Esercitazione 4:
Vettori e Matrici

Richiami di teoria: Norme di vettore

Principali norme di vettore:

1. $||\mathbf{x}||_1 = \sum_{i=1}^{n} |x_i|$
2. $||\mathbf{x}||_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2}$
3. $||\mathbf{x}||_\infty = \max_{1\leq i\leq n} |x_i|$

Ad esempio dato il vettore $\mathbf{x} = (1, -2, 3, -4)$ abbiamo

1. $||\mathbf{x}||_1 = |1| + |-2| + |3| + |-4| = 10$
2. $||\mathbf{x}||_2 = \sqrt{(1)^2 + (-2)^2 + (3)^2 + (-4)^2} = \sqrt{30} \simeq 5.4772$
3. $||\mathbf{x}||_\infty = \max(|1|, |-2|, |3|, |-4|) = 4$

Esercizio 1

Scrivere tre funzioni MatLab per il calcolo delle norme di vettore. Provare il codice su un vettore a scelta e confrontare i risultati con quelli forniti dalla function `norm`. Utilizzare i comandi `sum`, `max` e `sqrt`.

Richiami di teoria: Norme di matrice

Principali norme di matrice:

1. $||\mathbf{A}||_1 = \max_{1\leq k\leq n} \sum_{i=1}^{n} |a_{ik}|$ (norma indotta da $||\mathbf{x}||_1$, per colonne)
2. $||\mathbf{A}||_\infty = \max_{1\leq i\leq n} \sum_{k=1}^{n} |a_{ik}|$ (norma indotta da $||\mathbf{x}||_\infty$, per righe)
3. $||\mathbf{A}||_2 = \sqrt{\rho(A^*A)} = \sqrt{\rho(AA^*)} = ||\mathbf{A}^*||_2$ (norma indotta da $||\mathbf{x}||_2$)

Ad esempio data la matrice

$$\mathbf{A} = \begin{bmatrix} 4 & -1 & 1 \\ 1 & 3 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

abbiamo

1. $||\mathbf{A}||_1 = 5$

(colonna 1)	$\|4\| + \|1\| + \|0\| = 5$
(colonna 2)	$\|-1\| + \|3\| + \|1\| = 5$
(colonna 3)	$\|1\| + \|1\| + \|1\| = 3$
max colonna	5

2. $||\mathbf{A}||_\infty = 6$

(riga 1)	$\|4\| + \|-1\| + \|1\| = 6$
(riga 2)	$\|1\| + \|3\| + \|1\| = 5$
(riga 3)	$\|0\| + \|1\| + \|1\| = 2$
max riga	6

3. $||\mathbf{A}||_2 \simeq 4.3131$
   Da
   $$\mathbf{A} \cdot \mathbf{A}^T = \begin{bmatrix} 18 & 2 & 0 \\ 2 & 11 & 4 \\ 0 & 4 & 2 \end{bmatrix}$$
   ed utilizzando il comando `eig` per il calcolo degli autovalori si ottiene
   $$\sigma(\mathbf{A} \cdot \mathbf{A}^T) = \{0.4499, 11.9475, 18.6025)\} \quad \text{(spettro)}$$
   da cui il raggio spettrale
   $$\rho(\mathbf{A} \cdot \mathbf{A}^T) = 18.6025\,.$$

Esercizio 2

Scrivere tre function MatLab per il calcolo delle norme di matrice. Provare il codice su una matrice a scelta e confrontare i risultati con quelli forniti dalla function `norm`. Utilizzare i comandi `sum`, `max`, `eig`, `abs` e `sqrt`.

