Metodi numerici per la risoluzione di equazioni di Riccati di grandi dimensioni

Giuseppe Vacca

Alla mia famiglia

Sommario

Le equazioni algebriche di Riccati sono tra le equazioni matriciali più note ed analizzate nella matematica applicata. Tale interesse è dovuto tanto alla vasta gamma di campi in cui le equazioni di Riccati intervengono, tra i quali la teoria del controllo ottimo e stocastico, i giochi differenziali, i modelli di code, quanto alle innumerevoli relazioni delle suddette equazioni con concetti fondamentali dell'algebra lineare in generale, e dell'algebra lineare numerica in particolare: la nozione di sottospazi invarianti e sottospazi di deflazione di matrix pencil, split di autovalori. Le più importanti tipologie di equazioni di Riccati ($\mathcal{ARE}$) sono

- le **equazioni algebriche di Riccati non simmetriche** ($\mathcal{NARE}$) della forma

$$C + XA + DX - XBX = 0, \tag{1}$$

- le **equazioni algebriche di Riccati a tempi continui** ($\mathcal{CARE}$) del tipo

$$C + XD^* + DX - XBX = 0, \tag{2}$$

- le **equazioni algebriche di Riccati a tempi discreti** ($\mathcal{DARE}$) con formulazione

$$A^*XA + Q - (C + B^*XA)^*(R + B^*XB)^{-1}(C + B^*XA) - X = 0. \tag{3}$$

Si tratta, come evidente, di equazioni non lineari, in cui l'incognita X compare come fattore moltiplicativo sia destro che sinistro tanto nel termine di 'primo grado' quanto nel termine di 'secondo grado'. Tale peculiarità differezia le $\mathcal{ARE}$ dalle **equazioni matriciali unilaterali quadratiche** ($\mathcal{UQME}$).

Tra le soluzioni delle $\mathcal{ARE}$, particolarmente importanti sono le cosiddette **soluzioni estremali**, ovvero soluzioni che massimizzano e minimizzano ogni altra soluzione relativamente ad una specificata relazione d'ordine matriciale. È possibile dimostrare, sotto opportune ipotesi sui coefficienti delle $\mathcal{ARE}$, l'esistenza di tali soluzioni. A tal proposito, particolarmente significativo, è il caso in cui la **matrice dei coefficienti** $\mathcal{M} := \left(\begin{smallmatrix} A & -B \\ C & D \end{smallmatrix}\right)$ associata ad una $\mathcal{NARE}$ è una M-matrice, molto ricorrente anche nelle applicazioni.

Fondamentale, per una comprensione teorica dei metodi risolutivi, è l'esistenza di una corrispondenza tra soluzioni delle $\mathcal{ARE}$ e sottospazi invarianti o sottospazi di deflazione di matrix pencil: tale corrispondenza permette, quindi, di individuare una soluzione a partire dal relativo sottospazio, trasformando, dunque, un problema quadratico in un problema lineare. Per le soluzioni estremali, inoltre, la suddetta corrispondenza risuta particolarmente importante in quanto i relativi sottospazi presentano proprietà di c-stabilità o d-stabilità.

I metodi risolutivi per le $\mathcal{ARE}$ si dividono sostanzialmente in due macro gruppi: i **metodi classici** e i doubling algorithm. Tale distinzione si rende necessaria in quanto i primi metodi hanno una rilevanza più storica e teorica piuttosto che applicativa in quanto presentano un elevato costo computazionale e sono inficiati da problemi di stabilità numerica. Tra i metodi classici sono annoverati

- il **metodo di Schur** alla base del quale vi è quanto detto sulla corrispondenza tra sottospazi invarianti o di deflazione e soluzioni delle $\mathcal{ARE}$. Il metodo utilizza la fattorizzazione di Schur delle matrici 'in gioco' per determinare efficientemente tali sottospazi;

- il **metodo delle iterazioni funzionali** il quale definisce delle successioni definite per ricorrenza a partire dalle (1), (2), (3). Ponendo delle opportune ipotesi sui coefficienti, è possibile provare la convergenza di tali successioni alle soluzioni estremali. In generale l'ordine di convergenza è lineare, mentre il costo computazionale è di $O(n^3)$ operazioni per iterazione;

- il **metodo di Newton** applicato all'operatore di Riccati e definito a partire dalla derivata di Fréchet dell'operatore stesso. Anche in tal caso, ponendo ipotesi sui coefficienti, si dimostra la convergenza del motodo alle soluzioni estremali. Il metodo di Newton ha un costo computazionale di $O(n^3)$ operazioni elementari per passo e presenta una convergenza quadratica nel caso 'non critico' e lineare nel caso 'critico'.

I **doubling algorithm**, così denominati perché presentano convergenza quadratica, sono i metodi più competitivi presenti in letteratura, tra di essi i più importanti sono

- il **metodo doubling algorithm strutturato** ($\mathcal{SDA}$) il quale genera una successione di matrix pencil con la proprietà che ad ogni iterazione dell'algoritmo i sottospazi di deflazione rimangono inalterati, mentre i relativi autovalori vengono elevati al quadrato. Tale proprietà risulta molto utile se il sottospazio di deflazione è d-stabile, in tal caso, infatti, è evidente che se l'algoritmo può essere iterato senza interruzioni (ovvero senza breakdown), gli autovalori ad ogni passo hanno norma sempre più piccola, fino a tendere a zero. Questa proprietà rende particolarmente semplificata la ricerca del sottospazio di deflazione d-stabile e fornisce, quindi, un valido metodo per calcolare le soluzioni estremali;

- il **metodo di riduzione ciclica** ($\mathcal{CR}$) che si basa sulla medesima filosofia dell'elevamento a quadrato del metodo $\mathcal{SDA}$, generando una successione di polinomi matriciali quadratici i cui autovalori vengono elevati al quadrato ad ogni iterazione. Si rende dapprima necessario trasformare la $\mathcal{ARE}$ in una $\mathcal{UQME}$ ad essa associata, poi applicare ad essa il metodo $\mathcal{CR}$ per individuarne la soluzione, e da questa risalire, quindi, alla soluzione estremale della $\mathcal{ARE}$ di partenza.

Per poter applicare i doubling algorithm è necessario trasformare proprietà di c-stabilità o c-splitting, in proprietà di d-stabilità o d-splitting, a tal fine, si introducono due tipologie di trasformazioni, le **trasformazioni affini** e le **trasformazioni di Cayley**, che danno origine a due possibili varianti del metodo $\mathcal{SDA}$ e del metodo $\mathcal{CR}$.

I doubling algorithm, come detto, risultano i metodi più efficaci per la risoluzione di equazioni di Riccati in quanto presentano una convergenza quadratica ed hanno un costo computazionale per passo pari a $O(n^3)$ operazioni elementari. Tuttavia, per valori molto grandi di n, tale costo computazionale diviene 'insostenibile' ed anche i dobling algorithm risultano inapplicabili in quanto impiegherebbero tempi non ragionevoli.

I matematici cinesi Chang-Yi Weng, Tiexiang Li, Eric King-wah Chu e Wen-Wei Lin hanno osservato che se una $\mathcal{NARE}$ ha sì grandi dimensioni ma presenta particolari proprietà strutturali, è possibile apportare delle 'correzioni' al metodo $\mathcal{SDA}$ in modo da rendere l'algoritmo efficace anche in tale situazione. Propongono, dunque, un'estensione del metodo $\mathcal{SDA}$, il **large-scale** $\mathcal{SDA}$ ($\mathcal{SDA}_{ls}$) il cui costo computazionale è di $O(n)$ operazioni elementari per iterazione. A base del metodo $\mathcal{SDA}_{ls}$ vi è un astuto utilizzo della **Formula di Sherman-Morrison-Woodbury** ed un particolare 'meccanismo' di **troncamento e compressione** per limitare la crescita del rango di alcune matrici definite nel corso dell'algoritmo.

Nel presente lavoro di tesi è analizzata e commentata la valenza del metodo $\mathcal{SDA}_{ls}$ mettendone in luce sia gli aspetti positivi, tra i quali l'abbattimento del costo computazionale, sia i 'punti deboli' o comunque poco chiari. È, inoltre, studiata la portata di modifiche analoghe al metodo $\mathcal{CR}$, osservando che anche in tal caso è possibile ottenere una notevole miglioramento delle prestazioni. Il metodo proposto nel presente elaborato, il metodo **large-scale** $\mathcal{CR}$ ($\mathcal{CR}_{ls}$) risponde positivamente se testato a problemi con dimensione elevata e particolari proprietà di struttura in quanto presenta un costo computazionale di $O(n)$ operazioni aritmetiche per iterazione e risulta numericamente stabile.

Indice

Elenco dei codici

Capitolo 1

Definizioni e risultati preliminari

Indice

Le equazioni algebriche di Riccati sono tra le equazioni matriciali più studiate e meglio analizzate in letteratura. Sono innumerevoli i campi della matematica applicata in cui tali equazioni intervengono: in molti casi, infatti, determinare le soluzioni di problemi della teoria del controllo ottimo, descrivere modelli di code o modelli stocastici equivale a risolvere particolari equazioni di Riccati. L'importanza di tali equazioni però, non è esclusivamente applicativa, sono numerosi, infatti, i legami tra equazioni di Riccati e principi basilari dell'algebra lineare, quali il concetto di autospazio, il concetto di sottospazio di deflazione di matrix pencil, il concetto di c-stabilità o d-stabilità.

Nel paragrafo 1.1 viene in primo luogo definito il concetto di *equazione matriciale* e vengono poi introdotte le equazioni oggetto di studio: le *equazioni algebriche di Riccati* ($\mathcal{ARE}$). Sono dunque presentate nell'ordine le *equazioni algebriche di Riccati non simmetriche* ($\mathcal{NARE}$), le *equazioni algebriche di Riccati a tempi continui* ($\mathcal{CARE}$), le *equazioni algebriche di Riccati a tempi discreti* ($\mathcal{DARE}$), le *equazioni di Sylvester*, le *equazioni di Lyapunov*, le *equazioni di Stein* e le *equazioni matriciali unilaterali quadratiche* ($\mathcal{UQME}$).

Nel paragrafo 1.2 viene proposto una particolare applicazione delle equazioni di Riccati. Per rendere al meglio l'importanza di tali equazioni e per dare una parziale idea di cosa significa nella pratica risolvere un'istanza di un problema reale, si è scelto di partire da una situazione concreta, modellizzarla, ed arrivare alla formulazione matematica del problema in esame.

Nel paragrafo 1.3 sono introdotti alcuni concetti fondamentali di Algebra Lineare, basilari per l'analisi teorica delle equazioni di Riccati. Sono presentate le nozioni di *sot-*

tospazi invarianti, di *sottospazi di deflazione* associati a *matrix pencil* e relativi *autovalori*. Sono dunque illustrate due particolari tipologie di trasformazioni, le *trasformazioni affini* e le *trasformazioni di Cayley*, strumenti fondamentali per una efficace applicazione degli algoritmi risolutivi proposti.

Nel paragrafo 1.4 sono trattate le proprietà teoriche delle equazioni di Riccati: è presentata la corrispondenza tra particolari sottospazi vettoriali e soluzioni delle $\mathcal{ARE}$, sono mostrati risultati sull'esistenza, sotto opportune ipotesi sui coefficienti della $\mathcal{ARE}$, di *soluzioni estremali*, sono analizzate le proprietà spettrali della *matrice Hamiltoniana* e sono, infine, introdotte le definizioni di *splitting* di una matrice e *drift* di una $\mathcal{NARE}$.

1.1 Equazioni algebriche di Riccati

Si denotino con $\mathbb{R}$ e con $\mathbb{C}$ rispettivamente il campo reale e il campo complesso e con $\mathbb{R}^n$ e con $\mathbb{C}^n$ lo spazio vettoriale reale e complesso n-dimensionale. Allo stesso modo si indicano con $\mathbb{R}^{m\times n}$ e con $\mathbb{C}^{m\times n}$ l'insieme delle matrici reali e complesse di dimensione $m \times n$.

Una **equazione matriciale** è, parlando grossolanamente, un'espressione del tipo

$$\mathcal{F}(X) = 0,$$

dove $\mathcal{F}$ è un operatore da un sottospazio di $\mathbb{C}^{m\times n}$ in uno spazio vettoriale $\mathcal{V}$.

Nelle equazioni matriciali oggetto di questa tesi intervengono solo operazioni elementari, quali somme, moltiplicazioni (destre e sinistre), inversioni e trasposizioni.

Si osservi che risolvere un'equazione matriciale equivale sostanzialmente a risolvere un sistema (non necessariamente lineare) nelle incognite x_{ij} per $i = 1, \ldots, m$, $j = 1, \ldots, n$. Si preferisce tuttavia considerare come indeterminata l'intera matrice X, in quanto tale formulazione permette di sfruttare la particolare struttura delle matrici che intervengono nell'equazione stessa.

Le equazioni matriciali di Riccati derivano da una nota equazione differenziale non lineare presentata dal matematico trevigiano Jacopo Franceso Riccati (1676-1754) [45]:

$$x'(t) = a(t)x(t)^2 + b(t)x(t) + c(t),$$

dove a, b, c sono funzioni continue definite in un intervallo I e la funzione a non è identicamente nulla. L'equazione differenziale di Riccati è tuttora applicata in molti campi della matematica moderna. Dal punto di vista storico, invece, la rilevanza di tale equazione è da attribuire alla 'rivoluzionaria' impostazione dei metodi di integrazione proposti: grazie ad un cambiamento di variabile, Riccati riuscì a ridurre un'equazione del secondo ordine ad un'equazione differenziale del primo ordine.

1.1.1 Equazioni algebriche di Riccati non simmetriche

Una **equazione algebrica di Riccati non simmetrica** ($\mathcal{NARE}$) è un'equazione matriciale quadratica nella indeterminata X della forma:

$$C + XA + DX - XBX = 0, \tag{1.1}$$

dove $X \in \mathbb{C}^{m\times n}$, $A \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{n\times m}$, $C \in \mathbb{C}^{m\times n}$, $D \in \mathbb{C}^{m\times m}$.

Si osservi che la matrice incognita X viene moltiplicata a sinistra e a destra sia nei termini lineari, che nel termine quadratico (si noti a tal proposito la differenza con le $\mathcal{UQME}$).

Alla $\mathcal{NARE}$ (1.1) è possibile associare le seguenti matrici 2×2 a blocchi:

- la **matrice dei coefficienti**

$$\mathcal{M} = \begin{pmatrix} A & -B \\ C & D \end{pmatrix},$$

- la **matrice Hamiltoniana**

$$\mathcal{H} = \begin{pmatrix} A & -B \\ -C & -D \end{pmatrix}.$$

Si verifica immediatamente che vale la relazione

$$\mathcal{H} = \begin{pmatrix} I_n & 0 \\ 0 & -I_m \end{pmatrix} \mathcal{M},$$

dove I_k indica la matrice identità k-dimensionale.

Alla $\mathcal{NARE}$ (1.1) è solitamente accostata la seguente equazione detta **equazione di Riccati duale** di (1.1):

$$YCY + AY + YD - B = 0,$$

dove l'incognita $Y \in \mathbb{C}^{n\times m}$. Si osservi che tale equazione è ottenuta scambiando in (1.1) rispettivamente i ruoli dei coefficienti A e D e di B e C.

Tra le soluzioni di una $\mathcal{NARE}$, l'interesse degli algoritmi e dei modelli applicativi si focalizza principalmente sulla matrice soluzione X, detta **soluzione minimale non negativa** individuata dalla seguenti proprietà

- $x_{ij} \geq 0$ per $i = 1, \ldots, m$, e $j = 1, \ldots, n$, ovvero X è **non negativa**,
- se W soluzione non negativa della $\mathcal{NARE}$, allora $w_{ij} \geq x_{ij}$ per $i = 1, \ldots, m$, e $j = 1, \ldots, n$.

Nel seguito tale matrice è indicata con la notazione $X_{\min}$. Risultati sull'esistenza e sulle proprietà della matrice $X_{\min}$ sono oggetto del paragrafo 1.4.

1.1.2 Equazioni algebriche di Riccati a tempi continui

La definizione di $\mathcal{NARE}$ è volutamente una formulazione quanto più generale, molto più dettagliata e circostanziata è invece la definizione di **equazione algebrica di Riccati a tempi continui** ($\mathcal{CARE}$) data da:

$$C + XD^* + DX - XBX = 0, \tag{1.2}$$

dove $m = n$, e $C = C^*$, $B = B^*$. Si osservi dunque che il termine costante e il termine quadratico di una $\mathcal{CARE}$ sono hermitiani, mentre i termini lineari sono uno il trasposto coniugato dell'altro. È evidente dalle proprietà di simmetria, che se X risolve l'equazione (1.2), allora anche X^* è soluzione della $\mathcal{CARE}$ precedente.

Una $\mathcal{CARE}$ può essere viste come una particolare **equazione algebrica di Riccati a tempi continui generalizzata** ($\mathcal{GCARE}$), equazione della forma

$$C + E^*XA + A^*XE - E^*XBXE = 0, \tag{1.3}$$

con $E \in \mathbb{C}^{n\times n}$. È evidente che, se la matrice E è invertibile, allora la $\mathcal{GCARE}$ può essere ridotta ad una $\mathcal{CARE}$. Se infatti X risolve la (1.3), allora X è soluzione della $\mathcal{CARE}$:

$$E^{-*}CE^{-1} + XAE^{-1} - E^{-*}A^*X - XBX = 0.$$

dove $E^{-*} := (E^*)^{-1}$.

Individuare la soluzione di $\mathcal{CARE}$ è estramente importante in diversi rami della matematica applicata, quali la teoria del controllo ottimo e dei controlli stocastici [6, 50]. Tra le soluzioni di maggior interesse nelle implementazioni vanno annoverate le **soluzioni estremali**: le matrici X_+ e X_- si dicono rispettivamente **soluzione massimale** e **soluzione minimale** di una $\mathcal{CARE}$ se sono hermitiane e se per ogni soluzione hermitiana X vale

$$X_- \preceq X \preceq X_+$$

ove la scrittura $M \preceq N$ denota che M, N sono entrambe hermitiane e che $N - M$ è una matrice semidefinita positiva.

1.1.3 Equazioni algebriche di Riccati a tempi discreti

Una **equazione algebrica di Riccati a tempi discreti** ($\mathcal{DARE}$) è una equazione matriciale della forma

$$A^*XA + Q - (C + B^*XA)^*(R + B^*XB)^{-1}(C + B^*XA) - X = 0, \tag{1.4}$$

con $X \in \mathbb{C}^{n\times n}$, $A \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{n\times m}$, $C \in \mathbb{C}^{m\times n}$, $Q \in \mathbb{C}^{n\times n}$ e $R \in \mathbb{C}^{m\times m}$, e si suppone inoltre che le matrici Q e R siano hermitiane. Una soluzione X si dice **ammissibile** se la matrice $(R + B^*XB)$ risulta invertibile.

In analogia con quanto accade per le $\mathcal{CARE}$, è possibile generalizzare la formulazione di $\mathcal{DARE}$ introducendo le **equazioni di Riccati a tempi discreti generalizzata** ($\mathcal{GDARE}$):

$$A^*XA + Q - (C + B^*XA)^*(R + B^*XB)^{-1}(C + B^*XA) - E^*XE = 0, \tag{1.5}$$

dove $E \in \mathbb{C}^{n\times n}$. Qualora la matrice E risulti invertibile, dalla $\mathcal{GDARE}$ (1.5) si ricava immediatamente la seguente $\mathcal{DARE}$:

$$E^{-*}A^*XAE^{-1}+E^{-*}QE^{-1}-E^{-*}(C+B^*XA)^*(R+B^*XB)^{-1}(C+B^*XA)E^{-1}-X=0.$$

Anche per le $\mathcal{DARE}$ le soluzioni di maggior rilevanza nelle varie applicazioni risulteranno hermitiane, mentre dal punto di vista teorico sono particolarmente importanti le soluzioni estremali X_- e X_+.

La risoluzione di $\mathcal{DARE}$ è alla base di tutti problemi di controllo ottimo lineari e quadratici che sfruttano il principio del massimo di Pontryagin [47, 53]. È evidente dalla definizione stessa, che le $\mathcal{CARE}$ modellizzano sistemi a tempi continui, mentre le $\mathcal{DARE}$ risolvono istanze di sistemi dinamici discreti.

1.1.4 Equazioni algebriche di Sylvester, Lyapunov e Stein

Nell'ambito di una trattazione più esaustiva delle equazioni di Riccati, è opportuno introdurre tre tipologie di equazioni, che, sebbene non presentino un termine quadratico, possono essere considerate come particolari esempi di $\mathcal{NARE}$.

Una **equazione di Sylvester** è una equazione matriciale lineare del tipo:

$$AX + XB = Q, \tag{1.6}$$

dove $X \in \mathbb{C}^{m\times n}$, $A \in \mathbb{C}^{m\times m}$, $B \in \mathbb{C}^{n\times n}$, $Q \in \mathbb{C}^{m\times n}$.

Se nell'equazione di Sylvester si ha $m = n$, $B = A^*$ e si suppone Q hermitiana, si ottiene un'equazione matriciale della forma

$$AX + XA^* = Q, \tag{1.7}$$

detta **equazione di Lyapunov**.

Si osservi che, data la hermitianità dei coefficienti coinvolti nella (1.7), se X è soluzione dell'equazione di Lyapunov, lo è necessariamente anche X^*. Sotto opportune ipotesi sugli autovalori della matrice A, è possibile dimostrare che esiste una e una sola soluzione dell'equazione di Lyapunov, dall'osservazione precedente, quindi, si ha che tale soluzione risulterà hermitiana.

La panoramica sulle equazioni di Riccati si conclude con l'**equazione di Stein**:

$$X - AXB = Q \tag{1.8}$$

con $X \in \mathbb{C}^{m\times n}$, $A \in \mathbb{C}^{m\times m}$, $B \in \mathbb{C}^{n\times n}$, $Q \in \mathbb{C}^{m\times n}$. In particolare, se si pone nella (1.8) $B := A^*$ e si suppone Q hermitiana si ottiene una equazione della forma

$$X - AXA^* = Q,$$

detta **equazione di Stein simmetrica**.

1.1.5 Equazioni matriciali unilaterali quadratiche

Sebbene non propriamente appartenenti alla classe delle equazioni algebriche di Riccati, è opportuno illustrare nella presente introduzione una ulteriore tipologia di equazioni, le **equazioni matriciali unilaterali quadratiche** ($\mathcal{UQME}$):

$$A_0 + A_1 X + A_2 X^2 = 0$$

dove X, A_0, A_1, $A_2 \in \mathbb{C}^{n\times n}$, e A_2 è una matrice non nulla.

Si osservi che, a differenza di quanto accade nelle $\mathcal{NARE}$, nelle $\mathcal{UQME}$ il termine incognito X compare sia nel termine lineare che nel termine quadratico esclusivamente come fattore moltiplicativo destro.

Le $\mathcal{UQME}$ hanno innumerevoli applicazione nella risoluzione di modelli di code, nell'analisi di particolari sistemi dinamici e nello studio di particolari catene di Markov [11]. Tali equazioni, inoltre, come esposto nel capitolo 3, rappresentano un valido strumento per risolvere le $\mathcal{NARE}$.

1.2 Modellizzazione e applicazione di $\mathcal{ARE}$

Si consideri un'asta omogenea che si estende dal punto 0 al punto x e si denoti con z un generico punto sull'asta. Si supponga che delle particelle attraversino l'asta muovendosi da sinistra verso destra e da destra verso sinistra senza collidere tra loro ma interagendo con il materiale che compone l'asta. Si supponga, per il momento, che le particelle siano dello stesso tipo ed abbiano medesima velocità, l'obiettivo è dunque quello di ottenere informazioni sulla densità delle particelle in funzione della posizione attraversata z.

Si assuma che la probabilità che una particella nella posizione z (che si muove in qualsiasi delle direzioni possibili) interagisca con il materiale dell'asta mentre compie un percorso lungo $\Delta > 0$ è dato dall'espressione

$$\sigma(z)\Delta + o(\Delta), \tag{1.9}$$

dove la quantità $\sigma(\cdot)$ indica la **sezione macroscopica d'attraversamento** e con la notazione $o(\cdot)$ si denotano i termini di ordine più elevato. Come conseguenza di tale interazione vengono a crearsi esattamente nel punto z dell'asta un valore atteso di $f(z)$ nuove particelle che si muovono nella stessa direzione della particella originaria ed un valore atteso di $b(z)$ nuove particelli che si muovono nella direzione opposta. Si osservi che la denominazione 'valore atteso' si rende necessaria in quanto il processo che definisce il problema ivi trattato ha natura evidentemente stocastica.

Per la (1.9), la probabilità che una particella che percorre il tratto da z a $z + \Delta$ dell'asta non interagisca con la stessa è data da

$$1 - \sigma(z)\Delta + o(\Delta). \tag{1.10}$$

Le particelle che attraversano verso sinstra il punto $z = 0$ e verso destra il punto $z = x$ vengono perse dal sistema, mentre le particelle inserite dalle estremità sinistra e destra dell'asta e le nuove particelle generate delle varie interazioni particela-asta formano la popolazione del sistema.

Si supponga che il valore atteso della popolazione di particelle sull'asta sia stazionario, ovvero che sia indipendente dal particolare istante in cui viene osservato il sistema. Siano quindi

- $u(z)$ il numero atteso di particelle che si muovono verso destra che passano dal punto z ogni secondo,
- $v(z)$ il numero atteso di particelle che si muovono verso sinistra che passano dal punto z ogni secondo.

A partire dalle informazioni in possesso, è possibile determinare delle equazioni soddisfatte dalle funzioni $u(z)$ e $v(z)$.

Si consideri il flusso di particelle che si muovono da destra a sinistra $u(z)$ passanti dal punto z, allora il numero atteso di particelle $u(z+\Delta)$ si ottiene considerando i seguenti contributi:

- il contributo dato dalle particelle che non interagiscono con l'asta:

$$(1-\sigma(z)\Delta+o(\Delta))u(z)=(1-\sigma(z)\Delta)u(z)+o(\Delta)$$

  dove l'ultima uguaglianza è giustificata se $u(z)$ è un numero finito.

- se qualcuna delle particelle $u(z)$ interagisce con l'asta prima di raggiungere la posizione $z+\Delta$ produce un numero atteso di $f(z)$ particelle che procedono nella medesima direzione, dunque il contributo di tali reazioni è

$$(\sigma(z)\Delta+o(\Delta))f(z)u(z)=\sigma(z)f(z)u(z)\Delta+o(\Delta).$$

- se qualcuna delle particelle $v(z+\Delta)$ che si muovono a sinistra interagisce con l'asta prima di raggiungere il punto z, produce un numero atteso di particelle $b(z+\Delta)$ che si muovono nella direzione opposta, dunque il contributo di tali reazioni è

$$(\sigma(z+\Delta)\Delta+o(\Delta))b(z+\Delta)v(z+\Delta)=\sigma(z)b(z)v(z)\Delta+o(\Delta).$$

Sommando i vari contributi si ottiene pertanto

$$u(z+\Delta)=(1-\sigma(z)\Delta)u(z)+\sigma(z)f(z)u(z)\Delta+\sigma(z)b(z)v(z)\Delta+o(\Delta),$$

da cui

$$\frac{du(z)}{dz}=\lim_{\Delta\to 0}\frac{u(z+\Delta)-u(z)}{\Delta}=\sigma(z)\left((f(z)-1)u(z)+b(z)v(z)\right).$$

Si consideri ora il flusso di particelle che si muovono verso destra $v(z)$, allora il numero atteso di particelle $v(z-\Delta)$ si ottiene considerando i seguenti contributi:

- il contributo dato dalle particelle che non interagiscono con l'asta prima di raggiungere la posizione $z-\Delta$:

$$(1-\sigma(z)\Delta+o(\Delta))v(z)=(1-\sigma(z)\Delta)v(z)+o(\Delta)$$

  dove l'ultima uguaglianza è giustificata se $v(z)$ è un numero finito.

- se qualcuna delle particelle $v(z)$ interagisce con l'asta prima di raggiungere la posizione $z-\Delta$ produce un numero atteso di $f(z)$ particelle che procedono nella medesima direzione, dunque il contributo di tali reazioni è

$$(\sigma(z)\Delta+o(\Delta))f(z)v(z)=\sigma(z)f(z)u(z)\Delta+o(\Delta).$$

- se qualcuna delle particelle $u(z-\delta)$ che si muovono verso destra interagisce con l'asta prima di raggiungere il punto z, produce un numero atteso di particelle $b(z-\Delta)$ che si muovono nella direzione opposta, dunque il contributo di tali reazioni è

$$(\sigma(z-\Delta)\Delta+o(\Delta))b(z-\Delta)u(z-\Delta)=\sigma(z)b(z)u(z)\Delta+o(\Delta).$$

Sommando i vari contributi si ottiene quindi

$$v(z-\Delta)=(1-\sigma(z)\Delta)v(z)+\sigma(z)f(z)v(z)\Delta+\sigma(z)b(z)u(z)\Delta+o(\Delta),$$

da cui

$$-\frac{dv(z)}{dz}=\lim_{\Delta\to 0}\frac{v(z-\Delta)-v(z)}{\Delta}=\sigma(z)\left((f(z)-1)v(z)+b(z)u(z)\right).$$

Come condizioni al bordo compatibili con il sistema, si suppone che, ogni secondo, una particella venga inserita nell'asta nell'estremità $z = x$ mentre nessuna particella venga inserita all'estremità $z = 0$. Integrando tali condizioni alle equazioni differenziali sopra indicate, si ottiene il problema con condizioni al bordo

$$\begin{cases} \dfrac{du(z)}{dz} = \sigma(z)\left((f(z)-1)u(z) + b(z)v(z)\right), \\ -\dfrac{dv(z)}{dz} = \sigma(z)\left((f(z)-1)v(z) + b(z)u(z)\right), \\ u(0) = 0, \quad v(x) = 1. \end{cases} \tag{1.11}$$

È possibile considerare, in maniera del tutto analoga, anche modelli multistato in cui le particelle sono caratterizzate da n stati (per esempio velocità, energia, tipologia). In tal caso, per ogni stato j, si introduce la sezione macroscopica d'attraversamento $\sigma_j(\cdot) > 0$, dunque, la probabilità che una particella nello stato j e nella posizione z interagisca con l'asta mentre compie un percorso lungo Δ è data da

$$\sigma_j(z)\Delta + o(\Delta).$$

Sempre in analogia con quanto fatto precedentemente, siano

- $u_j(z)$ il numero atteso di particelle nello stato j che si muovono verso destra che passano dal punto z ogni secondo,
- $v_j(z)$ il numero atteso di particelle nello stato j che si muovono verso sinistra che passano dal punto z ogni secondo,
- $f_{ij}(z)$ il numero atteso di particelle nello stato i ottenute dopo una reazione di una particella nello stato j nella posizione z e che si muovono nella medesima direzione della particella originaria,
- $b_{ij}(z)$ il numero atteso di particelle nello stato i ottenute dopo una reazione di una particella nello stato j nella posizione z e che si muovono nella direzione opposta alla particella originaria.

Siano quindi

$$F(z) := (f_{ij}(z))_{i,j=1,\dots,n}, \quad B(z) := (b_{ij}(z))_{i,j=1,\dots,n}, \quad D(z) := diag(\sigma_i(z))_{i=1,\dots,n},$$

allora, per argomentazioni del tutto analoghe a qulle sopra esposte, si ottiene il problema con condizioni al contorno

$$\begin{cases} \dfrac{d\mathbf{u}(z)}{dz} = (F(z) - I_n)D(z)\mathbf{u}(z) + B(z)D(z)\mathbf{v}(z), \\ -\dfrac{d\mathbf{v}(z)}{dz} = (F(z) - I_n)D(z)\mathbf{v}(z) + B(z)D(z)\mathbf{u}(z), \\ \mathbf{u}(0) = 0, \quad \mathbf{v}(x) = \mathbf{e}, \end{cases} \tag{1.12}$$

dove

$$\mathbf{u}(z)^T := (u_i(z))_{i=1,\dots,n}, \quad \mathbf{v}(z)^T := (v_i(z))_{i=1,\dots,n}, \quad \mathbf{e}^T := \begin{pmatrix} 1, & \dots, & 1 \end{pmatrix}.$$

Si torni a considerare il caso in cui le particelle siano identificate da un unico stato. Attraverso le informazioni in possesso, è possibile descrivere il comportamento globale del modello, senza investigare la distribuzione interna delle particelle nel sistema. siano quindi

- $t(x)$ la **funzione di trasmissione**, ovvero il numero di particelle che emergono nel punto $z = 0$ per secondo,
- $r(x)$ la **funzione di riflessione**, ovvero il numero di particelle che emergono nel punto $z = x$ per secondo,

si osservi che tali funzioni non dipendono dal punto z in cui le particelle emergono, ma esclusivamente dalla lunghezza dell'asta x. Al solito, l'obiettivo è quello di individuare equazioni differenziali verificate dalle funzioni di trasmissione e riflessione.

Si consideri un'asta con le medesime caratteristiche fisiche della precedente avente però lunghezza $x + \Delta$, allora, utilizzando tecniche di **invariant imbedding** è possibile individuare delle relazioni tra i valori $r(x)$ e $r(x + \Delta)$, e tra i valori $t(x)$ e $t(x + \Delta)$.

Per il calcolo del numero di particelle $r(x+\Delta)$ che compaiono nell'estremità si devono considerare i seguenti contributi

- la particella che viene immessa ogni secondo all'estremità destra dell'asta $x + \Delta$ si muove verso sinistra e possono verificarsi le seguenti situazioni
  - se tale particella interagisce con l'asta produce un contributo di $b(x + \Delta)$ particelle che emergono in $z = x + \Delta$, e $f(x + \Delta)$ particelle che emergono in $z = x$ e dunque vanno ad aggiungersi alle $r(x)$ ivi presenti,
  - se la particella non interagisce con l'asta prima di raggiungere la posizione $z = x$, essa si aggiunge alle $r(x)$ particelle in $z = x$.

  Considerando le probabilità di ciascuno dei due eventi, si conclude che la particella immessa all'estremità destra dell'asta, comporta $\sigma(x)b(x)\Delta+o(\Delta)$ nuove particelle in $z = x + \Delta$, e $sr(x)$ nuove particelle in $z = x$, dove
  $$s := 1 - (f(x) - 1)\sigma(x)\Delta + o(\Delta).$$
- Le nuove particelle $sr(x)$ presenti in $z = x$ possono avere al solito due tipologie di comportamento
  - possono non interagire con l'asta prima di raggiungere la posizione $z = x + \Delta$ dando quindi un contributo di $sr(x)(1 - \sigma(x)\Delta) + o(\Delta)$ particelle,
  - possono interagire con l'asta dando vita a $sr(x)f(x)\sigma(x)\Delta + o(\Delta)$ nuove particelle in $z = x + \Delta$ e $sr^2(x)b(z)\Delta + o(\Delta)$ in $z = x$. Tali particelle sono reimmesse nel sistema e possono ritornare all'estremità destra dell'asta dando un contributo di $sr^2(x)b(x)\sigma(x)\Delta(1 - \sigma(x)\Delta) + o(\Delta)$.

Sommando i diversi contributi, si ottiene l'equazione

$$\begin{aligned} r(x + \Delta) &= \sigma(x)b(x)\Delta + sr(x)(1 - \sigma(x)\Delta) + sr(x)f(x)\sigma(x)\Delta + sr^2(x)b(x)\sigma(x)\Delta + o(\Delta) \\ &= \sigma(x)b(x)\Delta + sr(x)(1 + (f(x) - 1)\sigma(x)\Delta) + sr^2(x)b(x)\sigma(x)\Delta + o(\Delta) \\ &= r(x) + \sigma(x)b(x)\Delta + 2r(x)(1 - f(x))\sigma(x)\Delta + r^2(x)b(x)\sigma(x)\Delta + o(\Delta), \end{aligned}$$

da cui si ricava che la funzione di riflessione verifica l'equazione differenziale

$$\frac{dr(x)}{dx} = \lim_{\Delta\to\infty} \frac{r(x + \Delta) - r(x)}{\Delta} = \sigma(x)b(x)+2r(x)(1-f(x))\sigma(x)+r^2(x)b(x)\sigma(x). \quad (1.13)$$

Ripetendo argomentazioni analoghe si ottiene che la funzione di trasmissione realizza l'equazione

$$\frac{dt(x)}{dx} = (f(x) - 1 + b(x)r(x))\sigma(x)t(x). \quad (1.14)$$

Per quanto riguarda le condizioni iniziali si osservi che se l'asta ha lunghezza nulla allora non vi sono particelle 'riflesse', mentre tutte le particelle sono trasmesse, dunque si hanno le condizioni

$$r(0) = 0, \qquad t(1).$$

Nel caso a più stati si introducono la **matrice di riflessione** $R(x) := (r_{ij})_{i,j=1,\dots,n}$ e la **matrice di trasmissione** $T(x) := (t_{ij})_{i,j=1,\dots,n}$ definite come segue

- $r_{ij}(x)$ è il numero di particelle che emergono nel punto $z = x$ nello stato i quando l'unico 'input' è una particella per secondo nello stato j,
- $t_{ij}(x)$ è il numero di particelle che emergono nel punto $z = 0$ nello stato i quando l'unico 'input' è una particella per secondo nello stato j.

Generalizzando i calcoli svolti precedentemente ed adoperando le notazioni sopra menzionate, si ottiene il seguente problema di Cauchy

$$\begin{cases} \dfrac{dR(x)}{dx} = B(x)D(x) - R(x)(I_n - F(x))D(x) - (I_n - F(x))D(x)R(x) + R(x)B(x)D(x)R(x), \\ \dfrac{dT(x)}{dx} = T(x)((F(x) - I_n)D(x) + B(x)D(x)R(x)), \\ R(0) = 0, \quad T(0) = I_n. \end{cases} \tag{1.15}$$

Particolare interesse nelle applicazioni, hanno le soluzioni $R(x)$ a stati stazionari, ovvero le soluzioni $R(x)$ che verificano l'equazione

$$\frac{dR(x)}{dx} = 0 \quad \text{per ogni } x.$$

Si osservi ora che, per determinare tali soluzioni, occorre risolvere, per ogni valore di $\bar{x}$, l'equazione di Riccati nella variabile $R(\bar{x})$

$$B(\bar{x})D(\bar{x}) - R(\bar{x})(I_n - F(\bar{x}))D(\bar{x}) - (I_n - F(\bar{x}))D(\bar{x})R(\bar{x}) + R(\bar{x})B(\bar{x})D(\bar{x})R(\bar{x}). \tag{1.16}$$

Qualora vi fossero differenti sezioni di attraversamento macroscopiche $\sigma_j^l(\cdot)$ e $\sigma_j^r(\cdot)$ a seconda che le particelle si muovano rispettivamente verso sinistra o verso destra, allora la $\mathcal{NARE}$ (1.16) si riscrive come

$$B^l(\bar{x}) - R(\bar{x})F^l(\bar{x}) - F^r(\bar{x})R(\bar{x}) + R(\bar{x})B^r(\bar{x})R(\bar{x}), \tag{1.17}$$

dove

$$D^p(\cdot) := diag(\sigma_j^p(\cdot))_{j=1,\dots,n}, \quad B^p(\cdot) := B(\cdot)D^p(\cdot), \quad F^p(\cdot) := (I_n - F(\cdot))D^p(\cdot),$$

con $p = l, r$.

Si osservi, infine, che, se per ogni indice $i = 1, \dots, n$ gli elementi delle matrici $F(\bar{x})$ e $B(\bar{x})$ verificano la disuguaglianza

$$\sum_{j=1}^{n} f_{ij}(\bar{x}) + b_{ij}(\bar{x}) < 1,$$

allora la matrice dei coefficienti $\mathcal{M}(\bar{x})$ associata alla $\mathcal{NARE}$ (1.17) è una M-matrice non singolare. Vele infatti

$$\begin{aligned} \mathcal{M}(\bar{x}) &= \begin{pmatrix} (I_n - F(\bar{x}))D^l(\bar{x}) & B(\bar{x})D^r(\bar{x}) \\ -B(\bar{x})D^l(\bar{x}) & (I_n - F(\bar{x}))D^r(\bar{x}) \end{pmatrix} = \\ &= \left(\begin{pmatrix} I_n & 0 \\ 0 & I_n \end{pmatrix} - \begin{pmatrix} F(\bar{x}) & B(\bar{x}) \\ B(\bar{x}) & F(\bar{x}) \end{pmatrix} \right) \begin{pmatrix} D^l(\bar{x}) & 0 \\ 0 & D^r(\bar{x}) \end{pmatrix}. \end{aligned}$$

Dunque la matrice $\mathcal{M}(\bar{x})$ è in primo luogo una Z-matrice, inoltre, ponendo

$$v(\bar{x}) := \begin{pmatrix} (D^l)^{-1}(\bar{x})e \\ (D^r)^{-1}(\bar{x})e \end{pmatrix} > 0,$$

vale la relazione $\mathcal{M}(\bar{x})v(\bar{x}) > 0$ dunque, per le proprietà delle M–matrici, $\mathcal{M}(\bar{x})$ è una M-matrice non singolare.

1.3 Concetti basilari di Algebra lineare numerica

L'Algebra Lineare rappresenta indubbiamente uno degli ambiti fondamentali della matematica applicata: gran parte dei problemi concreti possono essere, infatti, riformulati matematicamente mediante concetti quali spazio vettoriale, sistema lineare, base di autovettori, ortonormalità. Si è dunque sviluppata un'importante branca dell'Analisi numerica che ha introdotto ed analizzato metodi per risolvere problemi classici dell'Algebra Lineare con criteri e caratteristiche proprie dell'Analisi numerica. I primi ad intraprendere questo percorso sono stati Alston Scott Householder [35] e James Hardy Wilkinson [56] rispettivamente nel 1964 e nel 1965, seguiti da Gene H. Golub e Charles F. Van Loan [21] nel 1983.

Nel seguito sono illustrate delle nozioni di base di Algebra lineare numerica indispensabili per la comprensione dei risultati teorici più rilevanti riguardanti le $\mathcal{ARE}$ presentati nel paragrafo successivo.

1.3.1 Sottospazi invarianti e di deflazione, stabilità e splitting

Siano $\mathcal{V}$ uno spazio vettoriale e $\mathcal{G}$ un automorfismo di $\mathcal{V}$, si consideri un sottospazio vettoriale $\mathcal{W} \subseteq \mathcal{V}$ e si definisca il seguente sottospazio:

$$\mathcal{GW} := \{\, y \in \mathcal{V} \mid y = \mathcal{G}x, x \in \mathcal{W} \,\}.$$

Un sottospazio $\mathcal{W} \subseteq \mathcal{V}$ si dice **sottospazio $\mathcal{G}$-invariante** se risulta $\mathcal{GW} \subseteq \mathcal{W}$.

Nel seguito si suppone che tutti gli spazi vettoriali trattati abbiano dimensione finita, pertanto è possibile considerare $\mathcal{V} \subseteq \mathbb{C}^n$ e identificare l'automorfismo $\mathcal{G}$ con una matrice $G \in \mathbb{C}^{n\times n}$. Se, dunque, si pone $m := \dim \mathcal{W}$, e si definisce la matrice

$$W := \big(w_1, \ldots, w_m\big), \qquad \text{con } \mathcal{W} = \mathtt{Span}(w_1, \ldots, w_m),$$

il sottospazio $\mathcal{W}$ è $\mathcal{G}$-invariante se e solo se esiste una matrice $\Gamma \in \mathbb{C}^{m\times m}$ tale che

$$GW = W\Gamma. \tag{1.18}$$

È importante osservare dalla (1.18) che lo spettro della matrice Γ è un sottoinsieme dello spettro della matrice G. Infatti, se esistono $v \in \mathbb{C}^m$ non nullo e $\lambda \in \mathbb{C}$ tali che $\Gamma v = \lambda v$, allora

$$G(Wv) = GWv = W\Gamma v = W(\Gamma v) = \lambda(Wv).$$

Si ha, pertanto, che la forma canonica di Jordan della matrice Γ è composta da sottoblocchi della forma canonica di Jordan della matrice G, in particolare, quindi, la molteplicità algebrica di un autovalore per Γ è minore o uguale alla molteplicità del medesimo autovalore per la matrice G.

Siano ora $L,\ K \in \mathbb{C}^{n\times n}$, risolvere il **problema generalizzato agli autovalori** relativo alla coppia (L, K) significa determinare i valori $z \in \mathbb{C}$ e i vettori $v \in \mathbb{C}^n$ non nulli che verificano la seguente relazione:

$$Lv = zKv,$$

dove z e v si dicono rispettivamente **autovalore** e **autovettore generalizzati**.

È chiaro che il problema generalizzato agli autovalori è intrinsecamente legato alla seguente funzione della variabile z

$$\mathcal{P}(z) := L - zK \tag{1.19}$$

detta **matrix pencil** associata alla coppia (L, K), ed all'equazione

$$\mathcal{C}(z) := \det(L - zK) = 0$$

detta **equazione caratteristica**. Si osservi in particolare che z è un autovalore generalizzato se e solo se $C(z) = 0$. Nel seguito, dunque, sono assimilati i concetti di autovalore generalizzato relativo ad una coppia di matrici e di **autovalore di una matrix pencil**.

Se esiste $\tilde{z}$ tale che $\mathcal{C}(\tilde{z}) \neq 0$, si parla di matrix pencil **regolare**. Si ha, per esempio, una matrix pencil non regolare se $\texttt{rank}(L) + \texttt{rank}(K) < n$, vale infatti

$$\texttt{rank}(L - zK) \leq \texttt{rank}(L) + \texttt{rank}(K) < n \qquad \forall z \in \mathbb{C}$$

allora necessariamente $\mathcal{C}(z) = 0$ per ogni arbitrario valore di $z \in \mathbb{C}$.

Si osservi che se la matrice L è singolare allora $z = 0$ è autovalore generalizzato, mentre se K è non invertibile allora $\mathcal{C}(z)$ ha grado $n-r$ per qualche intero r strettamente positivo. In tale situazione si pone convenzionalmente $z = \infty$ autovalore generalizzato di molteplicità r.

Un sottospazio vettoriale m-dimensionale $\mathcal{S} \subseteq \mathbb{C}^n$ si dice **sottospazio di deflazione** per la matrix pencil $\mathcal{P}(z)$ se esiste un sottospazio $\mathcal{T}$ di dimensione m tale che

$$L\mathcal{S} \subseteq \mathcal{T}, \qquad K\mathcal{S} \subseteq \mathcal{T}.$$

Utilizzando la notazione già adottata precendetemente, ponendo

$$S := \big(s_1, \ldots, s_m\big), \qquad \text{con } \texttt{Span}(s_1, \ldots, s_m),$$

$$T := \big(t_1, \ldots, t_m\big), \qquad \text{con } \texttt{Span}(t_1, \ldots, t_m),$$

si ottiene che $\mathcal{S}$ è un sottospazio di deflazione per $\mathcal{P}(z)$ se e solo se esistono Λ, $\Xi \in \mathbb{C}^{m\times m}$ tali che

$$LS = T\Lambda, \qquad KS = T\Xi. \tag{1.20}$$

È possibile riscrivere la (1.20) come

$$\mathcal{P}(z)S = (L - zK)S = T(\Lambda - z\Xi) = T\Pi(z),$$

dove $\Pi(z)$ indica la matrix pencil associata alla coppia (Λ, Ξ). Si ricava, dunque, che gli autovalori della matrix pencil $\Pi(z)$ sono un sottoinsieme degli autovalori di $\mathcal{P}(z)$.

È opportuno evidenziare che, se $\Pi(z)$ non ha autovalori nulli, e dunque, se Λ è non singolare, dalla (1.20) si ottiene $LS\Lambda^{-1} = T$, e quindi

$$LS\Lambda^{-1}\Xi = T\Xi = KS,$$

ponendo $\Delta := \Lambda^{-1}\Xi$, in forma più compatta, si ha $LS\Delta = KS$.

Allo stesso modo, se si suppone che $\Pi(z)$ abbia autovalori finiti e quindi che la matrice Ξ sia invertibile, ponendo $\Sigma := \Xi^{-1}\Lambda$, si ricava $KS\Sigma = LS$.

Il concetto di sottospazio di deflazione è evidentemente una generalizzazione della nozione di sottospazio invariante: se un sottospazio $\mathcal{S}$ è, infatti, A-invariante allora $\mathcal{S}$ risulta un sottospazio di deflazione relativo alla matrix pencil $A - zI_n$.

