



UNIVERSITÀ DI PISA

Seminario di Calcolo Scientifico

Il Metodo delle Iterazioni dei Sottospazi

Ivan Bioli

Università di Pisa

22 Marzo 2021

Setting e assunzioni

Analizzeremo varie varianti del *Metodo delle Iterazioni dei Sottospazi* nelle seguenti ipotesi:

- $A \in \mathbb{C}^{n \times n}$ matrice simmetrica reale o hermitiana
- autovalori $\lambda_1, \lambda_2, \dots, \lambda_n$ ordinati in ordine decrescente di modulo
- relativi autovettori u_1, \dots, u_n ortonormali
- p un intero $1 < p \ll n$, vogliamo calcolare i p autovalori dominanti nell'ipotesi che questi siano separati dal resto dello spettro:

$$|\lambda_1| \geq \dots \geq |\lambda_p| > |\lambda_{p+1}| \geq \dots \geq |\lambda_n|$$

Metodo delle Iterazioni Ortogonali

Algorithm Iterazioni ortogonali

- 1: Sia $X \in \mathbb{C}^{n \times p}$ matrice con colonne ortonormali, $X^*X = I_p$
 - 2: $X^{(0)} := X, k = 0$
 - 3: **repeat**
 - 4: $k := k + 1$
 - 5: $Z^{(k)} := AX^{(k-1)}$
 - 6: $Z^{(k)} =: Q^{(k)}R^{(k)}$ /*Fattorizzazione QR di $Z^{(k)}$ */
 - 7: $X^{(k)} = Q^{(k)}$
 - 8: $\lambda^{(k)} := \text{diag}(R^{(k)})$
 - 9: **until** $\|\lambda^{(k)} - \lambda^{(k-1)}\| / \|\lambda^{(k)}\| < tol$ **or** $k > itmax$
-

Analisi teorica della velocità di convergenza

Teorema (Convergenza dei sottospazi)

Sia $U_p := [u_1, \dots, u_p]$ e sia $X^{(0)} \in \mathbb{C}^{n \times p}$ tale che $W_p := U_p^* X^{(0)}$ sia non singolare. Se $|\lambda_p| > |\lambda_{p+1}|$ vale:

$$\tan \theta^{(k)} \leq \left| \frac{\lambda_{p+1}}{\lambda_p} \right|^k \tan \theta^{(0)}, \quad \theta^{(k)} = \angle(\mathcal{R}(X^{(k)}), \mathcal{R}(U_p))$$

Corollario

Supponiamo che $|\lambda_{j-1}| > |\lambda_j| > |\lambda_{j+1}|$ e che W_j sia non singolare. Allora:

$$\sin \angle(x_j^{(k)}, u_j) \leq c \cdot \max \left\{ \left| \frac{\lambda_j}{\lambda_{j-1}} \right|^k, \left| \frac{\lambda_{j+1}}{\lambda_j} \right|^k \right\}$$

Analisi teorica della velocità di convergenza

Teorema (Convergenza degli autovalori)

Supponiamo che

$$|\lambda_1| > \cdots > |\lambda_p| > |\lambda_{p+1}|$$

Allora per gli elementi diagonali della matrice $R^{(k)}$ vale la stima:

$$|r_{ii}^{(k)} - \lambda_i| = O\left(\left|\frac{\lambda_{i+1}}{\lambda_i}\right|^k + \left|\frac{\lambda_i}{\lambda_{i-1}}\right|^k\right)$$

Perché cercare di accelerare l'algoritmo?

Dobbiamo cercare di accelerare l'algoritmo perché:

- la velocità di convergenza dipende dai rapporti $|\lambda_{i+1}/\lambda_i| \implies$ molto lenta potenzialmente
- la convergenza delle autocopie non è assicurata

Perché cercare di accelerare l'algoritmo?

Dobbiamo cercare di accelerare l'algoritmo perché:

- la velocità di convergenza dipende dai rapporti $|\lambda_{i+1}/\lambda_i| \implies$ molto lenta potenzialmente
- la convergenza delle autocopie non è assicurata

Perché cercare di accelerare l'algoritmo?

Dobbiamo cercare di accelerare l'algoritmo perché:

- la velocità di convergenza dipende dai rapporti $|\lambda_{i+1}/\lambda_i| \implies$ molto lenta potenzialmente
- la convergenza delle autocopie non è assicurata

Perché cercare di accelerare l'algoritmo?

Dobbiamo cercare di accelerare l'algoritmo perché:

- la velocità di convergenza dipende dai rapporti $|\lambda_{i+1}/\lambda_i| \implies$ molto lenta potenzialmente
- la convergenza delle autocopie non è assicurata

Supporremo d'ora in poi che A sia una matrice simmetrica reale o hermitiana definita positiva

Perché cercare di accelerare l'algoritmo?

