# Laboratorio di Calcolo Numerico: Algebra lineare numerica

Giacomo Elefante

Laboratorio di calcolo numerico
23/05/23

# Scopo di oggi

Analizzare algoritmi per l'algebra lineare numerica:

- Norme vettoriali e matriciali
- Risoluzione di sistemi lineari
- Condizionamento
- Decomposizione LU
- Sistemi sovradeterminati

Dato un vettore $x \in \mathbb{R}^n$, le norme vettoriali più usate sono

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|, \qquad \|x\|_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2}, \qquad \|x\|_\infty = \max_{i=1,\dots,n} |x_i|$$

Mentre nel caso di una matrice $A \in \mathbb{R}^{n\times n}$

$$\|A\|_1 = \max_{j=1,\dots,n} \sum_{i=1}^{n} |a_{i,j}|, \qquad \|A\|_2 = \sqrt{\max_{i=1,\dots,n} |\lambda_i(A^t A)|},$$

$$\|A\|_\infty = \max_{i=1,\dots,n} \sum_{j=1}^{n} |a_{i,j}|$$

dove $\lambda_i(B)$ sono gli autovalori della matrice $B$

# Esercizio

In Matlab tali norme sono implementabili facilmente con il comando `norm`, ad esempio `norm(x,1)` restituirà la norma 1 del vettore (o matrice) `x`, per la norma infinito basta sostituire 1 con `inf`/`Inf`/`'Inf'`/`'inf'`.

## Esercizio

Si crei una function `norme_varie` che prende in input un vettore o una matrice `x` e un parametro `p` che può prendere solo (altrimenti viene segnalato un errore) i valori 1, 2, 'Inf'; La function inoltre resituisce come output il valore `s` corrispondente alla norma di `x`
Dato quindi `x`, la function dovrà distinguere il caso esso sia un vettore o una matrice.
Si costruisca la function **senza** usare il comando `norm` di Matlab ma attraverso le definizioni.

Data una matrice $A \in \mathbb{R}^{n \times n}$ e un vettore $b \in \mathbb{R}^n$, in Matlab è possibile ricavare il vettore $x \in \mathbb{R}^n$ soluzione del sistema lineare

$$Ax = b,$$

attraverso il comando backslash \, ovvero indicando

```
x = A\b;
```

Il condizionamento di una matrice $\kappa_p(A)$ associato ad una certa norma $\|\cdot\|_p$ e definito come

$$\kappa_p(A) = \|A\|_p \|A^{-1}\|_p$$

e, similmente al comando `norm`, calcolabile in Matlab con il comando `cond`.

# Esercizio

## Esercizio

Ricordando la stima, vista a lezione, per l'errore relativo della soluzione di un sistema lineare perturbato $\widetilde{A}\widetilde{x} = \widetilde{b}$

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{\kappa(A)}{1 - \kappa(A)\frac{\|\delta A\|}{\|A\|}} \left( \frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|b\|} \right),$$

valida se $\kappa(A)\frac{\|\delta A\|}{\|A\|} < 1$ e dove $\widetilde{A} = A + \delta A$, $\widetilde{x} = x + \delta x$, $\widetilde{b} = b + \delta b$, si crei una funzione `sistema_perturbato` senza output e che prenda come input i valori `A`, `b`, `Aap`, `bap` ovvero le matrici e vettori $A, b, \widetilde{A}, \widetilde{b}$.
Internamente alla funzione si calcoli le soluzioni `x`, `xap` relative, rispettivamente, al sistema lineare e al sistema lineare perturbato attraverso il comando backslash.
In seguito, dopo aver eseguito un controllo per $\kappa(A)\frac{\|\delta A\|}{\|A\|} < 1$ (e definito un comando di errore nel caso contrario), si controlli la stima calcolando il valore a sinistra e a destra della disuguaglianza, stampandoli a schermo in formato esponenziale con 1 cifra prima della virgola e due dopo.

