

Analisi di un metodo iterativo per sistemi lineari con matrice Toeplitz non simmetrica

Università di Pisa, corso di Calcolo Scientifico
Nikita Deniskin

16 Marzo 2021

Indice

1	Introduzione	2
2	Definizioni base	3
2.1	Matrici Toeplitz, circolanti e Fourier	3
2.2	Precondizionatori circolanti	5
2.3	Algoritmo MINRES standard	7
2.4	Stime sulla convergenza di MINRES	8
3	Algoritmo YMINRES	10
3.1	Versione base	10
3.2	Precondizionamento	11
3.3	Stime sulla velocità di convergenza di YMINRES	13
4	Sperimentazione	16
4.1	Matrici utilizzate	16
4.2	Descrizione della sperimentazione e commenti	18
5	Tabelle e grafici	21
5.1	Tavola 1: blocco di Jordan	21
5.2	Tavola 2: GRCAR	25
5.3	Tavola 3: GRCAR ₀	29
5.4	Tavola 4: Toeplitz(1,1,t)	33
5.5	Tavola 5: Toeplitz([1 1],1,[t t])	38
5.6	Tavola 6: Toeplitz([1 1 1],1,[t t t])	42
5.7	Tavola 7: simbolo $f(x) = x e^{ix}$	46
5.8	Calcolo numerico della stima in proposizione 5	50

1 Introduzione

In questa trattazione analizziamo l'articolo di Pestana e Wathen [1] “*A preconditioned MINRES method for nonsymmetric Toeplitz matrices*”, facendo degli esperimenti numerici sul nuovo metodo iterativo proposto dagli autori.

Data $A \in \mathbb{R}^{n \times n}$ matrice di Toeplitz non simmetrica, si vuole risolvere il sistema lineare

$$Ax = b$$

Nel caso di A simmetrica si può applicare l'algoritmo MINRES; inoltre, per A Toeplitz simmetrica esistono numerosi risultati teorici di convergenza di metodi iterativi. Invece per A Toeplitz non simmetrica c'è meno letteratura, e la teoria non è tanto consolidata. Descriveremo l'algoritmo introdotto da Pestana e Wathen e lo chiameremo YMINRES, per distinguerlo da MINRES “standard”. L'idea è di “capovolgere” la matrice A , trasformandola in una matrice di Hankel, ottenendo un sistema simmetrico risolvibile con MINRES

$$YAx = Yb \tag{1}$$

Nella sezione 2 vengono presentati gli strumenti che ci serviranno, tra cui la convenzione utilizzata per le matrici di Fourier, i tre preconditionatori circolanti che useremo (Strang, Ottimale, Superottimale), una descrizione dell'algoritmo MINRES e infine dei metodi per stimare la velocità di convergenza, tra cui la proposizione 3.

Nella sezione 3 è descritto l'algoritmo YMINRES, per il quale deriviamo alcune stime teoriche per la velocità di convergenza. L'analisi è svolta con particolare attenzione a YMINRES preconditionato, perché alcune formule dell'articolo di riferimento [1] sono errate. Infatti, l'algoritmo genera dei residui r_k in modo che sia minimizzata la norma $\left\| |C|^{-\frac{1}{2}} r_k \right\|_2$ in un certo sottospazio di Krylov, mentre nell'articolo viene indicata $\left\| |C|^{-1} r_k \right\|_2$ come norma da minimizzare. Infine mostriamo la proposizione 5, che dimostra la convergenza del metodo dal punto di vista teorico.

Nella sezione 4 è descritta la sperimentazione numerica svolta con commenti sui risultati, mentre nella sezione 5 sono presenti le tabelle e i grafici risultanti dalla sperimentazione, con commenti su ogni classe di matrice provata. È stata ripercorsa la sperimentazione descritta nell'articolo di riferimento [1], ampliandola sia con altre tipologie di matrici, sia provando più valori di n . Infine nella sezione 5.8 viene calcolata numericamente la stima teorica derivata nella proposizione 5.

2 Definizioni base

2.1 Matrici Toeplitz, circolanti e Fourier

Una matrice di Toeplitz A ha elementi costanti sulle diagonali. L'entrata i, j dipende solo dalla differenza $i - j$, e la indicheremo con una lettera minuscola: $A_{ij} = a_{i-j}$. Le matrici di Toeplitz di dimensioni $n \times n$ hanno $2n - 1$ parametri liberi, uno per diagonale.

$$\begin{pmatrix} a_0 & a_{-1} & a_{-2} & \cdots & a_{-(n-2)} & a_{-(n-1)} \\ & a_1 & a_2 & \cdots & a_{n-2} & \\ & a_2 & & \cdots & & a_{-2} \\ & \vdots & & \cdots & & \vdots \\ a_{n-2} & & & & & a_{-1} \\ a_{n-1} & a_{n-2} & \cdots & a_2 & a_1 & a_0 \end{pmatrix}$$

Una matrice circolante C è una matrice di Toeplitz, con la proprietà aggiuntiva che la k -esima sottodiagonale ha le stesse entrate della $n - k$ -esima sopradiagonale.

$$\begin{pmatrix} c_0 & c_{n-1} & c_{n-2} & \cdots & c_2 & c_1 \\ c_1 & & & \cdots & & c_2 \\ c_2 & & & \cdots & & \vdots \\ \vdots & & & \cdots & & c_{n-2} \\ c_{n-2} & & & & & c_{n-1} \\ c_{n-1} & c_{n-2} & \cdots & c_2 & c_1 & c_0 \end{pmatrix}$$

Sia Q la seguente matrice circolante:

$$Q = \begin{pmatrix} 0 & & & & 1 \\ 1 & 0 & & & \\ & 1 & \ddots & & \\ & & \ddots & \ddots & \\ & & & 1 & 0 \end{pmatrix} \quad (2)$$

Una qualsiasi matrice circolante C si può scrivere come combinazione lineare di potenze di Q :

$$C = \sum_{k=0}^{n-1} c_k Q^k$$

Sia $\omega_n = e^{\frac{2\pi i}{n}}$ la radice n -esima dell'unità. Definiamo le seguenti matrici, utili per rappresentare la trasformata discreta di Fourier:

- Ω . In ciascuna entrata $\Omega_{ij} = \omega_n^{ij}$. Alcune proprietà:
 - $\Omega^H \Omega = n I$.
 - È simmetrica: $\Omega^T = \Omega$.

- $U = \frac{1}{\sqrt{n}}\Omega$.
 - U è unitaria, $U^H U = I$
- $F = \overline{\Omega} = \Omega^H$
 - Entrate: $F_{ij} = \omega_n^{-ij}$.
 - $F = \sqrt{n}U^H$ e $U^H = \frac{1}{\sqrt{n}}F$
 - Fx corrisponde a fare `fft(x)` in MATLAB.
- $F^{-1} = \frac{1}{n}\Omega$.
 - Entrate: $F_{ij}^{-1} = \frac{1}{n}\omega_n^{ij}$
 - $F^{-1} = \frac{1}{\sqrt{n}}U$ e $U = \sqrt{n}F^{-1}$
 - $F^{-1}F = I$
 - $F^{-1}x$ corrisponde a fare `ifft(x)` in MATLAB.

La matrice Q è unitariamente diagonalizzabile, con matrice di similitudine U .

$$Q = U^H S U = F S F^{-1}$$

Dove $S = \text{diag}(1, \omega_n, \omega_n^2, \dots, \omega_n^{n-1})$.

Per una generica matrice circolante C :

$$C = \sum_{k=0}^{n-1} c_k Q^k = U^H \left(\sum_{k=0}^{n-1} c_k S^k \right) U \quad (3)$$

Osserviamo un legame tra la prima colonna di C e i suoi autovalori. Chiamato $e_1 = (1, 0, \dots, 0)^T$ e $w = (1, 1, \dots, 1)^T$ si ha:

$$C(:, 1) = C e_1 = (U^H D U) e_1 = U^H D (U e_1)$$

La prima colonna di U ha tutte entrate $\frac{1}{\sqrt{n}}$, dunque $U e_1 = \frac{1}{\sqrt{n}}w$.

$$C(:, 1) = \frac{1}{\sqrt{n}}U^H D w = \frac{1}{\sqrt{n}}U^H \text{diag}(D) = F^{-1} \text{diag}(D)$$

Questo è un modo veloce per passare dalle entrate di una matrice circolante ai suoi autovalori.

$$C(:, 1) = F^{-1} \text{diag}(D) = \text{ifft}(\text{diag}(D)) \quad \text{diag}(D) = F \cdot C(:, 1) = \text{fft}(C(:, 1)) \quad (4)$$

2.2 Precondizionatori circolanti

Definizione 1. Data $A = X^{-1}DX$ matrice diagonalizzabile, definiamo la funzione valore assoluto di una matrice $|A|$ nel seguente modo:

$$|A| = X^{-1}|D|X \quad (5)$$

■

Questa è una buona definizione (come in generale per le funzioni di matrice), ovvero non dipende dalla scelta di X .

In questa trattazione ci concentreremo su tre preconditionatori circolanti, relativi a matrici di Toeplitz: Strang, Ottimale e Superottimale [3, Capitolo 2].

Definizione 2. Sia A una matrice Toeplitz, con entrate $A_{ij} = a_{i-j}$. Il preconditionatore di Strang C relativo ad A si ottiene mantenendo la diagonali “centrali” di A , e completando le altre in modo da ottenere una matrice circolante.

$$A = \begin{pmatrix} 1 & 16 & 17 & 18 & 19 \\ 2 & 1 & 16 & 17 & 18 \\ 3 & 2 & 1 & 16 & 17 \\ 4 & 3 & 2 & 1 & 16 \\ 5 & 4 & 3 & 2 & 1 \end{pmatrix} \quad C = \begin{pmatrix} 1 & 16 & 17 & 3 & 2 \\ 2 & 1 & 16 & 17 & 3 \\ 3 & 2 & 1 & 16 & 17 \\ 17 & 3 & 2 & 1 & 16 \\ 16 & 17 & 3 & 2 & 1 \end{pmatrix}$$

Preso un entrata $C_{ij} = c_{i-j}$, e $A_{ij} = a_{i-j}$, sia $k = i - j$. Distinguiamo il caso n pari e dispari.

Caso n dispari. $n = 2m + 1$.

$$c_k = \begin{cases} a_{k-n} & k > m & \text{ultime } m \text{ entrate della prima colonna} \\ a_k & -m \leq k \leq m & \text{sono le } 2m + 1 \text{ entrate centrali} \\ a_{k+n} & k < -m & \text{ultime } m \text{ entrate della prima riga} \end{cases} \quad (6)$$

Caso n pari. $n = 2m$.

Qui sorge un piccolo problema per la scelta di c_m : oltre alla diagonale centrale relativa a $c_0 = a_0$, ci sono altre $2m - 1$ diagonali; si dovrebbero quindi mantenere o m sottodiagonali e $m - 1$ sopradiagonali scegliendo $c_m = a_m$ (o viceversa, con $c_m = a_{-m}$). Un modo un po' più “simmetrico” è di porre $c_m = 0$ o $c_m = \frac{1}{2}(a_m + a_{-m})$. Useremo quest'ultimo, perché sperimentalmente è più raro che C sia singolare.

$$c_k = \begin{cases} a_{k-n} & k > m & \text{ultime } m - 1 \text{ entrate della prima colonna} \\ \frac{a_m + a_{-m}}{2} & k = m & \text{entrata a metà colonna} \\ a_k & -m < k < m & \text{sono le } 2m - 1 \text{ entrate centrali} \\ \frac{a_m + a_{-m}}{2} & k = -m & \text{entrata a metà riga} \\ a_{k+n} & k < -m & \text{ultime } m - 1 \text{ entrate della prima riga} \end{cases} \quad (7)$$

■

Definizione 3. Sia A matrice generica, e V una generica matrice unitaria. Definiamo $\mathcal{M}_V = \{V^H \Lambda V \mid \Lambda \text{ è diagonale}\}$. Il preconditionatore ottimale per A è la matrice $C_V(A) \in \mathcal{M}_V$ che risolve il problema di minimo $\min_{W \in \mathcal{M}_V} \|A - W\|_F$, dove $\|\cdot\|_F$ è la norma di Frobenius. ■

Proposizione 1. [3, Teorema 2.7] Si hanno i seguenti risultati per il preconditionatore ottimale.

1. $C_V(A)$ è univocamente determinata da A , ed è uguale a:

$$C_V(A) = V^H \text{diag}(V A V^H) V$$

2. Se A è Hermitiana, allora anche $C_V(A)$ è Hermitiana e vale

$$\lambda_{\min}(A) \leq \lambda_{\min}(C_V(A)) \leq \lambda_{\max}(C_V(A)) \leq \lambda_{\max}(A)$$

In particolare, se A è definita positiva, lo è anche $C_V(A)$.

3. Se $V = U$ matrice unitaria di Fourier, allora \mathcal{M}_U è l'insieme di tutte le matrici circolanti di taglia n .
4. Sempre nel caso $V = U$, $C_U(A)$ si può calcolare nel seguente modo

$$C_U(A) = \sum_{k=0}^{n-1} \left(\frac{1}{n} \sum_{\substack{p-q \equiv k \\ \text{mod } n}} a_{pq} \right) Q^k$$

5. Se A è Toeplitz, allora la formula si può semplificare:

$$C_U(A) = \sum_{k=0}^{n-1} \left(\frac{(n-k)a_k + k a_{k-n}}{n} \right) Q^k$$

Cioè $c_k = \frac{(n-k)a_k + k a_{k-n}}{n}$ per $k > 0$ (quelle negative si ottengono ciclando). ■

Definizione 4. Sia A matrice generica, e V una generica matrice unitaria. Sia \mathcal{M}_V come prima. Il preconditionatore superottimale è la matrice $T_V(A) \in \mathcal{M}_V$ che risolve il problema di minimo $\min_{W \in \mathcal{M}_V} \|I - W^{-1}A\|_F$ ■

Proposizione 2. [3, Teorema 2.12] Sia A matrice non singolare, tale che $C_V(A)$ sia non singolare. Allora il preconditionatore superottimale di A esiste, è unico e si può calcolare come:

$$T_V(A) = C_V(AA^H) (C_V(A^H))^{-1}.$$

Inoltre, $T_V(A)^{-1}$ si può esprimere nel seguente modo:

$$T_V(A)^{-1} = V^H \text{diag}(V A^H V^H) (\text{diag}(V A A^H V^H))^{-1} V.$$

Questo risultato vale per A matrice qualsiasi e V matrice unitaria. La adotteremo utilizzando $V = U$ matrice di Fourier.

Queste espressioni sono utilizzate nei codici `diag_strang`, `diag_opt`, `diag_super` per calcolare $\text{diag}(D^{-1})$, dove il preconditionatore $C = U^H D U$. Viene calcolata D^{-1} e non C o C^{-1} perché l'algoritmo YMINRES usa come preconditionatori $|C| = U^H |D| U$; inoltre dovendo sempre calcolare prodotti della forma $|C|^{-1}v$ o $C^{-1}v$, conviene usare direttamente l'inversa. L'implementazione non calcola esplicitamente i prodotti per le matrici U o U^H , ma tramite le funzioni `fft(·)` e `ifft(·)` di MATLAB.

2.3 Algoritmo MINRES standard

Definizione 5. Indicheremo nel seguente modo il sottospazio di Krylov di dimensione k con matrice A e vettore iniziale r :

$$\mathcal{K}_k(A, r) = \text{span}(r, Ar, A^2r, \dots, A^{k-1}r)$$

Ci sarà utile anche lo sottospazio “traslato di A ”:

$$\mathcal{K}_k^{+1}(A, r) = \mathcal{K}_k(A, Ar) = \text{span}(Ar, A^2r, A^3r, \dots, A^k r)$$

■

Algoritmo MINRES. [2, Capitolo 2.4]

Sia A matrice reale simmetrica, in generale indefinita. L'algoritmo MINRES risolve il sistema $Ax = b$, partendo da un guess x_0 e generando una successione x_k tale che i residui $r_k = b - Ax_k$ soddisfino:

$$\begin{aligned} x_k &\in x_0 + \mathcal{K}_k(A, r_0) \\ r_k &\in r_0 + \mathcal{K}_k^{+1}(A, r_0) \\ \|r_k\|_2 &\leq \|s\|_2 \quad \forall s \in r_0 + \mathcal{K}_k^{+1}(A, r_0) \end{aligned}$$

Ovverosia si sceglie l'iterata x_k in modo che venga minimizzata la norma 2 del residuo r_k tra tutti i vettori del sottospazio $r_0 + \mathcal{K}_k^{+1}(A, r_0)$.

L'algoritmo si può applicare anche usando tecniche di preconditionamento, con un preconditionatore M simmetrico e definito positivo. Descriveremo in dettaglio come avviene il preconditionamento nella sezione 3.2.

Mostriamo di seguito l'algoritmo MINRES preconditionato [2, Algoritmo 6.1]:

```

r(0)=0; w(0)=0; w(1)=0;
Scegliere x(0);
r(1) = b - A * x(0);
Risolvere M * z(1) = r(1);
γ(1) = sqrt( z(1)T * r(1) )
η = γ(1);
s(0)=s(1)=0; c(0)=c(1)=1;
```

```

for j = 1, 2, ... fino a convergenza
  z(j) = z(j) / γ(j);
  δ(j) = z(j)⊤ * A * r(j);
  v(j+1) = A * z(j) - (δ(j) / γ(j)) * v(j) - (γ(j) / γ(j-1)) * v(j-1)

  Risolvere M * z(j+1) = v(j+1);
  γ(j+1) = sqrt( z(j+1)⊤ * r(j+1) )

  a(0) = c(j) * δ(j) - c(j-1) * s(j) * γ(j);
  a(1) = sqrt( a(0)2 + γ(j)2 )
  a(2) = s(j) * δ(j) - c(j-1) * c(j) * γ(j);
  a(3) = s(j-1) * γ(j);
  c(j+1) = a(0) / a(1);
  s(j+1) = γ(j+1) / a(1);

  w(j+1) = ( z(j) - a(3) * w(j-1) - a(2) * w(j) ) / a(1);
  x(j) = x(j-1) + c(j+1) * w(j+1);
  η = - s(j+1) * η;

  Controllare la convergenza
  if
    ||v(j+1)|| ≤ tol * ||b||
    return
  end
end

```

2.4 Stime sulla convergenza di MINRES

Cerchiamo ora una stima su $\|r_k\|_2$. Definiamo il seguente insieme di polinomi:

$$\Pi_k = \{p(x) \text{ polinomio di grado } \leq k \text{ tale che } p(0) = 1\}$$

Ogni $s \in r_0 + \mathcal{K}_k^{+1}(A, r_0)$ si può scrivere come $s = p(A)r_0$ con $p \in \Pi_k$.

$$\|r_k\|_2 = \min_{p \in \Pi_k} \|p(A)r_0\|_2$$

A è simmetrica, dunque ha una base di autovettori ortogonali w_i con autovalori λ_i .

Scriviamo r_0 in questa base: $r_0 = \sum_{i=1}^n \alpha_i w_i$, allora si ha

$$p(A)r_0 = p(A) \sum_{i=1}^n \alpha_i w_i = \sum_{i=1}^n \alpha_i p(A) w_i = \sum_{i=1}^n \alpha_i p(\lambda_i) w_i$$

Calcoliamo la norma 2, usando il fatto che i w_i sono ortonormali, dunque $\|w_i + w_j\|_2 = \|w_i\|_2 + \|w_j\|_2$:

$$\begin{aligned}
\|p(A)r_0\|_2 &= \left\| \sum_{i=1}^n \alpha_i p(\lambda_i) w_i \right\|_2 = \sum_{i=1}^n \|\alpha_i p(\lambda_i) w_i\|_2 \leq \\
&\leq \left(\max_i |p(\lambda_i)| \right) \sum_{i=1}^n \|\alpha_i w_i\|_2 = \left(\max_i |p(\lambda_i)| \right) \|r_0\|_2
\end{aligned} \tag{8}$$

Vale quindi:

$$\|r_k\|_2 \leq \left(\min_{p \in \Pi_k} \max_{\lambda \in \text{Sp}(A)} |p(\lambda)| \right) \cdot \|r_0\|_2$$

Se siamo a conoscenza di informazioni sugli autovalori di A , si possono sfruttare per trovare un buon polinomio $p \in \Pi_k$ in modo che $\max_{\lambda \in \text{Sp}(A)} |p(\lambda)|$ sia piccolo. Per esempio, se si hanno a disposizione tutti gli autovalori di A , il polinomio minimizzante si trova risolvendo un problema di minimi quadrati; tuttavia dal punto di vista computazionale è dispendioso calcolare esattamente gli autovalori di A . Per questo motivo sono state proposte stime basate su una conoscenza approssimata della posizione degli autovalori. Per esempio, se $\text{Sp}(A) \subseteq [a, b]$ (intervallo reale), allora $p(x) = T_k\left(\frac{b+a-2x}{b-a}\right) \cdot T_k\left(\frac{b+a}{b-a}\right)^{-1}$ con T_k polinomio di Chebycheff di prima specie.