Dobbiamo cercare di accelerare l'algoritmo perché:

- la velocità di convergenza dipende dai rapporti $|\lambda_{i+1}/\lambda_i| \Rightarrow$ molto lenta potenzialmente
- la convergenza delle autocopie non è assicurata

Supporremo d'ora in poi che A sia una matrice simmetrica reale o hermitiana definita positiva

Teorema

Sia $X^{(0)}$ come nel Teorema 1 e siano $u_i, 1 \leq i \leq p$ gli autovettori corrispondenti ai p autovalori dominanti $\lambda_1, \dots, \lambda_p$ di A . Allora:

$$\min_{x \in \mathcal{R}(X^{(k)})} \sin \angle(u_i, x) \leq c \left(\frac{\lambda_{p+1}}{\lambda_i} \right)^k$$

Dimostrazione

Per induzione

$$A^k X^{(0)} = X^{(k)} R, \quad R = R^{(k)} R^{(k-1)} \dots R^{(1)}$$

Sia $U \in \mathbb{C}^{n \times n}$ matrice unitaria, tale che $U^* A U = \text{diag}(\lambda_1, \dots, \lambda_n) =: \Lambda$.

$$\Lambda = \text{diag}(\Lambda_1, \Lambda_2), \quad U^* X^{(k)} =: \hat{X}^{(k)} = \begin{bmatrix} \hat{X}_1^{(k)} \\ \hat{X}_2^{(k)} \end{bmatrix} \quad \Lambda_1, \hat{X}_1^{(k)} \in \mathbb{C}^{p \times p}$$

$$X^{(k)} R = A^k X^{(0)} = U \Lambda^k U^* X^{(0)} = U \begin{bmatrix} \Lambda_1^k \hat{X}_1^{(0)} \\ \Lambda_2^k \hat{X}_2^{(0)} \end{bmatrix} = U \begin{bmatrix} I_p \\ S^{(k)} \end{bmatrix} \Lambda_1^k \hat{X}_1^{(0)}$$

dove $S^{(k)} = \Lambda_2^k \hat{X}_2^{(0)} \hat{X}_1^{(0)-1} \Lambda_1^{-k} \in \mathbb{C}^{(n-p) \times p}$ e vale

$$s_{ij}^{(k)} = s_{ij} \left(\frac{\lambda_{p+i}}{\lambda_j} \right)^k, \quad s_{ij} := s_{ij}^{(0)} \quad 1 \leq i \leq n-p, \quad 1 \leq j \leq p$$

$$\text{Dunque } \mathcal{R}(X^{(k)}) = \mathcal{R} \left(U \begin{bmatrix} I_p \\ S^{(k)} \end{bmatrix} \right)$$

Dimostrazione

$$\min_{x \in \mathcal{R}(X^{(k)})} \sin \angle(u_i, x) \leq \sin \angle \left(u_i, U \begin{pmatrix} I_p \\ S^{(k)} \end{pmatrix} e_i \right) =$$

$$= \left\| (I - u_i u_i^*) U \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ s_{1i}(\lambda_{p+1}/\lambda_i)^k \\ \vdots \\ s_{n-p,i}(\lambda_n/\lambda_i)^k \end{bmatrix} \right\| / \left\| \begin{pmatrix} I_p \\ S^{(k)} \end{pmatrix} e_i \right\| \leq$$

$$\leq \left\| (I - u_i u_i^*) \left(u_i + \sum_{j=p+1}^n s_{j-p,i} \left(\frac{\lambda_j}{\lambda_i} \right)^k u_j \right) \right\| = \sqrt{\sum_{j=1}^{n-p} s_{ji}^2 \frac{\lambda_{p+j}^{2k}}{\lambda_i^{2k}}} \leq \left(\frac{\lambda_{p+1}}{\lambda_i} \right)^k \sqrt{\sum_{j=1}^{n-p} s_{ji}^2}$$



Accelerazione di Rayleigh-Ritz

IDEA: Fare un cambio di base. Consideriamo la restrizione di A al sottospazio $\mathcal{R}(Q^{(k)})$

$$H^{(k)} := Q^{(k)*} A Q^{(k)} \in \mathbb{C}^{p \times p} \quad (\text{quoziente di Rayleigh generalizzato})$$

Accelerazione di Rayleigh-Ritz

Algorithm Iterazioni ortogonali con accelerazione di Rayleigh-Ritz, versione 1

- 1: Sia $X \in \mathbb{C}^{n \times p}$ matrice con colonne ortonormali, $X^*X = I_p$
 - 2: $X^{(0)} := X, k = 0$
 - 3: **repeat**
 - 4: $k := k + 1$
 - 5: $Z^{(k)} := AX^{(k-1)}$
 - 6: $Z^{(k)} =: Q^{(k)}R^{(k)}$ /*Fattorizzazione QR di $Z^{(k)*}$ */
 - 7: $H^{(k)} := Q^{(k)*}AQ^{(k)}$
 - 8: $H^{(k)} =: F^{(k)}D^{(k)}F^{(k)*}$ /*Decomposizione spettrale di $H^{(k)*}$ */
 - 9: $X^{(k)} = Q^{(k)}F^{(k)}$
 - 10: $\lambda^{(k)} := \text{diag}(D^{(k)})$
 - 11: **until** $\|\lambda^{(k)} - \lambda^{(k-1)}\| / \|\lambda^{(k)}\| < \text{tol}$ **or** $k > \text{itmax}$
-

Limiti della prima versione

PROBLEMA: Il costo del quoziente di Rayleigh generalizzato è troppo alto

Iterazione Algoritmo 1

$$Z^{(k)} := AX^{(k-1)}$$

p chiamate di OP

$$Z^{(k)} =: Q^{(k)}R^{(k)}$$

$p(p+1)n$ operaz.

$$X^{(k)} = Q^{(k)}$$

0 operaz.

