# Calcolo di autovalori e autovettori

Alvise Sommariva

Università degli Studi di Padova
Dipartimento di Matematica

15 aprile 2014

# Autovalori

Il problema del calcolo degli autovalori di una matrice quadrata $A$ di ordine $n$ consiste nel trovare gli $n$ numeri (possibilmente complessi) $\lambda$ tali che

$$Ax = \lambda x, \; x \neq 0 \tag{1}$$

Si osservi che a seconda delle esigenze

- talvolta è richiesto solamente il calcolo di alcuni autovalori (ad esempio quelli di massimo modulo, per determinare lo spettro della matrice),
- talvolta si vogliono determinare tutti gli $n$ autovalori in $\mathbb{C}$.

## Autovalori

Per semplicità , dopo i teoremi di localizzazione di Gershgorin, mostreremo solo due metodi classici, uno per ognuna di queste classi, quello delle potenze e il metodo QR, rimandando per completezza alla monografia di Saad o a manuali di algebra lineare [2], [13].

Una interessante applicazione è l'algoritmo di **PageRank** [11], utilizzato da Google per fornire i risultati migliori tra i siti web relativamente a certe parole chiave ed in prima approssimazione basato sul calcolo di un autovettore relativo all'autovalore 1 (ad esempio via metodo delle potenze) di una matrice stocastica di dimensioni *enormi*.

## Teoremi di Gershgorin

In questo paragrafo mostriamo tre teoremi di localizzazione di autovalori dovuti a Gershgorin (cf. [2, p.76], [15]).

**Teorema**

*(Primo teorema di Gershgorin). Gli autovalori di una matrice $A$ di ordine $n$ sono tutti contenuti nell'unione dei cerchi di Gershgorin*

$$K_i = \{z \in \mathbb{C} : |z - a_{i,i}| \leq \sum_{j=1, j \neq i}^{n} |a_{i,j}|\}$$

Vediamo quale esempio la matrice

$$A = \begin{pmatrix} 15 & -2 & 2 \\ 1 & 10 & -3 \\ -2 & 1 & 0 \end{pmatrix} \quad (2)$$

## Teoremi di Gershgorin

Il primo teorema di Gershgorin stabilisce che gli autovalori stanno nell'unione dei cerchi di Gershgorin

$$\begin{aligned} K_1 &= \{z \in \mathbb{C} : |z-15| \leq |-2| + |2| = 4\} \\ K_2 &= \{z \in \mathbb{C} : |z-10| \leq |1| + |-3| = 4\} \\ K_3 &= \{z \in \mathbb{C} : |z-0| \leq |-2| + |1| = 3\} \end{aligned}$$

**Teorema**

*(Secondo teorema di Gershgorin). Se l'unione $M_1$ di $k$ cerchi di Gershgorin è disgiunta dall'unione $M_2$ dei rimanenti $n-k$, allora $k$ autovalori appartengono a $M_1$ e $n-k$ appartengono a $M_2$.*

Applicando il secondo teorema di Gershgorin, dal confronto con la figura 6 abbiamo che 1 autovalore sta nel cerchio $K_3$ mentre 2 stanno nell'unione dei cerchi $K_1$, $K_2$.

## Teoremi di Gershgorin

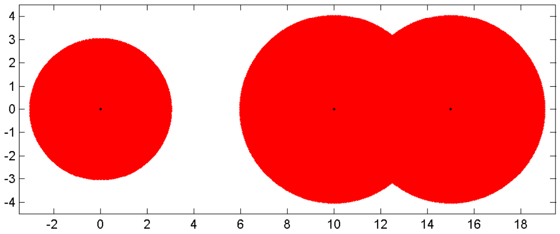


Figura: Cerchi di Gershgorin della matrice A definita in (4)

## Teoremi di Gershgorin

### Definizione

*Una matrice di ordine $n \geq 2$ è* **riducibile** *se esiste una matrice di permutazione $\Pi$ e un intero $k$, $0 < k < n$, tale che*

$$B = \Pi A \Pi^T = \begin{pmatrix} A_{1,1} & A_{1,2} \\ 0 & A_{2,2} \end{pmatrix}$$

*in cui $A_{1,1} \in \mathbb{C}^{k \times k}$, $A_{2,2} \in \mathbb{C}^{(n-k)\times(n-k)}$.*

### Definizione

*Una matrice si dice* **irriducibile** *se non è riducibile.*

## Teoremi di Gershgorin

**Nota.**

*Per verificare se una matrice $A = (a_{i,j})$ sia irriducibile (cf. [8]), ricordiamo che data una qualsiasi matrice, è possibile costruire un grafo avente come nodi gli indici della matrice. In particolare, il nodo i-esimo è connesso al nodo j-esimo se l'elemento $a_{i,j}$ è diverso da 0.*

*Il grafo associato si dice fortemente connesso se per ogni coppia $(i, j)$ posso raggiungere $j$ a partire da $i$.*

*Una matrice è irriducibile se e solo se il grafo ad essa associata (detto di adiacenza) è fortemente connesso.*

*In altre parole, una matrice  riducibile se e solo se il grafo di adiacenza ad esso associato non è fortemente connesso.*

# Teoremi di Gershgorin

**Teorema**

*(Terzo teorema di Gershgorin). Se la matrice di ordine $n$ è irriducibile e un autovalore $\lambda$ sta sulla frontiera dell'unione dei cerchi di Gershgorin, allora sta sulla frontiera di ogni cerchio di Gershgorin.*

**Esempio**. La matrice

$$B = \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{pmatrix}$$

è irriducibile, in quanto il grafico di adiacenza è fortemente connesso. Gli autovalori stanno nella disco centrato in $(2, 0)$ e raggio 2 (primo teorema di Gershgorin), ma non sulla frontiera (terzo teorema di Gershgorin). Quindi è non singolare.