Richiami di Teoria: Condizionamento di un sistema lineare

Teorema: Sia $\delta\mathbf{A} \in \mathbb{C}^{n\times n}$ e $\delta\mathbf{b} \in \mathbb{C}^n$ rispettivamente la matrice e il vettore delle perturbazioni sui dati del sistema lineare $\mathbf{Ax} = \mathbf{b}$ con $\mathbf{A} \in \mathbb{C}^{n\times n}$ e $\mathbf{b} \in \mathbb{C}^n$ dove $\mathbf{b} \neq \mathbf{0}$ e sia $||\cdot||$ una qualunque norma matriciale indotta. Se $\mathbf{A}$ è non singolare e se $||\mathbf{A}^{-1}||\,||\delta\mathbf{A}|| < 1$, allora anche la matrice $\mathbf{A}+\delta\mathbf{A}$ è non singolare. Indicata con $||\mathbf{x}+\delta\mathbf{x}||$ la soluzione del sistema perturbato

$$(\mathbf{A}+\delta\mathbf{A})(\mathbf{x}+\delta\mathbf{x}) = \mathbf{b}+\delta\mathbf{b}$$

risulta

$$\frac{||\delta\mathbf{x}||}{||\mathbf{x}||} \leq \frac{\mathrm{cond}(\mathbf{A})}{1-\mathrm{cond}(\mathbf{A})||\delta\mathbf{A}||/||\mathbf{A}||}\left(\frac{||\delta\mathbf{A}||}{||\mathbf{A}||}+\frac{||\delta\mathbf{b}||}{||\mathbf{b}||}\right)$$

in cui $\mathrm{cond}(\mathbf{A}) = ||\mathbf{A}||\,||\mathbf{A}^{-1}||$ è il numero di condizionamento della matrice $\mathbf{A}$ misurato in una qualunque norma.

Osservazione 0: Indicando con $\epsilon_A = ||\delta\mathbf{A}||/||\mathbf{A}||$ e $\epsilon_b = ||\delta\mathbf{b}||/||\mathbf{b}||$ le perturbazioni relative della matrice $\mathbf{A}$ e del vettore $\mathbf{b}$ e con $\epsilon_x = ||\delta\mathbf{x}||/||\mathbf{x}||$ la perturbazione relativa indotta sul vettore $\mathbf{x}$, il teorema precedente può essere così formulato: la perturbazione relativa ϵ_x della soluzione, indotta dalle perturbazioni relative dei dati ϵ_A e ϵ_b, è maggiorata dall'espressione

$$\epsilon_x \leq \frac{\mathrm{cond}(\mathbf{A})}{1-\mathrm{cond}(\mathbf{A})\cdot\epsilon_A}(\epsilon_A+\epsilon_b)\,.$$

Osservazione 1: Si osservi che il numero di condizionamento è sempre maggiore o uguale a 1. Infatti:

$$\mathrm{cond}(\mathbf{A}) = ||\mathbf{A}||\,||\mathbf{A}^{-1}|| \geq ||\mathbf{A}\mathbf{A}^{-1}|| = 1\,.$$

Osservazione 2: Dalla precedente stima risulta che se $\mathrm{cond}(\mathbf{A})$ assume valori piccoli, allora piccole perturbazioni sui dati inducono piccole perturbazioni sulla soluzione e quindi il problema è ben posto: in questo caso la matrice del sistema si dice *ben condizionata*; se $\mathrm{cond}(\mathbf{A})$ assume valori grandi, allora piccole variazioni sui dati possono indurre grandi perturbazioni nella soluzione e quindi il problema può essere mal posto: in questo caso la matrice del sistema si dice *mal condizionata*.

Osservazione 3: Si osservi che la maggiorazione, può fornire una stima eccessiva dell'errore della soluzione indotta dall'errore nei dati, tenuto conto anche che tale maggiorazione vale per qualunque vettore b.

Osservazione 4: Nel caso in cui $||\delta A|| \leq 1/||A||$ si può dimostrare che:

$$\frac{||\delta x||}{||x||} \leq \mathrm{cond}(A) \left(\frac{||\delta A||}{||A||} + \frac{||\delta b||}{||b||} \right) .$$

Esempio: Condizionamento di un sistema lineare

Consideriamo, al variare di $\epsilon > 0$, la matrice

$$\mathbf{A} = \begin{bmatrix} 1 & 1-\epsilon \\ 1+\epsilon & 1 \end{bmatrix}$$

che possiede la seguente inverva

$$\mathbf{A}^{-1} = \epsilon^{-2} \begin{bmatrix} 1 & -1+\epsilon \\ -1-\epsilon & 1 \end{bmatrix} .$$