Iterazione Algoritmo 2

$$Z^{(k)} := AX^{(k-1)}$$

p chiamate di OP, $Z^{(k)}$ sovrascrive $X^{(k)}$

$$Z^{(k)} =: Q^{(k)}R^{(k)}$$

$p(p+1)n$ operaz., $Q^{(k)}$ sovrascrive $Z^{(k)}$

$$H^{(k)} := Q^{(k)*}(AQ^{(k)})$$

p chiamate di OP, $\frac{1}{2}p(p+1)n$ operaz.

$$H^{(k)} =: F^{(k)}D^{(k)}F^{(k)*}$$

kp^3 operaz. (k dipende dal metodo)

$$X^{(k)} = Q^{(k)}F^{(k)}$$

kp^2n operaz. $X^{(k)}$ sovrascrive $Q^{(k)}$

OP: costo della moltiplicazione di un vettore per A

Seconda versione

Cerchiamo di calcolare implicitamente il quoziente di Rayleigh generalizzato, scrivendo $X^{(k)}$ nella forma:

$$X^{(k)} = Z^{(k)} G^{(k)}, \quad G^{(k)} \in \mathbb{C}^{p \times p} \text{ non singolare}$$

richiedendo:

$$G^{(k)*} Z^{(k)*} Z^{(k)} G^{(k)} = I_p, \quad \left(Z^{(k)} G^{(k)} \right)^* A^{-2} (Z^{(k)} G^{(k)}) = \Delta^{(k)-2}$$

Sostituendo si ottiene:

$$\left(A X^{(k-1)} G^{(k)} \right)^* A^{-2} (A X^{(k-1)} G^{(k)}) = G^{(k)*} G^{(k)} = \Delta^{(k)-2}$$

Dunque la matrice $Y^{(k)} := G^{(k)} \Delta^{(k)}$ è ortogonale e:

$$Y^{(k)*} Z^{(k)*} Z^{(k)} Y^{(k)} = \Delta^{(k)2}$$

Possiamo quindi ricavare $X^{(k)}$ come:

$$X^{(k)} = Z^{(k)} G^{(k)} = Z^{(k)} Y^{(k)} \Delta^{(k)-1}$$

Algorithm Iterazioni ortogonali con accelerazione di Rayleigh-Ritz, versione 2

- 1: Sia $X \in \mathbb{C}^{n \times p}$ matrice con colonne ortonormali, $X^*X = I_p$
 - 2: $X^{(0)} := X, k = 0$
 - 3: **repeat**
 - 4: $k := k + 1$
 - 5: $Z^{(k)} := AX^{(k-1)}$
 - 6: $\hat{H}^{(k)} := Z^{(k)*} Z^{(k)}$
 - 7: $\hat{H}^{(k)} =: Y^{(k)} \Delta^{(k)^2} Y^{(k)*}$ /*Decomposizione spettrale di $\hat{H}^{(k)*}$ /*
 - 8: $X^{(k)} = Z^{(k)} Y^{(k)} \Delta^{(k)^{-1}}$ /* $= Z^{(k)} G^{(k)*}$ */
 - 9: $\lambda^{(k)} := \text{diag}(\Delta^{(k)})$
 - 10: **until** $\|\lambda^{(k)} - \lambda^{(k-1)}\| / \|\lambda^{(k)}\| < \text{tol}$ **or** $k > \text{itmax}$
-

Algoritmo ritzritz

Algorithm Iterazioni ortogonali con accelerazione di Rayleigh-Ritz, versione 3 (ritzritz)

- 1: Sia $X \in \mathbb{C}^{n \times p}$ matrice con colonne ortonormali, $X^*X = I_p$
 - 2: $X^{(0)} := X, k = 0$
 - 3: **repeat**
 - 4: $k := k + 1$
 - 5: $Z^{(k)} := AX^{(k-1)}$
 - 6: $Z^{(k)} =: Q^{(k)}R^{(k)}$ /*Fattorizzazione QR di $Z^{(k)}$ */
 - 7: $H^{(k)} := R^{(k)}R^{(k)*}$
 - 8: $H^{(k)} =: P^{(k)}\Delta^{(k)^2}P^{(k)*}$ /*Decomposizione spettrale di $H^{(k)}$ */
 - 9: $X^{(k)} = Q^{(k)}P^{(k)}$
 - 10: $\lambda^{(k)} := \text{diag}(\Delta^{(k)})$
 - 11: **until** $\|\lambda^{(k)} - \lambda^{(k-1)}\| / \|\lambda^{(k)}\| < \text{tol}$ **or** $k > \text{itmax}$
-

Velocità di convergenza: autovettori

Teorema

Supponiamo valgano le ipotesi del Teorema 1. Sia $x_i^{(k)} = X^{(k)} e_i$ e sia $y_i^{(k)} = U \begin{pmatrix} I_p \\ S^{(k)} \end{pmatrix} e_i$ (cfr. dimostrazione del Teorema 2). Allora:

$$\sin \angle(x_i^{(k)}, y_i^{(k)}) \leq c \left(\frac{\lambda_{p+1}}{\lambda_i} \right)^k, \quad 1 \leq i \leq p$$

Usando la dimostrazione vista in precedenza:

Corollario

$$\sin \angle(x_i^{(k)}, u_i) \leq c_3 \left(\frac{\lambda_{p+1}}{\lambda_i} \right)^k$$

Velocità di convergenza: autovalori

- Algoritmo 2:

$$|\lambda_i^{(k)} - \lambda_i| = O\left(\left|\frac{\lambda_{p+1}}{\lambda_i}\right|^k\right)$$

- Algoritmi 3 e 4, se A è simmetrica definita positiva:

$$\lambda_j^{(k+1)^2} = \frac{\|Ax_j^{(k)}\|^2}{\|x_j^{(k)}\|^2} = x_j^{(k)*} A^2 x_j^{(k)}$$

e vale:

$$|\lambda_j^{(k+1)} - \lambda_j| = O\left(\frac{\lambda_{p+1}}{\lambda_i}\right)^{2k}$$