# Teoremi di Gershgorin

Vediamo ora in Matlab quali sono effettivamente gli autovalori. si ha

```
>> A=[15 -2 2; 1 10 -3; -2 1 0]
A =
    15    -2     2
     1    10    -3
    -2     1     0

>> eig(A)
ans =
    0.5121
   14.1026
   10.3854

>>
```

a conferma di quanto stabilito dai primi due teoremi di Gershgorin.

## Teoremi di Gershgorin

**Nota.**

*Ricordiamo che $A$ è una matrice a coefficienti reali, allora $A$ e $A^T$ hanno gli stessi autovalori (cf. [2, p.47]) e quindi applicando i teoremi di Gershgorin alla matrice trasposta possiamo ottenere nuove localizzazioni degli autovalori.*

*Nel caso $A$ sia a coefficienti complessi, se $\lambda$ è un autovalore di $A$ allora il suo coniugato $\overline{\lambda}$ è autovalore della sua trasposta coniugata $\overline{A}$. Da qui si possono fare nuove stime degli autovalori di $A$.*

*Cosa possiamo dire relativamente agli autovalori di $A$ se applichiamo i teoremi di Gershgorin ad $A^T$ invece che ad $A$?*

## Metodo delle potenze

Il metodo delle potenze è stato suggerito nel 1913 da Muntz ed è particolarmente indicato per il calcolo dell'autovalore di massimo modulo di una matrice.

Sia $A$ una matrice quadrata di ordine $n$ con

- $n$ autovettori $x_1, \ldots, x_n$ linearmente indipendenti,
- autovalori $\lambda_1, \ldots, \lambda_n$ tali che

$$|\lambda_1| > |\lambda_2| \geq \ldots \geq |\lambda_n|. \tag{3}$$

## Metodo delle potenze

A tal proposito ricordiamo (cf. [3], p. 951) i seguenti risultati.

- Una matrice $A$ è **diagonalizzabile** se e solo se possiede $n$ autovettori linearmente indipendenti.
- Se tutti gli autovalori di $A$ sono distinti la matrice è diagonalizzabile; l'opposto è ovviamente falso (si pensi alla matrice identica).
- Una matrice simmetrica (hermitiana) è diagonalizzabile. L'opposto è ovviamente falso, visto che la matrice

$$A = \begin{pmatrix} 15 & 0 \\ 1 & 10 \end{pmatrix} \tag{4}$$

è diagonalizzabile visto che ha tutti gli autovalori distinti ma non è simmetrica.

## Metodo delle potenze

Il metodo delle potenze è definito come segue. Sia $t_0 \in \mathcal{R}^n$ definito da

$$t_0 = \sum_{i=1}^{n} \alpha_i x_i, \ \alpha_1 \neq 0,$$

e si generi la successione

$$\begin{aligned} y_0 &= t_0 && (5) \\ y_k &= A y_{k-1}, \ k = 1, 2, \ldots && (6) \end{aligned}$$

## Metodo delle potenze

### Teorema

*Sia $A$ è una matrice quadrata diagonalizzabile avente autovalori $\lambda_k$ tali che*

$$|\lambda_1| > |\lambda_2| \geq \ldots \geq |\lambda_n|.$$

*Siano $u_k \neq 0$ autovettori relativi all'autovalore $\lambda_k$, cioè*

$$Au_k = \lambda_k u_k.$$

*Sia*

$$y_0 = \sum_k \alpha_k u_k, \quad \alpha_1 \neq 0.$$

*La successione $\{y_s\}$ definita da $y_{s+1} = Ay_s$ converge ad un vettore parallelo a $x_1$ e che il coefficiente di Rayleigh (relativo al prodotto scalare euclideo)*

$$\rho(y_s, A) := \frac{(y_s, Ay_s)}{(y_s, y_s)} \to \lambda_1. \tag{7}$$

## Metodo delle potenze

### Dimostrazione.

*Per la dimostrazione si confronti [7, p.171]. Essendo la matrice $A$ diagonalizzabile, esistono $n$ autovettori $u_k$ (relativi rispettivamente agli autovalori $\lambda_k$) che sono linearmente indipendenti e quindi formano una base di $\mathbb{R}^n$. Sia*

$$y_0 = \sum_k \alpha_k u_k, \quad \alpha_1 \neq 0.$$

*Essendo $Au_k = \lambda_k u_k$ abbiamo*

$$y_1 = Ay_0 = A(\sum_k \alpha_k u_k) = \sum_k \alpha_k A u_k = \sum_k \alpha_k \lambda_k u_k$$

$$y_2 = Ay_1 = A(\sum_k \alpha_k \lambda_k u_k) = \sum_k \alpha_k \lambda_k A u_k = \sum_k \alpha_k \lambda_k^2 u_k$$

*e più in generale*

$$y_{s+1} = Ay_s = A(\sum_k \alpha_k \lambda_k^s u_k) = \sum_k \alpha_k \lambda_k^s A u_k = \sum_k \alpha_k \lambda_k^{s+1} u_k.$$

## Metodo delle potenze

*Osserviamo ora che*

$$\frac{y_{s+1}}{\lambda_1^{s+1}} = \sum_k \alpha_k \frac{\lambda_k^{s+1}}{\lambda_1^{s+1}} u_k \quad (8)$$

*per cui essendo per* $k > 1$

$$\left| \frac{\lambda_k^{s+1}}{\lambda_1^{s+1}} \right| < 1,$$

*si ha*

$$\lim_{s \to +\infty} \left( \frac{\lambda_k}{\lambda_1} \right)^s = 0$$

*e quindi la direzione di* $\frac{y_s}{\lambda_1^s}$*, che è la stessa di* $y_s$*, tende a quella dell'autovettore* $u_1$.