Un'interessante generalizzazione della stima precedente è dovuta ad Axelsson-Lindskog [5] originariamente derivata per il gradiente coniugato, ma che presentiamo qui riformulata per GMRES: infatti in entrambe le formulazioni ci si è ricondotti a trovare un buon polinomio minimizzante $p \in \Pi_k$ in funzione di condizioni su $\text{Sp}(A)$.

Per A Hermitiana semidefinita positiva, sotto l'ipotesi

$$\text{Sp}(A) \subseteq \left(\bigcup_{i=1}^p \lambda_i \right) \cup [a, b] \cup \left(\bigcup_{j=1}^q \lambda_j \right) \quad \text{con } \lambda_i < a \text{ e } \lambda_j > b,$$

ovvero che quasi tutto lo spettro di A sia contenuto nell'intervallo $[a, b]$ ($a, b \geq 0$), con l'eccezione di $p + q$ "outliers", si ottiene la seguente stima:

$$\frac{\|r_{k+2\max(p,q)}\|_2}{\|r_0\|_2} \leq M\sigma^k$$

Dove M è un'opportuna costante dipendente dai λ , a, b, p, q e $\sigma = \frac{\sqrt{b}-\sqrt{a}}{\sqrt{b}+\sqrt{a}} < 1$.

Nella nostra trattazione useremo un'altra stima dovuta a Campbell, Ipsen, Kelley e Meyer [4] valida per GMRES (e di conseguenza anche MINRES e YMINRES). Per essa al posto di un intervallo $[a, b]$, si considerano P palle di raggio ε centrate in dei punti $\gamma_h \neq 0$ del piano complesso. Gli autovalori di A si suddividono in $n - d$ contenuti nei P cluster, e d "outliers" fuori da ogni palla. Similmente a prima, possiamo stimare le norme dei residui $\|r_k\|_2$ solo per $k \geq d$.

Proposizione 3. [4, Proposizione 5.1] Dato $\varepsilon > 0$ e $\gamma_1, \dots, \gamma_P \neq 0$ numeri complessi distinti, si considerino le palle di centro γ_h e raggio $\varepsilon\gamma_h$. Si suddividano gli autovalori di A nei seguenti insiemi, a seconda di in quali palle sono contenuti:

$$\Lambda_h = \{\lambda \in \text{Sp}(A) \mid |\lambda - \gamma_h| < \varepsilon\gamma_h\} = \text{Sp}(A) \cap B(\gamma_h, \varepsilon\gamma_h)$$

$$\Lambda_{out} = \Lambda \setminus \bigcup_1^P \Lambda_h = \text{outliers, autovalori fuori da ogni palla}$$

Chiamati r_k i residui di GMRES applicato al sistema $Ax = b$, vale la seguente stima

$$\frac{\|r_{d+Pk}\|_2}{\|r_0\|_2} \leq M(\sigma^{P-1}\varepsilon)^k \quad (9)$$

con le seguenti quantità. δ indica la distanza massima degli outliers dai cluster:

$$\delta = \max_{1 \leq h \leq P} \max_{|z - \gamma_h| = \varepsilon \gamma_h} \max_{\lambda \in \Lambda_{out}} \frac{|\lambda - z|}{|\lambda|}$$

σ indica la distanza massima tra due cluster:

$$\sigma = \max_{1 \leq h \leq P} \max_{|z - \gamma_h| = \varepsilon \gamma_h} \max_{l \neq h} \frac{|\gamma_l - z|}{|\gamma_l|}$$

Infine, chiamato B_h la proiezione di B sulla somma degli autospazi con autovalore in Λ_h , la costante M è pari a:

$$M = P \varepsilon \delta^d \max_{1 \leq h \leq P} \max_{|z - \gamma_h| = \varepsilon \gamma_h} \|(zI - B_h)^{-1}\|_2$$

■

Se ci sono tanti outliers, la stima è valida solo dopo tante iterazioni. Inoltre, se ci sono P cluster, la proposizione dà una riduzione dell'errore da r_k a r_{k+P} di un fattore $\sigma^{P-1}\varepsilon$; questo vuol dire che ad ogni passo, la riduzione dell'errore è mediamente $\sigma^{1-1/P}\varepsilon^{1/P}$.

Nell'espressione di δ compare un λ al denominatore, così come nell'espressione di σ compare un γ_l : questo vuol dire che se ci sono autovalori oppure centri di cerchi vicini a 0, la stima peggiora notevolmente. Infine, nella stima della costante M compare la norma di $(zI - B_h)^{-1}$, che è molto grande se ci sono degli autovalori $\lambda \in \Lambda_h$ vicini al bordo della palla centrata in γ_h . Questo vuol dire che le palle non devono essere scelte troppo "strette" attorno ai cluster di autovalori, per avere una buona stima.

Questa proposizione è efficace quando gli autovalori di A sono molto raggruppati in alcuni cluster, che sono copribili da palle di raggio abbastanza piccolo. Nel nostro caso, vedremo che le matrici preconditionate $|C|^{-1}YA$ hanno autovalori molto raggruppati attorno a 1 e -1 , per cui tale stima sembrerebbe essere efficace nel nostro caso. Tuttavia, come vedremo nella sezione 5.8 la stima numerica della convergenza del metodo non è tanto buona: la costante M può arrivare fino a 10^{100} , mentre il fattore di riduzione $\sigma^{P-1}\varepsilon$ è appena dell'ordine $10^{-1}, 10^{-2}$. Tuttavia questa proposizione ci permette comunque di concludere dal punto di vista teorico e dimostrare la convergenza dell'algoritmo YMINRES.

3 Algoritmo YMINRES

3.1 Versione base

In questa sezione descriviamo l'algoritmo introdotto da Pestana e Wathen nell'articolo di riferimento [1].

Sia A una matrice Toeplitz di dimensioni $n \times n$ a coefficienti reali, in generale non simmetrica. Si vuole risolvere il seguente sistema lineare

$$Ax = b.$$

Sia Y la seguente matrice:

$$Y = \begin{pmatrix} & & & & 1 \\ & & & & \dots \\ & & & & \dots \\ & & & & \dots \\ 1 & \dots & \dots & \dots & \dots \end{pmatrix}$$

Dato v vettore generico, Yv è il ribaltato di v : la prima componente è scambiata con l'ultima, la seconda con la penultima e così via. Non è necessario svolgere nessun calcolo per ottenere Yv . Analogamente, moltiplicare una matrice A per Y equivale a capovolgere la matrice, senza doverle moltiplicare esplicitamente. Nel nostro caso, YA è una matrice di Hankel, quindi è simmetrica. Questo vuol dire che il sistema

$$(YA)x = Yb \tag{10}$$

può essere risolto con l'applicazione del metodo MINRES alla matrice YA . Si noti che non è necessario calcolare esplicitamente tale matrice, ma ogni volta che bisogna moltiplicare YA per un vettore v , si calcola Av e poi lo si capovolge.

Nel corso della trattazione chiameremo questo algoritmo YMINRES. Come vedremo successivamente nella sperimentazione, in questa forma non è molto efficiente, ma le sue prestazioni possono essere aumentate notevolmente utilizzando delle tecniche di pre-condizionamento. Ci concentreremo sui seguenti preconditionatori circolanti: Strang, Ottimale e Superottimale.

3.2 Precondizionamento

Data C matrice circolante, si può diagonalizzare $C = U^H D U$ con U matrice di Fourier. Utilizzeremo come preconditionatore $|C| = U^H |D| U$, che resta comunque una matrice circolante. Inoltre è simmetrica e semidefinita positiva, ma d'ora in poi assumeremo che sia definita positiva (sperimentalmente nei casi analizzati lo è quasi sempre).

Cerchiamo una stima della convergenza con preconditionatore $|C|$, seguendo l'approccio presentato in [2, Capitolo 6]. Essendo $|C|$ simmetrica e definita positiva, possiamo fattorizzarla nel seguente modo:

$$|C| = LL^\top \quad \text{con } L \in \mathbb{R}^{n \times n}$$

D'ora in poi tutti i vettori e matrici considerate saranno a coefficienti reali. A e b lo sono per ipotesi, mentre possiamo prendere $L = U^H \sqrt{|D|} U = |C|^{\frac{1}{2}}$, che ha coefficienti reali. Il sistema 10 è equivalente a

$$(L^{-1}YAL^{-\top})(L^\top x) = L^{-1}Yb \tag{11}$$

Chiamiamo $\tilde{A} = L^{-1}YAL^{-\top}$, $\tilde{x} = L^\top x$, e $\tilde{b} = L^{-1}Yb$.

La matrice \tilde{A} è congruente a YA , dunque è simmetrica e si può applicare l'algoritmo MINRES al sistema

$$\tilde{A}\tilde{x} = \tilde{b}.$$

Partendo da un vettore iniziale \tilde{x}_0 , denotiamo con \tilde{x}_k la successione che si otterrebbe applicando MINRES a [11](#); nella pratica \tilde{x}_k non verrà calcolata esplicitamente, (come neanche la matrice L). Indichiamo la successione realmente calcolata x_k , che è legata alla precedente dalla relazione $L^\top x_k = \tilde{x}_k$.

L'algoritmo MINRES applicato a [11](#), oltre a \tilde{x}_k , genererebbe una successione di residui $\tilde{r}_k = \tilde{b} - \tilde{A}\tilde{x}_k$, in modo che il k -esimo residuo sia la soluzione al seguente problema di minimo:

$$\tilde{r}_k = \operatorname{argmin}\{ \|v\|_2 \mid v \in \tilde{r}_0 + \mathcal{K}_k^{+1}(\tilde{A}, \tilde{r}_0) \} \quad (12)$$

Possiamo riscrivere tale spazio tramite A, C, r_0 :

$$\begin{aligned} \tilde{r}_0 + \mathcal{K}_k^{+1}(\tilde{A}, \tilde{r}_0) &= L^{-1}Y r_0 + \mathcal{K}_k^{+1}(L^{-1}YAL^{-H}, L^{-1}r_0) = \\ &= L^{-1}Y (r_0 + \mathcal{K}_k^{+1}(A|C|^{-1}Y, r_0)) \end{aligned} \quad (13)$$

Ricordiamo che \tilde{x}_k e \tilde{r}_k non vengono esplicitamente calcolati nell'algoritmo MINRES preconditionato, bensì vengono calcolati x_k, r_k e $|C|^{-1}r_k$, con le seguenti relazioni:

$$\begin{aligned} \tilde{x}_k &= L^\top x_k \\ \tilde{r}_k &= \tilde{b} - \tilde{A}\tilde{x}_k = L^{-1}Yb - L^{-1}YAL^{-\top}L^\top x_k = L^{-1}Y(b - Ax_k) = L^{-1}Yr_k \end{aligned}$$

Riscriviamo la condizione [12](#) in termini di r_k :

$$\begin{aligned} r_k &= YL\tilde{r}_k = YL \cdot \operatorname{argmin}\{ \|v\|_2 \mid v \in \tilde{r}_0 + \mathcal{K}_k^{+1}(\tilde{A}, \tilde{r}_0) \} = \\ &= YL \cdot \operatorname{argmin}\{ \|v\|_2 \mid v \in L^{-1}Y (r_0 + \mathcal{K}_k^{+1}(A|C|^{-1}Y, r_0)) \} = \\ &\quad \text{cambio di variabile } w = YLv \\ &= \operatorname{argmin}\{ \|L^{-1}Yw\|_2 \mid w \in r_0 + \mathcal{K}_k^{+1}(A|C|^{-1}Y, r_0) \} \end{aligned} \quad (14)$$

Definizione 6. Data C una matrice circolante con $|C|$ definita positiva, definiamo

$$\|y\|_c = (y^\top |C|^{-1}y)^{\frac{1}{2}} = \left\| |C|^{-\frac{1}{2}} y \right\|_2 \quad (15)$$

Questa è una norma indotta da un prodotto scalare non degenere, in quanto $|C|^{-1}$ è simmetrica definita positiva. ■

Riscriviamo [14](#) con questa nuova norma. Visto che $|C|^{-1}$ è simmetrica, si ha $Y|C|^{-1} = |C|^{-1}Y$.

$$\|L^{-1}Yw\|_2^2 = w^\top YL^{-\top}L^{-1}Yw = w^\top Y|C|^{-1}Yw = w^\top |C|^{-1}Y^2w = w^\top |C|^{-1}w = \|w\|_c^2$$

Concludendo, abbiamo mostrato che MINRES preconditionato calcola r_k come minimo del problema:

$$r_k = \operatorname{argmin}\{ \|w\|_c \mid w \in r_0 + \mathcal{K}_k^{+1}(A|C|^{-1}Y, r_0) \} \quad (16)$$

Abbiamo fatto questa derivazione per mostrare un errore presente nell'articolo di riferimento [1]. Nelle ultime due equazioni a pag. 276 viene detto che la norma da minimizzare è

$$\| |C|^{-1} r_k \|_2 = (r_k^\top |C|^{-2} r_k)^{\frac{1}{2}}$$

che è diversa da $\| r_k \|_c = (r_k^\top |C|^{-1} r_k)^{\frac{1}{2}}$; essendo forme quadratiche diverse, in generale avranno minimi in punti diversi. Nell'articolo non viene specificato in quale spazio venga scelto r_k (soltanto che $r_k = b - Ax_k$), ma guardando la derivazione teorica di MINRES preconditionato, dovrebbe essere lo stesso che abbiamo utilizzando qui.

Questo errore si ripercuote anche nelle equazioni successive come (3.5), tuttavia nella dimostrazione del lemma 3.1 che viene subito dopo, per la minimizzazione viene usata la norma corretta $\| |C|^{-\frac{1}{2}} Y r_k \|_2$.

3.3 Stime sulla velocità di convergenza di YMINRES

Definiamo Π_k come prima, ogni vettore $v \in \tilde{r}_0 + \mathcal{K}_k(\tilde{A}, \tilde{r}_0)$ può essere scritto come $v = p(\tilde{A})\tilde{r}_0$ con $p \in \Pi_k$. La condizione 12 si può riscrivere come:

$$\| \tilde{r}_k \|_2 = \min_{p \in \Pi_k} \| p(\tilde{A})\tilde{r}_0 \|_2.$$

\tilde{A} è simmetrica, dunque vale la dimostrazione fatta in 8 (con MINRES non preconditionato applicato a $\tilde{A}\tilde{x} = \tilde{b}$) e si ottiene:

$$\| \tilde{r}_k \|_2 \leq \min_{p \in \Pi_k} \max_{\lambda \in \text{Sp}(\tilde{A})} |p(\lambda)| \| \tilde{r}_0 \|_2 \quad (17)$$

Ricordiamo che $\tilde{r}_k = LYr_k$ e $\| \tilde{r}_k \|_2 = \| r_k \|_c$, dunque si ottiene:

$$\frac{\| r_k \|_c}{\| r_0 \|_c} \leq \min_{p \in \Pi_k} \max_{\lambda \in \text{Sp}(A|C|^{-1}Y)} |p(\lambda)|. \quad (18)$$

Ancora una volta, questa espressione è diversa da quella presentata nell'articolo di riferimento [1] a pag 276 (ultima equazione), dove al posto di $\| r_k \|_c$ c'è $\| |C|^{-1} r_k \|_2$.

Osserviamo inoltre che \tilde{A} , $|C|^{-1}YA$ e $A|C|^{-1}Y$ sono tutte simili, per cui l'insieme degli autovalori tra cui si cerca il massimo di $|p(\lambda)|$ è lo stesso.

Cerchiamo ora un buon polinomio $p \in \Pi_k$ in modo che minimizzi la stima al membro di destra. La seguente è una particolarizzazione della proposizione 3. Anche la proposizione (4.4) dell'articolo di riferimento [1] si basa su quest'ultima, però introduce alcune imprecisioni che riteniamo giusto evidenziare, ovvero sull'equivalenza di GMRES e MINRES preconditionati.

Proposizione 4. Sia B una matrice $n \times n$. Dato $\varepsilon > 0$, dividiamo gli autovalori di B in tre gruppi:

- $\Lambda_{+1} = \{ \lambda \in \text{Sp}(B) \text{ tali che } |\lambda - 1| < \varepsilon \}$. Sia p il numero di autovalori in Λ_{+1} .
- $\Lambda_{-1} = \{ \lambda \in \text{Sp}(B) \text{ tali che } |\lambda + 1| < \varepsilon \}$. Sia q il numero di autovalori in Λ_{-1} .

- $\Lambda_{out} = \{\lambda \in \text{Sp}(B) \text{ tali che } |\lambda - 1| \geq \varepsilon \text{ e } |\lambda + 1| \geq \varepsilon\}$. Sia $d = n - p - q$ il numero di autovalori in Λ_{out} , e li chiameremo outliers.

Applicando l'algoritmo GMRES al sistema $By = f$, otteniamo una sequenza di residui $s_k = f - By_k$, con $\|s_k\|_2$ minima nel sottospazio di Krylov $s_0 + \mathcal{K}_k^{+1}(B, s_0)$. Gli s_k soddisfano la seguente relazione:

$$\frac{\|s_{d+2k}\|_2}{\|s_0\|_2} \leq M(2\varepsilon + \varepsilon^2)^k \quad (19)$$

M dipende da ε e dagli autovalori, ma non da k . $\sigma = 2 + \varepsilon$ ed è già inglobata nell'equazione di sopra. Il valore di δ , interpretabile come distanza degli outliers dal bordo dei cerchi di centro $\gamma = \pm 1$ e raggio ε (nel piano complesso, ovvero con $z \in \mathbb{C}$) è:

$$\delta = \max_{\gamma=\pm 1} \max_{|z-\gamma|=\varepsilon} \max_{\lambda \text{ outlier}} \frac{|\lambda - z|}{|\lambda|}$$

Chiamate B_{+1} e B_{-1} le proiezioni di B sugli autospazi generati dagli autovalori vicini a $+1$ e -1 rispettivamente, l'espressione per M è la seguente:

$$M = 2\varepsilon \delta^d \max_{\gamma=\pm 1} \max_{|z-\gamma|=\varepsilon} \|(zI - B_\gamma)^{-1}\|$$

■

Se B è simmetrica, allora GMRES e MINRES sono matematicamente equivalenti: in aritmetica esatta calcolano le stesse iterate s_k . Analizziamo cosa succede utilizzando i preconditionatori. D'ora in poi indicheremo con r_k la sequenza generata da YMINRES preconditionato come descritto nella sezione 3.2, mentre con s_k^1, s_k^2, \dots i residui ottenuti da GMRES applicati a diverse varianti di preconditionamento. Per tutti i metodi si parte con lo stesso guess iniziale x_0 , in particolare $r_0 = b - Ax_0$ non dipende dal metodo scelto.