Velocità di convergenza: autovalori

- Algoritmo 2:

$$|\lambda_i^{(k)} - \lambda_i| = O\left(\left|\frac{\lambda_{p+1}}{\lambda_i}\right|^k\right)$$

- Algoritmi 3 e 4, se A è simmetrica definita positiva:

$$\lambda_j^{(k+1)^2} = \frac{\|Ax_j^{(k)}\|^2}{\|x_j^{(k)}\|^2} = x_j^{(k)*} A^2 x_j^{(k)}$$

e vale:

$$|\lambda_j^{(k+1)} - \lambda_j| = O\left(\frac{\lambda_{p+1}}{\lambda_i}\right)^{2k}$$

Sperimentazione numerica: confronto tra algoritmi

I quattro algoritmi sono stati confrontati, al variare di n e p , in termini di:

- `it`: numero di iterazioni svolte per ciascuna istanza;
- `time`: tempo di risoluzione di un'istanza, in secondi;
- `it_time`: tempo di esecuzione di un'iterazione, in secondi;
- `values_err`: errore relativo massimo commesso sull'approssimazione degli autovalori (assumendo come esatti quelli calcolati da MATLAB)
- `vectors_err`: errore relativo in norma massimo commesso sull'approssimazione degli autovettori (assumendo come esatti quelli calcolati da MATLAB)

Sperimentazione numerica: risultati sperimentali

É stata eseguita una sperimentazione nel caso di A matrice sparsa per i seguenti valori di p e n :

- $n = 1000$, $p = 5$ [► Risultati](#)
- $n = 1000$, $p = 10$ [► Risultati](#)
- $n = 1000$, $p = 15$ [► Risultati](#)
- $n = 1000$, $p = 20$ [► Risultati](#)
- $n = 5000$, $p = 5$ [► Risultati](#)
- $n = 10000$, $p = 5$ [► Risultati](#)

Sperimentazione numerica: risultati sperimentali

È stata eseguita una sperimentazione nel caso di A matrice sparsa per i seguenti valori di p e n :

- $n = 1000$, $p = 5$ [► Risultati](#)
- $n = 1000$, $p = 10$ [► Risultati](#)
- $n = 1000$, $p = 15$ [► Risultati](#)
- $n = 1000$, $p = 20$ [► Risultati](#)
- $n = 5000$, $p = 5$ [► Risultati](#)
- $n = 10000$, $p = 5$ [► Risultati](#)

Sono stati ricavati interessanti risultati interessanti risultati:

- confronto con l'analisi teorica del costo e della convergenza

Sperimentazione numerica: risultati sperimentali

È stata eseguita una sperimentazione nel caso di A matrice sparsa per i seguenti valori di p e n :

- $n = 1000$, $p = 5$ [► Risultati](#)
- $n = 1000$, $p = 10$ [► Risultati](#)
- $n = 1000$, $p = 15$ [► Risultati](#)
- $n = 1000$, $p = 20$ [► Risultati](#)
- $n = 5000$, $p = 5$ [► Risultati](#)
- $n = 10000$, $p = 5$ [► Risultati](#)

Sono stati ricavati interessanti risultati interessanti risultati:

- confronto con l'analisi teorica del costo e della convergenza
- comportamento al crescere di p

Sperimentazione numerica: risultati sperimentali

É stata eseguita una sperimentazione nel caso di A matrice sparsa per i seguenti valori di p e n :

- $n = 1000$, $p = 5$ [► Risultati](#)
- $n = 1000$, $p = 10$ [► Risultati](#)
- $n = 1000$, $p = 15$ [► Risultati](#)
- $n = 1000$, $p = 20$ [► Risultati](#)
- $n = 5000$, $p = 5$ [► Risultati](#)
- $n = 10000$, $p = 5$ [► Risultati](#)

Sono stati ricavati interessanti risultati interessanti risultati:

- confronto con l'analisi teorica del costo e della convergenza
- comportamento al crescere di p
- differenze tra A simmetrica qualsiasi o definita positiva

Sperimentazione numerica: risultati sperimentali

È stata eseguita una sperimentazione nel caso di A matrice sparsa per i seguenti valori di p e n :

- $n = 1000$, $p = 5$ [► Risultati](#)
- $n = 1000$, $p = 10$ [► Risultati](#)
- $n = 1000$, $p = 15$ [► Risultati](#)
- $n = 1000$, $p = 20$ [► Risultati](#)
- $n = 5000$, $p = 5$ [► Risultati](#)
- $n = 10000$, $p = 5$ [► Risultati](#)

Sono stati ricavati interessanti risultati interessanti risultati:

- confronto con l'analisi teorica del costo e della convergenza
- comportamento al crescere di p
- differenze tra A simmetrica qualsiasi o definita positiva
- comportamento al crescere di n

Tabella: $n = 1000$, $p = 5$, $\text{itmax} = 5000$, $\text{tol} = 10^{-10}$

Algoritmo	it	time	it_time	values_err	vectors_err
1	3846	0.996	2.580×10^{-4}	1.857×10^{-6}	0.030
2	3287	1.805	5.482×10^{-4}	5.545×10^{-4}	5.251×10^{-4}
3	2765	0.795	2.853×10^{-4}	8.536×10^{-4}	7.385×10^{-4}
4	2765	0.942	3.402×10^{-4}	8.536×10^{-4}	7.385×10^{-4}