## Metodo delle potenze

*Si osservi che il coefficiente di Rayleigh $\rho(\cdot, A) = (x, Ax)/(x, x)$ è continuo in ogni $x \neq 0$, $x \in \mathbb{R}^n$ in quanto tanto il numeratore quanto il denominatore sono funzioni (multivariate) polinomiali (quadratiche) delle componenti $x_k$ di $x = (x_k)_k \in \mathbb{R}^n$, che sono appunto continue.*

*Per continuità , se $y_s/\lambda^s \to \alpha_1 u_1$ allora, essendo $\lambda_1 \neq 0$, da*

$$\begin{aligned} \lim_s \rho(y_s, A) &:= \lim_s \frac{(y_s, Ay_s)}{(y_s, y_s)} = \lim_s \frac{(y_s/\lambda_1^s, A(y_s/\lambda_1^s))}{(y_s/\lambda_1^s, y_s/\lambda_1^s)} \\ &= \frac{(\alpha_1 u_1, A(\alpha_1 u_1))}{(\alpha_1 u_1, \alpha_1 u_1)} = \frac{(u_1, Au_1)}{(u_1 u_1)} = \lambda_1, \end{aligned} \tag{9}$$

*ricaviamo che il coefficiente di Rayleigh $\rho(y_s, A)$ converge a $\lambda_1$.*

## Metodo delle potenze

### Nota.

*Il metodo converge anche nel caso in cui*

$$\lambda_1 = \ldots = \lambda_r$$

*per $r > 1$, tuttavia non è da applicarsi quando l'autovalore di modulo massimo non è unico.*

### Nota.

*In virtú di possibili underflow e underflow si preferisce normalizzare il vettore $y_k$ definito in (5). Così l'algoritmo diventa*

$$\begin{aligned} u_k &= At_{k-1} && (10)\\ t_k &= \frac{u_k}{\beta_k},\ \beta_k = \|u_k\|_2 && (11)\\ l_k &= \rho(t_k, A) && (12) \end{aligned}$$

*dove $\rho(t_k, A)$ è il coefficiente di Rayleigh definito in (7).*

## Metodo delle potenze inverse

Una variante particolarmente interessante del metodo delle potenze è stata scoperta da Wielandt nel 1944 [9] ed e' particolarmente utile nel caso in cui $A$ sia una matrice quadrata con $n$ autovettori linearmente indipendenti,

$$|\lambda_1| \geq |\lambda_2| \geq \ldots > |\lambda_n| > 0. \qquad (13)$$

e si desideri calcolare il **piú piccolo autovalore in modulo**, cioè $\lambda_n$, applicando il metodo delle potenze ad $A^{-1}$.

## Metodo delle potenze inverse

Si ottiene cosí la successione $\{t_k\}$ definita da

$$\begin{aligned} Au_k &= t_{k-1} && (14) \\ t_k &= \frac{u_k}{\beta_k},\ \beta_k = \|u_k\|_2 && (15) \end{aligned}$$

e convergente ad un vettore parallelo a $x_n$. La successione di coefficienti di Rayleigh è tale che

$$\rho(t_k, A^{-1}) := \frac{(t_k, A^{-1}t_k)}{(t_k, t_k)} = \frac{(t_k, u_{k+1})}{(t_k, t_k)} \to 1/\lambda_n. \quad (16)$$

da cui è immediato calcolare $\lambda_n$.

## Metodo delle potenze inverse

Vediamo in dettaglio questo punto. Se $\{\xi_i\}$ sono gli autovalori di $A^{-1}$ con

$$|\xi_1| > |\xi_2| \geq |\xi_3| \geq \ldots \geq |\xi_n|$$

allora il metodo delle potenze inverse calcola un'approssimazione di $\xi_1$ e di un suo autoversore $x$.

Si osserva subito che se $A^{-1}x = \xi_i x$ (con $\xi_i \neq 0$) allora moltiplicando ambo membri per $\xi_i^{-1}A$ si ottiene leggendo da destra a sinistra $Ax = \xi_i^{-1}x$ cioè $\xi_i^{-1}$ è un autovalore di $A$ e $x$ è non solo autovettore di $A^{-1}$ relativo all'autovalore $\xi_i$, ma pure autovettore di $A$ relativo all'autovalore $\xi_i^{-1}$.

## Metodo delle potenze inverse

Conseguentemente se $\xi_1$ è l'autovalore di massimo modulo di $A^{-1}$ e $\lambda_n$ è l'autovalore di minimo modulo di $A$ si ha $\lambda_n = \xi_i^{-1}$ e che

$$A^{-1}x = \xi_1 x \Rightarrow Ax = \xi_1^{-1}x = \lambda_n x$$

Notiamo che il metodo delle potenze inverse, calcola $\xi_1 = \lambda_n^{-1}$ e il relativo autovettore $x$.

Per ottenere $\lambda_n$ viene naturale calcolare $\xi_1^{-1}$, ma usualmente essendo $x$ autovettore di $A$ relativo a $\lambda_n$ si preferisce calcolare $\lambda_n$ via il coefficente di Rayleigh

$$\rho(x, A) := \frac{(x, Ax)}{(x, x)}.$$

## Metodo delle potenze inverse

In generale, fissato $\mu \in \mathbb{C}$ è possibile calcolare, se esiste unico, l'autovalore $\lambda$ più vicino a $\mu$ considerando il seguente pseudocodice [6, p.181]:

$$
\begin{aligned}
(A - \mu I)\, z_k &= q_{k-1} && (17)\\
q_k &= z_k / \|z_k\|_2 && (18)\\
\sigma_k &= (q_k)^H A q_k && (19)
\end{aligned}
$$

## Metodo delle potenze inverse

Ricordiamo che se $\lambda$ è autovalore di $A$ allora

$$Ax = \lambda x \Rightarrow (A - \mu I)x = \lambda x - \mu x = (\lambda - \mu)x$$

e quindi $\lambda - \mu$ è autovalore di $A - \mu I$.