Siano L_1, $K_1 \in \mathbb{C}^{n\times n}$ e sia $\mathcal{P}_1(z)$ la matrix pencil associata alla coppia (L_1, K_1). La matrix pencil $\mathcal{P}(z)$ definita in (1.19) si dice **simile a destra** a $\mathcal{P}_1(z)$ se esiste una matrice $R \in \mathbb{C}^{n\times n}$ tale che

$$\mathcal{P}_1(z) = R\mathcal{P}(z).$$

Analogamente $\mathcal{P}(z)$ è detta **simile a sinistra** a $\mathcal{P}_1(z)$ se esiste una matrice $R \in \mathbb{C}^{n\times n}$ tale che

$$\mathcal{P}_1(z) = \mathcal{P}(z)R.$$

I concetti di sottospazio invariante e sottospazio di deflazione, come mostrato, permettono sostanzialmente di selezionare taluni autovalori e i corrispettivi autovettori del problema di partenza. Tale situazione risulta estremamente comoda in quanto permette di studiare autospazi relativi a autovalori che giacciono su particolari regioni del piano complesso $\mathbb{C}$. Molti degli algoritmi dell'Algebra Lineare Numerica, infatti, sfruttano proprietà quali, per esempio, quella di convergenza di serie, che sono spesso riscontrabili solo in specifici domini del piano complesso. In tale ottica è opportuno introdurre le seguenti regioni particolarmente significative in gran parte degli algoritmi:

- $\mathcal{D} := \{\, z \in \mathbb{C} : |\, z| < 1 \,\}$ il disco unitario aperto,

- $\bar{\mathcal{D}} := \{ z \in \mathbb{C} : |z| \leq 1 \}$ il disco unitario chiuso,
- $S^1 := \{ z \in \mathbb{C} : |z| = 1 \}$ la circonferenza unitaria,
- $\mathbb{C}_l := \{ z \in \mathbb{C} : Re(z) < 0 \}$ il semipiano aperto sinistro,
- $\mathbb{C}_r := \{ z \in \mathbb{C} : Re(z) > 0 \}$ il semipiano aperto destro,
- $\mathbb{C}_0 := \{ z \in \mathbb{C} : Re(z) = 0 \}$ l'asse immaginario,
- $\mathbb{C}_{l,0} := \mathbb{C}_0 \cup \mathbb{C}_l$ il semipiano chiuso sinistro,
- $\mathbb{C}_{r,0} := \mathbb{C}_0 \cup \mathbb{C}_r$ il semipiano chiuso destro.

Alla luce delle precedenti notazioni, una matrice A si dice

- **d-stabile** se tutti gli autovalori di A sono in $\mathcal{D}$,
- **d-debolmente stabile** se tutti gli autovalori di A sono in $\bar{\mathcal{D}}$,
- **d-antistabile** se tutti gli autovalori di A sono in $\bar{\mathcal{D}}^c$,
- **d-debolmente antistabile** se tutti gli autovalori di A sono in $\mathcal{D}^c$,
- **c-stabile** se il suo spettro è contenuto in $\mathbb{C}_l$,
- **c-debolmente stabile** se il suo spettro è contenuto in $\mathbb{C}_{l,0}$,
- **c-antistabile** se il suo spettro è contenuto in C_r.
- **c-debolmente antistabile** se il suo spettro è contenuto in $\mathbb{C}_{r,0}$.

Un autovalore di A si dice **stabile** (**debolmente stabile**) se si trova nella regione di stabilità (nella chiusura della regione di stabilità). La nomenclatura adottata nelle precedenti definizioni è giustificata dalle applicazioni in cui nei problemi a tempi discreti la regione di interesse è $\mathcal{D}$, mentre nei sistemi continui il dominio di interesse è $\mathbb{C}_l$. Inoltre un sottospazio invariante o un sottospazio di deflazione $\mathcal{W}$ è detto **stabile** (**debolmente stabile**) se gli autovalori ad esso relativo sono tutti stabili (debolmente stabili).

Sia $A \in \mathbb{C}^{n \times n}$ e sia $\sigma(A) = \{ \lambda_1, \ldots, \lambda_n \}$ il suo spettro ordinato per norma

$$|\lambda_1| \leq \cdots \leq |\lambda_n|.$$

Siano inoltre n_1, $n_2 \geq 1$ con $n_1 + n_2 = n$, allora si dice che la matrice A ha un $(\mathbf{n_1}, \mathbf{n_2})$ **d-splitting** se

$$\lambda_i \in \bar{\mathcal{D}} \quad \text{per } i = 1, \ldots, n_1 \qquad \lambda_{n_1+i} \in \mathcal{D}^c \quad \text{per } i = 1, \ldots, n_2.$$

Entrando maggiormente nel dettaglio lo (n_1, n_2) d-splitting si dice

- **forte** se

$$\lambda_i \in \mathcal{D} \quad \text{per } i = 1, \ldots, n_1 \qquad \lambda_{n_1+i} \in \bar{\mathcal{D}}^c \quad \text{per } i = 1, \ldots, n_2,$$

- **debole** se

$$\lambda_i \in \bar{\mathcal{D}} \quad \text{per } i = 1, \ldots, n_1 \qquad \lambda_{n_1+i} \in \mathcal{D}^c \quad \text{per } i = 1, \ldots, n_2,$$

  e le due partizioni dello spettro si intersecano su S^1,

- **proprio** se

$$\lambda_i \in \mathcal{D} \quad \text{per } i = 1, \ldots, n_1 \qquad \lambda_{n_1+i} \in \mathcal{D} \quad \text{per } i = 1, \ldots, n_2,$$

  ma le due partizioni dello spettro non si intersecano su S^1.

Analogamente, apportando le modifiche del caso alla definizione sopra menzionata, si definisce un $(\mathbf{n_1}, \mathbf{n_2})$ **c-splitting** (è sufficiente porre $\mathbb{C}_l$ in luogo di $\mathcal{D}$ e $\mathbb{C}_0$ in luogo di S^1).

1.3.2 Trasformazioni di autovalori

I concetti introdotti nella precedente sezione sono fondamentali nell'Algebra Lineare Numerica in quanto gran parte degli algoritmi si basano su proprietà di c-stabilità e d-stabilità. Non sempre, però, nelle applicazioni queste proprietà sono immediatamente 'spendibili', si rendono, quindi, spesso necessarie delle particolari trasformazioni matriciali, che operino opportunamente sullo spettro. In tale ottica è essenziale il seguente teorema ([10, 32]):

Teorema 1.3.1. *Siano $\Omega \subseteq \mathbb{C}$ un dominio aperto e $\mathcal{T}\colon \Omega \to \mathbb{C}$ una funzione analitica complessa. Sia $L \in \mathbb{C}^{n\times n}$ con spettro $\sigma(L) = \{\lambda_1, \ldots, \lambda_n\} \subset \Omega$, allora gli autovalori di $\mathcal{T}(L)$ sono $\mathcal{T}(\lambda_1), \ldots, \mathcal{T}(\lambda_n)$. Sia inoltre $\mathcal{W}$ un sottospazio m-dimensionale $\mathcal{T}$-invariante, ovvero, con le notazioni solite*

$$LW = W\Lambda,$$

allora vale

$$\mathcal{T}(L)W = W\mathcal{T}(\Lambda).$$

Il risultato appena esposto ha, dunque, due importanti conseguenze: in primis illustra come una funzione analitica agisce sugli autovalori, in secondo luogo mostra che una funzione analitica conserva i sottospazi invarianti. Dopo la trattazione di carattere generale si entra nel merito delle trasformazioni utilizzate dagli algoritmi illustrati nei capitoli successivi.

Una **trasformazione affine** è una funzione analitica definita da

$$\mathcal{A}_\alpha(z) = \alpha z - 1, \tag{1.21}$$

dipendente dal parametro α. Si osservi che, ponendo $\alpha = 1/\gamma$, $\gamma > 0$, la trasformazione affine (1.21) mappa la circonferenza di centro γ e raggio γ in S^1. Le trasformazioni affini, quindi, portano regioni limitate di $\mathbb{C}_r$ in $\mathcal{D}$. Si consideri ora la matrix pencil $\mathcal{P}(z)$ (1.19) e si supponga che la matrice K sia invertibile, allora, se λ è un autovalore di $\mathcal{P}(z)$, si ottiene

$$\begin{aligned}\det(L - \lambda K) &= \det(K^{-1}L - \lambda I_n) = \\ &= \det(\alpha K^{-1}L - I_n - (\alpha\lambda - 1)I_n) = \det(\mathcal{A}_\alpha(K^{-1}L) - \mathcal{A}_\alpha(\lambda)I_n) = 0,\end{aligned}$$

ovvero $\mathcal{A}_\alpha(\lambda)$ è un autovalore della matrix pencil $\mathcal{Q}(z) := \mathcal{A}_\alpha(K^{-1}L) - zI_n$. D'altro canto, la matrix pencil $\mathcal{Q}(z)$ è evidentemente simile a destra a $\alpha L - K - zK$, dunque, a meno di tale similitudine, è possibile denotare

$$\mathcal{A}_\alpha(\mathcal{P}(z)) = \mathcal{A}_\alpha(L - zK) := \alpha L - K - zK.$$

È opportuno osservare che le matrix pencil $\mathcal{P}(z)$ e $\mathcal{A}_\alpha(\mathcal{P}(z))$ abbiano medesimi sottospazi di deflazione, infatti le condizioni

$$LS = T\Lambda, \qquad KS = T\Xi$$

implicano

$$(\alpha L - K)S = T(\alpha\Lambda - \Xi), \qquad KS = T\Xi,$$

da cui la tesi.

Un'altra classe di trasformazioni particolarmente importante è data dalle **trasformazioni di Cayley**:

$$\mathcal{C}_\gamma(z) = \frac{z - \gamma}{z + \gamma}$$

dipendenti dal parametro $\gamma \neq 0$ e definite per $z \neq -\gamma$. Il comportamento delle trasformazioni di Cayley è presentato dal seguente teorema:

Teorema 1.3.2. *Le trasformazioni di Cayley $\mathcal{C}_\gamma$ verificano le seguenti proprietà:*

i) $\mathcal{C}_\gamma(0) = -1$, $\lim_{|z|\to\infty} \mathcal{C}_\gamma(z) = 1$;

ii) *se* $\gamma \in \mathbb{R}$ *allora* $\mathcal{C}_\gamma(\mathbb{R} \cup \{\infty\}) = \mathbb{R} \cup \{\infty\}$;

iii) *se* $\gamma \in \mathbb{R}$ *allora* $\mathcal{C}_\gamma(\mathbb{C}_0) = S^1$

iv) *se* $\gamma > 0$ *allora* $\mathcal{C}_\gamma(\mathbb{C}_r) = \mathcal{D}$;

v) *se* $\gamma < 0$ *allora* $\mathcal{C}_\gamma(\mathbb{C}_l) = \mathcal{D}$.

Sia $\mathcal{P}(z)$ la matrix pencil (1.19), allora, svolgendo calcoli analoghi a quelli fatti per le trasformazioni affini, si ha

$$\mathcal{C}_\gamma(\mathcal{P}(z)) = \mathcal{C}_\gamma(L - zK) := L - \gamma K - z(L + \gamma K).$$

Anche le trasformazioni di Cayley lasciano inalterati i sottospazi di deflazione. Se infatti vale

$$LS = T\Lambda, \qquad KS = T\Xi,$$

si conclude facilmente

$$(L - \gamma K)S = T(\Lambda - \gamma\Xi), \qquad (L + \gamma K)S = T(\Lambda + \gamma\Xi).$$

Entrambe le classi di trasformazioni proposte hanno quindi due proprietà importanti: trasformano 'bene' gli autovalori, lasciando invariati i sottospazi di deflazione.

1.4 Proprietà teoriche delle $\mathcal{ARE}$

Il presente paragrafo si propone di illustrare i risultati teorici di maggior interesse nell'analisi delle equazioni di Riccati, tali risultati si basano principalmente sulle nozioni introdotte nel paragrafo 1.3 e sulle ben note proprietà delle M-matrici. Come mostrato nei capitoli successivi, i suddetti risultati, inoltre, sono alla base degli algoritmi risolutivi più noti. La prima parte dell'analisi mostra il legame presente tra soluzioni delle $\mathcal{ARE}$ e sottospazi invarianti o sottospazi di deflazione associati a matrix pencil, in particolare si studia la relazione tra spettro della matrice $\mathcal{H}$ e autovalori realtivi a tali autospazi. La seconda parte invece si occupa di dare risultati di esistenza delle soluzioni delle $\mathcal{ARE}$ di maggior rilevanza nelle applicazioni: le soluzioni estremali. Nell'ultima sezione sono infine analizzate le proprietà spettrali e lo splitting della matrice dei coefficienti ed è introdotto il concetto di drift di una $\mathcal{NARE}$.

1.4.1 Corrispondenza tra soluzioni di $\mathcal{ARE}$ e spazi invarianti e di deflazione

Per ciascuna delle tipologie di $\mathcal{ARE}$ prensentate nel Paregrago 1.3 è possibile individuare una relazione tra soluzioni e sottospazi invarianti o sottospazi di deflazione di matrix pencil e dare, dunque, una caratterizzazione geometrica di tali soluzioni.

Si consideri la $\mathcal{NARE}$

$$C + XA + DX - XBX = 0, \tag{1.22}$$

e sia $\mathcal{H}$ la relativa matrice Hamiltoniana. Vale il seguente fondamentale risultato:

Teorema 1.4.1. *Una matrice* $X \in \mathbb{C}^{m\times n}$ *è soluzione della* $\mathcal{NARE}$ (1.22) *se e solo se verifica*

$$\mathcal{H}\begin{pmatrix} I_n \\ X \end{pmatrix} = \begin{pmatrix} I_n \\ X \end{pmatrix}(A - BX). \tag{1.23}$$

In particolare $X \in \mathbb{C}^{m\times n}$ *è soluzione della* $\mathcal{NARE}$ (1.22) *se e solo se esiste un sottospazio* n*-dimensionale* $\mathcal{H}$*-invariante* $\mathcal{V}$ *tale che, con le notazioni della sezione 1.3.1, si abbia*

$$V = \begin{pmatrix} I_n \\ X \end{pmatrix}.$$

Dimostrazione. È banale osservare che dalla relazione (1.23) si ottiene

$$\begin{pmatrix} A-BX \\ -C-DX \end{pmatrix} = \mathcal{H}\begin{pmatrix} I_n \\ X \end{pmatrix} = \begin{pmatrix} I_n \\ X \end{pmatrix}(A-BX) = \begin{pmatrix} A-BX \\ XA-XBX \end{pmatrix},$$

uguagliando ora la seconda riga a blocchi di primo e ultimo membro, si ha che X è soluzione della $\mathcal{NARE}$ (1.22) se e solo se verifica la (1.23).

Per i risultati illustrati nella sezione 1.3.1, quindi, X è una soluzione della (1.22), se e solo se il sottospazio n-dimensionale

$$\mathcal{V} := \mathtt{Span}\left(\begin{pmatrix} e_1 \\ x_1 \end{pmatrix}, \ldots, \begin{pmatrix} e_n \\ x_n \end{pmatrix}\right),$$

dove $e_1, \ldots, e_n$ sono i vettori che compongono la base canonica di $\mathbb{C}^n$ e $x_1, \ldots, x_n$ sono le colonne della matrice X, è $\mathcal{H}$-invariante. □

Si osservi, inoltre, che se X è una soluzione della $\mathcal{NARE}$ (1.22), allora vale la relazione

$$\mathcal{H}\begin{pmatrix} I_n & 0 \\ X & I_m \end{pmatrix} = \begin{pmatrix} I_n & 0 \\ X & I_m \end{pmatrix}\begin{pmatrix} A-BX & -B \\ 0 & XB-D \end{pmatrix},$$

da cui si ottiene

$$\begin{pmatrix} I_n & 0 \\ X & I_m \end{pmatrix}^{-1}\mathcal{H}\begin{pmatrix} I_n & 0 \\ X & I_m \end{pmatrix} = \begin{pmatrix} A-BX & -B \\ 0 & XB-D \end{pmatrix}.$$

Dall'uguaglianza precedente è possibile osservare che il polinomio caratteristico della matrice $\mathcal{H}$ è il prodotto dei polinomio caratteristici delle matrici $A-BX$ e $XB-D$. Dunque, indicando con $\sigma(M)$ lo spettro di una matrice M, si ha la seguente caratterizzazione degli autovalori della matrice Hamiltoniana:

$$\sigma(\mathcal{H}) = \sigma(A-BX) \cup \sigma(XB-D). \tag{1.24}$$

Inoltre, dall'equazione (1.23), si ha in particolare che il sottospazio $\mathcal{V}$ (utilizzando le notazioni del teorema 1.4.1) è il sottospazio $\mathcal{H}$-invariante relativo agli autovalori della matrice $A-BX$.

È possibile generalizzare il risultato presentato nel teorema 1.4.1 introducendo il concetto graph subspace definito da Lancaster e Rodman in [38]: un sottospazio k-dimensionale $\mathcal{V} \in \mathbb{C}^n$ si dice **graph subspace** se la relativa matrice $V \in \mathbb{C}^{n\times k}$ ha matrice $k \times k$ di testa invertibile. Alla luce di tale definizione si ha:

Teorema 1.4.2. *È possibile stabilire una corrispondenza biunivoca tra le soluzioni della $\mathcal{NARE}$ (1.22) e i graph subspace n-dimensionali $\mathcal{H}$-invarianti.*

Dimostrazione. Nel teorema 1.4.1 si è mostrato che se X è una soluzione della (1.22), allora il sottospazio n-dimensionale generato dalle colonne della matrice $\left(\begin{smallmatrix} I_n \\ X \end{smallmatrix}\right)$ è $\mathcal{H}$-invariante e dunque è banalmente un graph subspace.

Sia ora $\mathcal{V} \subseteq \mathbb{C}^{(n+m)}$ un graph subspace n-dimensionale $\mathcal{H}$-invariante generato dalle colonne della matrice $V \in \mathbb{C}^{(n+m)\times n}$. Vale dunque

- $\mathcal{H}V = V\Lambda$, con $\Lambda \in \mathbb{C}^{n\times n}$,
- $V := \left(\begin{smallmatrix} Y \\ Z \end{smallmatrix}\right)$, dove $Y \in \mathbb{C}^{n\times n}$ invertibile.

Ponendo $X := ZY^{-1}$, si ottiene pertanto

$$\mathcal{H}\begin{pmatrix} I_n \\ X \end{pmatrix} Y = \begin{pmatrix} I_n \\ X \end{pmatrix} Y\Lambda,$$

da cui

$$\mathcal{H}\begin{pmatrix} I_n \\ X \end{pmatrix} = \begin{pmatrix} I_n \\ X \end{pmatrix} Y\Lambda Y^{-1},$$

e quindi, per il teorema 1.4.1 X è soluzione della $\mathcal{NARE}$ (1.22).

Rimane da provare che tale corrispondenza è biunivoca, ovvero che la soluzione X sopra calcolata non dipenda dalla particolare scelta della matrice V. Si supponga allora che le colonne della matrice $\binom{Y_1}{Z_1}$ generino il sottospazio $\mathcal{V}$ con Y_1 invertibile. Esiste, dunque, una matrice $W^{n\times n}$ invertibile tale che $\binom{Y_1}{Z_1} = \binom{Y}{Z}W$, è chiaro quindi che

$$Z_1 Y_1^{-1} = ZWW^{-1}Y^{-1} = X.$$

□

Il teorema dianzi presentato ha quindi una notevole importanza: risolvere una $\mathcal{NARE}$, ovvero un'equazione matriciale non lineare, equivale a risolvere un problema di Algebra Lineare, quello di individuare i sottospazi invarianti della matrice Hamiltoniana $\mathcal{H}$.

È opportuno precisare il contenuto del teorema 1.4.2, esso afferma che esiste una bigezione tra soluzioni le soluzioni X delle $\mathcal{NARE}$ (1.22) e gli n autovalori autovalori di $\mathcal{H}$ corrispondenti allo spettro di $A - BX$, non che esiste una bigezione tra soluzioni ed un qualsiasi sottoinsieme dello spettro di $\mathcal{H}$. Se, infatti, $\mathcal{H}$ ha autovalori con molteplicità maggiore di uno, a n medesimi autovalori possono corrispondere differenti graph subspace e dunque differenti soluzioni della $\mathcal{NARE}$.

È possibile effettuare una analisi analoga per l'equazione duale della (1.22):

$$YCY + AY + YD - B = 0,$$

osservando che una soluzione Y verifica la relazione

$$\mathcal{H}\begin{pmatrix} Y \\ I_m \end{pmatrix} = -\begin{pmatrix} Y \\ I_m \end{pmatrix}(D + CY).$$

Si consideri ora la $\mathcal{CARE}$

$$C + XA + A^*X - XBX = 0, \tag{1.25}$$

dove A, B, $C \in \mathbb{C}^{n\times n}$ e le matrici B, C sono hermitiane, in tal caso la matrice Hamiltoniana $\mathcal{H}$ assume la forma

$$\mathcal{H} := \begin{pmatrix} A & -B \\ -C & -A^* \end{pmatrix}.$$

Essendo le $\mathcal{CARE}$ una particolare $\mathcal{NARE}$, i risultati sopra esposti rimangono a fortiori validi.

Il caso delle $\mathcal{GCARE}$ si tratta in modo leggermente diffente, ma sostanzialmente analogo. Si consideri la $\mathcal{GCARE}$

$$C + E^*XA + A^*XE - E^*XBXE = 0, \tag{1.26}$$

e si supponga che la matrice E sia invertibile. Vale il seguente risultato

Teorema 1.4.3. *Siano*

$$L := \begin{pmatrix} A & -B \\ -C & -A^* \end{pmatrix} \qquad K := \begin{pmatrix} E & 0 \\ 0 & E* \end{pmatrix},$$

e sia $\mathcal{P}(z)$ la matrix pencil definita da $\mathcal{P}(z) := L - zK$. Esiste una corrispondenza biunivoca tra le soluzioni X della $\mathcal{GCARE}$ (1.26) e i graph subspace di deflazione n-dimensionali relativi alla matrix pencil $\mathcal{P}(z)$.

Dimostrazione. Sia X soluzione di (1.26) e si osservi allora che valgono le seguenti relazioni

$$L\begin{pmatrix} I_n \\ XE \end{pmatrix} = \begin{pmatrix} A - BXE \\ -C - A^*XE \end{pmatrix}, \qquad K\begin{pmatrix} I_n \\ XE \end{pmatrix} = \begin{pmatrix} E \\ E^*XE \end{pmatrix}.$$

Essendo inoltre

$$\begin{pmatrix} A - BXE \\ -C - A^*XE \end{pmatrix} = \begin{pmatrix} E \\ E^*XE \end{pmatrix} E^{-1}(A - BXE),$$

il sottospazio $\mathcal{V}$ generato dalle colonne della matrice $\left(\begin{smallmatrix} I_n \\ XE \end{smallmatrix}\right)$, verifica la tesi del teorema.

Sia ora $\mathcal{V}$ un sottospazio di deflazione generato dalle colonne $\left(\begin{smallmatrix} Y \\ Z \end{smallmatrix}\right)$ con $Y \in \mathbb{C}^{n\times n}$ invertibile, allora esiste una matrice $\Lambda \in \mathbb{C}^{n\times n}$ tale che

$$L\begin{pmatrix} Y \\ Z \end{pmatrix} = \begin{pmatrix} AY - BZ \\ -CY - A^*Z \end{pmatrix} = K\begin{pmatrix} Y \\ Z \end{pmatrix}\Lambda = \begin{pmatrix} EY\Lambda \\ E^*Z\Lambda \end{pmatrix},$$

e dunque

$$\begin{pmatrix} AY - BZ \\ -CY - A^*Z \end{pmatrix} = \begin{pmatrix} EY\Lambda \\ E^*Z\Lambda \end{pmatrix}.$$

Uguagliando le righe a blocchi si ottiene

$$\Lambda = (EY)^{-1}(AY - BZ)$$

e

$$-CY - A^*Z = E^*Z(EY)^{-1}(AY - BZ)$$

pertanto, ponendo $X := Z(EY)^{-1}$ si ha

$$-C - A^*XE = EXA - E^*XBXE,$$

da cui la tesi. □

Anche per le $\mathcal{DARE}$ è possibile dare una descrizione delle soluzioni in termini di sottospazi di deflazione di particolari matrix pencil. Si considerino dunque la $\mathcal{DARE}$

$$A^*XA + Q - (C + BXA)^*(R + B^*XB)^{-1}(C + BXA) - X = 0, \tag{1.27}$$

e la $\mathcal{GDARE}$

$$A^*XA + Q - (C + BXA)^*(R + B^*XB)^{-1}(C + BXA) - E^*XE = 0, \tag{1.28}$$

dove le matrici R, Q sono matrici hermitiane. Vale Il seguente risultato

Teorema 1.4.4. *Siano*

$$L := \begin{pmatrix} A - BR^{-1}C & 0 \\ -Q + C^*R^{-1}C & E^* \end{pmatrix} \qquad K := \begin{pmatrix} E & BR^{-1}B^* \\ 0 & (A - BR^{-1}C)^* \end{pmatrix},$$

e sia $\mathcal{P}(z)$ la matrix pencil definita da $\mathcal{P}(z) := L - zK$. Esiste una corrispondenza biunivoca tra le soluzioni X della $\mathcal{GDARE}$ (1.28) e i graph subspace di deflazione n-dimensionali relativi alla matrix pencil $\mathcal{P}(z)$.

Per i dettagli della dimostrazione si rimanda a [37]. Si osservi che, ponendo nell'enunciato del teorema $E = I_n$, il risultato è applicabile, in particolare, alla $\mathcal{DARE}$ (1.27).

1.4.2 Esistenza e proprietà delle soluzioni estremali

In fase di presentazione delle $\mathcal{ARE}$ si è evidenziata l'importanza nelle applicazioni di particolari soluzioni, scopo della presente sezione è illustrare i principali risultati di esistenza per tali soluzioni ed enfatizzarne le principali proprietà.

Tra le soluzioni di una $\mathcal{NARE}$, particolare rilevanza assume la soluzione minimale non negativa, denotata con $X_{\min}$. Tale soluzione, infatti, consente di ottenere importanti informazioni riguardanti le proprietà spettrali della matrice Hamiltoniana $\mathcal{H}$ e dunque, sulla maggiore o minore efficacia che i metodi numerici presentano se applicati alla $\mathcal{NARE}$ in oggetto.

Nel seguito si suppongono note le proprietà più importanti e le caratterizzazioni delle M-matrici. Viene ora esposto un risultato che assicura, sotto determinate condizioni, l'esistenza della soluzione minimale non negativa:

Teorema 1.4.5. *Sia $\mathcal{M}$ la matrice dei coefficienti associata alla $\mathcal{NARE}$ (1.22). Se $\mathcal{M}$ è una M-matrice non singolare o una M-matrice singolare irriducibile, allora la $\mathcal{NARE}$ (1.22) ammette una soluzione minimale non negativa $X_{\min}$. Se $\mathcal{M}$ è non singolare, allora le matrici*

$$D - X_{\min}B \qquad A - BX_{\min}$$

sono M-matrici non singolari. Se $\mathcal{M}$ è irriducibile allora la soluzione $X_{\min}$ ha elementi positivi e le matrici

$$D - X_{\min}B \qquad A - BX_{\min}$$

sono M-matrici irriducibili.

Dimostrazione. Si presenta la dimostrazione nel caso in cui $\mathcal{M}$ sia una M-matrice non singolare, il caso in cui $\mathcal{M}$ è una M-matrice singolare irriducibile è analogo salvo piccoli accorgimenti.

L'idea di fondo della dimostrazione è generare una successione $\{X_k\}_{k\in\mathbb{N}}$ definita per ricorrenza che sfrutti una strategia di punto fisso. La convergenza si otterrà verificando le seguenti proprietà della $\{X_k\}_{k\in\mathbb{N}}$:

a) $X_{k+1} \geq X_k$ per $k = 0, 1, \ldots$, dunque la successione è non decrescente,

b) la successione $\{X_k\}_{k\in\mathbb{N}}$ è limitata superiormente,

c) esiste il $\lim_k X_k$ e, posto $S = \lim_k X_k$, si ha S soluzione non negativa della $\mathcal{NARE}$ (1.22),

d) la S definita al punto c) è effettivamente la soluzione minimale non negativa della $\mathcal{NARE}$, ovvero $S = X_{\min}$.

Si osservi che per la struttura della matrice $\mathcal{M}$ A e D risultano M-matrici. Siano

- $A = A_1 - A_2$, dove

$$a_{ij}^{(1)} := \begin{cases} a_{ij} & \text{se } i = j \\ 0 & \text{altrimenti} \end{cases} \qquad \text{e} \qquad a_{ij}^{(2)} := \begin{cases} -a_{ij} & \text{se } i \neq j \\ 0 & \text{altrimenti} \end{cases};$$

- $D = D_1 - D_2$, dove

$$d_{ij}^{(1)} := \begin{cases} d_{ij} & \text{se } i = j \\ 0 & \text{altrimenti} \end{cases} \qquad \text{e} \qquad d_{ij}^{(2)} := \begin{cases} -d_{ij} & \text{se } i \neq j \\ 0 & \text{altrimenti} \end{cases}.$$

Usando le matrici sopra introdotte la $\mathcal{NARE}$ (1.22) si riscrive come segue:

$$-C + XA_2 + D_2X + XBX = XA_1 + D_1X. \tag{1.29}$$

A partire dalla precedente relazione, si introduce la seguente successione definita per ricorrenza:

$$\begin{cases} X_0 = 0 \\ X_{k+1}A_1 + D_1X_{k+1} = -C + X_kA_2 + D_2X_k + X_kBX_k. \end{cases} \tag{1.30}$$

Utilizzando la funzione `vec` e le proprietà del prodotto di Kronecker è possibile trasformare l'equazione matriciale (1.30) in un sistema lineare:

$$(A_1^T \otimes I_m + I_n \otimes D_1)\texttt{vec}(X_{k+1}) = \texttt{vec}(-C + X_kA_2 + D_2X_k + X_kBX_k),$$

poiché, inoltre, la matrice $(A_1^T \otimes I_m + I_n \otimes D_1)$ è invertibile, si è in grado di esplicitare il termine $\texttt{vec}(X_{k+1})$:

$$vec(X_{k+1}) = (A_1^T \otimes I_m + I_n \otimes D_1)^{-1}\texttt{vec}(-C + X_kA_2 + D_2X_k + X_kBX_k).$$

Si procede con la dimostrazione seguendo i punti sopra indicati.

a) La prima parte si ottiene utilizzando il Principio di Induzione.

Per $k = 0$, il primo termine della successione è definito da

$$X_1 A_1 + D_1 X_1 = -C,$$

allora

$$\texttt{vec}(X_1) = (A_1^T \otimes I_m + I_n \otimes D_1)^{-1} \texttt{vec}(-C).$$

Si osservi quindi che $(A_1^T \otimes I_m + I_n \otimes D_1)^{-1}$, $-C \geq 0$ in quanto $(A_1^T \otimes I_m + I_n \otimes D_1)$ e $\mathcal{M}$ sono M-matrice. Si conclude pertanto

$$X_1 \geq 0 = X_0,$$

ovvero il passo base dell'induzione.

È immediato, applicando l'ipotesi induttiva, verificare le seguenti relazioni:

$$\begin{aligned}(X_{k+1} - X_k)A_1 + D_1(X_{k+1} - X_k) &= -C + X_k A_2 + D_2 X_k + X_k B X_k - X_k A_1 - D_1 X_k \\ &\geq -C + X_{k-1} A_2 + D_2 X_{k-1} + X_{k-1} B X_{k-1} - X_k A_1 - D_1 X_k \\ &= A_1 X_k + X_k D_1 - A_1 X_k - X_k D_1 = 0.\end{aligned}$$

Dunque, per la positività di $(A_1^T \otimes I_m + I_n \otimes D_1)^{-1}$, si ha $\texttt{vec}(X_{k+1} - X_k) \geq 0$ e quindi $X_{k+1} \geq X_k$, pertanto la successione $\{ X_k \}_{k \in \mathbb{N}}$ risulta effettivamente crescente.

b) Per ipotesi $\mathcal{M}$ è una M-matrice non singolare, allora, per le ben note proprietà delle M-matrici, esiste un vettore $v = \left(\begin{smallmatrix} v_1 \\ v_2 \end{smallmatrix}\right) > 0$ tale che

$$\mathcal{M} v = \begin{pmatrix} A & -B \\ C & D \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} > 0 \qquad \text{ovvero} \qquad \begin{cases} A v_1 - B v_2 = u_1 \\ C v_1 + D v_2 = u_2. \end{cases} \tag{1.31}$$

Se la successione verifica

$$X_k v_1 \leq v_2 - D_1^{-1} u_2 \qquad \text{per } k = 0, 1, \ldots \tag{1.32}$$

essendo $v_1 > 0$, gli elementi di X_k risultano necessariamente limitati per ogni k, allora l'intera successione $\{X_k\}_{k \in \mathbb{N}}$ risulta limitata superiormente.

Al solito si procede per induzione.

Per $k = 0$ occorre provare che: $v_2 - D_1^{-1} u_2 \geq 0$. Poiché $D^{-1} \geq D_1^{-1}$ è sufficiente dimostrare che

$$v_2 - D^{-1} u_2 \geq 0.$$

Per la (1.31) si ha:

$$v_2 - D^{-1} u_2 = -D^{-1} C v_1 \geq 0$$

in quanto C è non positiva, e D^{-1} è non negativa poiché D è una M-matrice non singolare.

Sia $k \geq 1$, dalla definizione di $\{X_k\}_{k \in \mathbb{N}}$ si ottiene:

$$X_{k+1} A_1 v_1 + D_1 X_{k+1} v_1 = -C v_1 + X_k A_2 v_1 + D_2 X_k v_1 + X_k B X_k v_1,$$

applicando l'ipotesi induttiva e sfruttando la non negatività di X_k, B, D_1^{-1}, D_2 e u_2 è possibile ricavare la seguente successione di disequazioni:

$$\begin{aligned}X_{k+1} A_1 v_1 + D_1 X_{k+1} v_1 &\leq -C v_1 + X_k A_2 v_1 + D_2(v_2 - D_1^{-1} u_2) + X_k B (v_2 - D_1^{-1} u_2) \\ &\leq -C v_1 + X_k A_2 v_1 + D_2 v_2 + X_k B v_2 \\ &\leq D v_2 - u_2 + X_k A_2 v_1 + D_2 v_2 + X_k B v_2 \\ &\leq D v_2 - u_2 + X_k A_2 v_1 + D_2 v_2 + X_k (A v_1 - u_1) \\ &\leq -u_2 + D_1 v_2 + X_k A_1 v_1 \\ &\leq -u_2 + D_1 v_2 + X_{k+1} A_1 v_1.\end{aligned}$$

Si conclude quindi che $X_{k+1} v_1 \leq v_2 - D_1^{-1} u_2$, come voluto.

c) Poiché per i punti a) e b) la successione $\{X_k\}_{k\in\mathbb{N}}$ è monotona non decrescente e limitata superiormente, esiste $S := \lim_k X_k$, tale limite è necessariamente non negativo in quanto tutti i termini della successione sono non negativi.
Dall'equazione (1.22) segue

$$\lim_{k\to\infty} XA_1 + D_1X = \lim_{k\to\infty} -C + XA_2 + D_2X + XBX,$$

pertanto

$$SA_1 + D_1S = -C + SA_2 + D_2S + SBS$$

e quindi la matrice S è una soluzione non negativa della $\mathcal{NARE}$ (1.22).

d) Si supponga esista una soluzione T della $\mathcal{NARE}$ (1.22) tale che $0 \le T < S$.
Si dimostra ancora per induzione che

$$T - X_k \ge 0 \qquad \text{per } k = 0, 1 \dots$$

e, dunque, $T - S \ge 0$ da cui l'assurdo. Per $k = 0$ la tesi è vera in quanto si è supposto T non negativa.

Si supponga ora che valga $T - X_k \ge 0$, allora sottraendo membro a membro le relazioni

$$TA_1 + D_1T = -C + TA_2 + D_2T + TBT$$
$$X_{k+1}A_1 + D_1X_{k+1} = -C + X_kA_2 + D_2X_k - X_kBX_k,$$

si ha

$$(T - X_{k+1})A_1 + D_1T - X_{k+1}) = (T - X_k)A_2 + D_2(T - X_k) + TBT - X_kBX_k \ge 0.$$

Utilizzando la funzione `vec` si esplicita il termine $T - X_{k+1}$:

$$\texttt{vec}(T - X_{k+1}) = (A_1^T \otimes I_m + I_n \otimes D_1)^{-1} \texttt{vec}((T - X_k)A_2 + D_2(T - X_k) + TBT - X_kBX_k) \ge 0 \quad (1.33)$$

e quindi la tesi è provata per induzione, dunque si conclude $S = X_{\min}$.

Per concludere la dimostrazione del teorema occorre provare che

$$A - BX_{\min} \qquad \text{e} \qquad D - X_{\min}B$$

sono M-matrici: si riporta solo la prima la verifica in quanto le seconda è del tutto analoga.

Le matrici A, B, $X_{\min}$ sono non negative pertanto $A - BX_{\min}$ è una Z-matrice, inoltre per quanto dimostrato nel punto b)

$$X_kv_1 \le v_2 - D_1^{-1}u_2,$$

per ogni $k \in \mathbb{N}$, pertanto $X_{\min}v_1 \le v_2 - D_1^{-1}u_2$, ovvero $X_{\min}v_1 + D_1^{-1}u_2 \le v_2$.
Allora:

$$(A - BX_{\min})v_1 = Av_1 - BX_{\min}v_1 \ge Av_1 - Bv_2 = u_1 > 0,$$

dove nell'ultima uguaglianza e nell'ultima disuguaglianza si è fatto riferimento alla (1.31). La tesi segue dal teorema è evidente per le note caratterizzazioni delle M-matrici. □

Si osservi che tale dimostrazione è costruttiva, ovvero oltre a provare l'esistenza di $X_{\min}$, fornisce un metodo per individuare tale soluzione. Tale metodo, o meglio, tale classe di metodi (quella delle iterazioni funzionali), saranno approfonditi nel paragrafo 2.3.

Per quanto riguarda le $\mathcal{CARE}$, come esposto nella sezione 1.1.2, le soluzioni di maggior interesse sono le soluzioni estremali X_- e X_+. I teoremi che seguono illustrano sotto quali condizioni è assicurata l'esistenza e l'unicità di tali soluzioni, per una trattazione più dettagliata e completa si consulti [37].

Teorema 1.4.6. *Si supponga $B \succeq 0$. Allora esiste un'unica soluzione T della $\mathcal{CARE}$ (1.25) tale che*

$$\sigma(A - BT) \subseteq \mathbb{C}_{l,0}$$

se e solo se

*i) la coppia di matrici (A, B) è **c-stabilizzabile**, ovvero se*

$$\texttt{rank}([A - \lambda I_n, B]) = n$$

per ogni $\lambda \in \mathbb{C}_{l,0}$,

ii) gli autovalori di $\mathcal{H}$ immaginari puri hanno, se esistono, tutti molteplicità pari.

Teorema 1.4.7. *Si supponga $B \succeq 0$. Allora esiste un'unica soluzione S della $\mathcal{CARE}$ (1.25) tale che*

$$\sigma(A - BS) \subseteq \mathbb{C}_{r,0}$$

se e solo se

*i) la coppia di matrici $(-A, B)$ è **c-stabilizzabile**,*

ii) gli autovalori di $\mathcal{H}$ immaginari puri hanno, se esistono, tutti molteplicità pari.

I precedenti teoremi sono propedeutici per il seguente

Teorema 1.4.8. *Si supponga che $B \succeq 0$ e che esistano le soluzioni T, S introdotte nel teorema 1.4.6 e nel teorema 1.4.7. Se X è una soluzione hermitiana della $\mathcal{CARE}$ (1.25), allora $S \preceq X \preceq T$, ovvero*

$$X_- = S \qquad X_+ = T.$$

Il teorema 1.4.6 e il teorema 1.4.7 danno inoltre un'importante caratterizzazione delle matrici X_+ e X_-. Utilizzando una terminologia consolidata nella teoria dei controlli una soluzione X della $\mathcal{CARE}$ (1.25) si dice

- **c-stabilizzante** se $\sigma(A - BX) \subseteq \mathbb{C}_l$,
- **quasi c-stabilizzante** se $\sigma(A - BX) \subseteq \mathbb{C}_{l,0}$,
- **c-antistabilizzante** se $\sigma(A - BX) \subseteq \mathbb{C}_r$,
- **quasi c-antistabilizzante** se $\sigma(A - BX) \subseteq \mathbb{C}_{r,0}$.

Pertanto con le definizioni testé introdotte, la matrice X_+ risulta quasi c-stabilizzante, mentre la matrice X_- è quasi c-antistabile.

Lo studio sull'esistenza e sulle proprietà delle soluzioni estremali delle $\mathcal{DARE}$ è estremamente tecnica e esula dagli scopi di tale lavoro. Si elencano, per dovere di completezza, esclusivamente i risultati principali, per una trattazione più esaustiva e corredata di dimostrazioni si rimanda a [38]

Si consideri la $\mathcal{DARE}$ (1.27) e si definisca la seguente funzione matriciale

$$\Psi(z) := \begin{pmatrix} B^*(z^{-1}I_n - A^*)^{-1} & I_m \end{pmatrix} \begin{pmatrix} Q & C^* \\ C & R \end{pmatrix} \begin{pmatrix} (z^{-1}I_n - A)^{-1}B \\ I_m \end{pmatrix}, \tag{1.34}$$

Si supponga che le matrici A e $D = R - CA^{-1}B$ siano non singolari e si ponga

$$W := C - B^*A^{-*}Q,$$

si definisce quindi

$$U := \begin{pmatrix} A - BD^{-*}W & -BD^{-*}B^*A^{-*} \\ A^{-*}(-Q + C^*D^{-1}W) & A^{-*}(I_n + C^*D^{-*}B^*A^{-*}) \end{pmatrix}.$$

Alla luce delle precedenti notazioni, è possibile dare la seguente caratterizzazione delle matrici X_+ e X_-:

Teorema 1.4.9. *Si supponga che*

i) la coppia (A, B) *sia* ***controllabile****, ovvero che*

$$\texttt{rank}([A - \lambda I_n, B]) = n$$

per ogni $\lambda \in \mathbb{C}$*,*

ii) le matrici A *e* $D = R - CA^{-1}B$ *siano non singolari,*

iii) esista $\nu \in \mathbb{C}$ *di modulo unitario, tale che* $\Psi(\nu) \succeq 0$*.*

Allora la $\mathcal{DARE}$ (1.27) *ammette una soluzione hermitiana se e solo se gli autovalori di* U *di modulo unitario, se esistono, hanno tutti molteplicità pari.*

Teorema 1.4.10. *Si supponga che*

i) la coppia (A, B) *sia controllabile,*

ii) esista $\nu \in \mathbb{C}$ *di modulo unitario, tale che* $\Psi(\nu) \succeq 0$*,*

iii) esista una matrice $K \in \mathbb{C}^{m\times n}$ *tale che*

$$A - BK \qquad R - (C - RK)(A - BK)^{-1}B$$

siano non singolari.

Se la $\mathcal{DARE}$ (1.27) *ha una soluzione hermitiana, allora esistono ed uniche le soluzioni estremali* X_+ *e* X_-*.*

Mutuando le definizioni date per le $\mathcal{DARE}$, una soluzione X della $\mathcal{DARE}$ (1.27) si dice

- **d-stabilizzante** se $\sigma(A - BX) \subseteq \mathcal{D}$,
- **quasi d-stabilizzante** se $\sigma(A - BX) \subseteq \bar{\mathcal{D}}$,
- **d-antistabilizzante** se $\sigma(A - BX) \subseteq \bar{\mathcal{D}}^c$,
- **quasi d-antistabilizzante** se $\sigma(A - BX) \subseteq \mathcal{D}^c$.

Come nel caso precedente, è possibile dimostrare che la soluzione massimale X_+ risulta quasi d-stabilizzante, mentre la soluzione minimale X_- è quasi d-antistabile.

1.4.3 Proprietà spettrali della matrice Hamiltoniana e drift di una $\mathcal{NARE}$

Il teorema 1.4.5 mostra che, data la $\mathcal{NARE}$ (1.22), e detta $X_{\min}$ la sua soluzione minimale non negativa, la matrice $A - BX_{\min}$ è una M-matrice, e dunque, ha tutti gli autovalori con parte reale non negativa. Inoltre, per le osservazioni trattate nella sezione 1.4.1, lo spettro di $\mathcal{H}$ contiene lo spettro della matrice $A - BX_{\min}$, e pertanto la matrice Hamiltoniana $\mathcal{H}$ presenta n autovalori con parte reale non negativa. Nel seguito si indagherà sulla natura dei rimanenti m autovalori dello spettro di $\mathcal{H}$.

Teorema 1.4.11. *Sia* $\mathcal{M}$ *la matrice dei coefficienti associata alla* $\mathcal{NARE}$ (1.22) *una M-matrice non singolare. Allora la matrice hamiltoniana* $\mathcal{H}$ *ha un* (m, n) *c-splitting forte.*

Se $\mathcal{M}$ *è una M-matrice singolare irriducibile, allora* $\mathcal{H}$ *ha un* (m, n) *c-splitting e il solo autovalore che giace sull'asse immaginario è* $\lambda = 0$*.*

Dimostrazione. Si consideri dapprima il caso in cui $\mathcal{M}$ sia una M-matrice non singolare, dunque esiste $v > 0$ tale che $\mathcal{M}v = u > 0$.

Si ponga

$$\hat{\mathcal{M}} = (diag(v))^{-1}\mathcal{M}(diag(v))$$

e si osservi che $\hat{\mathcal{M}}$ è una Z-matrice.

È immediato verificare che:

$$\hat{\mathcal{M}}\begin{pmatrix}1\\ \vdots \\ 1\end{pmatrix} = (diag(v))^{-1}\mathcal{M}v > 0,$$

allora

$$\begin{cases} \sum_{j=1}^{n} \hat{m}_{ij} > 0 \quad i = 1, \ldots, n+m \\ \\ \hat{m}_{ii} > 0 \quad i = 1, \ldots, n+m \\ \hat{m}_{ij} \leq 0 \quad i = 1, \ldots, n+m. \end{cases}$$

Risulta quindi

$$\begin{cases} \hat{m}_{ii} > \sum_{j=1, j\neq i}^{n+m} |\hat{m}_{ij}| \quad i = 1, \ldots, n+m \\ \\ \hat{m}_{ii} > 0 \quad i = 1, \ldots, n+m. \end{cases}$$

Per il teorema di Gerschgorin gli autovalori di $\hat{\mathcal{M}}$ (e dunque di $\mathcal{M}$) hanno tutti parte reale positiva.

La matrice $\mathcal{H}$ corrisponde per le prime n righe a $\mathcal{M}$, per le restanti m a $-\mathcal{M}$, dunque presenta n autovalori con parte reale positiva e m con parte reale negativa, ovvero un ha un (n, m) d-splitting forte.

Per quanto riguarda il caso in cui $\mathcal{M}$ sia una M-matrice singolare irriducibile si segue il medesimo procedimento. Per il teorema di Perron-Frobenius esiste $v > 0$ tale che $\mathcal{M}v = 0$.

Si considera la medesima trasformazione per similitudine della precedente verifica e si ottiene il sistema:

$$\begin{cases} \sum_{j=1}^{n} \hat{m}_{ij} = 0 \quad i = 1, \ldots, n+m \\ \\ \hat{m}_{ii} \geq 0 \quad i = 1, \ldots, n+m \\ \\ \hat{m}_{ij} \leq 0 \quad i = 1, \ldots, n+m \end{cases}$$

risulta quindi

$$\begin{cases} \hat{m}_{ii} \geq \sum_{j=1, j\neq i}^{n+m} |\hat{m}_{ij}| \quad i = 1, \ldots, n+m \\ \\ \hat{m}_{ii} \geq 0 \quad i = 1, \ldots, n+m. \end{cases}$$

Sempre per il teorema di Ghershgorin $\mathcal{M}$ ha $n+m$ autovalori non negativi, e per le medesime argomentazioni utilizzate precedentemente $\mathcal{H}$ ha n autovalori con parte reale non negativa e m autovalori con parte reale non positiva, dunque l'unico autovalore sull'asse Immaginario è $\lambda = 0$. □

Per quanto dimostrato nel teorema 1.4.11 se $\mathcal{M}$ è una M-matrice singolare irriducibile è possibile discernere tre diversi casi:

positivo ricorrente: $Re(\lambda_{n+m}) \leq \cdots \leq Re(\lambda_{n+1}) < 0 = \lambda_n \leq \cdots \leq Re(\lambda_1)$;

transiente: $Re(\lambda_{n+m}) \leq \cdots \leq \lambda_{n+1} = 0 < Re(\lambda_n) \leq \cdots \leq Re(\lambda_1)$;

ricorrente nullo: $Re(\lambda_{n+m}) \leq \cdots < \lambda_{n+1} = 0 = \lambda_n < \cdots \leq Re(\lambda_1)$.