Tabella: $n = 1000$, A def. positiva, $p = 5$, $itmax = 5000$, $tol = 10^{-10}$

Algoritmo	it	time	it_time	values_err	vectors_err
1	3468	0.915	2.652×10^{-4}	1.242×10^{-6}	0.030
2	1832	1.031	5.653×10^{-4}	1.393×10^{-7}	3.363×10^{-4}
3	1832	0.524	2.872×10^{-4}	1.391×10^{-7}	3.362×10^{-4}
4	1832	0.623	3.437×10^{-4}	1.391×10^{-7}	3.362×10^{-4}

◀ Back

Tabella: $n = 1000$, $p = 10$, $\text{itmax} = 7500$, $\text{tol} = 10^{-10}$

Algoritmo	it	time	it_time	values_err	vectors_err
1	4647	2.363	5.095×10^{-4}	1.124×10^{-5}	0.073
2	3273	3.136	9.595×10^{-4}	1.552×10^{-4}	8.051×10^{-4}
3	2737	1.319	4.824×10^{-4}	2.248×10^{-4}	8.769×10^{-4}
4	2737	1.697	6.199×10^{-4}	2.249×10^{-4}	8.769×10^{-4}

Tabella: $n = 1000$, A def. positiva, $p = 10$, $\text{itmax} = 7500$, $\text{tol} = 10^{-10}$

Algoritmo	it	time	it_time	values_err	vectors_err
1	4613	2.347	5.087×10^{-4}	3.610×10^{-5}	0.186
2	2377	2.325	9.755×10^{-4}	4.866×10^{-8}	1.833×10^{-4}
3	2376	2.161	4.887×10^{-4}	4.874×10^{-8}	1.837×10^{-4}
4	2376	1.501	6.314×10^{-4}	4.873×10^{-8}	1.847×10^{-4}

◀ Back

Tabella: $n = 1000$, $p = 15$, $\text{itmax} = 7500$, $\text{tol} = 10^{-10}$

Algoritmo	it	time	it_time	values_err	vectors_err
1	7087	5.881	8.294×10^{-4}	1.076×10^{-5}	0.146
2	4571	7.071	1.532×10^{-3}	1.554×10^{-4}	1.994×10^{-4}
3	3687	2.712	7.328×10^{-4}	1.412×10^{-4}	3.282×10^{-4}
4	3687	3.747	1.047×10^{-3}	1.411×10^{-4}	3.282×10^{-4}

Tabella: $n = 1000$, A def. positiva, $p = 15$, $\text{itmax} = 7500$, $\text{tol} = 10^{-10}$

Algoritmo	it	time	it_time	values_err	vectors_err
1	6970	5.891	8.465×10^{-4}	0.019	0.173
2	4420	6.961	1.644×10^{-3}	4.328×10^{-7}	0.006
3	4399	2.302	7.537×10^{-4}	4.405×10^{-7}	0.006
4	4385	4.494	1.079×10^{-3}	4.411×10^{-7}	0.006

Tabella: $n = 5000$, $p = 5$, $\text{itmax} = 5000$, $\text{tol} = 10^{-10}$

Algoritmo	it	time	it_time	values_err	vectors_err
1	3336	6.388	1.945×10^{-3}	9.361×10^{-7}	0.033
2	2163	8.214	3.939×10^{-3}	1.266×10^{-4}	0.003
3	1883	3.578	1.971×10^{-3}	1.833×10^{-3}	0.004
4	1883	3.963	2.184×10^{-3}	1.833×10^{-3}	0.004

Tabella: $n = 5000$, A def. positiva, $p = 5$, $\text{itmax} = 5000$, $\text{tol} = 10^{-10}$

Algoritmo	it	time	it_time	values_err	vectors_err
1	2657	4.986	1.947×10^{-3}	3.867×10^{-8}	0.039
2	1729	6.591	3.866×10^{-3}	4.757×10^{-7}	0.003
3	1730	3.248	1.915×10^{-3}	4.735×10^{-7}	0.003
4	1730	3.556	2.101×10^{-3}	4.735×10^{-7}	0.003

Tabella: $n = 10000$, $p = 5$, $\text{itmax} = 5000$, $\text{tol} = 10^{-10}$

Algoritmo	it	time	it_time	values_err	vectors_err
1	4515	10.116	2.229×10^{-3}	1.851×10^{-6}	0.050
2	2167	10.088	4.715×10^{-3}	1.006×10^{-4}	5.528×10^{-4}
3	1527	3.427	2.206×10^{-3}	1.875×10^{-3}	0.003
4	1527	3.925	2.613×10^{-3}	1.859×10^{-3}	0.003

Tabella: $n = 10000$, A def. positiva, $p = 5$, $\text{itmax} = 5000$, $\text{tol} = 10^{-10}$

Algoritmo	it	time	it_time	values_err	vectors_err
1	4417	9.870	2.235×10^{-3}	6.695×10^{-5}	0.374
2	2896	13.422	4.607×10^{-3}	3.575×10^{-8}	0.020
3	2897	6.402	2.221×10^{-3}	3.574×10^{-8}	0.020
4	2897	7.168	2.549×10^{-3}	3.574×10^{-8}	0.020

◀ Back

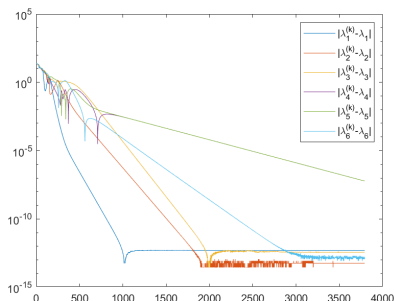
Sperimentazione numerica: confronto con la velocità di convergenza prevista

Possiamo inoltre analizzare graficamente la velocità di convergenza di autovalori e autovettori e in particolare chiederci se:

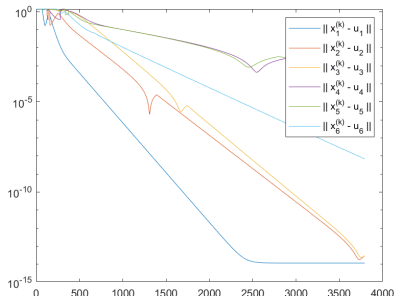
- la convergenza è effettivamente lineare
- il fattore di convergenza è quello ricavato teoricamente

Algoritmo 1

Figura: Algoritmo 1. Plot in scala semilogaritmica dell'errore.

