Università degli Studi di Pisa

DIPARTIMENTO DI MATEMATICA
Corso di Laurea Triennale in Matematica

Tesi di laurea triennale

Metodi numerici per il calcolo della funzione segno e della decomposizione polare

Candidato:
Hergert Gjoni

Relatore:
Beatrice Meini

Anno Accademico 2018–2019

Indice

Introduzione

In questo elaborato parleremo di funzione segno e decomposizione polare di matrice, definizioni che daremo rispettivamente su matrici quadrate a elementi complessi di dimensione aventi autovalori non immaginari puri e su matrici complesse invertibili.

La funzione segno viene usata per diverse applicazioni, per esempio per la ricerca di determinati sottospazi invarianti di $\mathbb{C}^{n\times n}$, oppure per trovare il numero di autovalori di una data matrice in determinate regioni del piano complesso. La funzione segno presenta anche dei forti legami con la decomposizione polare di una matrice, altro argomento che tratteremo nella tesi. Anche la decomposizione polare viene usata per diverse applicazioni. Per esempio, data una matrice $A \in \mathbb{R}^{n\times n}$, viene usata per cercare la matrice ortogonale Q più "vicina" ad A, utile per altre applicazioni, come ad esempio cercare soluzioni ortogonali di determinate equazioni differenziali in cui l'incognita è una matrice.

L'obiettivo della tesi è quello di studiare metodi numerici per il calcolo della funzione segno e della decomposizione polare.

Tra i metodi numerici studieremo il metodo di Newton e le famiglie di iterazioni di Padé. Di queste fa parte anche il metodo di Newton, che è uno dei più utilizzati, e sono interessanti perché è possibile costruire successioni con ordine di convergenza locale o globale più alto rispetto all'ordine del metodo di Newton, che ha ordine 2. Tuttavia questi metodi fanno uso di funzioni razionali, quindi valutarle su una matrice A può essere problematico. Esistono infatti diversi approcci sia per quanto riguarda la valutazione di polinomi sia per quanto riguarda la valutazione di funzioni razionali. Questi approcci possono essere diversi, perché una stessa funzione razionale può essere scritta in modi differenti, e a seconda dei casi un approccio può essere più vantaggioso rispetto ad un altro.

Abbiamo detto in precedenza che la funzione segno presenta dei legami con la decomposizione polare, e uno di questi, che è anche quello che ci interessa, è il legame tra i metodi numerici per il calcolo di esse. Infatti i metodi numerici per il calcolo della decomposizione polare vengono ereditati direttamente dai metodi numerici per il calcolo della funzione segno, con lo stesso ordine di convergenza.

Un altro aspetto importante è la stabilità numerica, che non è legata alla velocità di convergenza, ma all'accuratezza delle operazioni. Infatti in tutti i metodi proposti si effettueranno inversioni di matrici o risoluzioni di sistemi lineari multipli, operazioni su cui viene richiesta una buona accuratezza. Usando il metodo di Newton per esempio si effettua un'inversione di matrice su ogni elemento della successione. Se si usa tale metodo per il calcolo della decomposizione polare di una matrice, se l'algoritmo di inversione usato su ognuna delle matrici della successione non è stabile in avanti e all'indietro non si ha la garanzia che il metodo sia stabile all'indietro, o per meglio dire la decomposizione polare effettivamente calcolata potrebbe non essere di buona qualità, soprattutto su matrici malcondizionate.

L'elaborato si suddivide in tre capitoli.

Nel primo capitolo introdurremo le funzioni di matrice, in particolar modo la funzione segno, e la decomposizione polare.

Nel secondo capitolo vedremo i metodi numerici sia per il calcolo della funzione segno sia per il calcolo della decomposizione polare, vedendo con maggiori dettagli ciò che abbiamo appena accennato nell'introduzione.

Nel terzo capitolo mostreremo degli esperimenti numerici applicando il metodo di Newton per calcolare la decomposizione polare di una matrice. Mostreremo tramite gli esperimenti che l'inversione gioca un ruolo fondamentale sull'accuratezza della soluzione, vedremo come se un algoritmo di inversione di matrice ha una proprietà leggermente più debole rispetto alla stabilità in avanti e all'indietro la soluzione ottenuta può allontanarsi in maniera non trascurabile rispetto a quella vera.

Capitolo 1

Funzioni di matrici

1.1 Definizioni

In questo contenuto ci concentreremo sulla famiglia di matrici $\{A \mid A \in \mathbb{C}^{n\times n}\}$, in particolar modo ci concentreremo sulle loro decomposizioni polari e sulle loro funzioni segno.
Inizieremo quindi col dare le prime definizioni, definendo cos'è una funzione di matrice. Data una funzione scalare $f : \Omega \subseteq \mathbb{C} \to \mathbb{C}$ e data una matrice $A \in \mathbb{C}^{n\times n}$ ci sono molti significati differenti per la funzione $f(A)$.
Alcuni esempi (su cui noi non ci concentreremo) sono i seguenti:

- $f(A) = (f(a_{ij}))$, cioè f agisce su ogni elemento di A
- una funzione scalare, cioè $f(A) = \lambda$ con $\lambda \in \mathbb{C}$, ad esempio il determinante
- funzioni che vanno da $\mathbb{C}^{n\times n}$ a $\mathbb{C}^{m\times m}$ che non derivano da funzioni scalari, per esempio la trasposizione, estrazione di un suo minore, oppure un fattore di una sua fattorizzazione.
- f può essere una funzione scalare che va da $\mathbb{C}$ a $\mathbb{C}^{n\times n}$, per esempio $f(t) = B(tI - A)^{-1}C$, dove $A, B, C \in \mathbb{C}^{n\times n}$.

In questo elaborato tratteremo il caso in cui, se $A \in \mathbb{C}^{n\times n}$, allora $f(A) \in \mathbb{C}^{n\times n}$. Diremo che l'applicazione $f : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ è una funzione di matrice.
Esistono varie definizioni equivalenti di funzione di matrice. Ne descriviamo di seguito una basata sulla forma normale di Jordan.
Sappiamo che per ogni $A \in \mathbb{C}^{n\times n}$ esiste ed è unica (a meno di permutazione di blocchi) una relativa forma di Jordan J, cioè esiste $Z \in GL(\mathbb{C})$ tale che $Z^{-1}AZ = J$. Sia m_k il relativo ordine di J_k dove J_k è uno dei blocchi di Jordan dove $J = \mathrm{diag}(J_1, J_2, \ldots, J_p)$ e siano $\lambda_1, \ldots, \lambda_s$ gli autovalori di A, con $s \leqslant n$ naturalmente, e siano $n_1, \ldots, n_s$ i relativi ordini dei più grandi blocchi di Jordan in cui i λ_i appaiono dove $i = 1, \ldots, s$.
Allora affinchè possiamo definire $f(A)$ è necessario che per ogni λ_i esistano i vari $f^{(j)}(\lambda_i)$ con $j = 1, \ldots, n_i$, dove con $f^{(j)}(\lambda)$ indicheremo la derivata j-esima di f calcolata in λ. Detto questo possiamo allora definire $f(A)$ nel seguente modo:

$$f(A) := Zf(J)Z^{-1} = Z\mathrm{diag}(f(J_1), \ldots, f(J_p))Z^{-1}$$

dove

$$f(J_k) := \begin{bmatrix} f(\lambda_k) & f'(\lambda_k) & \ldots & \frac{f^{(m_k-1)}(\lambda_k)}{(m_k-1)!} \\ & f(\lambda_k) & \ddots & \vdots \\ & & \ddots & f'(\lambda_k) \\ & & & f(\lambda_k) \end{bmatrix}$$

Le funzioni di matrice hanno numerose applicazioni, per esempio:

- **Equazioni differenziali**

  Il problema ai valori iniziali scalare classico

  $$\frac{dy}{dt} = ay, \qquad y(0) = c$$

  ha soluzione $y(t) = e^{at}c$, mentre l'analogo problema vettoriale

  $$\frac{d\mathbf{y}}{dt} = A\mathbf{y}, \quad \mathbf{y}(0) = \mathbf{c}, \quad \mathbf{y} \in \mathbb{C}^n, A \in \mathbb{C}^{n\times n},$$

  ha soluzione $\mathbf{y}(t) = e^{At}\mathbf{c}$. Più in generale, la soluzione del sistema non omogeneo

  $$\frac{d\mathbf{y}}{dt} = A\mathbf{y} + f(t, \mathbf{y}), \quad \mathbf{y}(0) = \mathbf{c}, \quad \mathbf{y} \in \mathbb{C}^n, A \in \mathbb{C}$$

  assumendo che f sia sufficientemente regolare, è data da

  $$\mathbf{y}(t) = e^{At}\mathbf{c} + \int_0^t e^{A(t-s)} f(s, \mathbf{y}) ds,$$

  che è una formula esplicita nel caso in cui f non dipende da $\mathbf{y}$.
  Tali formule non sono necessariamente il modo migliore per calcolare le soluzioni numericamente. Esiste una larga letteratura sulle soluzioni numeriche di equazioni differenziali ordinarie che offre tecniche alternative. Tuttavia l'esponenziale di matrice viene usato in certi metodi, in particolare gli *Integratori esponenziali* (Vedere [2, Subsect 2.1.1]).
  Esistono anche equazioni differenziali matriciali le quali hanno soluzioni che possono essere espresse in termini di esponenziali di matrice. Un esempio è dato dalla seguente equazione

  $$\frac{dY}{dt} = AY + YB, \quad Y(0) = C, \quad A, B, C, Y \in \mathbb{C}^{n\times n}$$

  avente soluzione $Y(t) = e^{At}Ce^{Bt}$.

  Sia $A \in \mathbb{C}^{n\times n}$ invertibile e tale che esiste B tale che $B^2 = A$ (B la chiameremo $\sqrt{A}$). Un'equazione della forma

  $$\frac{d^2\mathbf{y}}{dt^2} + A\mathbf{y} = \mathbf{0}, \quad \mathbf{y}(0) = \mathbf{y}_0, \quad \mathbf{y}'(0) = \mathbf{y}'_0$$

  ha soluzione

  $$\mathbf{y}(t) = \cos(\sqrt{A}t)\mathbf{y}_0 + (\sqrt{A})^{-1}\sin(\sqrt{A}t)\mathbf{y}'_0$$

- **Teoria del controllo**, che fa uso della funzione segno, funzione che vedremo meglio nei dettagli nella prossima sezione.

- **Il problema agli autovalori non-simmetrico**, anche in questo problema viene usata la funzione segno.

- **Problema di ortogonalizzazione**, che fa uso della decomposizione polare, che definiremo in seguito. (La decomposizione polare non è considerata una funzione di matrice, ma ha dei forti legami con la funzione segno).

- **Altre applicazioni**

È possibile trovare ulteriori applicazioni e maggiori dettagli in [2, Cap.2].

1.2 Funzione segno e decomposizione polare

Nel caso scalare, dato $z \in \mathbb{C} \setminus \{\text{Re } z = 0\}$ definiamo la funzione segno

$$\text{sign}(z) := \begin{cases} 1 & \text{se Re}(z) > 0 \\ -1 & \text{se Re}(z) < 0 \end{cases}$$

Dunque,utilizzando la definizione di funzione di matrice, se $A = ZJZ^{-1}$ e $J = \text{diag}(J_-, J_+)$ dove J_- è formato dai blocchi di Jordan relativi agli autovalori con parte reale negativa, e J_+ è formato da quelli relativi agli autovalori con parte reale positiva allora la funzione segno di A è definita nel seguente modo:

$$\text{sign}(A) := Z \begin{bmatrix} -I_p & \\ & I_q \end{bmatrix} Z^{-1}$$

dove p è l'ordine di J_-, q è l'ordine di J_+.
Osservazione: La funzione segno scalare non è definita sugli elementi immaginari puri di $\mathbb{C}$, quindi affinchè la funzione segno possa essere ben definita su A è necessario che gli autovalori di A non siano immaginari puri.