Se $\mathcal{M}$ è una M-matrice singolare irriducibile per il teorema di Perron-Frobenius

$$\exists v > 0 \quad \text{tale che} \quad \mathcal{M}v = 0 \qquad \text{e} \qquad \exists u > 0 \quad \text{tale che} \quad u^T\mathcal{M} = 0 \tag{1.35}$$

dove $u^T v = 1$.
A partire da questa osservazione si definisce un coefficiente che indica la localizzazione degli autovalori della matrice $\mathcal{H}$, e determina, come mostrato nei capitoli successivi, se i metodi risolutivi applicati alla $\mathcal{NARE}$ in oggetto sono più o meno efficaci: il drift. Data la $\mathcal{NARE}$ (1.22), si supponga che la matrice dei coefficienti $\mathcal{M}$ sia una M-matrice singolare irriducibile. Sia data la seguente partizione dei vettori u, v introdotti in (1.35):

$$u = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \qquad \text{e} \qquad v = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix},$$

con u_1, $v_1 \in \mathbb{R}^n$ e u_2, $v_2 \in \mathbb{R}^m$. Si definisce **drift della $\mathcal{NARE}$** o **deriva della $\mathcal{NARE}$** lo scalare

$$\mu = u_2^T v_2 - u_1^T v_1 = \begin{pmatrix} u_1 & u_2 \end{pmatrix} \begin{pmatrix} -I_n & 0 \\ 0 & I_m \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}.$$

Dal valore del drift è possibile risalire alla natura della $\mathcal{NARE}$ ([22, 26]), in particolare si ha:

Teorema 1.4.12. *Si supponga che la matrice dei coefficienti $\mathcal{M}$ associata alla $\mathcal{NARE}$ (1.22) sia singolare irriducibile, allora valgono le seguenti implicazioni:*

- *$\mu < 0$ se e solo se la $\mathcal{NARE}$ è positiva ricorrente,*
- *$\mu > 0$ se e solo se la $\mathcal{NARE}$ è transiente ,*
- *$\mu = 0$ se e solo se la $\mathcal{NARE}$ è ricorrente nulla ed esiste un solo autovettore di $\mathcal{H}$, a meno di moltiplicazione per scalare, relativo all'autovalore $\lambda = 0$.*

Dimostrazione. Sia $\mathcal{M}$ una matrice singolare irriducibile e siano v, u i vettori definiti in (1.35), allora si osservi che

$$\mathcal{H}v = \begin{pmatrix} A & -B \\ -C & -D \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = 0,$$

e

$$\begin{pmatrix} u_1^T & -u_2^T \end{pmatrix} \mathcal{H} = \begin{pmatrix} u_1^T & -u_2^T \end{pmatrix} \begin{pmatrix} A & -B \\ -C & -D \end{pmatrix} = 0,$$

dunque i vettori v e $\left(\begin{smallmatrix} u_1 \\ -u_2 \end{smallmatrix}\right)$ sono rispettivamente autovalore destro e sinistro della matrice Hamiltoniana $\mathcal{H}$ relativo all'autovalore $\lambda = 0$.

Se il drift μ della $\mathcal{NARE}$ è nullo, allora i

$$-\mu = \begin{pmatrix} u_1^T & -u_2^T \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = 0$$

ovvero i due autovettori sono ortogonali allora necessariamente $\lambda_{n+1} = 0 = \lambda_n$, quindi la $\mathcal{NARE}$ è ricorrente nulla.

Se $\mu \neq 0$ allora uno solo tra λ_n e λ_{n+1} è vincolato ad essere nullo. Sia $X_{\min}$ la soluzione minimale non negativa della $\mathcal{NARE}$ (1.22), allora valgono le seguenti uguaglianze:

$$\begin{pmatrix} I_n & 0 \\ -X_{\min} & I_m \end{pmatrix} \mathcal{H} = \begin{pmatrix} A - BX_{\min} & -B \\ 0 & -(D - X_{\min}B) \end{pmatrix} \begin{pmatrix} I_n & 0 \\ -X_{\min} & I_m \end{pmatrix},$$

$$\mathcal{H} \begin{pmatrix} I_n & 0 \\ X_{\min} & I_m \end{pmatrix} = \begin{pmatrix} I_n & 0 \\ X_{\min} & I_m \end{pmatrix} \begin{pmatrix} A - BX_{\min} & -B \\ 0 & -(D - X_{\min}B) \end{pmatrix}.$$

Moltiplicando la prima equazione per v, ed essendo v autovettore destro, si ottiene

$$\begin{pmatrix} I_n & 0 \\ -X_{\min} & I_m \end{pmatrix} \mathcal{H} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} * \\ -(D - X_{\min}B)(-X_{\min}v_1 + v_2) \end{pmatrix} = 0, \tag{1.36}$$

analogamente dalla seconda equazione si ha

$$\begin{pmatrix} u_1^T & -u_2^T \end{pmatrix} \mathcal{H} \begin{pmatrix} I_n & 0 \\ X_{\min} & I_m \end{pmatrix} = \begin{pmatrix} (u_1^T - u_2^T X_{\min})(A - BX_{\min}) & * \end{pmatrix} = 0. \tag{1.37}$$

Se $\lambda_{n+1} \neq 0$, allora la matrice $A - BX_{\min}$ risulta invertibile, sicché la (1.37) implica

$$u_1^T = u_2^T X_{\min}. \tag{1.38}$$

Inoltre, sfuttando quanto dimostrato nel punto b) del teorema 1.4.5, si ha

$$X_{\min}v_1 \leq v_2 - D^{-1}u_2,$$

e dunque, a fortiori,

$$X_{\min}v_1 \leq v_2,$$

essendo D una M-matrice e $u_2 > 0$. Si conclude quindi

$$\mu = u_2^T v_2 - u_1^T v_1 = u_2^T v_2 - u_2^T X_{\min} v_1 \leq 0,$$

ovvero $\mu > 0$, in quanto si è supposto $\mu \neq 0$.

Sfruttando argomentazioni del tutto analoghe, si prova anche il caso in cui $\mu \neq 0$ e $\lambda_n \neq 0$. □

Come detto in precedenza il drift esprime un paramentro per valutare se i metodi numerici hanno efficacia (ovvero se convergono, e a quale velocità convergono) se applicati alla $\mathcal{NARE}$ in oggetto. Nei capitoli successivi sarà chiarito che i metodi risultano meno efficienti nel caso ricorrente nullo; per quanto riguarda il caso positivo ricorrente o transiente si osserva che questi sono interscambiabili, ovvero è possibile ricondursi da un caso all'altro per dualità.

L'enunciato del teorema precedente può essere riformulato in termini di algebra lineare e dunque dare una caratterizzazione geometrica del drift di una $\mathcal{NARE}$:

Teorema 1.4.13. *Siano $\mathcal{M}$ la matrice dei coefficienti e $\mathcal{H}$ la Matrice Hamiltoniana associate alla $\mathcal{NARE}$ (1.22), e si supponga che $\mathcal{M}$ sia una M-matrice non singolare o singolare irriducibile. Vale una delle seguenti implicazioni:*

- *se $\mathcal{M}$ è invertibile allora $\mathcal{H}$ ha un (m, n) c-splitting forte e $\mathcal{H}$ ha un unico autospazio m-dimensionale c-stabile e un unico autospazio n-dimensionale c-antistabile,*
- *altrimenti:*
  - *se $\mu < 0$ allora $\mathcal{H}$ ha un (m, n) c-splitting proprio e $\mathcal{H}$ ha un unico autospazio m-dimensionale c-stabile e un unico autospazio n-dimensionale c-antistabile debole,*
  - *se $\mu > 0$ allora $\mathcal{H}$ ha un (m, n) c-splitting proprio e $\mathcal{H}$ ha un unico autospazio m-dimensionale c-stabile debole e un unico autospazio n-dimensionale c-antistabile,*
  - *se $\mu = 0$ allora $\mathcal{H}$ ha un (m, n) c-splitting debole e $\mathcal{H}$.*

In ultima analisi si studiano le proprietà spettrali della matrice Hamiltoniana associata ad una $\mathcal{CARE}$.

Sia $\mathcal{J} := \left(\begin{smallmatrix} 0 & I_n \\ -I_n & 0 \end{smallmatrix}\right)$, allora la matrice Hamiltoniana

$$\mathcal{H} := \begin{pmatrix} A & -B \\ -C & -A^* \end{pmatrix},$$

verifica la seguente relazione

$$\mathcal{J}^*\mathcal{H}\mathcal{J} = -\mathcal{H}^*. \tag{1.39}$$

La relazione precedente, in particolare, implica che le matrici $\mathcal{H}$ e $-\mathcal{H}^*$ hanno lo stesso spettro e dunque se $\lambda \in \sigma(\mathcal{H})$, allora $-\bar{\lambda} \in \sigma(\mathcal{H})$. Si conclude, quindi, che gli autovalori λ_i $(i = 1, \ldots, 2n)$ di $\mathcal{H}$ verificano la catena di disuguaglianze

$$Re(\lambda_{2n}) \leq \cdots \leq \Re(\lambda_{n+1}) \leq 0 \leq Re(\lambda_n) \leq \cdots \leq Re(\lambda_1),$$

pertanto la matrice Hamiltoniana $\mathcal{H}$ ha nel caso delle $\mathcal{CARE}$ un (n, n) c-splitting.

Capitolo 2

Metodi risolutivi classici

Indice

Nel presente capitolo sono presentati i metodi classici per la risoluzione di equazioni algebriche di Riccati. È bene sottolineare che l'interesse di tali algoritmi è più storico e teorico che prettamente numerico. Suddetti metodi, infatti, non offrono in termini di efficienza computazionale e di stabilità numerica, le medesime garanzie assicurate dai *doubling algorithms* illustrati nel capitolo succesivo. Si è pertanto preferito omettere, per tali algoritmi, la parte dedicata all'implementazione in quanto le prestazioni fornite dai metodi classici risultano significativamente peggiori di quelle ottenute dai doubling algorithms.

Il paragrafo 2.1 espone brevemente i metodi per la risoluzione delle *equazioni di Sylvester, Lyapunov e Stein*: gli *algoritmi di Bartels e Stewart*. Nella descrizione degli algoritmi di Bartels e Stewart è essenziale sfruttare le ben note proprietà della *funzione* `vec`, del *prodotto di Kronecker* e della *fattorizzazione di Schur*. La risoluzione delle equazioni di Sylvester, Lyapunov e Stein è fondamentale nello sviluppo dei metodi classici, pertanto tale praragrafo può considerarsi propedeutico per successivi.

Il paragrafo 2.2 illustra il *metodo di Schur*, metodo che si propone di individuare i *graph invariant subspace* e i *graph deflating subspace* introdotti nella sezione 1.4.1. Alla base di tale motodo e dunque, a giustificazione della sua denominazione, vi sono la fattorizzazione di Schur e la *fattorizzazione generalizzata di Schur*. Vengono quindi analizzate le proprietà numeriche degli algoritmi.

Il paragrafo 2.3 presenta i *metodi di iterazione funzionale* basati su una tecnica di punto fisso. Tali metodi ricalcano sostanzialmente la dimostrazione del teorema 1.4.5, con opportune scelte delle matrici A_1, A_2, D_1, D_2. Per taluna di tali scelte vengono confrontati costi computazionali e velocità di convergenza alla soluzione $X_{\min}$.

Il paragrafo 2.4 mostra un'applicazione del *metodo di Newton* alle equazioni di Riccati. Sono dapprima richiamate le generalità del metodo di Newton, sono poi introdotte la *derivata di Fréchet* e l'*operatore di Riccati*. Si dimostra la convergenza del metodo alla soluzione estremale e viene descritta l'equazione definita per ricorrenza da risolvere ad ogni passo del metodo di Newton. Sono quindi illustrate le proprietà numeriche degli algoritmi: costo computazionale, ordine di convergenza, stabilità numerica.

2.1 Metodi risolutivi per equazioni di Sylvester, Lyapunov e Stein

Le equazioni di Sylvester, Lyapunov e Stein, introdotte nella sezione 1.1.4, sono equazioni matriciali lineari molto utilizzate in teoria dei controlli. La prima formulazione di tali equazioni è descritta nel 1884 nel testo *Sur l'equations en matrices px = xq* del matematico britannico James Joseph Sylvester (1814-1897).

I metodi illustrati nel presente paragrafo sono metodi standard, basati esclusivamente sull'utilizzo appropriato della funzione `vec`, del prodotto di Kronecker e della fattorizzazione di Schur. Se le matrici che intervengono nelle equazioni sono sparse e di grandi dimensioni e hanno rango basso, i metodi adottati hanno tutt'altra filosofia, e si basano su tecniche analoghe a quelle illustrate nel capitolo 4 ([42]).

2.1.1 Equazioni di Sylvester

Si consideri l'equazione di Sylvester nell'indeterminata X

$$AX + XB = Q, \tag{2.1}$$

dove $X \in \mathbb{C}^{m\times n}$, $A \in \mathbb{C}^{m\times m}$, $B \in \mathbb{C}^{n\times n}$ e $Q \in \mathbb{C}^{m\times n}$.

Utilizzando le proprietà della funzione `vec` e del prodotto di Kronecker, è possibile riformulare l'equazione (2.1) come

$$(I_n \otimes A + B^T \otimes I_m)\mathtt{vec}(X) = \mathtt{vec}(Q). \tag{2.2}$$

Si osservi che, per le ben note proprietà del prodotto di Kronecker, la matrice $(I_n \otimes A + B^T \otimes I_m)$ ha autovalori del tipo $\lambda_i - \mu_j$ dove λ_i per $i = 1, \dots m$ sono gli autovalori di A, mentre μ_j per $j = 1, \dots n$ sono gli autovalori della matrice B. Pertanto il sistema lineare (2.2) ammette una e una sola soluzione se e solo se le matrici A e $-B$ non hanno autovalori comuni.

Risolvere, dunque, l'equazione di Sylvester (2.1), equivale a risolvere un sistema lineare di dimensione mn. Sebbene tale strategia risolutiva sia semplice ed immediata, dal punto di vista numerico non risulta praticabile in quanto, applicando l'algoritmo di **eliminazione gaussiana** al sistema (2.2), si avrebbe un costo computazionale di $O((mn)^3)$ operazioni elementari.

Un metodo di gran lunga più efficiente è l'**algoritmo di Bartels e Stewart** presentato nel 1972 in [3]. Utilizzando la **fattorizzazione reale di Schur**, è possibile porre

$$A = UTU^T \qquad B^T = VSV^T,$$

dove le matrice U e V sono ortogonali e le matrici T e S sono triangolari superiori a blocchi.

Si osservi che sfuttando le fattorizzazioni appena introdotte, l'equazione (2.1) si riscrive come

$$UTU^T X + XVS^T V^T = Q,$$

da cui, ponendo $\tilde{X} := (U^T XV)$ e $\tilde{Q} := (U^T QV)$ si ottiene

$$T\tilde{X} + \tilde{X}S^T = \tilde{Q}. \tag{2.3}$$

Si ricava pertanto una nuova equazione di Sylvester nella incognita $\tilde{X}$, in una forma molto particolare in quanto le matrici T e S risultano triangolari superiori a blocchi. Siano p e q i blocchi che compongono rispettivamente le matrici T e S, allora, dall'equazione (2.3), si ricava

$$T_{kk}\tilde{X}_{kl} + \tilde{X}_{kl}S_{ll}^T = \tilde{Q}_{kl} - \sum_{i=k+1}^{p} T_{ki}\tilde{X}_{ki} - \sum_{j=l+1}^{q} \tilde{X}_{kj}S_{jl}^T,$$

per $k = p, \ldots, 1$ e $l = q, \ldots, 1$, dove $\tilde{X}_{kl}$ denota il blocco (k, l) della matrice $\tilde{X}$.

È possibile applicare la medesima strategia considerando la **fattorizzazione complessa di Schur** ed ottenere una forma semplificata dell'algoritmo in quanto le matrici T e S risultano triangolari supriori e non triangolari superiori a blocchi.

Il costo computazionale dell'algoritmo di Bartels e Stewart, se $m = n$, ammonta a circa $60n^3$ operazioni elementari che comprendono il costo per il calcolo delle fattorizzazioni di Schur delle matrici A e B^T, ciascuna delle quali ha costo pari a circa $25n^3$ operazioni elementari.

2.1.2 Equazioni di Lyapunov

Si consideri l'equazione di Lyapunov

$$AX + XA^* = Q, \tag{2.4}$$

dove X, A, $Q \in \mathbb{C}^{n\times n}$ e Q è una matrice hermitiana.

Un primo approccio per risolvere tale equazione è passare attraverso la funzione *vect* e riscrivere la (2.4) come

$$\big(I_n \otimes A + (A^*)^T \otimes I_n\big)\mathtt{vec}(X) = \mathtt{vec}(Q). \tag{2.5}$$

Si osservi che per le ben note proprietà del prodotto di Kronecker gli autovalori della matrice $\big(I_n \otimes A + (A^*)^T \otimes I_n\big)$ sono della forma $\lambda + \bar{\mu}$, dove λ e μ sono autovalori di A, è evidente, quindi, che il sistema precedente ammette una e una sola soluzione se e solo se la matrice A non ha coppie di autovalori simmetrici rispetto all'asse immaginario, ovvero non esistono λ e μ nello spettro di A tali che

$$\lambda = -\bar{\mu}.$$

Per esempio, quindi, se la matrice A è c-stabile o c-antistabile o è una M-matrice, allora verifica la proprietà spettrale voluta.

Come nel caso delle equazioni di Sylvester, risolvere direttamente il sistema (2.5) con il metodo dell'eliminazione gaussiana non è numericamente efficiente in quanto richiede un costo computazionale di $O((mn)^3)$ operazioni elementari.

Una strategia migliore è rappresentata dell'**algoritmo di Bartels e Stewart**. Sia quindi $A = UTU^*$ con U unitaria e T triangolare superiore la fattorizzazione complessa di Schur della matrice A e si adoperi tale decomposizione per riscrivere la (2.4):

$$UTU^*X + XUT^*U^* = Q.$$

Ponendo $\tilde{X} := U^*XU$ e $\tilde{Q} := U^*QU$, è possibile riscrivere la precedente equazione come

$$T\tilde{X} + \tilde{X}T^* = \tilde{Q}. \tag{2.6}$$

Si ottiene, quindi, una nuova equazione di Lyapunov, definita a partire dalla matrice triangolare superiore T e dalla matrice hermitiana $\tilde{Q}$. Tale formulazione permette di calcolare facilmente ciscun termine della matrice $\tilde{X}$:

$$t_{kk}\tilde{x}_{kl} + \tilde{x}_{kl}\bar{t}_{ll} = \tilde{q}_{kl} - \sum_{i=k+1}^{n} t_{ki}\tilde{x}_{il} - \sum_{j=l+1}^{n} \tilde{x}_{jl}\tilde{t}_{jl},$$

con k, $l = n, \ldots 1$.

In analogia con quanto esposto nella sezione 2.1.1, il costo dell'algoritmo di Bartels e Stewart per risolvere una Equazione di Lyapunov è di circa $35n^3$ operazioni elementari comprensivo del calcolo della fattorizzazione di Schur della matrice A.

2.1.3 Equazioni di Stein

Si consideri l'equazione di Stein

$$X - AXB = Q, \tag{2.7}$$

dove X, $Q \in \mathbb{C}^{m\times n}$, $A \in \mathbb{C}^{m\times m}$, $B \in \mathbb{C}^{n\times n}$. Al solito, servendosi della funzione *vect* e delle proprietà del prodotto di Kronecker, è possibile riscrivere l'equazione precedente nella forma più compatta:

$$(I_{mn} - B^T \otimes A)\mathtt{vec}(X) = \mathtt{vec}(Q). \tag{2.8}$$

Lo spettro della matrice $(I_m n - B^T \otimes A)$ è composto da elementi del tipo $1 - \lambda\mu$, con λ e μ autovalori rispettivamente di A e B, pertanto il sistema (2.8) ammette una e una sola soluzione se e solo se $\lambda\mu \neq 1$ per ogni coppia di autovalori λ di A e μ di B.

Anche in tal caso risolvere l'equazione di Stein passando attraverso il sistema (2.8) risulta una strategia inefficiente, è opportuno, invece, utilizzare l'**algoritmo di Bertels e Stewart**. Siano, quindi,

$$A = UTU^* \qquad B^T = V^T S^T (V^T)^*$$

con U, V^T matrici unitarie e T, S^T matrici triangolari superiori, le fattorizzazioni complesse di A e B^T. A partire dalle decomposizioni testé introdotte, la (2.7) si riscrive come

$$X - UTU^*XV^*SV = Q,$$

da cui si ricava

$$\tilde{X} - T\tilde{X}S = \tilde{Q} \tag{2.9}$$

dove si è posto $\tilde{X} := U^*XV^*$, $\tilde{Q} := U^*QV^*$. La (2.9) è, dunque, una equazione di Stein definita dai termine triangolare superiore T e dalla matrice triangolare inferiore S, pertanto si ha

$$\tilde{x}_{kl} - \sum_{j=l}^{n} (T\tilde{X})_{kj} s_{jl} = \tilde{q}_{kl},$$

ovvero

$$\tilde{x}_{kl} - \sum_{j=l}^{n} \left(\sum_{i=k}^{m} t_{ki}\tilde{x}_{ij} \right) s_{jl} = \tilde{q}_{kl},$$

da cui si conclude

$$\tilde{x}_{kl} - t_{kk}s_{ll}\tilde{x}_{kl} = \tilde{q}_{kl} + \left(\sum_{i=k+1}^{m} t_{ki}\tilde{x}_{ij} \right) s_{ll} + \sum_{j=l+1}^{n} \left(\sum_{i=k}^{m} t_{ki}\tilde{x}_{ij} \right) s_{jl},$$

per $k = m, \ldots, 1$ e $l = n, \ldots 1$.

Il costo computazionale per l'algoritmo di Bertels e Stewart per le equazioni di Stein è analogo a quello delle equazioni di Sylvester e dunque, nel caso $n = m$, di circa $60n^3$ operazioni elementari comprensivo del calcolo delle due fattorizzazioni di Schur delle matrici A e B^T.

Si consideri l'equazione di Stein simmetrica

$$A - AXA^* = Q, \tag{2.10}$$

dove X, A, $Q \in \mathbb{C}^{n\times n}$ e la matrice Q è hermitiana. Applicando quanto sopra esposto, è possibile osservare che l'equazione (2.10) ammette una e una sola soluzione se e solo se lo spettro di A non ammette coppie di autovalori simmetrici rispetto alla circonferenza unitaria S^1, ovvero non esistano autovalori λ e μ di A tali che

$$\lambda = \frac{\mu}{|\mu|^2}.$$

L'algoritmo di Bartels e Stewart per le equazioni di Stein simmetriche ricalca i passi soliti, con il vantaggio, ovviamente, di calcolare una sola fattorizzazione di Schur.

2.2 Metodo di Schur

Nella sezione 1.4.1 si è stabilito il legame tra sottospazi invarianti e soluzioni di $\mathcal{NARE}$ e $\mathcal{CARE}$, e tra sottospazi di deflazione e soluzioni di $\mathcal{DARE}$, $\mathcal{GCARE}$ e $\mathcal{GDARE}$. Individuare tali sottospazi permette, dunque, di determinare le soluzioni delle equazioni di Riccati: il metodo di Schur segue esattamente questa strategia per risolvere le $\mathcal{ARE}$. Sfruttando, inoltre, le proprietà delle soluzioni estremali, illustrate nella sezione 1.4.2, è possibile individuare i suddetti sottospazi mediante proprietà di c-stabilità o d-stabilità di particolari matrici.

La prima versione del metodo di Schur è stata proposta nel 1979 da Alan J. Laub in [40], a base di tale metodo vi è un efficace utilizzo della fattorizzazione reale di Schur per la determinazione di sottospazi invarianti e della fattorizzazione generalizzata di Schur per il calcolo di sottospazi di deflazione.

2.2.1 Metodo di Schur per $\mathcal{NARE}$

Si consideri la $\mathcal{NARE}$

$$C + XA + DX - XBX = 0 \tag{2.11}$$

e sia $\mathcal{H}$ la relativa matrice Hamiltoniana. Per quanto descritto nel teorema 1.4.1, indicati con $\mathcal{V}$ un graph subspace n-dimensionale $\mathcal{H}$-invariante e con $V := \binom{Y}{Z}$ la matrice le cui colonne sono generatori di $\mathcal{V}$, è possibile ottenere una soluzione X della *nare* (2.11), ponendo $X := ZY^{-1}$.

Si supponga ora che la matrice dei Coefficienti $\mathcal{M}$ associata alla (2.11) sia una M-matrice non singolare o singolare irriducibile, allora, per quanto esposto nella sezione 1.4.3, esiste la soluzione minimale $X_{\min}$ ed il graph subspace n-dimensionale $\mathcal{H}$-invariante corrispondente, è relativo agli n autovalori di $\mathcal{H}$ con parte reale massima, in particolare al sottospazio $\mathcal{H}$-invairante n-dimensionale c-antistabile o debolemte c-antistabile.

Si osservi che per le argomentazioni del teorema 1.4.13, se la matrice $\mathcal{M}$ è non singolare o singolare irriducibile con drift μ non nullo, tale sottospazio è univocamente determinato. Se $\mathcal{M}$ è singolare irriducibile con drift nullo, è possibile dimostrare che esiste uno e uno solo sottospazio $\mathcal{H}$-invariante c-antistabile **canonico**, ovvero un sottospazio generato dagli autovettori relativi ad autovalori c-antistabili e dall'unico autovettore relativo a $\lambda = 0$ (c.f.r teorema 1.4.12).

Si conclude, quindi, che, se la matrice dei Coefficienti $\mathcal{M}$ associata alla (2.11) è una M-matrice non singolare o singolare irriducibile, allora, è sempre possibile individuare univocamente il graph subspace n-dimensionale $\mathcal{H}$-invariante (debolmente) c-antistabile, in particolare è sufficiente determinare il graph subspace n-dimensionale $\mathcal{H}$-invariante relativo agli n autovalori di $\mathcal{H}$ con parte reale massima, in quanto, per il teorema 1.4.11, $\mathcal{H}$ presenta un (m, n) c-splitting.

Sia quindi $\mathcal{H} = UTU^T$ la fattorizzazione reale di Schur della matrice $\mathcal{H}$ con U matrice ortogonale e T matrice triangolare superiore a blocchi. Si supponga, inoltre, che le matrici U e T siano partizionate come segue:

$$U := \begin{pmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{pmatrix} \qquad T := \begin{pmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{pmatrix},$$

dove i blocchi di testa sono matrici $n \times n$ e la matrice T_{11} ha come autovalori gli n autovalori di $\mathcal{H}$ con parte reale massima. È possibile effettuare tale costruzione applicando una versione semiordinata dell'algoritmo della fattorizzazione reale di Schur, imponendo che gli n autovalori con parte reale massima, non necessariamente ordinati, siano posti nella sottomatrice di testa T_{11}.

Dalla decomposizione, si ottiene immediatamente

$$\mathcal{H} \begin{pmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{pmatrix} = \begin{pmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{pmatrix} \begin{pmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{pmatrix},$$

e quindi, considerando la prima colonna a blocchi

$$\mathcal{H}\begin{pmatrix} U_{11} \\ U_{21} \end{pmatrix} = \begin{pmatrix} U_{11} \\ U_{21} \end{pmatrix} T_{11}.$$

Si evince, dunque, che il sottospazio $\mathcal{V}$ generato dalle prime n colonne della matrice U è il graph subspace $\mathcal{H}$-invariante n-dimensionale realativo agli n autovalori di $\mathcal{H}$ con parte reale massima. Risulta, pertanto, $X_{\min} := U_{21}U_{11}^{-1}$.

Il metodo di Schur, sebbene concettualmente molto immediato, non risulta particolarmente efficiente per la risoluzione di $\mathcal{NARE}$, in quanto dal punto di vista numerico presenta le seguenti caratteristiche:

costo computazionale: nel caso $n = m$ l'algoritmo richiede circa $200n^3$ operazioni elementari, dovuto in gran parte alla fattorizzazione di Schur della matrice $\mathcal{H}$ e all'inversione della matrice U_{11} ([26]),

stabilità e condizionamento: se il drift della $\mathcal{NARE}$ è vicino a zero, il sottospazio c-antistabile è mal condizionato in quanto $\mathcal{H}$ ha due autovettori linearmente indipendenti molto vicini tra loro. Una strategia per superare tale difficoltà, è stata elaborata da Guo in [26].

2.2.2 Metodo di Schur per $\mathcal{CARE}$

Si consideri la $\mathcal{CARE}$:

$$C + XA + A^*X - XBX = 0, \tag{2.12}$$

dove tutte le matrici sono in $\mathbb{C}^{n\times n}$ e le matrici B e C sono hermitiane.

Si supponga che la (2.12) verifichi le ipotesi del teorema 1.4.6 e del teorema 1.4.7, allora esistono le soluzioni estremali X_- e X_+ della $\mathcal{CARE}$ e tali soluzioni sono sono rispettivamente quasi c-stabilizzante e quasi c-antistabilizzante, ovvero, indicando con $\sigma(M)$ lo spettro della matrice M X_+ e X_- verificano:

$$\sigma(A - BX_+) \subseteq \mathbb{C}_{l,0} \qquad \sigma(A - BX_-) \subseteq \mathbb{C}_{r,0}.$$

Sia $\mathcal{H}$ la matrice Hamiltoniana relativa alla $\mathcal{CARE}$ (2.12), allora, per le ben note relazioni

$$\mathcal{H}\begin{pmatrix} I_n \\ X_+ \end{pmatrix} = \begin{pmatrix} I_n \\ X_+ \end{pmatrix}(A - BX_+) \qquad \mathcal{H}\begin{pmatrix} I_n \\ X_- \end{pmatrix} = \begin{pmatrix} I_n \\ X_- \end{pmatrix}(A - BX_-),$$

e poiché $\mathcal{H}$ ha un (n, n) c-splitting, si ha che

- il graph subspace n-dimensionale $\mathcal{H}$-invairante corrispondente a X_+ è (debolmente) c-stabile,
- il graph subspace n-dimensionale $\mathcal{H}$-invairante corrispondente a X_- è (debolmente) c-antistabile.

Utilizzanodo la fattorizzazione reale di Schur, è possible porre

$$\mathcal{H} = UTU^T \qquad \mathcal{H} = VSV^T,$$

dove

$$U := \begin{pmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{pmatrix} \qquad V := \begin{pmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{pmatrix}$$

sono matrici ortogonali, e

$$T := \begin{pmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{pmatrix} \qquad S := \begin{pmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{pmatrix}$$

sono matrici triangolari superiori a blocchi, con T_{11} c-stabile e S_{11} c-antistabile.

Ripercorrendo i medesimi passi svolti nella sezione precedente si ottiene dunque

$$X_+ = U_{21}U_{11}^{-1} \qquad X_- = V_{21}V_{11}^{-1}.$$

Le caratteristiche numeriche del metodo di Schur per le $\mathcal{CARE}$ sono, ovviamente, analoghe a quelle del metodo di Schur per le $\mathcal{NARE}$.

Per risolvere le $\mathcal{GDARE}$ si utilizza una strategia leggermente differente in quanto non si deve individuare un sottospazio invariante, bensì un sottospazio di deflazione. Si consideri, dunque la $\mathcal{GDARE}$

$$C + E^*XA + A^*XE - E^*XBXE = 0, \tag{2.13}$$

dove $E \in \mathbb{C}^{n\times n}$ è una matrice invertibile. Siano

$$L := \begin{pmatrix} A & -B \\ -C & -A^* \end{pmatrix} \qquad E := \begin{pmatrix} E & 0 \\ 0 & E* \end{pmatrix},$$

e sia $\mathcal{P}(z) := L - zK$. Sia inoltre $V := \left(\begin{smallmatrix} Y \\ Z \end{smallmatrix}\right)$ la matrice le cui colonne sono i generatori di un graph subspace n-dimensionale relativo alla matrix pencil $\mathcal{P}(z)$, allora, per il teorema 1.4.3, una soluzione X della $\mathcal{GCARE}$ (2.13), è data da $X := Z(EY)^{-1}$.

Si consideri la **fattorizzazione generalizzata di Schur** delle matrici L e K

$$L = USV^T \qquad K = UTV^T, \tag{2.14}$$

con U e V matrici ortogonali e S e T matrici triangolari superiori a blocchi.

Sia data la seguente partizione della matrici S e T:

$$T := \begin{pmatrix} T_{11} & T_{12} \\ 0 & T_{22} \end{pmatrix} \qquad S := \begin{pmatrix} S_{11} & S_{12} \\ 0 & S_{22} \end{pmatrix}$$

dove tutte le sottomatrici hanno dimensione $n \times n$.

Si osservi, quindi, che valgono le seguenti relazioni

$$USV^T \begin{pmatrix} V_{11} \\ V_{21} \end{pmatrix} = US \begin{pmatrix} I_n \\ 0 \end{pmatrix} = U \begin{pmatrix} S_{11} \\ 0 \end{pmatrix} = \begin{pmatrix} U_{11} \\ U_{22} \end{pmatrix} S_{11},$$
$$UTV^T \begin{pmatrix} V_{11} \\ V_{21} \end{pmatrix} = UT \begin{pmatrix} I_n \\ 0 \end{pmatrix} = U \begin{pmatrix} T_{11} \\ 0 \end{pmatrix} = \begin{pmatrix} U_{11} \\ U_{22} \end{pmatrix} T_{11},$$

ovvero le prime n colonne della matrice V generano un sottospazio di deflazione $\mathcal{V}$ della matrix pencil $\mathcal{P}(z)$, e dunque, per quanto sopra esposto, la matrice $X := V_{21}(EV_{11})^{-1}$, risolve la $\mathcal{GCARE}$ (2.13).

2.2.3 Metodo di Schur per $\mathcal{DARE}$

Si consideri la $\mathcal{DARE}$

$$A^*XA + Q - (C + BXA)^*(R + B^*XB)^{-1}(C + BXA) - X = 0, \tag{2.15}$$

e siano

$$L := \begin{pmatrix} A - BR^{-1}C & 0 \\ -Q + C^*R^{-1}C & I_n \end{pmatrix} \qquad K := \begin{pmatrix} I_n & BR^{-1}B^* \\ 0 & (A - BR^{-1}C)^* \end{pmatrix},$$

e sia $\mathcal{P}(z)$ la matrix pencil definita da $\mathcal{P}(z) := L - zK$. Allora, per quanto esposto nel teorema 1.4.4, vi è una corrispondenza tra i graph subspace di deflazione n-dimensionali $\mathcal{V}$ relativi alla matrix pencil $\mathcal{P}(z)$ e le soluzioni X della $\mathcal{DARE}$. In particolare se le colonne della matrice $V = \left(\begin{smallmatrix} Y \\ Z \end{smallmatrix}\right)$ generano un $\mathcal{V}$, allora la soluzione corrispondente della $\mathcal{DARE}$ (2.15) è data da $X := ZY^{-1}$.

Sia

$$L = USV^T \qquad K = UTV^T,$$

la fattorizzazione generalizzata di Schur della matrici L e K, dunque, adoperando i medesimi calcoli e le medesime notazioni del metodo di Schur per le $\mathcal{GCARE}$, si ha che la matrice $X := V_{21}V_{11}^{-1}$ è soluzione della (2.15).

Per la $\mathcal{GDARE}$

$$A^*XA + Q - (C + BXA)^*(R + B^*XB)^{-1}(C + BXA) - E^*XE = 0, \tag{2.16}$$

dove le matrici R, Q sono matrici hermitiane, è sufficente porre

$$L := \begin{pmatrix} A - BR^{-1}C & 0 \\ -Q + C^*R^{-1}C & E^* \end{pmatrix} \qquad K := \begin{pmatrix} E & BR^{-1}B^* \\ 0 & (A - BR^{-1}C)^* \end{pmatrix},$$

e considerare, al solito, e i graph subspace di deflazione n-dimensionali relativi alla matrix pencil $\mathcal{P}(z) := L - zK$. Considerando, quindi, la fattorizzazione generalizzata di Schur delle matrici L e K

$$L = USV^T \qquad K = UTV^T,$$

si ottiene che la matrice $X := V_{21}(EV_{11})^{-1}$ verifica la $\mathcal{GDARE}$ (2.16)

2.3 Metodi di Iterazione Funzionale

La dimostrazione del teorema 1.4.5 per verificare l'esistenza della soluzione minimale non negativa $X_{\min}$ di una $\mathcal{NARE}$, è una 'dimostrazione costruttiva', ovvero, oltre a mostrare teoricamente che la $X_{\min}$ effettivamente esiste, fornisce anche una strategia per costruire tale matrice, propone, quindi, un metodo risolutivo per individuare la soluzione $X_{\min}$.

Il **metodo delle iterazioni funzionali**, si basa proprio sulla filosofia risolutiva della dimostrazione teorema 1.4.5, ovvero sulla risoluzione di successioni definite per ricorrenza che, per argomentazioni analoghe a quelle del teorema 1.4.5, convergono alla soluzione minimale non negativa $X_{\min}$ della $\mathcal{NARE}$.

Si consideri la equazione algebrica di Riccati non simmetrica:

$$C + XA + DX - XBX = 0 \tag{2.17}$$

dove $A \in \mathbb{C}^{n\times n}$, $B \in \mathbb{C}^{n\times m}$, $C \in \mathbb{C}^{m\times n}$, $D \in \mathbb{C}^{m\times m}$, $X \in \mathbb{R}^{m\times n}$ e si supponga che la matrice dei Coefficienti $\mathcal{M}$ associata alla (2.17) sia una M-matrice non singolare o singolare irriducibile.

Si ponga allora

$$A = A_1 - A_2 \qquad D = D_1 - D_2$$

con A_1 e D_1 M-matrici e $A_2, D_2 \geq 0$.

È evidente, dunque, che risolvere la $\mathcal{NARE}$ (2.17) equivale a risolvere l'equazione matriciale:

$$XA_1 + D_1X = -C + XA_2 + D_2X + XBX.$$

A partire dalla precedente equazione matriciale è possibile generare la successione $\{ X_k \}_{k\in\mathbb{N}}$ definita per ricorrenza:

$$\begin{cases} X_0 = 0 \\ X_{k+1}A_1 + D_1X_{k+1} = -C + X_kA_2 + D_2X_k + X_kBX_k, \quad k = 0, 1 \ldots, \end{cases} \tag{2.18}$$

e pertanto, utilizzando la funzione `vec` e le proprietà del prodotto di Kronecker, ottenere il sistema:

$$\begin{cases} X_0 = 0 \\ \texttt{vec}(X_{k+1}) = (A_1^T \otimes I_m + I_n \otimes D_1)^{-1}\texttt{vec}(-C + X_kA_2 + D_2X_k + X_kBX_k) \quad k = 0, 1 \ldots. \end{cases} \tag{2.19}$$

Si osservi che il metodo è applicabile se la matrice $(A_1^T \otimes I_m + I_n \otimes D_1)$ è invertibile, ovvero se, indicati con λ_i per $i = 1, \dots n$ gli autovalori di A_1 e con μ_j per $j = 1, \dots m$ gli autovalori di D_1, si abbia

$$\lambda_i + \mu_j \neq 0, \qquad \text{per } i = 1, \dots n \text{ e } j = 1, \dots m.$$

Se, quindi, la matrice $(A_1^T \otimes I_m + I_n \otimes D_1)$ è invertibile, ad ogni passo k del metodo di iterazione funzionale, per determinare il termine X_{k+1}, è necessario risolvere un sistema lineare di dimensione nm, risoluzione che ha un costo computazionale $O((nm)^3)$ operazioni aritmetiche. Con una scelta oculata delle matrici A_1 e D_1, è possibile abbattere l'oneroso costo dell'algoritmo portandolo, nel caso $m = n$, a $O(n^3)$ operazioni.

In [22], Chun-Hua Guo analizza le seguenti possibili scelte delle matrici A_1 e D_1, valutando, per ciascuna di tali opzioni, il costo computazionale dell'algoritmo di iterazione funzionale.

- Come prima scelta si ponga:

$$a_{ij}^{(1)} := \begin{cases} a_{ij} & \text{se } i = j \\ 0 & \text{altrimenti} \end{cases}$$

e

$$d_{ij}^{(1)} := \begin{cases} d_{ij} & \text{se } i = j \\ 0 & \text{altrimenti} \end{cases}$$

risoluzione sistema In questo caso la matrice $(A_1^T \otimes I_m + I_n \otimes D_1)$ risulta diagonale, dunque, la risoluzione del sistema, per $m = n$, ha un costo computazionale di $O(n^2)$ operazioni.

termine noto Più significativo è invece il costo relativo al calcolo del termine noto $\texttt{vec}(-C + X_k A_2 + D_2 X_k + X_k B X_k)$ in quanto sono necessari 3 prodotti di matrici di dimensione $n \times n$ con un costo di $6n^3$ operazioni aritmetiche.

costo totale ad iterazione Ad ogni passo dell'algoritmo occorrono, quindi, circa $6n^3$ operazioni.

- Per la seconda scelta si ponga:

$$a_{ij}^{(1)} := \begin{cases} a_{ij} & \text{se } i \geq j \\ 0 & \text{altrimenti} \end{cases}$$

e

$$d_{ij}^{(1)} := \begin{cases} d_{ij} & \text{se } i \geq j \\ 0 & \text{altrimenti.} \end{cases}$$

risoluzione sistema La matrice $(A_1^T \otimes I_m + I_n \otimes D_1)$ è triangolare superiore a blocchi con blocchi triangolari superiori, pertanto il costo computazionale per la risoluzione del sistema lineare associato è di $2n^3$ operazioni.

termine noto Il calcolo del termine noto è del tutto analogo al caso precedente e dunque comporta un costo di $6n^3$ operazioni aritmetiche.

costo totale ad iterazione Sono in tutto necessarie $8n^3$ operazioni.

- Come terza scelta si ponga semplicemente

$$A_1 = A \qquad \text{e} \qquad D_1 = D.$$

risoluzione dell'equazione matriciale In questo caso si ha un'impostazione differente dai precedenti casi: al passo k occorre risolvere l'equazione di Sylvester:

$$X_{k+1} A + D X_{k+1} = -C + X_k B X_k.$$

L'algoritmo di Bartels e Stewart per la risoluzione di equazioni di Sylvester (si veda la sezione 2.1.1) ha un costo di $60n^3$ operazioni che però comprende il costo per portare le matrici A e D in forma normale di Schur, operazione eseguita una sola volta durante lo svolgimento dell'algoritmo e quindi 'ammortizzabile'.

termine noto Per il calcolo del termine noto si rendono necessarie due moltiplicazioni di matrici $n \times n$ con un costo di $4n^3$ operazioni.

costo totale per passo Si rendono pertanto necessarie $10n^3$ operazioni.

Si osservi, inoltre, che nei primi due casi, ad ogni iterazione si risolvono sitemi lineari associati ad M-matrici, questo comporta che non vi siano problemi di cancellazione e dunque il metodo di Schur può considerarsi numericamente ben condizionato.

Oltre al costo computazione, un parametro fondamentale per valutare l'efficienza di un metodo iterativo è determinare la velocità di convergenza della successione definita per ricorrenza alla soluzione minimale non negativa $X_{\min}$. In [22] vengono comparati in base a tale parametro, i tre metodi di iterazione funzionale sopra introdotti. L'analisi parte dal seguente teorema di cui si presenta solo l'enunciato:

Teorema 2.3.1. *Data la successione definita per ricorrenza (2.18) ed indicato con $\varrho(\cdot)$ il raggio spettrale di una matrice, vale la seguente relazione:*

$$\limsup \sqrt[k]{\|X_k - X_{\min}\|} = \varrho\big((A_1^T \otimes I_m + I_n \otimes D_1)^{-1}((A_2 + BX_{\min})^T \otimes I_m + I_n \otimes (D_2 + X_{\min}B))\big) \quad (2.20)$$

dove $X_{\min}$ è la soluzione minimale non negativa della $\mathcal{NARE}$ (2.17).

Il teorema 2.3.1 mostra, quindi, che i metodi di iterazione funzionale hanno **ordine di convergenza lineare**, ovvero che la successione $\{ X_k \}_{k\in\mathbb{N}}$ verifica:

$$\lim_{k\to\infty} \frac{\|X_{k+1} - X_{\min}\|}{\|X_k - X_{\min}\|} = \gamma \qquad \text{e dunque} \qquad \|X_k - X_{\min}\| \approx \gamma^k \|X_0 - X_{\min}\|,$$

con

$$\gamma \approx \varrho\big((A_1^T \otimes I_m + I_n \otimes D_1)^{-1}((A_2 + BX_{\min})^T \otimes I_m + I_n \otimes (D_2 + X_{\min}B))\big).$$

Alla luce del risultato appena esposto, si valuta quale scelta di A_1 e D_1 comporta una velocità di convergenza maggiore.

Si ponga quindi:

$$U := A_1^T \otimes I_m + I_n \otimes D_1 \qquad \text{e} \qquad V := (A_2 + X_{\min}B)^T \otimes I_m + I_n \otimes (D_2 + BX_{\min})$$

e

$$W := U - V = (A - X_{\min}B)^T \otimes I_m + I_n \otimes (D - BX_{\min})$$

e si osservi che

- U è una M-matrice in quanto A_1 e D_1 sono M-matrici,
- W è una M-matrice essendo $(A - X_{\min}B)$ e $(D - BX_{\min})$ M-matrici per il teorema 1.4.5,
- $V \geq 0$ poiché A_2, D_2, B e $X_{\min}$ sono matrici non negative.

Fatte le dovute osservazioni, si enuncia il seguente teorema che permette di stabilire quale scelta di A_1 e D_1 dà luogo alla convergenza più rapida.

Teorema 2.3.2. *Sia*

$$W = U_1 - V_1 = U_2 - V_2$$

con W, U_1, U_2 M-matrici non singolari e V_1, $V_2 \geq 0$. Se $U_1 \leq U_2$ allora

$$\varrho(U_1^{-1}V_1) \leq \varrho(U_2^{-1}V_2).$$

Dimostrazione. Le matrici W, U_1, U_2 sono M-matrici non singolari ed hanno pertanto inversa positiva, valgono, dunque, le seguenti relazioni:

$$0 \leq U_i^{-1}V_i = (W + V_i)^{-1}V_i = (I + W^{-1}V_i)^{-1}W^{-1}V_i \qquad \text{per } i = 1, 2.$$

Per ipotesi, inoltre, si ha $V_1 = U_1 - W \leq U_2 - W = V_2$, quindi vale

$$0 \leq W^{-1}V_1 \leq W^{-1}V_2,$$

dunque, per il teorema di Perron-Frobenius, si ha:

$$\varrho(W^{-1}V_1) \leq \varrho(W^{-1}V_2).$$

Si osservi ora che gli autovalori delle matrici $(I + W^{-1}V_i)^{-1}W^{-1}V_i$ per $i = 1, 2$ sono del tipo

$$\frac{\lambda_i}{1 + \lambda_i}$$

dove λ_i è un autovalore di $W^{-1}V_i$. La funzione $x \mapsto \frac{x}{1+x}$ è monotona crescente, pertanto si ottiene:

$$\frac{\varrho(W^{-1}V_1)}{1 + \varrho(W^{-1}V_1)} \leq \frac{\varrho(W^{-1}V_2)}{1 + \varrho(W^{-1}V_2)}$$

ovvero $\varrho(U_1^{-1}V_1) \leq \varrho(U_2^{-1}V_2)$. □

Per quanto dimostrato nel precedente teorema, è evidente che l'iterazione funzionale che garantisce la velocità di convergenza più rapida (pur essendo l'algoritmo con più alto costo computazionale per passo) è quella ottenuta ponendo $A_1 = A$ e $D_1 = D$ che ha dunque velocità di convergenza

$$\gamma \approx \varrho\big((A^T \otimes I_m + I_n \otimes D)^{-1}((BX_{\min})^T \otimes I_m + I_n \otimes (X_{\min}B))\big).$$

È possibile applicare metodi di iterazione funzionale che sfruttino successioni definite per ricorrenza convergenti alle soluzioni estremali anche alle $\mathcal{CARE}$ e alle $\mathcal{DARE}$, per maggiori dettagli si rimanda a [53].