Il metodo delle potenze inverse applicato a $A - \mu I$ calcola il minimo autovalore $\sigma = \lambda - \mu$ in modulo di $A - \mu I$ cioè il $\sigma$ che rende minimo il valore di $|\sigma| = |\lambda_i - \mu|$, dove $\lambda_i$ sono gli autovalori di $A$.

Quindi essendo $\lambda_i = \sigma_i - \mu$ si ottiene pure il $\lambda_i$ più vicino a $\mu$.
Per versioni piú sofisticate di questa tecnica detta di *shift* (o in norma infinito invece che in norma 2) si confronti [2, p.379].

**Problema**. Si può applicare il metodo delle potenze inverse con shift $\mu$ nel caso $\mu$ sia proprio un autovalore di A?

## Metodo QR

Il metodo QR, considerato tra i 10 algoritmi più rilevanti del ventesimo secolo, cerca di calcolare tutti gli autovalori di una matrice $A$.

Sia $A$ una matrice quadrata di ordine $n$. Utilizzando il metodo di Gram-Schmidt è possibile fattorizzare la matrice A come prodotto di due matrici $Q$ ed $R$ con $Q$ unitaria (cioè $Q^T * Q = Q * Q^T = I$) ed $R$ triangolare superiore.

## Metodo QR

Citiamo alcune cose:

- La matrice $A$ ha quale sola particolarità di essere quadrata. Nel caso generale però la sua fattorizzazione QR in generale non è unica bensì determinata a meno di una matrice di *fase* (cf. [2, p.149]) .
- Come osservato in [1] p. 614, tale fattorizzazione non è unica (per una opportuna $Q$ unitaria, i segni delle componenti sulla diagonale della matrice $R$ possono essere scelti arbitrariamente). Nel caso sia non singolare, allora tale fattorizzazione è unica qualora si chieda che i coefficienti diagonali di $R$ siano positivi.
- La routine Matlab `qr` effettua tale fattorizzazione. Si consiglia di consultare l'`help` di Matlab, per consultare le particolarità di tale routine.

# Metodo QR

- Se la matrice $H$ è simile a $K$ (cioè esiste una matrice non singolare $S$ tale che $H = S^{-1}KS$) allora $H$ e $K$ hanno gli stessi autovalori.

  Si può vedere facilmente che la relazione di similitudine è transitiva, cioè se $H_1$ è simile ad $H_2$ e $H_2$ è simile ad $H_3$ allora $H_1$ è simile ad $H_3$.
- Dal teorema di Bauer-Fike, segue che se $A = V\Lambda V^{-1}$ con $\Lambda$ diagonale, e $\mu$ è autovalore di $A + \Delta$, allora esiste $\lambda$ autovalore di $A$ tale che $|\lambda - \mu| \leq \mathcal{K}_\infty(A)\|\Delta\|_\infty$.

Il metodo QR venne pubblicato indipendentemente nel 1961 da Francis e da Kublanovskaya e successivamente implementato in EISPACK. Ci limiteremo a considerare versioni di base del metodo.

# Metodo QR

Sia

$$A_0 = A = Q_0 R_0$$

e si ponga

$$A_1 := R_0 Q_0.$$

Poichè

$$Q_0 A_1 Q_0^T = Q_0 A_1 Q_0^T = Q_0 R_0 Q_0 Q_0^T = A_0$$

la matrice $A_1$ è simile ad $A_0$ (si ponga $S = Q_0^{-1} = Q_0^T$) e quindi ha gli stessi autovalori. Sia quindi in generale

$$A_k = Q_k R_k$$

$$A_{k+1} = R_k Q_k.$$

Per le stesse motivazioni $A_{k+1}$ è simile ad $A_k$, e per transitività ad $A_0$. Quindi $A_{k+1}$ ha gli stessi autovalori di $A_0$.

## Metodo QR

Per la convergenza del metodo esistono vari risultati (cf. [3, p.393], [4, p.352], [7, p.180]). Da [6, p.169]

### Teorema

*Se* $A \in \mathbb{R}^{n \times n}$ *ha autovalori tutti distinti in modulo, con*

$$|\lambda_1| > \ldots > |\lambda_n| \tag{20}$$

*allora l'alg. QR converge a* $A_\infty = (a_{i,j}^\infty)$ *triangolare sup., cioè*

$$\lim_k A_k = \begin{pmatrix} a_{1,1}^\infty & a_{1,2}^\infty & \ldots & \ldots & a_{1,n}^\infty \\ 0 & a_{2,2}^\infty & a_{2,3}^\infty & \ldots & a_{2,n}^\infty \\ 0 & 0 & a_{3,3}^\infty & \ldots & a_{3,n}^\infty \\ 0 & 0 & \ldots & \ldots & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & a_{n-1,n-1}^\infty & a_{n-1,n}^\infty \\ 0 & 0 & 0 & 0 & a_{n,n}^\infty \end{pmatrix} \tag{21}$$

*con* $\lambda_k = a_{k,k}^\infty$.