Prima di tutto consideriamo il sistema con due preconditionatori generici P_1 e P_2

$$(P_1^{-1}YAP_2^{-1})(P_2x) = P_1^{-1}Yb$$

L'algoritmo GMRES applicato a questo sistema, con $B = P_1^{-1}YAP_2^{-1}$, $y = P_2x$ e $f = P_1^{-1}Yb$, genera una sequenza di residui

$$s_k = \operatorname{argmin}\{ \|v\|_2 \mid v \in s_0 + \mathcal{K}_k^{+1}(B, s_0) \}$$

Similmente a come fatto in 13 e 14, possiamo riscrivere come:

$$s_k = \operatorname{argmin}\{ \|v\|_2 \mid v \in P_1^{-1}Y(r_0 + \mathcal{K}_k^{+1}(AP_2^{-1}P_1^{-1}Y, r_0)) \}$$

$$s_k = P_1^{-1}Yw \quad \text{con} \quad w = \operatorname{argmin}\{ \|P_1^{-1}Yw\|_2 \mid w \in r_0 + \mathcal{K}_k^{+1}(AP_2^{-1}P_1^{-1}Y, r_0) \}$$

Cerchiamo ora P_1, P_2 tali che i residui calcolati s_k coincidano (eventualmente moltiplicati per $P_1^{-1}Y$) con r_k ; in questo modo avremo la garanzia che YMINRES preconditionato con $|C|$ e GMRES con preconditionatori P_1, P_2 diano le stesse iterate, e dunque possiamo utilizzare la proposizione 4 per stimare la velocità di convergenza di YMINRES. Una condizione necessaria è che la matrice generatrice sia la stessa per entrambi i sottospazi di Krylov in cui si cerca il minimo: $AP_2^{-1}P_1^{-1}Y = A|C|^{-1}Y$, da cui $P_1P_2 = |C|$.

- Scegliamo $P_1 = |C|$, $P_2 = I$ e sia s_k^1 la sequenza di residui generata. Si ha:

$$s_k^1 = |C|^{-1}Yw \quad \text{con} \quad w = \operatorname{argmin}\{ \| |C|^{-1}Yw \|_2 \mid w \in r_0 + \mathcal{K}_k^{+1}(A|C|^{-1}Y, r_0) \}$$

- Scegliamo $P_1 = I$, $P_2 = |C|$ e sia s_k^2 la sequenza di residui generata. Si ha:

$$s_k^2 = Yw \quad \text{con} \quad w = \operatorname{argmin}\{ \|w\|_2 \mid w \in r_0 + \mathcal{K}_k^{+1}(A|C|^{-1}Y, r_0) \}$$

- Scegliamo $P_1 = P_2 = |C|^{\frac{1}{2}}$ e sia s_k^3 la sequenza di residui generata. Si ha:

$$s_k^3 = |C|^{-\frac{1}{2}}Yw \quad \text{con} \quad w = \operatorname{argmin}\{ \| |C|^{-\frac{1}{2}}Yw \|_2 \mid w \in r_0 + \mathcal{K}_k^{+1}(A|C|^{-1}Y, r_0) \}$$

In tutti i tre casi il sottospazio di Krylov in cui si cerca w minimizzante è lo stesso dell'algoritmo YMINRES preconditionato 16, però sono diverse le norme da minimizzare: $\| |C|^{-1}Yw \|_2$ per il primo, $\|w\|_c$ per il secondo e $\| |C|^{-\frac{1}{2}}Yw \|_2$ per il terzo, che è la stessa che serve per YMINRES.

Questo vuol dire che in generale gli w e s_k^1, s_k^2 generati dai primi due metodi saranno diversi dai r_k generati da YMINRES preconditionato con $|C|$ (non coincidono neanche facendo cambi di base con $P^{-1}Y$). Invece confrontando il terzo caso con 14 si vede che $s_k^1 = |C|^{-\frac{1}{2}}Yr_k$. La proposizione 4 dà la seguente stima:

$$M(2\varepsilon + \varepsilon^2)^k \geq \frac{\|s_{d+2k}^3\|_2}{\|s_0^3\|_2} = \frac{\| |C|^{-\frac{1}{2}}Yr_{d+2k} \|_2}{\| |C|^{-\frac{1}{2}}Yr_0 \|_2} = \frac{\| |C|^{-\frac{1}{2}}r_{d+2k} \|_2}{\| |C|^{-\frac{1}{2}}r_0 \|_2} = \frac{\|r_{d+2k}\|_c}{\|r_0\|_c}$$

Osserviamo che la stima ottenuta è diversa da quella presentata nell'articolo di riferimento [1, equazione 4.3], in cui c'è scritto:

$$\frac{\| |C|^{-1}r_{d+2k} \|_2}{\| |C|^{-1}r_0 \|_2} \leq M(2\varepsilon + \varepsilon^2)^k$$

Questa è lo stesso errore che abbiamo notato precedentemente. Abbiamo fatto questa analisi con diversi modi di preconditionatori per mostrare che MINRES e GMRES preconditionati non sono sempre matematicamente equivalenti, ma bisogna scegliere in modo opportuno i preconditionatori. Infatti, si vede con gli stessi ragionamenti fatti sopra che gli unici preconditionatori per GMRES che danno esattamente le stesse iterate di MINRES sono $P_1 = P_2 = |C|^{-\frac{1}{2}}$.

A monte di tali osservazioni, riassumiamo i risultati ottenuti per questa scelta di preconditionatori nella seguente proposizione:

Proposizione 5. Data A Toeplitz, C preconditionatore circolante e $\varepsilon > 0$, dividiamo gli autovalori di $|C|^{-1}YA$ (o equivalentemente $A|C|^{-1}Y$) in tre gruppi:

- Ci sono p autovalori $\lambda \in \operatorname{Sp}(|C|^{-1}YA)$ tali che $|\lambda - 1| < \varepsilon$; denotiamo il loro insieme come Λ_{+1} .

- Ci sono q autovalori $\lambda \in \text{Sp}(|C|^{-1}YA)$ tali che $|\lambda + 1| < \varepsilon$; denotiamo il loro insieme come Λ_{-1} .
- Ci sono $d = n - p - q$ autovalori $\lambda \in \text{Sp}(|C|^{-1}YA)$ tali che $|\lambda - 1| \geq \varepsilon$ e $|\lambda + 1| \geq \varepsilon$; denotiamo il loro insieme come Λ_{out} .

Applicando l'algoritmo YMINRES al sistema 10 con preconditionatore $|C|$, otteniamo una sequenza di residui r_k (soluzione del problema di minimo 16) che soddisfano:

$$\frac{\left\| |C|^{-\frac{1}{2}} r_{d+2k} \right\|_2}{\left\| |C|^{-\frac{1}{2}} r_0 \right\|_2} \leq M(2\varepsilon + \varepsilon^2)^k \quad (20)$$

L'espressione di M è la seguente, dove possiamo calcolare esplicitamente $\left\| (zI - B_\gamma)^{-1} \right\|_2$ in funzione degli autovalori perché $B = YA$ è simmetrica.

$$\delta = \max_{\gamma=\pm 1} \max_{|z-\gamma|=\varepsilon} \max_{\lambda \in \Lambda_{out}} \frac{|\lambda - z|}{|\lambda|}$$

$$M = 2\varepsilon \delta^d \max_{\gamma=\pm 1} \max_{\lambda \in \Lambda_\gamma} \frac{1}{\varepsilon - |\lambda - \gamma|}$$

L'ultimo termine, $\varepsilon - |\lambda - \gamma|$, è la minima distanza tra un autovalore $\lambda \in \Lambda_\gamma$ dal bordo del cerchio di raggio ε e centro γ . ■

4 Sperimentazione

4.1 Matrici utilizzate

La sperimentazione è stata eseguita sulle seguenti tipologie di matrici. Esse sono state generate dallo script `matrixgen(n,s,t)`, dove n è la dimensione, s indica il tipo di matrice e t è un parametro per alcune tipologie matrici (il valore usato è $t = 0.01$ per $s = 4, 5, 6$). Indicheremo con `Toeplitz(v,b,w)` la matrice di Toeplitz A che ha $A(1,1) = b$, $A(1,2:n) = [w \ 0]$ e $A(2:n,1) = [v; 0]$, dove 0 indica un blocco di 0 di dimensione opportuna.

- $s = 1$, A_n è blocco di Jordan $n \times n$ con autovalore $\lambda = 1.1$.

$$A = \begin{pmatrix} 1.1 & 1 & & & \\ & 1.1 & \dots & & \\ & & \dots & \dots & \\ & & & \dots & 1 \\ & & & & 1.1 \end{pmatrix}$$

- Siccome LSQR per ogni iterazione fa due moltiplicazioni matrice-vettore (per A e per A^T), nella tabella viene indicato $2 \cdot$ numero di iterazioni, per rimanere in linea con l'articolo di riferimento [1].
- Le matrici di tipo $s = 1, 2$ sono ben condizionate, mentre $s = 3, 4, 5, 6$ sono mal condizionate. In particolare il simbolo relativo (con parametro $t = 0$) per $s = 3, 4$ si annulla in un punto, per $s = 5$ in due punti e per $s = 6$ in tre punti. Questa cosa si nota anche sperimentalmente, poiché per $n \geq 2000$ i metodi si fermano dopo poche iterazioni ma con residui molto grandi, per via dei numeri di condizionamento elevati (tali situazioni sono evidenziate in rosso).

Inoltre, per $1 \leq s \leq 6$ le matrici sono sparse (ci sono meno di $7n$ entrate non nulle), mentre per $s = 7$ la matrice è piena. Il suo simbolo ha uno zero singolo, ma i metodi riescono comunque a convergere abbastanza bene.

- Per ciascuna scelta di n , preconditionatore e metodo, si sono fatte 10 prove multiple di risoluzione dei sistemi $Ax = b$, per vedere quanto cambiasse il numero di iterazioni a seconda della scelta di b . Non riportiamo i dati numerici ottenuti, però nella maggior parte dei casi con preconditionamento, il numero di iterazioni rimaneva esattamente uguale; nel resto cambiava di qualche unità. I casi non preconditionati erano più sensibili: il numero di iterazioni variava di 5-10; tuttavia visto che servono 200 – 300 iterazioni, tali oscillazioni sono trascurabili.
- I metodi non preconditionati richiedono molte iterazioni per convergere, e spesso non ci riescono in MAXIT iterazioni. GMRES funziona un po' meglio di YMINRES e LSQR, ma comunque molto lentamente.

Questo ci porta a concludere che l'uso dei preconditionatori migliora notevolmente l'efficacia degli algoritmi, in particolare anche per YMINRES.

- Tra tutti i preconditionatori, quello che da risultati migliori è quello di Strang, richiedendo tra 5 e 20 iterazioni per convergere, anche per valori di n piuttosto grandi. Il preconditionatore Ottimale è il secondo migliore, richiedendo 20 – 30 iterazioni per convergere; il Superottimale è invece il più lento tra i tre.

Evidenziamo però che per $1 \leq s \leq 6$ (ovvero la maggior parte della sperimentazione), le matrici erano sparse, per cui anche il preconditionatore di Strang viene sparso; quelli Ottimale e Superottimale sono in generale pieni, indipendentemente dalla sparsità di A . Invece per $s = 7$ la matrice considerata è piena, e i preconditionatori Strang e Ottimale richiedono circa lo stesso numero di iterazioni.

- Guardando gli autovalori di $|C|^{-1}A$, per $|C| = \text{Strang}$ o Ottimale gli autovalori sono ben raggruppati molto bene vicino ± 1 , nella maggior parte dei casi contenuti in $(-1.01, -0.99) \cup (0.99, 1.01)$. Invece per $|C| = \text{Superottimale}$ ci sono molti autovalori nell'intervallo $(-1, 1)$. Questo comportamento non solo è riscontrabile con peggiori bound teorici (Axelsson-Lindskog e proposizione 3), ma anche sperimentalmente si richiedono molte più iterazioni per arrivare a convergenza.
- Possiamo notare un andamento “a scalini” nelle figure 3.1 (Superottimale), 4.1 (Ottimale), 5.1 (Ottimale e Superottimale), 7.1 (precondizionato). Il grafico ha

dei tratti “orizzontali”, in cui la norma del residuo decresce poco, ma ci sono alcuni punti in cui la velocità di convergenza aumenta notevolmente e la norma del residuo decresce molto rapidamente.

Per GMRES una volta raggiunta la regione di convergenza veloce, si arriva alla tolleranza richiesta con grande velocità; per gli altri metodi, a volte si arriva ad altre regioni “piatte” in cui la convergenza diventa nuovamente lenta. Per LSQR spesso bastano due “scalini” per arrivare alla tolleranza.

È curioso osservare che gli scalini per YMINRES hanno lunghezza 2: quando si arriva alla regione di convergenza veloce, c’è un passo che diminuisce molto la norma, e un altro che la lascia quasi invariata.

- Una spiegazione euristica di questo comportamento è dato dalla proposizione 5. Se tutti gli autovalori tranne d sono contenuti in P cluster di raggio ε , allora dopo d iterazioni del metodo, la norma dei residui si riduce di $(2\varepsilon + \varepsilon^2)$ ad ogni passo. I tratti orizzontali possono essere considerati come quelli in cui si minimizza la norma dovuta ai d autovalori outliers, mentre la convergenza veloce è quando si arriva alle palle di raggio ε .

Una possibile spiegazione dell’andamento a scalini è che, dopo alcuni passi di convergenza veloce, si arrivi di nuovo ad una situazione con alcuni autovalori “cattivi”; l’andamento orizzontale serve per migliorare la stima negli autospazi generati da loro.

- Per vedere quanto questa spiegazione euristica (derivante dalle proposizioni 3 e 5) sia valida, nella sezione 5.8 abbiamo riportato il calcolo della stima al membro di destra in 20. Si è usato $n = 1000$ per quattro tipologie di matrici, e si è usata una tolleranza più piccola $\text{tol} = 10^{-14}$.

Vediamo tuttavia che la stima è troppo larga rispetto al numero di iterazioni effettivamente necessario: si sono considerati valori di ε di ordine di grandezza 10^{-1} , 10^{-2} , per cui il valore di M è dell’ordine di 10^{10} o addirittura 10^{100} . Questo rende la stima poco utile dal punto di vista pratico, perché mentre il residuo r_k ha già raggiunto la tolleranza richiesta, la stima teorica è maggiore di 1.

Inoltre dato ε e il corrispondente numero di outliers d , possiamo notare che il fattore di riduzione $(2\varepsilon + \varepsilon^2)$ stima abbastanza bene la riduzione della norma al passo d : $\|r_{d+1}\|_2 / \|r_d\|$. Infatti ad occhio le rette corrispondenti alla stima sono parallele alle tangenti al grafico nel punto $(d, \|r_d\|)$; tuttavia, il fatto che la costante M davanti sia molto grande rende troppo elevata la stima, ottenendo quindi una “retta traslata parallelamente”.

Possiamo concludere che tale stima è importante per dimostrare la convergenza, ma numericamente è poco efficace.

5 Tabelle e grafici

5.1 Tavola 1: blocco di Jordan

A_n = blocco di Jordan con autovalore $\lambda = 1.1$

Numero di iterazioni per metodo:

Precondizionatore	n	YMINRES	GMRES	LSQR
Non preconditionato	10	10	10	20
	100	100	96	200
	500	350	171	356
	1000	356	174	360
	2000	360	177	362
	5000	362	177	362
	10000	360	177	362
Strang	10	4	2	6
	100	4	2	6
	500	4	2	6
	1000	4	2	6
	2000	4	2	6
	5000	4	2	6
	10000	4	2	6
Ottimale	10	10	8	14
	100	13	7	16
	500	10	5	14
	1000	10	5	12
	2000	8	4	12
	5000	8	4	10
	10000	8	3	10
Superottimale	10	10	9	18
	100	21	11	30
	500	18	8	20
	1000	14	7	18
	2000	12	6	16
	5000	10	5	14
	10000	10	5	12

Numero di condizionamento di A_n :

n	10	100	500	1000	2000	5000	10000
$\kappa(A_n)$	9.2	20.1	20.9	21.0	21.0	21.0	21.0