2.4 Metodo di Newton

Il **metodo di Newton** è un metodo iterativo utile per individuare gli zeri di funzioni reali sufficientemente regolari. Tale metodo è stato descritto da Isaac Newton nel 1669 nel *De analysi per aequationes numero terminorum infinitas*, tuttavia l'attuale formulazione dell'algortimo è significativamente diversa dalla formulazione originale, in quanto Newton applicava il suo metodo esclusivamente ai polinomi ed i suoi interessi erano prettamente algebrici piuttosto che analitici. Inoltre, nel metodo di Newton originario viene generata una successione di polinomi, e non di approssimanti x_n della radice α, e solo in un secondo momento viene individuata un'approssimazione della radice. Una versione più vicina all'attuale del metodo di Newton, è stata proposta nel 1690 da Joseph Raphson in *Analysis aequationum universalis*: Raphson applica nuovamente l'algoritmo esclusivamente ai polinomi, ma contrariamente a quanto fatto da Newton, il suo metodo genera direttamente una successione di approssimanti x_n della radice α. Pertanto il metodo di Newton è noto anche come **metodo di Newton-Raphson**.

Si consideri una funzione differenziabile $f\colon \Omega \to \mathbb{R}^n$, con $\Omega \subseteq \mathbb{R}^n$, e sia $\alpha \in \Omega$ tale che $f(\alpha) = 0$. Il metodo di Newton genera la seguente successione definita per ricorrenza:

$$\begin{cases} x_0 \in \Omega_1 \\ x_{k+1} = x_k - \mathcal{J}_f(x_k)^{-1} f(x_k), \end{cases} \tag{2.21}$$

dove $\Omega_1 \subseteq \Omega$ è un intorno relativamente piccolo dello zero α e $\mathcal{J}_f(x_k)$ indica il valore dello jacobiano di f valutato in x_k.

Il teorema che segue fornisce delle ipotesi che assicurino la convergenza della successione (2.21) e ne stabiliscano l'ordine di convergenza ([7]):

Teorema 2.4.1. *Sia $f\colon \Omega \to \mathbb{R}^n$, con $\Omega \subseteq \mathbb{R}^n$, e sia $\alpha \in \Omega$ tale che $f(\alpha) = 0$. Se $\mathcal{J}_f(x)$ è invertibile in Ω, allora esiste un intorno $\Omega_1 \subseteq \Omega$ di α tale che, se $x_0 \in \Omega_1$, la successione $\{ x_k \}_{k\in\mathbb{N}}$ definita in (2.21) verifica le seguenti relazioni:*

$$lim_{k\to\infty} x_k = \alpha$$
$$\|x_{k+1} - \alpha\| \leq \beta \|x_k - \alpha\|^2$$

per $k = 0, 1, \ldots,$ e per una opportuna costante β.

2.4.1 Derivata di Fréchet e operatore di Riccati

È possibile estendere la formulazione del metodo di Newton data per sottoinsiemi Ω di $\mathbb{R}^n$ a qualsiasi spazio di Banach $\mathcal{V}$. Sia, dunque, $\mathcal{F}\colon \mathcal{V} \to \mathcal{V}$ un operatore tra spazi di Banach e si indichi con $\mathcal{A}ut(V)$ lo spazio vettoriale degli automorfismi di $\mathcal{V}$, ovvero

$$\mathcal{A}ut(V) := \{ F\colon \mathcal{V} \to \mathcal{V} \mid F \text{ operatore lineare e continuo} \}.$$

La **derivata di Fréchet** di $\mathcal{F}$ è una applicazione

$$\mathcal{F}'\colon \mathcal{V} \to \mathcal{A}ut(\mathcal{V}) \qquad x \mapsto \mathcal{F}'_x[\cdot]$$

tale che per ogni $x \in \mathcal{V}$ l'applicazione $\mathcal{F}'_x[\cdot]$ verifica

$$lim_{h\to 0} \frac{\|\mathcal{F}(x+h) - \mathcal{F}(x) - \mathcal{F}'_x[h]\|}{\|h\|} = 0.$$

Un operatore $\mathcal{F}$ si dice **derivabile secondo Fréchet** se ammette derivata di Fréchet.

Alla luce delle suddette definizioniè possibile definire il metodo di Newton per un qualsiasi spazio di Banach $\mathcal{V}$: sia $\mathcal{F}\colon \mathcal{V} \to \mathcal{V}$ un operatore derivabile secondo Fréchet e sia $\alpha \in \mathcal{V}$ tale che $\mathcal{F}(\alpha) = 0$. Il metodo di Newton genera la successione definita per ricorrenza

$$\begin{cases} x_0 \in \mathcal{V}_1 \\ x_{k+1} = x_k - (\mathcal{F}'_{x_k})^{-1}[\mathcal{F}(x_k)], \end{cases} \tag{2.22}$$

dove $\mathcal{V}_1 \subseteq \mathcal{V}$ è un intorno dello zero α.

È tuttavia consigliabile in ciascuna iterazione del metodo non invertire la derivata di Fréchet, pertanto nell'implementazione è preferibile risolvere il seguente sistema lineare in luogo della (2.22)

$$\begin{cases} \mathcal{F}'_{x_k}[h_k] = -\mathcal{F}(x_k), \\ x_{k+1} = x_k + h_k. \end{cases} \tag{2.23}$$

Si consideri la $\mathcal{NARE}$

$$C + XA + DX - XBX = 0, \tag{2.24}$$

si dice **operatore di Riccati** della $\mathcal{NARE}$ (2.24) l'operatore $\mathcal{R}\colon \mathbb{C}^{m\times n} \to \mathbb{C}^{m\times n}$ definito da

$$\mathcal{R}(X) := C + XA + DX - XBX. \tag{2.25}$$

Si osservi che l'operatore di Riccati (2.25) è derivabile secondo Fréchet ed ammette derivata di Fréchet data da

$$\mathcal{R}'_X[H] := HA + DH - HBX - XBH, \tag{2.26}$$

infatti

$$\mathcal{R}(X+H) - \mathcal{R}(X) - \mathcal{R}'_H[H] = HBH,$$

e dunque

$$lim_{H\to 0} \frac{\|\mathcal{R}(X+H) - \mathcal{R}(X) - \mathcal{R}'_X[H]\|}{\|H\|} = lim_{H\to 0} \frac{\|HBH\|}{\|H\|} = 0.$$

Utilizzando le proprietà della funzione vec e del prodotto di Kronecker, è possibile riscrivere l'azione della derivata di Fréchet $\mathcal{R}'_X[\cdot]$:

$$\texttt{vec}\big(\mathcal{R}'_X[H]\big) = \texttt{vec}(HA + DH - HBX - XBH) = \Delta_X \texttt{vec}(H)$$

con

$$\Delta_X := (A - BX)^T \otimes I_m + I_n \otimes (D - XB). \tag{2.27}$$

Una soluzione S della $\mathcal{NARE}$ (2.24) si dice **critica** se la matrice Δ_S non è invertibile.

Si osservi che per le proprietà del prodotto di Kronecker, gli autovalori della matrice Δ_X sono della forma $\lambda_i + \mu_j$ dove λ_i per $i = 1, \dots n$ sono autovalori di $(A - BX)$ e μ_j per $j = 1, \dots, m$ sono autovalori di $(D - XB)$. Una soluzione X risulta quindi critica se e solo se le matrici $(A - BX)$ e $(XB - D)$ hanno un autovalore comune. D'altro canto, per quanto esposto nella Sezione 1.4.1, gli autovalori della Matrice Hamiltoniana verificano la relazione

$$\sigma(\mathcal{H}) = \sigma(A - BX) \cup \sigma(XB - D),$$

da cui il particolare legame tra soluzioni critiche e proprietà spettrali della matrice $\mathcal{H}$.

Se si suppone, inoltre, che la Matrice dei Coefficienti $\mathcal{M}$ sia una M-matrice non singolare o singolare irriducibile, allora, per il Teorema 1.4.5, le matrici $A - BX_{\min}$ e $D - X_{\min}B$ sono M-matrici e dunque, la matrice $\Delta_{X_{\min}}$ risulta una M-matrice. Gli autovalori di $\Delta_{X_{\min}}$ hanno, pertanto, parte reale positiva, si conclude quindi che la soluzione minimale $X_{\min}$ è critica se e solo se $A - BX_{\min}$ e $D - X_{\min}B$ sono non invertibili. Per quanto argomentato nella Sezione 1.4.3, tale situazione si verifica se e solo se $\lambda_m = \lambda_{m+1} = 0$ dove λ_i con $i = n + m \dots, 1$ sono gli autovalori di $\mathcal{M}$, ovvero se e solo se la $\mathcal{NARE}$ ha drift nullo. Si conclude, quindi, che la soluzione $X_{\min}$ è critica se e solo la $\mathcal{NARE}$ (2.24) è ricorrente nulla.

Si consideri ora la $\mathcal{CARE}$

$$C + XA + A^*X - XBX = 0, \tag{2.28}$$

allora in questo caso la derivata di Fréchet dell'operatore di Riccati assume la forma

$$\mathcal{R}'_X[H] := HA + A^*H - HBX - XBH, \tag{2.29}$$

e dunque, utilizzando la funzione vec, si ha

$$\texttt{vec}(\mathcal{R}'_X[H]) = \Delta_X \texttt{vec}(H)$$

con

$$\Delta_X := (A - BX)^T \otimes I_m + I_n \otimes (A^* - XB). \tag{2.30}$$

Anche per le $\mathcal{CARE}$, una soluzione S della (2.28) si dice **critica** se la matrice Δ_S è singolare.

Per le proprietà del prodotto di Kronecker, lo spettro della matrice Δ_X è composto da elementi del tipo $\lambda + \mu$ con λ autovalore di $(A - BX)$ e μ autovalore di $(A^* - XB)$. Dunque una soluzione X della $\mathcal{CARE}$ (2.28) è critica se e solo se le matrici $(A - BX)$ e $(XB - A^*)$ hanno un autovalore comune.

Come nel caso precedente, è opportuno ricordare che, se X è soluzione di (2.28), allora la Matrice Hamiltoniana verifica.

$$\sigma(\mathcal{H}) = \sigma(A - BX) \cup \sigma(XB - A^*),$$

e che $\mathcal{H}$ ha un (n, n) c-splitting. Pertanto, se si suppone che $\mathcal{H}$ non abbia autovalori immaginari puri allora ogni soluzione X c-stabilizzabile o c-antistabilizzabile non può essere critica.

Infine, si consideri la $\mathcal{DARE}$

$$A^*XA + Q - (C + B^*XA)^*(R + B^*XB)^{-1}(C + B^*XA) - X = 0, \tag{2.31}$$

ed il relativo operatore di riaccati

$$\mathcal{R}(X) := A^*XA + Q - (C + B^*XA)^*(R + B^*XB)^{-1}(C + B^*XA) - X. \tag{2.32}$$

È possibile dimostrare che tale operatore è derivabile secondo Fréchet e che ammette derivata di Fréchet

$$\mathcal{R}'_X[H] := \hat{A}^*H\hat{A} - H, \tag{2.33}$$

con $\hat{A} = B(R+B^*XB)^{-1}(C+B^*XA)$. Al solito, utilizzando la funzione $\mathtt{vec}$, è possibile riformulare la derivata di Fréchet $\mathcal{R}'_X$ come

$$\mathtt{vec}\big(\mathcal{R}'_X[H]\big) = \Delta_X \mathtt{vec}(H)$$

con

$$\Delta_X := \hat{A}^T \otimes \hat{A}^* - I_{n^2}. \tag{2.34}$$

Una soluzione S della $\mathcal{DARE}$ (2.31) si dirà **critica** se la matrice Δ_S è singolare.

Per le solite proprietà del prodotto di Kronecker, gli autovalori della matrice Δ_X hanno la forma $\lambda\bar{\mu} - 1$ con λ e μ autovalori di $\hat{A}$. Se, per esempio, la matrice $\hat{A}$ è d-stabile o d-antistabile, ogni soluzione X della $\mathcal{DARE}$ risulta non critica.

2.4.2 Metodo di Newton per $\mathcal{NARE}$

Si consideri la $\mathcal{NARE}$ (2.24), e si supponga che la Matrice dei Coefficienti $\mathcal{M}$ ad essa relativa sia una M-matrice non singolare o singolare irriducibile. Il **metodo di Newton per $\mathcal{NARE}$** genera la seguente successione definita per ricorrenza

$$\begin{cases} X_0 \in \mathcal{V}_1 \\ X_{k+1} = X_k - (\mathcal{R}'_{X_k})^{-1}[C + X_kA + DX_k - X_kBX_k], \end{cases} \tag{2.35}$$

dove $\mathcal{V}_1$ è un'opportuna regione di $\mathbb{C}^{m\times n}$ e $\mathcal{R}'_X$ è la derivata di Fréchet dell'operatore di Riccati definita in (2.26). La precedente formulazione equivale a

$$\begin{aligned} \mathcal{R}'_{X_k}[X_{k+1} - X_k] &= (X_{k+1} - X_k)A + D(X_{k+1} - X_k) - (X_{k+1} - X_k)BX_k - X_kB(X_{k+1} - X_k) \\ &= -C - X_kA - DX_k + X_kBX_k, \end{aligned}$$

da cui

$$X_{k+1}A + DX_{k+1} - X_{k+1}BX_k - X_kBX_{k+1} = -C - X_kBX_k, \tag{2.36}$$

e quindi, con le notazioni di (2.27)

$$\Delta_{X_k}\mathtt{vec}(X_{k+1}) = \mathtt{vec}(-C - X_kBX_k). \tag{2.37}$$

Alla luce delle precedenti argomentazioni si enuncia il teorema dovuto a Guo e Laub ([29]) che dimostra la convergenza del metodo di Newton alla soluzione minimale non negativa $X_{\min}$, cui si premette il seguente lemma dovuto ai medesimi autori.

Lemma 2.4.1. *Si supponga che la $\mathcal{NARE}$ (2.24) abbia Matrice dei Coefficienti $\mathcal{M}$ M-matrice irriducibile. Allora la soluzione minimale non negativa $X_{\min}$ è strettamente positiva.*

Teorema 2.4.2. *Si consideri la $\mathcal{NARE}$ (2.24) e si supponga che la rispettiva Matrice dei Coefficienti $\mathcal{M}$ sia una M-matrice non singolare o una M-matrice singolare irriducibile e sia $X_{\min}$ la soluzione minimale non negativa.*

La succesione $\{X_k\}_{k\in\mathbb{N}}$ definita dal metodo di Newton con valore iniziale $X_0 = 0$ rispetta le seguenti proprietà:

- *la successione $\{X_k\}_{k\in\mathbb{N}}$ è ben definita,*
- *$X_k \le X_{k+1} \le X_{\min}$ per ogni $k \in \mathbb{N}$,*
- *$\lim_{k\to+\infty} X_k = X_{\min}$.*

Dimostrazione. Si consideri dapprima il caso in cui $\mathcal{M}$ sia una M-matrice non singolare. La dimostrazione si svuilluppa in tre parti, la cui verifica è provata per induzione:

a) $X_k \leq X_{\min}$,

b) Δ_{X_k} è una M-matrice non singolare,

c) $X_k \leq X_{k+1}$.

Per $k = 0$ si ha:

a) il primo termine della successione definita per ricorrenza è $X_0 = 0$ pertanto $X_0 \leq X_{\min}$;

b) le matrici A e D sono M-matrici non singolari, pertanto $\Delta_{X_0} = (A^T \otimes I_m + I_n \otimes D)$ risulta una M-matrice non singolare;

c) per la (2.36) il secondo termine della successione è definito da:

$$X_1 A + B X_1 = -C,$$

dunque

$$(A^T \otimes I_m + I_n \otimes D)\mathtt{vec}(X_1) = \mathtt{vec}(-C).$$

Si conclude, quindi

$$\mathtt{vec}(X_1) = (\Delta_{X_0})^{-1}\mathtt{vec}(-C) \geq 0 = X_0.$$

essendo $C \leq 0$ e $(\Delta_{X_0})^{-1} \geq 0$ per le ben note proprietà delle M-matrici.

Si supponga che le proprietà sopra indicate siano verificate per ogni $i \leq k$, allora

a) Dalla (2.36)) seguono le uguaglianze:

$$\begin{gathered}(X_{k+1} - X_{\min})(A - BX_k) + (D - X_K B)(X_{k+1} - X_{\min}) = \\ -C - X_k B X_k - X_{\min} A + X_{\min} B X_k - D X_{\min} + X_k B X_{\min} = \\ -C - X_k B X_k + C - X_{\min} B X_{\min} + X_{\min} B X_k + X_k B X_{\min} = \\ -(X_k - X_{\min}) B (X_k - X_{\min}).\end{gathered}$$

Utilizzando la funzione vec per primo e ultimo membro si ottiene:

$$\Delta_{X_k}\mathtt{vec}(X_{k+1} - X_{\min}) = -\mathtt{vec}\big((X_k - X_{\min})B(X_k - X_{\min})\big).$$

Per ipotesi induttiva si ha che $X_k \leq X_{\min}$ e che Δ_{X_k} è una M-matrice non singolare, pertanto:

$$\mathtt{vec}(X_{k+1} - X_{\min}) = (\Delta_{X_k})^{-1}\big(-\mathtt{vec}\big((X_k - X_{\min})B(X_k - X_{\min})\big)\big) \leq 0$$

da cui segue la tesi.

b) Per il punto precedente $X_{k+1} \leq X_{\min}$, dunque:

$$A - BX_{k+1} \geq A - BX_{\min} \qquad \text{e} \qquad D - X_{k+1}B \geq D - X_{\min}B$$

e allora:

$$\big((A - BX_{k+1})^T \otimes I_m + I_n \otimes (D - X_{k+1}B)\big) \geq \big((A - BX_{\min})^T \otimes I_m + I_n \otimes (D - X_{\min}B)\big).$$

Le matrici $\Delta_{X_{k+1}}$ e $\Delta_{X_{\min}}$ sono rispettivamente una Z-matrice ed una M-matrice per costruzione, pertanto la disuguaglianza precedente implica che $\Delta_{X_{k+1}}$ è una M-matrice, da cui la tesi.

c) È immediato verificare la seguente catena di uguaglianze uguaglianze:

$$\begin{gathered} X_{k+1}(A - BX_{k+1}) + (D - X_{k+1}B)X_{k+1} = \\ X_{k+1}(A - BX_k - B(X_{k+1} - X_k)) + (D - X_kB - (X_{k+1} - X_k)B)X_{k+1} = \\ -C - X_kBX_k - X_{k+1}B(X_{k+1} - X_k) - (X_{k+1} - X_k)BX_{k+1} = \\ -C - X_{k+1}BX_{k+1} - (X_{k+1} - X_k)B(X_{k+1} - X_k). \end{gathered}$$

Dalle eguaglianze precedenti segue che:

$$\begin{gathered} (X_{k+1} - X_{k+2})(A - BX_{k+1}) + (D - X_{k+1}B)(X_{k+1} - X_{k+2}) = \\ -C - X_{k+1}BX_{k+1} - (X_{k+1} - X_k)B(X_{k+1} - X_k) + C + X_{k+1}BX_{k+1} = \\ -(X_{k+1} - X_k)B(X_{k+1} - X_k), \end{gathered}$$

e dunque, essendo per il punto precedente $\Delta_{X_{k+1}}$ una M-matrice non singolare e per ipotesi induttiva $X_k \leq X_{k+1}$, si ottiene:

$$X_{k+1} \leq X_{k+2}.$$

Per quanto riguarda il caso in cui la Matrice dei Coefficienti $\mathcal{M}$ sia una M-matrice singolare irriducibile la dimostrazione è analoga: si prova per induzione su k che

a) $X_k < X_{\min}$,

b) Δ_{X_k} è una M-matrice non singolare,

c) $X_k \leq X_{k+1}$.

Per $k = 0$ si ha

a) è evidente essendo $X_0 = 0 < X_{\min}$ per il Lemma 2.4.1;

b) le matrici A e D sono M-matrici non singolari, pertanto:

$$\Delta_{X_0} = (A^T \otimes I_m + I_n \otimes D)$$

è una M-matrice non singolare;

c) la dimostrzione è analoga al caso in cui $\mathcal{M}$ sia una M-matrice non singolare.

Si supponga di aver dimostrato le tre proprietà per ogni $i \leq k$ e le si verificano per $k + 1$.

a) È possibile rifarsi al caso precedente sostituendo la maggiorazione (minorazione) stretta alla maggiorazione (minorazione) semplice.

b) Si osservi che vale la seguente serie di uguaglianze:

$$\begin{gathered} (X_{\min} - X_{k+1})(A - BX_{k+1}) + (D - X_{k+1}B)(X_{\min} - X_{k+1}) = \\ C + X_{k+1}BX_{k+1} + (X_{k+1} - X_k)B(X_{k+1} - X_k) + \\ -C + X_{\min}BX_{\min} - X_{\min}BX_{k+1} - X_{k+1}BX_{\min} = \\ (X_{k+1} - X_k)B(X_{k+1} - X_k) + (X_{k+1} - X_{\min})B(X_{k+1} - X_{\min}). \end{gathered}$$

Essendo $X_{\min} - X_{k+1} > 0$ per il punto precedente, e, $X_{k+1} \geq X_k$ per ipotesi induttiva, risulta:

$$\Delta_{X_{k+1}} \mathtt{vec}(X_{\min} - X_{k+1}) > 0$$

e dunque, per le caratterizzazioni delle M-matrici, $\Delta_{X_{k+1}}$ è una M-matrice non singolare.

c) Analogo al caso di $\mathcal{M}$ non singolare.

In entrambi i casi le proprietà sono quindi verificate per ogni $k = 0, 1, \ldots$ pertanto la successione $\{X_k\}_{k\in\mathbb{N}}$ è ben definita e monotona crescente e limitata, risulta, pertanto, convergente. Si supponga

$$\lim_{k\to+\infty} X_k = X_*$$

allora X_* è una soluzione non negativa della $\mathcal{NARE}$ (2.24) e poiché si avrebbe $X_* \leq X_{\min}$, e $X_{\min}$ è la soluzione minimale si ottiene:

$$\lim_{k\to+\infty} X_k = X_{\min}.$$

Dunque il metodo di Newton con valore iniziale $X_0 = 0$ è convergente. □

Per quanto riguarda l'analisi sull'ordine di convergenza del metodo di Newton, vale il seguente risultato ([29])

Teorema 2.4.3. *Sia $\mathcal{M}$ la Matrice dei Coefficienti associata alla $\mathcal{NARE}$ (2.24) e sia $\{X_k\}_{k\in\mathbb{N}}$ la successione generata dal metodo di Newton. Allora*

- *se $\mathcal{M}$ è una M-matrice non singolare o singolare irriducibile con drift non nullo il metodo di Newton ha convergenza quadratica, ovvero esiste una costante γ tale che*

$$\|X_{k+1} - X_{\min}\| \approx \gamma\|X_k - X_{\min}\|^2,$$

- *se $\mathcal{M}$ è una M-matrice irriducibile con drift nullo e quindi se la soluzione minimale $X_{\min}$ è critica il metodo di Newton ha convergenza lineare con raggio $\frac{1}{2}$, ovvero*

$$\|X_{k+1} - X_{\min}\| \approx \frac{1}{2}\|X_k - X_{\min}\|.$$

È possibile applicare tecniche particolari nel caso in cui $X_{\min}$ sia critica in modo da ottenere una convergenza quadratica. Una prima strategia è quella si 'shiftare' la Matrice Hamiltoniana $\mathcal{H}$ in una matrice $\tilde{\mathcal{H}}$ che abbia medesimo autospazio invariante c-antistabile e tale che la dericata di Fréchet del nuovo operatore di Riccati $\mathcal{R}'_X$ sia non singolare in $X_{\min}$. Un'altra tecnica è quella di individuare un valore opportuno di X_0 in modo da baipassare la singolarità della matrice $\Delta_{X_{\min}}$. Per un approfondimento di tali artifici si rimanda a [10].

Riassumendo gli aspetti numerici del metodo si ha:

costo computazionale per passo per individuare X_{k+1} è necessario risolvere il sistema (si veda (2.23))

$$\begin{cases} H_k(A - BX_k) + (D - X_kB)H_K = \mathcal{R}(X_k) \\ X_{k+1} = X_k + H_k. \end{cases}$$

Si osservi che la prima relazione è una equazione di Sylvester, la cui risoluzione richiede circa $60n^3$ operazioni con l'algoritmo di Bartels e Stewart. Il calcolo del termine noto e dei coefficienti che definiscono l'equazione è di circa $6n^3$ operazioni. Il costo totale è dunque di circa $66n^3$ operazioni aritmetiche. Si osservi, inoltre, che il calcolo di $\mathcal{R}(X_k)$ fornisce, ad ogni passo, una stima della bontà della approssimazione.

convergenza Il metodo presenta convergenza quadratica nel caso non critico, e convergenza lineare (con raggio $\frac{1}{2}$) nel caso critico pertanto in tal caso si rendono necessarie tecniche per mantenere la convergenza quadratica.

stabilità numerica La stabilità del metodo dipende dal condizionamento della matrice $\mathcal{R}'_{X_+}$.

2.4.3 Metodo di Newton per $\mathcal{CARE}$

Si consideri la $\mathcal{CARE}$ (2.28) con operatore di Riccati

$$\mathcal{R}(X) := C + XA + DX - XBX \tag{2.38}$$

e derivata di Fréchet

$$\mathcal{R}'_X[X] := HA - A^*H - HBX - XBH. \tag{2.39}$$

Alla luce delle suddette definizioni, il termine k-esimo della successione $\{ X_k \}_{k\in\mathbb{N}}$ generata dal **metodo di Newton per $\mathcal{CARE}$** (si veda (2.23)) è dato da

$$\begin{cases} (A^* - X_kB)H_k + H_k(A - BX_k) = -\mathcal{R}(X) \\ X_{k+1} = X_k + H_k. \end{cases} \tag{2.40}$$

Risoltati significativi sulla convergenza della successione $\{ X_k \}_{k\in\mathbb{N}}$ alla soluzione estremale X_+ sono stati provatati da P. Lancaster e R. Rodman in [38], di seguito si riportano le conclusioni più importanti:

Teorema 2.4.4. *Si consideri la $\mathcal{CARE}$ (2.28) e si supponga che le matrici B e C siano semidefinite positive e che le coppie di matrici (A, B) e (A^*, C) siano c-stabilizzabili. Se il primo termine X_0 della successione (2.40) è hermitiano e verifica $\sigma(A-BX_0)subseteq\mathbb{C}_l$, allora la successione $\{ X_k \}_{k\in\mathbb{N}}$ ha le seguenti proprietà:*

- *$\sigma(A - BX_k)subseteq\mathbb{C}_l$ per ogni $k \in \mathbb{N}$,*
- *$X_1 \succeq X_2 \succeq \cdots \succeq X_+$*
- *$\lim_{k\to\infty} X_k = X_+$.*

Inoltre la convergenza della successione $\{ X_k \}_{k\in\mathbb{N}}$ è quadratica, ovvero

$$\|X_{k+1} - X_+\| \leq \gamma \|X_k - X + \|^2,$$

per $k \in \mathbb{N}$ e per una opportuna costante positiva γ.

Si osservi che la successione $\{ X_k \}_{k\in\mathbb{N}}$ risulta monotona decrescente (per l'ordinamento $\succeq$) non dal termine X_0, ma solo dal successivo, in quanto, in generale, può non verificarsi $X_0 \succeq X_1$. Particolarmente problematico nell'implementazione dell'algoritmo è l'individuazione del valore iniziale X_0 in quanto le ipotesi richieste dal Teorema 2.4.4 sono molto restrittive. Se la matrice A è c-stabile, allora è possibile porre $X_0 = 0$, altrimenti si rendono necessarie tecniche particolari che sfruttano un 'temine fittizio' X_{-1} non necessariamente hermitiano tale che $\sigma(A - BX_{-1}) \subset \mathbb{C}_l$, per poi calcolare il termine X_0. Per maggiori dettagli si rimanda a [10].

Le caratteristiche numeriche del metodo si possono riassumere come

costo computazionale per passo per individuare X_{k+1} e dunque, per risolvere il sistema (2.40), occore risolvere l'equazione di Lyapunov

$$(A^* - X_kB)H_k + H_k(A - BX_k) = -\mathcal{R}(X)$$

la cui risoluzione richiede circa $35n^3$ operazioni con l'algoritmo di Bartels e Stewart. Si osservi che il calcolo di $\mathcal{R}(X_k)$ fornisce, per ogni iterazione, una stima della bontà della approssimazione.

convergenza Il metodo presenta convergenza quadratica se il termine X_0 è hermitiano e verifica $\sigma(A_BX_0) \subset \mathbb{C}_l$. Per individuare il termine X_0 è necessario adottare un artificio particolare.

stabilità numerica La stabilità del metodo dipende dal condizionamento della matrice $\mathcal{R}'_{X_+}$, dunque se la soluzione estremale X_+ è critica possono esserci problemi di mal condizionamento.

2.4.4 Metodo di Newton per $\mathcal{DARE}$

Si consideri la $\mathcal{DARE}$ (2.31), allora, per la (2.23), il **metodo di Newton per** $\mathcal{DARE}$ genera la successione $\{ X_k \}_{k\in\mathbb{N}}$ definita dal sistema:

$$\begin{cases} \hat{A}_k^* H_k \hat{A}_k - H_k = -\mathcal{R}(X) \\ X_{k+1} = X_k + H_k \end{cases} \tag{2.41}$$

dove

$$\hat{A}^* := A - B(R + B^* X_k B)^{-1}(B^* X_k A + C)$$

E $\mathcal{R}(X)$ è l'operatore di Riccati definito in (2.32).

La convergenza della successione $\{ X_k \}_{k\in\mathbb{N}}$ alla soluzione estremale X_+ è assicurata dal seguente teorema ([38]:

Teorema 2.4.5. *Si supponga siano verificate le ipotesi del Teorema 1.4.10 e che la soluzione estremale X_+ della* (2.31) *verifichi la relazione $R + B^* X_+ B \succ 0$. Se il primo termine X_0 della successione* (2.41) *è hermitiana d-stabilizzante, allora la successione $\{ X_k \}_{k\in\mathbb{N}}$ verifica*

- *X_k è d-stabilizzante per ogni $k \in \mathbb{N}$,*
- *$X_1 \succeq X_2 \succeq \cdots \succeq X_+$,*
- *$\lim_{k\to\infty} X_k = X_+$.*

Inoltre se la soluzione estremale X_+ è d-stabilizzante, allora la convergenza della successione è quadratica, ovvero

$$\|X_{k+1} - X_+\| \leq \gamma \|X_k - X + \|^2,$$

per $k \in \mathbb{N}$ e per una opportuna costante positiva γ.

Per quanto riguarda gli aspetti numerici del metodo si ha:

costo computazionale per passo per individuare X_{k+1} e dunque, per risolvere il sistema (2.41), occore risolvere l'equazione di Stein simmetrica

$$\hat{A}_k^* H_k A_k - H_k = \mathcal{R}(X),$$

la cui risoluzione richiede circa $35n^3$ operazioni con l'algoritmo di Bartels e Stewart. Anche in questo caso il calcolo di $\mathcal{R}(X_k)$ fornisce, per ogni iterazione, una stima della bontà della approssimazione.

convergenza Il metodo presenta convergenza quadratica se il termine X_0 è hermitiano d-stabilizzante. Tale richiesta è molto restrittiva e limita molto l'utilizzo del metodo di Newton per la risoluzione di $\mathcal{DARE}$.

stabilità numerica La stabilità del metodo dipende dal condizionamento della matrice $\mathcal{R}'_{X_+}$.

Capitolo 3

Doubling algorithms

Indice

I METODI RISOLUTIVI introdotti nel capitolo precedente, come già sottolineato, non risultano particolarmente efficaci per la risoluzioni di equazioni di Riccati in quanto presentano un oneroso costo computazionale e sono inficiati da problemi di stabilità numerica. Il presente capitolo offre una panoramica sui metodi numericamente più robusti e performanti per la risoluzione di $\mathcal{ARE}$: i *doubling algorithm*. Tale denominazione è stata introdotta da Sima ([53]) e per enfatizzare la convergenza quadratica dei suddetti algoritmi. I doubling algorithm sfruttano e reiterpretano le proprietà delle $\mathcal{ARE}$ illustrate nella sezione 1.4.1, ovvero la corrispondenza tra sottospazi invarianti o di deflazione e soluzioni delle equazioni di Riccati. Per meglio sviluppare le potenzialità di tali algoritmi e garantire la convergenza dei metodi, è opportuno convertire le proprietà di c-stabilità delle soluzioni estramali delle $\mathcal{ARE}$ con proprietà di d-stabilità, si rendono pertanto necessarie le trasformazioni affini e le trasformazioni di Cayley introdotte nella sezione 1.3.2.

Al fine di rendere al meglio le caratteristiche dei doubling algorithms, sono inoltre presentate alcune implementazioni degli algoritmi proposti.

Il paragrafo 3.1 illustra il metodo *doubling algorithm strutturato* (denotato con l'acronimo inglese $\mathcal{SDA}$). La filosofia genearale del metodo $\mathcal{SDA}$ è quella di generare una successione di matrix pencil che ammettono medesimi sottospazi di deflazione ma relativi ad autovalori che vengono elevati al quandrato ad ogni passo dell'algoritmo. Fondamentale per il corretto utilizzo del metodo è sfruttare proprietà di d-stabilità di tali sottospazi di deflazione. Sono quindi riportate le applicazioni del metodo $\mathcal{SDA}$ per la individuazione delle soluzioni estremali di $\mathcal{NARE}$, $\mathcal{CARE}$ e $\mathcal{DARE}$. Per le prime due equazioni è necessario, quindi, convertire le caratteristiche di c-stabilità in proprietà di d-stabilità di tali soluzioni, e dunque vengono proposti due differenti approcci basati sulle trasformazioni affini e sulle trasformazioni di Cayley.

Nel paragrafo 3.2 viene descritto il metodo di *riduzione ciclica* (indicato con la dicitura $\mathcal{CR}$). L'idea di fondo del metodo $\mathcal{CR}$ è quella di generare una successione di polinomi matriciali quadratici i cui autovalori vengono, ad ogni passo, elevati al quadrato. In analogia con il metodo $\mathcal{SDA}$, alla base dell'algoritmo vi sono le proprietà di d-splitting di tali autovalori che garantiscono la convergenza del metodo. Sono quindi presentate le caratteristiche salienti delle equazioni unilaterali quadratiche, è introdotto il concetto di autovalori di una matrix pencil quadratica, e sono illustrate due modi per portare una $\mathcal{NARE}$ nella forma $\mathcal{UQME}$: le trasformazioni semplici e le trasformazioni basate sulla fattorizzazione UL. Dopo una descrizione di carattere generale del metodo di $\mathcal{CR}$, sono presentate le applicazioni della riduzione ciclica alle $\mathcal{NARE}$ e ad una particolare $\mathcal{DARE}$. Anche per il metodo di $\mathcal{CR}$ sono illustrati, due differenti algoritmi per la risoluzione di $\mathcal{NARE}$ a seconda che vengano adoperate le trasformazioni affini o le trasformazioni di Cayley.

Nel paragrafo 3.3 si traducono gli algoritmi descritti nei precedenti paragrafi in codici. Sono quindi analizzate le proprietà numeriche di ciascun metodo: costo computazionale, velocità di convergenza e stabilità, e vengono presentate alcune implementazioni degli algoritmi $\mathcal{SDA}$ e $\mathcal{CR}$ per le $\mathcal{ARE}$. Per ciascuna implementazione vengono sottolineati numero di iterazioni necessarie, errore relativo e tempo di utilizzo della CPU. Le sperimentzioni sono effettuate in MATLAB 2008*b*.

3.1 Doubling algorithm strutturato

Il metodo **doubling algorithm strutturato** genera una successione di matrix pencil con la proprietà che ad ogni iterazione dell'algoritmo i sottospazi di deflazione rimangono inalterati, mentre i relativi autovalori vengono elevati al quadrato. Tale proprietà risulta molto utile se il sottospazio di deflazione è d-stabile, in tal caso, infatti, è evidente che se l'algoritmo può essere iterato senza interruzioni (ovvero senza **breakdown**), gli autovalori ad ogni passo hanno norma sempre più piccola, fino a tendere a zero. Questa proprietà rende particolarmente semplificata la ricerca del sottospazio di deflazione d-stabile e fornisce, quindi, un valido metodo per calcolarlo.

Come sopra esposto, la denominazione **doubling** di tali algoritmi vuole sottolineare la convergenza quadratica dei metodi stessi, mentre per **strutturato** si intende la capacità dell'algoritmo di preservare nel corso delle iterazioni la particolare struttura delle matrici che definiscono le matrix pencil generata in ciascun passo.

L'attuale formulazione del metodo $\mathcal{SDA}$ è dovuta ai matematici cinesi E.K.-W. Chu, H.-Y. Fan, W.-W. Lin e C.-S. Wang che, in [18, 19], applicano tale metodo alle equazioni di Riccati. Altri importanti risultati sono stati provati da C.-H. Guo, W.-W. Lin e S.-F. Xu in [30].

3.1.1 Descrizione generale del metodo $\mathcal{SDA}$

Siano N, $K \in \mathbb{C}^{(m+n)\times(m+n)}$ e si consideri la matrix pencil $\mathcal{P}(z) := N - zK$. Sia, inoltre, $\mathcal{V} \subseteq \mathbb{C}^{m+n}$ un graph deflating subspace n-dimensionale relativo alla matrix pencil $\mathcal{P}(z)$, esistono dunque una matrice $X \in \mathbb{C}^{m\times n}$ ed una matrice $W \in \mathbb{C}^{n\times n}$ tali che

$$N \begin{pmatrix} I_n \\ X \end{pmatrix} = K \begin{pmatrix} I_n \\ X \end{pmatrix} W. \tag{3.1}$$

L'obiettivo è ora quello di costruire una successione di matrix pencil

$$\{\, \mathcal{P}_k(z) := N_k - zK_k \,\}_{k\in\mathbb{N}}$$

che abbiano medesimo graph deflating subspace $\mathcal{V}$ ma relativo ad autovalori differenti, in particolare, utilizzando le notazioni precedenti, si richiede che la successione $\{\, \mathcal{P}_k(z) \,\}_{k\in\mathbb{N}}$ verifichi la seguente relazione

$$N_k \begin{pmatrix} I_n \\ X \end{pmatrix} = K_k \begin{pmatrix} I_n \\ X \end{pmatrix} W^{2^k} \tag{3.2}$$

per $k = 0, 1 \dots$.

Per generare una successione di matrix pencil $\{ \mathcal{P}_k(z) \}_{k\in\mathbb{N}}$ che verifica le proprietà volute sono necessarie le seguenti proposizioni:

Proposizione 3.1.1. *Si consideri la matrix pencil $\mathcal{P}(z) := N - zK$ definita dalla* (3.1) *e siano $\{ N_k \}_{k\in\mathbb{N}}$ e $\{ K_k \}_{k\in\mathbb{N}}$ successioni di matrici definite da*

$$\begin{cases} K_{k+1}^{-1} N_{k+1} = (K_k^{-1} N_k)^2, \\ N_0 = N, \\ K_0 = K. \end{cases} \tag{3.3}$$

Allora, per ogni valore di k, le matrici N_k e K_k verificano la relazione (3.2) *e dunque, ponendo $\mathcal{P}_k(z) := N_k - zK_k$ per $k = 0, 1, \dots$, la successione di matrix pencil $\{ \mathcal{P}_k(z) \}_{k\in\mathbb{N}}$ verifica le proprietà volute.*

Dimostrazione. La tesi è provata per induzione. Per $k = 0$ la tesi è banalmente vera. Si supponga ora che N_k e K_k verifichino la relazione (3.2) e che la matrice K_k sia invertibile, allora

$$(K_k^{-1} N_k)^2 \begin{pmatrix} I_n \\ X \end{pmatrix} = K_k^{-1} N_k \begin{pmatrix} I_n \\ X \end{pmatrix} W^{2^k} = \begin{pmatrix} I_n \\ X \end{pmatrix} W^{2^k} W^{2^k},$$

pertanto

$$N_{k+1} \begin{pmatrix} I_n \\ X \end{pmatrix} = K_{k+1} \begin{pmatrix} I_n \\ X \end{pmatrix} W^{2^{k+1}},$$

e dunque la matrix pencil $\{ \mathcal{P}_k(z) \}_{k\in\mathbb{N}}$ verifica le proprietà volute. □

La proposizione precedente è fortemente vincolata alla invertibilità delle matrici K_k, qualora tale condizione non fosse garantita, è possibile utilizzare la seguente proposizione di carattere più generale:

Proposizione 3.1.2. *Si consideri la matrix pencil $\mathcal{P}(z) := N - zK$ definita dalla* (3.1). *Si supponga che esistano le successioni di matrici $\{ L_k \}_{k\in\mathbb{N}}$, $\{ U_k \}_{k\in\mathbb{N}}$, $\{ N_k \}_{k\in\mathbb{N}}$ e $\{ K_k \}_{k\in\mathbb{N}}$ definite da*

$$\begin{cases} N_{k+1} = L_k N_k, \\ K_{k+1} = U_k K_k, \\ N_0 = N, \\ K_0 = K. \end{cases} \tag{3.4}$$

con

$$U_k N_k = L_k K_k \qquad \text{per } k = 0, 1 \dots$$

Allora, per ogni valore di k, le matrici N_k e K_k verificano la relazione (3.2) *e dunque, ponendo $\mathcal{P}_k(z) := N_k - zK_k$ per $k = 0, 1, \dots$, la successione di matrix pencil $\{ \mathcal{P}_k(z) \}_{k\in\mathbb{N}}$ verifica le proprietà volute.*

Dimostrazione. La tesi è provata per induzione. Per $k = 0$ la tesi è vera per ipotesi. Se si suppone che la tesi sia verificata per ogni $j \le k$, allora

$$N_{k+1} \begin{pmatrix} I_n \\ X \end{pmatrix} = L_k N_k \begin{pmatrix} I_n \\ X \end{pmatrix} = L_k K_k \begin{pmatrix} I_n \\ X \end{pmatrix} W^{2^k} = \\ U_k N_k \begin{pmatrix} I_n \\ X \end{pmatrix} W^{2^k} = U_k K_k \begin{pmatrix} I_n \\ X \end{pmatrix} W^{2^k} W^{2^k} = K_{k+1} \begin{pmatrix} I_n \\ X \end{pmatrix} W^{2^{k+1}},$$

da cui la tesi. □

Nelle ipotesi della proposizione precedente, si ha il seguente teorema

Teorema 3.1.1. *Si supponga che la matrix pencil* $\mathcal{P}(z) := N - zK$ *definita in* (3.1) *sia regolare e che esistano le successioni introdotte nella proposizione 3.1.2. Allora le matrix pencil* $\mathcal{P}_k(z) := N_k - zK_k$ *sono regolari e ad ogni iterazione, gli autovalori della nuova matrix pencil sono i quadrati degli autovalori della matrix pencil calcolata al passo precedente, ovvero*

$$\det(\mathcal{P}_k(\lambda)) = \det(N_k - \lambda K_k) = 0 \quad \textit{implica} \quad \det(\mathcal{P}_{k+1}(\lambda^2)) = \det(N_{k+1} - \lambda^2 K_{k+1}) = 0$$

per $k = 0, 1 \ldots$.

Si osservi che la tesi del teorema 3.1.1 può essere riscritta come segue: se λ è un'autovalore della matrix pencil $\mathcal{P}(z)$, allora λ^{2^k} è un autovalore della matrix pencil $\mathcal{P}_k(z)$ per ogni k naturale.

La costruzione testé introdotta si rivela particolarmente significativa se la matrice W risulta d-stabile. Se si suppone, infatti, che la successione $\{ K_k \}_{k\in\mathbb{N}}$ sia limitata in norma, allora, per la (3.2), si ha

$$\lim_{k\to\infty} N_k \begin{pmatrix} I_n \\ X \end{pmatrix} = \lim_{k\to\infty} K_k \begin{pmatrix} I_n \\ X \end{pmatrix} W^{2^k} = 0,$$

pertanto se la successione $\{ N_k \}_{k\in\mathbb{N}}$ ammettesse limite $\bar{N}$, si otterrebbe la relazione

$$\bar{N} \begin{pmatrix} I_n \\ X \end{pmatrix} = 0,$$

e dunque, per individuare il graph subspace n-dimensionale in oggetto, sarebbe sostanzialmente sufficiente risolvere un sistema lineare.

Espresse le generalità del metodo $\mathcal{SDA}$ ed il fondamento teorica su cui si sviluppa, occorre descrivere in modo concreto come calcolare la successione di matrix pencil $\{ \mathcal{P}_k(z) \}_{k\in\mathbb{N}}$ definita nella proposizione 3.1.2. In tale ottica, un criterio rilevante nella costruzione di tale tale successione, è quello di 'preservare' nelle iterazioni dell'algoritmo la particolare struttura iniziale delle matrici N e K.

Si supponga che le matrici N, $K \in \mathbb{C}^{(m+n)\times(m+n)}$ abbiano la seguente forma

$$N = \begin{pmatrix} E & 0 \\ -P & I_m \end{pmatrix}, \qquad K = \begin{pmatrix} I_n & -G \\ 0 & F \end{pmatrix}, \tag{3.5}$$

in tal caso si dice che le matrici N e K sono in **forma strutturata standard-I**. L'obiettivo prefissato è quello di determianare una successione di matrici $\{ N_k \}_{k\in\mathbb{N}}$ e $\{ K_k \}_{k\in\mathbb{N}}$ che abbiano medesima struttura, ovvero abbiano la forma

$$N_k := \begin{pmatrix} E_k & 0 \\ -P_k & I_m \end{pmatrix}, \qquad K_k := \begin{pmatrix} I_n & -G_k \\ 0 & F_k \end{pmatrix}. \tag{3.6}$$

Teorema 3.1.2. *Siano, per* $k \geq 0$, N_k *e* K_k *in forma* (3.5)*, e si supponga che la matrice* $I_n - G_k P_k$ *sia invertibile. Allora esistono due matrici* L_k *e* U_k *in forma strutturata standard-I con*

$$L_k := \begin{pmatrix} L_{11}^{(k)} & 0 \\ L_{21}^{(k)} & I_m \end{pmatrix}, \qquad U_k := \begin{pmatrix} I_n & U_{12}^{(k)} \\ 0 & U_{22}^{(k)} \end{pmatrix},$$

tali che $U_k N_k = L_k K_k$. *È possibile dunque definire la matrici*

$$N_{k+1} := L_k N_k \qquad K_{k+1} := U_k K_k$$

che presentano la struttura (3.6) *con*

$$\begin{aligned} E_{k+1} &= E_k (I_n - G_k P_k)^{-1} E_k, \\ P_{k+1} &= P_k + F_k (I_m - P_k G_k)^{-1} P_k E_k \\ F_{k+1} &= F_k (I_m - P_k G_k)^{-1} F_k, \\ G_{k+1} &= G_k + E_k (I_n - G_k P_k)^{-1} G_k F_k. \end{aligned} \tag{3.7}$$

Dimostrazione. Si osservi preliminarmente che se la matrice $I_n - G_kP_k$ è non singolare, allora la matrice G_kP_k non ha $\lambda = 1$ come autovalore, pertanto anche la matrice P_kG_k non ha $\lambda = 1$ come autovalore e la matrice $I_m - P_kG_k$ risulta a sua volta invertibile.

Dalle relazioni

$$U_kN_k = \begin{pmatrix} I_n & U_{12}^{(k)} \\ 0 & U_{22}^{(k)} \end{pmatrix} \begin{pmatrix} E_k & 0 \\ -P_k & I_m \end{pmatrix} = \begin{pmatrix} E_k - U_{12}^{(k)}P_k & U_{12}^{(k)} \\ -U_{22}^{(k)}P_k & U_{22}^{(k)} \end{pmatrix}$$

e

$$L_kK_k = \begin{pmatrix} L_{11}^{(k)} & 0 \\ L_{21}^{(k)} & I_m \end{pmatrix} \begin{pmatrix} I_n & -G_k \\ 0 & F_k \end{pmatrix} = \begin{pmatrix} L_{11}^{(k)} & -L_{11}^{(k)}G_k \\ L_{21}^{(k)} & -L_{21}^{(k)}G_k + F_k \end{pmatrix}$$

ed imponendo la condizione $U_kN_k = L_kK_k$ si ottiene

$$\begin{aligned} L_{11}^{(k)} &= E_k(I_n - G_kP_k)^{-1} & L_{21}^{(k)} &= -F_k(I_m - P_kG_k)^{-1}P_k, \\ U_{12}^{(k)} &= -E_k(I_n - G_kP_k)^{-1}G_k & U_{22}^{(k)} &= F_k(I_m - P_kG_k)^{-1}, \end{aligned}$$

e dunque per le ipotesi di invertibilità di $I_n - G_kP_k$ e $I_m - P_kG_k$, le matrici L_k e U_k sono ben definite.