Data una matrice quadrata $A \in \mathbb{R}^{n \times n}$, per decomposizione $LU$ si riferisce alla fattorizzazione della matrice $A$ in due matrici $L, U \in \mathbb{R}^{n \times n}$, la prima triangolare inferiore, la seconda triangolare superiore, tali che

$$A = LU.$$

Tale fattorizzazione però non è sempre possibile da fare, come ad esempio nel caso in cui $a_{11} = 0$ e $A$ è una matrice non singolare. Infatti, il primo elemento di $A$ uguale a 0 implica che o $l_{11} = 0$ o $u_{11} = 0$, e, essendo le matrici triangolari, si ha che o $L$ o $U$ risulta essere una matrice singolare, ma tale fatto è impossibile.
Per cui, è preferibile la fattorizzazione $LU$ con pivoting parziale

$$PA = LU.$$

Tale decomposizione può essere effettuata su Matlab attraverso il comando `lu`.
Nel caso si chiedano solo due output, essi saranno due matrici, una la matrice $U$ e l'altra la matrice $L$ permutata, ovvero $P^t L$

# Esercizio

La decomposizione può essere usata per trovare la soluzione di un sistema lineare $Ax = b$, infatti, considerando la decomposizione, è possibile andare a risolvere direttamente due sistemi triangolari.

$$\begin{cases} Ly = Pb \\ Ux = y \end{cases}$$

## Esercizio

Si crei la function `lu_solver` che prende in input una matrice $A$ e un vettore colonna $b$ e ricava la soluzione del sistema, facendo la decomposizione $PA = LU$ della matrice.
Si testi in seguito la funzione sulla matrice

$$\mathtt{A} = \mathtt{magic}(9), \qquad b = \Big(910, 1034, 1113, 1264, 1172, 981, 1060, 941, 750,\Big)^t$$

Nel caso di sistemi sovradeterminati, ovvero $Ax = b$ con $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ e $x \in \mathbb{R}^n$, con $m > n$. In generale non esiste una soluzione $x$ tale che $Ax = b$, si cerca una soluzione tale che il residuo $\|Ax - b\|_2$ sia minima.
Una soluzione del problema ai minimi quadrati può essere data dalla soluzione del problema alle equazioni normali

$$A^t Ax = A^t b$$

# Esercizio

## Esercizio

Si crei uno script dove si cerca il polinomio di grado $n = 1$ (la retta di regressione) che meglio approssimi dei dati perturbati della retta $y = 2x + 0.2$, ottenuti aggiungendo ai dati dei valori random ottenuti con la funzione Matlab `rand`, ovvero

`y = 2 x + 0.2 + 0.1*rand(size(x))`

si prendano come ascisse, un vettore `x` di 20 punti equidistanti in $[0, 1]$.

Per generare la matrice di Vandermonde relativa ai nodi `x`, si usi il comando

`V = fliplr(vander(x));`

Tale comando genera la matrice (quadrata) di Vandermonde dove ogni elemento è $\mathtt{V}_{i,j} = \mathtt{x}_i^j$. Volendo il polinomio di grado 1 che meglio approssima i dati si definisca la matrice `A` del problema ai minimi quadrati, prendendo solo le prime due colonne della matrice `V`.

Si trovino quindi i coefficienti della retta, ovvero la soluzione (attraverso le equazioni normali) del problema ai minimi quadrati generato , e successivamente si faccia il plot sovrapponendo la retta di regressione in blu e i punti $(\mathtt{x}, \mathtt{y})$ in pallini verdi con bordo nero.

Il condizionamento della matrice $A$ (e anche di $A^tA$ nel caso del problema ai minimi quadrati) può portare a non trovare la soluzione corretta del poblema.

## Esercizio

Si ripeta l'esercizio con `lu_solver` ma usando la matrice $\mathtt{A} = \mathtt{magic}(10)$ e come $b$ il risultato di $\mathtt{b} = \mathtt{A} * [1:10]'$, (da cui si ha ovviamente che la soluzione esatta è $[1:10]'$)