Grafici per $n = 1000$

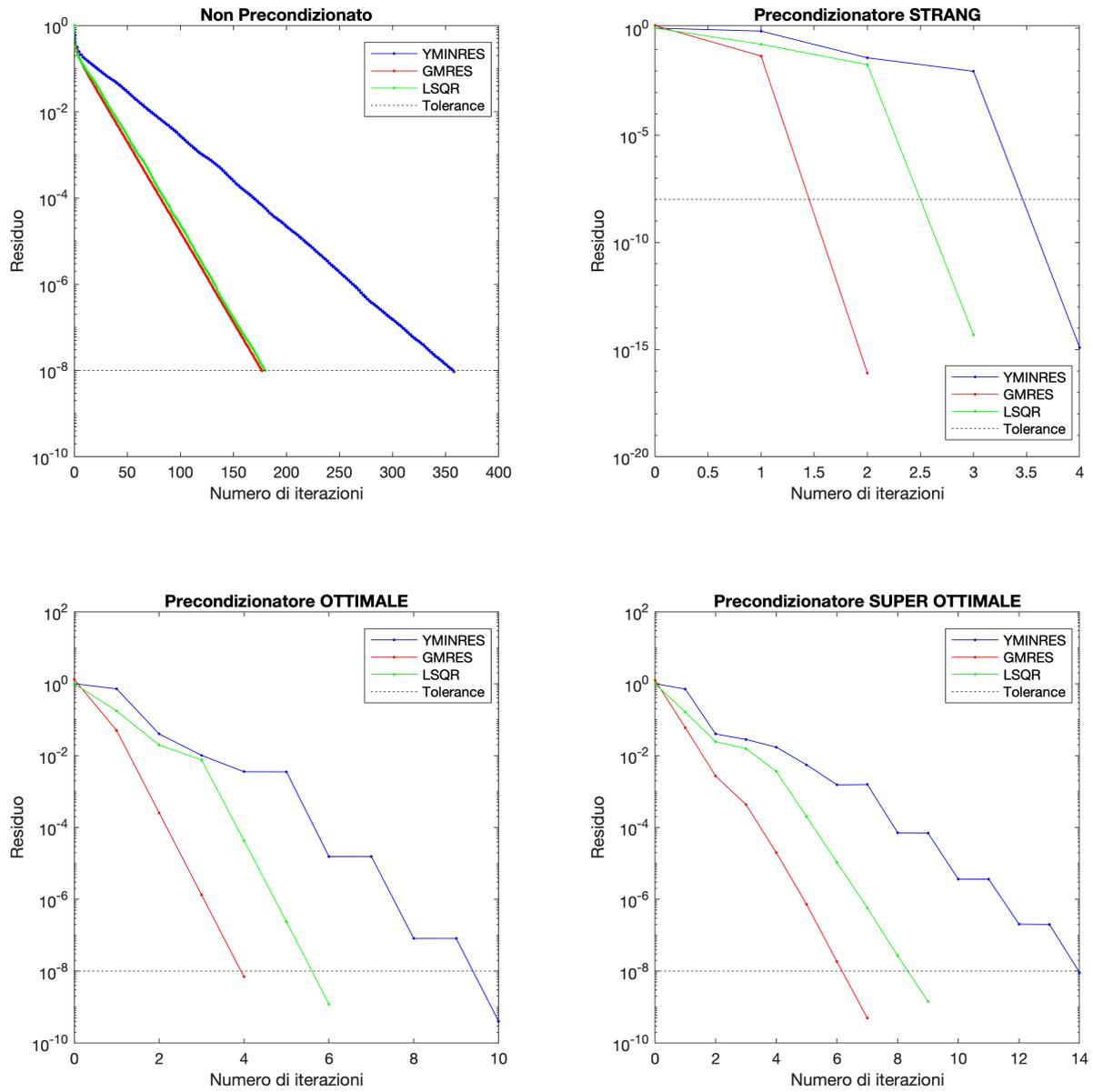


Figura 1.1: Confronto dei diversi metodi, a seconda del preconditionatore usato

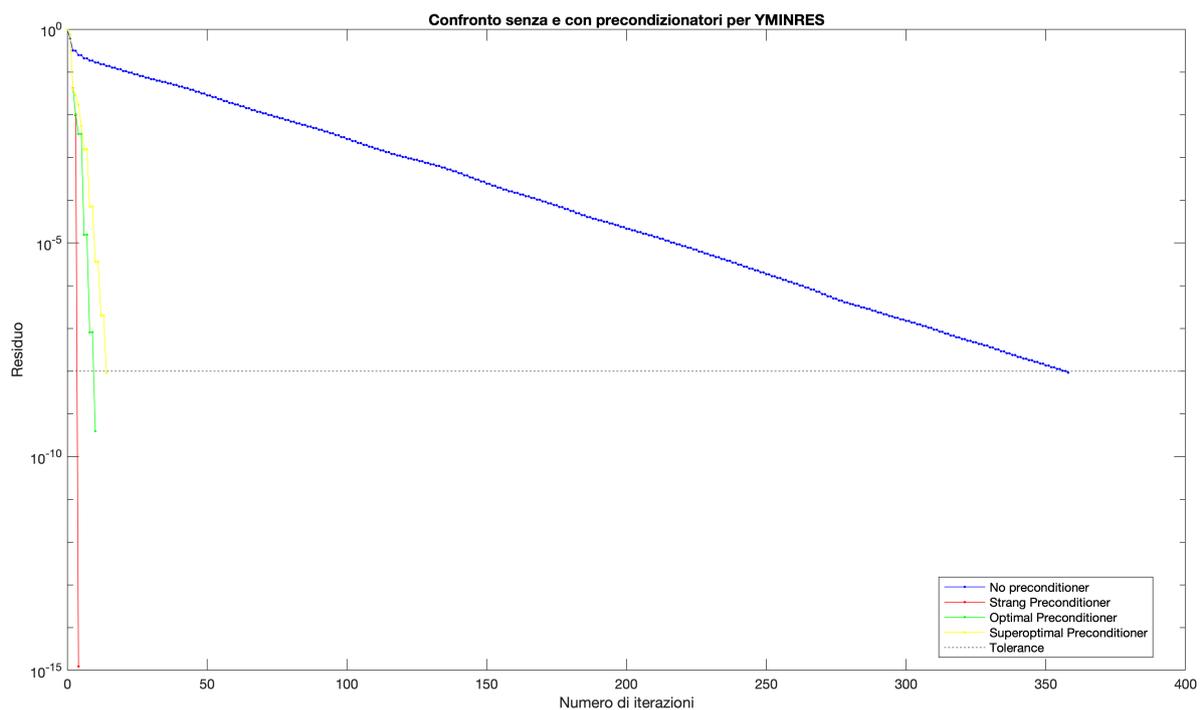


Figura 1.2: Confronto dei diversi preconditionatori per YMINRES

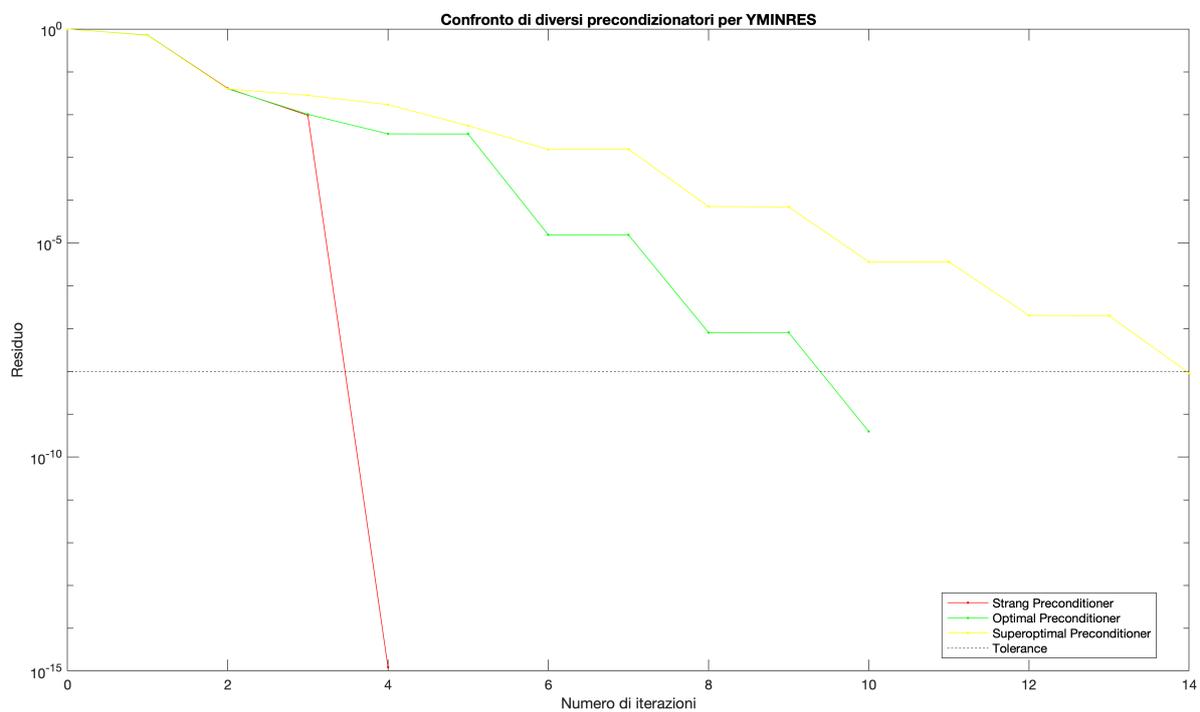


Figura 1.3: Zoom sui tre metodi, tolto YMINRES non preconditionatore

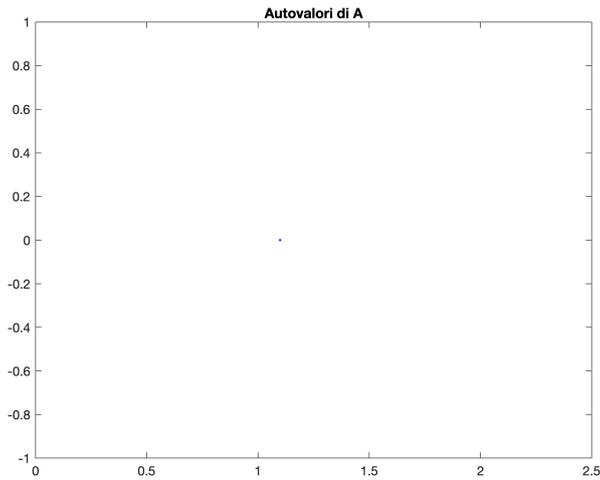


Figura 1.4: Plot degli autovalori di A_{1000}

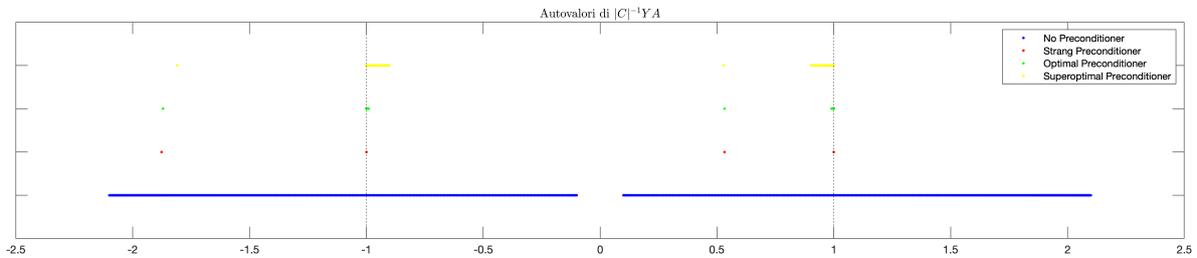


Figura 1.5: Autovalori delle matrici preconditionate $|C|^{-1}A_{1000}$.

Commenti:

- Facendo più esperimenti variando il vettore b del sistema $Ax = b$, il numero di iterazioni per arrivare a convergenza resta lo stesso per i metodi preconditionati. Nel caso non preconditionato, variano sensibilmente poco (± 5 iterazioni).
- Osserviamo che il numero di iterazioni viene esattamente lo stesso dell'articolo di riferimento [1, Tabella 2], tranne che per il caso non preconditionato e per il metodo GMRES, dove per $n = 10, 100$ viene un'iterazione in meno, mentre per $n = 1000$ è uguale.
- Osservando i grafici notiamo che la convergenza nel caso non preconditionato è lineare per tutti i tre metodi, con YMINRES più lento di GMRES e LSQR. Con il preconditionamento i metodi convergono in poche iterazioni, ottenendo risultati migliori usando il preconditionatore di Strang.
- Notiamo due andamenti curiosi: il primo è che per $n \geq 100$ con preconditionamento, il numero di iterazioni diminuisce all'aumentare di n . Il secondo è che YMINRES ha una convergenza "a scalini", che inizia a mostrarsi dopo i primi passi.

5.2 Tavola 2: GRCAR

$$A_n = \text{Toeplitz}(-1, 1, [1 \ 1 \ 1])$$

Numero di iterazioni per metodo:

Precondizionatore	n	YMINRES	GMRES	LSQR
Non preconditionato	10	10	10	20
	100	58	81	62
	500	64	255	64
	1000	64	288	64
	2000	64	289	64
	5000	64	290	64
	10000	64	291	64
	Strang	10	10	5
100		10	4	18
500		10	4	18
1000		10	4	18
2000		10	4	18
5000		10	4	18
10000		10	4	18
Ottimale		10	10	10
	100	16	7	22
	500	14	6	20
	1000	14	6	20
	2000	12	6	20
	5000	12	5	20
	10000	12	5	20
	Superottimale	10	10	10
100		17	8	22
500		14	7	20
1000		14	6	20
2000		14	6	20
5000		12	6	20
10000		12	6	20

Numero di condizionamento di A_n :

n	10	100	500	1000	2000	5000	10000
$\kappa(A_n)$	2.89	3.59	3.63	3.63	3.63	3.63	3.63

Grafici per $n = 1000$

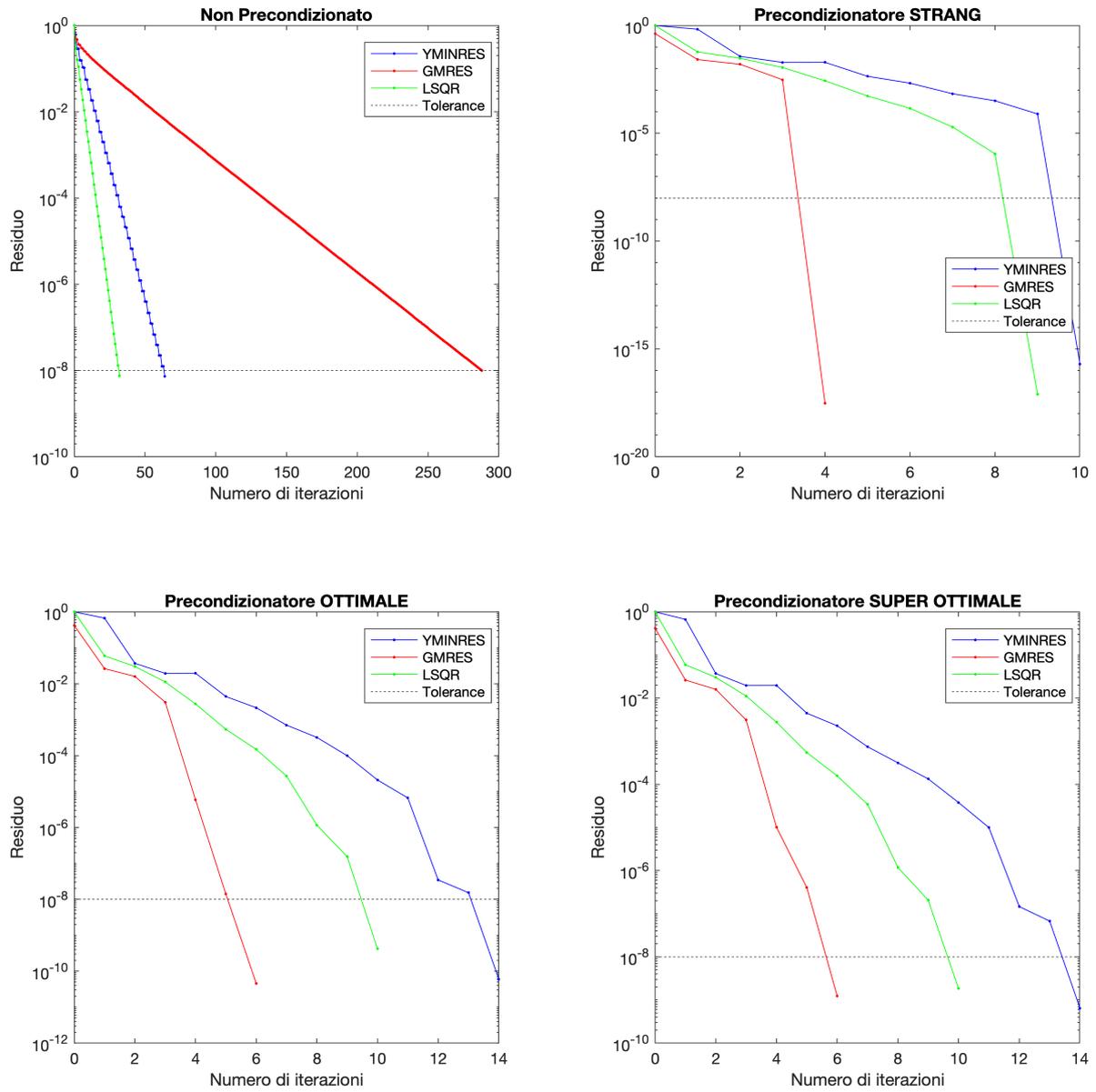


Figura 2.1: Confronto dei diversi metodi, a seconda del preconditionatore usato

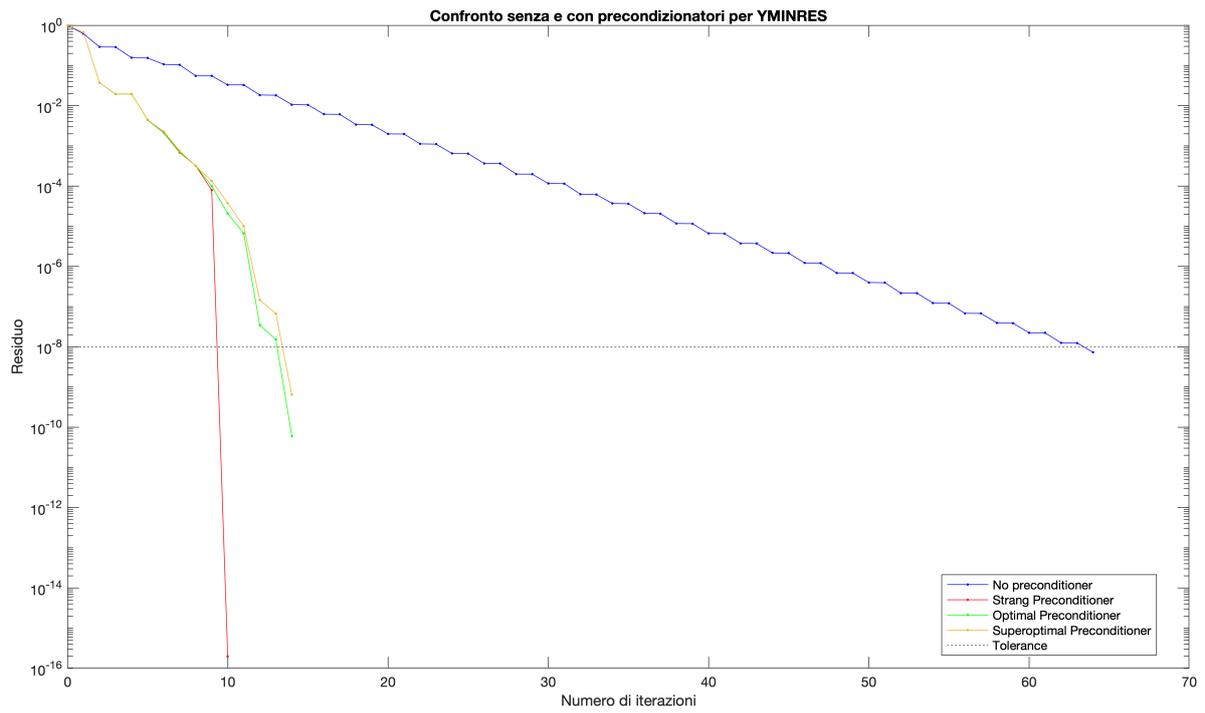


Figura 2.2: Confronto dei diversi preconditionatori per YMINRES

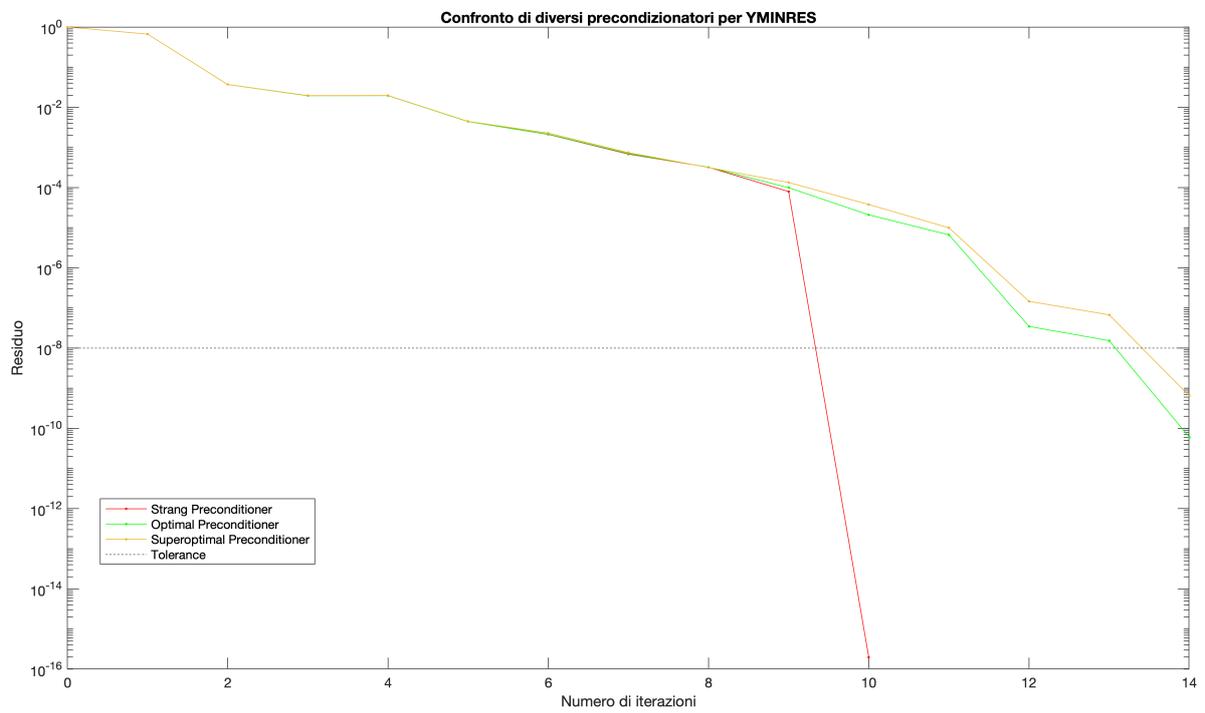


Figura 2.3: Zoom sui tre metodi, tolto YMINRES non preconditionatore

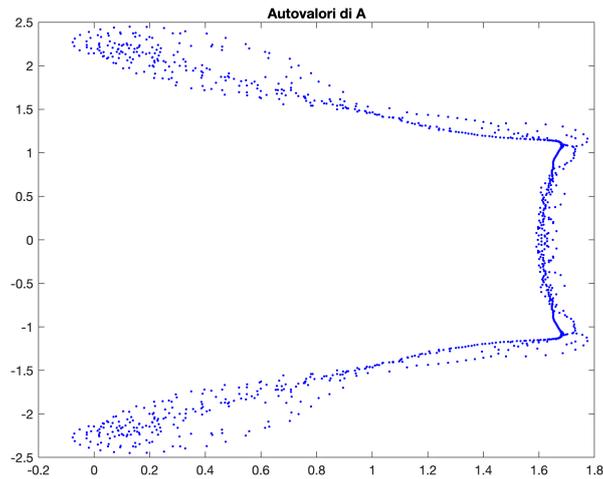


Figura 2.4: Plot degli autovalori di A

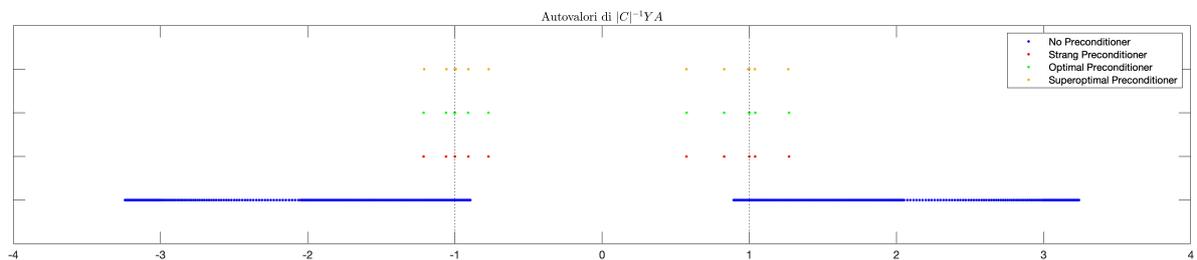


Figura 2.5: Autovalori delle matrici preconditionate.