Si ha:

- $\mathrm{cond}(\mathbf{A})_1 = (2+\epsilon)^2/\epsilon^2 \geq 4/\epsilon^2$
- $\mathrm{cond}(\mathbf{A})_\infty = (2+\epsilon)^2/\epsilon^2 \geq 4/\epsilon^2$
- $\mathrm{cond}(\mathbf{A})_2 = \frac{2+\epsilon^2+2\sqrt{1+\epsilon^2}}{2+\epsilon^2-2\sqrt{1+\epsilon^2}}$

Per $\epsilon = 10^{-2}$ abbiamo:

$$\mathbf{A} = \begin{bmatrix} 1 & 0.99 \\ 1.01 & 1 \end{bmatrix}$$

e

$$\mathbf{A}^{-1} = 10^4 \begin{bmatrix} 1 & -0.99 \\ -1.01 & 1 \end{bmatrix} .$$

ed inoltre

- $\mathrm{cond}(\mathbf{A})_1 \approx 40000$
- $\mathrm{cond}(\mathbf{A})_\infty \approx 40000$
- $\mathrm{cond}(\mathbf{A})_2 \approx 40000$

Se scegliamo $\mathbf{b} = (1.99, 2.01)^T$ il sistema $\mathbf{Ax} = \mathbf{b}$ ha per soluzione $\mathbf{x} = (1,1)^T$, mentre se perturbiamo il termine noto in $\mathbf{b} = (2.0, 2.0)^T$, il sistema ha per soluzione $\mathbf{x} = (200, -200)^T$.

Poichè $\delta\mathbf{b} = (-0.01, 0.01)^T$ e $\delta\mathbf{x} = (-199, 201)^T$, si ha che ad un errore relativo sui dati di $||\delta\mathbf{b}||_\infty/||\mathbf{b}||_\infty \approx 0.5 \times 10^{-2}$ porta ad un errore relativo sui risultati pari a $||\delta\mathbf{x}||_\infty/||\mathbf{x}||_\infty \approx 0.2 \times 10^3$.

Richiami di Teoria: Matrice di Hilbert

La matrice $\mathbf{A}^{(n)}$ di ordine n, definita da

$$a_{ij}^{(n)} = \frac{1}{i+j-1} \quad i,j = 1,\cdots,n,$$

è detta matrice di $Hilbert$. Per $n = 5$ si ha

$$A^{(5)} = \begin{bmatrix} \frac{1}{1} & \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} \end{bmatrix}$$

Il comando MatLab `hilb` crea la matrice di Hilbert. La matrice di Hilbert è una matrice mal condizionata con numero di condizionamento elevato anche per valori bassi di n.

Esercizio 3

Calcolare il condizionamento $\mathrm{cond}(\mathbf{A}^{(n)})$ della matrice di Hilbert per valori di n da 2 a 10. Per calcolare il condizionamento si utilizzi la function `cond`. Si riporti inoltre il condizionamento $\mathrm{cond}_\infty(\mathbf{A}^{(n)})$ calcolato utilizzando l'inversa della matrice di hilbert calcolata numericamente (comando `inv`) oppure calcolata esattamente (comando `invhilb`).

Risultati

n	cond_matlab	stima1	stima2
2	1.928e+001	2.700e+001	2.700e+001
3	5.241e+002	7.480e+002	7.480e+002
4	1.551e+004	2.837e+004	2.837e+004
5	4.766e+005	9.437e+005	9.437e+005
6	1.495e+007	2.907e+007	2.907e+007
7	4.754e+008	9.852e+008	9.852e+008
8	1.526e+010	3.387e+010	3.387e+010
9	4.932e+011	1.100e+012	1.100e+012
10	1.603e+013	3.536e+013	3.535e+013

Esercizio 4

Per vedere l'effetto del condizionamento di una matrice sui sistemi lineari si può considerare la seguente idea:

(a) Si fissi una matrice $\mathbf{A}$ di dimensione n;