Per migliorare si può utilizzare la decomposizione $QR$ della matrice. Infatti data $A \in \mathbb{R}^{m\times n}$ tale che $\mathtt{rank}(A) = n$, allora esistono $Q \in \mathbb{R}^{m\times n}$ (ortogonale) e $R \in \mathbb{R}^{n\times n}$ (triangolare superiore non singolare) tali che

$$A = QR.$$

Da questo si può dimostrare che il sistema alle equazioni normali può essere trasformato (attraverso la decomposizione QR di $A^tA$) nel sistema

$$Rx = Q^tb$$

che è risolvibile facilmente con il meg ma in particolare si ha che

$$\kappa(A^tA) = (\kappa(A))^2 >> \kappa(R)$$

Attenzione che utilizzando il comando per la decomposizione QR di Matlab `qr` su $A \in \mathbb{R}^{m\times n}$ viene decomposta in due matrici $Q, R$ la prima ortogonale e la seconda triangolare superiore di dimensioni $Q \in \mathbb{R}^{m\times m}, R \in \mathbb{R}^{m\times n}$, che possiamo descrivere come

$$Q = [Q_1 Q_2]$$

$Q_1 \in \mathbb{R}^{m\times n}$ e $Q_2 \in \mathbb{R}^{m\times(m-n)}$ e

$$R = \begin{bmatrix} R1 \\ 0 \end{bmatrix}$$

con $R_1 \in \mathbb{R}^{n\times n}$ e $0 \in \mathbb{R}^{(m-n)\times n}$, da cui le matrici che servono per risolvere il sistema lineare sono $Q_1$ e $R_1$.

# Esercizio

## Esercizio

Data la funzione $f(x) = sin(2x) - x^2$, si costruisca un vettore $\mathtt{x}$ di $\mathtt{n} = 100$ punti in $[-5, 5]$, dove valutare la funzione in un vettore $\mathtt{y}$, a cui possiamo aggiungere un rumore aggiungendo $0.5 * \mathtt{rand(size(y))}$. In seguito si definisca un grado $\mathtt{d} = 8$ e si vada a cercare il polinomio di grado $\mathtt{d}$ che meglio approssima ai minimi quadrati i valori $\mathtt{y}$ in $\mathtt{x}$, ovvero che i suoi coefficienti $\mathtt{a} = [\mathtt{a}_0, \ldots, \mathtt{a}_\mathtt{d}]$ risolvano il sistema

$$Va = y$$

con $(V)_{i,j} = \mathtt{x}_i^j$.
Si esegua l'esercizio sia andando a risolverlo con le equazioni normali e sia con la decomposizione QR.
Si stampino a schermo il condizionamento della matrice $A^tA$ e la matrice $R$.
E infine si facciano due grafici, uno sovrapponendo la funzione $f$ al polinomio approssimante trovato attraverso le equazioni normali e un secondo sovrapponendo la funzione $f$ al polinomio approssimante trovato attraverso la decomposizione QR.

# Esercizio (per casa)

## Esercizio

Sapendo che la nostra funzione ha come dati il vettore y nella prossima slide in 40 punti equidistanti tra $[-\pi, \pi]$. Si calcolino i coefficienti del polinomio approssimante ai minimi quadrati e si cerchi il valore di tale polinomio in $\mathtt{x0} = \pi/4$.
Si stampi a schermo tale valore in formato esponenziale con 1 cifra prime della virgola e 10 dopo.
Ora si ripeta la medesima cosa prendendo come dati solo gli elementi di indice pari del vettore y.
Sapendo che i valori derivano dalla funzione $f(x) = sin(x) - cos(3x)$ a cui è stato aggiunto un rumore, si calcoli gli errori assoluti che fanno i due polinomi in x0 e si stampino anch'essi a schermo

# Esercizio (per casa)

## Esercizio

y= [1.081472368639318; 0.815623938503011; 0.264095434559032;
-0.252848906174545; -0.892111226657974; -1.459959155014975;
-1.766075861433004; -1.819704100115936; -1.613278176259043;
-1.250824907620662; -0.862891009748491; -0.314666627135050;
0.046156477792090; 0.182512161087846; 0.190879110714435;
-0.080869277646918; -0.371753017609851; -0.654997944383704;
-0.908605679502703; -0.955459143503488; -0.824901178793667;
-0.505623916026125; 0.122274653404417; 0.748401831158879;
1.299060920457728; 1.735835000538698; 1.940338650596930;
1.859694970292042; 1.613403187791179; 1.136844347208049;
0.708708595857479; 0.215190648098014; -0.039799084319580;
-0.143340812469280; -0.017595122703702; 0.328483160028172;
0.654742714596673; 0.916442688538714; 1.140889511394805;
1.003444608050291]