## Metodo QR

*Inoltre*

- *Se la condizione (20) non è verificata si può dimostrare che la successione $\{A_k\}$ tende a una forma triangolare a blocchi.*
- *se $A_k = (a_{i,j}^{(k)})$, e $\lambda_{i-1} \neq 0$*

$$|a_{i,i-1}^{(k)}| = \mathcal{O}\left(\frac{|\lambda_i|}{|\lambda_{i-1}|}\right)^k, \; i = 2, \ldots, n, \;\; k \to \infty. \qquad (22)$$

- *se la matrice è simmetrica, allora*

$$A_\infty = diag(\lambda_1, \ldots, \lambda_1).$$

- *se $A$ è una matrice Hessenberg superiore allora l'algoritmo QR converge ad $A_\infty$ triangolare a blocchi, simile ad $A$ e con gli autovalori di ogni blocco diagonale tutti uguali in modulo.*

## Implementazione del metodo QR

Nelle implementazioni si calcola con un metodo scoperto da Householder (ma esiste un metodo alternativo dovuto a Givens) una matrice di Hessenberg $T$

$$T = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & a_{2,3} & \dots & a_{2,n} \\ 0 & a_{3,2} & a_{3,3} & \dots & a_{3,n} \\ 0 & 0 & a_{4,3} & \dots & a_{4,n} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & a_{n,n-1} & a_{n,n} \end{pmatrix}$$

simile ad $A$ ed in seguito si applica il metodo QR relativamente alla matrice $T$. Se $A$ è simmetrica la matrice $T$ risulta tridiagonale simmetrica.

## Implementazione del metodo QR

Si può mostrare che se $A$ è una matrice di Hessenberg superiore, allora $A = QR$ con $Q$ di Hessenberg superiore.

Inoltre, se $A$ è tridiagonale allora $A = QR$ con $Q$ di Hessenberg e $R$ triangolare superiore con $r_{i,j} = 0$ qualora $j - i \geq 2$.

In entrambi i casi le iterazioni mantengono la struttura, cioè se $A_0 = T$ è di Hessenberg, allora $A_k$ è di Hessenberg, se $A_0 = T$ è tridiagonale allora $A_k$ è tridiagonale.

## Implementazione del metodo QR

Il numero di moltiplicazioni necessarie

- all'algoritmo di Givens per calcolare tale matrice $T$ a partire da $A$ è approssimativamente $10n^3/3$;
- all'algoritmo di Householder per calcolare tale matrice $T$ a partire da $A$ è approssimativamente $5n^3/3$.

Il metodo QR applicato ad una matrice A in forma di Hessenberg superiore ha ad ogni passo un costo di $2n^2$ operazioni moltiplicative.

Per versioni piú sofisticate come il metodo QR con shift, si veda [3], p. 394.

# Il metodo delle potenze in Matlab

Partiamo con una versione semplice `power_basic` del metodo delle potenze

```
function [lambda1, x1, niter, err]=power_basic(A,z0,toll,nmax)

% INPUT:
% A    : MATRICE DI CUI VOGLIAMO CALCOLARE L'AUTOVALORE DI MASSIMO MODULO.
% z0   : VETTORE INIZIALE (NON NULLO).
% toll: TOLLERANZA.
% nmax: NUMERO MASSIMO DI ITERAZIONI.
%
% OUTPUT:
% lambda1  : VETTORE DELLE APPROSSIMAZIONI DELL'AUTOVALORE DI MASSIMO MODULO.
% x1       : AUTOVETTORE RELATIVO ALL'AUTOVALORE DI MASSIMO MODULO.
% niter    : NUMERO DI ITERAZIONI.
% err      : VETTORE DEI RESIDUI PESATI RELATIVI A "lambda1".
%
% TRATTO DA QUARTERONI-SALERI, "MATEMATICA NUMERICA", p. 184.
%
```

# Il metodo delle potenze in Matlab

```
q=z0/norm(z0); q2=q; err=[]; lambda1=[];
res=toll+1; niter=0; z=A*q;
while (res >= toll & niter <= nmax)
    q=z/norm(z); z=A*q; lam=q'*z; x1=q;
    z2=q2'*A; q2=z2/norm(z2); q2=q2'; y1=q2; costheta=abs(y1'*x1);
    niter=niter+1; res=norm(z-lam*q)/costheta;
    err=[err; res]; lambda1=[lambda1; lam];
end
```

## Il metodo delle potenze in Matlab

Qualche nota

- il vettore iniziale `z0` e' normalizzato ed in `err`, `lambda1` vengono memorizzati rispettivamente i valori dell'errore compiuto e dell'autovalore di massimo modulo $\lambda_{\max}$;
- l'assegnazione `res=toll+1;` forza l'algoritmo ad entrare nel ciclo `while`, mentre `z=A*q;` è una quantità da utilizzarsi per il calcolo dell'autovalore $\lambda_{\max}$;

## Il metodo delle potenze in Matlab

- nel ciclo `while`, `q` è un'approssimazione di un autoversore relativo a $\lambda_{\max}$, mentre `lam` di $\lambda_{\max}$;
- il ciclo si interrompe se un numero massimo di iterazioni `niter` è raggiunto oppure

$$\frac{||Aq^k - \lambda^k||_2}{|\cos(\theta_{\lambda_k})|} < \text{tol}$$

dove $\theta_{\lambda_k}$ è l'angolo formato tra (un'approssimazione del)l'autovalore destro `x1` e sinistro `y1` associati a `lam` (cf. [6, p.180])

## Il metodo delle potenze in Matlab: esempio I

Testiamo il codice per il calcolo dell'autoval. di massimo modulo di

$$A = \begin{pmatrix} -15.5 & 7.5 & 1.5 \\ -51 & 25 & 3 \\ -25.5 & 7.5 & 11.5 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 2 & 3 \\ 2 & 5 & 6 \\ 7 & 9 & 3 \end{pmatrix} \cdot \begin{pmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 & 3 \\ 2 & 5 & 6 \\ 7 & 9 & 3 \end{pmatrix}^{-1} \quad (23)$$

La matrice $A$ è diagonalizzabile e ha autovalori 10, 10, 1. Si può vedere che una base di autovettori relativa agli autovalori 10, 10, 1 è composta da $(1, 2, 7)$, $(2, 5, 9)$, $(3, 6, 3)$. Quale vettore iniziale del metodo delle potenze consideriamo

$$z_0 = (1, 1, 1) = (7/6) \cdot (1, 2, 7) - 1 \cdot (2, 5, 9) + (11/18) \cdot (3, 6, 3)$$

e quindi il metodo delle potenze applicato ad $A$, e avente quale punto iniziale $z_0$ può essere utilizzato per il calcolo dell'autovalore di massimo modulo di $A$, poichè $\alpha_1 = 7/6 \neq 0$.