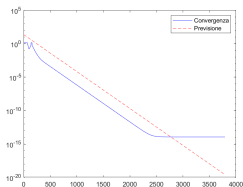


(a) Autovalori

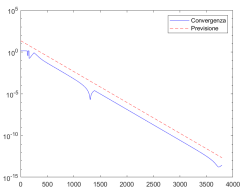


(b) Autovettori

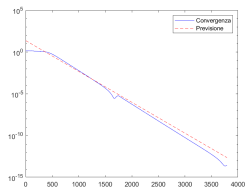
Figura: Algoritmo 1. Confronto, in scala semilogaritmica, tra la velocità di convergenza prevista teoricamente e quella sperimentale per gli autovettori.



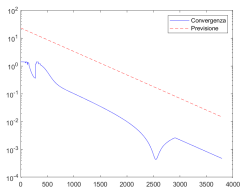
(a) $\|x_1^{(k)} - u_1\|$



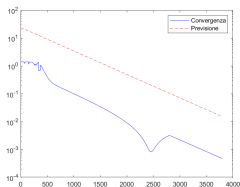
(b) $\|x_2^{(k)} - u_2\|$



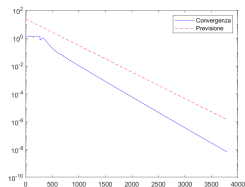
(c) $\|x_3^{(k)} - u_3\|$



(d) $\|x_4^{(k)} - u_4\|$

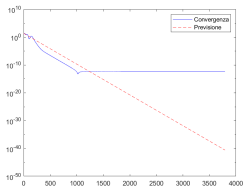


(e) $\|x_5^{(k)} - u_5\|$

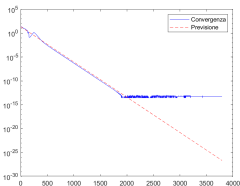


(f) $\|x_6^{(k)} - u_6\|$

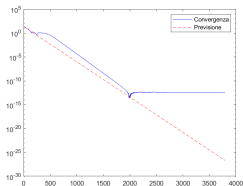
Figura: Algoritmo 1. Confronto, in scala semilogaritmica, tra la velocità di convergenza prevista teoricamente e quella sperimentale per gli autovalori.



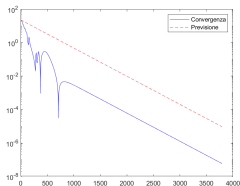
(a) $|\lambda_1^{(k)} - \lambda_1|$



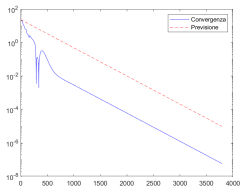
(b) $|\lambda_2^{(k)} - \lambda_2|$



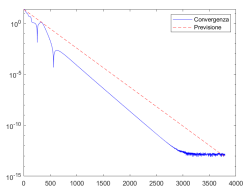
(c) $|\lambda_3^{(k)} - \lambda_3|$



(d) $|\lambda_4^{(k)} - \lambda_4|$



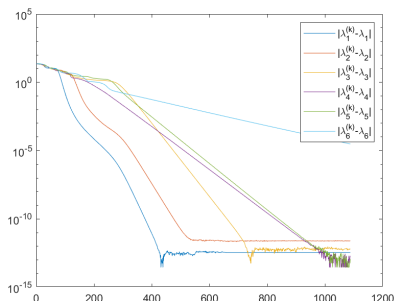
(e) $|\lambda_5^{(k)} - \lambda_5|$



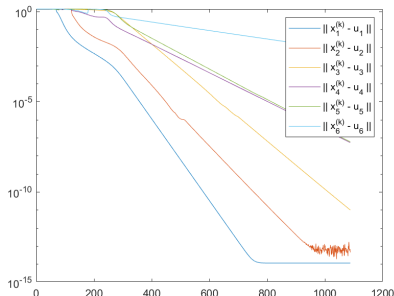
(f) $|\lambda_6^{(k)} - \lambda_6|$

Algoritmo 2

Figura: Algoritmo 2. Plot in scala semilogaritmica dell'errore.

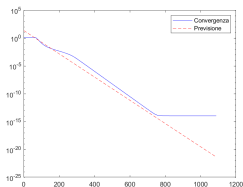


(a) Autovalori

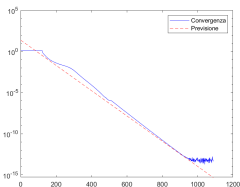


(b) Autovettori

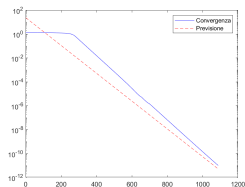
Figura: Algoritmo 2. Confronto, in scala semilogaritmica, tra la velocità di convergenza prevista teoricamente e quella sperimentale per gli autovettori.