Infine, per definizione

$$N_{k+1} = L_kN_k = \begin{pmatrix} E_k(I_n - G_kP_K)^{-1}E_k & 0 \\ -(P_k + F_k(I_m - P_kG_K)^{-1}P_kE_k) & I_m \end{pmatrix}$$

e

$$K_{k+1} = U_kK_k = \begin{pmatrix} I_n & -(G_k + E_k(I_n - G_kP_K)^{-1}G_kF_k) \\ 0 & F_k(I_m - P_kG_K)^{-1}F_k \end{pmatrix},$$

dunque le matrici N_{k+1} e K_{k+1} sono in forma strutturata standard-I (3.6) con sottomatrici definite dalla (3.7). □

Se le matrici definite nel teorema precedente possono essere calcolate per ogni valore di k, ovvero se la matrici $I_n - G_kP_k$ sono invertibili per ogni k, è possibile generare le successioni $\{ N_k \}_{k\in\mathbb{N}}$ e $\{ K_k \}_{k\in\mathbb{N}}$ e tali successioni verificano le ipotesi della proposizione 3.1.2 e dunque vale l'uguaglianza

$$N_k \begin{pmatrix} I_n \\ X \end{pmatrix} = K_k \begin{pmatrix} I_n \\ X \end{pmatrix} W^{2^k}. \tag{3.8}$$

Alla luce della precedente relazione è possibile enunciare e provare il seguente fondamentale risultato:

Teorema 3.1.3. *Si supponga che le matrici $I_n - G_kP_k$ siano invertibili per ogni valore di k e che*

- *la matrice W sia d-stabile,*
- *esista $\bar{P} := \lim_{k\to\infty} P_k$,*
- *la successione $\{ F_k \}_{k\in\mathbb{N}}$ sia limitata per qualche norma $\|\cdot\|$,*

allora sono verificate le seguenti relazioni

$$X = \bar{P} \qquad \limsup_{k\to\infty} \sqrt[2^k]{\|X - P_k\|} \le \varrho(W)$$

dove $\varrho(\cdot)$ indica il raggio spettrale di una matrice.

Dimostrazione. Si osservi che per la (3.8) e per la (3.6), vale

$$\begin{pmatrix} E_k & 0 \\ -P_k & I_m \end{pmatrix} \begin{pmatrix} I_n \\ X \end{pmatrix} = \begin{pmatrix} I_n & -G_k \\ 0 & F_k \end{pmatrix} \begin{pmatrix} I_n \\ X \end{pmatrix} W^{2^k}. \tag{3.9}$$

Considerando la seconda riga a blocchi dall'equazione precedente si ottiene pertanto

$$\lim_{k\to\infty} -P_k + X = \lim_{k\to\infty} F_k W^{2^k} \tag{3.10}$$

da cui, essendo $\varrho(W) < 1$ e F_k limitata, si ricava $X = \bar{P}$.

Allo stesso modo, dalla seconda riga a blocchi della 3.10, si ottiene

$$\|X - P_k\| \leq \|F_k\|\|X\|\|W\|^{2^k},$$

da cui, per le ipotesi su W e sulla successione $\{ F_k \}_{k\in\mathbb{N}}$, si ha

$$\limsup_{k\to\infty} \sqrt[2^k]{\|X - P_k\|} \leq \|W\| \leq \varrho(W).$$

□

La strategia risulutiva per individuare un sottospazio di deflazione d-stabile relativo alla matrix pencil $\mathcal{P}(z)$ introdotta dai Tereome 3.1.2 e 3.1.3 va sotto il nome di **doubling algorithm strutturato-I** ed è in letteratura indicato con l'acronimo $\mathcal{SDA}$-I. Si dice che il metodo $\mathcal{SDA}$-I ha un **breakdown** al k-esimo passo se la matrice $I_n - G_k P_k$ non è invertibile e, dunque, se l'algoritmo deve necessariamente arrestarsi alla k-esima iterazione.

Il teorema che segue, illustrato in [10], rende più forti le maggiorazioni espresse nel teorema 3.1.3 sotto ipotesi più restrittive sull'esistenza di graph deflating subspace della matrix pencil $\mathcal{P}(z)$.

Teorema 3.1.4. *Si supponga che il metodo $\mathcal{SDA}$-I non abbia breakdown e che esistano $X \in \mathbb{C}^{m\times n}$, $Y \in \mathbb{C}^{n\times m}$ e $W \in \mathbb{C}^{n\times n}$, $V \in \mathbb{C}^{m\times m}$ tali che*

$$N \begin{pmatrix} I_n \\ X \end{pmatrix} = K \begin{pmatrix} I_n \\ X \end{pmatrix} W, \qquad N \begin{pmatrix} Y \\ I_m \end{pmatrix} V = K \begin{pmatrix} Y \\ I_m \end{pmatrix},$$

con

$$\varrho(W) \leq 1, \qquad \varrho(V) \leq 1, \qquad \varrho(W)\varrho(V) < 1.$$

Allora valgono le maggiorazioni

$$\limsup_{k\to\infty} \sqrt[2^k]{\|X - P_k\|}, \leq \varrho(W)\varrho(V) \qquad \limsup_{k\to\infty} \sqrt[2^k]{\|F_k\|} \leq \varrho(V),$$
$$\limsup_{k\to\infty} \sqrt[2^k]{\|Y - G_k\|}, \leq \varrho(W)\varrho(V) \qquad \limsup_{k\to\infty} \sqrt[2^k]{\|E_k\|} \leq \varrho(W).$$

Inoltre

$$X = \lim_{k\to\infty} P_k \qquad Y = \lim_{k\to\infty} G_k.$$

Dimostrazione. Adottando un procedimento analogo a quello utilizzato nella Proposzione 3.1.2 è possibile dimostrare che vale la relazione

$$N_k \begin{pmatrix} Y \\ I_m \end{pmatrix} V^{2^k} = K_k \begin{pmatrix} Y \\ I_m \end{pmatrix}$$

e dunque, per la (3.6),

$$\begin{pmatrix} E_k & 0 \\ -P_k & I_m \end{pmatrix} \begin{pmatrix} Y \\ I_m \end{pmatrix} V^{2^k} = \begin{pmatrix} I_n & -G_k \\ 0 & F_k \end{pmatrix} \begin{pmatrix} Y \\ I_m \end{pmatrix}. \tag{3.11}$$

Esplicitando la (3.9) e la (3.11), si ottengono le uguaglianze

$$E_k = (I_n - G_k X)W^{2^k} \qquad X - P_k = F_k X W^{2^k}, \tag{3.12}$$

$$F_k = (I_m - P_k Y)V^{2^k} \qquad Y - G_k = E_k X V^{2^k}, \tag{3.13}$$

da cui

$$X - P_k = (I_m - P_k Y)V^{2^k} X W^{2^k}, \qquad Y - G_k = (I_n - G_k X)W^{2^k} X V^{2^k}, \tag{3.14}$$

e quindi, passando alle norme ed estreando la radice 2^k ed entrambi i membri delle equazioni precedenti si ottiene

$$\limsup_{k\to\infty} \sqrt[2^k]{\|X - P_k\|}, \leq \varrho(W)\varrho(V) \qquad \limsup_{k\to\infty} \sqrt[2^k]{\|F_k\|} \leq \varrho(V), \tag{3.15}$$

$$\limsup_{k\to\infty} \sqrt[2^k]{\|Y - G_k\|}, \leq \varrho(W)\varrho(V) \qquad \limsup_{k\to\infty} \sqrt[2^k]{\|E_k\|} \leq \varrho(W). \tag{3.16}$$

Per la (3.14), inoltre, si ha

$$X - V^{2^k} X W^{2^k} = P_k - P_k Y V^{2^k} X W^{2^k} = P_k(I_n - Y V^{2^k} X W^{2^k})$$
$$Y - W^{2^k} Y V^{2^k} = G_k - G_k X W^{2^k} Y V^{2^k} = G_k(I_m - X W^{2^k} Y V^{2^k})$$

pertanto, essendo $\varrho(W)\varrho(V) < 1$, si ottiene

$$\lim_{k\to\infty} P_k = X \qquad \lim_{k\to\infty} G_k = Y.$$

□

Come illustrato nei paragrafi successivi, non sempre nelle matrix pencil oggetto di studio sono immediatamente riscontrabili le proprietà strutturali richieste dal metodo $\mathcal{SDA}$-I. Il seguente risultato, provato da Poloni in [49], presenta delle condizioni affinché una matrix pencil generica sia simile ad una matrix pencil definita da matrici in forma strutturata standard-I.

Teorema 3.1.5. *Siano $\tilde{\mathcal{P}}(z) := \tilde{N} - z\tilde{K}$ una matrix pencil regolare con $\tilde{N}$ e $\tilde{K}$ matrici quadrate $m+n$-dimensionali partizionate come segue*

$$\tilde{N} := \begin{pmatrix}\tilde{N}_1 & \tilde{N}_2\end{pmatrix}, \qquad \tilde{K} := \begin{pmatrix}\tilde{K}_1 & \tilde{K}_2\end{pmatrix}$$

dove $\tilde{N}_1$ e $\tilde{K}_1 \in \mathbb{C}^{(m+n)\times n}$. Allora esiste una matrix pencil $\mathcal{P}(z) := N - zK$ definita da matrici in forma strutturata standard-I (3.5) *simile a destra alla matrix pencil $\tilde{\mathcal{P}}(z)$ se e solo se la matrice*

$$S := \begin{pmatrix}\tilde{K}_1 & \tilde{N}_2\end{pmatrix}$$

è invertibile. In tal caso, utilizzando le notazioni adottate in (3.5)*, vale la relazione*

$$\begin{pmatrix} E & -G \\ -P & F \end{pmatrix} = S^{-1}\begin{pmatrix}\tilde{N}_1 & \tilde{K}_2\end{pmatrix}.$$

Dimostrazione. La matrix pencil $\mathcal{P}(z)$ risulta simile a destra alla matrix pencil $\tilde{\mathcal{P}}(z)$ se e solo se esiste una matrice $R \in \mathbb{C}^{(m+n)\times(m+n)}$ tale che

$$\mathcal{P}(z) = N - zK = R(\tilde{N} - z\tilde{K}) = R\tilde{\mathcal{P}}(z).$$

Sfruttando le partizioni delle matrici in gioco, la precedente equazione si riscrive come

$$R\left(\begin{pmatrix}\tilde{N}_1 & \tilde{N}_2\end{pmatrix} - z\begin{pmatrix}\tilde{K}_1 & \tilde{K}_2\end{pmatrix}\right) = \begin{pmatrix} E & 0 \\ -P & I_m \end{pmatrix} - z\begin{pmatrix} I_n & -G \\ 0 & F \end{pmatrix},$$

da cui

$$R\begin{pmatrix}\tilde{K}_1 & \tilde{N}_2\end{pmatrix} = RS = \begin{pmatrix}I_n & 0\\ 0 & I_m\end{pmatrix}, \qquad R\begin{pmatrix}\tilde{N}_1 & \tilde{K}_2\end{pmatrix} = \begin{pmatrix}E & -G\\ -P & F\end{pmatrix}.$$

La prima relazione implica che la matrice S è invertibile con inversa $S^{-1} = R$, mentre dalla seconda equazione si ottiene

$$\begin{pmatrix}E & -G\\ -P & F\end{pmatrix} = S^{-1}\begin{pmatrix}\tilde{N}_1 & \tilde{K}_2\end{pmatrix},$$

da cui la tesi. □

Oltre al metodo $\mathcal{SDA}$-I esistono altre tipologie di doubling algorithm strutturato che sfruttano strutture alternative delle matrici N e K. Per dovere di completezza è opportuno ricordare il metodo $\mathcal{SDA}$-**II** in cui si suppone che $n = m$ e che le matrici N e K siano in **forma strutturata standard-II** ovvero siano del tipo

$$N := \begin{pmatrix}-P & I_n\\ E & 0\end{pmatrix}, \qquad K := \begin{pmatrix}F & 0\\ G & -I_n\end{pmatrix}.$$

In tal caso, sotto opportune ipotesi di non singolarità, è possibile generare successioni di matrici in forma strutturata standard-II in analogia a quanto ottenuto per il metodo $\mathcal{SDA}$-I, con medesime proprietà di convergenza.

Tutt'altra filosofia è invece alla base del metodo $\mathcal{SDA}$-QR introdotto da Benner e Byers in [4]. Tale metodo genera le successioni $\{N_k\}_{k\in\mathbb{N}}$ e $\{K_k\}_{k\in\mathbb{N}}$, utilizzando ad ogni passo la fattorizzazione QR della matrice $\left(\begin{smallmatrix}K_k\\ -N_k\end{smallmatrix}\right)$. Tra i vantaggi del $\mathcal{SDA}$-QR va annoverata la mancanza di breakdown e dunque la possibilità di portare l'algoritmo a termine, nel contempo, però, il metodo non preserva la particolare struttura delle matrici N e K e presenta, quindi, un elevato costo computazionale.

Per una presentazione puntuale dei suddetti metodi si rimanda a [10]. Nel seguito il metodo $\mathcal{SDA}$-I sarà chiamato senza ambiguità semplicemente $\mathcal{SDA}$.

3.1.2 Metodo $\mathcal{SDA}$ per $\mathcal{NARE}$

Si consideri la $\mathcal{NARE}$

$$C + XA + DX - XBX = 0 \tag{3.17}$$

e siano $\mathcal{M}$ e $\mathcal{H}$ rispettivamente la matrice dei coefficienti e la matrice hamiltoniana associata alla $\mathcal{NARE}$. Per quanto mostrato nella sezione 1.4.2, se la matrice dei coefficienti $\mathcal{M}$ è una M-matrice non singolare o singolare irriducibile, allora esista la soluzione minimale non negativa $X_{\min}$ della $\mathcal{NARE}$ (3.17). In tal caso, inoltre, la matrice hamiltoniana $\mathcal{H}$ presenta un (m, n) c-splitting, la matrice $A - BX_{\min}$ è una M-matrice e pertanto la soluzione $X_{\min}$ risulta c-antistabilizzante. Dunque, per la ben nota relazione

$$\mathcal{H}\begin{pmatrix}I_n\\ X_{\min}\end{pmatrix} = \begin{pmatrix}I_n\\ X_{\min}\end{pmatrix}(A - BX_{\min}),$$

si ottiene che le colonne della matrice $\left(\begin{smallmatrix}I_n\\ X_{\min}\end{smallmatrix}\right)$, generano il sottospazio di deflazione c-antistabile della matrix pencil $\mathcal{Q}(z) := \mathcal{H} - zI_{m+n}$.

Per poter applicare il metodo $\mathcal{SDA}$ illustrato nella sezione precedente, occorre trasformare la matrix pencil $\mathcal{Q}(z)$ in una nuova matrix pencil $\tilde{\mathcal{P}}(z) := \tilde{N} - z\tilde{K}$ tale che il sottospazio di deflazione c-antistabile della matrix pencil $\mathcal{Q}(z)$ sia il sottospazio di deflazione d-stabile di $\tilde{\mathcal{P}}(z)$. Occorre dunque individuare una funzione analitica $\mathcal{F}$ tale che la matrice $\mathcal{F}(A - BX_{\min})$ sia d-stabile. Nella sezione 1.3.2 sono state proposte due tipologie di trasformazioni, le trasformazioni affini $\mathcal{A}_\alpha$ e le trasformazioni di Cayley $\mathcal{C}_\gamma$. È necessario, quindi, individuare delle condizioni sui parametri α e su γ affinché le suddette trasformazioni verifichino la proprietà voluta.

Per le trasformazioni affini si ponga dunque

$$\tilde{\mathcal{P}}_\alpha(z) = \tilde{N}_\alpha - z\tilde{K}_\alpha = \mathcal{A}_\alpha(\mathcal{Q}(z)) = \mathcal{A}_\alpha(\mathcal{H} - zI_{m+n}) = \alpha\mathcal{H} - I_{m+n} - zI_{m+n}, \quad (3.18)$$

quindi, per il teorema 1.3.1,

$$(\alpha\mathcal{H} - I_{m+n}) \begin{pmatrix} I_n \\ X_{\min} \end{pmatrix} = \begin{pmatrix} I_n \\ X_{\min} \end{pmatrix} W_\alpha$$

con $W_\alpha := \mathcal{A}_\alpha(A - BX_{\min})$.

Si osservi ora che gli autovalori della matrice $A - BX_{\min}$ corrispondono agli n autovalori di $\mathcal{H}$ con parte reale positiva. Per costruzione inoltre, tali autovalori si trovano nell'unione dei cerchi di Gershgorin relativi alle prime n righe della matrice $\mathcal{M}$. Essendo $\mathcal{M}$ una M-matrice non singolare o singolare irriducibile, quindi, gli autovalori si trovano nella circonferenza di centro δ e e raggio δ, dove

$$\delta := \max_{i=1,\dots,n} a_{ii},$$

con a_{ij} elemento sulla riga i e sulla colonna j della matrice A. Ponendo, dunque, $\alpha \leq \frac{1}{\delta}$ e indicati con λ_i per $i = 1, \dots n$ gli autovalori di $A - BX_{\min}$, si ottiene $|\mathcal{A}_\alpha(\lambda_i)| \leq 1$, come voluto.

Un secondo passaggio 'propedeutico' all'utilizzo del metodo $\mathcal{SDA}$ è quello di portare la matrix pencil $\tilde{\mathcal{P}}_\alpha(z)$ in forma strutturata standard-I. Per il teorema 3.1.5 esiste una matrix pencil $\mathcal{P}_\alpha(z) := N_\alpha - zK_\alpha$ simile a destra a $\tilde{\mathcal{P}}_\alpha(z)$ se e solo se la matrice

$$S_\alpha := \begin{pmatrix} I_n & -\alpha B \\ 0 & -D_\alpha \end{pmatrix}$$

con $D_\alpha := \alpha D + I_m$ è invertibile. In tal caso, vale

$$\begin{aligned} \begin{pmatrix} E^{(\alpha)} & -G^{(\alpha)} \\ -P^{(\alpha)} & F^{(\alpha)} \end{pmatrix} &= \begin{pmatrix} I_n & -\alpha B \\ 0 & -D_\alpha \end{pmatrix}^{-1} \begin{pmatrix} \alpha A - I_n & 0 \\ -\alpha C & I_m \end{pmatrix} \\ &= \begin{pmatrix} I_n & -\alpha B D_\alpha^{-1} \\ 0 & -D_\alpha^{-1} \end{pmatrix} \begin{pmatrix} \alpha A - I_n & 0 \\ -\alpha C & I_m \end{pmatrix} \\ &= \begin{pmatrix} \alpha A - I_n - \alpha^2 B D_\alpha^{-1} C & -\alpha B D_\alpha^{-1} \\ \alpha D_\alpha^{-1} C & -D_\alpha^{-1} \end{pmatrix}. \end{aligned} \quad (3.19)$$

La matrici appena calcolate, possono essere dunque utilizzate per inizializzare il metodo $\mathcal{SDA}$ e generare le successioni $\{ E_k^{(\alpha)} \}_{k\in N}$, $\{ F_k^{(\alpha)} \}_{k\in N}$, $\{ G_k^{(\alpha)} \}_{k\in N}$ e $\{ P_k^{(\alpha)} \}_{k\in N}$. Per quanto riguarda la convergenza del metodo $\mathcal{SDA}$ vale il seguente risultato presentato in [13]

Teorema 3.1.6. *Si consideri la $\mathcal{NARE}$ (3.17) e siano $X_{\min}$ e Y_{min} le soluzioni minimali non negative rispettivamente della $\mathcal{NARE}$ e della sua duale. Si supponga che la matrice dei coefficienti $\mathcal{M}$ associata alla $\mathcal{NARE}$ (3.17) sia non singolare o singolare irriducibile. Sia $\alpha \leq \frac{1}{\delta}$, allora le matrici*

$$I_n - G_k^{(\alpha)} P_k^{(\alpha)}, \qquad I_m - P_k^{(\alpha)} G_k^{(\alpha)}$$

generate dal metodo $\mathcal{SDA}$ a partire dai valori iniziali definiti in (3.19) sono M-matrici non singolari, dunque il metodo $\mathcal{SDA}$ non presenta breakdown, e risulta $E_k^{(\alpha)}$, $F_k^{(\alpha)}$, $G_k^{(\alpha)}$, $P_k^{(\alpha)} \geq 0$.

Inoltre se la matrice $\mathcal{M}$ è non singolare o singolare irriducibile con drift non nullo, allora

$$\limsup_{k\to\infty} \sqrt[2^k]{\|P_k^{(\alpha)} - X_{\min}\|} \le \frac{\tau_1}{\tau_2},$$
$$\limsup_{k\to\infty} \sqrt[2^k]{\|G_k^{(\alpha)} - Y_{min}\|} \le \frac{\tau_1}{\tau_2},$$
$$\limsup_{k\to\infty} \sqrt[2^k]{\|E_k\|} \le \tau_1,$$
$$\limsup_{k\to\infty} \sqrt[2^k]{\|F_k\|} \le \frac{1}{\tau_2},$$

con

$$\tau_1 := \max_{i=n,\dots,1} |\mathcal{A}_\alpha(\lambda_i)| \le 1 \qquad \tau_2 := \min_{i=m+n,\dots,n+1} |\mathcal{A}_\alpha(\lambda_i)| \ge 1,$$

dove λ_i sono gli autovalori di $\mathcal{H}$ per $i = 1,\dots,m+n$, con parti reali ordinate in modo decrescente.

Per quanto riguarda le trasformazioni di Cayley, si ponga

$$\tilde{\mathcal{P}}_\gamma(z) = \tilde{N}_\gamma - z\tilde{K}_\gamma = \mathcal{C}_\gamma(\mathcal{Q}(z)) = \mathcal{C}_\gamma(\mathcal{H} - zI_{m+n}) = \mathcal{H} - \gamma I_{m+n} - z(\mathcal{H} + \gamma I_{m+n}), \quad (3.20)$$

quindi, per il teorema 1.3.1

$$(\mathcal{H} - \gamma I_{m+n}) \begin{pmatrix} I_n \\ X_{\min} \end{pmatrix} = \begin{pmatrix} I_n \\ X_{\min} \end{pmatrix} W_\gamma$$

con $W_\gamma := \mathcal{C}_\gamma(A - BX_{\min})$.

Si osservi che, per il teorema 1.3.2, scegliendo $\gamma > 0$, la trasformazione $\mathcal{C}_\gamma$ mappa il semipiano complesso destro $\mathbb{C}_r$ nel disco aperto $\mathcal{D}$ ed il semipiano complesso chiuso sinistro $\mathbb{C}_{l,0}$ in $\bar{\mathcal{D}}^c$. È evidente, quindi, che per ogni autovalore λ_i di $A - BX_{\min}$ per $i = 1,\dots n$, $|\mathcal{C}_\gamma(\lambda_i)| \le 1$, dunque tale scelta di γ soddisfa le proprietà volute.

Come per le trasformazioni affini, occorre portare la matrix pencil $\tilde{\mathcal{P}}_\gamma(z)$ in forma strutturata standard-I. In tal caso, utilizzando le notazioni del teorema 3.1.5, si ha

$$S := \begin{pmatrix} A + \gamma I_n & -B \\ -C & -D - \gamma I_m \end{pmatrix},$$

e

$$\begin{aligned} \begin{pmatrix} E^{(\gamma)} & -G^{(\gamma)} \\ -P^{(\gamma)} & F^{(\gamma)} \end{pmatrix} &= \begin{pmatrix} A + \gamma I_n & -B \\ -C & -D - \gamma I_m \end{pmatrix}^{-1} \begin{pmatrix} \alpha A - \gamma I_n & -B \\ -C & -D + \gamma I_m \end{pmatrix} \\ &= \begin{pmatrix} A + \gamma I_n & -B \\ C & D + \gamma I_m \end{pmatrix}^{-1} \begin{pmatrix} \alpha A - \gamma I_n & -B \\ C & D - \gamma I_m \end{pmatrix} \\ &= (\mathcal{M} + \gamma I_{m+n})^{-1}(\mathcal{M} - \gamma I_{m+n}) = \mathcal{C}_\gamma(\mathcal{M}). \end{aligned} \quad (3.21)$$

Si osservi che, essendo $\mathcal{M}$ una M-matrice, la matrice $\mathcal{M} + \gamma I_{m+n}$ è, per valori di $\gamma > 0$, una M-matrice non invertibile, in particolare per una delle solite caratterizzazioni delle M-matrici, ammette inversa non negativa. Se, inoltre, si sceglie

$$\gamma \ge \max\{ \max_{i=1,\dots,n} a_{ii}, \max_{i=j,\dots,m} d_{jj} \}, \quad (3.22)$$

allora la matrice $M - \gamma I_{m+n}$ risulta non positiva. Con tale scelta di γ, quindi, la matrice $\mathcal{C}_\gamma(\mathcal{M})$ è non negativa e può essere calcolata senza alcuna cancellazione. In particolare si ha

$$E^{(\gamma)}, F^{(\gamma)} \le 0, \qquad P^{(\gamma)}, G^{(\gamma)} \ge 0.$$

Le matrici testé introdotte, possono essere utilizzate come valori iniziali per generare le successioni definite dal metodo $\mathcal{SDA}$. Per quanto riguarda la convergenza, vale il seguente teorema ([28]):

Teorema 3.1.7. *Si consideri la $\mathcal{NARE}$ (3.17) e siano $X_{\min}$ e Y_{min} le soluzioni minimali non negative rispettivamente della $\mathcal{NARE}$ e della sua duale. Si supponga che la matrice dei coefficienti $\mathcal{M}$ associata alla $\mathcal{NARE}$ (3.17) sia non singolare o singolare irriducibile. Sia γ scelto secondo la maggiorazione (3.22), allora le matrici*

$$I_n - G_k^{(\gamma)} P_k^{(\gamma)}, \qquad I_m - P_k^{(\gamma)} G_k^{(\gamma)}$$

generate dal metodo $\mathcal{SDA}$ a partire dai valori iniziali definiti in (3.21) sono M-matrici non singolari, dunque il metodo $\mathcal{SDA}$ non presenta breakdown, e risulta

$$0 \le P_k^{(\gamma)} < P_{k+1}^{(\gamma)} < X_{\min}, \qquad 0 \le G_k^{(\gamma)} < G_{k+1}^{(\gamma)} < Y_{min},$$

ovvero le successioni $\{ P_k^{(\gamma)} \}_{k\in\mathbb{N}}$ e $\{ G_k^{(\gamma)} \}_{k\in\mathbb{N}}$ sono monotone convergenti. Inoltre, se la matrice $\mathcal{M}$ è non singolare o singolare irriducibile con drift non nullo, allora

$$\limsup_{k\to\infty} \sqrt[2^k]{\|P_k^{(\gamma)} - X_{\min}\|} \le \frac{\sigma_1}{\sigma_2},$$
$$\limsup_{k\to\infty} \sqrt[2^k]{\|G_k^{(\gamma)} - Y_{min}\|} \le \frac{\sigma_1}{\sigma_2},$$
$$\limsup_{k\to\infty} \sqrt[2^k]{\|E_k^{(\gamma)}\|} \le \sigma_1,$$
$$\limsup_{k\to\infty} \sqrt[2^k]{\|F_k^{(\gamma)}\|} \le \frac{1}{\sigma_2},$$

con

$$\sigma_1 := \max_{i=n,\dots,1} |\mathcal{C}_\gamma(\lambda_i)| \le 1 \qquad \sigma_2 := \min_{i=m+n,\dots,n+1} |\mathcal{C}_\gamma(\lambda_i)| \ge 1,$$

dove λ_i sono gli autovalori di $\mathcal{H}$ per $i = 1, \dots, m+n$, con parti reali ordinate in modo decrescente.

I teoremi 3.1.6 e 3.1.7 stabiliscono i raggi di convergenza del metodo $\mathcal{SDA}$ nel caso in cui si utilizzi una trasformazione affine o una trasformazione di Cayley. È opportuno dunque individuare quale trasformazione garantisce la convergenza più rapida. Si osservi che

$$|\mathcal{C}_\gamma(\lambda)| = \left|\frac{\lambda-\gamma}{\lambda+\gamma}\right| = \left(1 - \frac{\lambda}{\gamma}\right)^2 + O\left(\left(\frac{\lambda}{\gamma}\right)^2\right) = \left(\mathcal{A}_{\frac{1}{\gamma}}(\lambda)\right)^2 + O\left(\left(\frac{\lambda}{\gamma}\right)^2\right),$$

ponendo, quindi, $\alpha := \frac{1}{\gamma}$, e mutuando le notazioni dei teoremi 3.1.6 e 3.1.7 si ottiene

$$\frac{\sigma_1}{\sigma_2} = \left|\frac{\mathcal{C}_\gamma(\lambda_n)}{\mathcal{C}_\gamma(\lambda_{n+1})}\right| = \left|\frac{(\mathcal{A}_\alpha(\lambda_n))^2}{(\mathcal{A}_\alpha(\lambda_{n+1}))^2}\right| = \left(\frac{\tau_1}{\tau_2}\right)^2.$$

La precedente relazione esprime che il metodo $\mathcal{SDA}$ ottenuto a partire da una trasformazione di Cayley necessita, per raggiungere la convergenza, di meno passi del metodo $\mathcal{SDA}$ ottenuto da una trasformazione affine, soprattutto nel caso in cui gli elementi diagonali della matrice A sono maggiori degli elementi diagonali della matrice D. Se, invece, gli elementi della matrice A sono minori degli elementi diagonali dalla matrice D, allora, scegliendo un valore $\alpha < \frac{1}{\gamma}$ il metodo $\mathcal{SDA}$ ottenuto da una trasformazione affine risulta avere raggio di convergenza minore di quello ottenuto da una trasformazione di Cayley.

3.1.3 Metodo $\mathcal{SDA}$ per $\mathcal{CARE}$

Si consideri la $\mathcal{CARE}$

$$C + XA + A^*X - XBX = 0, \tag{3.23}$$

dove le matrici B, C sono hermitiane. L'interesse degli algoritmi e delle applicazioni, come ampiamente descritto, si focalizza principalmente sulle soluzioni estremali X_- e X_+.

Riassumendo brevemente quanto mostrato nel paragrafo 1.4, si ricorda che la matrice hamiltoniana $\mathcal{H}$ presenta un (n, n) c-splitting ovvero gli autovalori sono ordinati nel modo seguente

$$Re(\lambda_{2n}) \leq \cdots \leq Re(\lambda_{n+1}) \leq 0 \leq Re(\lambda_n) \leq \cdots \leq Re(\lambda_1),$$

mentre, per la simpletticità della matrice $\mathcal{H}$, se λ è autovalore di $\mathcal{H}$, lo è anche $-\bar{\lambda}$. Inoltre le soluzioni estremali X_- e X_+ verificano

$$\mathcal{H}\begin{pmatrix} I_n \\ X_- \end{pmatrix} = \begin{pmatrix} I_n \\ X_- \end{pmatrix}(A - BX_-), \qquad \mathcal{H}\begin{pmatrix} I_n \\ X_+ \end{pmatrix} = \begin{pmatrix} I_n \\ X_+ \end{pmatrix}(A - BX_+)$$

con

$$\sigma(A - BX_-) \subseteq \mathbb{C}_{r,0}, \qquad \sigma(A - BX_+) \subseteq \mathbb{C}_{l,0},$$

dove $\sigma(\cdot)$ indica lo spettro di una matrice, ovvero la soluzione X_- risulta (quasi) c-antistabilizzante e la soluzione X_+ è (quasi) c-stabilizzante.

Per la soluzione X_-, se la matrice dei coefficienti $\mathcal{M}$ è una M-matrice non singolare o singolare irriducibile rimane valido l'impianto presentato nella sezione precedente per il calcolo della soluzione $X_{\min}$.

Al solito, per trasformare le proprietà di c-stabilità (c-antistabilità) in proprietà di d-stabilità, è opportuno utilizzare le trasformazioni affini $\mathcal{A}_\alpha$ o le trasformazioni di Cayley $\mathcal{C}_\gamma$. Per la soluzione minimale X_-, occorre trasformare la c-antistabilità in d-stabilità, per quanto mostrato nella sezione 1.3.2, è possibile dunque utilizzare

- le trasformazioni affini $\mathcal{A}_\alpha$ con $\alpha \leq \frac{1}{\delta}$, dove $\delta := \max_{i=1,\ldots,n} |\lambda_i|$,
- le trasformazioni di Cayley $\mathcal{C}_\gamma$ con $\gamma > 0$.

Per la soluzione massimale X_+, è necessario, invece, trasformare la c-stabilità in d-stabilità, pertanto le trasformazioni idonee sono

- le trasformazioni affini $\tilde{\mathcal{A}}_\alpha$ con $\alpha \leq \frac{1}{\delta}$ $\tilde{\mathcal{A}}_\alpha(z) := \alpha z + 1$,
- le trasformazioni di Cayley $\mathcal{A}_\gamma$ con $\gamma < 0$.

Si osservi che l'utilizzo delle trasformazioni affini implica il calcolo degli autovalori della matrice hamiltoniana $\mathcal{H}$, dunque tale strategia si rende particolarmente svantaggiosa.

Si consideri dunque la matrix pencil

$$\tilde{\mathcal{P}}_\gamma(z) := \mathcal{C}_\gamma(\mathcal{H} - zI_{m+n}) = \mathcal{H} - \gamma I_{m+n} - z(\mathcal{H} + \gamma I_{m+n})$$

con $\gamma > 0$ per il calcolo della soluzione minimale X_- o con $\gamma < 0$ per il calcolo della soluzione massimale X_+. Sfruttando l'hermitianità della matrix pencil $\tilde{\mathcal{P}}_\gamma(z)$, è possibile portare la matrix pencil $\tilde{\mathcal{P}}_\gamma(z)$ in forma strutturata simplettica standard-I $\hat{\mathcal{P}}_\gamma(z) := \hat{N}_\gamma - z\hat{K}_\gamma$, con

$$\hat{N}_\gamma := \begin{pmatrix} \hat{E}_\gamma & 0 \\ -\hat{P}_\gamma & I_m \end{pmatrix}, \qquad \hat{K}_\gamma := \begin{pmatrix} I_n & -\hat{G}_\gamma \\ 0 & \hat{E}_\gamma^* \end{pmatrix}.$$

Imponendo le condizioni del teorema 3.1.5, si ottiene l'equazione

$$\begin{pmatrix} \hat{E}_\gamma & -\hat{G}_\gamma \\ -\hat{P}_\gamma & \hat{E}_\gamma^* \end{pmatrix} = \begin{pmatrix} A + \gamma I_n & -B \\ -C & -A^* - \gamma I_n \end{pmatrix}^{-1} \begin{pmatrix} A - \gamma I_n & -B \\ -C & -A^* + \gamma I_n \end{pmatrix}$$

da cui si ottiene

$$\begin{aligned} \hat{E}_\gamma &:= W_\gamma(A - \gamma I_n + B(A^* + \gamma I_n)^{-1}C), \\ \hat{F}_\gamma &:= W_\gamma B(I_n - (A^* + \gamma I_n)^{-1}(A^* - \gamma I_n)), \\ \hat{P}_\gamma &:= -W_\gamma^* C(I_n - (A^* + \gamma I_n)^{-1}(A^* - \gamma I_n)), \end{aligned}$$

con

$$W_\gamma := (A + \gamma I_n + B(A^* + \gamma I_n)^{-1}C)^{-1}.$$

In tal caso, partendo dai valori iniziali

$$E_0^{(\gamma)} := \hat{E}_\gamma, \quad G_0^{(\gamma)} := \hat{G}_\gamma, \quad P_0^{(\gamma)} := \hat{P}_\gamma,$$

il metodo $\mathcal{SDA}$ genera le successioni $\{ E_k^\gamma \}_{k\in\mathbb{N}}$, $\{ G_k^\gamma \}_{k\in\mathbb{N}}$, $\{ P_k^\gamma \}_{k\in\mathbb{N}}$ definite da

$$\begin{aligned} E_{k+1}^{(\gamma)} &:= E_k^{(\gamma)}(I_n - G_k^{(\gamma)}P_k^{(\gamma)})^{-1}E_k^{(\gamma)}, \\ G_{k+1}^{(\gamma)} &:= G_k^{(\gamma)} + E_k^{(\gamma)}(I_n - G_k^{(\gamma)}P_k^{(\gamma)})^{-1}G_k^{(\gamma)}(E_k^{(\gamma)})^*, \\ F_{k+1}^{(\gamma)} &:= P_k^{(\gamma)} + (E_k^{(\gamma)})^*(I_n - P_k^{(\gamma)}G_k^{(\gamma)})^{-1}P_k^{(\gamma)}E_k^{(\gamma)}. \end{aligned} \tag{3.24}$$

Per quanto riguarda la convergenza del metodo $\mathcal{SDA}$, vale il seguente risultato illustrato in [18].

Teorema 3.1.8. *Si supponga che la $\mathcal{CARE}$ (3.23) ammetta soluzioni estremali X_- e X_+. Allora le successioni definite in (3.24) dal metodo $\mathcal{SDA}$ verificano*

$$P_k^{(\gamma)} \succeq 0 \qquad G_k^{(\gamma)} \preceq 0,$$

e le matrici $I_n - G_k^{(\gamma)}P_k^{(\gamma)}$ e $I_n - P_k^{(\gamma)}G_k^{(\gamma)}$ risultano invertibili, dunque il metdo $\mathcal{SDA}$ può essere iterato senza breakdown. Inoltre, siano $\mu > 0$ e $\eta < 0$, allora

$$\begin{aligned} \limsup_{k\to\infty} \sqrt[2^k]{\|P_k^{(\mu)} - X_-\|} &\le \sigma^2, \\ \limsup_{k\to\infty} \sqrt[2^k]{\|P_k^{(\eta)} - X_+\|} &\le \sigma^2, \\ \limsup_{k\to\infty} \sqrt[2^k]{\|E_k^{(\cdot)}\|} &\le \sigma \end{aligned}$$

dove $\sigma := \max_{i=1,\dots,n} |\mathcal{C}_\gamma(\lambda_i)|$.

3.1.4 Metodo $\mathcal{SDA}$ per $\mathcal{DARE}$

Si consideri la $\mathcal{DARE}$

$$X = A^*XA - A^*XB(R + B^*XB)^{-1}B^*XA + Q. \tag{3.25}$$

Si osservi che dalla precedente $\mathcal{DARE}$ si ricava

$$\begin{aligned} X &= A^*XA - A^*XB(RB^{-1}(I_n + BR^{-1}B^*X)B)^{-1}B^*XA + Q \\ &= A^*XA - A^*X(I_n + BR^{-1}B^*X)BR^{-1}B^*XA + Q \\ &= A^*X(I_n - (I_n + BR^{-1}B^*X)BR^{-1}B^*X)A + Q \\ &= A^*X(I_n + BR^{-1}B^*X)^{-1}A + Q, \end{aligned}$$

da cui, ponendo $G := BR^{-1}B^*$, si ottiene la $\mathcal{DARE}$ nella nuova formulazione

$$X = A^*X(I_n + GX)^{-1}A + Q. \tag{3.26}$$

Siano ora

$$N := \begin{pmatrix} A & 0 \\ -Q & I_n \end{pmatrix}, \qquad K := \begin{pmatrix} I_n & G \\ 0 & A^* \end{pmatrix},$$

e si consideri la matrix pencil $\mathcal{P}(z) := N - zK$. Sia X una soluzione della $\mathcal{DARE}$ (3.26), allora

$$\begin{aligned} N\begin{pmatrix} I_n \\ X \end{pmatrix} = \begin{pmatrix} A & 0 \\ -Q & I_n \end{pmatrix}\begin{pmatrix} I_n \\ X \end{pmatrix} = \begin{pmatrix} A \\ -Q + X \end{pmatrix} = \begin{pmatrix} I_n + GX \\ A^*X \end{pmatrix}(I_n + GX)^{-1}A = \\ = \begin{pmatrix} I_n & G \\ 0 & A^* \end{pmatrix}\begin{pmatrix} I_n \\ X \end{pmatrix}(I_n + GX)^{-1}A = K\begin{pmatrix} I_n \\ X \end{pmatrix}(I_n + GX)^{-1}A, \end{aligned}$$

ovvero le colonne della matrice $\binom{I_n}{X}$ generano un sottospazio di deflazione n-dimensionale relativo alla matrix pencil $\mathcal{P}(z)$.

Si supponga che esista una soluzione X della $\mathcal{DARE}$ (3.26) d-stabilizzante ovvero tale che la matrice $(I_n + GX)^{-1}A$ sia d-stabile, allora, per individuare la soluzione X è possibile utilizzare il metodo $\mathcal{SDA}$. Si osservi che la matrix pencil $\mathcal{P}(z)$ è in forma strutturata standard-I, allora, per il teorema 3.1.2, il metodo $\mathcal{SDA}$ genera le successioni $\{ A_k \}_{k\in\mathbb{N}}$, $\{ G_k \}_{k\in\mathbb{N}}$, $\{ Q_k \}_{k\in\mathbb{N}}$ definite da

$$\begin{aligned} A_{k+1} &= A_k(I_n + G_kQ_k)^{-1}A_k, \\ G_{k+1} &= G_k + A_kG_k(I_n + Q_kG_k)^{-1}A_k^*, \\ Q_{k+1} &= Q_k + A_k^*(I_n + Q_kG_k)^{-1}Q_kA_k. \end{aligned}$$

Il seguente teorema, illustrato in [19], mostra quali ipotesi devono essere verificate affinchè il metodo $\mathcal{SDA}$ converga alla soluzione d-stabilizzante:

Teorema 3.1.9. *Sia X una soluzione d-stabilizzante della $\mathcal{DARE}$ (3.26) e si supponga che la matrix pencil $\mathcal{P}(z)$ non abbia autovalori sulla circonferenza unitaria S^1, che esistano due matrici non singolari U, V tali che*

$$UNV = \begin{pmatrix} J_s & 0 \\ 0 & I_n \end{pmatrix}, \qquad UKV = \begin{pmatrix} I_n & 0 \\ 0 & J_s \end{pmatrix},$$

con $\sigma(J_s) \subseteq \mathcal{D}$, dove $\sigma(\cdot)$ indica lo spettro di una matrice e si consideri la seguente partizione della matrice V

$$V = \begin{pmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{pmatrix}$$

con $V_{11} \in \mathbb{C}^{n\times n}$. Allora, se le matrici V_{11} V_{22} sono invertibili, il metodo $\mathcal{SDA}$ non presenta breakdown e sono verificate le relazioni:

$$\begin{aligned} X &= \lim_{k\to\infty} Q_k, \\ \limsup_{k\to\infty} \sqrt[2^k]{\|X - Q_k\|} &= |\lambda_n|^2, \\ \limsup_{k\to\infty} \sqrt[2^k]{\|A_k\|} &= |\lambda_n|, \end{aligned}$$

dove

$$|\lambda_1| \leq \ldots |\lambda_n| < 1 < \frac{1}{|\lambda_n|} \leq \cdots \leq \frac{1}{|\lambda_1|}$$

sono gli autovalori della matrix pencil $\mathcal{P}(z)$ (ponendo $\infty = \frac{1}{0}$).

3.2 Riduzione ciclica

Come ampiamente descritto nel paragrafo precedente, il metodo $\mathcal{SDA}$ genera una successione di matrix pencil i cui autovalori sono i quadrati degli autovalori della matrix pencil calcolata all'iterazione precedente. Il metodo di **riduzione ciclica** si basa sulla medesima idea di elevamento al quadrato generando una successione di polinomi matriciali quadratici $\{ \mathcal{A}_k(z) \}_{k\in\mathbb{N}}$.

È dunque opportuno, prima di descrivere il metodo $\mathcal{CR}$, presentare le proprietà delle equazioni matriciali unilaterali quadratiche definite nella sezione 1.1.5 e studiare le proprietà spettrali delle soluzioni.

La prima formulazione del metodo di riduazione ciclica è dovuta a G.H. Golub e R.W. Hockney ([33]): gli autori utilizzano il metodo $\mathcal{CR}$ per risolvere sistemi lineari ottenuti discretizzando un'equazione di Poisson definita su un rettangolo. Una versione attualizzata del metodo $\mathcal{CR}$ è stata proposta da D.A. Bini e B. Meini ([9]) per risolvere equazioni matriciali quadratiche ed equazioni matriciali di serie di potenze.

3.2.1 Proprietà delle $\mathcal{UQME}$ e relazioni con le $\mathcal{ARE}$

Si consideri il polinomio matriciale quadratico nell'indeterminata X

$$\mathcal{A}(X) := A_0 + A_1X + A_2X^2 \tag{3.27}$$

con A_0, A_1, A_2, $X \in \mathbb{C}^{n\times n}$ e A_2 matrice non nulla.

Alla (3.27) è possibile associare la $\mathcal{UQME}$

$$\mathcal{A}(X) := A_0 + A_1X + A_2X^2 = 0 \tag{3.28}$$

e la **matrix pencil quadratica**

$$\mathcal{A}(z) := A_0 + zA_1z + z^2A_2 = 0. \tag{3.29}$$

Uno scalare λ si dice **autovalore** della matrix pencil quadratica (3.29) se è una radice del polinomio

$$a(z) := \det(\mathcal{A}(z)).$$

Si osservi che $a(z)$ è un polinomio di grado al più $2n$, in particolare, se $\det(A_2) \neq 0$, si ha

$$a(z) = \det(\mathcal{A}(z)) = \det(A_2)\det(A_2^{-1}A_0 + zA_2^{-1}A_1 + z^2I_n),$$

dunque $a(z)$ ha grado esattamente $2n$ e pertanto ha esattamente $2n$ radici. Se invece la matrice A_2 è non invertibile, $a(z)$ ha grado $r < n$, quindi ha r radici. In tal caso si pongono convenzionalmene $n - r$ autovalori $\lambda = \infty$.

Particolarmente importante nella risoluzione della $\mathcal{UQME}$ $\mathcal{A}(X) = 0$ è sfruttare evenutuali proprietà spettrali della matrix pencil $\mathcal{A}(z)$. In gran parte delle applicazioni, infatti, gli autovalori di $\mathcal{A}(z)$ presentano un (n, n)-d splitting, in tal caso sono verificate le proprietà menzionate nel seguente teorema.

Teorema 3.2.1. *Si supponga che la matrix pencil* (3.29) *abbia esattamente n autovalori (contati con molteplicità) nel disco chiuso $\bar{\mathcal{D}}$ e sia X una soluzione d-stabile della $\mathcal{UQME}$* (3.28)*. Allora X è l'unica soluzione d-stabile ed è la soluzione con raggio spettrale minimo.*

Dimostrazione. Si osservi che gli autovalori della matrice X sono esattamente gli n autovalori di $\mathcal{A}(z)$ di modulo al più unitario. Sia, infatti, λ un autovalore di X e v il relativo autovettore, allora, essendo X soluzione della (3.28), si ha

$$\mathcal{A}(X)v = (A_0 + A_1X + A_2X^2)v = (A_0 + A_1\lambda + A_2\lambda^2)v = \mathcal{A}(\lambda)v = 0,$$

dunque $a(\lambda) = \det(\mathcal{A}(\lambda)) = 0$, e quindi λ è un autovalore della matrix pencil $\mathcal{A}(z)$.

Se $\tilde{X}$ è un'altra soluzione d-stabile della (3.28), allora deve avere necessariamente i medesimi autovalori di X ovvero i medesimi autovalori di $\mathcal{A}(z)$ nel disco $\bar{\mathcal{D}}$. In particolare, quindi, la matrici $\tilde{X}$ e X hanno la medesima forma canonica di Jordan e dunque devono necessariamente coincidere, da cui l'unicità della soluzione d-stabile. È evidente, quindi, che ogni soluzione $\tilde{X}$ differente da X deve avere almeno un autovalore di $\mathcal{A}(z)$ in $\mathcal{D}^c$ e quindi vale

$$\varrho(X) \leq 1 < \varrho(\tilde{X}),$$

pertanto X è la soluzione con raggio spettrale minimo. □

L'equazione matriciale quadratica

$$A_2 + YA_1 + Y^2A_0 = 0 \tag{3.30}$$

si dice $\mathcal{UQME}$ **duale** della $\mathcal{UQME}$ (3.28).