# Il metodo delle potenze in Matlab: esempio I

Dalla shell di Matlab/Octave:

```
>> S=[1 2 3; 2 5 6; 7 9 3];
>> D=diag([10 10 1]);
>> A=S*D*inv(S)
A =
  -15.5000     7.5000     1.5000
  -51.0000    25.0000     3.0000
  -25.5000     7.5000    11.5000
>> z0=[1 1 1]';
>> toll=10^(-8);
>> nmax=10;
>> format short e;
```

# Il metodo delle potenze in Matlab: esempio I

```
>> [lambda1, x1, niter, err]=power_basic(A,z0,toll,nmax)
lambda1 =
   1.1587e+001
   1.0138e+001
   1.0014e+001
   1.0001e+001
     ...
   1.0000e+001
   1.0000e+001
   1.0000e+001
x1 =
  -2.8583e-001
  -9.1466e-001
  -2.8583e-001
niter =
     10
err =
   2.2466e+000
   2.1028e-001
   2.0934e-002
   2.0925e-003
   ?
   2.0924e-007
   2.0924e-008
   2.0924e-009
>>
```

# Il metodo delle potenze in Matlab: esempio I

La convergenza è abbastanza veloce come si vede dalla quantità err, che consiste in un particolare residuo pesato.
Una questione sorge spontanea. Cosa sarebbe successo se avessimo utilizzato l'algoritmo senza normalizzazione come ad esempio power_method definito da

```
function [lambda,v]=power_method(A,x0,maxit)

v=x0;

for index=1:maxit
    v_old=v;
    v=A*v_old;
    lambda=(v_old'*v)/(v_old'*v_old);
end
```

## Il metodo delle potenze in Matlab: esempio I

Proviamo il test, facendo iterare il metodo prima 5, poi 100 volte e alla fine 1000 volte (si noti il settaggio della variabile `maxit` relativa al numero di iterazioni da compiere):

```
>> x0=[1 1 1]'
x0 =
     1
     1
     1
>> A=[-15.5 7.5 1.5; -51 25 3; -25.5 7.5 11.5]
A =
  -15.5000    7.5000    1.5000
  -51.0000   25.0000    3.0000
  -25.5000    7.5000   11.5000
>> [lambda,v]=power_method(A,x0,5)
lambda =
   10.0014
v =
  1.0e+005 *
   -0.8333
   -2.6666
   -0.8333
```

# Il metodo delle potenze in Matlab: esempio I

```
>> [lambda,v]=power_method(A,x0,100)
lambda =
    10.0000
v =
   1.0e+100 *
    -0.8333
    -2.6667
    -0.8333
>> [lambda,v]=power_method(A,x0,1000)
lambda =
    NaN
v =
    NaN
    NaN
    NaN
>>
```

## Il metodo delle potenze in Matlab: esempio I

La ragione è semplice. Per $k$ relativamente piccolo si ha $A \cdot t_k \approx 10 \cdot t_k$ e quindi per $s \geq k$

$$t_s \approx A^{s-k} \cdot t_k \approx 10 \cdot A^{s-k-1} \cdot t_k \approx \cdots \approx 10^{s-k} \cdot t_k$$

da cui

$$\|t_s\|_2 \approx 10^{s-k} \cdot \|t_k\|_2$$

spiegando quindi perchè si possano avere problemi di overflow applicando l'algoritmo di base.

## Il metodo delle potenze in Matlab: esempio II

Proviamo un test diverso, questa volta con la matrice (diagonalizzabile)

$$A = \begin{pmatrix} 1 & 2 \\ 0 & -1 \end{pmatrix},$$

avente autovalori $\lambda_1 = 1$ e $\lambda_2 = -1$ e autovettori linearmente indipendenti (1, 0), (−1, 1). Quale vettore iniziale poniamo

$$x_0 = (1, 3) = 4 \cdot (1, 0) + 3 \cdot (-1, 1)$$

e quindi il metodo delle potenze applicato ad A, partendo da $x_0$ può essere sicuramente applicato. D'altra parte dubitiamo converga in quanto $|\lambda_1| = |\lambda_2| = 1$ pur essendo $\lambda_1 \neq \lambda_2$.

# Il metodo delle potenze in Matlab: esempio II

Dalla shell di Matlab/Octave:

```
>> A=[1 2; 0 -1]
A =
     1     2
     0    -1
>> [lambda1, x1, niter, err]=power_basic(A,[1; 3],10^(-8),15)
lambda1 =
  -3.4483e-002
  -2.0000e-001
  ...
  -3.4483e-002
  -2.0000e-001
  -3.4483e-002
  -2.0000e-001
x1 =
   3.1623e-001
   9.4868e-001
niter =
    16
err =
   4.4567e-001
   2.4000e+000
   ...
   4.4567e-001
   2.4000e+000
   4.4567e-001
   2.4000e+000
>>
```



Dal residuo pesato è chiaro che il metodo non converge, e come anticipato il motivo è la presenza di autovalori distinti aventi modulo massimo.