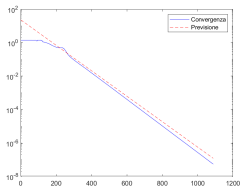
(a) $\|x_1^{(k)} - u_1\|$



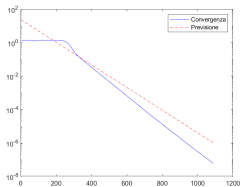
(b) $\|x_2^{(k)} - u_2\|$



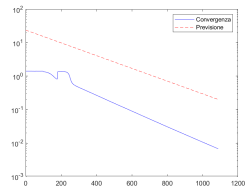
(c) $\|x_3^{(k)} - u_3\|$



(d) $\|x_4^{(k)} - u_4\|$

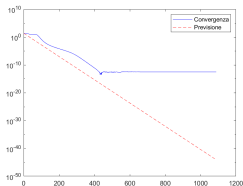


(e) $\|x_5^{(k)} - u_5\|$

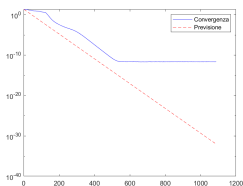


(f) $\|x_6^{(k)} - u_6\|$

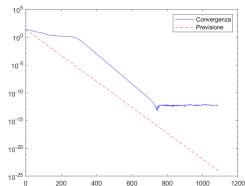
Figura: Algoritmo 2. Confronto, in scala semilogaritmica, tra la velocità di convergenza prevista teoricamente e quella sperimentale per gli autovalori.



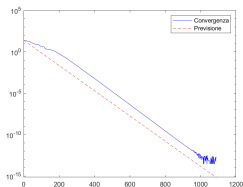
(a) $|\lambda_1^{(k)} - \lambda_1|$



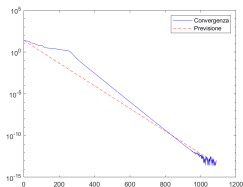
(b) $|\lambda_2^{(k)} - \lambda_2|$



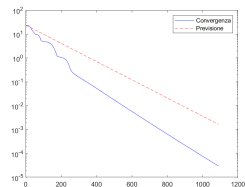
(c) $|\lambda_3^{(k)} - \lambda_3|$



(d) $|\lambda_4^{(k)} - \lambda_4|$



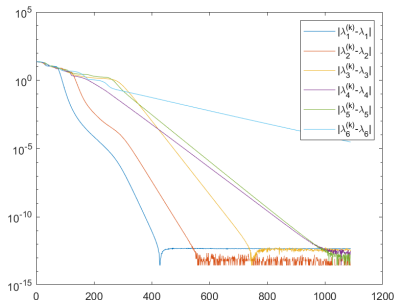
(e) $|\lambda_5^{(k)} - \lambda_5|$



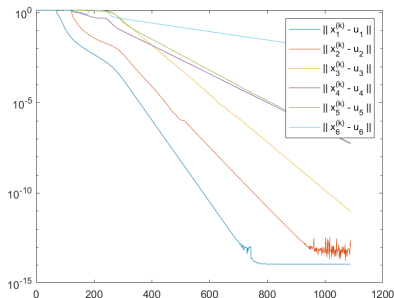
(f) $|\lambda_6^{(k)} - \lambda_6|$

Algoritmo 3

Figura: Algoritmo 3. Plot in scala semilogaritmica dell'errore.

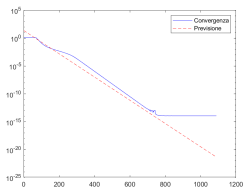


(a) Autovalori

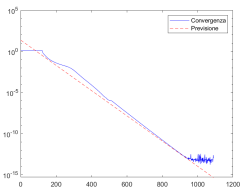


(b) Autovettori

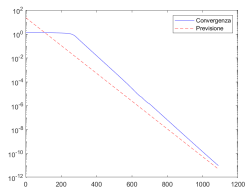
Figura: Algoritmo 3. Confronto, in scala semilogaritmica, tra la velocità di convergenza prevista teoricamente e quella sperimentale per autovettori.



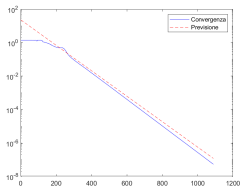
(a) $\|x_1^{(k)} - u_1\|$



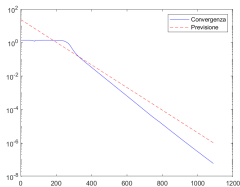
(b) $\|x_2^{(k)} - u_2\|$



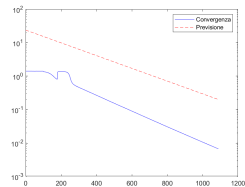
(c) $\|x_3^{(k)} - u_3\|$



(d) $\|x_4^{(k)} - u_4\|$

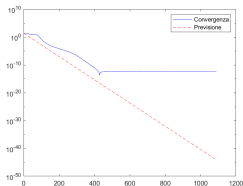


(e) $\|x_5^{(k)} - u_5\|$

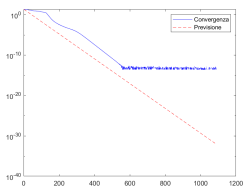


(f) $\|x_6^{(k)} - u_6\|$

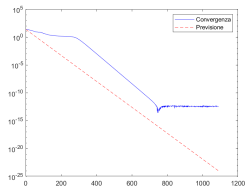
Figura: Algoritmo 3. Confronto, in scala semilogaritmica, tra la velocità di convergenza prevista teoricamente e quella sperimentale per autovalori.



