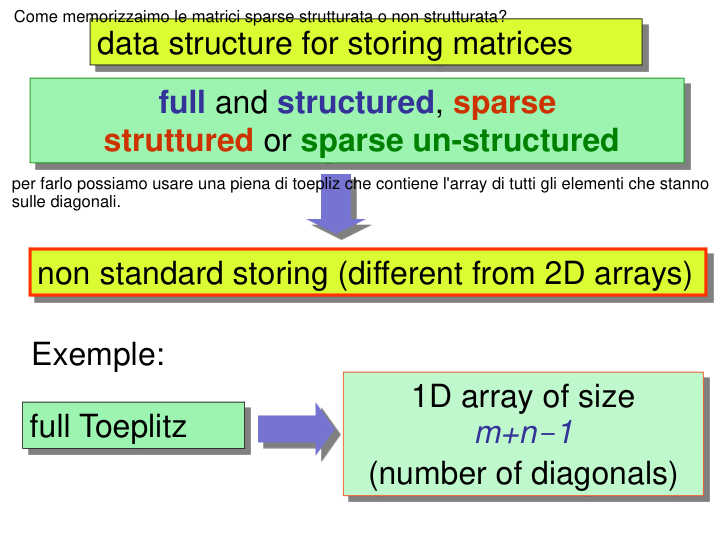


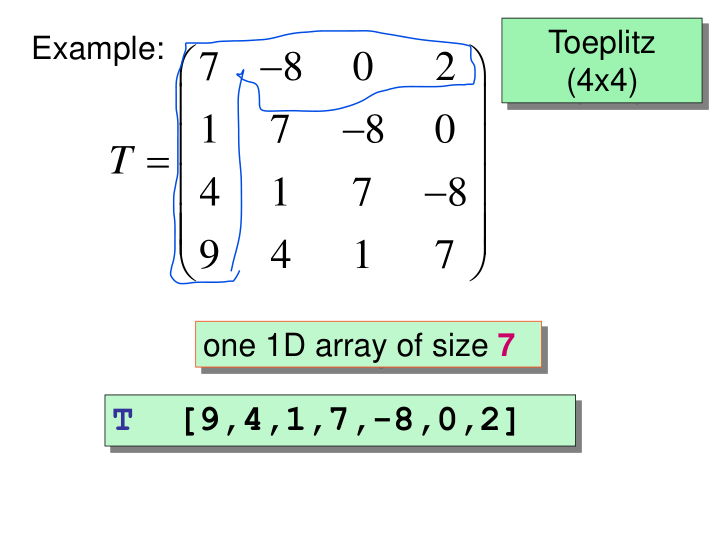
Questa è una matrice diagonale a blocchi.  
Quando vediamo una matrice sparsa dobbiamo trovare la sua struttura, che può essere anche più complessa delle strutture viste



Come memorizzaimo le matrici sparse strutturata o non strutturata?

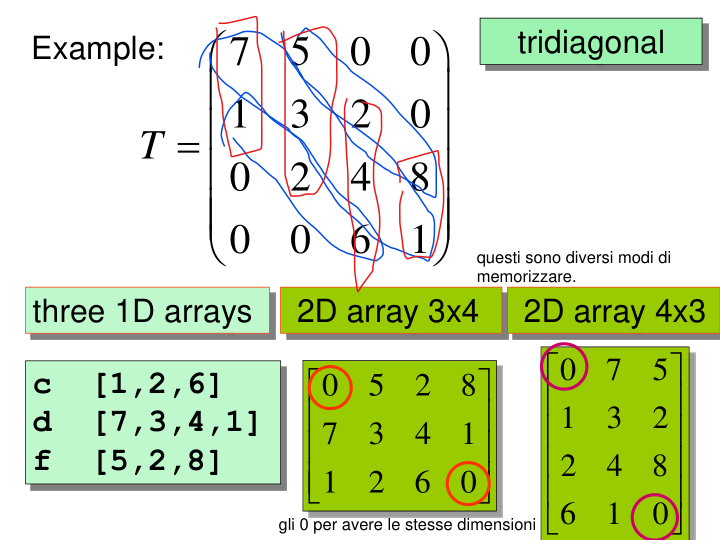
per farlo possiamo usare una piena di toepliz che contiene l'array di tutti gli elementi che stanno sulle diagonali.