Anche dalle soluzioni dell'equazione duale di una $\mathcal{UQME}$ è possibile descrivere proprietà spettrali interessanti. Vale infatti il seguente risultato:

Proposizione 3.2.1. *Sia X una soluzione della* (3.28) *e si supponga che la matrice $A_1 + A_2X$ sia non singolare. Allora esiste una soluzione Y della $\mathcal{UQME}$ duale* (3.30) *tale che se λ è un autovalore di $\mathcal{A}(z)$ allora o λ è un autovalore di X o $\frac{1}{\lambda}$ è un autovalore di Y dove si pone $\frac{1}{0} = \infty$.*

Dimostrazione. Si ponga

$$Y := -A_2(A_1 + A_2X)^{-1},$$

allora $Y(A_1 + A_2X) + A_2 = 0$, quindi, essendo X soluzione della $\mathcal{UQME}$

$$Y(A_1+A_2X)X+A_2X = -YA_0+A_2X = -Y^2A_0+YA_2X = -Y^2A_0-YA_1+Y(A_1+A_2X) = 0,$$

e quindi la matrice Y risolve la $\mathcal{UQME}$ (3.30).

Si osservi ora che valgono le uguaglianze

$$\begin{aligned}\mathcal{A}(z) &= A_0 + zA_1 + z^2A_2 = -A_1X - A_2X^2 + zA_1 + z^2A_2 = \\ &= A_1(zI_n - X) - A_2X^2 + zA_2X - zA_2X + z^2A_2 \\ &= (A_1 + A_2X + zA_2)(zI_n - X) = (I_n - zY)(A_1 + A_2X)(zI_n - X),\end{aligned}$$

allora λ è un autovalore di $\mathcal{A}(z)$ se e solo λ è un autovalore di X oppure $\frac{1}{\lambda}$ è un autovalore di Y. □

Meritevole di citazione è il teorema che segue, presentato in [11], in cui ancora una volta vengono descritte l'esistenza e le proprietà spettrali di una soluzione della $\mathcal{UQME}$ a partire dalle proprietà spettrali della realativa matrix pencil quadratica.

Teorema 3.2.2. *Si consideri la $\mathcal{UQME}$* (3.28) *e la matrix pencil* (3.29)*, sia*

$$\mathcal{L}(z) := z^{-1}\mathcal{A}(z) = z^{-1}A_0 + A_1 + zA_2$$

e si supponga che esista una fattorizzazione $\mathcal{L}(z) = \mathcal{P}(z)\mathcal{N}(z)$ con

$$\mathcal{N}(z) := z^{-1}N_{-1} + N_0, \qquad \mathcal{P}(z) := P_0 + zP_1$$

tale che $\det(\mathcal{P}(z))$ *non si annulla in $\bar{\mathcal{D}}$, mentre* $\det(\mathcal{N}(z))$ *non si annulla in $\mathcal{D}^c$. Allora*

$$X := -N_0^{-1}N_{-1},$$

è l'unica soluzione strettamente d-stabile della $\mathcal{UQME}$ (3.28)*.*

Viceversa se esiste una soluzione strettamente d-stabile X della $\mathcal{UQME}$ (3.28) *e se $\mathcal{A}(z)$ ha un (n, n) d-splitting forte, allora la seguente fattorizzazione*

$$\mathcal{L}(z) = (A_1 + A_2X + zA_2)(-z^{-1}X + I_n).$$

verifica le richieste precedenti.

Dimostrazione. Si supponga che esista la fattorizzazione $\mathcal{L}(z) = \mathcal{P}(z)\mathcal{N}(z)$, con le proprietà richieste, allora essendo $\det(\mathcal{N}(z)) \neq 0$ in $\mathcal{D}^c$, allora necessariamente la matrice N_0 è invertibile e dunque è ben definita la matrice $X = -N_0^{-1}N_{-1}$. Esplicitando la fattorizzazione si ottiene

$$\mathcal{A}(z) = z\mathcal{L}(z) = z(P_0 + zP_1)(z^{-1}N_{-1} + N_0) = P_0N_{-1} + z(P_0N_0 + P_1N_{-1}) + z^2P_1N_0,$$

quindi

$$\mathcal{A}(-N_0^{-1}N_{-1}) = 0$$

dunque la matrice X è effettivamente soluzione della $\mathcal{UQME}$ (3.28). Si osservi che

$$\mathcal{N}(z) = z^{-1}N_0(zI_n + N_0^{-1}N_{-1}) = z^{-1}N_0(zI_n - X)$$

allora, essendo $\det(\mathcal{N}(z)) \neq 0$ in $\mathcal{D}^c$, la matrice X risultra strettamente d-stabile. Inoltre, poiché la matrix pencil $\mathcal{P}$ non ha autovalori in $\bar{\mathcal{D}}$, la matrix pencil quadratica $\mathcal{A}(z)$ ha

esattamente n autovalori in $\bar{\mathcal{D}}$, allora per il teorema 3.2.1, la matrice X è l'unica soluzione strettamente d-stabile.

Per quanto riguarda la seconda parte dell'enuciato, sfruttando i calcoli svolti nella proposizione 3.2.1 si ha

$$\mathcal{L}(z) = z^{-1}\mathcal{A}(z) = (A_1 + A_2X + zA_2)(I_n - z^{-1}X).$$

Essendo X strettamente d-stabile, $\det(I_n - z^{-1}X)$ non si annulla su $\mathcal{D}^c$, quindi, poiché $\mathcal{A}(z)$ ha un (n,n) d-splitting forte, necessariamente $\det(A_1 + A_2X + zA_2) \neq 0$ su $\bar{\mathcal{D}}$. □

Come per le $\mathcal{ARE}$, è possibile stabilire una corrispondenza biunivoca tra soluzioni di una $\mathcal{UQME}$ e sottospazi di deflazione di particolari matrix pencil. Tale corrispondenza determina, come illustrato in seguito, un metodo per passare da una $\mathcal{ARE}$ ad una $\mathcal{UQME}$ e viceversa.

Teorema 3.2.3. *Si consideri la $\mathcal{UQME}$ (3.28) e si ponga*

$$N := \begin{pmatrix} 0 & I_n \\ -A_0 & -A_1 \end{pmatrix}, \qquad K := \begin{pmatrix} I_n & 0 \\ 0 & A_2 \end{pmatrix}.$$

Allora è possibile stabilire una corrispondenza biunivoca tra le soluzioni X della $\mathcal{UQME}$ (3.28) e i graph deflating subspace n-dimensionali della matrix pencil $\mathcal{P}(z) := N - zK$. In particolare X è soluzione della $\mathcal{UQME}$ (3.28) se e solo se

$$N \begin{pmatrix} I_n \\ X \end{pmatrix} = K \begin{pmatrix} I_n \\ X \end{pmatrix} X.$$

Dimostrazione. È sufficente osservare, essendo X una soluzione della $\mathcal{UQME}$, che

$$\begin{aligned} N \begin{pmatrix} I_n \\ X \end{pmatrix} &= \begin{pmatrix} 0 & I_n \\ -A_0 & -A_1 \end{pmatrix} \begin{pmatrix} I_n \\ X \end{pmatrix} = \begin{pmatrix} X \\ -A_0 - A_1X \end{pmatrix} \\ &= \begin{pmatrix} X \\ A_2X^2 \end{pmatrix} = \begin{pmatrix} I_n & 0 \\ 0 & A_2 \end{pmatrix} \begin{pmatrix} I_n \\ X \end{pmatrix} X = K \begin{pmatrix} I_n \\ X \end{pmatrix} X. \end{aligned}$$

□

Le proprietà spettrali delle $\mathcal{UQME}$ e delle relative soluzioni e la corrispondenza tra sottospazi di deflazione e soluzione permettono di descrivere dei metodi per trasformare una $\mathcal{NARE}$ in una $\mathcal{UQME}$.

Si consideri la $\mathcal{NARE}$

$$C + XA + DX - XBX \tag{3.31}$$

sia $\mathcal{H}$ la relativa matrice hamiltoniana e si supponga che la matrice dei coefficienti $\mathcal{M}$ sia una M-matrice non singolare o singolare irriducibile. Per quanto mostrato nella sezione 3.2.1, se X è soluzione della $\mathcal{NARE}$ (3.31), vale la relazione

$$\mathcal{H} \begin{pmatrix} I_n \\ X \end{pmatrix} = \begin{pmatrix} I_n \\ X \end{pmatrix} (A - BX). \tag{3.32}$$

in particolare se si considera la soluzione minimale non negativa $X_{\min}$, tale soluzione risulta c-antistabilizzante ovvero la matrice $A - BX_{\min}$ risulta c-antistabile.

Nell'ottica di trasformare la $\mathcal{NARE}$ (3.31) in una $\mathcal{UQME}$ è dapprima opportuno trasformare le prorietà di c-stabilità (c-antistabilità) presenti solitamente in una $\mathcal{NARE}$ in proprietà di d-stabilità (d-antistabilità) più ricorrenti nello studio di una $\mathcal{UQME}$. Si rendono pertanto necessarie le trasforazioni affini e le trasformazioni di Cayley definite nella sezione 1.3.2.

Il primo metodo per trasformare una $\mathcal{NARE}$ in una $\mathcal{UQME}$ è stato proposto da Ramaswami in [51] ed è detto **trasformazione semplice** in quanto non utilizza particolari strutture matriciali.

Si indichi con $\mathcal{F}\colon \Omega \to \mathbb{C}$ una funzione analitica che sia o una trasformazione affine $\mathcal{A}_\alpha$ o una trasformazione di Cayley $\mathcal{C}_\gamma$, definita in un aperto Ω contenente gli autovalori di $\mathcal{H}$, allora per il teorema 1.3.1, vale la relazione

$$\mathcal{F}(\mathcal{H}) \begin{pmatrix} I_n \\ X \end{pmatrix} = \begin{pmatrix} I_n \\ X \end{pmatrix} \mathcal{F}(A - BX). \tag{3.33}$$

Si ponga $K := \mathcal{F}(\mathcal{H})$, $R := \mathcal{F}(A - BX)$ e si consideri la seguente partizione di K:

$$K := \begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix}$$

con $K_{11} \in \mathbb{C}^{n\times n}$.

Si consideri, inoltre, la matrix pencil $\mathcal{P}(z) := K - zI_{n+m}$ e si moltiplichi la seconda colonna a blocchi dei $\mathcal{P}(z)$ per z. Si ottiene, quindi, la matrix pencil quadratica

$$\mathcal{A}(z) := \left(\begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix} - z \begin{pmatrix} I_n & 0 \\ 0 & I_m \end{pmatrix} \right) \begin{pmatrix} I_n & 0 \\ 0 & zI_m \end{pmatrix} = A_0 + zA_1 + z^2 A_2 \tag{3.34}$$

con

$$A_0 := \begin{pmatrix} K_{11} & 0 \\ K_{21} & 0 \end{pmatrix}, \quad A_1 := \begin{pmatrix} -I_n & K_{12} \\ 0 & K_{22} \end{pmatrix}, \quad A_2 := \begin{pmatrix} 0 & 0 \\ 0 & -I_m \end{pmatrix},$$

cui è associata la $\mathcal{UQME}$

$$\mathcal{A}(\mathcal{X}) := A_0 + A_1\mathcal{X} + A_2\mathcal{X}^2 = 0. \tag{3.35}$$

Il risultato che segue esplora la natura degli autovalori della matrix pencil $\mathcal{A}(z)$.

Teorema 3.2.4. *Gli autovalori della matrix pencil quadratica $\mathcal{A}(z)$ definita in* (3.34) *sono*

- 0 *con molteplicità* m,
- λ_i *con* $i = 1, \dots m + n$ *autovalori di* K,
- ∞ *con molteplicità* n.

In particolare, quindi, se la matrice K ha un (n, m) d-splitting, allora gli autovalori di $\mathcal{A}(z)$ presentano un $(m + n, m + n)$ d-splitting.

Dimostrazione. Per la (3.34)

$$\det(\mathcal{A}(z)) = z^m \det(K - zI_{m+n}),$$

dunque la tesi è evidente. □

Tale costruzione, apparentemente artificiosa, è molto importante in quanto esiste una interessante relazione tra le soluzioni X della $\mathcal{NARE}$ (3.31) e le soluzioni $\mathcal{X}$ della $\mathcal{UQME}$ (3.35).

Teorema 3.2.5. *Sia X una soluzione della $\mathcal{NARE}$* (3.31)*. Allora la matrice*

$$\mathcal{X} := \begin{pmatrix} R & 0 \\ X & 0 \end{pmatrix}$$

risolve la $\mathcal{UQME}$ (3.35)*. In particolare, se la matrice R è d-stabile, allora $\mathcal{X}$ è una soluzione d-stabile della $\mathcal{UQME}$.*

Dimostrazione. Poiché X è soluzione della $\mathcal{NARE}$, per la (3.33),

$$\begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix} \begin{pmatrix} I_n \\ X \end{pmatrix} = \begin{pmatrix} I_n \\ X \end{pmatrix} R$$

e quindi

$$\begin{cases} K_{11} + K_{12}X - R = 0 \\ K_{21} + K_{22}X - XR = 0. \end{cases}$$

Svolgendo semplicemente i calcoli si ha

$$\begin{aligned} \mathcal{A}(\mathcal{X}) &= \begin{pmatrix} K_{11} & 0 \\ K_{21} & 0 \end{pmatrix} + \begin{pmatrix} -I_n & K_{12} \\ 0 & K_{22} \end{pmatrix} \begin{pmatrix} R & 0 \\ X & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & -I_m \end{pmatrix} \begin{pmatrix} R^2 & 0 \\ XR & 0 \end{pmatrix} \\ &= \begin{pmatrix} K_{11} - R + K_{12}X & 0 \\ K_{21} + K_{22}X - XR & 0 \end{pmatrix}, \end{aligned}$$

e quindi, per le relazioni, sopra indicate, la $\mathcal{X}$ risolve la $\mathcal{UQME}$ (3.35). □

Un metodo alternativo per trasformare una $\mathcal{NARE}$ in una $\mathcal{UQME}$ e che sfrutta nel contempo particolari strutture matriciali è illustrato in [10] ed è detto **trasformazione basata sulla fattorizzazione UL**.

Si consideri la matrix pencil $\mathcal{P}(z) := N - zK$, dove le matrici N e K sono in forma strutturata standard-I, ovvero

$$N := \begin{pmatrix} N_1 & 0 \\ -N_2 & I_m \end{pmatrix}, \qquad K := \begin{pmatrix} I_n & -K_1 \\ 0 & K_2 \end{pmatrix}.$$

L'obiettivo è determinare delle condizioni sulla matrice hamiltoniana $\mathcal{H}$ affinché la matrix pencil $\mathcal{P}(z)$ sia simile a destra alla matrix pencil $\mathcal{F}(\mathcal{H}) - zI_{m+n}$, dove $\mathcal{F}$ è una funzione analitica definita in un aperto contenente gli autovalori della matrice $\mathcal{H}$. Come nel caso precedente è possibile considere $\mathcal{F}$ come una trasformazione affine $\mathcal{A}_\alpha$ o una trasformazione di Cayley $\mathcal{C}_\gamma$ per opportuni valori di α e γ.

Proposizione 3.2.2. *Si ponga*

$$H := \mathcal{F}(\mathcal{H}) = \begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix},$$

con $H_{11} \in \mathbb{C}^{n\times n}$. Se la matrice H_{22} è invertibile, allora esiste una matrix pencil $\mathcal{P}(z) := N - zK$ in forma strutturata standard-I, tale che la matrix pencil $H - zI_{m+n}$ è simile a destra alla $\mathcal{P}(z)$.

Dimostrazione. Siano N e K tali che $H = K^{-1}N$ e dunque $KH = N$. L'esistenza di tali matrici è assicurata dall'invertibilità di K_{22}, infatti

$$\begin{pmatrix} I_n & -K_1 \\ 0 & K_2 \end{pmatrix} \begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix} = \begin{pmatrix} H_{11} - K_1H_{21} & H_{12} - K_1H_{22} \\ K_2H_{21} & K_2H_{22} \end{pmatrix} = \begin{pmatrix} N_1 & 0 \\ -N_2 & I_m \end{pmatrix},$$

quindi

$$N := \begin{pmatrix} H_{11} - H_{12}H_{22}^{-1}H_{21} & 0 \\ H_{22}^{-1}H_{21} & I_m \end{pmatrix}, \qquad K := \begin{pmatrix} I_n & H_{12}H_{22}^{-1} \\ 0 & H_{22}^{-1} \end{pmatrix},$$

dunque siffatte matrici sono ben definite.

Per provare la tesi è sufficiente oservare che

$$H - zI_{m+n} = K^{-1}N - zI_{m+n} = K^{-1}(N - zK) = K^{-1}\mathcal{P}(z).$$

□

Come nella trasformazione precedente, è possibile trasformare la matrix pencil lineare $\mathcal{P}(z)$ in una matrix pencil quadratica, moltiplicando la seconda riga a blocchi di $\mathcal{P}(z)$ per $-z$, ottenendo quindi la matrix pencil

$$\mathcal{A}(z) := \begin{pmatrix} I_n & 0 \\ 0 & -zI_m \end{pmatrix} \left(\begin{pmatrix} N_1 & 0 \\ -N_2 & I_m \end{pmatrix} - z \begin{pmatrix} I_n & -K_1 \\ 0 & K_2 \end{pmatrix} \right) = A_0 + zA_1 + z^2 A_2 \quad (3.36)$$

con

$$A_0 := \begin{pmatrix} N_1 & 0 \\ 0 & 0 \end{pmatrix}, \quad A_1 := \begin{pmatrix} -I_n & K_1 \\ N_2 & -I_m \end{pmatrix}, \quad A_2 := \begin{pmatrix} 0 & 0 \\ 0 & K_2 \end{pmatrix},$$

e la $\mathcal{UQME}$

$$\mathcal{A}(\mathcal{X}) := A_0 + A_1\mathcal{X} + A_2\mathcal{X}^2. \quad (3.37)$$

È importante evidenziare le relazioni tra gli autovalori della matrix pencil $\mathcal{A}(z)$ e gli autovalori di $\mathcal{F}(\mathcal{H})$.

Teorema 3.2.6. *Gli autovalori della matrix pencil quadratica $\mathcal{A}(z)$ definita in* (3.36) *sono*

- 0 *con molteplicità* m,
- λ_i *con* $i = 1, \dots m+n$ *autovalori di* H,
- ∞ *con molteplicità* n.

In particolare, quindi, se la matrice H ha un (n, m) d-splitting, allora gli autovalori di $\mathcal{A}(z)$ presentano un $(m+n, m+n)$ d-splitting.

Dimostrazione. Per definizione di $\mathcal{A}(z)$ e per la proposizione 3.2.2 si ha

$$\det(\mathcal{A}(z)) = (-1)^m z^m \det(N - zK) = (-1)^m z^m \det(K) \det(H - zI_{m+n}),$$

e quindi la tesi è evidente. □

Anche per le trasformazioni basate sulla fattorizzazione UL è possibile dare una descrizione delle soluzioni della $\mathcal{UQME}$ a partire dalle soluzioni della relativa $\mathcal{NARE}$.

Teorema 3.2.7. *Sia X una soluzione della $\mathcal{NARE}$* (3.31) *e si ponga $R := \mathcal{F}(A - BX)$. Allora la matrice*

$$\mathcal{X} := \begin{pmatrix} R & 0 \\ XR & 0 \end{pmatrix}$$

risolve la $\mathcal{UQME}$ (3.37)*. In particolare, se la matrice R è d-stabile, allora $\mathcal{X}$ è una soluzione d-stabile della $\mathcal{UQME}$.*

Dimostrazione. Poiché X è soluzione della $\mathcal{NARE}$, si ha

$$H \begin{pmatrix} I_n \\ X \end{pmatrix} = \begin{pmatrix} I_n \\ X \end{pmatrix} R,$$

dunque, essendo $H = K^{-1}N$, si ha

$$\begin{pmatrix} N_1 & 0 \\ -N_2 & I_m \end{pmatrix} \begin{pmatrix} I_n \\ X \end{pmatrix} = \begin{pmatrix} I_n & -K_1 \\ 0 & K_2 \end{pmatrix} \begin{pmatrix} I_n \\ X \end{pmatrix} R$$

e pertanto valgono le seguenti relazioni:

$$\begin{cases} N_1 - R + K_1 XR = 0 \\ -N_2 + X - K_2 XR = 0. \end{cases}$$

Da una verifica diretta si ottiene

$$\begin{aligned} \mathcal{A}(\mathcal{X}) &= \begin{pmatrix} N_1 & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} -I_n & K_1 \\ N_2 & -I_m \end{pmatrix} \begin{pmatrix} R & 0 \\ XR & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & K_2 \end{pmatrix} \begin{pmatrix} R^2 & 0 \\ XR^2 & 0 \end{pmatrix} \\ &= \begin{pmatrix} N_1 - R + K_1 XR & 0 \\ N_2 R - XR + K_2 XR^2 & 0 \end{pmatrix}, \end{aligned}$$

quindi $\mathcal{X}$ è effettivamente soluzione della $\mathcal{UQME}$ (3.37). □

3.2.2 Descrizione generale del metodo $\mathcal{CR}$

Il metodo $\mathcal{SDA}$ genera delle successioni di matrix pencil con medesimi sottospazi di deflazione ed autovalori elevati al quadrato ad ogni passo, il metodo $\mathcal{CR}$ si basa sul un principio analogo: genera una successione di matrix pencil quadratiche i cui autovalori vengono elevati al quadrato ad ogni iterazione.

Si consideri la matrix pencil quadratica regolare

$$\mathcal{A}(z) := A_0 + zA_1 + z^2A_2 \tag{3.38}$$

con A_0, A_1, A_2, $X \in \mathbb{C}^{n\times n}$ e A_2 matrice non nulla, e sia

$$\mathcal{A}(X) := A_0 + A_1X + A_2X^2 = 0. \tag{3.39}$$

la relativa $\mathcal{UQME}$.

Vale il seguente fondamentale teorema:

Teorema 3.2.8. *Si supponga che le successioni*

$$\begin{aligned} A_0^{(k+1)} &= -A_0^{(k)}(A_1^{(k)})^{-1}A_0^{(k)} \\ A_1^{(k+1)} &= A_1^{(k)} - A_0^{(k)}(A_1^{(k)})^{-1}A_2^{(k)} - A_2^{(k)}(A_1^{(k)})^{-1}A_0^{(k)} \\ A_2^{(k+1)} &= -A_2^{(k)}(A_1^{(k)})^{-1}A_2^{(k)} \end{aligned} \tag{3.40}$$

con $A_0^{(0)} = A_0$, $A_1^{(0)} = A_1$, $A_2^{(0)} = A_2$ siano ben definite, e si consideri la successione di matrix pencil quadratiche $\{\,\mathcal{A}_k(z)\,\}_{k\in\mathbb{N}}$ definite da

$$\mathcal{A}^{(k)}(z) := A_0^{(k)} + zA_1^{(k)} + z^2A_2^{(k)}. \tag{3.41}$$

Allora se λ è un autovalore di $\mathcal{A}(z)$, λ^{2^k} è un autovalore di $\mathcal{A}^{(k)}(z)$.

Dimostrazione. La tesi è provata per induzione. Per $k = 1$, si osservi che

$$\begin{aligned} -\mathcal{A}(z)A_1^{-1}\mathcal{A}(-z) &= -(A_0 + zA_1 + z^2A_2)A_1^{-1}(A_0 - zA_1 + z^2A_2) \\ &= -A_0A_1^{-1}A_0 + z^2(A_1 - A_0A_1^{-1}A_2 - A_2A_1^{-1}A_0) - z^4A_2A_1^{-1}A_2 \\ &= \mathcal{A}^{(1)}(z^2), \end{aligned}$$

pertanto, portando il determinante a primo e ultimo membro, se λ è un autovalore di $\mathcal{A}(z)$, allora λ^2 è autovalore $\mathcal{A}^{(1)}(z)$.

Si supponga la tesi vera per ogni intero $j \le k$, allora, ripetendo calcoli del tutto analoghi, si ha

$$\begin{aligned} &-\mathcal{A}^k(z)(A_1^{(k)})^{-1}\mathcal{A}^{(k)}(-z) = -(A_0^{(k)} + zA_1^{(k)} + z^2A_2^{(k)})(A_1^{(k)})^{-1}(A_0^{(k)} - zA_1^{(k)} + z^2A_2^{(k)}) \\ &= -A_0^{(k)}(A_1^{(k)})^{-1} + z^2(A_1^{(k)} - A_0^{(k)}(A_1^{(k)})^{-1}A_2^{(k)} - A_2^{(k)}(A_1^{(k)})^{-1}A_0^{(k)}) - z^4A_2^{(k)}(A_1^{(k)})^{-1}A_2^{(k)} \\ &= \mathcal{A}^{(k+1)}(z^2), \end{aligned}$$

quindi, se μ è un autovalore di $\mathcal{A}^{(k)}(z)$, μ^2 è un autovalore di $\mathcal{A}^{(k+1)}(z)$. Applicando l'ipotesi induttiva si ha la tesi. □

Il metodo $\mathcal{CR}$ consiste nel generare le successioni definite nel teorema 3.2.8. Se la matrice $A_1^{(k)}$ risulta non invertibile, allora si dice che il metodo $\mathcal{CR}$ ha un **breakdown** al k-esimo passo. Il metodo di riduzione ciclica risulta particolarmente efficace per la risoluzione di $\mathcal{UQME}$. Vale infatti il seguente teorema:

Teorema 3.2.9. *Sia X una soluzione della $\mathcal{UQME}$ (3.39) e si supponga che il metodo $\mathcal{CR}$ non abbia breakdown. Allora la matrice X^{2^k} risolve la $\mathcal{UQME}$ associata alla matrix pencil* (3.41)*.*

Dimostrazione. La tesi è provata per induzione. Per $k = 1$, sia X una soluzione della $\mathcal{UQME}$, allora

$$\begin{aligned} A_0A_1^{-1}(A_0 + A_1X + A_2X^2) &= 0 \\ -(A_0 + A_1X + A_2X^2)X &= 0 \\ A_2A_1^{-1}(A_0 + A_1X + A_2X^2)X^2 &= 0 \end{aligned}$$

sommando le precedenti equazioni si ottiene

$$A_0A_1^-A_0 + (A_1 - A_0A_1^{-1}A_2 - A_2A_1^{-1}A_0)X^2 + A_2A_1^{-1}A_2X^4 = \mathcal{A}^{(1)}(X^2) = 0$$

da cui la tesi.

Si supponga la tesi vera per ogni $j \leq k$, allora utilizzando i medesimi calcoli svolti sopra (utilizzando $A_i^{(k)}$ in luogo di A_i per $i = 0, 1, 2$ e X^{2^k} in luogo di X) si ottiene

$$\mathcal{A}^{(k+1)}\left(X^{2^{k+1}}\right) = 0$$

e quindi la tesi è provata per ogni valore di k. □

Il teorema che segue, illustra, concretamente, come il metodo $\mathcal{CR}$ può essere utilizzato per risolvere una $\mathcal{UQME}$.

Teorema 3.2.10. *Sia X una soluzione della $\mathcal{UQME}$ (3.39) e si supponga che il metodo $\mathcal{CR}$ non presenti breakdown. Si consideri la successione*

$$\begin{cases} \hat{A}_1^{(0)} = A_1, \\ \hat{A}_1^{(k+1)} = \hat{A}_1^{(k)} - A_2^{(k)}(A_1^{(k)})^{-1}A_0^{(k)}. \end{cases} \tag{3.42}$$

Allora per ogni k sono verificate le relazioni

$$A_0 + \hat{A}_1^{(k)}X + A_2^{(k)}X^{2^k+1} = 0 \tag{3.43}$$

e

$$\hat{A}_1^{(k)} + A_2^{(k)}X^{2^k} = A_1 + A_2X. \tag{3.44}$$

Dimostrazione. Al solito per la dimostrazione si utilizza il principio di induzione. Per $k = 0$, le due relazioni sono banalmente vere, in quanto la prima relazione si riduce a

$$A_0 + A_1X + A_2X^2 = 0,$$

mentre la seconda è semplicemente

$$A_1 + A_2X = A_1 + A_2X = 0.$$

Si suppone che le relazioni siano verificate per ogni valore $j \leq k$, allora per ipotesi induttiva vale

$$A_0 + \hat{A}_1^{(k)}X + A_2^{(k)}X^{2^k+1}$$

dunque, per il teorema 3.2.9

$$\begin{aligned} A_0 + \hat{A}_1^{(k)}X + A_2^{(k)}X^{2^k+1} - A_2^{(k)}(A_1^{(k)})^{-1}(A_0 + A_1^{(k)}X^{2^k} + A_2^{(k)}X^{2^{k+1}})X = \\ = A_0 + \hat{A}_1^{(k+1)}X + A_2^{(k+1)}X^{2^{(k+1)}+1} = 0. \end{aligned}$$

Per quanto riguarda la seconda relazione, si osservi che, applicando l'ipotesi induttiva ed il teorema 3.2.9, valgono le uguaglianze

$$\begin{aligned} \hat{A}_1^{(k+1)} + A_2^{(k+1)}X^{2^{k+1}} &= \hat{A}_1^{(k)} - A_2^{(k)}(A_1^{(k)})^{-1}A_0^{(k)} - A_2^{(k)}(A_1^{(k)})^{-1}A_2^{(k)}X^{2^{k+1}} \\ &= \hat{A}_1^{(k)} - A_2^{(k)}(A_1^{(k)})^{-1}A_0^{(k)} + A_2^{(k)}(A_1^{(k)})^{-1}(A_0^{(k)} + A_1^{(k)}X^{2^k}) \\ &= \hat{A}_1^{(k)} + A_2X^{2^k} = A_1 + A_2X. \end{aligned}$$

□

Sia X una soluzione d-stabile della $\mathcal{UQME}$ (3.39) e si supponga che le successioni $\{ A_2^{(k)} \}_{k\in\mathbb{N}}$ e $\{ (\hat{A}_1^{(k)})^{-1} \}_{k\in\mathbb{N}}$ siano limitate. Allora per la (3.43) si ha

$$(-\hat{A}_1^{(k)})^{-1}A_0 = X + (\hat{A}_1^{(k)})^{-1}A_2^{(k)}X^{2^k+1},$$

e dunque, portando il limite ad entrambi i mebri, si ottiene

$$X = \lim_{k\to\infty} (-\hat{A}_1^{(k)})^{-1}A_0. \tag{3.45}$$

Applicando il teorema 3.2.10, quindi, è possibile individuare una soluzione d-stabile di una $\mathcal{UQME}$, a partire dalle successioni $\{ A_2^{(k)} \}_{k\in\mathbb{N}}$ e $\{ (\hat{A}_1^{(k)})^{-1} \}_{k\in\mathbb{N}}$. È interessante, a questo punto, osservare che se X è una soluzione d-stabile di una $\mathcal{UQME}$ (3.39), allora

$$\limsup_{k\to\infty} \sqrt[2^k]{X + (\hat{A}_1^{(k)})^{-1}A_0} \le \limsup_{k\to\infty} \sqrt[2^k]{(\hat{A}_1^{(k)})^{-1}A_2^{(k)}X^{2^k+1}} \le \varrho(X) < 1.$$

La precedente disequazione mostra, dunque, che il metodo $\mathcal{CR}$, sotto ipotesi di limitezza delle successioni $\{ A_2^{(k)} \}_{k\in\mathbb{N}}$ e $\{ (\hat{A}_1^{(k)})^{-1} \}_{k\in\mathbb{N}}$, ha convergenza quadratica ad una soluzione d-stabile X.

Il teorema che segue rende più precise le stime sopra menzionate. Per la dimostrazione e per ulteriori risultati riguardanti la convergenza del metodo $\mathcal{CR}$ sono presentati in [10].

Teorema 3.2.11. *Si supponga che gli autovalori λ_i con $i = 1, \dots 2n$, della matrix pencil quadratica $\mathcal{A}(z)$ (3.38) abbiano un (n, n) d-splitting forte, ovvero verifichino*

$$|\lambda_{2n}| \le \cdots \le |\lambda_{n+1}| < 1 < |\lambda_n| \le \cdots \le |\lambda_1|,$$

e che il metodo $\mathcal{CR}$ non abbia breakdown. Sia X una soluzione d-stabile della $\mathcal{UQME}$ (3.39), allora

$$\lim_{k\to\infty} A_0^{(k)} = 0 \qquad \lim_{k\to\infty} A_2^{(k)} = 0,$$

inoltre sono verificate le stime

$$\limsup_{k\to\infty} \sqrt[2^k]{\|A_0^{(k)}\|} \le |\lambda_{n+1}|, \tag{3.46}$$

$$\limsup_{k\to\infty} \sqrt[2^k]{\|A_2^{(k)}\|} \le \frac{1}{|\lambda_n|}, \tag{3.47}$$

$$\limsup_{k\to\infty} \sqrt[2^k]{\|X + (\hat{A}_1^{(k)})^{-1}A_0\|} \le \frac{|\lambda_{n+1}|}{|\lambda_n|}. \tag{3.48}$$

Dopo aver introdotto le generalità del metodo $\mathcal{CR}$ ed aver enunciato i principali risultati di convergenza, è opportuno ossorvere quali relazioni intercorrono tra il metodo $\mathcal{CR}$ ed il metodo $\mathcal{SDA}$. Come sarà esposto nella prossima sezione, tali relazioni si rivelano particolarmente importanti per risolvere $\mathcal{UQME}$ definite a partire da una $\mathcal{NARE}$.

Teorema 3.2.12. *Si consideri la matrix pencil quadratica*

$$\mathcal{A}(z) := \begin{pmatrix} E & 0 \\ 0 & 0 \end{pmatrix} + z \begin{pmatrix} -I_n & G \\ P & -I_m \end{pmatrix} + z^2 \begin{pmatrix} 0 & 0 \\ 0 & F \end{pmatrix}. \tag{3.49}$$

Allora il metodo $\mathcal{CR}$ definito a partire dalla matrix pencil $\mathcal{A}(z)$ genera la successione di matrix pencil quadratiche $\{ \mathcal{A}^{(k)}(z) \}_{k\in\mathbb{N}}$ date da

$$\mathcal{A}^{(k)}(z) := \begin{pmatrix} E_k & 0 \\ 0 & 0 \end{pmatrix} + z \begin{pmatrix} -I_n & G_k \\ P_k & -I_m \end{pmatrix} + z^2 \begin{pmatrix} 0 & 0 \\ 0 & F_k \end{pmatrix},$$

dove la successione di matrici $\{ E_k \}_{k\in\mathbb{N}}$, $\{ F_k \}_{k\in\mathbb{N}}$, $\{ G_k \}_{k\in\mathbb{N}}$, $\{ P_k \}_{k\in\mathbb{N}}$ sono le successioni definite dal metodo $\mathcal{SDA}$ con valori iniziali E, F, G, P.

Dimostrazione. Al solito la dimostrazione segue dal principio di induzione. Per $k = 0$ la tesi è banalmente vera. Si supponga, quindi, che la tesi sia verficata per ogni intero $j \leq k$, allora

$$(A_1^{(k)})^{-1} = \begin{pmatrix} -I_n & G_k \\ P_k & -I_m \end{pmatrix}^{-1} = \begin{pmatrix} (G_kP_k - I_n)^{-1} & (G_kP_k - I_n)^{-1}G_k \\ (P_kG_k - I_m)^{-1}P_k & (P_kG_k - I_m)^{-1} \end{pmatrix},$$

da cui

$$\begin{aligned}
A_0^{(k)} &= -\begin{pmatrix} E_k & 0 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} -I_n & G_k \\ P_k & -I_m \end{pmatrix}^{-1}\begin{pmatrix} E_k & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} -E_k(G_kP_k - I_n)^{-1}E_k & 0 \\ 0 & 0 \end{pmatrix}, \\
A_1^{(k)} &= \begin{pmatrix} -I_n & G_k \\ P_k & -I_m \end{pmatrix} - \begin{pmatrix} E_k & 0 \\ 0 & 0 \end{pmatrix}(A_1^{(k)})^{-1}\begin{pmatrix} 0 & 0 \\ 0 & F_k \end{pmatrix} - \begin{pmatrix} 0 & 0 \\ 0 & F_k \end{pmatrix}(A_1^{(k)})^{-1}\begin{pmatrix} E_k & 0 \\ 0 & 0 \end{pmatrix} = \\
&= \begin{pmatrix} -I_n & G_k + E_k(I_n - G_kP_k)^{-1}G_kF_k \\ P_k + F_k(I_m - P_kG_k)^{-1}P_kE_k & -I_m \end{pmatrix}, \\
A_2^{(k)} &= -\begin{pmatrix} 0 & 0 \\ 0 & F_k \end{pmatrix}\begin{pmatrix} -I_n & G_k \\ P_k & -I_m \end{pmatrix}^{-1}\begin{pmatrix} 0 & 0 \\ 0 & F_k \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & -F_k(P_kG_k - I_n)^{-1}F_k \end{pmatrix}.
\end{aligned}$$

Dalle precedenti relazioni segue immediatamente che la matrix pencil $\mathcal{A}^{(k)}(z)$ è nella forma voluta e che le successioni $\{E_k\}_{k\in\mathbb{N}}$, $\{F_k\}_{k\in\mathbb{N}}$, $\{G_k\}_{k\in\mathbb{N}}$, $\{P_k\}_{k\in\mathbb{N}}$ coincidono con quelle introdotte nel teorema 3.1.2. □

Il teorema appena dimostrato illustra, quindi, che il metodo $\mathcal{SDA}$ risolve una specifica istanza del metodo $\mathcal{CR}$ applicato ad una matrix pencil quadratica della forma (3.49).

3.2.3 Metodo $\mathcal{CR}$ per $\mathcal{NARE}$

Nella sezione 3.2.1 si è mostrato che risolvere una $\mathcal{NARE}$ è sostanzialmente equivalente a risolvere una particolare $\mathcal{UQME}$. Utilizzando il metodo $\mathcal{CR}$, dunque, è possibile determinare le soluzioni di tale $\mathcal{UQME}$ e quindi risalire alle corrispondenti soluzioni della $\mathcal{NARE}$.

Si consideri la $\mathcal{NARE}$

$$C + XA + DX - XBX = 0, \tag{3.50}$$

e si supponga che la matrice dei coefficienti $\mathcal{M}$ sia una M-matrice non singolare o singolare irriducibile. Allora, per quanto mostrato nella sezione 1.4.3, la matrice hamiltoniana $\mathcal{H}$ presenta un (m, n) c-splitting. Per poter applicare correttamente le trasformazioni introdotte nella sezione 3.2.1, è neccassario dapprima individuare una funzione analitica $\mathcal{F}$ definita in un aperto contenente gli autovalori di $\mathcal{H}$, tale che la matrice $\mathcal{F}(\mathcal{H})$ presenti un (n, m) d-splitting. In tal caso, infatti, per i teoremi 3.2.4 e 3.2.6, le matrix pencil quadratiche oggetto di studio hanno un $(m + n, m + n)$ d-splitting, dunque hanno le proprietà spettrali che garantiscono la convergenza dal metodo $\mathcal{CR}$.

Utilizzando le medesime argomentazioni della sezione 3.1.2, si ha che tra le funzioni analitiche che verificano le proprietà di cui sopra vi sono

- le trasformazioni affini $\mathcal{A}_\alpha$ dove $\alpha \leq \frac{1}{\delta}$, con $\delta := \max_{i=1,\dots,n} a_{ii}$,
- le trasformazioni di Cayley $\mathcal{C}_\gamma$ con $\gamma \geq 0$,

con a_{ii} elementi diagonali della matrice A.

Il primo metodo per trasformare una $\mathcal{NARE}$ in una $\mathcal{UQME}$ è la trasformazione semplice. Sia, dunque $\mathcal{F}$ una funzione analitica che trasformi opportunamente gli autovalori della matrice $\mathcal{H}$ e sia $K := \mathcal{F}(\mathcal{H})$, la trasformazione semplice della $\mathcal{NARE}$ (3.50) genera la matrix pencil quadratica

$$\mathcal{A}(z) := \begin{pmatrix} R_1 & 0 \\ R_2 & 0 \end{pmatrix} + z\begin{pmatrix} -I_n & R_3 \\ R_4 & R_5 \end{pmatrix} + z^2\begin{pmatrix} 0 & 0 \\ 0 & R_6 \end{pmatrix} \tag{3.51}$$

con

$$R_1 := K_{11}, \quad R_2 := K_{21}, \quad R_3 := K_{12}, \quad R_4 := 0, \quad R_5 := K_{22}, \quad R_6 := -I_m.$$

La precedente matrix pencil è legata alla $\mathcal{NARE}$ (3.50) dal seguente teorema:

Teorema 3.2.13. *Si indichi con $\mathcal{F}$ o una trasformazione affine $\mathcal{A}_\alpha$ o una trasformazione di Cayley $\mathcal{C}_\gamma$ con α e γ che verificano le disuguaglianze sopra elencate. Sia $X_{\min}$ la soluzione minimale non negativa della $\mathcal{NARE}$ (3.50) e si ponga $R := \mathcal{F}(A - BX_{\min})$. Allora la matrice*

$$\mathcal{X} := \begin{pmatrix} R & 0 \\ X_{\min} & 0 \end{pmatrix}$$

è la soluzione d-stabile (d-debolmente stabile) della $\mathcal{UQME}$ associata alla matrix pencil (3.51).

Dimostrazione. Si osservi che per il teorema 3.2.5 la matrice $\mathcal{X}$ è soluzione della $\mathcal{UQME}$, inoltre, essendo $A - BX_{\min}$ c-antistabile (c debolmente-antistabile), la matrice $R = \mathcal{F}(A - BX_{\min})$ e quindi anche la matrice $\mathcal{X}$ risultano d-stabili (d-debolmente stabili). Per il teorema 3.2.4 la matrix pencil (3.51) presenta un $(m+n, m+n)$ d-splitting e quindi per il teorema 3.2.1 si ha che $\mathcal{X}$ è l'unica soluzione d-stabile (d-debolmente stabile) della $\mathcal{UQME}$. □

È importante osservare che la particolare struttura della matrix pencil (3.51) si 'preserva' nelle iterazioni del metodo $\mathcal{CR}$. Vale infatti il seguente teorema, la cui verifica si basa esclusivamente sulle equazioni per ricorrenza definite nel teorema 3.2.8.

Teorema 3.2.14. *Il metodo $\mathcal{CR}$ con matrix pencil iniziale* (3.51) *genera la successione di matrix pencil $\{\, \mathcal{A}^{(k)}(z) \,\}_{k\in\mathbb{N}}$ data da*

$$\mathcal{A}^{(k)}(z) := \begin{pmatrix} R_1^{(k)} & 0 \\ R_2^{(k)} & 0 \end{pmatrix} + z \begin{pmatrix} -I_n & R_3^{(k)} \\ R_4^{(k)} & R_5^{(5)} \end{pmatrix} + z^2 \begin{pmatrix} 0 & 0 \\ 0 & R_6^{(k)} \end{pmatrix},$$

con i blocchi matriciali $R_i^{(k)}$ con $i = 1, \ldots, 6$ definiti dalle relazioni

$$\begin{aligned} R_1^{(k+1)} &:= -R_1^{(k)} X^{(k)}, & R_2^{(k+1)} &:= -R_2^{(k)} X^{(k)}, & R_3^{(k+1)} &:= R_3^{(k)} - R_1^{(k)} T^{(k)}, \\ R_4^{(k+1)} &:= R_4^{(k)} - R_6^{(k)} Y^{(k)}, & R_5^{(k+1)} &:= R_5^{(k)} - R_2^{(k)} T^{(k)}, & R_6^{(k+1)} &:= -R_6^{(k)} Z^{(k)}, \end{aligned}$$

con

$$\begin{aligned} S^{(k)} &:= R_5^{(k)} + R_4^{(k)} R_3^{(k)}, & Y^{(k)} &:= (S^{(k)})^{-1}(R_2^{(k)} + R_4^{(k)} R_1^{(k)}), \\ X^{(k)} &:= R_3^{(k)} Y^{(k)} - R_1^{(k)}, & Z^{(k)} &:= (S^{(k)})^{-1} R_6^{(k)}, & T^{(k)} &:= R_3^{(k)} Z^{(k)}. \end{aligned}$$

Inoltre la successione $\{\, \hat{A}_1^{(k)} \,\}_{k\in\mathbb{N}}$ definita in 3.42 è definita da

$$\hat{A}_1^{(k)} := \begin{pmatrix} -I_n & R_3 \\ R_4^{(k)} & R_5 \end{pmatrix},$$

dunque solo un blocco della matrice $\hat{A}_1^{(k)}$ dipende dai blocchi calcolati alla k-esima iterazione.

I blocchi della matrix pencil iniziale $\mathcal{A}(z)$, dipendono dal tipo di funzione analitica utilizzata per trasformare la matrice hamiltoniana. Se si utilizza una trasformazione affine $\mathcal{A}_\alpha$ si ottiene

$$\begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix} := \mathcal{A}_\alpha \left(\begin{pmatrix} A & -B \\ -C & -D \end{pmatrix} \right) = \begin{pmatrix} \alpha A - I_n & -\alpha B \\ -\alpha C & -\alpha D - I_m \end{pmatrix},$$

da cui

$$R_1 = \alpha A - I_n, \qquad R_2 = -\alpha C, \qquad R_3 = -\alpha B,$$
$$R_4 = 0, \qquad R_5 = -\alpha D - I_m, \qquad R_6 = -I_m.$$

A partire da tele scelta iniziale delle matrici R_i con $i = 1, \ldots, 6$, il seguente teorema, illustrato in [10], mostra che il metodo $\mathcal{CR}$ è numericamente stabile e non presenta brakdown:

Teorema 3.2.15. *Si supponga che la matrice dei coefficienti $\mathcal{M}$ associata alla $\mathcal{NARE}$ (3.50) sia una M-matrice non singolara o singolare irriducibile. Allora il metodo $\mathcal{CR}$ ottenuto a partire dalle trasformazioni $\mathcal{A}_\alpha$ con $\alpha \leq \frac{1}{\delta}$ genera senza brakdown le successioni di matrici $\{ R_i^{(k)} \}_{k\in\mathbb{N}}$ con $i = 1, \ldots, 6$. Inoltre valgono le relazioni*

$$R_1^{(k)} \leq 0, \quad R_2^{(k)} \geq 0, \quad R_3^{(k)} \leq 0, \quad R_4^{(k)} \leq 0, \quad R_6^{(k)} \leq 0,$$

e le matrici $-S^{(k)}$ e $-R_5^{(k)}$ risultano M-matrici non singolari.

Utilizzando, invece, una trasformazione di Cayley $\mathcal{C}_\gamma$, la matrice K è definita da

$$\begin{pmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{pmatrix} := \mathcal{C}_\gamma(\mathcal{H}) = (\mathcal{H} + \gamma I_{m+n})^{-1}(\mathcal{H} - \gamma I_{m+n}) = I_{m+n} - 2\gamma(\mathcal{H} + \gamma I_{m+n})^{-1}$$
$$= \begin{pmatrix} I_n - 2\gamma W^{-1} & -2\gamma(A + \gamma I_n)^{-1} B V^{-1} \\ -2\gamma(-D + \gamma I_m)^{-1} C W^{-1} & I_m - 2\gamma V^{-1} \end{pmatrix},$$

con

$$W := A + \gamma I_n - B(-D + \gamma I_m)^{-1} C \qquad V := -D + \gamma I_m - C(A + \gamma I_n)^{-1} B,$$

dunque, per la (3.51), si ha

$$R_1 := I_n - 2\gamma W^{-1}, \quad R_2 := -2\gamma(-D + \gamma I_m)^{-1} C W^{-1}, \quad R_3 := -2\gamma(A + \gamma I_n)^{-1} B V^{-1},$$
$$R_4 := 0, \qquad R_5 := I_m - 2\gamma V^{-1}, \qquad R_6 := -I_m.$$

Contrariamente a quanto mostrato per le trasformazioni affini, il metodo $\mathcal{CR}$ ottenuto a partire dalle trasformazioni di Cayley non preservano e particolari proprietà di non negatività delle matrici $A_i^{(k)}$ con $i = 0, 1, 2$.