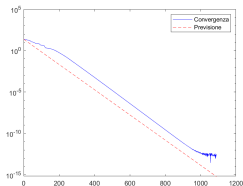
(a) $|\lambda_1^{(k)} - \lambda_1|$



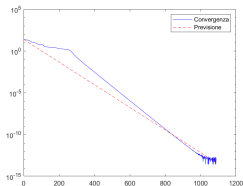
(b) $|\lambda_2^{(k)} - \lambda_2|$



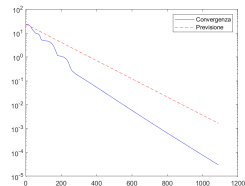
(c) $|\lambda_3^{(k)} - \lambda_3|$



(d) $|\lambda_4^{(k)} - \lambda_4|$



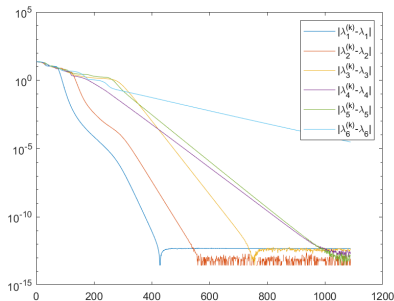
(e) $|\lambda_5^{(k)} - \lambda_5|$



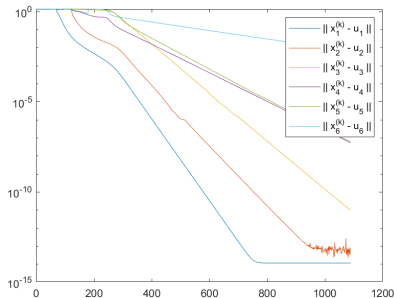
(f) $|\lambda_6^{(k)} - \lambda_6|$

Algoritmo 4

Figura: Algoritmo 4. Plot in scala semilogaritmica dell'errore.

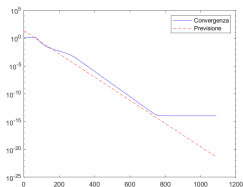


(a) Autovalori

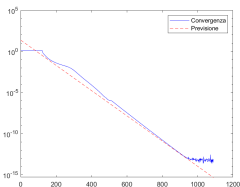


(b) Autovettori

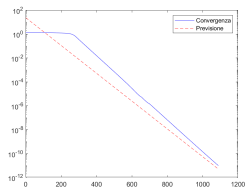
Figura: Algoritmo 4. Confronto, in scala semilogaritmica, tra la velocità di convergenza prevista teoricamente e quella sperimentale per gli autovettori.



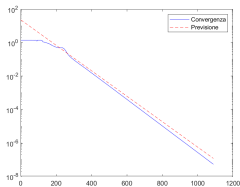
(a) $\|x_1^{(k)} - u_1\|$



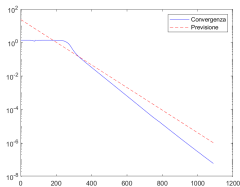
(b) $\|x_2^{(k)} - u_2\|$



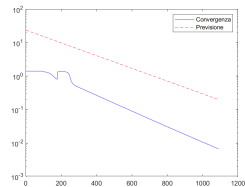
(c) $\|x_3^{(k)} - u_3\|$



(d) $\|x_4^{(k)} - u_4\|$

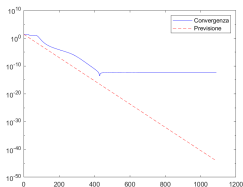


(e) $\|x_5^{(k)} - u_5\|$

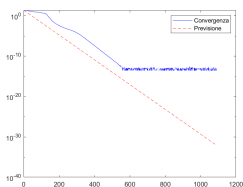


(f) $\|x_6^{(k)} - u_6\|$

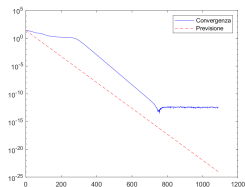
Figura: Algoritmo 4. Confronto, in scala semilogaritmica, tra la velocità di convergenza prevista teoricamente e quella sperimentale per gli autovalori.



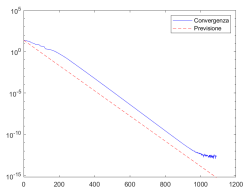
(a) $|\lambda_1^{(k)} - \lambda_1|$



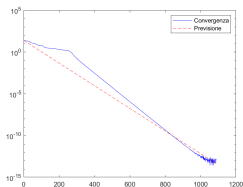
(b) $|\lambda_2^{(k)} - \lambda_2|$



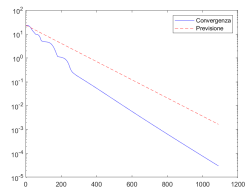
(c) $|\lambda_3^{(k)} - \lambda_3|$



(d) $|\lambda_4^{(k)} - \lambda_4|$



(e) $|\lambda_5^{(k)} - \lambda_5|$



(f) $|\lambda_6^{(k)} - \lambda_6|$

Riferimenti bibliografici

- [1] Peter Arbenz. *Numerical Methods for Solving Large Scale Eigenvalue Problems*. 2016.
- [2] Dario Bini. *Problemi di vibrazioni*. 2020.
- [3] Dario Bini, Milvio Capovani e Ornella Menchi. *Metodi numerici per l'algebra lineare*. Zanichelli, 1988.
- [4] Beresford N. Parlett. «14. Subspace Iteration». In: *The Symmetric Eigenvalue Problem*. Society for Industrial e Applied Mathematics, pp. 323–337.
- [5] H. Rutishauser. «Simultaneous Iteration Method for Symmetric Matrices». In: *Handbook for Automatic Computation: Volume II: Linear Algebra*. Springer Berlin Heidelberg, 1971, pp. 284–302.