(b) Si fissi una soluzione di riferimento $\mathbf{xsol}$ (ad esempio vettore di tutti 1) su cui misurare successivamente gli errori assoluti e relativi (norma 2 di vettore);

(c) Si costruisca il termine nodo $\mathbf{b}$ del sistema lineare che garantisca la soluzione esatta ($\mathbf{b} = \mathbf{A} \cdot \mathbf{xsol}$);

(d) Si risolva il sistema lineare $\mathbf{Ax} = \mathbf{b}$ (ad esempio mediante il comando MatLab `x = A\b`);

(e) Si calcolino gli errori (assoluto e relativo) tra $\mathbf{x}$ e $\mathbf{xsol}$;

(f) Si costruisca un nuovo vettore $\hat{\mathbf{b}}$ perturbando $\mathbf{b}$ di un errore piccolo (ad esempio si utilizzi la seguente istruzione `bhat = b + 1e-3*(rand(n,1)-0.5);` per generare un vettore perturbato si è utilizzata l'istruzione `rand` che genera un numeri casuali tra 0 e 1, si è tolto 0.5 in nodo tale che i numeri casuali siano distrubuiti tra $-1/2$ e $1/2$ e quindi a media nulla, ed infine si è moltiplicato per `1e-3` per variarne l'ampiezza);

(g) Si risolva il nuovo sistema lineare $\mathbf{A}\hat{\mathbf{x}} = \hat{\mathbf{b}}$ trovando $\hat{\mathbf{x}}$;

(h) Si calcolino gli errori (assoluto e relativo) tra $\hat{\mathbf{x}}$ e $\mathbf{xsol}$;

(i) Si calcolino gli errori (assoluto e relativo) tra $\hat{\mathbf{x}}$ e $\mathbf{x}$;

Provare la suddetta idea su una matrice mal condizionata quale quella di Hilbert e su una matrice ben condizionata quale la matrice tridiagonale simmetrica con elementi sulla diagonale pari a 1 ed elementi sulla sopra e sotto-diagonale generati in modo casuale (`A = eye(n) + diag(rand(n-1,1),1)+ diag(rand(n-1,1),-1)`).

Risultati

```
MATRICE DI HILBERT
------------------

err_ass_x_xsol =
  6.1431e-004
err_rel_x_xsol =
  1.9426e-004
err_ass_xhat_xsol =
  2.6119e+008
err_rel_xhat_xsol =
```

```
  8.2596e+007
err_ass_xhat_x =
  2.6119e+008
err_rel_xhat_x =
  8.2596e+007

MATRICE TRIDIAGONALE
--------------------

err_ass_x_xsol =
  4.5776e-016
err_rel_x_xsol =
  1.4476e-016
err_ass_xhat_xsol =
  9.3529e-004
err_rel_xhat_xsol =
  2.9577e-004
err_ass_xhat_x =
  9.3529e-004
err_rel_xhat_x =
  2.9577e-004
```

Esercizio 5 (Matrici sparse)[1]

Nelle applicazioni tipicamente si ha a che fare con matrici in formato pieno (denso) per dimensioni dell'ordine di $n = 100$ elementi mentre per dimensioni superiori la matrice è tipicamente sparsa. Per matrice sparsa si intende una matrice in cui gli elementi non nulli (diversi da zero, nnz) sono proporzionali alla dimensione n della matrice, ovvero $\mathrm{nnz} = \mathcal{O}(n)$, cioè sono "in numero costante" per ogni riga della matrice.
Lo scopo dell'esercizio è confrontare i seguenti tre comandi MatLab per risolvere un sistema lineare $\mathbf{A}\mathbf{x} = \mathbf{b}$:

- `x = A\b`) con la matrice $\mathbf{A}$ memorizzata in formato pieno;
- `x = inv(A)*b`) soluzione mediante calcolo dell'inversa (matrice $\mathbf{A}$ memorizzata in formato pieno; l'inversa di una matrice sparsa è tipicamente una matrice in densa);
- `x = A\b`) con la matrice $\mathbf{A}$ memorizzata in formato sparso;