## Il metodo delle potenze in Matlab: esempio III

Vediamo il caso della matrice diagonalizzabile (autovalori distinti!)

$$A = \begin{pmatrix} 1 & 2 \\ 0 & 10 \end{pmatrix},$$

in cui il metodo funziona rapidamente ($\lambda_{\max} = 10$).

```
>> A=[1 2; 0 10];  [lambda1 , x1 , niter , err]=power_basic(A,[1; 3],10^(-8),15)
lambda1 =
   9.9779e+000
   9.9979e+000
   9.9998e+000
   1.0000e+001
   1.0000e+001
   1.0000e+001
   1.0000e+001
   1.0000e+001
x1 =
   2.1693e-001
   9.7619e-001
niter =
     8
err =
   9.6726e-002
   9.7529e-003
   9.7610e-004
   9.7618e-005
   9.7619e-006
   9.7619e-007
   9.7619e-008
   9.7619e-009
```



Si osservi che il metodo termina in quanto l'errore pesato `err` è minore della tolleranza $toll = 10^{-8}$.

# Il metodo delle potenze inverse in Matlab

Una versione di base `invpower` del metodo delle potenze inverse [6, p.184] è

```
function [lambda, x, niter, err]=invpower(A,z0,mu,toll,nmax)

% DATO UN VALORE mu, SI CALCOLA L'AUTOVALORE "lambda_mu" PIU' VICINO A mu.

% INPUT:
% A    : MATRICE DI CUI VOGLIAMO CALCOLARE L'AUTOVALORE "lambda_mu".
% z0   : VETTORE INIZIALE (NON NULLO).
% mu   : VALORE DI CUI VOGLIAMO CALCOLARE L'AUTOVALORE PIU' VICINO.
% toll: TOLLERANZA.
% nmax: NUMERO MASSIMO DI ITERAZIONI.
%
% OUTPUT:
% lambda  : VETTORE DELLE APPROSSIMAZIONI DELL'AUTOVALORE DI MINIMO MODULO.
% x       : AUTOVETTORE RELATIVO ALL'AUTOVALORE DI MINIMO MODULO.
% niter   : NUMERO DI ITERAZIONI.
% err     : VETTORE DEI RESIDUI PESATI RELATIVI A "lambda".
%
% TRATTO DA QUARTERONI-SALERI, "MATEMATICA NUMERICA", p. 184.
%
```

# Il metodo delle potenze inverse in Matlab

```
n=max(size(A)); M=A-mu*eye(n); [L,U,P]=lu(M);
q=z0/norm(z0); q2=q'; err=[]; lambda=[];
res=toll+1; niter=0;
while (res >= toll & niter <= nmax)
    niter=niter+1; b=P*q; y=L\b; z=U\y;
    q=z/norm(z); z=A*q; lam=q'*z;
    b=q2'; y=U'\b; w=L'\y;
    q2=(P'*w)'; q2=q2/norm(q2); costheta=abs(q2*q);
    if (costheta > 5e-2)
        res=norm(z-lam*q)/costheta; err=[err; res]; lambda=[lambda; lam];
    else
        disp('\n \t [ATTENZIONE]: AUTOVALORE MULTIPLO'); break;
    end
    x=q;
end
```

## Il metodo delle potenze inverse in Matlab

Forniamo ora alcune spiegazioni del codice in `invpower`.

- Per risolvere il sistema lineare in 17, si effettua una fattorizzazione $PM = LU$ della matrice $M = A - \mu I$;
- All'interno del ciclo while, nella prima riga si calcola $z_k$, mentre nella successiva un suo versore $q_k$, e $\sigma_k$ è immagazzinato in `lam`;
- Similmente al metodo diretto si effettua il prodotto scalare di un'autovalore sinistro con uno destro.

## Il metodo delle potenze inverse in Matlab: esempio I

Applichiamo il metodo delle potenze inverse per il calcolo dell'autovalore più piccolo in modulo della matrice

$$A = \begin{pmatrix} -15.5 & 7.5 & 1.5 \\ -51 & 25 & 3 \\ -25.5 & 7.5 & 11.5 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 5 & 6 \\ 7 & 9 & 3 \end{pmatrix} \cdot \begin{pmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 2 & 3 \\ 2 & 5 & 6 \\ 7 & 9 & 3 \end{pmatrix}^{-1} \quad (24)$$

Come visto la matrice $A$ è quindi diagonalizzabile, ha autovalori 10, 10, 1 e relativi autovettori è $(1,2,7)$, $(2,5,9)$, $(3,6,3)$ formanti una base di $\mathbb{R}^3$. Quale vettore iniziale consideriamo

$$z_0 = (1,1,1) = (7/6) \cdot (1,2,7) - 1 \cdot (2,5,9) + (11/18) \cdot (3,6,3)$$

e quindi il metodo delle potenze inv. applicato ad $A$, e avente quale punto iniziale $z_0$ può essere utilizzato per il calcolo dell'autovalore di minimo modulo di $A$.

## Il metodo delle potenze inverse in Matlab: esempio I

```
>> z0=[1;1;1]; mu=0; toll=10^(-8); nmax=10;
>> A=[-15.5 7.5 1.5; -51 25 3; -25.5 7.5 11.5];
>> [lambda, x, niter, err]=invpower(A,z0,mu,toll,nmax)
lambda =
   0.39016115351993
   0.94237563941268
   0.99426922936854
   0.99942723776656
   0.99994272692315
   0.99999427272378
   0.99999942727270
   0.99999994272728
   0.99999999427273
x =
   0.40824829053809
   0.81649658085350
   0.40824829053809
niter =
     9
err =
   0.81535216507377
   0.08358101289062
   0.00838126258396
   0.00083836078891
   0.00008383842712
   0.00000838386620
   0.00000083838685
   0.00000008383868
   0.00000000838387
>>
```



La convergenza è lineare (come si intuisce dalle approssimazioni contenute nel vettore `lambda`).