Per entrambe le trasformazioni, invece, valgono le seguenti proprietà di convergenza

Teorema 3.2.16. *Si supponga che la matrice dei coefficienti $\mathcal{M}$ associata alla $\mathcal{NARE}$ (3.50) sia una M-matrice non singolare o singolare irriducibile con drift μ non nullo. Si supponga inoltre che il metodo $\mathcal{CR}$ non presenti breakdown e che la matrice $R_5 + R_4^{(k)} R_3$ sia non singolare. Si indichi con $\mathcal{F}$ una trasformazione affine $\mathcal{A}_\alpha$ con $\alpha \leq \frac{1}{\delta}$ o una trasformazione di Cayley con $\gamma > 0$ e si definisca la successione $\{ X_k \}_{k\in\mathbb{N}}$ data da*

$$X_k := -(R_5 + R_4^{(k)} R_3)^{-1}(R_2 + R_4^{(k)} R_1).$$

Allora

$$\lim_{k\to\infty} X_k = X_{\min},$$
$$\limsup_{k\to\infty} \sqrt[2^k]{\|X_{\min} - X_k\|} \leq \frac{\omega_1}{\omega_2}$$

dove $X_{\min}$ è la soluzione minimale non negativa della $\mathcal{NARE}$ (3.50), $\omega_1 = |\mathcal{F}(\lambda_n)|$ e $\omega_2 = |\mathcal{F}(\lambda_{n+1})|$, con

$$Re(\lambda_{m+n}) \leq \ldots Re(\lambda_{n+1}) \leq 0 \leq Re(\lambda_n) \leq \ldots Re(\lambda_1)$$

autovalori della matrice hamiltoniana $\mathcal{H}$.

Dimostrazione. Per il teorema 3.2.13 la matrice

$$\mathcal{X} := \begin{pmatrix} R & 0 \\ X_{\min} & 0 \end{pmatrix}$$

è la soluzione d-stabile (d-debolmente stabile) della $\mathcal{UQME}$, pertanto, per il teorema 3.2.11, vale la stima

$$\limsup_{k\to\infty} \sqrt[2^k]{\|\mathcal{X} + (\hat{A}_1^{(k)})^{-1}A_0\|} \leq \frac{|\mathcal{F}(\lambda_n)|}{|\mathcal{F}(\lambda_{n+1})|}.$$

Risulta, dunque $\lim_{k\to\infty} -(\hat{A}_1^{(k)})A_0 = \mathcal{X}$, pertanto dalle definizioni di $\hat{A}_1^{(k)}$ e A_0 si ha

$$\begin{aligned} X_{\min} &= \lim_{k\to\infty} \begin{pmatrix} 0 & I_m \end{pmatrix} \left(-(\hat{A}_1^{(k)})A_0\right) \begin{pmatrix} I_n \\ 0 \end{pmatrix} \\ &= -\lim_{k\to\infty} \begin{pmatrix} (R_5 + R_4^{(k)}R_1)^{-1}R_4^{(k)} & (R_5 + R_4^{(k)}R_1)^{-1} \end{pmatrix} \begin{pmatrix} R_1 \\ R_2 \end{pmatrix} = \\ &= -(R_5 + R_4^{(k)}R_1)^{-1}(R_4^{(k)}R_1 + R_2), \end{aligned}$$

da cui la tesi. □

Utilizzando le trasformazioni basate sulla fattorizzazione UL, invece, per quanto illustrato nela sezione 3.2.1, si avrebbe una matrix pencil quadratica del tipo

$$\mathcal{A}(z) := \begin{pmatrix} N_1 & 0 \\ 0 & 0 \end{pmatrix} + z \begin{pmatrix} -I_n & -K_1 \\ N_2 & -I_m \end{pmatrix} + z^2 \begin{pmatrix} 0 & 0 \\ 0 & K_2 \end{pmatrix},$$

con $\mathcal{F}(\mathcal{H}) = K^{-1}N$, dove $\mathcal{F}$ indica o una trasformazione affine $\mathcal{A}_\alpha$ con $\alpha \leq \frac{1}{\delta}$ o una trasformazione affine $\mathcal{C}_\gamma$ con $\gamma > 0$. Si osservi che la precedente matrix pencil quadratica rientra nelle ipotesi del teorema 3.2.12, pertanto per le trasformazioni basate sulla fattorizzazione UL, il metodo $\mathcal{CR}$ si riconduce al metodo $\mathcal{SDA}$ trattato nella sezione 3.1.1.

Le strategie adottate per risolvere le $\mathcal{NARE}$ possono ovviamente essere adattate alle $\mathcal{CARE}$, ma tali tecniche non sfruttano le proprietà di hermitianità dei coefficienti delle $\mathcal{CARE}$, pertanto, contrariamente a quanto accade per il metodo $\mathcal{SDA}$, per il metodo $\mathcal{CR}$ non vi sono differenze tra l'algoritmo per le $\mathcal{NARE}$ ed il corrispettivo per le $\mathcal{CARE}$.

3.2.4 Metodo $\mathcal{CR}$ per una particolare $\mathcal{DARE}$

Si consideri una particolare $\mathcal{DARE}$ nella forma

$$X + C^*X^{-1}C = Q, \tag{3.52}$$

con X, C, $Q \in \mathbb{C}^{n\times n}$ e Q matrice hermitiana. Al solito, le soluzioni di maggior interesse nelle applicazioni sono le soluzioni estremali X^- e X^+, la cui esistenza è assicurata dal seguente teorema ([20]):

Teorema 3.2.17. *Si consideri la funzione razionale matriciale*

$$\Phi(\lambda) := \lambda C + Q + \lambda^{-1}C^*, \tag{3.53}$$

definita sul cerchio unitario del piano complesso S^1 e si supponga che sia regolare, ovvero che esista almeno un valore λ tale che $\Phi(\lambda) \neq 0$. Allora la $\mathcal{DARE}$ (3.52) *ammette una soluzione X definita positiva se e solo se $\Phi(\lambda) \succeq 0$ per ogni $\lambda \in S^1$. Inoltre, se l'equazione* (3.52) *ammette una soluzione definita positiva, allora ammette anche soluzioni estremali X_- e X_+.*

Si consideri la $\mathcal{DARE}$

$$Y + CY^{-1}C^* = Q, \tag{3.54}$$

e si osservi che, se la matrice C è invertibile e Y è soluzione della $\mathcal{DARE}$ 3.54 allora $X = Q - Y$ è soluzione della (3.52). Si avrebbe, infatti, $CY^{-1}C^* = X$ da cui

$$C^{-1}XC^{-*} = (Q - X)^{-1}$$

invertendo entrambi i membri della precedente relazione, si ottiene

$$X + C^*X^{-1}C = Q.$$

Sia ora X una soluzione della $\mathcal{DARE}$ (3.52), allora

$$-I_n + QX^{-1} - C^*X^{-1}CX^{-1} = 0,$$

ponendo, quindi, $G := X^{-1}C$, si ottiene che la matrice G risolve la $\mathcal{UQME}$

$$-C + QG - C^*G^2 = 0. \tag{3.55}$$

Allo stesso modo, se Y è una soluzione della $\mathcal{DARE}$ (3.54), allora la matrice $H := Y^{-1}C^*$ risolve la $\mathcal{UQME}$

$$-C^* + QH - CH^2 = 0. \tag{3.56}$$

Dunque si è ricondotto lo studio di due $\mathcal{DARE}$ all'analisi di due $\mathcal{UQME}$, importante è pertanto evidenziare le proprietà spettrali delle soluzioni delle $\mathcal{UQME}$ (3.55) e (3.56). Si considerino le matrix pencil quadratiche

$$\begin{aligned}\mathcal{A}(z) &:= -C + zQ - z^2C^*,\\ \mathcal{B}(z) &:= -C^* + zQ - z^2C,\end{aligned}$$

e si osservi che se λ è un autovalore di $\mathcal{A}(z)$, allora $\bar{\lambda}$ è un autovalore di $\mathcal{B}(z)$, infatti

$$\det(\mathcal{A}(\lambda)) = \det(\mathcal{A}(\lambda)^*) = \det(-C^* + \bar{\lambda}Q - \bar{\lambda}^2C) = \det(\mathcal{B}(\bar{\lambda}) = 0,$$

Vale, inoltre, il seguente fondamentale teorema ([20]):

Teorema 3.2.18. *Sia X^+ la soluzione massimale della $\mathcal{UQME}$ (3.52) e $\Phi(\cdot)$ la funzione razionale matriciale (3.53). Allora $\varrho(X_+^{-1}C) < 1$ se e solo se $\Phi(\lambda) \succ 0$ per ogni $\lambda \in S^1$, dove $\varrho(\cdot)$ indica il raggio spettrale di una matrice.*

Per il teorema precedente e per le osservazioni sopra menzionate, se, dunque, $\Phi(\lambda) \succ 0$ per ogni $\lambda \in S^1$, la matrici $G_+ := X_+^{-1}C$ e $H_+ := Y_+^{-1}C^*$ risultano rispettivamente soluzioni d-stabili delle $\mathcal{UQME}$ (3.55) e (3.56). Inoltre se la matrice C è non singolare, è possibile individuare la soluzione minimale X_- mediante la relazione $X_- = Q - Y_+$.

Sotto le ipotesi del teorema 3.2.18, è possibile applicare il metodo di riduzione ciclica alle $\mathcal{UQME}$ (3.55) e (3.56) e generare le successioni $\{ C_k \}_{k\in\mathbb{N}}$, $\{ Q_k \}_{k\in\mathbb{N}}$, $\{ X_k \}_{k\in\mathbb{N}}$ e $\{ Y_k \}_{k\in\mathbb{N}}$ definite da

$$\begin{aligned}C_{n+1} &= C_nQ_n^{-1}C_n,\\ Q_{n+1} &= Q_n - C_nQ_n^{-1}C_n^* - C_n^*Q_n^{-1}C_n,\\ X_{n+1} &= X_n - C_n^*Q_n^{-1}C_n,\\ Y_{n+1} &= Y_n - C_nQ_n^{-1}C_n^*.\end{aligned} \tag{3.57}$$

Per quanto riguarda la convergenza vale il seguente risultato dimostrato in [48].

Teorema 3.2.19. *Le matrici Q_n, X_n, Y_n per $n \geq 0$ sono definite positive e valgono le relazioni*

$$0 \prec Q_{n+1} \preceq Q_n, \quad 0 \prec X_{n+1} \preceq X_n, \quad 0 \prec Y_{n+1} \preceq Y_n.$$

Inoltre se $\Phi(\lambda) \succ 0$ *per ogni* $\lambda \in S^1$*, allora per ogni* $\varepsilon > 0$ *e per ogni norma matriciale* $\|\cdot\|$ *vale*

$$\|I_n - X_n X_+^{-1}\| = O\left((\omega+\varepsilon)^{2^{n+1}}\right), \quad \|I_n - Y_n Y_+^{-1}\| = O\left((\omega+\varepsilon)^{2^{n+1}}\right),$$
$$\|C_n\| = O\left((\omega+\varepsilon)^{2^{n+1}}\right),$$

con $\omega = \varrho(X_+^{-1}C)$.

3.3 Implementazioni

Nel presente paragrafo vengono tradotti in codice gli algoritmi illustrati nelle pagine precedenti. Sono dapprima analizzate le proprietà numeriche di ciascun metodo, sono dunque descritti alcuni esempi di applicazione dei suddetti algoritmi, i quali vengono comparati per numero di iterazioni richieste, tempo di impiego della CPU ed errore relativo generato. Le sperimentazioni sono eseguite in MATLAB 2008*b*.

3.3.1 Algoritmi per $\mathcal{NARE}$

Per quanto mostrato nei paragrafi precedenti, per risolvere una $\mathcal{NARE}$ avente come matrice dei coefficienti una M-matrice non singolare o singolare irriducibile, i doubling algorithm più idonei sono i seguenti

- il metodo $\mathcal{SDA}$ i cui termini iniziali sono ottenuti mediante una trasformazione affine,
- il metodo $\mathcal{SDA}$ i cui termini iniziali sono ottenuti mediante una trasformazione di Cayley,
- il metodo $\mathcal{CR}$ applicato ad una $\mathcal{UQME}$ ottenuta attraverso una trasformazione semplice di una $\mathcal{NARE}$ basata su una trasformazione affine;
- il metodo $\mathcal{CR}$ applicato ad una $\mathcal{UQME}$ ottenuta attraverso una trasformazione semplice di una $\mathcal{NARE}$ basata su una trasformazione di Cayley.

I codici 3.1, 3.2, 3.3 traducono in 'linguaggio macchina' i metodi presentati nella sezione 3.1.2, in particolare il codice 3.1 calcola le successioni di matrici generate dal metodo $\mathcal{SDA}$, con criteri d'arresto il numero di iterazioni k (posto 30 come limite superiore) e la norma matriciale delle matrici E_k e F_k, che, come noto, tendono a zero.

Listing 3.1: Metodo $\mathcal{SDA}$.

```
function [X,Y] = sda(E,F,G,P)

% il codice applica il metodo SDA a partire dalle matrici E,F,G,P.
% X = limite della successione (P_k)
% Y = limite della successione (G_k)


tol = 1e-13;
kmax = 30;
err = 1;
k = 0;
n = size(G,1);
m = size(P,1);

while err > tol && k < kmax
    M1 = eye(n) - G*P;
    M2 = eye(m) - P*G;
    E1 = E/M1;
```

```
    F1 = F/M2;
    G = G + E1*G*F;
    P = P + F1*P*E;
    E = E1*E;
    F = F1*F;
    err = min(norm(E,1),norm(F,1));
    k = k + 1;
end
X = P;
Y = G;
if k == kmax
    disp('Warning: raggiunto il massimo numero di iterazioni')
end
```

Il codice 3.2 calcola, seguendo le indicazioni della sezione 3.1.2, i termini iniziali del metodo $\mathcal{SDA}$ adoperando una trasformazione affine, si osservi la particolare scelta del parametro a.

Listing 3.2: Metodo $\mathcal{SDA}$ per $\mathcal{NARE}$ con trasformazione affine.

```
function [X,Y] = sda_nare_aff(A,B,C,D)

% il codice risolve la NARE
%        C + XA + DX - XBX = 0
% utilizzando il metodo SDA a partire da una trasformazione affine.
% X = soluzione minimale NARE
% Y = soluzione minimale NARE duale


a = 1/max(diag(A));
n = size(A,1);
m = size(D,1);

% calcolo coefficienti iniziali sda mediante trasformazione affine
IDa = inv(eye(m) + a*D);
F = -IDa;
G = a*B*IDa;
P = -a*IDa*C;
E = a*A - eye(n) + a*G*C;

% implementazione SDA
[X,Y] = sda(E,F,G,P);
```

Il codice 3.3 calcola, invece, i termini iniziali del metodo $\mathcal{SDA}$ ottenuti a partire da una trasformazione di Cayley, anche in questo caso, è importante osservare come si è scelto il paramentro g.

Listing 3.3: Metodo $\mathcal{SDA}$ per $\mathcal{NARE}$ con trasformazione di Cayley.

```
function [X,Y] = sda_nare_cay(A,B,C,D)

% il programma risolve la NARE
%        C + XA + DX - XBX = 0
% utilizzando il metodo SDA a partire da una trasformazione Cayley.
% X = soluzione minimale NARE
% Y = soluzione minimale NARE duale


g = max(max(diag(A)),max(diag(D)));
n = size(A,1);
m = size(D,1);
```

```
% calcolo coefficienti iniziali sda mediante trasformazione affine
U = [A + g*eye(n), -B ; C, D + g*eye(m)];
V = [A - g*eye(n), -B ; C, D - g*eye(m)];
W = U\V;
E = W(1:n,1:n);
G = -W(1:n,n+1:n+m);
P = -W(n+1:n+m,1:n);
F = W(n+1:n+m,n+1:n+m);

% implementazione SDA
[X,Y] = sda(E,F,G,P);
```

Le proprietà numeriche dei metodi $\mathcal{SDA}$ per la risoluzione di una $\mathcal{NARE}$ avente come matrice dei coefficienti una M-matrice non singolare o singolare irriducibile, possono riassumersi come segue

costo computazionale nel caso $n = m$, per il metodo $\mathcal{SDA}$ si rendono necessarie:

- per l'inizializzazione delle matrici ottenute con una trasformazione affine $8n^3$ operazioni elementari dovute a
  - una inversione di una matrice n-dimensionale,
  - tre prodotti matriciali di matrici di dimensione $n \times n$,
- per l'inizializzazione delle matrici ottenute con una trasformazione di Cayley $\frac{64}{3}n^3$ operazioni elementari dovute alla risoluzione di un sistema lineare di dimensione $2n$,
- per una singola iterazione del metodo $\mathcal{SDA}$ circa $\frac{64}{3}n^3$ operazioni elementari dovute a
  - una risoluzione di un sistema lineare n-dimensionale,
  - otto prodotti matriciali di matrici di dimensione $n \times n$.

velocità di convergenza Per i teoremi 3.1.6 e 3.1.7 il metodo $\mathcal{SDA}$ presenta convergenza quadratica, inoltre, se viene utilizzata una trasformazione affine, il raggio di convergenza delle successioni è

$$\Delta_\alpha = \left| \frac{\alpha\lambda_n - 1}{\alpha\lambda_{n+1} - 1} \right|,$$

se viene, invece adoperata una trasformazione di Cayley, il raggio di convergenza è

$$\Delta_\gamma = \left| \frac{(\lambda_n - \gamma)(\lambda_{n+1} + \gamma)}{(\lambda_n + \gamma)(\lambda_{n+1} - \gamma)} \right|.$$

stabilità numerica La stabilità dei metodi è fortemente condizionata dal drift della $\mathcal{NARE}$. Per una $\mathcal{NARE}$ ricorrente nulla, si rendono necessarie tecniche di shift.

I codici di seguito utilizzano le argomentazioni esposte nella sezione 3.2.3 per la risoluzione di $\mathcal{NARE}$ con matrice dei coefficienti una M-matrice non singolare o singolare irriducibile. In particolare, il codice 3.4 utilizza la riduzione ciclica per risolvere la $\mathcal{UQME}$ ottenuta dalla $\mathcal{NARE}$ oggetto di studio mediante una trasformazione semplice.

Listing 3.4: Metodo $\mathcal{CR}$ per $\mathcal{UQME}$ ottenute da $\mathcal{NARE}$.

```
function [X] = cr(R1,R2,R3,R4,R5,R6)

% il codice applica il metodo CR alla UQME
%   |R1 0| + z|-I R3| + z^2|0 0 |
%   |R2 0|     |R4 R5|       |0 R6|
%
% ottenuta da una NARE.
```

```
tol = 1e-13;
kmax = 30;
err = 1;
k = 0;
n = size(R1,1);

R10 = R1; R20 = R2; R30 = R3; R50 = R5;

while err > tol && k < kmax
    S = R5 + R4*R3;
    Y = S\(R2 + R4*R1);
    Z = S\R6;
    X = R3*Y - R1;
    T = R3*Z;
    R3 = R3 - R1*T;
    R1 = -R1*X;
    R4 = R4 - R6*Y;
    R5 = R5 - R2*T;
    R2 = -R2*X;
    R6 = -R6*Z;
    err = min(max(norm(R1,1),norm(R2,1)),norm(R6,1));
    k = k + 1;
end
X = -(R50 + R4*R30)\(R20 + R4*R10);

if k == kmax
    disp('Warning: raggiunto il massimo numero di iterazioni')
end
```

Il codice 3.5 calcola i coefficienti iniziali della $\mathcal{UQME}$ di cui sopra utilizzando una trasformazione affine, mentre il codice 3.5 calcola i medesimi coefficienti adoperando una trasformazione di Cayley.

Listing 3.5: Metodo $\mathcal{CR}$ per $\mathcal{NARE}$ con trasformazione affine.

```
function[X] = cr_nare_aff(A,B,C,D)

% il codice risolve la NARE
%         C + XA + DX - XBX = 0
% utilizzando il metodo CR a partire da una trasformazione affine.
% X = soluzione minimale NARE


a = 1/max(diag(A));
n = size(A,1);
m = size(D,1);

% calcolo coefficienti iniziali sda mediante trasformazione affine
R1 = a*A - eye(n);
R2 = -a*C;
R3 = -a*B;
R4 = zeros(m,n);
R5 = -a*D - eye(m);
R6 = - eye(m);

% implementazione CR
[X] = cr(R1,R2,R3,R4,R5,R6);
```

Listing 3.6: Metodo $\mathcal{CR}$ per $\mathcal{NARE}$ con trasformazione di Cayley.

```
function [X] = cr_nare_cay(A,B,C,D,g)

% il codice risolve la NARE
%         C + XA + DX - XBX = 0
% utilizzando il metodo CR a partire da una trasformazione di Cayley
% di parametro g>0.
% X = soluzione minimale NARE

% calcolo coefficienti iniziali sda mediante trasformazione affine
M1 = (A + g*eye(n))\B;
M2 = (-D + g*eye(m))\C;
W = A + g*eye(n) - B*M2;
V = -D + g*eye(m) - C*M1;

IW = inv(W);
IV = inv(V);

R1 = eye(n) - 2*g*IW;
R2 = -2*g*M2*IW;
R3 = -2*g*M1*IV;
R4 = zeros(m,n);
R5 = eye(m) - 2*g*IV;
R6 = -eye(m);

% implementazione CR
[X] = cr(R1,R2,R3,R4,R5,R6);
```

Le proprietà numeriche dei metodi $\mathcal{CR}$ per la risoluzione di una $\mathcal{NARE}$ avente come matrice dei coefficienti una M-matrice non singolare o singolare irriducibile, possono riassumersi come segue

costo computazionale per $n = m$, per il metodo $\mathcal{CR}$ si rendono necessarie:

- per l'inizializzazione delle matrici ottenute con una trasformazione affine $4n^2$ operazioni elementari dovute a semplici manipolazioni sulle matrici,
- per l'inizializzazione delle matrici ottenute con una trasformazione di Cayley $\frac{52}{3}n^3$ operazioni elementari dovute a
  - due risoluzioni di sistemi lineari n-dimensionali,
  - quattro prodotti matriciali di matrici di dimensione $n \times n$,
  - due inversioni di matrici n-dimensionali,
- per una singola iterazione del metodo $\mathcal{CR}$ circa $\frac{76}{3}n^3$ operazioni elementari dovute a
  - due risoluzioni di sistemi lineari n-dimensionali,
  - dieci prodotti matriciali di matrici di dimensione $n \times n$.

velocità di convergenza Per il teorema 3.2.16 il metodo $\mathcal{CR}$ presenta convergenza quadratica, inoltre, se viene utilizzata una trasformazione affine, il raggio di convergenza delle successioni è

$$\Delta_\alpha = \left| \frac{\alpha\lambda_n - 1}{\alpha\lambda_{n+1} - 1} \right|,$$

se viene, invece adoperata una trasformazione di Cayley, il raggio di convergenza è

$$\Delta_\gamma = \left| \frac{(\lambda_n - \gamma)(\lambda_{n+1} + \gamma)}{(\lambda_n + \gamma)(\lambda_{n+1} - \gamma)} \right|.$$

stabilità numerica Anche per il metodo $\mathcal{CR}$, la stabilità dei metodi è fortemente condizionata dal drift della matrice dei coefficienti associata alla $\mathcal{NARE}$. Nel caso di $\mathcal{NARE}$ ricorrente nulla, per ottenre risultati accettabili, sono necessarie tecniche di shift.

Esempio 3.3.1. *Si consideri la $\mathcal{NARE}$ con matrici definite da*

$$A = \begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix}, \quad B = \begin{pmatrix} 1.5 & 1.5 \\ 2.9 & 0.1 \end{pmatrix}, \quad C = \begin{pmatrix} -1.9 & -1 \\ -1.9 & -1 \end{pmatrix}, \quad D = \begin{pmatrix} 3 & -0.1 \\ -0.1 & 3 \end{pmatrix}.$$

È possibile dimostrare che la matrice dei coefficienti $\mathcal{M}$ è una M-matrice singolare, che la $\mathcal{NARE}$ ha drift positivo ed ammette la seguente soluzione minimale

$$X_{\min} = \frac{1}{3} \begin{pmatrix} 1.9 & 1 \\ 1.9 & 1 \end{pmatrix}.$$

Implementando gli algoritmi sopra menzionati (per il metodo $\mathcal{CR}$ con trasformazione di Cayley si pone $g = 1$) si ottengono i seguenti risultati

	iterazioni necessarie	*errore relativo*	*tempo CPU*
$\mathcal{SDA}_{aff}$	11	$5.26e-016$	0
$\mathcal{SDA}_{cay}$	10	$4.21e-015$	0
$\mathcal{CR}_{aff}$	10	$3.42e-015$	0
$\mathcal{CR}_{cay}$	8	$3.94e-015$	0

Esempio 3.3.2. *Si consideri la $\mathcal{NARE}$ con matrici definite da*

$$A = \varepsilon \begin{pmatrix} 3 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & & & -1 & 2 \end{pmatrix}, \quad B = \varepsilon \begin{pmatrix} 1 & 1 & & \\ & 1 & \ddots & \\ & & \ddots & 1 \\ & & & 1 \end{pmatrix},$$

$$C = \begin{pmatrix} -1 & & & \\ -1 & -1 & & \\ & \ddots & \ddots & \\ & & -1 & -1 \end{pmatrix}, \quad D = \begin{pmatrix} n & -1 & \dots & -1 \\ -1 & n+1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & -1 \\ -1 & \dots & -1 & n+1 \end{pmatrix}.$$

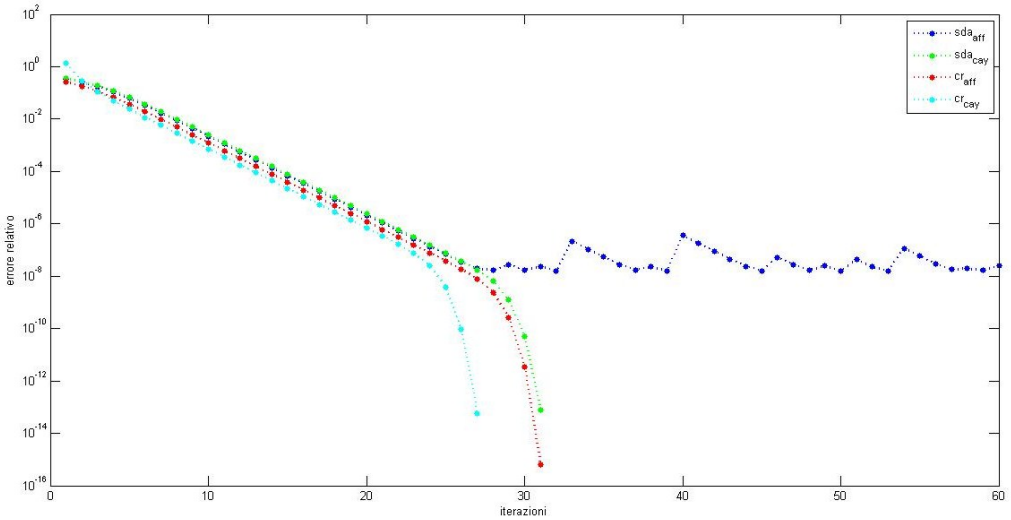
È possibile dimostrare che la matrice dei coefficienti $\mathcal{M}$ è una M-matrice singolare e che la $\mathcal{NARE}$ ha

- *drift $\mu < 0$ per valori di $\varepsilon < 1$,*
- *drift $\mu = 0$ per $\varepsilon = 1$,*
- *drift $\mu > 0$ per valori di $\varepsilon > 1$.*

Si osservi, inoltre, che per valori piccoli di ε gli elementi della matrice A, sono molto più piccoli degli elementi della matrice D, pertanto si ha $\alpha \ll \gamma$, dove α e γ sono i parametri rispettivamente delle trasformazioni affini e delle trasformazioni di Cayley. È lecito aspettarsi, quindi, che i metodi basati sulle trasformzioni affini siano più performanti di quelli basati sulle trasformazioni di Cayley. Tale aspettativa è confermata dalle sperimentazioni ottenute ponendo $\varepsilon = 0.0001$ di cui si riportano i risultati in tabella

n		$\mathcal{SDA}_{aff}$	$\mathcal{SDA}_{cay}$	$\mathcal{CR}_{aff}$	$\mathcal{CR}_{cay}$
8	*iterazioni*	2	7	3	7
	tempo CPU	0	0	0	0
16	*iterazioni*	2	8	3	8
	tempo CPU	0	0	0	0
32	*iterazioni*	2	8	3	9
	tempo CPU	0.015	0.031	0.031	0.031
64	*iterazioni*	2	9	3	9
	tempo CPU	0.031	0.062	0.031	0.093
128	*iterazioni*	2	10	3	10
	tempo CPU	0.046	0.312	0.062	0.234
256	*iterazioni*	2	11	3	11
	tempo CPU	0.296	1.653	0.452	1.778
512	*iterazioni*	2	12	3	12
	tempo CPU	2.121	12.230	3.416	14.477
1024	*iterazioni*	2	13	3	13
	tempo CPU	14.835	85.207	22.932	105.581

Il grafico sottostante mostra, per tutti i metodi, l'evoluzione dell'errore relativo al variare dell'iterazione. Nell'implementazione si è scelto $\varepsilon = 1$ *e* $n = 8$.



Esempio 3.3.3. *Si consideri la* $\mathcal{NARE}$ *con matrici definite da*

$$A = \begin{pmatrix} 2 & -1 & & \\ & 3 & \ddots & \\ & & \ddots & -1 \\ -1 & & & 3 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 1 & & \\ & 1 & \ddots & \\ & & \ddots & 1 \\ & & & 1 \end{pmatrix},$$

$$C = \begin{pmatrix} -1 & -1 & & \\ & -1 & \ddots & \\ & & \ddots & -1 \\ & & & -0.9 \end{pmatrix}, \quad D = \begin{pmatrix} 3 & -1 & & \\ & \ddots & \ddots & \\ & & 3 & -1 \\ -1 & & & 1.9 \end{pmatrix}.$$

È possibile dimostrare che la matrice dei coefficienti $\mathcal{M}$ è una M-matrice singolare e che la $\mathcal{NARE}$ ha drift negativo. Si riportano in tabella i risultati ottenuti dalle sperimentazioni

n		$\mathcal{SDA}_{aff}$	$\mathcal{SDA}_{cay}$	$\mathcal{CR}_{aff}$	$\mathcal{CR}_{cay}$
8	*iterazioni*	2	7	3	7
	tempo CPU	0	0	0	0
16	*iterazioni*	13	12	12	9
	tempo CPU	0	0	0	0
32	*iterazioni*	15	14	13	11
	tempo CPU	0.062	0.031	0.093	0.047
64	*iterazioni*	15	14	14	12
	tempo CPU	0.109	0.109	0.125	0.062
128	*iterazioni*	16	15	15	13
	tempo CPU	0.281	0.343	0.343	0.296
256	*iterazioni*	17	16	16	14
	tempo CPU	2.231	2.121	2.605	2.543
512	*iterazioni*	18	17	17	14
	tempo CPU	48.2041	28.423	46.66	19.578

3.3.2 Algoritmi per $\mathcal{CARE}$

Nella sezione 3.1.3 si è mostrato che è possibile applicare il metodo $\mathcal{SDA}$ per individuare le soluzioni estremali X_- e X_+ delle $\mathcal{CARE}$. In particolare tale metodo sfrutta le proprietà di hermitianità dei coefficienti delle $\mathcal{CARE}$ stesse riducendo sensibilmente il costo computazionale degli algoritmi. Il codice 3.7 sottostante descrive l'agoritmo $\mathcal{SDA}$ per le $\mathcal{CARE}$: richiamado la function con un valore del parametro $g > 0$, si ottiene la soluzione minimale X_-, mentre richiamandola con $g > 0$ restituisce la soluzione massimale X_+. Come ampiamente descritto nella sezione 3.1.3, l'algoritmo calcola dapprima le matrici iniziali del metodo $\mathcal{SDA}$ mediante una trasformazione di Cayley di paramentro γ, per poi generare le successioni matriciali solite, sfruttando le proprietà di hermitianità. Come per i codici della precedente sezione, viene scelto come numero massimo di iterazioni $k = 30$ e come soglia di tolleranza il valore 10^{-13}.

Listing 3.7: Metodo $\mathcal{SDA}$ per $\mathcal{CARE}$.

```
function [X] = sda_care(A,B,C,g)

% il codice individua le soluzioni estremali della care
%          C + XA + A'X - XBX = 0
% utilizzando il metdo sda a partire da una trasformazione di
% Cayley con parametro g>0 per la soluzione minimale Xm e g<0
% per la soluzione massimale Xp.


tol = 1e-13;
kmax = 30;
n = size(A,1);

% calcolo coefficienti iniziali sda per calcolo di X
% mediante trasformazione di Cayley

IA = inv(A + g*eye(n));
R = B*IA'*C;
```

```
S1 = inv(A + g*eye(n) + R);
E = S1*(A - g*eye(n) + R);
R = eye(n) - IA*(A - g*eye(n));
G = S1*B*R';
P = -S1'*C*R;

% implementazione sda per calcolo X

err = 1;
k = 0;
while err > tol && k < kmax
    M1 = eye(n) - G*P;
    Z = [E;P']/M1;
    E1 = Z(1:n,:);
    P1 = Z(n+1:2*n,:);
    G = G + E1*G*E';
    P = P + E'*P1'*E;
    E = E1*E;
    err = norm(E,1);
    k = k+1;
end
X = P;
if k == kmax
    disp('Warning: raggiunto il massimo numero di iterazioni')
end
```

Le caratterisitiche salienti del metodo $\mathcal{SDA}$ per le $\mathcal{CARE}$ possono riassumersi come segue:

costo computazionale Il metodo richiede

- $20n^3$ operazioni elementari per l'inizializzazione delle matrici ottenute con una trasformazione di Cayley dovute a
  - otto prodotti matriciali di matrici di dimensione $n \times n$,
  - due inversioni di matrici n-dimensionali,
- circa $\frac{52}{3}n^3$ operazioni elementari per una singola iterazione del metodo $\mathcal{CR}$ dovute a
  - due risoluzioni di sistemi lineari n-dimensionali,
  - sei prodotti matriciali di matrici di dimensione $n \times n$.

velocità di convergenza Per il teorema 3.1.8 il metodo $\mathcal{CR}$ presenta convergenza quadratica con raggio di convergenza delle successioni

$$\Delta_\gamma = |\mathcal{C}_\gamma(\lambda_n)|^2 = \left|\frac{(\lambda_n - \gamma)}{(\lambda_n + \gamma)}\right|^2.$$

stabilità numerica La stabilità del metodo dipende dallo split degli autovalori: se lo split è debole, infatti, possono verificarsi problemi di stabilità.

Esempio 3.3.4. *Si consideri la $\mathcal{CARE}$ definita dai coefficienti*

$$A = \begin{pmatrix} \frac{2}{3} & 0 \\ 0 & -\frac{1}{3} \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 0 \\ 0 & -\frac{4}{9} \end{pmatrix}, \quad C = \begin{pmatrix} 0 & 1 \\ 1 & 2 \end{pmatrix},$$

con soluzioni massimale e minimale rispettivamente

$$X_+ = \begin{pmatrix} \frac{2}{39}(13 + 2\sqrt{13}) & \frac{3}{\sqrt{13}} \\ \frac{3}{\sqrt{13}} & \frac{3}{52}(-13 + 9\sqrt{13}) \end{pmatrix}, \quad X_- = \begin{pmatrix} \frac{2}{39}(13 - 2\sqrt{13}) & -\frac{3}{\sqrt{13}} \\ -\frac{3}{\sqrt{13}} & -\frac{3}{52}(13 + 9\sqrt{13}) \end{pmatrix}.$$

Il codice 3.7, implementato con $g = -1$ indivudua la soluzione massimale X_+ in 0.0312 *secondi sebbene necessiti di 28 iterazioni e genera un errore relativo pari a* $4.283e-008$. *Per il calcolo della soluzione minimale X_- viene adoperato il valore $g = 1$ con prestazioni, in termini di utilizzo della CPU e numero di iterazioni, analoghe al caso precedente, e errore relativo di* $5.294e-009$.

Esempio 3.3.5. *Si consideri la $\mathcal{CARE}$ definita dai coefficienti*

$$A = \begin{pmatrix} \varepsilon+1 & 1 \\ 1 & \varepsilon+1 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad C = \begin{pmatrix} \varepsilon^2 & 0 \\ 0 & \varepsilon^2 \end{pmatrix}.$$

È possibile dimostrare che la $\mathcal{CARE}$ precedente ammette soluzione massimale X_+ data da $X_+ = \left(\begin{smallmatrix} x_1 & x_2 \\ x_2 & x_1 \end{smallmatrix}\right)$ con

$$x_1 := \frac{1}{2}\left(2(\varepsilon+1) + \sqrt{2(\varepsilon+1)^2+2} + \sqrt{2}\varepsilon\right), \quad x_2 := \frac{x_1}{x_1 - (\varepsilon+1)}.$$

Implementando il codice 3.7 con $g = -1$ si ottengono i risultati illustrati in tabella

ε	*iterazioni necessarie*	*errore relativo*	*tempo CPU*
10^-1	6	$1.109e-015$	0
10^-2	10	$4.397e-014$	0
10^-3	13	$1.308e-011$	0.0312
10^-4	16	$2.102e-010$	0.0312
10^-5	20	$9.877e-009$	0.0312
10^-6	22	$4.743e-007$	0.0312
10^-7	23	$3.380e-004$	0.0312

Esempio 3.3.6. *Siano*

$$A = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \dots & \dots & 0 & 1 \\ 0 & \dots & \dots & 0 & 0 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ \vdots \\ \vdots \\ 0 \\ 1 \end{pmatrix}, \quad c = \sqrt{q}\begin{pmatrix} 1 \\ \vdots \\ \vdots \\ 0 \\ 0 \end{pmatrix},$$

e si consideri la $\mathcal{CARE}$ con coefficienti A, $B := \frac{1}{r}b\dot{b}*$, $C := c\dot{c}*$, con r parametro che dipende dalle applicazioni.

Non è possibile esplicitare una formulazione della soluzione massimale X_+, è bensì noto che l'elemento $x_{1,n}$ di tale soluzione assume valore $\sqrt{rq}$. Pertanto nelle sperimentazioni, come stima dell'errore relativo, si pone

$$rel := \frac{|x_{1,n} - \sqrt{rq}|}{\sqrt{rq}}.$$

Al solito si riassumo i dati salienti delle implementazioni nella tabella sottostante

n	*iterazioni necessarie*	*errore relativo*	*tempo CPU*
4	6	$6.661e-016$	0
6	7	$1.147e-008$	0
8	7	$4.498e-011$	0
10	8	$1.147e-008$	0.0312
12	9	$2.763e-004$	0.0312

L'algoritmo non può essere implentato con valori di n più elevati in quanto si generano problemi di singolarità delle matrici coinvolte.

3.3.3 Algoritmi per $\mathcal{DARE}$

Nella sezione 3.1.4 si è mostrato che, ponendo $G := BR^{-1}B^*$, è possibile portare la $\mathcal{DARE}$

$$X = A^*XA - A^*XB(R + B^*XB)^{-1}B^*XA + Q$$

nella forma

$$X = A^*X(I_n + GX)^{-1}A + Q,$$

e dunque calcolare le soluzioni X come graph invariant subspace di una particolare matrix pencil. Il codice di seguito individua la soluzione d-stabilizzante generando le successioni definite dal matodo $\mathcal{SDA}$.

Listing 3.8: Metodo $\mathcal{SDA}$ per $\mathcal{DARE}$.

```
function [X] = sda_dare(A,B,R,Q)

% il codice individua la soluzione d-stabilizzante della dare
%          X= A'XA - A'XB(R + B'XB)^{-1}B'XA + Q.
%
% X = soluzione d-stabilizzante della dare


tol = 1e-13;
kmax = 30;
n = size(A,1);
err = 1;
k = 0;
G = B/R;
G = G*B';

while err > tol && k < kmax
    M1 = eye(n) + G*Q;
    M2 = eye(n) + Q*G;
    A1 = A/M1;
    G1 = G/M1;
    Q1 = A'/M2;
    G = G + A*G1*A';
    Q = Q + Q1*Q*A;
    A = A1*A;
    err = norm(A,1);
    k = k + 1;
end

X = Q;
if k == kmax
    disp('Warning: raggiunto il massimo numero di iterazioni')
end
```

Alla luce del codice 3.8, il metodo $\mathcal{SDA}$ per le $\mathcal{DARE}$ presenta le seguenti caratteristiche

costo computazionale Implementare l'algoritmo comporta

- $\frac{14}{3}n^3$ operazioni elementari per l'inizializzazione della matrice G dovute a
  - una risoluzione di un sistema lineare n-dimensionale,
  - un prodotto matriciale di matrici di dimensione $n \times n$,
- $22n^3$ operazioni elementari per un singolo passo del metodo $\mathcal{SDA}$ dovute a
  - tre risoluzioni di sistemi lineari n-dimensionali,
  - sette prodotti matriciali di matrici di dimensione $n \times n$.

velocità di convergenza Per il teorema 3.1.9 il metodo $\mathcal{SDA}$ presenta convergenza quadratica con raggio di convergenza delle successioni

$$\Delta = |(\lambda_n)|^2.$$

stabilità numerica In analogia con i metodi precedenti, la stabilità del metodo dipende dallo split degli autovalori della matrix pencil che genera il metodo $\mathcal{SDA}$.

Esempio 3.3.7. *Si consideri la $\mathcal{DARE}$ definita dai coefficienti*

$$A = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \dots & \dots & 0 & 1 \\ 0 & \dots & \dots & 0 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ \vdots \\ \vdots \\ 0 \\ 1 \end{pmatrix}, \quad R = r, \quad Q = I_n.$$

In tal caso la soluzione d-stabilizzante X assume la seguente forma

$$X = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ \vdots & 2 & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \dots & \dots & n-1 & 0 \\ 0 & \dots & \dots & 0 & n \end{pmatrix}.$$

Al solito si riassumo i dati salienti delle implementazioni nella tabella sottostante

n	*iterazioni necessarie*	*errore relativo*	*tempo CPU*
8	2	0	0
16	3	0	0
32	4	0	0.062
64	5	0	0.062
128	7	0	0.109
256	7	$3.527e-015$	0.951
512	8	$6.364e-013$	7.347

La 'carrellata' di codici si conclude con il codice 3.9 che illustra l'applicazione del metodo $\mathcal{CR}$ alla $\mathcal{DARE}$ della forma

$$X + C^* X^{-1} C = Q.$$

Listing 3.9: Metodo $\mathcal{CR}$ per una particolare $\mathcal{DARE}$.

```
function [Xm,Xp] = cr_dare(C,Q)

% il programma risolve la particolare DARE
%              X +C'X^{-1}C= Q
% utilizzando il metodo CR.
%
% Xm = soluzione minimale DARE
% Xm = soluzione massimale DARE

tol = 1e-13;
kmax = 30;
err = 1;
k = 0;
n = size(Q,1);
Q0 = Q;
```

```
Xp = Q;
Yp = Q;
while err > tol && k < kmax
    M1 = Q\C;
    M2 = Q\C';
    N1 = C'*M1;
    N2 = C*M2;
    C = C*M1;
    Q = Q - N1 - N2;
    Xp = Xp - N1;
    Yp = Yp - N2;
    err = norm(C,1);
    k = k + 1;
end
Xm = Q0 - Yp;
if k == kmax
    disp('Warning: raggiunto il massimo numero di iterazioni')
end
```

costo computazionale L'algoritmo necessita di $\frac{34}{3}n^3$ operazioni elementari dovute a

- due risoluzioni di sistemi lineari n-dimensionale,
- sei prodotti matriciali di matrici di dimensione $n \times n$,

velocità di convergenza Per il teorema 3.2.19 il metodo $\mathcal{CR}$ presenta convergenza quadratica.

stabilità numerica La stabilità del metodo dipende dallo split degli autovalori della matrix pencil razionale $\Phi(\cdot)$ definita in 3.53.

Esempio 3.3.8. *Si consideri la $\mathcal{DARE}$ particolare con*

$$C = \begin{pmatrix} 50 & 20 \\ 10 & 60 \end{pmatrix}, \quad Q = \begin{pmatrix} 3 & 2 \\ 2 & 4 \end{pmatrix}.$$

Il codice 3.9 individua le soluzioni estremali della $\mathcal{DARE}$ con tolleranza di $1e-13$ *in* 36 *iterazioni e necessita di* 0.0312 *secondi.*

Capitolo 4

Metodo $\mathcal{SDA}$ per equazioni di Riccati di grandi dimensioni

Indice

Gli algoritmi descritti nel capitolo precedente, come ampiamente mostrato, risultano i più efficaci per la risoluzione di equazioni di Riccati in quanto presentano una convergenza quadratica ed hanno un costo computazionale di $O(n^3)$ operazioni elementari per iterazione (se le matrici che definiscono le equazioni hanno dimensione $n \times n$), costo computazionale relativamente limitato se comparato a quello dei metodi classici. Tuttavia, per valori molto grandi di n, tale costo computazionale diviene 'insostenibile' e i dobling algorithm possono risultare inapplicabili in quanto richiederebbero tempi d'esecuzione eccessivi o comunque non accettabili. Per una trattazione dettagliata di tale casistica, va in primo luogo chiarito cosa si intende per *valore molto grande della dimensione di una matrice*: euristicamente, si dice che una dimensione è grande se operazioni quali il prodotto matrice-vettore o la risoluzione di sistemi lineari richiedono tempi di calcolo non ragionevoli. È evidente che tale definizione è tutt'altro che rigorosa e formale, in quanto il tempo di esecuzione di una determinata operazione dipende, tra l'altro, dal calcolatore a disposizione. Se, quindi, le matrici che definiscono un'equazione di Riccati hanno grandi dimensioni, può non essere possibile individuarne la soluzione in quanto, alla base degli algoritmi risolutivi vi sono prodotti matrice-vattore e risoluzioni di sistemi lineari che risulterebbero non calcolabili. In molte applicazioni però, le matrici hanno sì grandi dimensioni, ma presentano proprietà di sparsezza o di rango basso, proprietà che, se opportunamente sfruttate, potrebbero limitare il costo computazionale di suddette operazioni. Utilizzando tali proprietà strutturali delle matrici coinvolte, è possibile riadattare i doubling algorithm in modo da ridurre il costo computazionale a $O(n)$ operazioni elementari per passo ottenendo, quindi, un cospicuo risparmio dei tempi d'esecuzione degli algoritmi.

Il presente capitolo illustra le 'correzioni' apportate dai matematici cinesi Chang-Yi Weng, Tiexiang Li, Eric King-wah Chu e Wen-Wei Lin al doubling algorithm strutturato per equazioni di Riccati definite da matrici aventi grandi dimensioni e con particolari proprietà di struttura. Come comune denominatore delle modifiche apportate al metodo $\mathcal{SDA}$ vi è un accorto utilizzo della *Formula di Sherman-Morrison-Woodbury* ($\mathcal{SMWF}$)

e l'idea di *troncare e comprimere* la crescita delle correzioni di rango definite nel corso dell'algoritmo. Gli autori, quindi, estendono la versione standard del metodo $\mathcal{SDA}$ ad equazioni di Riccati di grandi dimensioni e determinate proprietà struttuarali elaborando il metodo *large-scale* $\mathcal{SDA}$ ($\mathcal{SDA}_{ls}$).

Il paragrafo 4.1 illustra le caratterisitiche del metodo $\mathcal{SDA}_{ls}$ per $\mathcal{NARE}$: dapprima vengono definite le proprietà richieste sulla struttura dei coefficienti, sono quindi descritte le successioni definite per ricorrenza del metodo $\mathcal{SDA}_{ls}$ con particolare attenzione all'utilizzo della $\mathcal{SMWF}$, è poi spiegato il meccanismo di troncamento e compressione della crescita, potenzialmente esponenziale, delle correzioni di rango. È, quindi, descritto una procedura per controllare la convergenza dell'algoritmo e calcolare il residuo relativo rimanendo nella soglia massima fissata di $O(n)$ operazioni elementari. Infine, vi è un'analisi della propagazione dell'errore generato dal procedimento di troncamento e compressione ed un resoconto sul costo computazionale dell'algoritmo e dell'impiego di memoria richiesto.

Nel paragrafo 4.2 sono raccolti i codici MATLAB del metodo $\mathcal{SDA}_{ls}$ e sono presentate delle implementazioni dell'algoritmo proposto, particolare attenzione è rivolta alla relazione tra i parametri che definiscono il procedimento di troncamento e compressione e l'errore relativo della corrispondente soluzione. Sono, inoltre, evidenziate le differenze in termini di prestazioni tra il metodo $\mathcal{SDA}$ e il metodo $\mathcal{SDA}_{ls}$ se testati a problemi di grandi dimensioni con le proprietà di struttura richieste.