Possiamo elencare alcune proprietà di tale matrice.
Sia $S = \text{sign}(A)$, allora:

- $S^2 = I$
- S è diagonalizzabile con autovalori ± 1
- $SA = AS$
- Se A è reale allora S è reale
- $(I + S)/2$ e $(I - S)/2$ sono proiezioni sui sottospazi invarianti associati agli autovalori con parte reale positiva e con parte reale negativa.

Motivazioni della funzione segno:
Come già deto nelle sezione precedente la funzione segno ha varie applicazioni, come per esempio la **Teoria del controllo**, e il **problema agli autovalori non-simmetrico**.

- **Problema agli autovalori non-simmetrico**

  La funzione segno di matrice può essere utilizzata per determinare quanti autovalori di una data matrice A si trovano in particolari regioni del piano complesso ed ottenere i corrispondenti sottospazi invarianti.
  Sappiamo dalle formule precedenti che la traccia di $\text{sign}(A)$ è uguale a $p-q$, dove sappiamo che $p + q = n$. Riusciamo quindi a dire immediatamente quanti autovalori si trovano sul semipiano sinistro e quanti vi si trovano sul semipiano destro.
  Abbiamo inoltre già detto che le colonne delle matrici $(I + S)/2$ e $(I - S)/2$ generano i corrispondenti spazi invarianti.
  Possiamo anche cercare il numero di autovalori in regioni più complicate del piano complesso. Ciò è possibile effettuando opportune sequenze di valutazioni di funzione segno di matrice,e usando tecniche come lo **spectrum splitting via sign function** (vedi [2, Teorema 2.1]), oppure effettuando valutazioni della funzione segno su opportune matrici.
  Per esempio il numero di autovalori di A che si trovano nell'insieme $\{z \in \mathbb{C} \mid \text{Re } z \in (\xi_1, \xi_2)\}$ è dato da $\frac{1}{2}\text{trace}(\text{sign}(A - \xi_1 I) - \text{sign}(A - \xi_2 I))$.
  Per maggiori dettagli si veda [1, Sect 2.5].

Introdotta la funzione segno possiamo adesso definire la decomposizione polare di una matrice.

Teorema 1.1. *Sia $A \in \mathbb{C}^{n\times n}$ invertibile. Allora esistono e sono uniche U_A unitaria e H_A hermitiana definita positiva tale che $A = U_A H_A$.*

Essa è detta **decomposizione polare** di A. Ciò che possiamo dire subito è che se si conosce U_A allora è $H_A = U_A^* A$, dove con U_A^* indicheremo la trasposta coniugata di U_A.
Esistono delle relazioni che intercorrono tra la funzione segno e la decomposizione polare, e se siamo in grado di calcolare la prima siamo anche in grado di calcolare la matrice U_A della decomposizione polare.
Un legame fondamentale tra la funzione segno di matrice e la decomposizione di una matrice $A \in \mathbb{C}^{n\times n}$ è dato dalla seguente relazione:

$$\operatorname{sign}\left(\begin{bmatrix} 0 & A \\ A^* & 0 \end{bmatrix} \right) = \begin{bmatrix} 0 & U_A \\ U_A^* & 0 \end{bmatrix}$$

(si veda [2, Teorema 5.2]).
Tale relazione non è però l'unico legame, esiste infatti un ulteriore legame che enunceremo nel prossimo capitolo, che ci permetterà, utilizzando alcuni metodi numerici per il calcolo della funzione segno, di trovare metodi numerici anche per il calcolo della decomposizione polare di una matrice.

Motivazioni della decomposizione polare:
In alcune applicazioni una matrice che dovrebbe essere ortogonale, per via di errori di macchina, risulta non essere ortogonale. Un approccio per risolvere il problema è quello di applicare l'ortogonalizzazione di Gram-Schmidt o, equivalentemente, calcolare il fattore ortogonale in una fattorizzazione QR. Un' alternativa , "ortogonalizzazione ottimale", è quella di rimpiazzare A con la matrice ortogonale che le si avvicina di più. Più in generale, sia $A \in \mathbb{C}^{n\times n}$ $(m \geq n)$ e definiamo la distanza della matrice più vicina ad A con colonne ortonormali

$$\min\{||A - Q|| : Q^*Q = I\}.$$

Per la norma-2 e la norma di Frobenius una Q ottimale è U_A nella decomposizione polare $A = U_A H_A$ (si veda [2, Teorema 8.4]).
Un vantaggio del fattore unitario U_A rispetto al fattore Q della fattorizzazione QR è che se $A \to W_1 A W_2$ è una trasformazione unitaria (cioè W_1, W_2 sono unitarie) allora $W_1 U_A W_2$ è il fattore unitario della decomposizione polare della nuova matrice. Infatti $W_1 A W_2 = W1 U_A H_A W_2 = W_1 U_A W_2 W_2^* H_A W_2$, quindi $W_1 A W_2 = (W_1 U_A W_2)(W_2^* H_A W_2)$ è la decomposizione polare di tale matrice.
La riortogonalizzazione ottimale viene usata in determinate applicazioni. Per esempio se ne fa uso se si cerca una soluzione numerica ortogonale di una data equazione differenziale

$$Y'(t) = F(t, Y(t)), \quad Y(0)Y^T(0) = I,$$

dove $Y(t) \in \mathbb{R}^{n\times n}$ e $Y(t)^T Y(t) = I$ per tutti i $t > 0$.
Per maggiori dettagli si veda [2, Sect 2.6].

Capitolo 2

Metodi numerici

2.1 Metodi numerici per la funzione segno

2.1.1 Metodo di Newton

Uno dei metodi più usati per il calcolo della funzione segno è il metodo di Newton.

$$X_{k+1} = \frac{1}{2}(X_k + X_k^{-1}), \quad k \geq 0 \qquad X_0 = A. \tag{2.1}$$

Tale successione si ottiene applicando il metodo di Newton all'equazione $X^2 - I = 0$.

Definizione: Diremo che una norma di matrice $||\cdot||$ è consistente con una norma vettoriale $||\cdot||'$ se per ogni matrice A e per ogni vettore colonna $\mathbf{x}$ vale $||A\mathbf{x}||' \leq ||A||||\mathbf{x}||'$.

Definizione: Diremo che una norma di matrice $||\cdot||$ è consistente se esiste una norma vettoriale $||\cdot||'$ tale che $||A\mathbf{x}||' \leq ||A||||\mathbf{x}||'$.

Il seguente teorema ci garantisce la convergenza della successione $\{X_k\}_{k\in\mathbb{N}}$ (si veda [1, pag. 113]).

Teorema 2.1. *Sia $A \in \mathbb{C}^{n\times n}$ avente autovalori non immaginari puri (quindi sarà anche invertibile). Allora (2.1) converge a $S = sign(A)$ in maniera quadratica con*

$$||X_{k+1} - S|| \leqslant \frac{1}{2}||X_k^{-1}||||X_{k+1} - S||^2 \tag{2.2}$$

per ogni norma consistente. Inoltre, per $k \geqslant 1$,

$$X_k = (I - G_0^{2^k})^{-1}(I + G_0^{2^k})S, \qquad dove \;\; G_0 = (A - S)(A + S)^{-1} \tag{2.3}$$

Dalla relazione (2.3) $X_k \longrightarrow S$ con $k \longrightarrow \infty$ se $G_0^{2k} \longrightarrow 0$, e infatti la convergenza c'è, perché gli autovalori di G_0 sono della forma $(\lambda - sign(\lambda))/(\lambda + sign(\lambda))$ con λ autovalore di A che, essendo non immaginari puri, hanno modulo minore di 1.
Poichè per ogni norma consistente $||G_0^{2^k}|| \geqslant \rho(G_0^{2^k})$, dove con $\rho(G)$ indichiamo il raggio spettrale di G, allora possiamo dire subito che se $\rho(A) \gg 1$ oppure se un autovalore si trova nei pressi dell'asse immaginario di $\mathbb{C}$ allora la convergenza sarà lenta.
Viene fatto uso allora del metodo di Newton con scaling.
Viene cioè introdotto un parametro μ_k positivo tale che la nuova formula sia la seguente:

$$X_{k+1} = \frac{1}{2}(\mu_k X_k + \mu_k^{-1} X_k^{-1}), \quad k \geq 0 \qquad X_0 = A. \tag{2.4}$$

Poichè μ_k è positivo allora la funzione segno di ogni iterazione viene preservata.
I principali parametri μ_k che vengono utilizzati sono i seguenti:

$$\text{determinantal scaling: } \mu_k = \mid \det(X_k) \mid^{-1/n} \tag{2.5}$$

$$\text{spectral scaling: } \mu_k = \sqrt{\rho(X_k^{-1})/\rho(X_k)} \tag{2.6}$$

$$\text{norm scaling: } \mu_k = \sqrt{||X_k^{-1}||/||X_k||} \tag{2.7}$$

dove $||\ .\ ||$ è un'opportuna norma.
Tutti e tre i parametri proposti permettono allo stesso modo di accelerare la convergenza iniziale. Tuttavia ognuno di essi ha dei vantaggi e degli svantaggi.
Per esempio una proprietà interessante dello spectral scaling è che se gli autovalori di A sono reali allora l'iterazione (2.4) convergerà a sign(A) in un numero finito di passi. Per tale scaling il metodo può essere svantaggioso se gli autovalori si raggruppano vicino all'asse immaginario.
Invece il determinantal scaling porta gli autovalori vicino alla circonferenza unitaria, in questo modo il raggio spettrale non sarà molto maggiore di 1, però potrebbe non funzionare bene se esiste un piccolo gruppo di autovalori che sono lontani dal punto di convergenza, mentre gli altri sono invece vicini.
Poichè $X_k \longrightarrow S$ allora in tutti i casi $\mu_k \longrightarrow 1$, questo vuol dire che il metodo di Newton con scaling non fa saltare la convergenza quadratica. È ragionevole porre $\mu_k = 1$ quando l'errore sarà sufficientemente vicino a 0.

2.1.2 Iterazioni di Padé

Il metodo di Newton non è l'unico metodo usato per il calcolo della funzione segno di una matrice A. Esiste in realtà una grande varietà di famiglie di metodi, con approcci differenti e con proprietà e ordini di convergenza diversi. Noi in questo testo ci concentreremo sulla **famiglia di iterazioni di Padé**.

Per una data funzione scalare $f(x)$ la funzione razionale $r_{km}(x) = p_{km}(x)/q_{km}(x)$, dove p_{km} è un polinomio di grado k e q_{km} è un polinomio di grado m, è una $[k/m]$ approssimante di Padé di f se $q_{km}(0) = 1$, e $f(x) - r_{km}(x) = O(x^{k+m+1})$.
Se l'approssimante $[k/m]$ di Padé esiste allora è unica. È di solito richiesto che p_{km} e q_{km} non abbiano zeri in comune, così i due polinomi sono unici.