Commenti:

- Il numero di iterazioni (con preconditionamento) è consistente con quelli ottenuti nell'articolo [1, Tabella 3], facendo qualche iterazione in meno per GMRES e nel resto quasi uguale.
- Senza preconditionamento GMRES è il metodo peggiore, ma in tutti i casi preconditionati richiede circa la metà delle iterazioni. YMINRES richiede più iterazioni di LSQR, ma in generale richiede meno moltiplicazioni matrice-vettore.
- Nel caso non preconditionato la convergenza è lineare. Con il preconditionatore di Strang, tutti e tre i metodi fanno alcune iterazioni diminuendo il poco il residuo, e poi in un solo passo arrivano a precisione di macchina. Per i preconditionatori ottimale e superottimale, la situazione è intermedia: ad occhio la convergenza è superlineare, ma c'è solo un graduale aumento della pendenza, senza alcun salto netto.
- Similmente a quanto accade per $A =$ Blocco di Jordan, notiamo che il numero di iterazioni diminuisce all'aumentare di $n \geq 100$ con i preconditionatori Ottimale e Superottimale. Per Strang il numero di iterazioni non dipende dalla dimensione.
- Per YMINRES, osserviamo dai grafici l'andamento a scalini delle iterazioni, ben visibile nel caso non preconditionato. Il preconditionatore di Strang risulta il migliore: alla decima iterazione il residuo passa da 10^{-4} alla precisione di macchina. Per Ottimale e Superottimale, l'andamento è leggermente superlineare.

5.3 Tavola 3: GRCAR_0

$$A_n = \text{Toeplitz}(-1, 0, [1 \ 1 \ 1])$$

Numero di iterazioni per metodo:

Precondizionatore	n	YMINRES	GMRES	LSQR
Non preconditionato	10	10	10	20
	100	95	100	160
	500	339	456	610
	1000	610	902	1152
	2000	-	-	-
	5000	-	-	-
	10000	-	-	-
Strang	11	10	5	18
	101	10	5	20
	501	10	5	20
	1001	10	5	20
	2001	10	5	20
	5001	10	5	20
	10001	10	5	22
Ottimale	10	10	10	22
	100	17	9	30
	500	15	9	28
	1000	15	9	30
	2000	15	8	30
	5000	15	8	30
	10000	15	8	30
Superottimale	10	10	10	20
	100	26	14	42
	500	35	19	64
	1000	42	23	78
	2000	53	27	100
	5000	71	36	144
	10000	94	46	222

Numero di condizionamento di A_n :

n	10	100	500	1000	2000	5000	10000
$\kappa(A_n)$	8.86	70.8	343	683	$1.36 \cdot 10^3$	$3.40 \cdot 10^3$	$6.80 \cdot 10^3$

Grafici per $n = 1001$

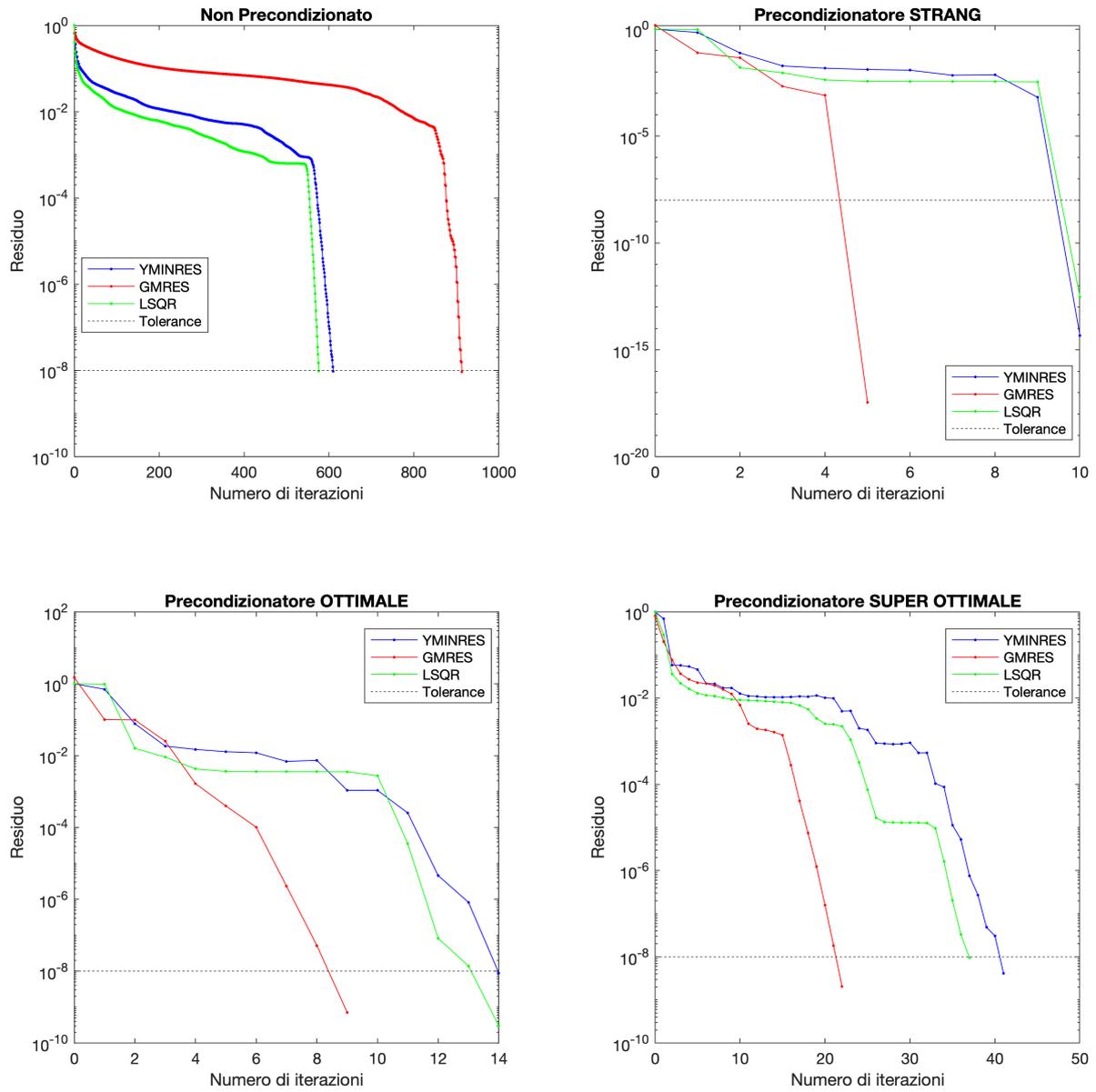


Figura 3.1: Confronto dei diversi metodi, a seconda del preconditionatore usato

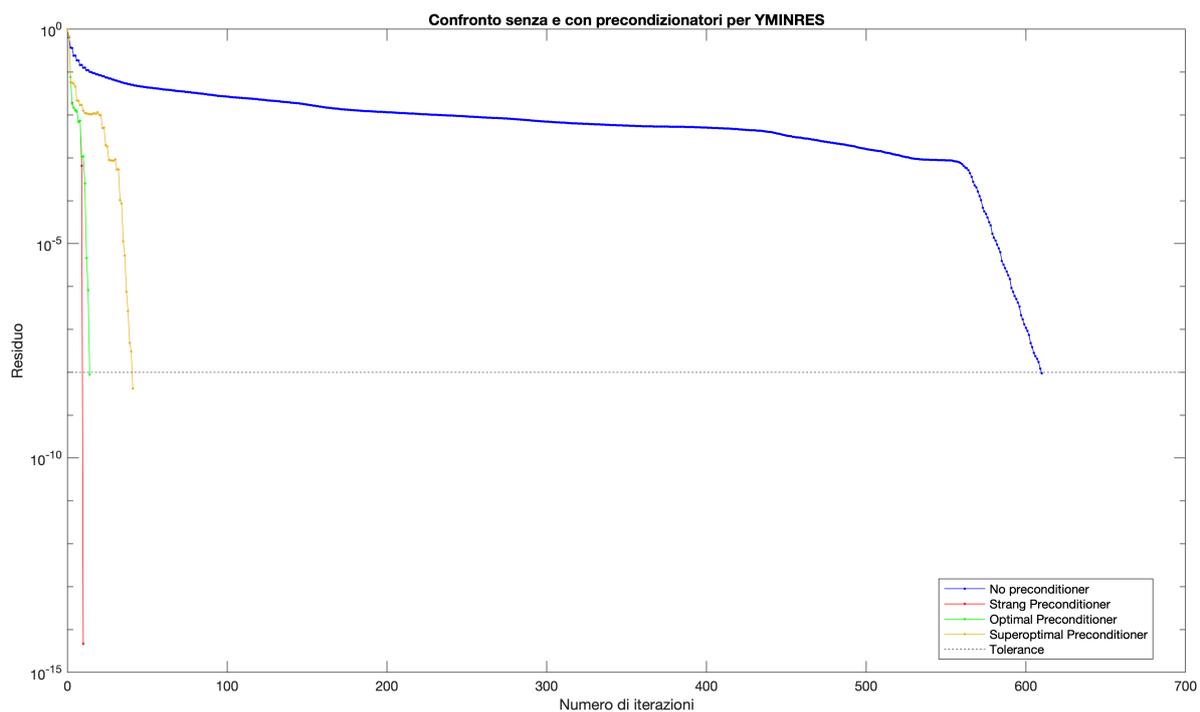


Figura 3.2: Confronto dei diversi preconditionatori per YMINRES

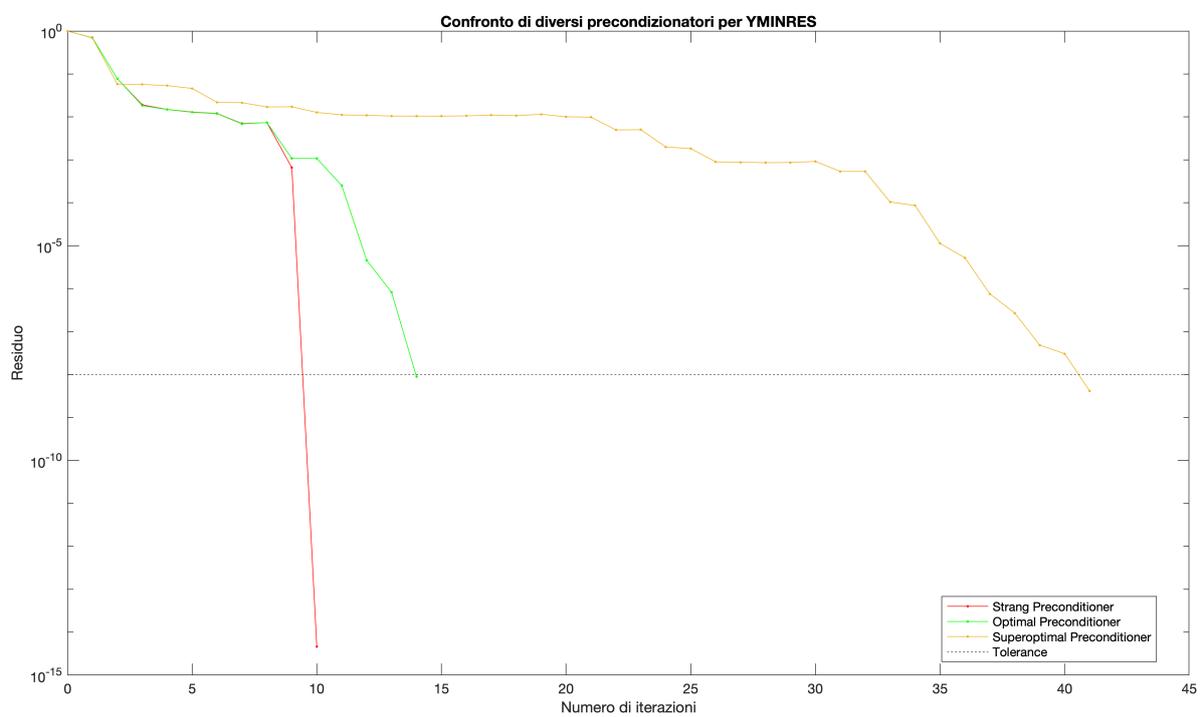


Figura 3.3: Zoom sui tre metodi, tolto YMINRES non preconditionatore

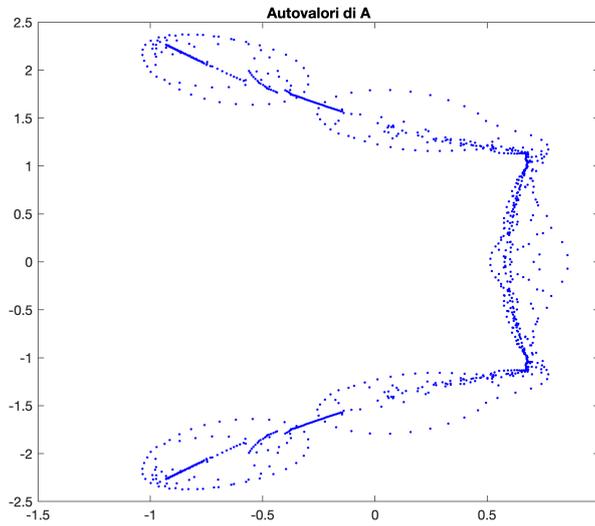


Figura 3.4: Plot degli autovalori di A_{1001}

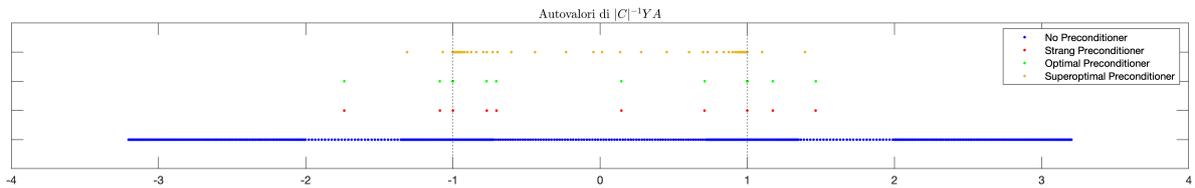


Figura 3.5: Autovalori delle matrici preconditionate $|C|^{-1}A_{1001}$.

Commenti:

- Per $n \geq 2000$, nel caso non preconditionato i metodi raggiungono $\text{MAXIT} = 1000$ iterazioni senza convergere.
- I risultati per il preconditionatore di Strang dipendono dalla parità di n . Per n dispari funziona molto bene e servono meno iterazioni degli altri metodi; per di più, il numero di iterazioni rimane lo stesso anche per n grandi. Tuttavia per n pari il preconditionatore di Strang è numericamente singolare, per cui il metodo dà problemi sin dal primo passo.
In figura e nella tabella relativa a Strang sono raffigurati i risultati con n dispari, ma per gli altri preconditionatori si sono utilizzati i valori "usuali" pari.
- Nel caso non preconditionato, i tre metodi fanno molte iterazioni diminuendo poco il residuo (circa 600 per YMINRES e LSQR, e 900 per GMRES), per poi aumentare di molto la velocità di convergenza e arrivare alla tolleranza in qualche decina di iterazioni.
Si può vedere questo andamento con il preconditionatore di Strang (in 10 iterazioni per YMINRES e LSQR, e 5 per GMRES). Anche quello superottimale ha un comportamento simile: si vede uno scalino iniziale per tutti i metodi, poi uno piccolo e uno grande per GMRES, uno piccolo e due medi per LSQR, mentre YMINRES fa tanti salti piccoli prima di convergere.
Questo comportamento non è presente con il preconditionatore ottimale, ma notiamo che necessita di poche iterazioni, quindi può essere ragionevolmente non visibile.
- Per YMINRES, il preconditionatore migliore è Strang per n dispari (mentre per n pari è singolare). Il secondo più veloce è quello Ottimale, che non ha problemi di singolarità e dà consistentemente buoni risultati.

5.4 Tavola 4: Toeplitz(1,1,t)

$$A_n = \text{Toeplitz}(1,1,t)$$

Numero di iterazioni per metodo:

Precondizionatore	n	YMINRES	GMRES	LSQR
Non preconditionato	10	10	10	20
	100	100	98	200
	500	502	490	1008
	1000	-	978	-
	2000	-	-	-
	5000	-	-	-
	10000	-	-	-
Strang	10	6	3	12
	100	6	3	12
	500	6	3	12
	1000	6	3	12
	2000	13	4	6
	5000	3	3	6
	10000	3	3	6
Ottimale	10	10	8	14
	100	13	7	18
	500	15	10	24
	1000	18	10	30
	2000	32	11	14
	5000	52	13	14
	10000	5	11	10
Superottimale	10	10	9	20
	100	29	17	48
	500	43	24	78
	1000	55	29	104
	2000	73	36	94
	5000	142	42	62
	10000	71	33	40

Numero di condizionamento di A_n :

n	10	100	500	1000	2000	5000	10000
$\kappa(A_n)$	13	128	637	$1.27 \cdot 10^3$	$2.55 \cdot 10^3$	$6.37 \cdot 10^3$	$1.27 \cdot 10^4$