Per avere una matrice significativa si cui provare l'esempio si utilizzi il seguente comando MatLab `A = gallery('poisson',n)` che genera una matrice sparsa di ordine n^2. Una matrice in formato pieno si può ottenere utilizzando il comando `full` applicato alla matrice $\mathbf{A}$, cioè `A = full(A)`. Si consideri il termine $\mathbf{b}$ in modo tale che la soluzione del sistema sia un vettore di tutti uno.
Riportare graficamente i tempi di calcolo e l'accuratezza delle soluzioni ottenute (norme errori relativi) al variare di n. In MatLab i tempi di calcolo di un'istruzione, ad esempio `Ainv = inv(A)` può essere ottenuto dai seguenti comandi `tic, Ainv = inv(A); t = toc;` dove l'istruzione `tic` imposta il timer e l'istruzione `toc` misurano il tempo trascorso salvandolo nella variabile `t`.

[1] Facoltativo.

Risultati

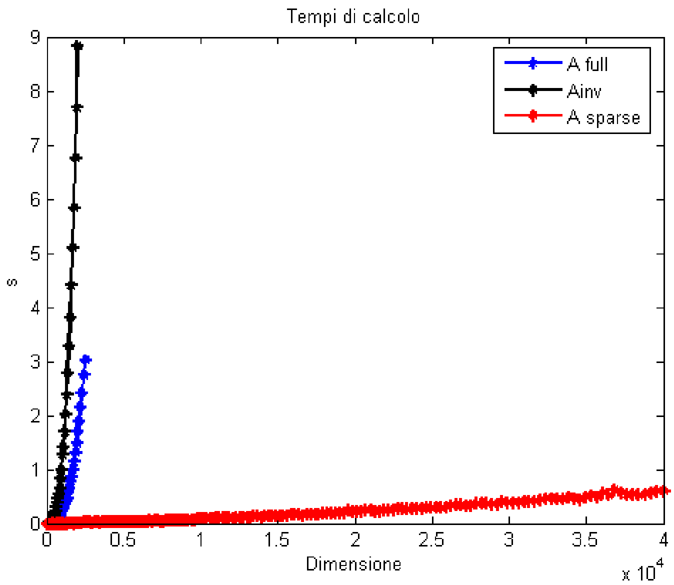

Figura 1: Tempi di calcolo.

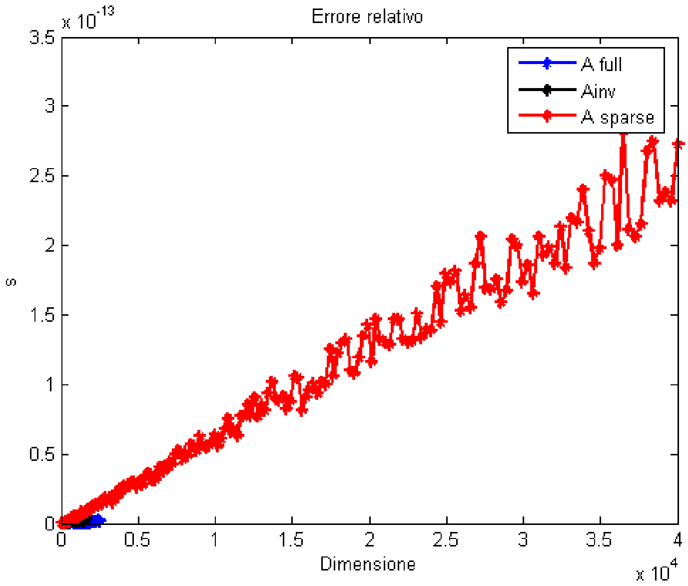

Figura 2: Errore relativo.