Premesso questo, dato $z \in \mathbb{C}$ esso vale $\text{sign}(z) = \dfrac{z}{(z^2)^{1/2}}$ dove $(\mu)^{1/2}$ con $\mu \in \mathbb{C}$ è la radice di μ tale che la sua parte reale sia positiva. Tale radice non sempre esiste ma vale un risultato generale non solo per scalari, ma anche per matrici:

Teorema 2.2 (Radice quadrata principale)**.** *Sia $A \in \mathbb{C}^{n\times n}$ con autovalori che non appartengono all'insieme $\mathbb{R}^-$. Esiste allora un'unica matrice $X \in \mathbb{C}^{n\times n}$ tale che $X^2 = A$ e i suoi autovalori sono tali che la loro parte reale sia strettamente positiva. X sarà chiamata la radice quadrata principale di A e scriveremo $X = A^{1/2}$. Se A è reale allora $A^{1/2}$ è reale.*

Per la dimostrazione del teorema si veda [1, Teorema 1.29].

Poiché z non è immaginario puro allora si vede facilmente che z^2 non apparterrà mai a $\mathbb{R}^-$, dunque $(z^2)^{1/2}$ esiste.
Possiamo dunque scrivere

$$\text{sign}(z) = \frac{z}{(z^2)^{1/2}} = \frac{z}{(1-(1-z^2))^{1/2}} = \frac{z}{(1-t)^{1/2}}$$

dove $t = 1 - z^2$. Dunque, per approssimare $\text{sign}(z)$ bisogna approssimare

$$h(t) = (1-t)^{-1/2}$$

Si dimostra che p_{km} e q_{km} che approssimano $h(t)$ sono:

$$p_{km} = {}_2F_1(-k, \tfrac{1}{2} - m; -k - m, t)$$
$$q_{km} = {}_2F_1(-m, \tfrac{1}{2} - k; -k - m, t)$$

dove ${}_2F_1(a, b; c, t)$ è la hypergeometric function di Gauss:

$${}_2F_1(a, b; c, t) = \sum_{j=0}^{\infty} \frac{(a)_j (b)_j}{j!(c)_j} t^j$$

e $(a)_j = a(a+1)...(a+j-1)$, $(a)_0 = 1$, a, b, c sono interi. È facile vedere che se $a < 0$, $j > |a|$ allora $(a)_j = 0$, ed esattamente si verifica facilmente che

$${}_2F_1(-m, b; c, t) = \sum_{j=0}^{m} (-1)^j \binom{m}{j} \frac{(b)_j}{(c)_j} t^j$$

quindi i due polinomi hanno grado esattamente k ed m.

Osservazione In entrambe le formule $c = -k - m$, quindi poiché in entrambe le formule $j \leq \max(k, m)$ allora $(-k-m)_j$ sarà diverso da zero, dunque i polinomi sono ben definiti.

Sulla base di ciò, l'idea di Kenney e Laub è di introdurre la famiglia di iterazioni

$$x_{k+1} = f_{\ell m}(x_k) := x_k \frac{p_{\ell m}(1 - x_k^2)}{q_{\ell m}(1 - x_k^2)}, \qquad x_0 = a \tag{2.8}$$

La versione matriciale diventa ovviamente

$$X_{k+1} = X_k (p_{\ell m}(1 - X_k^2))(q_{\ell m}(1 - X_k^2))^{-1}, \qquad X_0 = A \tag{2.9}$$

Le iterazioni (2.9) saranno chiamate *iterazioni di Padè*, in particolar modo, quando $\ell = m$ oppure $\ell = m - 1$ saranno chiamate *iterazioni Principali di Padé.*

Le iterazioni principali di Padé hanno delle interessanti proprietà. Sia infatti $g_r := f_{\ell m}$ con $r = \ell + m + 1$, allora g_r ha le seguenti proprietà.

(a) $g_r(x) = \dfrac{(1+x)^r - (1-x)^r}{(1+x)^r + (1-x)^r}$

(b) $g_r(x) = \tanh(r \operatorname{arctanh}(x))$

(c) $g_r(g_s(x)) = g_{rs}(x)$

(d) g_r segue l'espressione

$$g_r(x) = \frac{2}{r} {\sum_{j=0}^{\lceil \frac{r-2}{2} \rceil}}' \frac{x}{\sin^2\left(\frac{(2j+1)\pi}{2r}\right) + \cos^2\left(\frac{(2j+1)\pi}{2r}\right) x^2} \tag{2.10}$$

dove il simbolo *prime* indica che l'ultimo termine della sommatoria viene dimezzato quando r è dispari.

Si veda [1, Teorema 5.9].
Le iterazioni di Padé fanno uso di funzioni razionali. Per la valutazione di queste funzioni su una data matrice A esistono diversi approcci, in cui si fa uso di valutazioni di funzioni polinomiali, inversioni e risoluzioni di sistemi lineari multipli (si veda [1, Capitolo 4])), e ognuno di essi può essere più o meno efficiente a seconda dei casi. Nel caso delle iterazioni principali di Padé le quattro proprietà elencate possono rendere ancora più efficiente la valutazione di funzione: per esempio la proprietà (c) può essere utile nel caso in cui r non sia primo. Infatti generalmente al crescere di r la funzione $g_r(x)$ assume forme più complicate, con numeratore e denominatore di grado sempre più alto, quindi se $r = r_1 r_2$ valutare $g_{r_2}(A)$ e successivamente $g_{r_1}(g_{r_2}(A))$ può essere più efficiente rispetto al valutare direttamente $g_r(A)$.
Una caratteristica interessante della proprietà (d) è che l'espressione (2.10) comprende $\lceil \frac{r-2}{2} \rceil$ inversioni di matrice indipendenti (oppure sistemi lineari multipli indipendenti), che possono quindi essere eseguite in parallelo. Per maggiori dettagli si veda [1, Sect 5.4].

Un altro risultato interessante sulle approssimanti di Padé è che data $f(z) \in \mathbb{C}$ se p/q è la $[k/m]$ approssimante di Padé di f, se $f(0) \neq 0$ allora q/p è la $[m/k]$ approssimante di Padé di $1/f(z)$. (Vedere[3, p.473]).
Osserviamo che un altro modo di scrivere sign(z) è

$$\operatorname{sign}(z) = \frac{(z^2)^{1/2}}{z} = \frac{(1-t)^{1/2}}{z}$$

Con $t = 1 - z^2$.
Di conseguenza possiamo approssimare la funzione $g(t) = (1-t)^{1/2}$ con la funzione razionale $q_{km}(t)/p_{km}(t)$.
Introduciamo dunque una nuova famiglia di metodi:

$$x_{k+1} = \frac{p_{\ell m}(1 - x_k^2)}{x_k q_{\ell m}(1 - x_k^2)}, \qquad x_0 = a \tag{2.11}$$

Le iterazioni (2.11) le chiameremo *Reciprocal Padé iterations.*
La versione matriciale diventa ovviamente

$$X_{k+1} = q_{\ell m}(I - X_k^2)(X_k p_{\ell m}(I - X_k^2))^{-1}, \quad k \geq 0 \quad X_0 = A. \tag{2.12}$$

La funzione segno di una matrice A è un caso particolare della *p-sector function* di A definita come

$$\operatorname{sect}_p(A) = A(A^p)^{-1/p}$$

dove $(A)^{1/p}$ è la p-esima radice principale di A. (Guardare [1, Capitolo 7]) Un metodo iterativo proposto per il calcolo di $\operatorname{sect}_p(A)$ é dato dalla seguente successione:

$$Z_{k+1} = Z_k Q_{\ell m}(I - Z_k^{-p})(P_{\ell m}(I - Z_k^{-p})^{-1}, \ k \geq 0 \quad Z_0 = A \tag{2.13}$$

dove $P_{\ell m}(t)/Q_{\ell m}(t)$ è la $[\ell/m]$ approssimante di Padé della funzione $(1-t)^{-1/p}$, e inoltre vale

che $P_{\ell m}(t) = {}_2F_1(-\ell, \frac{1}{p} - m; -\ell - m, t)$, $Q_{\ell m}(t) = {}_2F_1(-m, \frac{1}{p} - \ell; -\ell - m, t)$.

Per maggiori dettagli si veda [4]. Noi ci concentreremo sul caso $p = 2$, per la quale la successione diventa

$$Z_{k+1} = Z_k q_{\ell m}(I - Z_k^{-2})(p_{\ell m}(I - Z_k^{-2})^{-1}, \quad Z_0 = A \tag{2.14}$$

dove $p_{\ell m}$ e $q_{\ell m}$ le abbiamo già definite in precedenza. Le iterazioni come la (2.13) vengono chiamate *Dual Padé iterations.*
Per $k = 0, m = 1$ l'iterazione duale di Padé è proprio il metodo di Newton (si veda [2]).
Le famiglie su cui effettueremo le nostre analisi saranno la (2.9),(2.12),(2.14).

2.1.3 Convergenza delle iterazioni di Padé

Studieremo adesso le proprietà di convergenza delle iterazioni di Padé che abbiamo introdotto per la funzione segno.

Enunciamo il seguente teorema ([1, pag.116]).

Teorema 2.3 (covergenza delle iterazioni di Padé)**.** *Sia $A \in \mathbb{C}^{n\times n}$ avente autovalori non immaginari puri. Consideriamo l'iterazione (2.9) con $\ell + m > 0$ e ogni norma di matrice subordinata.*

- *Per $\ell \geq m-1$, se $||I - A^2|| < 1$ allora $X_k \longrightarrow sign(A)$ con $k \longrightarrow \infty$ e $||I - X_k^2|| < ||I - A^2||^{(\ell+m+1)^k}$.*
- *Per $\ell = m-1$ e $\ell = m$*

$$(S - X_k)(S + X_k)^{-1} = [(S - A)(S + A)^{-1}]^{(\ell+m+1)^k}$$

  e quindi $X_k \longrightarrow sign(A)$ con $k \longrightarrow \infty$.

Il teorema 2.3 dunque ci dice che per $\ell = m-1$ e $\ell = m$ la successione (2.9) converge globalmente, mentre per $\ell \geq m+1$ converge localmente. In entrambi i casi l'ordine di convergenza sarà $\ell + m + 1$.

Per quanto riguarda iterationi Duali di Padé invece possiamo enunciare il seguente teorema enunciato nell'articolo [2, Corollario 2.2].

Teorema 2.4. *Sia $A \in \mathbb{C}^{n\times n}$ avente autovalori non immaginari puri. Consideriamo l'iterazione (2.14) con $\ell + m > 0$ e ogni norma di matrice subordinata. Se A è tale che $||I - A^{-2}|| < 1$ allora $Z_k \longrightarrow sign(A)$ con $k \longrightarrow \infty$ e inoltre $||I - Z_k^{-2}|| < ||I - A^{-2}||^{(\ell+m+1)}$.*

Questo teorema ci garantisce quindi una convergenza locale di ordine $\ell + m + 1$ delle iterazioni Duali di Padé.

Possiamo però dire qualcosa di più. Infatti se $\ell = m-1$ l'iterazione (2.14) e l'iterazione (2.11) sono i reciproci dell'iterazione(2.9), cioè le rispettive successioni sono generate dalla funzione razionale reciproca di quella che genera la sucessione (2.9). Inoltre con $\ell = m$ (2.14) coincide proprio con (2.9).
Sempre per $\ell = m$ la sequenza dell'iterazione (2.11) si suddivide in due sottosequenze:

- La prima sotto-sequenza $\{X_{2k}\}$ coincide con la sotto-sequenza $\{Z_{2k}\}$ generata dall'iterazione (2.9).
- La seconda sotto-sequenza $\{X_{2k+1}\}$ coincide con il reciproco della sotto-sequenza $\{Z_{2k+1}\}$ generata sempre dall'iterazione (2.9).