# Il metodo delle potenze inverse in Matlab: esempio I

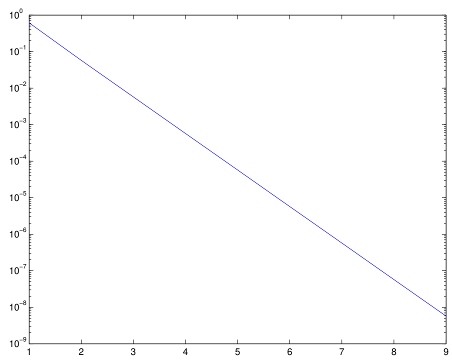


Figura: Grafico che illustra la convergenza lineare del metodo delle potenze inverse nell'esempio 1.

## Il metodo delle potenze inverse in Matlab: esempio I

Per vederlo, dalla shell di Matlab/Octave calcoliamo l'errore assoluto/relativo relativo all'autovalore 1:

```
>> s=1-lambda
s =
    0.60983884648007
    0.05762436058732
    0.00573077063146
    0.00057276223344
    0.00005727307685
    0.00000572727622
    0.00000057272730
    0.00000005727272
    0.00000000572727
>> semilogy(1:length(s),s)
```

generando il grafico in scala semi-logaritmica in figura che evidentemente sottolinea la convergenza lineare.

# Il metodo QR in Matlab

Una versione di base del metodo QR è la seguente. Si salvi il files `houshess.m` che implementa la trasformazione per similitudine di $A$ in una matrice di Hessenberg

```
function [H,Q]=houshess(A)

% REDUCTION OF A MATRIX TO A SIMILAR HESSENBERG ONE.
% SEE QUARTERONI, SACCO, SALERI P. 192.

n=max(size(A)); Q=eye(n); H=A;

for k=1:(n-2)
    [v,beta]=vhouse(H(k+1:n,k)); I=eye(k); N=zeros(k,n-k);
    m=length(v); R=eye(m)-beta*v*v'; H(k+1:n,k:n)=R*H(k+1:n,k:n);
    H(1:n,k+1:n)=H(1:n,k+1:n)*R; P=[I,N; N',R]; Q=Q*P;
end
```

## Il metodo QR in Matlab

ove `vhouse.m` è definito da

```
function [v,beta]=vhouse(x)

% BUILDING HOUSEHOLDER VECTOR.
% SEE QUARTERONI, SACCO, SALERI P. 197.

n=length(x); x=x/norm(x); s=x(2:n)'*x(2:n); v=[1; x(2:n)];
if (s==0)
    beta=0;
else
    mu=sqrt(x(1)^2+s);
    if (x(1) <= 0)
        v(1)=x(1)-mu;
    else
        v(1)=-s/(x(1)+mu);
    end
    beta=2*v(1)^2/(s+v(1)^2);
    v=v/v(1);
end
```

## Il metodo QR in Matlab

Quindi `QRbasicmethod.m` che calcola la matrice triangolare $T$ relativa a QR

```
function  [T, hist]=QRbasicmethod(T_input,maxit)

% QR METHOD FOR A SYMMETRIC TRIDIAGONAL MATRIX "T_input".

T=T_input;
hist=sort(diag(T));

for index=1:maxit
    [Q,R]=qr(T);
    T=R*Q;                          % NEW SIMILAR MATRIX.
    hist=[hist sort(diag(T))];      % IT STORES THE DIAGONAL ELEMENTS
                                    % OF THE "index" ITERATION.
end
```

## Esercizio

- Data la matrice di Hilbert di ordine 5, ottenibile in Matlab col comando `hilb(5)` si calcolino col metodo delle potenze i suoi minimi e massimi autovalori in modulo.
- Da questi si determini il condizionamento della matrice in norma 2 e lo si confronti con `cond(hilb(5),2)`. Eseguire lo stesso esercizio utilizzando il metodo QR.

# Bibliografia


- K. Atkinson, *Introduction to Numerical Analysis*, Wiley, 1989.
- D. Bini, M. Capovani e O. Menchi, *Metodi numerici per l'algebra lineare*, Zanichelli, 1988.
- V. Comincioli, *Analisi Numerica, metodi modelli applicazioni*, Mc Graw-Hill, 1990.
- G.H. Golub e C.F. Van Loan, *Matrix Computation, 3rd Edition*, The John Hopkins University Press 1996.
- Netlib,http://www.netlib.org/templates/matlab/
- A. Quarteroni, R. Sacco, F. Saleri, *Matematica numerica*, 2001.
- A. Quarteroni e F. Saleri, *Introduzione al calcolo scientifico*, Springer Verlag, 2006.
- Wikipedia (Matrice irriducibile) http://it.wikipedia.org/wiki/Matrice_irriducibile
- Wikipedia (Inverse iteration) http://en.wikipedia.org/wiki/Inverse_iteration.
- Wikipedia (Metodo delle potenze) http://it.wikipedia.org/wiki/Metodo_delle_potenze.
- Wikipedia (PageRank) http://it.wikipedia.org/wiki/Page_rank.
- Wikipedia (Rayleigh quotient) http://en.wikipedia.org/wiki/Rayleigh_quotient.
- Wikipedia (QR algorithm) http://en.wikipedia.org/wiki/QR_algorithm.
- Wikipedia (QR decomposition) http://en.wikipedia.org/wiki/QR_decomposition.
- Wikipedia (Teoremi di Gershgorin) http://it.wikipedia.org/wiki/Teoremi_di_Gershgorin.