Grafici per $n = 1000$

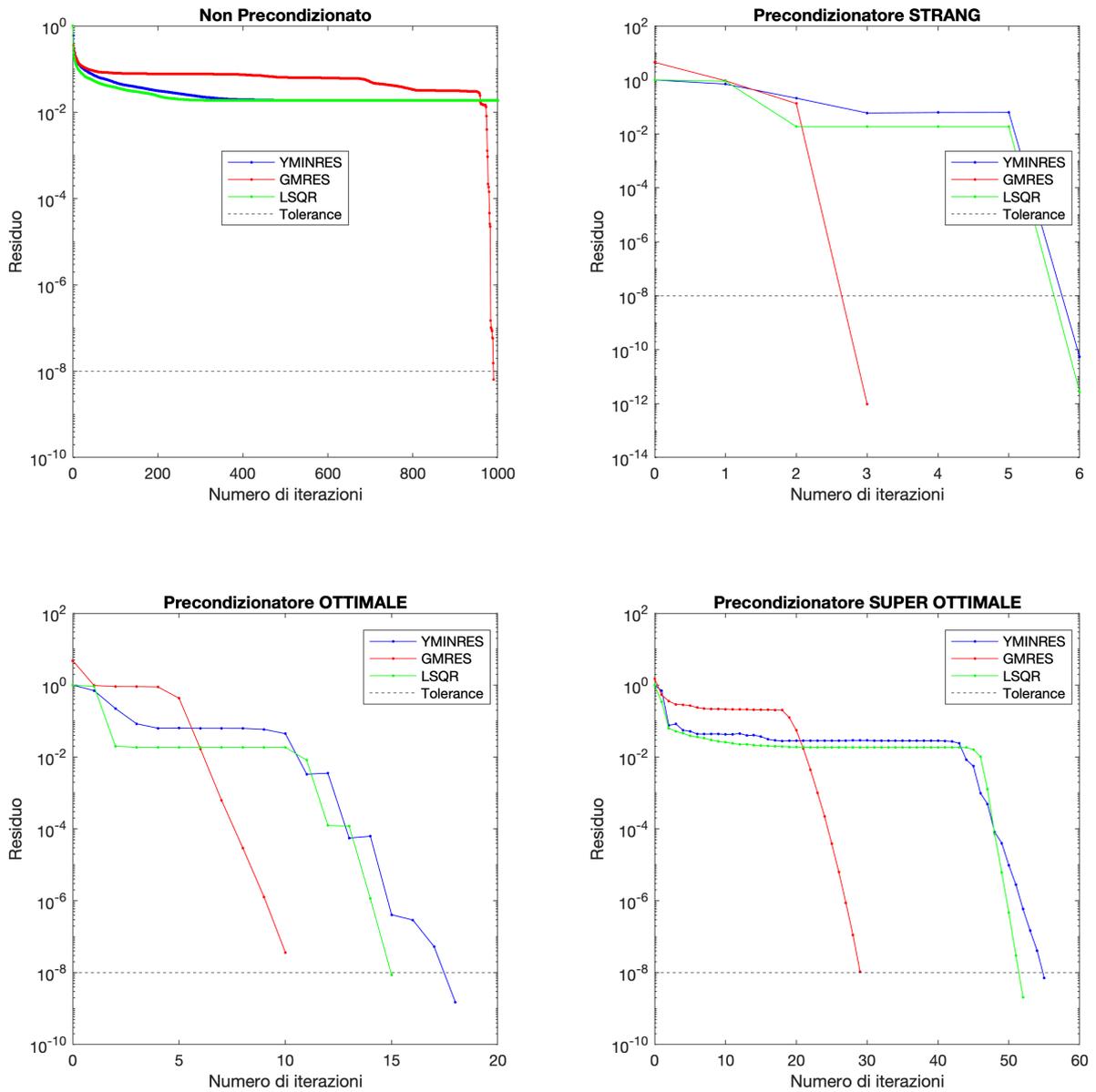


Figura 4.1: Confronto dei diversi metodi, a seconda del preconditionatore usato

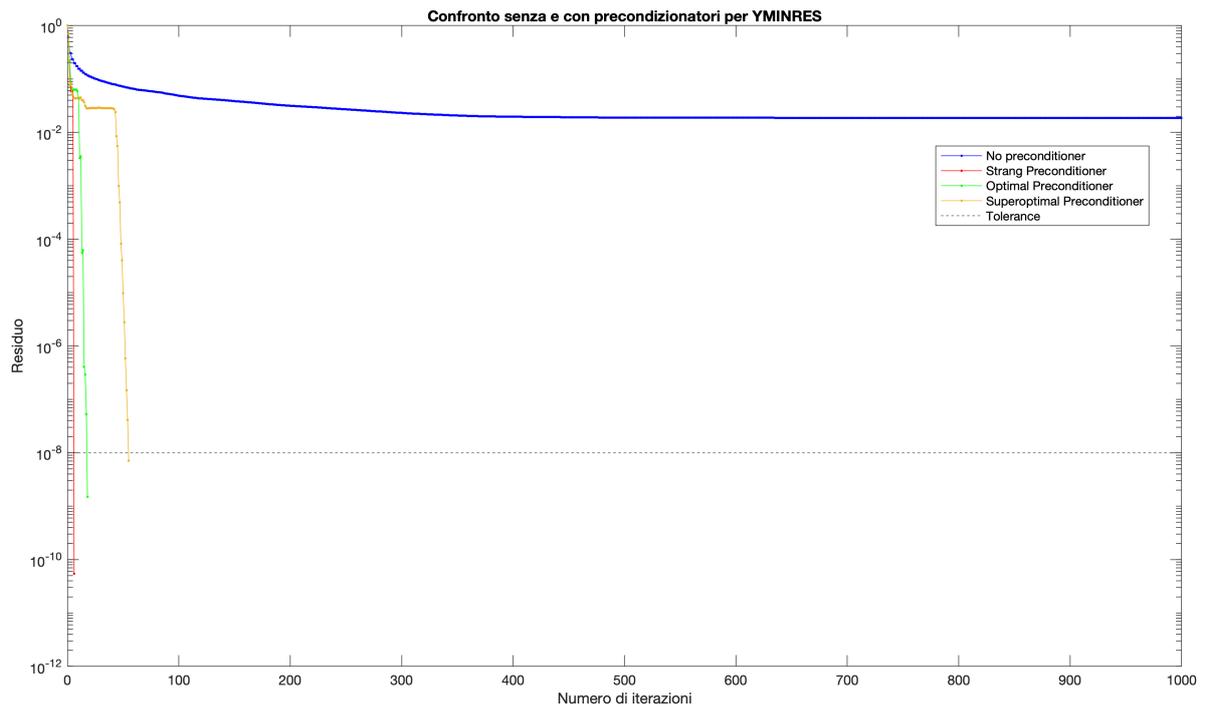


Figura 4.2: Confronto dei diversi preconditionatori per YMINRES

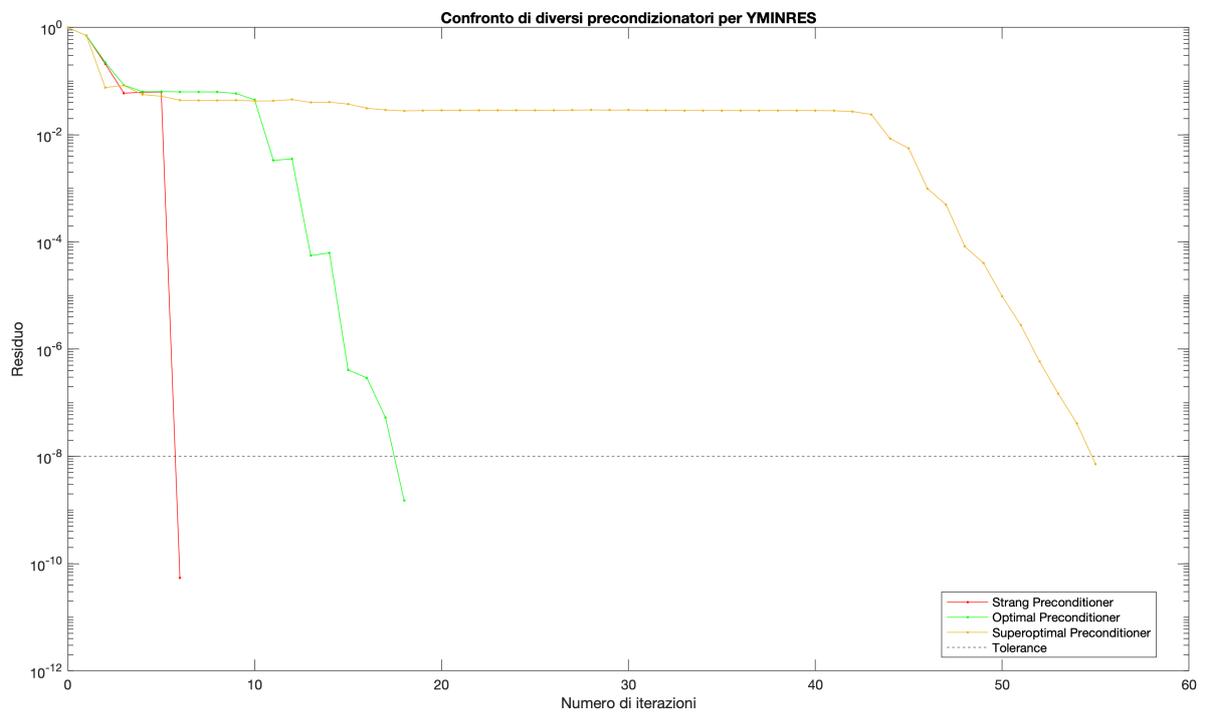


Figura 4.3: Zoom sui tre metodi, tolto YMINRES non preconditionatore

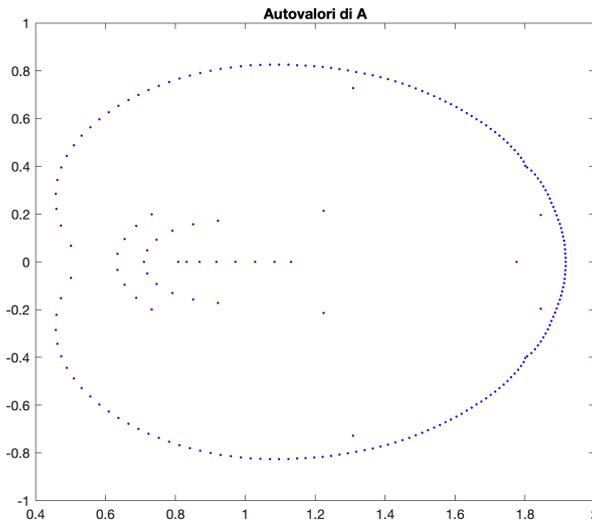


Figura 4.4: Plot degli autovalori di A_{1000}

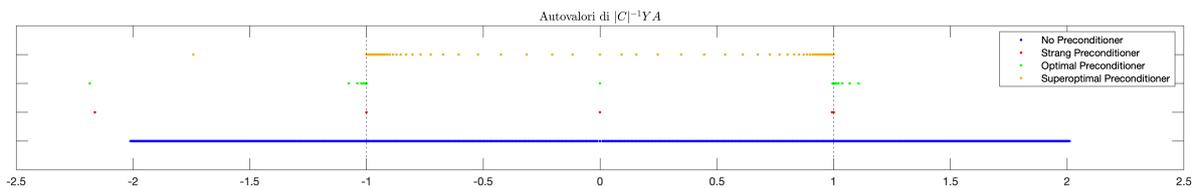


Figura 4.5: Autovalori delle matrici preconditionate $|C|^{-1}A_{1000}$.

Commenti:

- Il numero di iterazioni ottenuto sono simili a quelli dell'articolo di riferimento [1, Tabella 4]; si discostano di più rispetto alle matrici $A = \text{Jordan}$ o GRCAR , ma la differenza è solo di qualche unità.
- Notiamo che per $n \geq 1000$ non c'è più convergenza per i metodi non preconditionati (tranne per GMRES con $n = 1000$).
- Per $n \geq 1000$, le matrici preconditionate $|C|^{-1}A$ hanno degli autovalori vicini a 0. Per $n = 1000$ questi autovalori sono dell'ordine di 10^{-5} e il metodo converge bene, tuttavia per $n = 2000$ sono già dell'ordine di 10^{-10} , che crea problemi per la convergenza. I metodi iterativi si bloccano dopo qualche decina di passi, a volte raggiungendo e a volte senza raggiungere la tolleranza richiesta; in entrambi i casi, il vettore che trovano è molto distante dalla soluzione vera del problema. Notiamo inoltre che con Strang e Ottimale gli autovalori sono raggruppati attorno a ± 1 , mentre con Superottimale sono più distribuiti su $(-1, 1)$.
- Dai grafici per $n = 1000$, vediamo con i preconditionatori Strang e Ottimale un andamento molto piano, che poi si trasforma in un salto con convergenza molto veloce. Questa cosa si vede anche per GMRES nel caso non preconditionato.

Per il preconditionatore ottimale, le ultime iterazioni di GMRES sono meno "a picco" rispetto agli altri due preconditionatori, ma converge comunque abbastanza velocemente. Per YMINRES e LSQR, si nota maggiormente l'andamento a scalini.

- Il preconditionatore che da risultati migliori è quello di Strang, in specifico anche per YMINRES.

5.5 Tavola 5: Toeplitz([1 1],1,[t t])

$A_n = \text{Toeplitz}([1 \ 1],1,[t \ t])$ con $t = 0.01$.

Numero di iterazioni per metodo:

Precondizionatore	n	YMINRES	GMRES	LSQR
Non preconditionato	10	10	10	20
	100	144	80	316
	500	870	415	1772
	1000	-	861	-
	2000	-	-	-
	5000	-	-	-
	10000	-	-	-
Strang	10	10	5	14
	100	10	5	16
	500	10	5	20
	1000	10	5	22
	2000	12	5	26
	5000	32	10	8
	10000	-	6	8
Ottimale	10	10	10	18
	100	12	11	22
	500	18	12	36
	1000	24	13	44
	2000	28	15	60
	5000	82	19	16
	10000	-	22	16
Superottimale	10	10	10	20
	100	34	20	62
	500	65	34	126
	1000	84	43	194
	2000	112	55	362
	5000	258	77	130
	10000	26	81	82

Numero di condizionamento di A_n :

n	10	100	500	1000	2000	5000	10000
$\kappa(A_n)$	12.4	112	537	$1.10 \cdot 10^3$	$2.21 \cdot 10^3$	$5.52 \cdot 10^3$	$1.10 \cdot 10^4$