Si veda [2, Prop 2.2].

Grazie a ciò possiamo dire che per $\ell = m-1$ (2.14) e (2.11) convergono globalmente a sign(A). Per $\ell = m$ (2.14) converge globalmente a sign(A), e abbiamo convergenza globale anche per l'iterazione (2.11), poichè le relative sotto-sequenze convergono globalmente a sign(A).
Determinare regioni di convergenza locale per le Reciprocal Padé iterations (2.11) con generici ℓ ed m è ancora un problema aperto.

2.2 Metodi numerici per la decomposizione polare

Come abbiamo già detto nel Capitolo 1 siamo in grado di mettere in relazione la funzione segno e la decomposizione polare di una matrice. Possiamo vedere che ci è possibile ereditare dei metodi iterativi per il calcolo del fattore U_A della matrice A direttamente dai metodi iterativi relativi a $\text{sign}(A)$.
Questo ce lo permette il seguente teorema [1, Teorema 8.13]:

Teorema 2.5. *Sia $A \in \mathbb{C}^{n\times n}$ di rango n avente la decomposizione polare $A = U_A H_A$. Sia g una qualsiasi funzione della forma $g(X) = Xh(X^2)$ (dove h sarà una funzione) tale che l'iterazione $X_{k+1} = g(X_k)$ converge a $sign(X_0)$ per $X_0 = H_A$ con ordine di convergenza m. Assumiamo che g abbia la proprietà che $g(X)^* = g(X^*)$, Allora la successione*

$$Y_{k+1} = Y_k h(Y_k^* Y_k), \quad k \geq 0 \quad Y_0 = A \tag{2.15}$$

converge a U_A con ordine di convergenza m.

Osserviamo che la funzione segno di H_A esiste, perché essa è definita positiva, non avrà quindi autovalori immaginari puri. Grazie a questo teorema è possibile derivare dei metodi iterativi di Padé per ottenere U_A che avranno lo stesso ordine di convergenza dei corrispondenti metodi per il calcolo della funzione segno.

2.2.1 Metodo di Newton

Grazie al teorema (2.5) un metodo che possiamo proporre immediatamente è il metodo di Newton. L'iterazione sarà la seguente:

$$X_{k+1} = \frac{1}{2}(X_k + X_k^{-*}), \quad X_0 = A. \tag{2.16}$$

Tale successione convergerà a U_A in modo quadratico. Infatti seguirà la relazione

$$||X_{k+1} - U_A|| \leq \frac{1}{2}||X_k^{-1}||||X_k - U_A||^2 \tag{2.17}$$

Si veda [1, Teorema 8.12].

Per le stesse ragioni esposte nel caso della funzione segno, cioè quelle di accelerare la convergenza, viene applicato lo scaling al metodo di Newton, cioè si giunge alla seguente iterazione:

$$X_{k+1} = \frac{1}{2}(\mu_k X_k + \mu_k^{-1} X_k^{-*}), \quad k \geq 0 \quad X_0 = A \tag{2.18}$$

I tre scalings che vengono considerati sono i seguenti:

$$\text{optimal scaling:} \quad \mu_k^{\text{opt}} = (\sigma_1(X_k)\sigma_n(X_k))^{-1/2} \tag{2.19}$$

$$1,\infty\text{-norm scaling:} \quad \mu_k^{1,\infty} = \left(\frac{||X_k^{-1}||_1||X_k^{-1}||_\infty}{||X_k||_1||X_k||_\infty}\right)^{1/4} \tag{2.20}$$

$$\text{Frobenius norm scaling:} \quad \mu_k^F = \left(\frac{||X_k^{-1}||_F}{||X_k||_F}\right)^{1/2} \tag{2.21}$$

Dove i $\sigma_i(X_k)$ sono i valori singolari della matrice X_k.

Ognuno di questi tre scalings ha una sua particolarità:

Si verifica per esempio che l'optimal scaling è tale che $\kappa_2(X_{k+1}) \leq \kappa_2(X_k)^{1/2}$, dove $\kappa(A)$ è il numero di condizionamento di una matrice A, e ovviamente κ_2 è relativo alla norma-2 di una matrice.
Senza lo scaling, si vedrebbe che $\kappa_2(X_{k+1}) \approx \frac{1}{2}\kappa_2(X_k)$, che significa κ_2 si riduce molto più lentamente.
Inoltre si dimostra che con l'optimal scaling vale la seguente disuguaglianza:

$$||U_A - X_k||_2 \ \leq \ g^{(k)}\Big(\sigma_1(A)/\sigma_n(A)\Big) - 1, \quad g(x) = \frac{1}{2}(\sqrt{x} + 1/\sqrt{x}) \tag{2.22}$$

dove $g^{(i)}$ sarà la i-esima composizione di g.
Questo ci permetterà di stimare il numero di iterazioni necessarie.

Si verifica anche che $\frac{1}{n^{1/4}}\mu_k^{\text{opt}} \leq \mu_k^{1,\infty} \leq n^{1/4}\mu_k^{\text{opt}}$.
Una proprietà della Frobenius norm scaling è che minimizza $||X_{k+1}||_F$ tra tutte le scelte possibili di μ_k.
Per maggiori dettagli vi rimandiamo a [1, pag.205].

2.2.2 Iterazioni di Padé

Sempre grazie al Teorema 2.5 possiamo proporre le seguenti iterazioni di Padé.

- Padé iteration:

$$X_{k+1} = X_k p_{\ell m}(I - X_k^* X_k) q_{\ell m}(I - X_k^* X_k)^{-1}, \quad k \geq 0 \quad X_0 = A \tag{2.23}$$

- Dual Padé iteration:

$$X_{k+1} = X_k q_{\ell m}(I - X_k^{-1} X_k^{-*})(p_{\ell m}(I - X_k^{-1} X_k^{-*}))^{-1}, \quad k \geq 0 \quad X_0 = A \tag{2.24}$$

- Reciprocal Padé iteration:

$$X_{k+1} = q_{\ell m}(I - X_k^* X_k)(X_k p_{\ell m}(I - X_k^* X_k))^{-1}, \quad k \geq 0 \quad X_0 = A \tag{2.25}$$

Analizzeremo in seguito le proprietà di convergenza di queste 3 famiglie di iterazioni.

Attenzione! Gli scalings possono essere applicati anche alle Padé iterations (2.9), in particolare i μ_k dati da (2.19),(2.20),(2.21) sono applicabili alle Principal Padé iterations. Ci sono però degli inconvenienti:

- L'uso di questi scalings richiede il calcolo di X_k^{-1}, che nelle Padé iterations non compaiono.
- Esperimenti numerici mostrano che lo scaling influisce negativamente sulla stabilità numerica delle Padé iterations.

2.2.3 Convergenza delle iterazioni di Padé

Grazie alle proprietà di convergenza che abbiamo enunciato nella sezione 2.1.3, ci è possibile enunciare anche delle proprietà di convergenza relative alle iterazioni di Padé per la decomposizione polare.
Grazie al teorema (2.3) possiamo enunciare subito il seguente corollario (vedere [1, Corollario 8.14]):

Corollario 2.1. *Sia $A \in \mathbb{C}^{n\times n}$ invertibile, e sia $A = U_A H_A$ la sua decomposizione polare. Consideriamo l'iterazione (2.23) con $\ell + m > 0$ ed ogni norma subordinata.*

- *Per $\ell \geq m-1$, se $||I - A^*A|| < 1$ allora $X_k \longrightarrow U_A$ con $k \longrightarrow \infty$ e $||I - X_k^* X_k|| < ||I - A^*A||^{(\ell+m+1)^k}$.*
- *Per $\ell = m-1$ e $\ell = m$,*

  $$(I - H_k)(I - H_k)^{-1} = [(I - H)(I + H)^{-1}]^{(\ell+m+1)^k},$$

  dove $X_k = U_A H_k$ è la decomposizione polare relativa a X_k, e quindi $X_k \longrightarrow U_A$ con $k \longrightarrow \infty$ con ordine di convergenza $\ell + m + 1$.

Analogamente a come abbiamo fatto prima possiamo enunciare il segeunte corollario (Vedi [2, Corollario 3.3])

Corollario 2.2.

- *Sia $X_0 = A \in \{X \in \mathbb{C}^{n\times n} \ : \ ||I - (X^*X)^{-1}|| < 1\}$. Allora la Dual Padé iteration (2.24) converge a U_A.*
- *Sia $\ell = m$ e $\ell = m-1$ con $m \geq 1$. Allora la convergenza della Dual Padé iteration (2.24) e della Reciprocal Padé iteration (2.25) è globale.*

Capitolo 3

Analisi della stabilità e sperimentazione numerica

3.1 Stabilità numerica del metodo di Newton

Come abbiamo detto in precedenza, il metodo di Newton ha bisogno per ogni iterazione di un'inversione di matrice.
Supponiamo di voler invertire la nostra matrice X e che $G = fl(X^{-1})$ sia la nostra matrice inversa effettivamente calcolata con aritmetica floating point.
Assumiamo che sia $G = (X + \Delta_X)^{-1} + \Delta_G$ dove per qualche norma $||\cdot||$

$$||\Delta_X|| \leq w_X(n)u||X||_2 \qquad ||\Delta_G|| \leq w_G(n)u||G||_2 \tag{3.1}$$

con $w_X(n)$ e $w_G(n)$ funzioni che dipendono solo da n (per esempio gradi di polinomi bassi), e $u > 0$ un parametro molto piccolo, (ad esempio la precisione di macchina). Si assume che $w_X(n)u$ e $w_G(n)u$ siano multipli moderati di u. Insomma si richiede che l'inversione sia numericamente stabile in avanti e all'indietro.
Si dimostra che sotto questa ipotesi il metodo di Newton con scaling è stabile all'indietro (si veda [1]).Inoltre, gli esperimenti numerici mostrano esempi per cui la stabilità all'indietro non è garantita senza questa condizione su G.
Supponiamo che su Δ_X e su Δ_G valgano le seguenti condizioni:

$$||\Delta_X||_F \leq \varepsilon_X||X + \Delta_X||_2, \qquad ||\Delta_G||_F \leq \varepsilon_G||(X + \Delta_X)^{-1}||_2 \tag{3.2}$$

dove ε_X e ε_G saranno dell'ordine della precisione di macchina, allora essendo ε_X e ε_G minori di 1 vale che

$$||\Delta_X||_F \leq \frac{\varepsilon_X}{1 - \varepsilon_X}, \text{ e } ||\Delta_G||_F \leq \frac{\varepsilon_G}{1 - \varepsilon_G}$$

Possiamo quindi dire che la (3.1) e la (3.2) sono praticamente equivalenti.
Osserviamo inoltre che essendo ε_X e ε_G molto piccoli possiamo dire che $\frac{\varepsilon_X}{1 - \varepsilon_X} \approx \varepsilon_X$ e $\frac{\varepsilon_G}{1 - \varepsilon_G} \approx \varepsilon_G$.
Possiamo adesso dare la seguente definizione:

Definizione: Sia `Inv` un algoritmo di inversione di matrice, sia data una matrice X e sia G la sua inversa calcolata con `Inv`. Diremo che `Inv` è:

- Numericamente stabile per X (NS) se

$$||G - X^{-1}||_F \leq \varepsilon \text{ cond}_2(X)||G||_2 \tag{3.3}$$

- È left-residual stabile per X (LRS) se

$$||GX - I||_F \leq \varepsilon ||G||_2 ||X||_2 \tag{3.4}$$

- È right-residual stabile per X (RRS) se

$$||XG - I||_F \leq \varepsilon ||G||_2 ||X||_2 \tag{3.5}$$

- È numericamente corretto per X (NC) se esistono Δ_X e Δ_G tali che $G = (X + \Delta_X)^{-1} + \Delta_G$ tale che

$$||\Delta_X||_F \leq \varepsilon_X ||X||_2, \qquad ||\Delta_G||_F \leq \varepsilon_G ||G||_2 \tag{3.6}$$

In queste disuguaglianze, $\varepsilon, \varepsilon_X, \varepsilon_G$ saranno molto piccoli, dell'ordine di precisione di macchina. La (3.6) vuol dire che l'algoritmo è stabile in avanti e all'indietro.