Esercizio 6 (Speed-up prodotto matrice-matrice)[2]

Scopo dell'esercizio è il confronto sui tempi di calcolo del prodotto matrice-matrice in MatLab/Octave rispetto ad un codice scritto in C/C++ (o Fortran).
Data una matrice $\mathbf{A}$ di dimensione $n \times m$ e una matrice $\mathbf{B}$ di dimensioni $m \times p$ la matrice prodotto $\mathbf{C} = A \times B$ di dimensioni $n \times p$ è ottenuta attraverso la seguente formula:

$$C_{ij} = \sum_{k=1}^{m} A_{ik} B_{kj} \,.$$

L'obiettivo è mettere a confronto tre codici (ovviamente della sola parte che esegue il prodotto matrice-matrice)

1. Scrittura in C/C++ del prodotto matrice-matrice con ottimizzazione del codice mediante impostazioni di flag al compilatore;
2. (opzionale) Chiamata da un programma C della routine `dgemm` delle librerie `BLAS` ottimizzate (ad esempio si scarichi le ATLAS dal sito `http://math-atlas.sourceforge.net/` e se le compili per la propria macchina);
3. Utilizzo in MatLab del prodotto matrice-matrice.

Le matrici $\mathbf{A}$ e $\mathbf{B}$ possono essere scelte ad esempio quadrate e della forma di Hilbert o con numeri casuali.
In MatLab i tempi di calcolo possono essere ottenuti dal seguente comando `tic, C = A*B; t = toc;`.

[2]Facoltativo. Scopo di questo esercizio è capire l'effetto dell'utilizzo di librerie ottimizzate in un codice di calcolo numerico.

Risultati

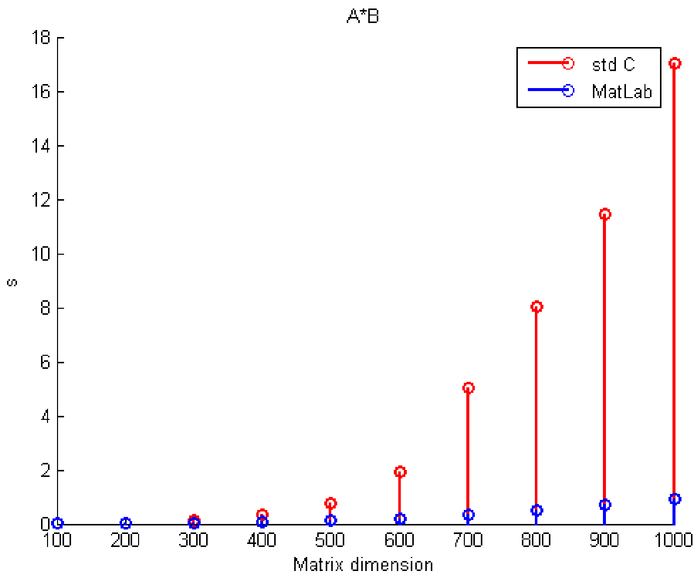

Figura 3: Tempi di calcolo del prodotto matrice-matrice.

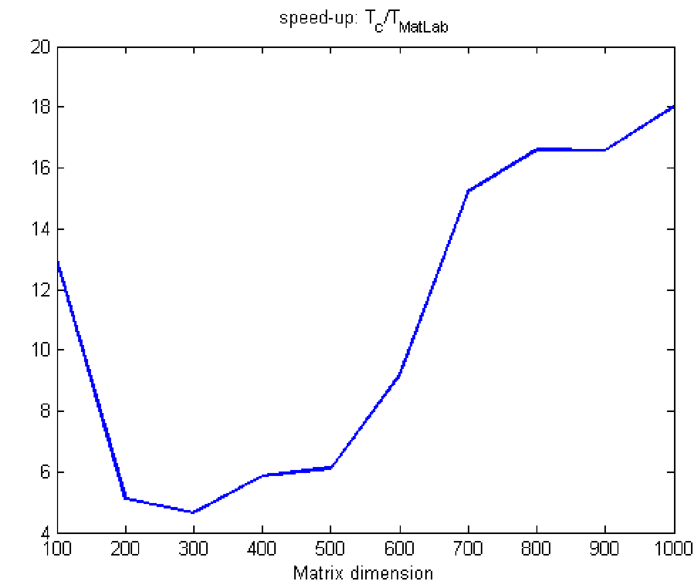

Figura 4: Speed-up.