Grafici per $n = 1000$

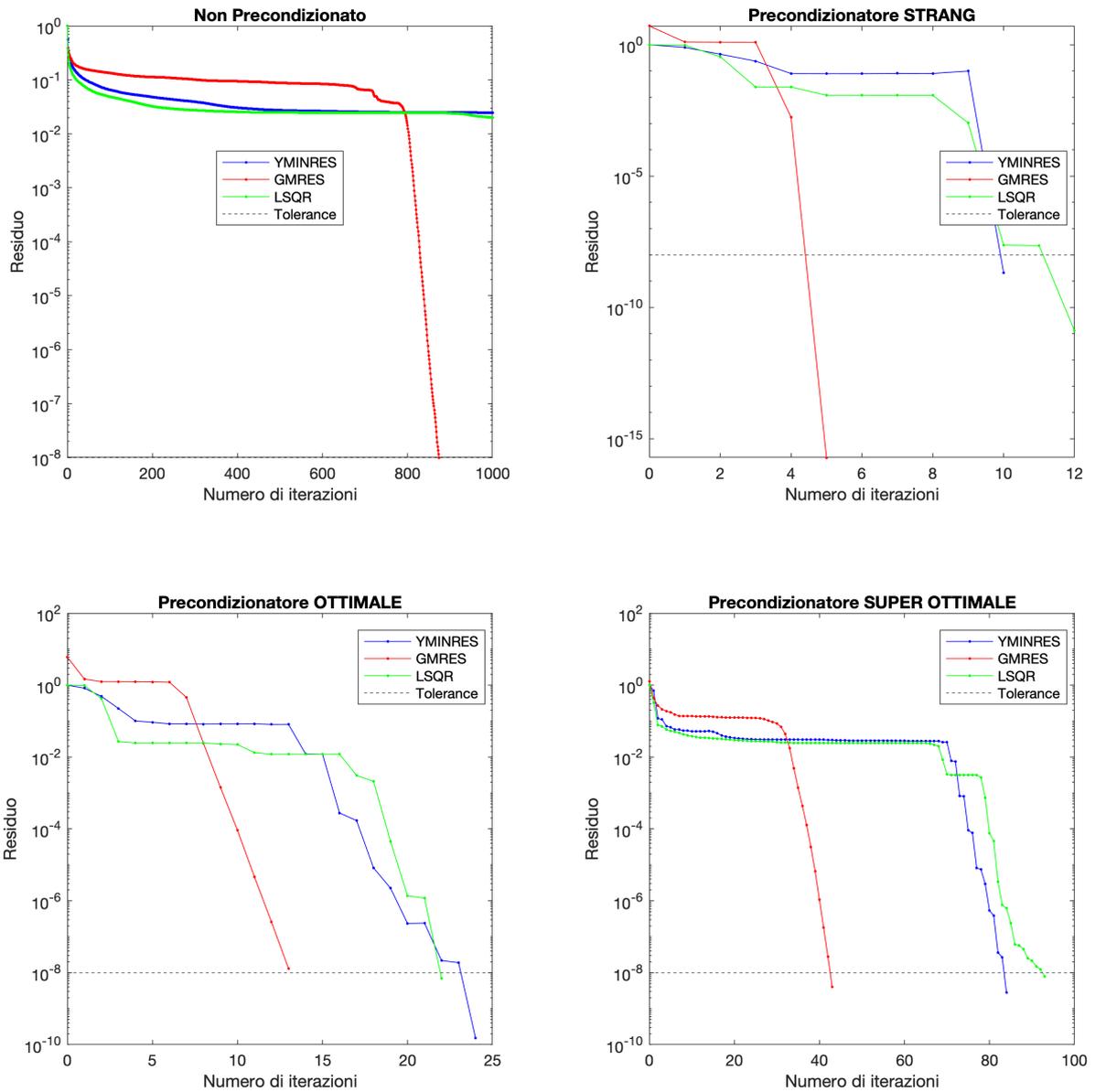


Figura 5.1: Confronto dei diversi metodi, a seconda del preconditionatore usato

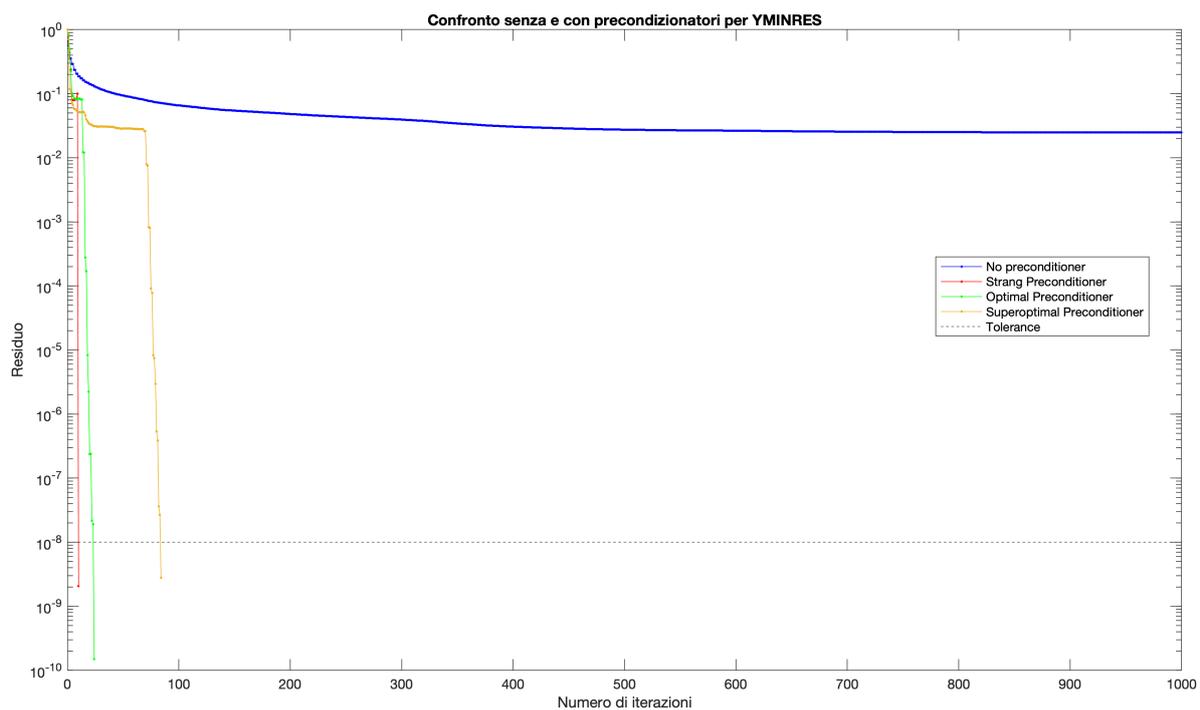


Figura 5.2: Confronto dei diversi preconditionatori per YMINRES

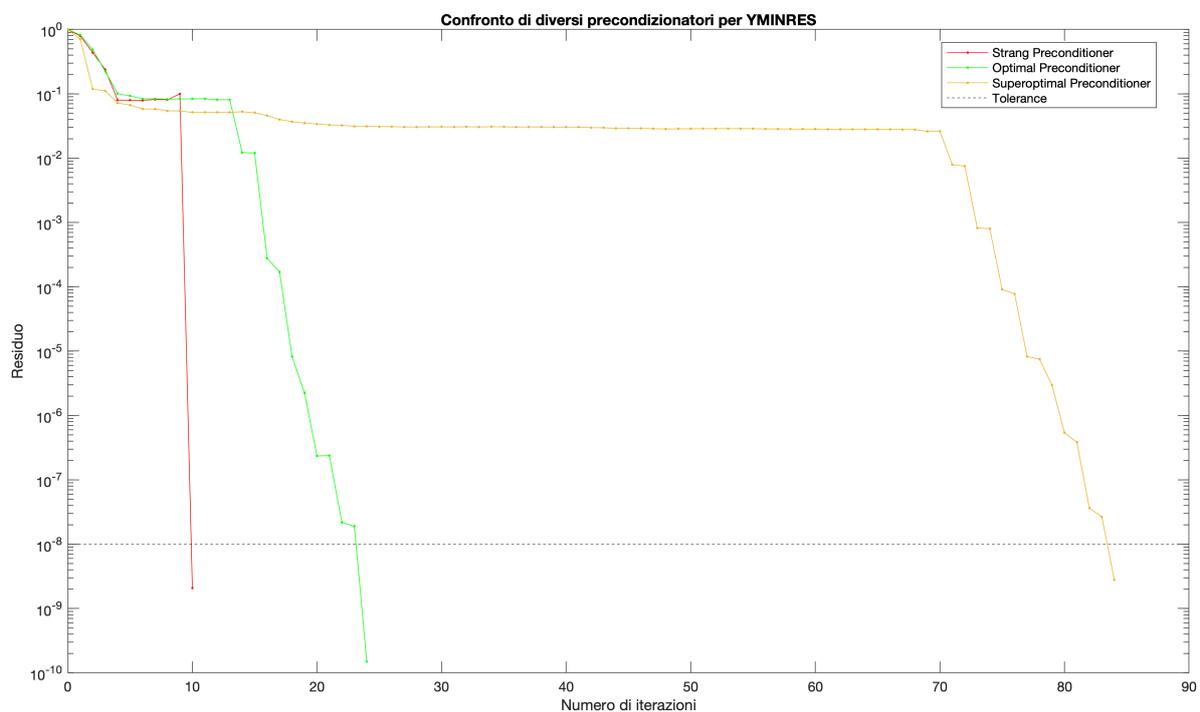


Figura 5.3: Zoom sui tre metodi, tolte le iterazioni senza preconditionatore

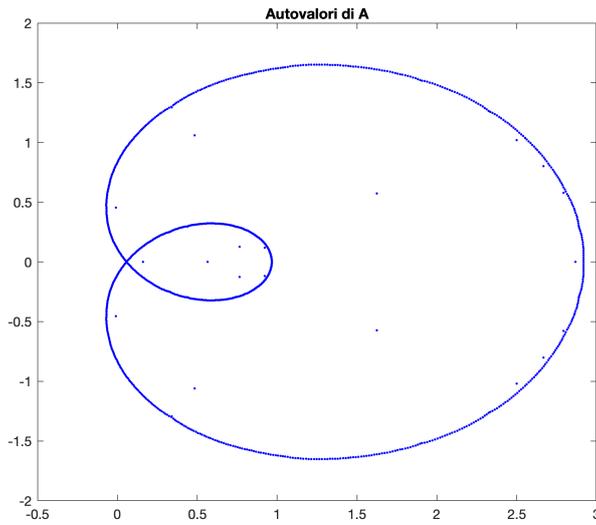


Figura 5.4: Plot degli autovalori di A_{1000}

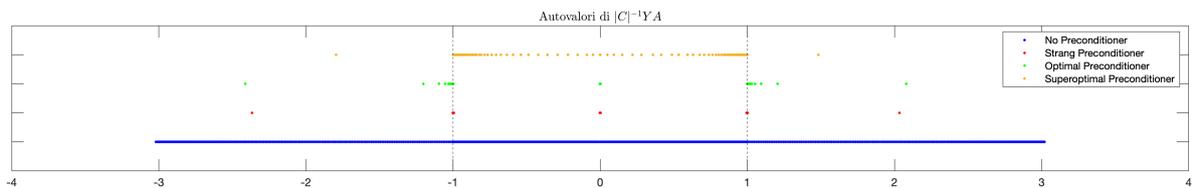


Figura 5.5: Autovalori delle matrici preconditionate $|C|^{-1}A_{1000}$.

Commenti:

- Per $n \geq 2000$ e per $n = 1000$ con YMINRES e LSQR non c'è più convergenza per i metodi non preconditionati.
- Per $n \geq 5000$, le matrici preconditionate $|C|^{-1}A$ hanno un autovalore molto prossimo a 0 (come si può vedere dalla figura 5.5). Per questo motivo, anche se il metodo iterativo si ferma dopo pochi passi, la soluzione trovata è lontana da quella effettiva.
- Come per l'esperimento 4, vediamo per il preconditionatore Strang prima una lenta decrescita, per poi una convergenza a picco estremamente rapida. Similmente per Ottimale e Superottimale, c'è un primo tratto orizzontale in cui il residuo decresce poco, e dopo si arriva a una convergenza molto rapida.
- YMINRES e LSQR hanno bisogno dello stesso numero di iterazioni, ma LSQR richiede anche qui più moltiplicazioni matrice-vettore.
- Notiamo l'andamento "a scalini" di YMINRES, particolarmente evidente nel preconditionatore superottimale.
- Vediamo che il preconditionatore di Strang è il più efficiente tra i tre a disposizione.

5.6 Tavola 6: Toeplitz([1 1 1],1,[t t t])

$A_n = \text{Toeplitz}([1 \ 1 \ 1],1,[t \ t \ t])$ con $t = 0.01$.

Numero di iterazioni per metodo:

Precondizionatore	n	YMINRES	GMRES	LSQR
Non preconditionato	10	10	10	22
	100	164	75	388
	500	-	407	-
	1000	-	838	-
	2000	-	-	-
	5000	-	-	-
	10000	-	-	-
Strang	10	11	6	20
	100	13	7	26
	500	14	7	28
	1000	14	7	32
	2000	17	7	38
	5000	41	11	10
	10000	-	8	10
Ottimale	10	10	10	22
	100	15	12	32
	500	23	14	48
	1000	27	15	60
	2000	34	17	78
	5000	56	23	22
	10000	-	29	18
Superottimale	10	10	10	20
	100	44	25	80
	500	80	42	168
	1000	107	53	272
	2000	153	68	518
	5000	378	96	390
	10000	-	120	190

Numero di condizionamento di A_n :

n	10	100	500	1000	2000	5000	10000
$\kappa(A_n)$	14.7	129	639	$1.28 \cdot 10^3$	$2.55 \cdot 10^3$	$6.37 \cdot 10^3$	$1.27 \cdot 10^4$

Grafici per $n = 1000$

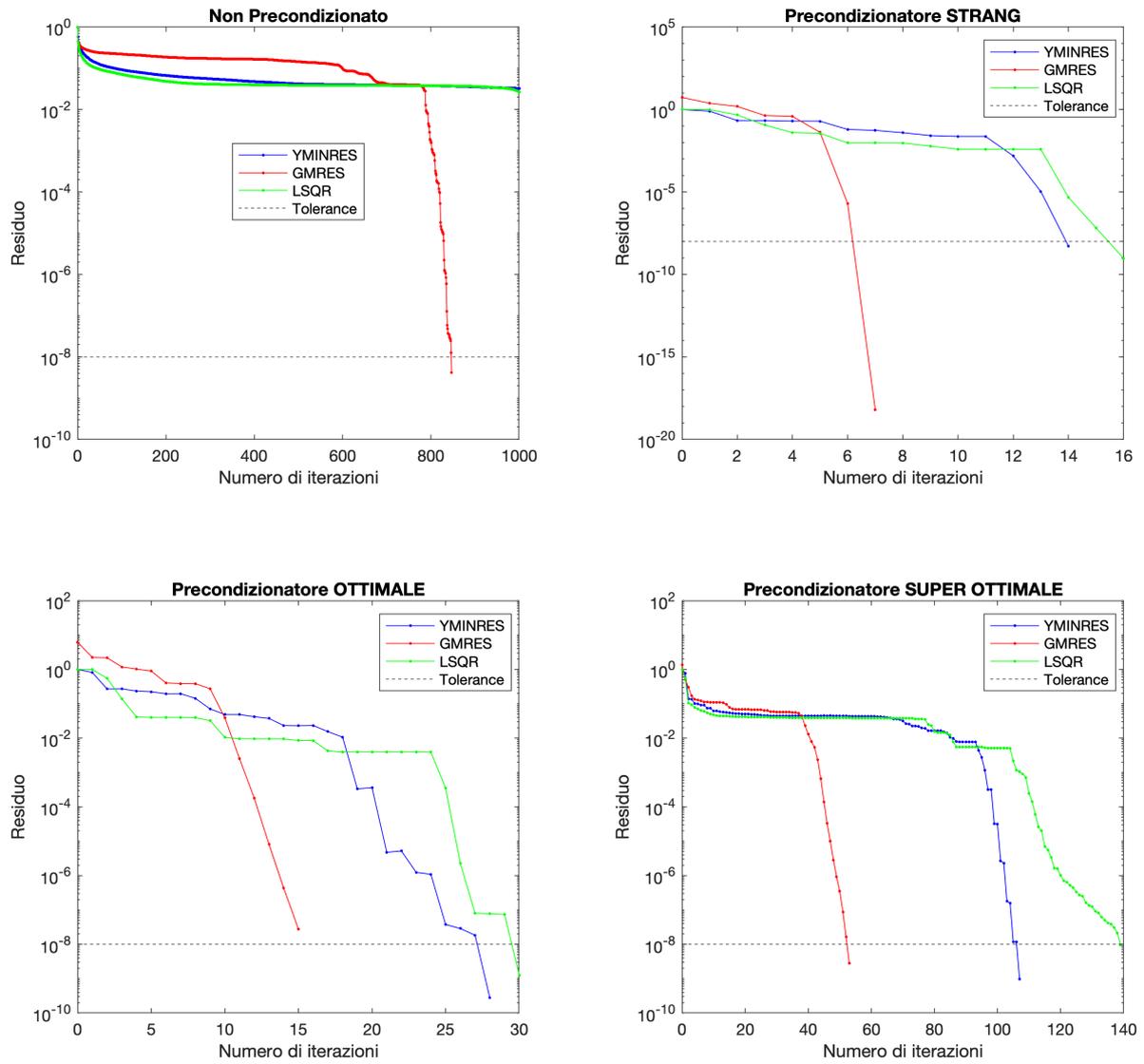


Figura 6.1: Confronto dei diversi metodi, a seconda del preconditionatore usato

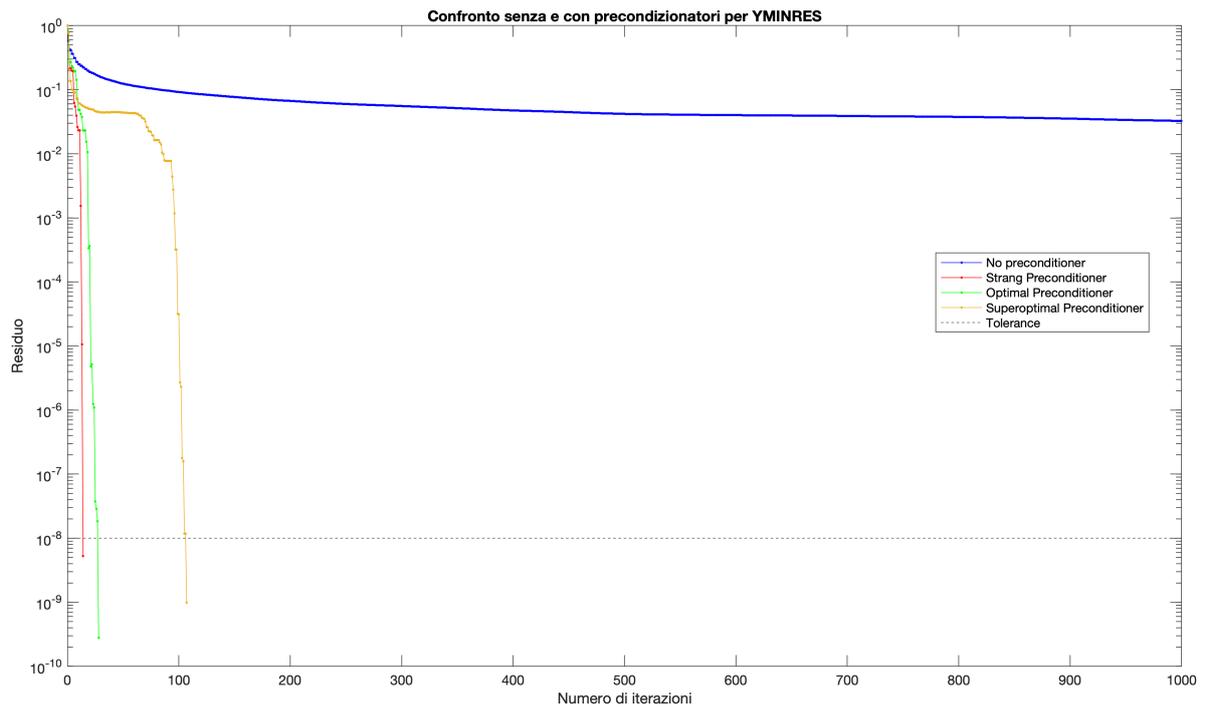


Figura 6.2: Confronto dei diversi preconditionatori per YMINRES

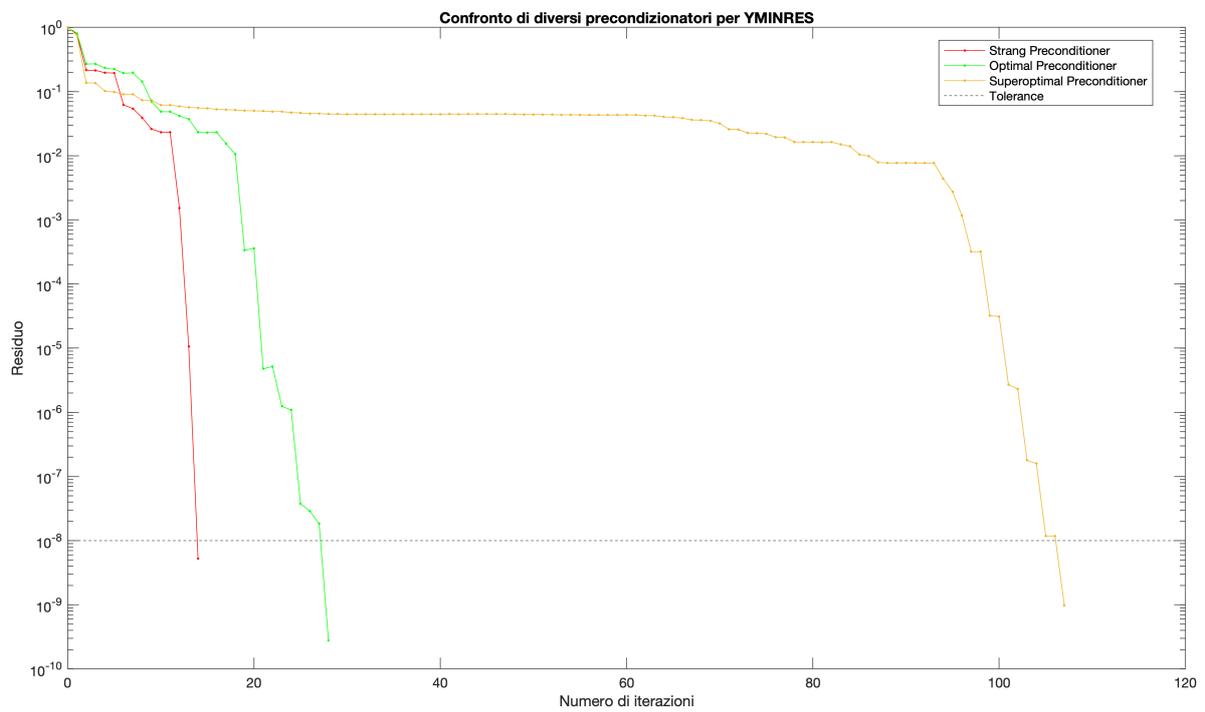


Figura 6.3: Zoom sui tre metodi, tolte le iterazioni senza preconditionatore

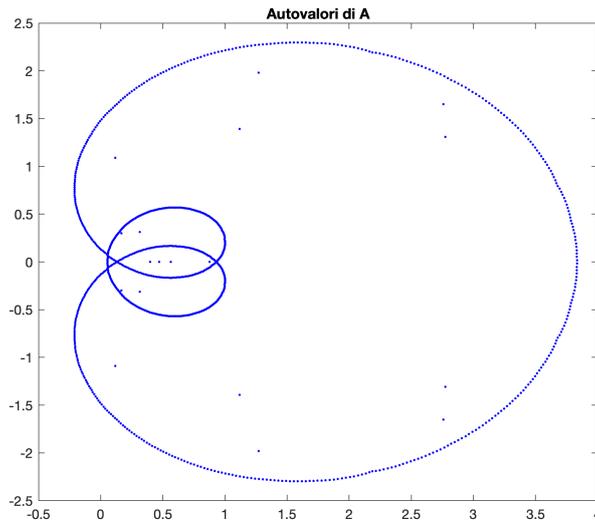


Figura 6.4: Plot degli autovalori di A_{1000}

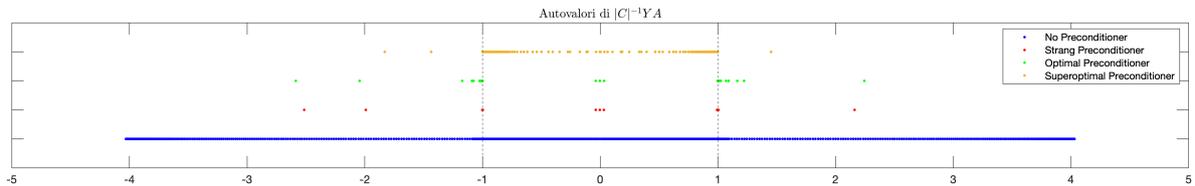


Figura 6.5: Autovalori delle matrici preconditionate $|C|^{-1}A_{1000}$.

Commenti:

- Come prima, per $n \geq 2000$ e per $n = 1000$ con YMINRES e LSQR i metodi preconditionati non convergono entro MAXIT=1000 iterazioni.
- Indifferentemente dal preconditionatore, GMRES è il metodo più efficiente, seguito da YMINRES e infine LSQR. Notiamo che YMINRES necessita meno iterazioni di LSQR.
- Per questa sperimentazione, il divario tra i diversi preconditionatori aumenta. Per $n = 1000$, Strang necessita sempre di 10-15 iterazioni, quello Ottimale di 15-30, mentre quello Superottimale di 50 per GMRES e più di 100 per gli altri due metodi.
- Dalla figura 6.5 vediamo che ci sono alcuni autovalori di $|C|^{-1}YA$ prossimi a 0; questo fa sì che per $n \geq 2000$, i metodi preconditionati non riescano a raggiungere la tolleranza richiesta e che la soluzione calcolata si discosti notevolmente da quella vera.
- In merito al punto sopra, è curioso osservare nella figura 6.1 l'andamento di LSQR con preconditionatore Superottimale: dopo un tratto orizzontale, attorno all'iterazione 110 inizi a convergere rapidamente, per poi diminuire la velocità di convergenza dopo l'iterazione 125. Questo probabilmente è dovuto al mal condizionamento di A e $|C|^{-1}YA$

5.7 Tavola 7: simbolo $f(x) = |x|e^{ix}$

A_n ha entrate $a_k = k$ -esimo coefficiente di Fourier di $f(x)$.