Definizione: Diciamo che `Inv` è

- `Alt` per X se è LRS oppure RRS per X
- `Conj` per X se è sia LRS che RRS per X

Enunciamo adesso la seguente
Proprietà: Date due matrici X e Y vale che

$$\sigma_j(XY) \leq \sigma_j(X)\sigma_1(Y) \tag{3.7}$$

dove con $\sigma_j(X)$ indichiamo i valori singolari della matrice X.

Questo ci porta immediatamente alla seguente relazione:

$$||XY||_F \leq ||X||_F ||Y||_2. \tag{3.8}$$

Inoltre data una matrice quadrata B si verifica facilmente che $||B||_F = ||B^*||_F$ e $||B||_2 = ||B^*||_2$. Dunque vale anche la seguente relazione:

$$||XY||_F = ||(XY)^*||_F = ||Y^*X^*||_F \leq ||Y^*||_F ||X^*||_2 = ||Y||_F ||X||_2. \tag{3.9}$$

Grazie a queste due proprietà è possibile esprimere le due seguenti relazioni:

$$||GX - I||_F \leq \text{cond}_2(X)||XG - I||_F, \qquad ||XG - I||_F \leq \text{cond}_2(X)||GX - I||_F. \tag{3.10}$$

Se vale che $\varepsilon_X + \varepsilon_G + \varepsilon_X\varepsilon_G \leq \varepsilon$ e $\varepsilon||X||_2||G||_2 < 1$ valgono le seguenti implicazioni:

$$\text{NC} \Longrightarrow \texttt{Conj} \Longrightarrow \texttt{Alt} \Longrightarrow \text{NS} \tag{3.11}$$

Se inoltre $\text{cond}_2(X)$ è piccolo allora NS implica NC. Infatti va dimostrato che esistono Δ_X e Δ_G che seguano la relazione (3.6).
Poniamo allora $\Delta_X = 0$ e $\Delta_G = G - X^{-1}$.
Poiché vale la proprietà NS allora $||\Delta_G||_F \leq \varepsilon \text{cond}_2(X)||G||_2 \equiv \varepsilon_G||G||_2$. Dunque se $\text{cond}_2(X)$ è piccolo allora la proprietà NC è verificata.

3.2 Esperimenti numerici

In questa sezione mostriamo degli esperimenti numerici per mostrare che se un algoritmo `Inv` usato per eseguire il metodo di Newton con scaling non segue la proprietà (3.6) (cioè la proprietà `NC`) su ogni matrice X_ℓ della successione ottenuta con il metodo di Newton con scaling con $X_0 = A$ allora l'approssimazione di $\{U_A, H_A\}$ può non essere buona.

Eseguiremo gli esperimenti facendo uso del seguente risultato:

$$||X_\ell - U_A||_2 \leq ||X_\ell - X_\ell^{-*}||_2,$$

dove X_ℓ è la ℓ-esima iterazione del metodo di Newton con scaling (2.18) (si veda [2]).

Definiamo $\{\tilde{X}_\ell\}_{\ell=0}^{s}$ la sequenza di iterazioni usando il metodo (2.18) calcolata in aritmetica floating-point.
Il fattore polare unitario $\tilde{U}$ sarà $\tilde{U} = \tilde{X}_s$, il fattore polare Hermitiano lo calcoliamo nel seguente modo:

$$\tilde{B} = \tilde{U}^* A \qquad \tilde{H} = \frac{1}{2}(\tilde{B} + \tilde{B}^*)$$

Calcoliamo dunque il fattore $\tilde{U}$ applicando ad una data matrice A il metodo di Newton con scaling scegliendo quale degli scalings sopra elencati usare.
Definiamo

$$\beta_\ell = ||\tilde{X}_\ell - G_\ell^*||_F \tag{3.12}$$

dove G_ℓ è l'inversa calcolata di $\tilde{X}_\ell$.
Se $\beta_\ell \leq 1.5$ allora si pone $\mu_\ell = 1$, inoltre se $\beta_\ell \leq \sqrt{2}\nu_d^{1/2} n^{1/4}$ fermiamo il processo, dove $\nu_d \approx 2.2 \times 10^{-16}$.

Si verifica facilmente che se U_A è il fattore polare unitario della decomposizione polare di A allora lo sarà anche per ogni X_ℓ ottenuta dall'iterazione (2.18).

Dunque per ogni ℓ la decomposizione polare della matrice X_ℓ sarà $X_\ell = U_A H_\ell$, dove H_ℓ è Hermitiana e definita positiva. Per $\ell = 0$ H_ℓ sarà proprio H_A.

Dunque, una volta calcolato $\tilde{U}$ esso sarà il fattore unitario(approssimato) relativo a tutti gli $\tilde{X}_\ell$ calcolati durante le iterazioni. Possiamo allora definire $\tilde{H}_\ell = \frac{1}{2}(\tilde{U}^*\tilde{X}_\ell + \tilde{X}_\ell^*\tilde{U})$, che sarà il fattore Hermitiano calcolato per $\tilde{X}_\ell$, cioè $\tilde{U}\tilde{H}_\ell$ è la decomposizione polare di $\tilde{X}_\ell$ calcolata in aritmetica floating point ($\tilde{H}_0$ sarà proprio $\tilde{H}$).

Possiamo ora introdurre le seguenti definizioni:

- $\delta_\ell = \dfrac{||\tilde{X}_\ell - \tilde{U}\tilde{H}_\ell||_F}{||\tilde{X}_\ell||_2}$
- $e_\ell^{(L)} = \dfrac{||I - G_\ell\tilde{X}_\ell||_F}{||\tilde{X}_\ell||_2||G_\ell||_2}$
- $e_\ell^{(R)} = \dfrac{||I - \tilde{X}_\ell G_\ell||_F}{||\tilde{X}_\ell||_2||G_\ell||_2}$

δ_ℓ rappresenterà l'accuratezza di tale decomposizione, dunque indicherà per ogni $\ell = s, s-1, ..., 0$ se essa è una buona decomposizione oppure no. Osserviamo che in particolar modo δ_0 rappresenterà la qualità della decomposizione polare ottenuta di A.
$e_\ell^{(L)}$ e $e_\ell^{(R)}$ sono gli errori relativi (destro e sinistro), che indicheranno se l'algoritmo di inversione `Inv` usato rispetta per ogni $\tilde{X}_\ell$ le proprietà `LRS` e `RRS` introdotte in precedenza.
Osserviamo che $e_\ell^{(L)}$,$e_\ell^{(R)}$ possono essere calcolati durante le iterazioni dell'esecuzione del metodo, mentre per calcolare δ_ℓ è necessaria sia la matrice $\tilde{U}$ che la matrice $\tilde{H}_\ell$, che si potranno ottenere entrambe solo dopo aver eseguito il metodo.

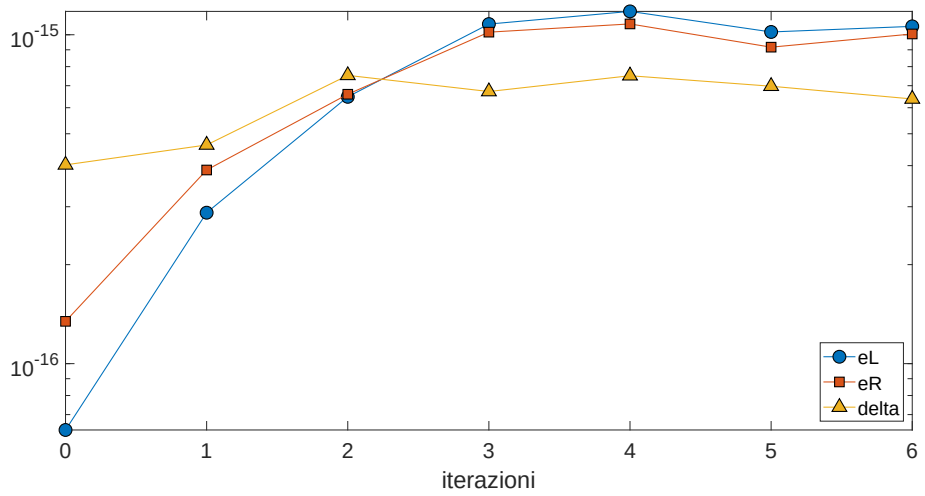
Nei nostri esperimenti noi ci interesseremo proprio a queste grandezze. Una volta trovato $\tilde{U}$ all'iterazione s ($\tilde{U} = \tilde{X}_s$) sappiamo che ℓ varia da 0 a s. Noi ci concentreremo su questi valori al variare di ℓ. Se per qualche ℓ risulta che $e_\ell^{(L)}$ non è sufficientemente piccolo vuol dire che l'algoritmo `Inv` non è stabile a sinistra per $\tilde{X}_\ell$, e possiamo dire lo stesso anche per $e_\ell^{(R)}$. Se per qualche ℓ δ_ℓ non è sufficientemente piccolo vuol dire che la coppia $\{\tilde{U}, \tilde{H}_\ell\}$ non è una buona decomposizione polare approssimata per la matrice $\tilde{X}_\ell$, in particolare per $\ell = 0$ vorrebbe dire che la decomposizione polare di A non é buona.
Abbiamo già detto che se l'algoritmo di inversione `Inv` ha la proprietà `NC` allora il metodo di Newton con scaling è stabile all'indietro, quindi la decomposizione polare calcolata è ben accurata.
Osserviamo che la proprietà `NC` è una proprietà molto forte che si possa dare ad un algoritmo di inversione di una matrice: se X è una matrice e G è la sua inversa effettivamente calcolata con un algoritmo `Inv` che ha la proprietà `NC` allora G è un'inversa leggermente perturbata di una leggera perturbazione di X. Nella gerarchia descritta dalla (3.11) è la proprietà più forte.
La (3.11) indica una catena di proprietà di un algoritmo `Inv` che sono una più debole dell'altra.Abbiamo già detto che se il numero di condizionamento della matrice X è piccolo allora sono tutte proprietà equivalenti, quindi possono essere distinte solo nel caso in cui il cond(X) sia grande.
Il nostro scopo è di mostrare con degli esempi numerici che tra queste proprietà solo la `NC` dà una condizione sufficiente per la stabilità all'indietro del metodo di Newton con scaling. L'algoritmo di cui faremo uso è l'algoritmo di inversione di Matlab. Esso ha la proprietà `Alt`, che su matrici ben condizionate significa che gode anche della proprietà `Conj`.In realtà anche su matrici malcondizionate accade frequentemente che l'algoritmo è stabile sia a sinistra che a destra, quindi la proprietà `Conj` è verificata lo stesso. Esistono eccezioni per cui si verifica la proprietà `Alt` ma non la proprietà `Conj`, però è molto probabile che se dispone della proprietà `Conj` allora avrà anche la proprietà `NC` (si veda [2, pag. 491-492]).