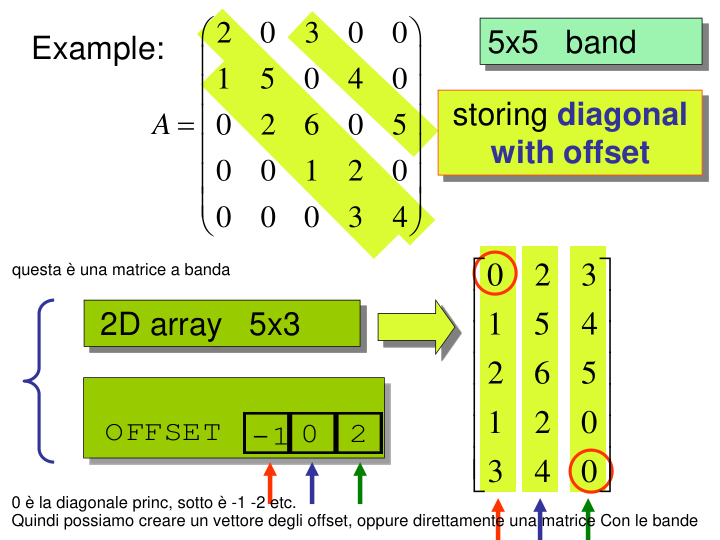
gli 0 per avere le stesse dimensioni

questi sono diversi modi di memorizzare.



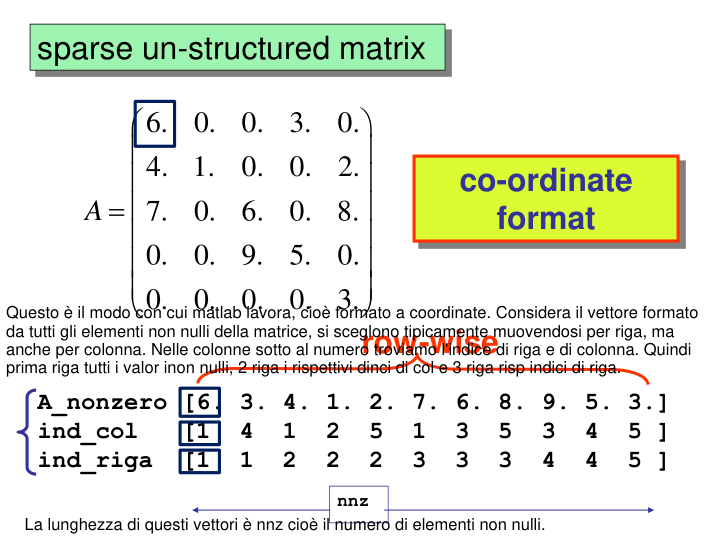
questa è una matrice a banda

0 è la diagonale princ, sotto è -1 -2 etc.   
Quindi possiamo creare un vettore degli offset, oppure direttamente una matrice Con le bande

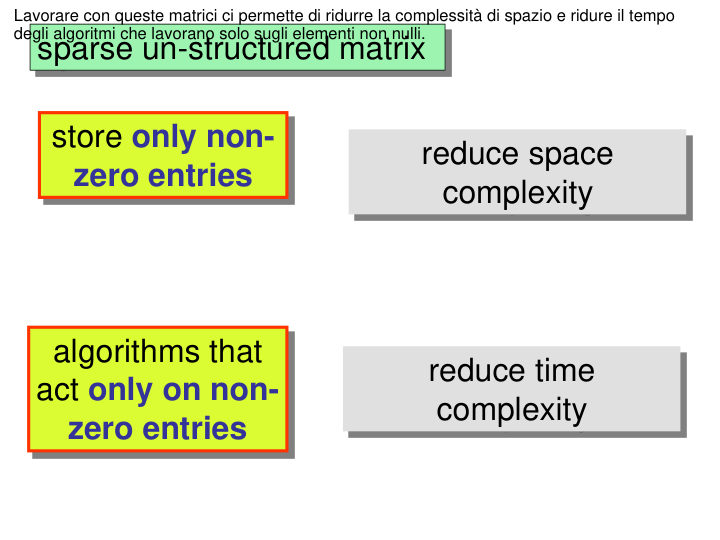


Questo è il modo con cui matlab lavora, cioè formato a coordinate. Considera il vettore formato da tutti gli elementi non nulli della matrice, si sceglono tipicamente muovendosi per riga, ma anche per colonna. Nelle colonne sotto al numero troviamo l'indice di riga e di colonna. Quindi prima riga tutti i valor inon nulli, 2 riga i rispettivi dinci di col e 3 riga risp indici di riga.

La lunghezza di questi vettori è nnz cioè il numero di elementi non nulli.



Lavorare con queste matrici ci permette di ridurre la complessità di spazio e ridure il tempo degli algoritmi che lavorano solo sugli elementi non nulli.



Matlab ridefinisce poi gli operatori per lavorare su matrici sparse ed ha function per creare matrici sparse, visualizzare e estrarre informazioni.

RIPRENDERE LA PROSSIMA LEZIONE DALLA REGISTRAZIONE SUCCESSIVA ALLA 01/10