Numero di iterazioni per metodo:

Precondizionatore	n	YMINRES	GMRES	LSQR
Non preconditionato	10	10	10	20
	100	124	100	272
	500	721	500	1470
	1000	-	-	-
	2000	-	-	-
	5000	-	-	-
	10000	-	-	-
Strang	10	10	7	20
	100	16	9	30
	500	17	9	36
	1000	19	9	38
	2000	19	9	38
	5000	19	9	42
	10000	19	9	42
Ottimale	10	10	9	20
	100	18	11	32
	500	19	10	38
	1000	19	10	38
	2000	19	11	36
	5000	19	10	42
	10000	19	10	42
Superottimale	10	10	10	20
	100	28	17	48
	500	51	28	96
	1000	65	34	126
	2000	83	42	178
	5000	121	60	358
	10000	173	77	538

Numero di condizionamento di A_n :

n	10	100	500	1000	2000	5000	10000
$\kappa(A_n)$	21.3	113	499	991	$1.98 \cdot 10^3$	$4.92 \cdot 10^3$	$9.81 \cdot 10^3$

Grafici per $n = 1000$

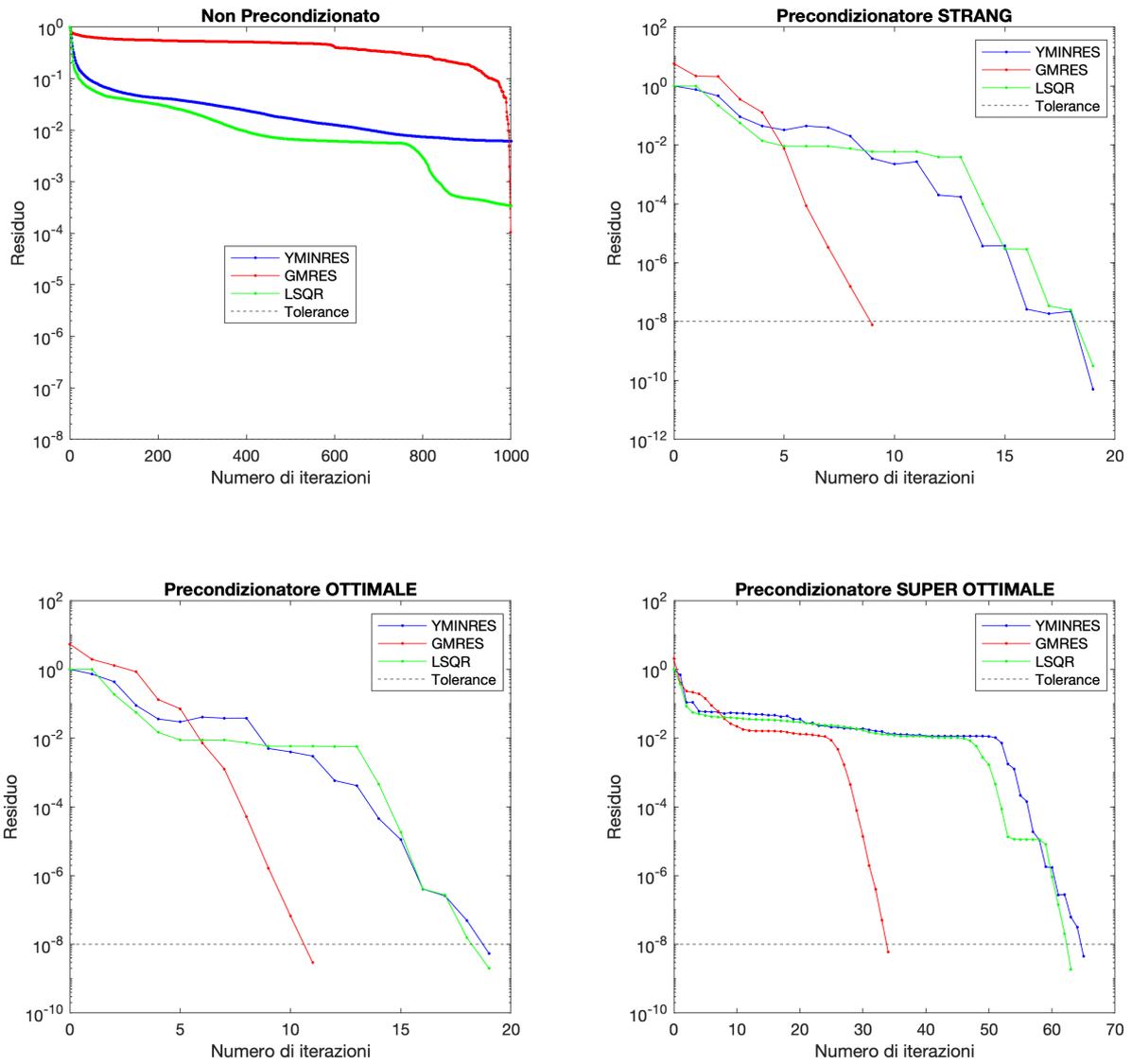


Figura 7.1: Confronto dei diversi metodi, a seconda del preconditionatore usato

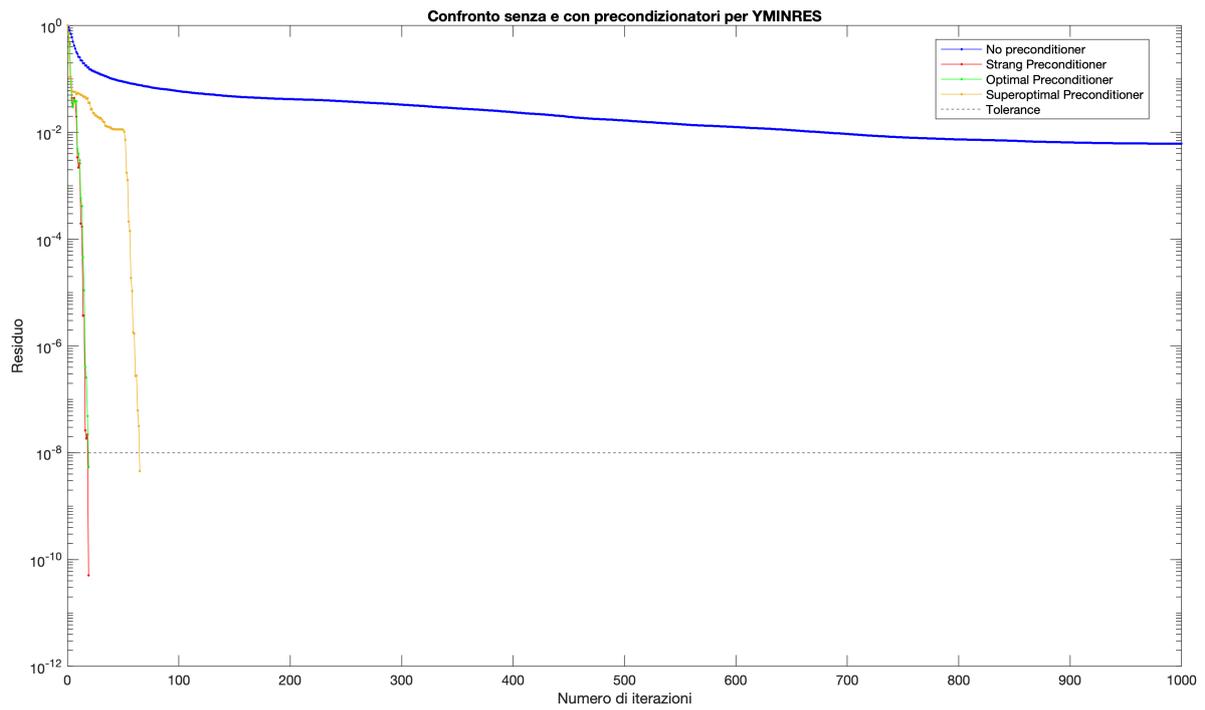


Figura 7.2: Confronto dei diversi preconditionatori per YMINRES

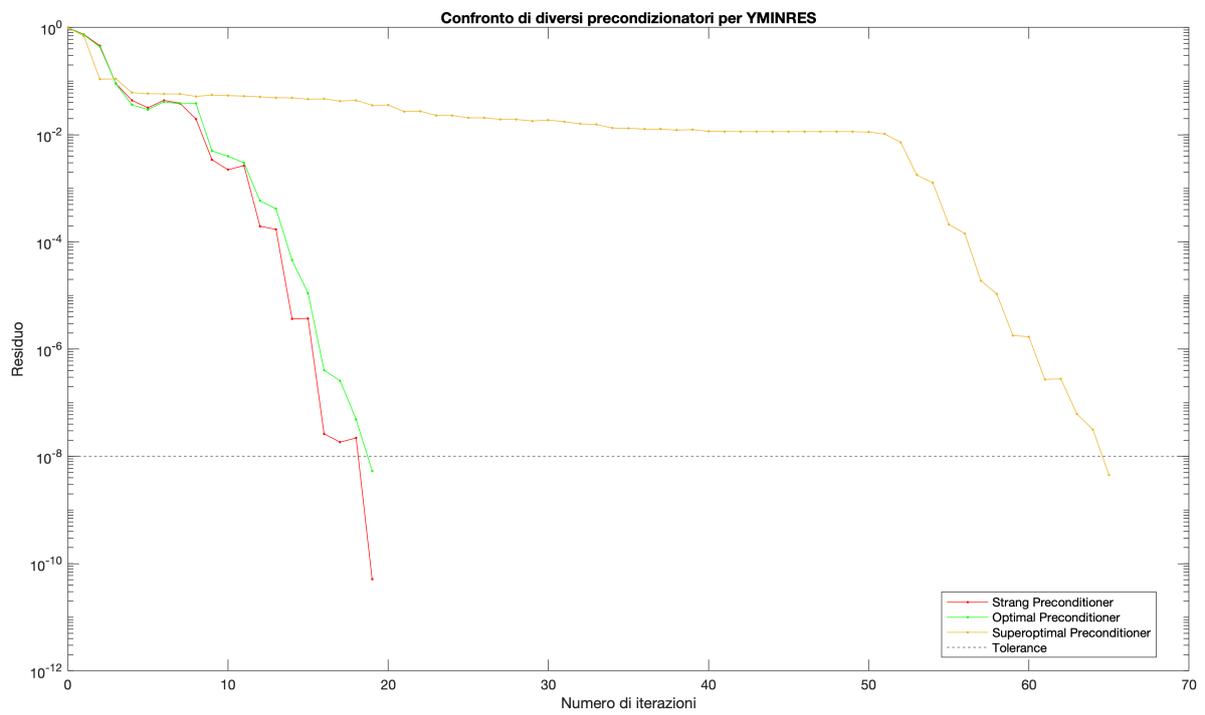


Figura 7.3: Zoom sui tre metodi, tolte le iterazioni senza preconditionatore

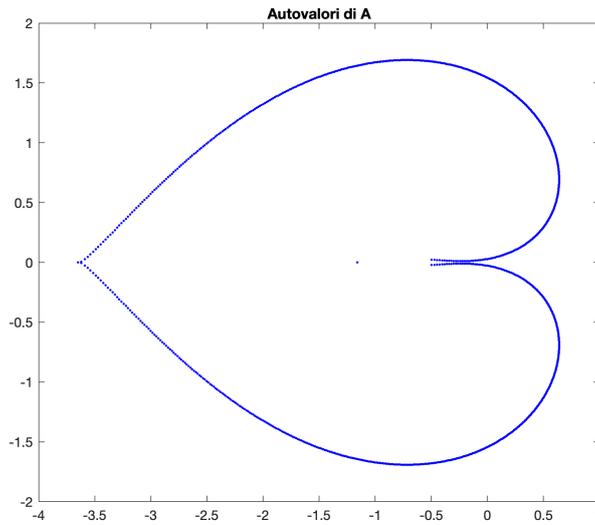


Figura 7.4: Plot degli autovalori di A_{1000}

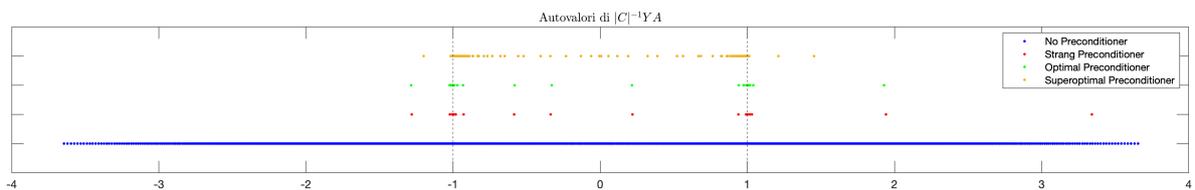


Figura 7.5: Autovalori delle matrici preconditionate $|C|^{-1}A_{1000}$.

Commenti:

- In questa sperimentazione le matrici sono piene, e non sono né troppo ben condizionate, né troppo mal condizionate. Notiamo infatti che a differenza delle 4 tipologie precedenti, i metodi preconditionati riescono a convergere alla soluzione anche per $n = 10000$. Quelli non preconditionati solo fino a $n = 500$.
- In questo caso Strang e Ottimale danno circa lo stesso numero di iterazioni, e hanno comportamento simile. Per entrambi GMRES è il metodo più veloce, mentre YMINRES e LSQR richiedono approssimativamente lo stesso numero di iterazioni.
- Per il preconditionatore Superottimale servono molte più iterazioni. GMRES converge più velocemente degli altri due. Per LSQR si può notare un doppio "salto" quando inizia la convergenza rapida (figura 7.1), mentre per YMINRES si vede l'andamento a scalini.
- In particolare per il metodo YMINRES, i preconditionatori Strang e Ottimale (figura 7.3) hanno pressappoco lo stesso comportamento.

5.8 Calcolo numerico della stima in proposizione 5

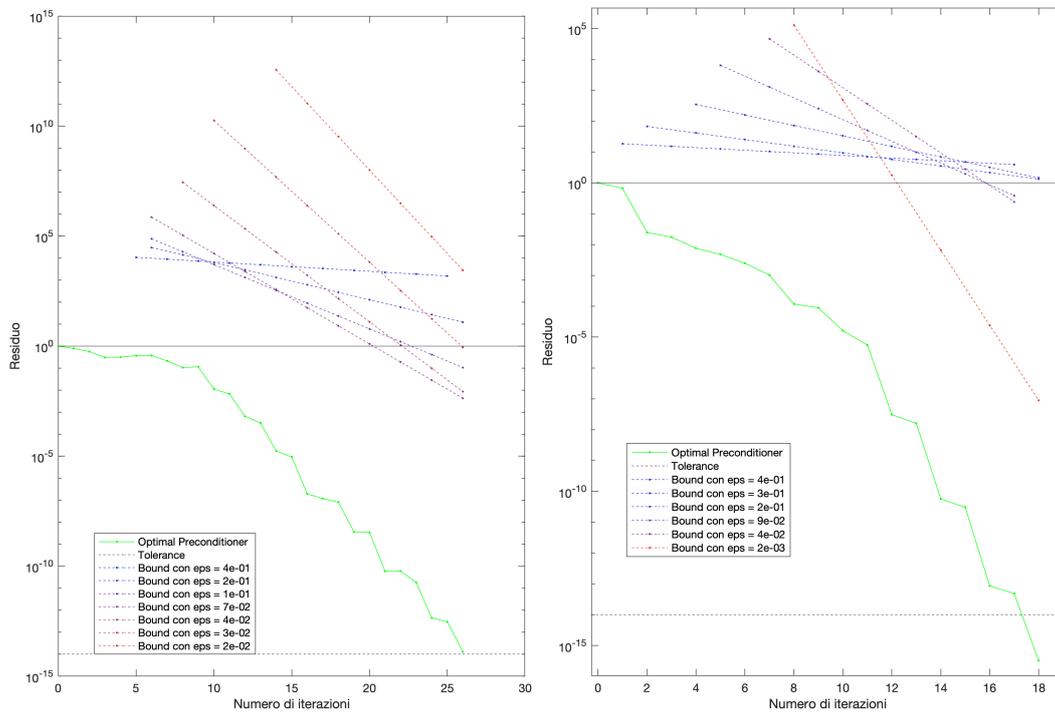


Figura 36: A sinistra, tipo = 7. A destra, tipo = 2

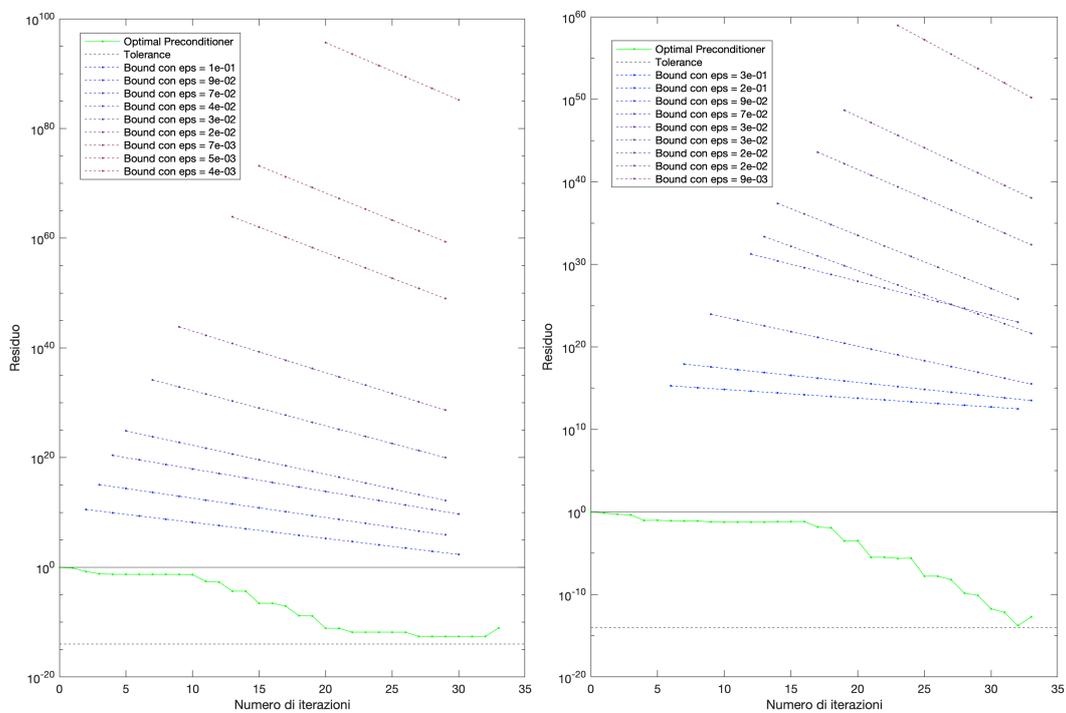


Figura 37: A sinistra, tipo = 4. A destra, tipo = 6.

In tutti e quattro i grafici si è usato $n = 1000$ e una tolleranza $\text{tol}=10^{-14}$ per poter osservare l'andamento per più iterazioni. Tutte le stime proposte sono lineari, e sono molto poco efficaci: in particolare si notino quelli della figura 37, dove le stime variano tra 10^{100} e 10^{10} .

Sorge la domanda perché per questi bound, i valori di ε sono piuttosto grandi, dell'ordine di 10^{-1} , 10^{-2} . Il problema è che diminuendoli ancora, aumenta considerevolmente il numero di outliers, rendendo la stima valida per k molto più grande del numero di iterazioni necessarie alla convergenza.

Per esempio in figura 36 a destra, per tipo =2, l'ultimo bound con $\varepsilon \approx 2 \cdot 10^{-3}$ è il migliore tra quelli qui raffigurati, però è comunque abbastanza grossolano. Diminuendo ε , il numero di outliers aumenta da 9 a diverse centinaia; tuttavia si ha convergenza in soli $k = 18$ passi, per cui tale stima non è nemmeno raffigurabile nel grafico.

Riferimenti bibliografici

- [1] J. Pestana, A.J. Wathen, *A preconditioned MINRES method for nonsymmetric Toeplitz matrices*, SIAM J. Matrix Anal. Appl. Vol. 36, No. 1, pp. 273–288 (2015)
- [2] H. Elman, D. Silvester, A. Wathen, *Finite Elements and Fast Iterative Solvers with Applications in Incompressible Fluid Dynamics*, Oxford University Press, New York, 2005.
- [3] R. H.-F. Chan, X.-Q. Jin, *An Introduction to Iterative Toeplitz Solvers*, SIAM (2007)
- [4] S. L. Campbell, I. C. Ipsen, C. T. Kelley, C. D. Meyer, *GMRES and the minimal polynomial*, BIT Numerical Mathematics Vol. 36, No. 4, pp. 664–675. (1996)
- [5] O. Axelsson, G. Lindskog. *On the eigenvalue distribution of a class of preconditioning methods*, Numer. Math. Vol. 48, 479–498 (1986)