I primi esperimenti li mostriamo su matrici di dimensione 10 generate casualmente. Mostriamo nello stesso piano i grafici di $e_\ell^{(L)}, e_\ell^{(R)}$ e δ_ℓ al variare di ℓ. Le ascisse indicheranno i valori naturali di ℓ, mentre le ordinate i corrispondenti valori che assumono sui vari ℓ gli errori relativi e i δ_ℓ.

Esperimento 1:

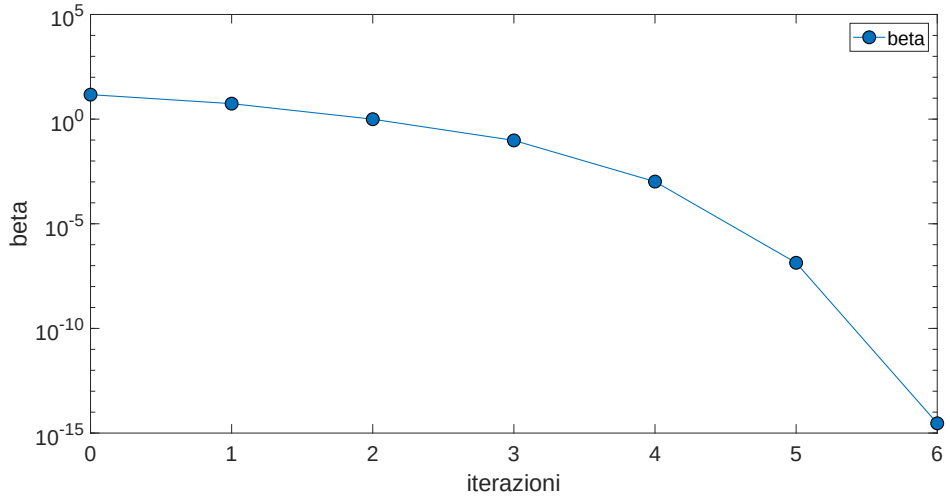
Generiamo una matrice A con il comando
```
A=rand(10);
```
e applichiamo ad A il metodo di Newton con scaling scegliendo l'optimal scaling e otteniamo la coppia di matrici $\{\tilde{U}, \tilde{H}\}$. Risulta che $||\tilde{U}^T\tilde{U} - I||_F \approx 7.3 \times 10^{-16}$, inoltre gli autovalori di $\tilde{H}$ sono tutti positivi.

Mostriamo adesso il grafico che rappresenta gli errori relativi $e_\ell^{(L)}$, $e_\ell^{(R)}$ e i δ_ℓ:



Il metodo è terminato dopo 6 iterazioni, e gli errori relativi hanno un ordine di grandezza di $10^{-17}, 10^{-16}, 10^{-15}$, mentre i δ_ℓ hanno un ordine di grandezza di 10^{-16}.
Sono valori che possiamo ritenere accettabili, quindi in questo caso il metodo ha restituito una coppia $\{\tilde{U}, \tilde{H}\}$ soddisfacente.

Ora mostriamo anche il grafico che rappresenta β_ℓ:

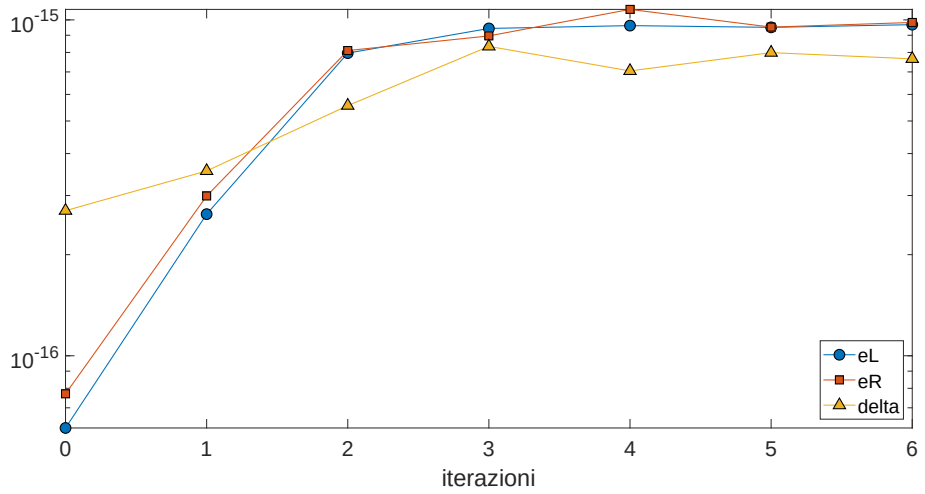


Possiamo notare che in poche iterazioni i valori di β_ℓ scendono rapidamente raggiungendo valori molto piccoli.

Esperimento 2:

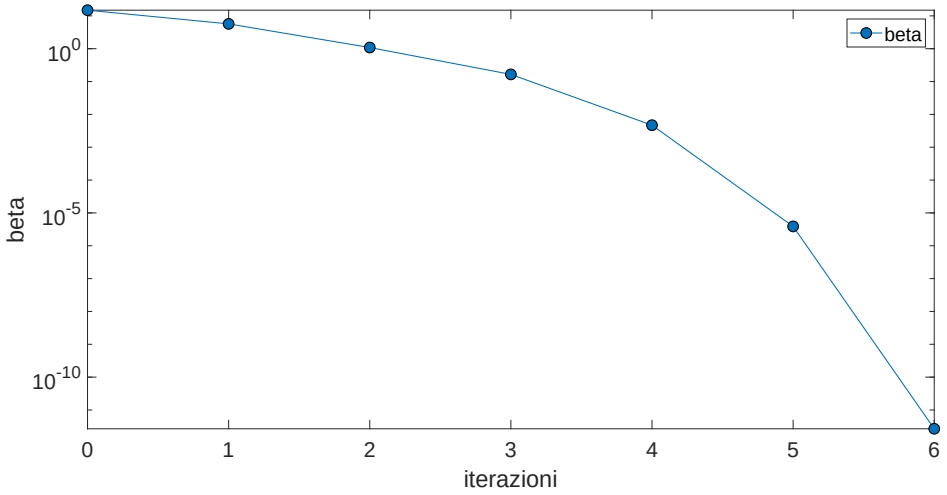
Generiamo una matrice A sempre con il comando
`A=rand(10);`
e applichiamo ad A il metodo di Newton con scaling scegliendo il Frobenius norm scaling e otteniamo la coppia di matrici $\{\tilde{U}, \tilde{H}\}$.Risulta che $||\tilde{U}^T\tilde{U} - I||_F \approx 9.07 \times 10^{-16}$, inoltre gli autovalori di $\tilde{H}$ sono tutti positivi.

Mostriamo adesso il grafico che rappresenta gli errori relativi $e_\ell^{(L)}$, $e_\ell^{(R)}$ e i δ_ℓ:



Il metodo è terminato dopo 7 iterazioni, e gli errori relativi hanno un ordine di grandezza di $10^{-17}, 10^{-16}, 10^{-15}$, mentre i δ_ℓ hanno un ordine di grandezza di 10^{-16}.
Sono valori che rispettano la nostra soglia di accettazione.

Mostriamo ora il grafico che rappresenta β_ℓ:

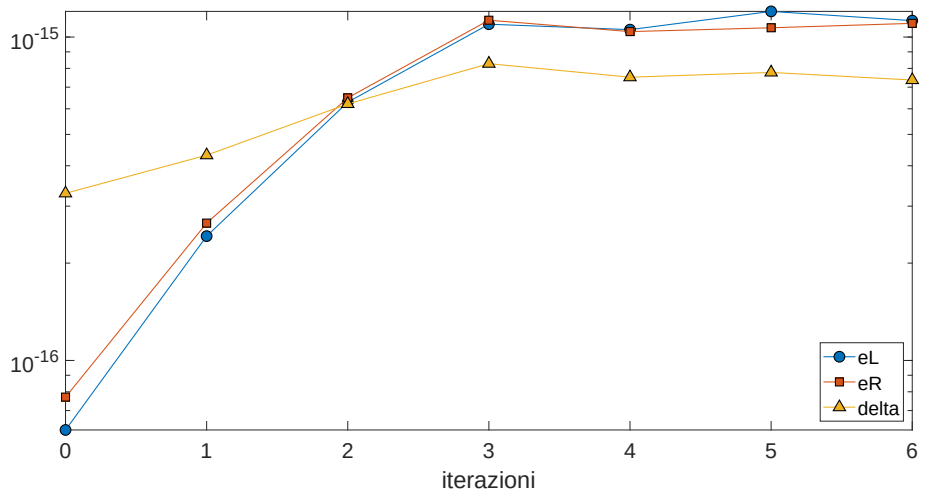


Anche in questo caso in poche iterazioni i valori di β_ℓ scendono rapidamente arrivando ad essere molto piccoli.

Esperimento 3:

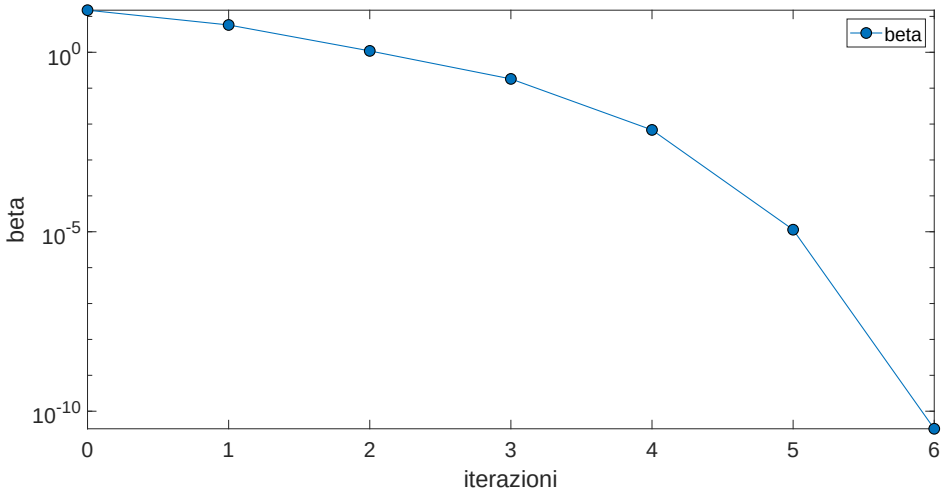
Generiamo una matrice A sempre con il comando
`A=rand(10);`
e applichiamo ad A il metodo di Newton con scaling scegliendo l'$1-$inf norm scaling e otteniamo la coppia di matrici $\{\tilde{U}, \tilde{H}\}$.Risulta che $||\tilde{U}^T\tilde{U} - I||_F \approx 8.85 \times 10^{-16}$, inoltre gli autovalori di $\tilde{H}$ sono tutti positivi.

Mostriamo adesso il grafico che rappresenta gli errori relativi $e_\ell^{(L)}$, $e_\ell^{(R)}$ e gli errori δ_ℓ:



Il metodo è terminato dopo 7 iterazioni, e in questo caso gli errori relativi hanno un ordine di grandezza di $10^{-17}, 10^{-16}, 10^{-15}$, mentre i δ_ℓ hanno un ordine di grandezza di 10^{-16}.
Sono valori che rispettano la nostra soglia di accettazione.

Mostriamo ora il grafico che rappresenta β_ℓ:



Anche in questo caso in poche iterazioni i valori di β_ℓ scendono rapidamente arrivando ad essere molto piccoli.

In tutti gli esperimenti visti finora gli errori relativi sono sufficientemente bassi, quindi possiamo dire che l'algoritmo di inversione su queste matrici dispone della proprietà `Conj`. Come abbiamo detto in precedenza è probabile che disponga anche della proprietà `NC`, infatti anche i δ_ℓ sono sufficientemente bassi, e ciò non ci permette di dire quindi se `Conj` e `NC` in questo caso sono distinte.

Esperimento 4:

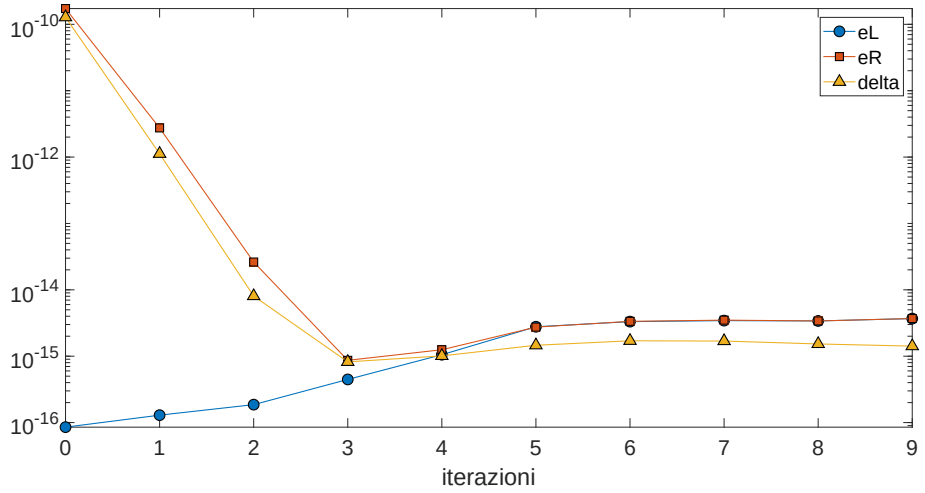
Mostriamo adesso un esperimento in cui, data una matrice A, l'algoritmo di inversione `Inv` usato ha la sola proprietà `Alt` su almeno una delle matrici $\tilde{X}_\ell$ (in particolare la avrà sulla matrice A stessa). Generiamo una matrice A nel seguente modo:

```
v=(1/25):(1/25):1;
```

```
V=fliplr(vander(v));
V=V';
[Q3,L3]=qr(V);
L3=L3';
Q=rand(25);
[Q,R]=qr(Q);
A=Q*L3';
```

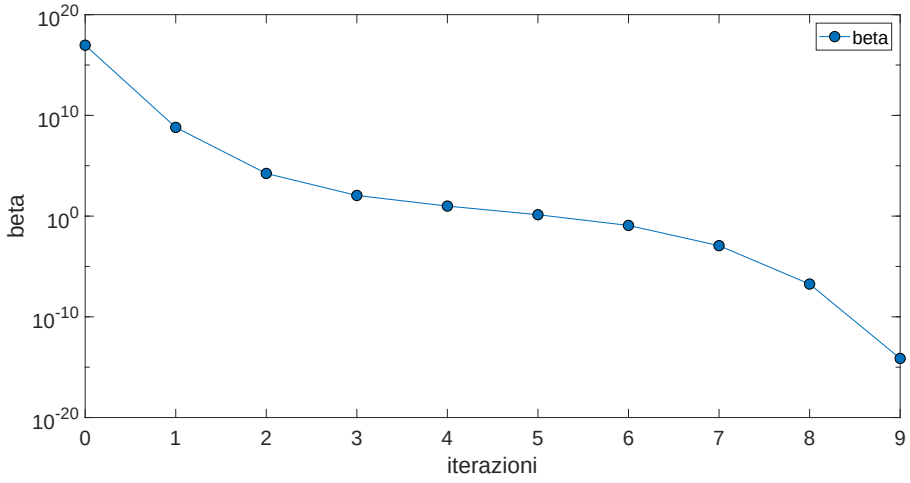
e applichiamo ad A il metodo di Newton con scaling scegliendo l'$1-\text{inf}$ norm scaling e otteniamo la coppia di matrici $\{\tilde{U}, \tilde{H}\}$.Risulta che $||\tilde{U}^T\tilde{U} - I||_F \approx 2.68 \times 10^{-15}$, ma gli autovalori di $\tilde{H}$ tendono ad avvicinarsi a zero, questo perché la matrice A data in input è molto vicino all'essere singolare.

Il metodo è terminato dopo 10 iterazioni e mostriamo adesso il grafico rappresentante gli errori relativi $e_\ell^{(L)}$,$e_\ell^{(R)}$ e i δ_ℓ:



Come si può vedere dal grafico l'algoritmo di inversione è stabile a sinistra ma non a destra. Per $\ell = 0, 1, 2$ non è stabile a destra, e anche le decomposizioni polari ottenute non sono di buona qualità. Per $\ell \geq 3$ l'algoritmo è stabile sia a sinistra che a destra, e anche le decomposizioni polari ottenute sono di buona qualità. Possiamo osservare che il grafico rappresentante δ_ℓ ha più o meno lo stesso andamento di quello rappresentante $e_\ell^{(R)}$, con all'incirca gli stessi ordini di grandezza.

Mostriamo adesso anche il grafico rappresentante β_ℓ:



Anche qui vediamo come in pochi passi β_ℓ scende da valori molto alti a valori molto piccoli.

Questo esempio ci mostra quindi che avere la proprietà `Alt` non è una condizione sufficiente per la stabilità all'indietro del metodo di Newton con scaling.

Otteniamo gli stessi risultati utilizzando il Frobenius norm scaling, mentre scegliendo l'optimal scaling Matlab ci restituisce `NaN` (Not a number), in quanto come detto in precedenza, la matrice A è molto vicina all'essere singolare, di conseguenza gli optimal scalings μ_k che fanno uso dei valori singolari sono dei pessimi scalings.
Questo esperimento ci mostra quindi un esempio in cui l'optimal scaling, nonostante i vantaggi indicati nel Capitolo 2, non è sempre lo scaling migliore da scegliere.

Personali esperimenti numerici mostrano che l'algoritmo `Inv` di Matlab mostra una maggiore stabilità a sinistra rispetto a quella destra e, in casi particolari, una forte stabilità a sinistra ma una mancanza di stabilità a destra. I grafici mostrati finora mostrano che δ_ℓ è molto legato a $e_\ell^{(R)}$, in particolar modo l'ordine di grandezza di δ_ℓ dipende da quello di $e_\ell^{(R)}$, anzi è quasi uguale.

Altri esempi:

Ora mostriamo però un esempio in cui si ha la proprietà `Conj` (ma non la `NC`).
Per le motivazioni date in precedenza, non possiamo utilizzare l'algoritmo `Inv` di Matlab per avere la sola proprietà `Conj`. Faremo uso allora del seguente algoritmo, che utilizza la decomposizione SVD:

- Sia $X = P\Sigma Q^T$, una decomposizione SVD di X, dove $\Sigma = \text{diag}(\sigma_i)$, P, Q ortogonali.
- L'inversa calcolata G di X sarà

$$G = Q(\Sigma^{-1} + \nu_d \frac{\sigma_{\max}}{\sigma_{\min}}\Psi)P^T, \quad \Psi = [\psi_{ij}], \quad \psi_{ij} = \frac{d_{ij}}{\max\{\sigma_i, \sigma_j\}}, \tag{3.13}$$

  dove d_{ij} sono numeri random tale che il loro modulo sia minore o uguale di 1.

Osserviamo che $|\sigma_i\psi_{ij}| \leq 1$ e analogamente $|\sigma_i\psi_{ij}| \leq 1$, e quindi vale che

$$||GX-I||_F = ||Q(I+\nu_d\frac{\sigma_{\max}}{\sigma_{\max}}\Psi\Sigma)Q^T||_F \leq \nu_d\frac{\sigma_{\max}}{\sigma_{\max}}||\Psi\Sigma||_F||Q||_2||Q^T||_2 = \nu_d\frac{\sigma_{\max}}{\sigma_{\max}}||\Psi\Sigma||_F \leq n\nu_d\frac{\sigma_{\max}}{\sigma_{\max}}.$$

Allo stesso modo si verifica che $||XG - I||_F \leq n\nu_d\frac{\sigma_{\max}}{\sigma_{\max}}$.

Negli esperimenti gli errori relativi saranno sempre dell'ordine di 10^{-15} (raramente dell'ordine di 10^{-14}), ed inoltre si verifica che se $\kappa_2(X)$ è abbastanza grande e ci sono valori singolari che coincidono con $\alpha = \sqrt{\sigma_{\max}\sigma_{\min}}$ tale algoritmo non rispetta la proprietà `NC`.
Mostriamo adesso tre esperimenti, nei quali è necessario che almeno 2 valori singolari si avvicinino ad α (si veda [2]):

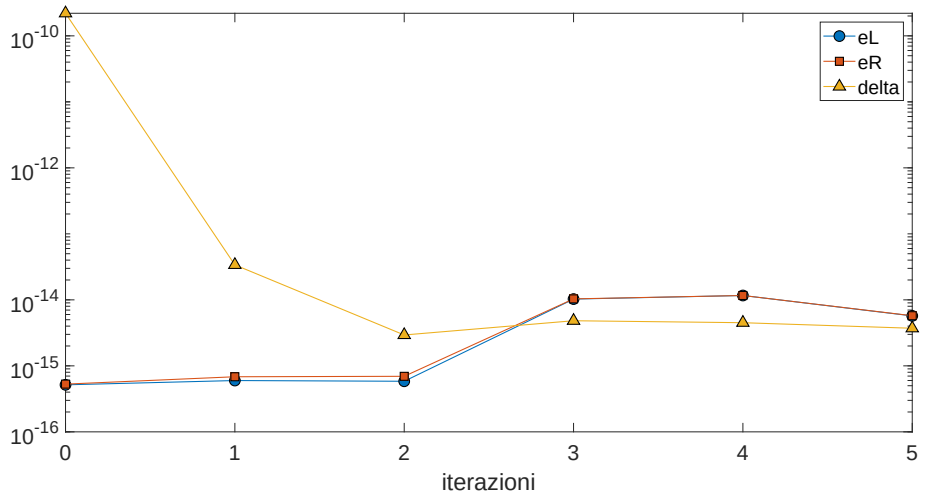
Esperimento 5:

Generiamo una matrice A nel segente modo:

```
d=[10^7; sqrt(2* 10^7) ; 1 ;1 ; sqrt(5 * 10^(-8)) ; 10^(-7)];
Q=rand(6); [Q,R]=qr(Q);
P=rand(6); [P,R]=qr(P);
A=Q*diag(d)*P';
```

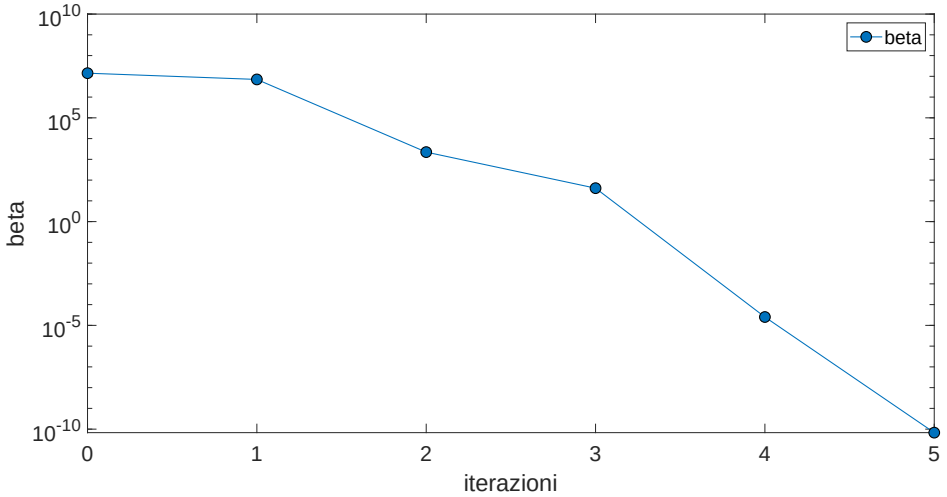
Applichiamo ad A il metodo di Newton con scaling scegliendo l'optimal scaling e, facendo uso su ogni matrice dell'algoritmo di inversione sopra esposto, otteniamo la coppia di matrici $\{\tilde{U}, \tilde{H}\}$. Risulta che $||\tilde{U}^T\tilde{U} - I||_F \approx 1.31 \times 10^{-15}$, inoltre gli autovalori di $\tilde{H}$ sono tutti positivi.

Mostriamo adesso il grafico che rappresenta gli errori relativi $e_\ell^{(L)}$, $e_\ell^{(R)}$ e i δ_ℓ:



Il metodo è terminato dopo 7 iterazioni, e in questo caso gli errori relativi hanno un ordine di grandezza di 10^{-15}, mentre i δ_ℓ hanno un ordine di grandezza di 10^{-16}.
Sono valori che rispettano la nostra soglia di accettazione.

Mostriamo ora il grafico che rappresenta β_ℓ:



Anche in questo caso in poche iterazioni i valori di β_ℓ scendono rapidamente arrivando ad essere molto piccoli.

Come abbiamo potuto vedere dai grafici l'algoritmo di inversione ha la proprietà `Conj`, ma non la proprietà `NC`, ed infatti δ_0 ha ordine di grandezza 10^{-10}, un valore troppo alto per ritenere la coppia $\{\tilde{U}, \tilde{H}\}$ una buona decomposizione polare calcolata.

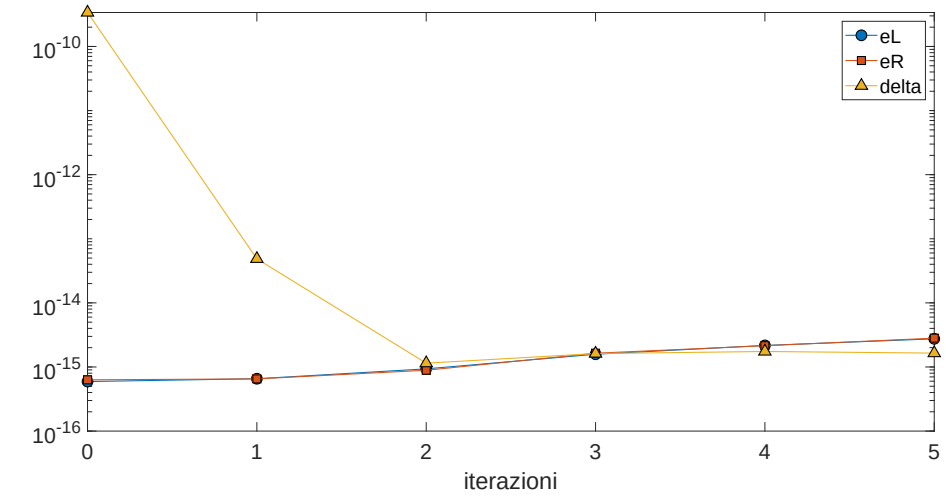
Esperimento 6:

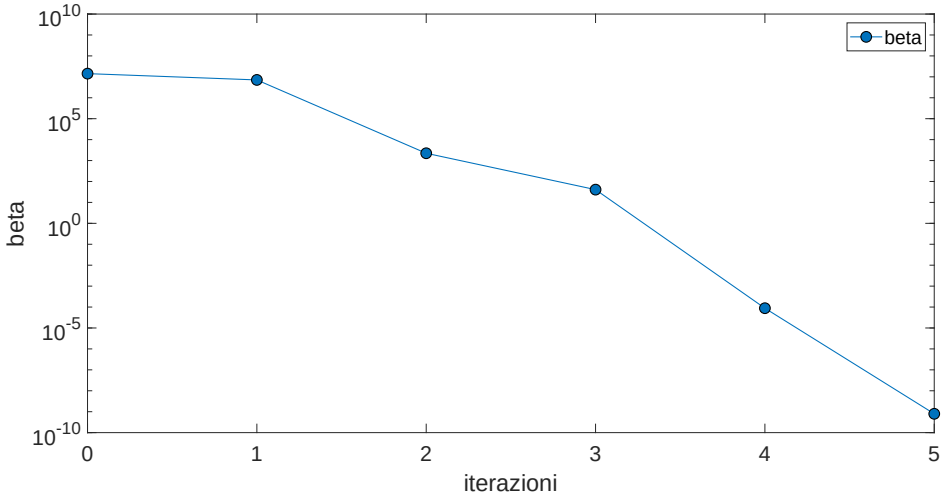
Mostriamo un ulteriore esperimento nel quale la matrice A la generiamo nel seguente modo:

```
10^7 * ones(1,18);
d=[10^(14),ans,1];
d=d';
Q=rand(20); [Q,R]=qr(Q);
P=rand(20); [P,R]=qr(P);
A=Q*diag(d)*P';
```

Otteniamo anche in questo caso una coppia $\{\tilde{U}, \tilde{H}\}$, risulta che $||\tilde{U}^T\tilde{U} - I||_F \approx 1.73 \times 10^{-14}$ e gli autovalori di $\tilde{H}$ sono tutti positivi.

Mostriamo ora i soliti 2 grafici:





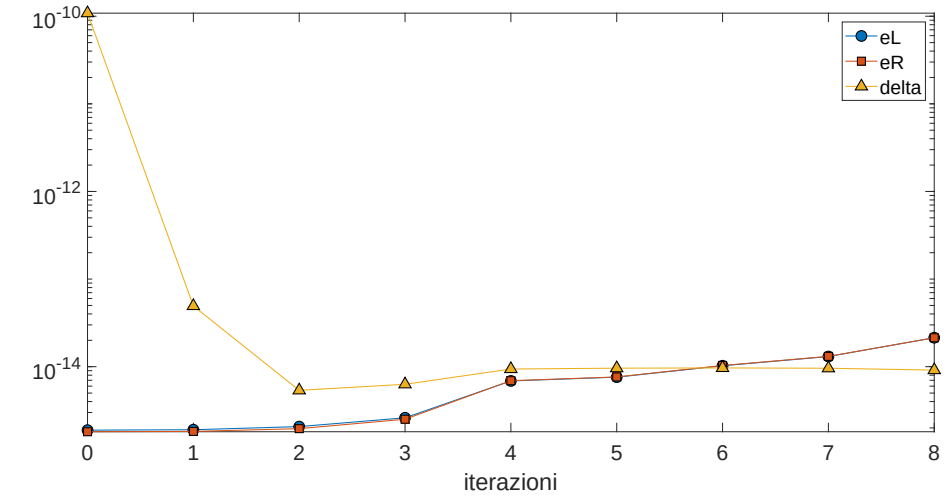
Esperimento 7

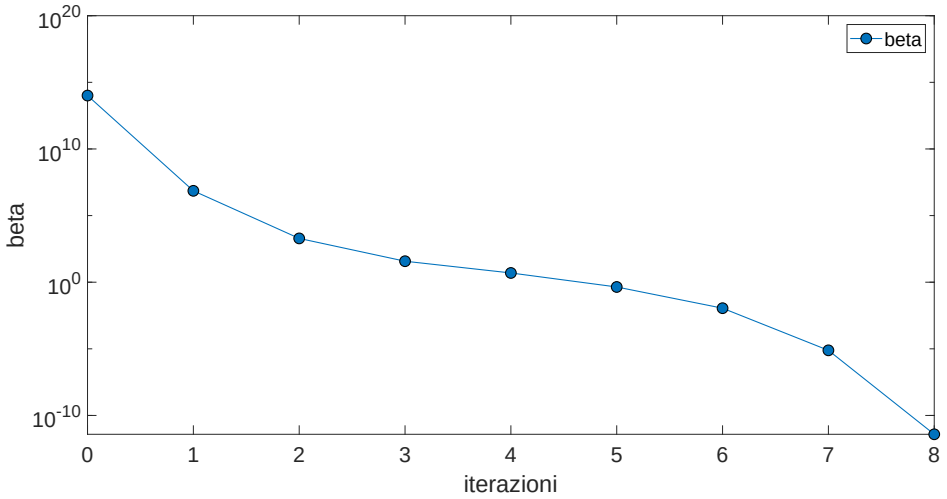
In questo esperimento la matrice A la generiamo nel seguente modo:

```
x=10^(14/19);
d=[];
for kk = 1:20 d=[d,x^(kk-1)];end
d=d';
Q=rand(20); [Q,R]=qr(Q);
P=rand(20); [P,R]=qr(P);
```

`A=Q*diag(d)*P';`

Risulta che $||\tilde{U}^T\tilde{U} - I||_F \approx 1.63 \times 10^{-14}$ e gli autovalori di $\tilde{H}$ sono tutti positivi.
Mostriamo i grafici:





Come abbiamo potuto vedere, sulle matrici mostrate negli esperimenti l'algoritmo di inversione usato è stabile a sinistra e a destra sulle matrici $\tilde{X}_\ell$, ma le coppie $\{\tilde{U}, \tilde{H}\}$ ottenute non sono buone decomposizioni polari per tali matrici. Dunque, abbiamo mostrato come per avere garanzia sulla stabilità all'indietro del metodo di Newton è necessario che l'algoritmo di inversione usato abbia la proprietà `NC`.

Bibliografia

[1] Higham, Nicholas J. *Functions of matrices: theory and computation*, volume 104. Siam, 2008.

[2] Kiełbasiński, Andrzej and Zieliński, Paweł and Ziętak, Krystyna. On iterative algorithms for the polar decomposition of a matrix and the matrix sign function. *Applied Mathematics and Computation*, 270:483–495, 2015.

[3] Greco, Federico and Iannazzo, Bruno and Poloni, Federico. The Padé iterations for the matrix sign function and their reciprocals are optimal. *Linear Algebra and Its Applications*, 436(3):472–477, 2012.

[4] Ziętak, Krystyna. The dual Padé families of iterations for the matrix pth root and the matrix p-sector function. *Journal of Computational and Applied Mathematics*, 272:468–486, 2014.