Nel paragrafo 4.3 sono riportati i commenti dell'autore del presente elaborato al metodo $\mathcal{SDA}_{ls}$. L'algortimo offre indubbiamente numerose garanzie dal punto di vista computazionale, ma sono al contempo diversi i punti poco chiari e gli aspetti che, a modesto parere dello scrivente, non sono trattati con la dovuta profondità.

4.1 Metodo $\mathcal{SDA}$ per $\mathcal{NARE}$ di grandi dimensioni

Si consideri la $\mathcal{NARE}$ nella indeterminata X

$$C + XA + DX - XBX = 0, \tag{4.1}$$

e si pongano sui coefficienti le seguenti ipotesi

- $A \in \mathbb{R}^{n\times n}$ matrice di grandi dimensioni sparsa tale che il calcolo dei prodotti $A^{-1}u$ e $A^{-T}u$, per un qualsivoglia vettore $u \in \mathbb{R}^n$, richieda $O(n)$ operazioni elementari,
- $D \in \mathbb{R}^{m\times m}$ matrice di grandi dimensioni sparsa tale che il calcolo dei prodotti $D^{-1}v$ e $D^{-T}v$, per un qualsivoglia vettore $v \in \mathbb{R}^m$, richieda $O(m)$ operazioni elementari,
- $B \in \mathbb{R}^{n\times m}$ ammettente la seguente fattorizzazione di rango basso

  $$B := B_1 R B_2^T,$$

  dove $B_1 \in \mathbb{R}^{n\times l}$, $B_2 \in \mathbb{R}^{m\times l}$, $R \in \mathbb{R}^{l\times l}$ con $l \ll \max\{\, n, m \,\}$,
- $C \in \mathbb{R}^{m\times n}$ ammettente la seguente fattorizzazione di rango basso

  $$C := C_1 T C_2^T,$$

  dove $C_1 \in \mathbb{R}^{m\times h}$, $C_2 \in \mathbb{R}^{n\times h}$, $R \in \mathbb{R}^{h\times h}$ con $h \ll \max\{\, n, m \,\}$.

Tali richieste sulla struttura dei coefficienti sono, per esempio, rispettate dalla $\mathcal{NARE}$ proposta nel paragrafo 1.2 che descrive il moto di particelle lungo un'asta omogenea. In tal caso, infatti, ridenominando opportunamente le matrici, si ha

- $A := (I_n - \Phi_1\Phi_2^T)\Delta^+ = \Delta^+ - \Phi_1\left(\Phi_2^T\Delta^+\right)$,
- $D := (I_n - \Phi_1\Phi_2^T)\Delta^- = \Delta^- - \Phi_1\left(\Phi_2^T\Delta^-\right)$,
- $B := -\Gamma_1\Gamma_2^T\Delta^- = -\Gamma_1\left(\Gamma_2^T\Delta^-\right)$,

- $C := -\Gamma_1\Gamma_2^T\Delta^+ = -\Gamma_1\left(\Gamma_2^T\Delta^+\right)$,

con Φ_1, Φ_2 matrici di rango basso sparse, Γ_1, Γ_2 matrici di rango basso, Δ^+, Δ^- matrici diagonali. Se si suppone, inoltre, che il sistema abbia un numero di stati n grande, si ritrovano le ipotesi sopra menzionate. Tale esempio sarà oggetto delle implementazioni illustrate nel paragrafo 4.2.

Un passaggio essenziale per lo sviluppo degli algoritmi illustrati nel presente e nel successivo capitolo, è individuare un modo efficiente per calcolare l'inversa di **una matrice sparsa di grandi dimensione più una matrice di rango basso**. Lo strumento migliore per calcolare tale inversa è la formula di Sherman-Morrison-Woodbury presentata dai medesimi autori in [52].

Teorema 4.1.1 (Formula di Sherman-Morrison-Woodbury)**.** *Sia $A \in \mathbb{R}^{n\times n}$ una matrice invertibile e siano $U, V \in \mathbb{R}^{m\times n}$ con $m \leq n$. Allora la matrice $A + UV^T$ è invertibile se e solo se lo è la matrice $I_m + V^TA^{-1}U$, e vale la relazione*

$$(A + UV^T)^{-1} = A^{-1} - A^{-1}U(I_m + V^TA^{-1}U)^{-1}V^TA^{-1},$$

detta ***formula di Sherman-Morrison-Woodbury****.*

Dimostrazione. È sufficiente svolgere i calcoli ottenendo

$$\begin{gathered}(A + UV^T)\left(A^{-1} - A^{-1}U(I_m + V^TA^{-1}U)^{-1}V^TA^{-1}\right) = \\ I_n + UV^TA^{-1} - U(I_m + V^TA^{-1}U)^{-1}V^TA^{-1} - UV^TA^{-1}U(I_m + V^TA^{-1}U)^{-1}V^TA^{-1} = \\ I_n + UV^TA^{-1} - U(I_m + V^TA^{-1}U)(I_m + V^TA^{-1}U)^{-1}V^TA^{-1} = \\ I_n + UV^TA^{-1} - UV^TA^{-1} = I_n,\end{gathered}$$

da cui la tesi. □

L'importanza del teorema 4.1.1 è evidente: se la matrice A è diagonale o comunque facilmente invertibile, la $\mathcal{SMWF}$ permette di ricondurre il calcolo di una matrice di grandi dimensioni al calcolo dell'inversa di una matrice di dimensioni minori, ottenendo una notevole riduzione delle operazioni necessarie.

I matematici cinesi Chang-Yi Weng, Tiexiang Li, Eric King-wah Chu e Wen-Wei Lin hanno apportato delle migliorie al metodo $\mathcal{SDA}$ standard per risolvere le $\mathcal{NARE}$ del tipo (4.1) che permettono di abbattere sensibilmente il costo computazionale portandolo da $O(n^3)$ operazioni elementari per iterazione a 'sole' $O(n)$ operazioni. I suddetti autori propongono in [[55]] una estensione del metodo $\mathcal{SDA}$, il **large-scale** $\mathcal{SDA}$ ($\mathcal{SDA}_{ls}$), sviluppato sulle seguenti idee fondamentali:

- l'utilizzo della formula di Sherman-Morrison-Woodbury per calcolare efficientemente l'inversa di una matrice sparsa più una matrice di rango basso,
- l'utilizzo di formule ricorsive che preservino le proprietà di struttura iniziali,
- il procedimento di troncamento e compressione per controllare la crescita potenzialmente esponenziale delle correzioni di rango cercando di raggiungere un 'compromesso' tra migliore efficienza dell'algortimo e perdita di accuratezza della soluzione,
- la descrizione di una condizione d'arresto e di un controllo di convergenza che mantenga il costo computazionale dell'algoritmo a $O(n)$ operazioni per iterazione.

Alla luce di quanto esposto nella sezione 3.1.2, il metodo $\mathcal{SDA}$ standard genera le successioni definite per ricorrenza

$$\begin{aligned}E_{k+1} &= E_k(I_n - G_kP_k)^{-1}E_k,\\ P_{k+1} &= P_k + F_k(I_m - P_kG_k)^{-1}P_kE_k,\\ F_{k+1} &= F_k(I_m - P_kG_k)^{-1}F_k,\\ G_{k+1} &= G_k + E_k(I_n - G_kP_k)^{-1}G_kF_k,\end{aligned} \tag{4.2}$$

che verificano le propretà

$$\lim_{k\to 0} E_k = 0, \quad \lim_{k\to 0} P_k = X_{\min}, \quad \lim_{k\to 0} F_k = 0, \quad \lim_{k\to 0} G_k = Y_{min}, \tag{4.3}$$

dove $X_{\min}$ e Y_{min} sono rispettivamente le soluzioni minimali della $\mathcal{NARE}$ (4.1) e della sua duale. I valori iniziali sono dati da

$$\begin{aligned} E_0 &= E^{(\alpha)} = \alpha A - I_n + \alpha^2 B D_\alpha^{-1} C, \\ P_0 &= P^{(\alpha)} = -\alpha D_\alpha^{-1} C, \\ F_0 &= F^{(\alpha)} = -D_\alpha^{-1}, \\ G_0 &= G^{(\alpha)} = \alpha B D_\alpha^{-1}, \end{aligned} \tag{4.4}$$

con $D_\alpha := \alpha D + I_m$, se si utilizza il metodo $\mathcal{SDA}$ inizializzato da una trasformazione affine $\mathcal{A}_\alpha$ con $\alpha > 0$. Se si adopera, invece, una trasformazione di Cayley $\mathcal{C}_\gamma$ si ha

$$\begin{aligned} E_0 &= E^{(\gamma)} = I_n - 2\gamma V_\gamma^{-1}, \\ P_0 &= P^{(\gamma)} = -2\gamma D_\gamma^{-1} C V_\gamma^{-1}, \\ F_0 &= F^{(\gamma)} = I_m - 2\gamma W_\gamma^{-1}, \\ G_0 &= G^{(\gamma)} = 2\gamma A_\gamma^{-1} B W_\gamma^{-1}, \end{aligned} \tag{4.5}$$

con

$$A_\gamma := A + \gamma I_n, \qquad D_\gamma := D + \gamma I_m,$$

e

$$V_\gamma := A_\gamma + B D_\gamma^{-1} C, \quad W_\gamma := D_\gamma + C A_\gamma^{-1} B,$$

dove il parametro γ verifica

$$\gamma \geq \max\{ \max_{i=1,\dots,n} a_{ii}, \max_{i=j,\dots,m} d_{jj} \}.$$

Sia $\bar{n} := \max\{ n, m \}$ e allora si osservi ora che, ponendo sui coefficienti le ipotesi di struttura della $\mathcal{NARE}$ (4.1), per inizializzare l'algoritmo

- se si utilizza una trasformazione affine $\mathcal{A}_\alpha$, è necessario invertire la matrice D_α, operazione che comporta un costo di $O(\bar{n})$ operazioni elementari,
- se si adopera una trasformazione di Cayley $\mathcal{C}_\gamma$ occorre
  - calcolare le inverse A_γ^{-1} e D_γ^{-1} il cui costo è per ipotesi di $O(\bar{n})$ operazioni,
  - determinare mediante la $\mathcal{SMWF}$ l'inversa

$$\begin{aligned} V_\gamma^{-1} &= (A_\gamma + B D_\gamma^{-1} C)^{-1} \\ &= A_\gamma^{-1} - A_\gamma^{-1} B_1 \left(I_l + R B_2^T D_\gamma^{-1} C A_\gamma^{-1} B_1 \right)^{-1} R B_2^T D_\gamma^{-1} C A_\gamma^{-1}, \\ &= A_\gamma^{-1} - A_\gamma^{-1} B D_\gamma^{-1} C_1 T \left(I_h + C_2^T A_\gamma^{-1} B D_\gamma^{-1} C_1 \right)^{-1} C_2^T A_\gamma^{-1}, \end{aligned} \tag{4.6}$$

    che richiede $O(\bar{n})$ operazioni elementari,
  - determinare in modo analogo l'inversa

$$\begin{aligned} W_\gamma^{-1} &= (D_\gamma + C A_\gamma^{-1} B)^{-1} \\ &= D_\gamma^{-1} - D_\gamma^{-1} C_1 \left(I_h + T C_2^T A_\gamma^{-1} B D_\gamma^{-1} C_1 \right)^{-1} T C_2^T A_\gamma^{-1} B D_\gamma^{-1}, \\ &= D_\gamma^{-1} - D_\gamma^{-1} C A_\gamma^{-1} B_1 R \left(I_l + B_2^T D_\gamma^{-1} C A_\gamma^{-1} B_1 R \right)^{-1} B_2^T D_\gamma^{-1}. \end{aligned} \tag{4.7}$$

    operazione che necessita di $O(\bar{n})$ operazioni aritmetiche.

Utilizzando le proprietà strutturali dei coefficienti della $\mathcal{NARE}$ ed applicando opportunamente la $\mathcal{SMWF}$ è, quindi, possibile calcolare i termini iniziali delle successioni che definiscono il metodo $\mathcal{SDA}$ con un costo computazionale di sole $O(\bar{n})$ operazioni elementari. Un'analisi più puntuale del costo computazionale per inizializzare il metodo $\mathcal{SDA}_{ls}$ è riportata nella sezione 4.1.5.

4.1.1 Descrizione dell'algoritmo

Un passo di fondamentale importanza nello sviluppo dell'algoritmo $\mathcal{SDA}_{ls}$ è riscrivere le successioni (4.2) mediante formule ricorsive in modo da ottenere una rappresentazione della matrici E_k, F_k, G_k e H_k che meglio sfrutti le proprietà di struttura inizialmente presenti nei coefficienti.

Si ponga dunque

$$\begin{aligned} E_k &:= E_{k-1}^2 + E_{1k}E_{2k}^T, \\ P_k &:= C_{1k}T_kC_{2k}^T, \\ F_k &:= F_{k-1}^2 + F_{1k}F_{2k}^T, \\ G_k &:= B_{1k}R_kB_{2k}^T, \end{aligned} \tag{4.8}$$

con $E_{ik} \in \mathbb{R}^{n\times r_{E_k}}$, $F_{ik} \in \mathbb{R}^{m\times r_{F_k}}$ per $i = 1, 2$, $B_{1k} \in \mathbb{R}^{n\times l_k}$, $B_{2k} \in \mathbb{R}^{m\times l_k}$, e $C_{1k} \in \mathbb{R}^{m\times h_k}$, $C_{2k} \in \mathbb{R}^{n\times h_k}$. Tale costruzione permette, tra l'altro, di non determinare esplicitamente i termini E_k e F_k perdendo ogni proprietà di struttura, ma calcolare i prodotti del tipo E_ku, E_k^Tu, F_kv, F_k^Tv, per qualsivoglia vettori u e v, applicando ricorsivamente le (4.8).

Occorre, pertanto, a partire dalla (4.2), individuare una formulazione ricorsiva delle matrici B_{ik}, C_{ik}, E_{ik}, F_{ik}, R_k e T_k. Si osservi che, applicando la $\mathcal{SMWF}$ si ottiene

$$\begin{aligned} (I_n - G_kP_k)^{-1} &= (I_n - B_{1k}R_kB_{2k}^TP_k)^{-1} = \\ &= I_n + B_{1k}(I_{l_k} - R_kB_{2k}^TP_kB_{1k})^{-1}R_kB_{2k}^TP_k, \\ &= (I_n - G_kC_{1k}T_kC_{2k}^T)^{-1} = \\ &= I_n + G_kC_{1k}T_k(I_{h_k} - C_{2k}^TG_kC_{1k}T_k)^{-1}C_{2k}^T, \end{aligned} \tag{4.9}$$

e

$$\begin{aligned} (I_m - P_kG_k)^{-1} &= (I_m - C_{1k}T_kC_{2k}^TG_k)^{-1} = \\ &= I_m + C_{1k}(I_{h_k} - T_kC_{2k}^TG_kC_{1k})^{-1}T_kC_{2k}^TG_k, \\ &= (I_m - P_kB_{1k}R_kB_{2k}^T)^{-1} = \\ &= I_m + P_kB_{1k}R_k(I_{l_k} - B_{2k}^TP_kB_{1k}R_k)^{-1}B_{2k}^T. \end{aligned} \tag{4.10}$$

La $\mathcal{SMWF}$ permette quindi di calcolare le inverse di tali matrici con un costo computazionale di $O(\bar{n})$ operazioni elementari. Utilizzando la (4.9) si ottiene

$$\begin{aligned} E_{k+1} &= E_k(I_n - G_kP_k)^{-1}E_k \\ &= E_k\left(I_n + B_{1k}(I_{l_k} - R_kB_{2k}^TC_{1k}T_kC_{2k}^TB_{1k})^{-1}R_kB_{2k}^TC_{1k}T_kC_{2k}^T\right)E_k \\ &= E_k^2 + E_kB_{1k}(I_{l_k} - R_kB_{2k}^TC_{1k}T_kC_{2k}^TB_{1k})^{-1}R_kB_{2k}^TC_{1k}T_kC_{2k}^TE_k, \\ &= E_k\left(I_n + B_{1k}R_kB_{2k}^TC_{1k}T_k(I_{h_k} - C_{2k}^TB_{1k}R_kB_{2k}^TC_{1k}T_k)^{-1}C_{2k}^T\right)E_k \\ &= E_k^2 + E_kB_{1k}R_kB_{2k}^TC_{1k}T_k(I_{h_k} - C_{2k}^TB_{1k}R_kB_{2k}^TC_{1k}T_k)^{-1}C_{2k}^TE_k, \end{aligned}$$

da cui

$$\begin{aligned} E_{1,k+1} &:= E_kB_{1k}(I_{l_k} - R_kB_{2k}^TC_{1k}T_kC_{2k}^TB_{1k})^{-1}R_kB_{2k}^TC_{1k}T_k \\ &:= E_kB_{1k}R_kB_{2k}C_{1k}T_k(I_{h_k} - C_{2k}^TB_{1k}R_kB_{2k}^TC_{1k}T_k)^{-1}, \end{aligned} \tag{4.11}$$

e

$$E_{2,k+1} = E_k^TC_{2k}. \tag{4.12}$$

Allo stesso modo, utilizzando la (4.10), si ricava

$$\begin{aligned} F_{k+1} &= F_k(I_m - P_kG_k)^{-1}F_k \\ &= F_k\left(I_m + C_{1k}(I_{h_k} - T_kC_{2k}^TB_{1k}R_kB_{2k}^TC_{1k})^{-1}T_kC_{2k}^TB_{1k}R_kB_{2k}^T\right)F_k \\ &= F_k^2 + F_kC_{1k}(I_{h_k} - T_kC_{2k}^TB_{1k}R_kB_{2k}^TC_{1k})^{-1}T_kC_{2k}^TB_{1k}R_kB_{2k}^TF_k, \\ &= F_k\left(I_m + C_{1k}T_kC_{2k}^TB_{1k}R_k(I_{l_k} - B_{2k}^TC_{1k}T_kC_{2k}^TB_{1k}R_k)^{-1}B_{2k}^T\right)F_k \\ &= F_k^2 + F_kC_{1k}T_kC_{2k}^TB_{1k}R_k(I_{l_k} - B_{2k}^TC_{1k}T_kC_{2k}^TB_{1k}R_k)^{-1}B_{2k}^TF_k, \end{aligned}$$

da cui

$$\begin{aligned} F_{1,k+1} &:= F_k C_{1k}(I_{h_k} - T_k C_{2k}^T B_{1k} R_k B_{2k}^T C_{1k})^{-1} T_k C_{2k}^T B_{1k} R_k \\ &:= F_k C_{1k} T_k C_{2k}^T B_{1k} R_k (I_{l_k} - B_{2k}^T C_{1k} T_k C_{2k}^T B_{1k} R_k)^{-1}, \end{aligned} \tag{4.13}$$

e

$$F_{2,k+1} := F_k^T B_{2k}. \tag{4.14}$$

Procedendo in modo analogo si ha

$$\begin{aligned} P_{k+1} &= P_k + F_k(I_m - P_k G_k)^{-1} P_k E_k \\ &= C_{1k} T_k C_{2k}^T + F_k \left(I_m + C_{1k}(I_{h_k} - T_k C_{2k}^T B_{1k} R_k B_{2k}^T C_{1k})^{-1} T_k C_{2k}^T B_{1k} R_k B_{2k}^T\right) C_{1k} T_k C_{2k}^T E_k, \\ &= C_{1k} T_k C_{2k}^T + F_k \left(I_m + C_{1k} T_k C_{2k}^T B_{1k} R_k (I_{l_k} - B_{2k}^T C_{1k} T_k C_{2k}^T B_{1k} R_k)^{-1} B_{2k}^T\right) C_{1k} T_k C_{2k}^T E_k, \end{aligned}$$

allora

$$C_{1,k+1} := \begin{pmatrix} C_{1k} & F_k C_{1k} \end{pmatrix}, \tag{4.15}$$

$$C_{2,k+1} := \begin{pmatrix} C_{2k} & E_k^T C_{2k} \end{pmatrix}, \tag{4.16}$$

$$\begin{aligned} T_{k+1} &:= \begin{pmatrix} T_k & 0 \\ 0 & T_k + (I_{h_k} - T_k C_{2k}^T B_{1k} R_k B_{2k}^T C_{1k})^{-1} T_k C_{2k}^T B_{1k} R_k B_{2k}^T C_{1k} T_k \end{pmatrix}, \\ &:= \begin{pmatrix} T_k & 0 \\ 0 & T_k + T_k C_{2k}^T B_{1k} R_k (I_{l_k} - B_{2k}^T C_{1k} T_k C_{2k}^T B_{1k} R_k)^{-1} B_{2k}^T C_{1k} T_k \end{pmatrix}. \end{aligned} \tag{4.17}$$

È possibile, infine, ottenere

$$\begin{aligned} G_{k+1} &= G_k + E_k(I_n - G_k P_k)^{-1} G_k F_k \\ &= B_{1k} R_k B_{2k}^T + E_k \left(I_n + B_{1k}(I_{l_k} - R_k B_{2k}^T C_{1k} T_k C_{2k}^T B_{1k})^{-1} R_k B_{2k}^T C_{1k} T_k C_{2k}^T\right) B_{1k} R_k B_{2k}^T F_k, \\ &= B_{1k} R_k B_{2k}^T + E_k \left(I_n + B_{1k} R_k B_{2k}^T C_{1k} T_k (I_{h_k} - C_{2k}^T B_{1k} R_k B_{2k}^T C_{1k} T_k)^{-1} C_{2k}^T\right) B_{1k} R_k B_{2k}^T F_k, \end{aligned}$$

allora

$$B_{1,k+1} := \begin{pmatrix} B_{1k} & E_k B_{1k} \end{pmatrix}, \tag{4.18}$$

$$B_{2,k+1} := \begin{pmatrix} B_{2k} & F_k^T B_{2k} \end{pmatrix}, \tag{4.19}$$

$$\begin{aligned} R_{k+1} &:= \begin{pmatrix} R_k & 0 \\ 0 & R_k + (I_{l_k} - R_k B_{2k}^T C_{1k} T_k C_{2k}^T B_{1k})^{-1} R_k B_{2k}^T C_{1k} T_k C_{2k}^T B_{1k} R_k \end{pmatrix}, \\ &:= \begin{pmatrix} R_k & 0 \\ 0 & R_k + R_k B_{2k}^T C_{1k} T_k (I_{h_k} - C_{2k}^T B_{1k} R_k B_{2k}^T C_{1k} T_k)^{-1} C_{2k}^T B_{1k} R_k \end{pmatrix}. \end{aligned} \tag{4.20}$$

Mediante la $\mathcal{SMWF}$, quindi, si sono ottenute delle formulazioni ricorsive delle matrici F_{ik}, E_{ik}, B_{ik}, C_{ik}, R_k, T_k. Per concludere la formulazione dell'algoritmo, dunque, occorre dare una espressione coerente con la (4.8) dei valori iniziali. Se si utilizza una trasformazione affine $\mathcal{A}_\alpha$, si pone

$$\begin{gathered} E_0 := E^{(\alpha)}, \\ C_{1,0} := D_\alpha^{-1} C_1 \qquad T_0 := -\alpha T \qquad C_{2,0} := C_2, \\ F_0 := F^{(\alpha)}, \\ B_{1,0} := B_1 \qquad R_0 := \alpha R \qquad B_{2,0} := D_\alpha^{-T} B_2, \end{gathered}$$

mentre, se si adopera una trasformazione di Cayley $\mathcal{C}_\gamma$, è possibile porre

$$\begin{gathered} E_0 := E^{(\gamma)}, \\ C_{1,0} := D_\gamma^{-1} C_1 \qquad T_0 := -2\gamma T \qquad C_{2,0} := V_\gamma^{-T} C_2, \\ F_0 := F^{(\gamma)}, \\ B_{1,0} := A_\gamma^{-1} B_1 \qquad R_0 := 2\gamma R \qquad B_{2,0} := W_\gamma^{-T} B_2. \end{gathered}$$

Dal lungo elenco di formule, si evince che il costo per il calcolo delle successioni definite dal metodo $\mathcal{SDA}_{ls}$ è dominato dall'oneroso costo per il calcolo di prodotti del tipo $E_k u$, $E_k^T u$, $F_k v$, $F_k^T v$ per qualsivoglia vettori u e v. Si osservi, però, che, applicando ricorisivamente le formule (4.8), è possibile ricondurre tali operazioni a prodotti del tipo $E^{(\delta)}u$, ${E^{(\delta)}}^T u$, $F^{(\delta)}v$, ${F^{(\delta)}}^T v$, con $\delta = \alpha, \gamma$, che, per le ipotesi poste sui coefficienti della $\mathcal{NARE}$, hanno un costo computazionale di $O(\bar{n})$ operazioni elementari. Si rimanda alla sezione 4.1.5 per una trattazione più dettagliata del costo dell'implementazione della generica iterazione k del metodo $\mathcal{SDA}_{ls}$.

4.1.2 Troncamento e compressione

Come più volte sottolineato, l'idea fondamentale dell'algortimo illustrato nella sezione precedente è quella di sfruttare la $\mathcal{SMWF}$ e le formule ricorsive (4.8) per preservare le proprietà di struttura presenti nei termini iniziali delle successioni. Dalle (4.15)-(4.20), è però evidente che la dimensione delle correzioni di rango ha crescita potenzialmente esponenziale. Tale rapidissima crescita potrebbe, in pochissime iterazioni, 'bruciare' il vantaggio di avere matrici di rango basso, non rendendo più conveniente l'utilizzo del metodo $\mathcal{SDA}_{ls}$ se comparato con il metodo $\mathcal{SDA}$ standard. Tuttavia, come ben osservato dagli autori dell'algoritmo, poiché per le (4.3) le matrici E_k e F_k diventano sempre più piccole in norma, le correzioni di rango definite dalle (4.18), (4.19), (4.15), (4.16), aumentano sì di dimensione ma hanno componenti di norma sempre minore. È pertanto sperabile che tali correzioni non siano abbiano numericamente rango pieno. Il procedimento di **troncamento e compressione** delle matrici B_{ik} e C_{ik} adotta le precedenti osservazioni per cercare di limitare la crescita esponenziale delle dimensioni di tali matrici a patto, però, di perdere accuratezza nelle soluzioni effettivamente calcolate.

Siano, quindi,

- τ^B, τ^C numeri reali sufficientemente piccoli che definiscono la soglia di tolleranza del troncamento,
- $l_{max}, h_{max} \ll n, m$ limiti superiori della compressione.

Utilizzando la fattorizzazione QR con pivoting, è possibile individuare la seguente decomposizione delle matrici B_{ik} e C_{ik} per $i = 1, 2$:

$$B_{1k} = Q_{1k}^B M_{1k}^B + V_{1k}^B, \qquad B_{2k} = Q_{2k}^B M_{2k}^B + V_{2k}^B, \tag{4.21}$$

$$C_{1k} = Q_{1k}^C M_{1k}^C + V_{1k}^C, \qquad C_{2k} = Q_{2k}^C M_{2k}^C + V_{2k}^C, \tag{4.22}$$

con

- $Q_{1k}^B \in \mathbb{R}^{n \times r_{1k}^B}$, $Q_{2k}^B \in \mathbb{R}^{m \times r_{2k}^B}$, $Q_{1k}^C \in \mathbb{R}^{m \times r_{1k}^C}$, $Q_{2k}^C \in \mathbb{R}^{n \times r_{2k}^C}$ matrici unitarie,
- $M_{1k}^B \in \mathbb{R}^{r_{1k}^B \times l_k}$, $M_{2k}^B \in \mathbb{R}^{r_{2k}^B \times l_k}$, $M_{1k}^C \in \mathbb{R}^{r_{1k}^C \times h_k}$, $M_{2k}^C \in \mathbb{R}^{r_{1k}^C \times h_k}$, matrici triangolari superiori,

tali che

$$\|V_{ik}^B\| \leq \tau^B \qquad \|V_{ik}^C\| \leq \tau^C,$$

e

$$r_{ik}^B = \texttt{rank} B_{ik} \leq l_k \leq l_{max} \ll \bar{n},$$
$$r_{ik}^C = \texttt{rank} C_{ik} \leq h_k \leq h_{max} \ll \bar{n},$$

per $i = 1, 2$.

Sebbene teoricamente non sia possibile provare l'esistenza di tale decomposizione, scegliendo ragionevolmente i parametri τ^B, τ^C, l_{max}, h_{max} del procedimento di troncamento e compressione, è empiricamente possibile determinarla. Alla luce delle precedenti espressioni, si ha

$$G_k = B_{1k} R_k B_{2k}^T = Q_{1k}^B M_{1k}^B R_k (Q_{2k}^B M_{2k}^B)^T + O(\tau^B),$$
$$P_k = C_{1k} T_k C_{2k}^T = Q_{1k}^C M_{1k}^C T_k (Q_{2k}^C M_{2k}^C)^T + O(\tau^C),$$

pertanto, trascurando i termini di norma minore delle soglie τ^B, τ^C, si può porre

$$\begin{aligned} B_{ik} &:= Q^B_{ik}, \\ R_k &:= M^B_{1k} R_k (M^B_{2k})^T, \\ C_{ik} &:= Q^C_{ik}, \\ T_k &:= M^C_{1k} T_k (M^C_{2k})^T. \end{aligned} \tag{4.23}$$

La chiave di volta per il successo dell'algoritmo è, quindi, la ricerca del giusto compromesso tra accuratezza della soluzione e efficienza dell'algoritmo sia in termini di costo computazionale che di memoria richiesta. Nelle sperimentazioni svolte dagli autori, come specificato nel paragrafo 4.2 è più volte sottolineata l'importanza di tale compromesso: scegliere soglie di tolleranza τ^B, τ^C troppo piccole o limiti dimensionali l_{max}, h_{max} elevati certamente implica un costo computazionale notevole, mentre non comporta necessariamente grandi miglioramenti dell'accuratezza delle soluzioni. È, pertanto fondamentale, per una corretta interpretazione del metodo $\mathcal{SDA}_{ls}$, scegliere i parametri del procedimento di troncamento e compressione proprio in base all'ottimizzazione di tale compromesso. Altra conseguenza importante di tale meccanismo è che le soluzioni minimali $X_{\min}$, Y_{min} della $\mathcal{NARE}$ e della sua duale, per la (4.3), hanno per costruzione numericamente rango basso.

4.1.3 Controllo di convergenza e residuale relativo

L'ultima fase del metodo $\mathcal{SDA}_{ls}$ è quella di individuare delle condizioni di arresto per le iterazioni dell'algoritmo e dunque dei criteri che assicurino, con buon margine di accuratezza, la convergenza del metodo. Tale controllo, ovviamente, non deve avere un costo computazionale superiore alle $O(\bar{n})$ operazioni per passo, per non vanificare gli 'sforzi' fatti nei passaggi precedenti.

Siano

$$\delta^P_k := \|P_k - P_{k-1}\| \qquad \delta^G_k := \|G_k - G_{k-1}\|,$$

allora, per la (4.8), si ha

$$\begin{aligned} P_k - P_{k-1} &= C_{1k} T_k C^T_{2k} - C_{1,k-1} T_{k-1} C^T_{2,k-1} = \tilde{C}_{1k} \tilde{T}_k \tilde{C}^T_{2k} \\ G_k - G_{k-1} &= B_{1k} R_k B^T_{2k} - B_{1,k-1} R_{k-1} B^T_{2,k-1} = \tilde{B}_{1k} \tilde{R}_k \tilde{B}^T_{2k} \end{aligned}$$

con

$$\tilde{C}_{1k} := \begin{pmatrix} C_{1k} & C_{1,k-1} \end{pmatrix} \qquad \tilde{T}_k := \begin{pmatrix} T_k & 0 \\ 0 & -T_{k-1} \end{pmatrix} \qquad \tilde{C}_{2k} := \begin{pmatrix} C_{2k} & C_{2,k-1} \end{pmatrix},$$

$$\tilde{B}_{1k} := \begin{pmatrix} B_{1k} & B_{1,k-1} \end{pmatrix} \qquad \tilde{R}_k := \begin{pmatrix} R_k & 0 \\ 0 & -R_{k-1} \end{pmatrix} \qquad \tilde{B}_{2k} := \begin{pmatrix} B_{2k} & B_{2,k-1} \end{pmatrix}.$$

Applicando alle matrici di basso rango $\tilde{B}_{ik}$, $\tilde{C}_{ik}$ il medesimo procedimento di compressione (senza troncamento) della sezione precedente e modificando opportunamente i nuclei $\tilde{T}_k$ e $\tilde{R}_k$, si ottengono le seguenti stime

$$\begin{aligned} \delta^P_k &= \|\tilde{C}_{1k} \tilde{T}_k \tilde{C}^T_{2k}\| = \|\tilde{T}_k\|, \\ \delta^G_k &= \|\tilde{B}_{1k} \tilde{R}_k \tilde{B}^T_{2k}\| = \|\tilde{R}_k\|, \end{aligned}$$

con un notevole risparmio di operazioni, essendo le dimensioni delle matrici $\tilde{T}_k$ e $\tilde{R}_k$ di gran lunga inferiori rispetto a quelle delle matrici P_k e G_k. Tale stima permette quindi di determinare l'ordine di convergenza dell'algoritmo senza però oltrepassare la soglia massima di $O(\bar{n})$ operazioni elementari scelta dagli autori come massimale per la buona riuscita dell'algoritmo.

Un altro parametro utile a stimare la buona riuscita delle implementazioni è il **residuale relativo** definito da

$$r_k := \frac{e_k}{t_0 + t_{1k} + t_{2k}}$$

dove

- $e_k := \|\mathcal{R}(P_k)\| = \|C + P_kA + DP_k - P_kBP_k\|$,
- $t_0 := \|C\|$,
- $t_{1k} := \|P_kA + DP_k\|$,
- $t_{2k} := \|P_kBP_k\|$.

È possibile calcolare ciascuno dei termini sopra menzionati applicando tecniche analoghe a quelle adottate per δ_k^P e δ_k^G. Per il termine t_0, si osservi che $C = C_1TC_2^T$, dunque, con il procedimento di compressione ai coefficienti C_1 e C_2 e con la relativa modifica del nucleo T, si pone semplicemente

$$t_0 = \|T\|.$$

Per il termine t_{1k} si osservi che

$$P_kA + DP_k = C_{1k}T_kC_{2k}^TA + DC_{1k}T_kC_{2k}^T =: \hat{C}_{1k}\hat{T}_k\hat{C}_{2k}^T,$$

con

$$\hat{C}_{1k} := \begin{pmatrix} DC_{1k} & C_{1k} \end{pmatrix}, \qquad \hat{T}_k := \begin{pmatrix} 0 & T_k \\ T_k & 0 \end{pmatrix}, \qquad \hat{C}_{2k} := \begin{pmatrix} A^TC_{2k} & C_{2k} \end{pmatrix}.$$

Utilizzando il solito procedimento di compressione, è sufficiente calcolare

$$t_{1k} = \|\hat{T}_k\|$$

con un notevole risparmio in termini di costo computazionale.

Il termine t_{2k} si tratta in modo analogo:

$$P_kBP_k = C_{1k}T_kC_{2k}^TB_1RB_2^TC_{1k}T_kC_{2k}^T =: C_{1k}\breve{T}_kC_{2k}^T,$$

con

$$\breve{T}_k := T_kC_{2k}^TB_1RB_2^TC_{1k}T_k,$$

e quindi, dopo i consueti cambiamenti,

$$t_{2k} = \|\breve{T}_k\|.$$

Infine, per e_k, si ha

$$\begin{aligned} C + P_kA + DP_k - P_kBP_k &= C_1TC_2^T + \hat{C}_{1k}\hat{T}_k\hat{C}_{2k}^T - C_{1k}\breve{T}_kC_{2k}^T \\ &= \bar{C}_{1k}\bar{T}_k\bar{C}_{2k}^T, \end{aligned}$$

con

$$\bar{C}_{1k} := \begin{pmatrix} \hat{C}_{1k} & C_1 \end{pmatrix}, \qquad \bar{T}_k := \begin{pmatrix} 0 & T_k & 0 \\ T_k & -\breve{T}_k & 0 \\ 0 & 0 & T \end{pmatrix}, \qquad \bar{C}_{2k} := \begin{pmatrix} \hat{C}_{2k} & C_2 \end{pmatrix},$$

pertanto, applicando il compressione, si pone semplicemente

$$e_k = \|\bar{T}_k\|.$$

Si conclude, quindi, che tale formulazione, sebbene abbastanza elaborata, permette di controllare l'andamento della convergenza e dei residuali relativi senza compromettere il costo computazionale dell'algoritmo di $O(\bar{n})$ operazioni aritmetiche per iterazione.

4.1.4 Propagazione dell'errore

Il procedimento di troncamento e compressione presentato nella sezione 1.4.2 è indubbiamente un vantaggio in quanto permette di mantenere le dimensioni delle correzioni di rango definite dal metodo $\mathcal{SDA}_{ls}$ sotto una determinata soglia a patto di perdere 'qualcosa' in accuratezza. È certamente possibile, però, che nel corso delle iterazioni, tali approssimazioni si amplifichino tanto da rendere la soluzione del tutto inattendibile. Gli autori accanto alla descrizione del metodo, presentano, quindi, un'analisi della propagazione di tali arrotondamenti, mostrando che nel corso delle iterazioni le approssimazioni si mantengano dello stesso ordine senza dare luogo a degenerazioni. Nel presente lavoro di tesi si è arricchito tale studio individuando una costante che indichi il fattore di amplificazione.

Siano E'_k, P'_k, F'_k, G'_k le matrici calcolate dal metodo $\mathcal{SDA}_{ls}$ dopo il procedimento di troncamento e compressione e siano

$$\Sigma_k^E := E'_k - E_k \qquad \Sigma_k^P := P'_k - P_k \qquad \Sigma_k^F := F'_k - F_k \qquad \Sigma_k^G := G'_k - G_k$$

gli errori maturati al passo k-esimo a seguito di tale procedimento.

Siano

$$\varepsilon_k := \max\{\, \|\Sigma_k^E\|, \|\Sigma_k^P\|, \|\Sigma_k^G\|, \|\Sigma_k^F\| \,\}, \qquad \tau := \max\{\, \tau^B, \tau^C \,\},$$

allora è evidente che $\varepsilon_0 = O(\tau)$.

Per studiare la propagazione dell'errore si introducono le matrici

$$\begin{aligned} &N_{1k} := I_n - G_k P_k \qquad N'_{1k} := I_n - G'_k P'_k \qquad \Delta_{1k} := G_k\Sigma_k^P + \Sigma_k^G P_k + \Sigma_k^G\Sigma_k^P, \\ &N_{2k} := I_m - P_k G_k \qquad N'_{2k} := I_m - P'_k G'_k \qquad \Delta_{2k} := P_k\Sigma_k^G + \Sigma_k^P G_k + \Sigma_k^P\Sigma_k^G. \end{aligned}$$

Allora si hanno le seguenti relazioni

$$\begin{aligned} \Sigma_{k+1}^E &= E'_{k+1} - E_{k+1} = E'_k {N'_{1k}}^{-1} E'_k - E_k N_{1k} E_k \\ &= (E_k + \Sigma_k^E)(I_n - (G_k + \Sigma_k^G)(P_k + \Sigma_k^P))^{-1}(E_k + \Sigma_k^E) - E_k N_{1k}^{-1} E_k \\ &= (E_k + \Sigma_k^E)(N_{1k} - \Delta_{1k})^{-1}(E_k + \Sigma_k^E) - E_k N_{1k}^{-1} E_k \\ &= (E_k + \Sigma_k^E)N_{1k}^{-1}(I_n + \Delta_{1k}N_{1k}^{-1})(E_k + \Sigma_k^E) - E_k N_{1k}^{-1} E_k + o(\varepsilon_k) \\ &= E_k N_{1k}^{-1}\Delta_{1k}N_{1k}^{-1} E_k + E_k N_{1k}^{-1}\Sigma_k^E + \Sigma_k^E N_{1k}^{-1} E_k + o(\varepsilon_k); \end{aligned}$$

$$\begin{aligned} \Sigma_{k+1}^F &= F'_{k+1} - F_{k+1} = F'_k {N'_{2k}}^{-1} F'_k - F_k N_{2k} F_k \\ &= (F_k + \Sigma_k^F)(I_m - (P_k + \Sigma_k^P)(G_k + \Sigma_k^G))^{-1}(F_k + \Sigma_k^F) - F_k N_{2k}^{-1} F_k \\ &= (F_k + \Sigma_k^F)(N_{2k} - \Delta_{2k})^{-1}(F_k + \Sigma_k^F) - F_k N_{2k}^{-1} F_k \\ &= (F_k + \Sigma_k^F)N_{2k}^{-1}(I_m + \Delta_{2k}N_{2k}^{-1})(F_k + \Sigma_k^F) - F_k N_{2k}^{-1} F_k + o(\varepsilon_k) \\ &= F_k N_{2k}^{-1}\Delta_{2k}N_{2k}^{-1} F_k + F_k N_{2k}^{-1}\Sigma_k^F + \Sigma_k^F N_{2k}^{-1} F_k + o(\varepsilon_k); \end{aligned}$$

$$\begin{aligned} \Sigma_{k+1}^P &= P'_{k+1} - P_{k+1} = P'_k + F'_k {N'_{2k}}^{-1} P'_k E'_k - P_k - F_k N_{2k}^{-1} P_k E_k \\ &= \Sigma_k^P + (F_k + \Sigma_k^F)(I_m - (P_k + \Sigma_k^P)(G_k + \Sigma_k^G))^{-1}(P_k + \Sigma_k^F)(E_k + \Sigma_k^E) - F_k N_{2k}^{-1} P_k E_k \\ &= \Sigma_k^P + (F_k + \Sigma_k^F)(N_{2k} - \Delta_{2k})^{-1}(P_k + \Sigma_k^P)(E_k + \Sigma_k^E) - F_k N_{2k}^{-1} P_k E_k \\ &= \Sigma_k^P + (F_k + \Sigma_k^F)N_{2k}^{-1}(I_m + \Delta_{2k}N_{2k}^{-1})(P_k + \Sigma_k^P)(E_k + \Sigma_k^E) - F_k N_{2k}^{-1} P_k E_k + o(\varepsilon_k) \\ &= \Sigma_k^P + F_k N_{2k}^{-1}\Delta_{2k}N_{2k}^{-1} P_k E_k + F_k N_{2k}^{-1} P_k\Sigma_k^E + F_k N_{2k}^{-1}\Sigma_k^P E_k + \Sigma_k^F N_{2k}^{-1} P_k E_k + o(\varepsilon_k); \end{aligned}$$

$$\begin{aligned} \Sigma_{k+1}^G &= G'_{k+1} - G_{k+1} = G'_k + E'_k {N'_{1k}}^{-1} G'_k F'_k - G_k - E_k N_{2k}^{-1} G_k F_k \\ &= \Sigma_k^G + (E_k + \Sigma_k^E)(I_n - (G_k + \Sigma_k^G)(P_k + \Sigma_k^P))^{-1}(G_k + \Sigma_k^G)(F_k + \Sigma_k^F) - E_k N_{1k}^{-1} G_k F_k \\ &= \Sigma_k^G + (E_k + \Sigma_k^E)(N_{1k} - \Delta_{1k})^{-1}(G_k + \Sigma_k^G)(F_k + \Sigma_k^F) - E_k N_{1k}^{-1} G_k F_k \\ &= \Sigma_k^G + (E_k + \Sigma_k^E)N_{1k}^{-1}(I_n + \Delta_{1k}N_{1k}^{-1})(G_k + \Sigma_k^G)(F_k + \Sigma_k^F) - E_k N_{1k}^{-1} G_k F_k + o(\varepsilon_k) \\ &= \Sigma_k^G + E_k N_{1k}^{-1}\Delta_{1k}N_{1k}^{-1} G_k F_k + E_k N_{1k}^{-1} G_k\Sigma_k^F + E_k N_{1k}^{-1}\Sigma_k^G F_k + \Sigma_k^E N_{1k}^{-1} G_k F_k + o(\varepsilon_k). \end{aligned}$$

Dopo questa laboriosa serie di calcoli, si hanno gli 'ingredienti' necessari per ottenere le stime sulla propagazione volute. Siano

- $\delta_k := \max\{\, \|E_k\|, \|F_k\| \,\}$,
- $\gamma_k := \delta_k \cdot \max\{\, \|N_{1k}^{-1}\|, \|N_{2k}^{-1}\| \,\}$,
- $\alpha_k := \max\{\, \|P_k\|, \|G_k\| \,\}$,

e si osservi che

$$\|\Delta_{1k}\|, \|\Delta_{2k}\| \leq 2\varepsilon_k\alpha_k + O(\varepsilon_k),$$

allora

$$\varepsilon_{k+1} \leq \max\{\, 2\gamma_k(1+\alpha_k\gamma_k), 1+\gamma_k(2\alpha_k + 2\gamma_k\alpha_k^2 + \delta_k) \,\}\,\varepsilon_k + o(\varepsilon_k). \tag{4.24}$$

Si ricordi che, per il teorema 3.1.6 e per il teorema 3.1.7, sono verificate le seguenti stime asintotiche

$$\lim_{k\to\infty} E_k = 0, \qquad \lim_{k\to\infty} P_k = X_{min}, \qquad \lim_{k\to\infty} F_k = 0 \qquad \lim_{k\to\infty} G_k = Y_{min},$$

in particolare si ha

$$\alpha_k \leq \alpha \qquad \gamma_k, \delta_k \leq \sigma^{2^k}$$

dove α è una costante limitata e $\sigma < 1$.

Alla luce delle precedenti stime, dalla (4.24) si ottiene

$$\varepsilon_{k+1} \leq (1 + \xi\sigma^{2^k})\varepsilon_k,$$

dunque l'errore al passo $(k+1)$-esimo rimane dello stesso ordine dell'errore al passo k-esimo. Inoltre, è possibile estendere quanto fatto dagli autori del metodo $\mathcal{SDA}_{ls}$ calcolando la costante che esprime il fattore di amplificazione dell'errore. Si osservi, infatti, che

$$\varepsilon_k \leq \left(\prod_{i=0}^{k-1}(1+\xi\sigma^{2^i})\right)\varepsilon_0 \leq \left(\sum_{i=0}^{\bar{k}}\xi^i\sigma^{2i}\right)\varepsilon_0,$$

pertanto

$$\lim_{k\to\infty}\varepsilon_k \leq \frac{1}{1-\xi\sigma^2}\varepsilon_0 \leq \frac{1}{1-\xi\sigma^2}\tau + O(\tau^2).$$

Dalla stima precedente, quindi, è possibile evincere che l'errore dovuto al procedimento di troncamento e compressione, non si amplifica nel corso delle iterazioni ma rimane del medesimo ordine del parametro τ scelto come soglia di accuratezza voluto.

4.1.5 Costo computazionale

Come più volte sottolineato nelle sezioni precedenti, il principale vantaggio del metodo $\mathcal{SDA}_{ls}$ è quello di abbattere il costo computazionale da $O(n^3)$ operazioni elementari per passo, ad un decisamente meno oneroso costo di $O(n)$ operazioni. Per ottenere tale riduzione, occorre svolgere opportunamente operazioni quali prodotti matriciali ed inversioni, facendo particolare attenzione anche all'ordine con cui vengono eseguite. È, inoltre, importante sfruttare ricorsivamente le equazioni (4.8) senza calcolare esplicitamente le matrici E_k e F_k ad ogni iterazione, in modo da preservare le proprietà di struttura inizialmente presenti e rendere meno pesante il calcolo dei prodotti matrice-vettore.

Per semplicità, si supponga $n = m$, e si facciano le seguenti assunzioni

- il calcolo dei prodotti

$$AR, \qquad A^TR, \qquad DR, \qquad D^TR \tag{4.25}$$

e la risoluzione deii sistemi

$$A_\gamma Z = R, \qquad A_\gamma^T Z = R, \qquad D_\gamma Z = R, \qquad D_\gamma^T Z = R, \tag{4.26}$$

con $R \in \mathbb{R}^{n\times s}$ comportano un costo computazionale di $c_\gamma ns$ operazioni elementari,

- il calcolo della fattorizzazione QR con pivoting di una matrice $n \times s$ ha un costo computazionale di $4ns^2$ operazioni elementari.

Alla luce delle precedenti assunzioni, è possibile calcolare nel dettaglio il numero di operazioni richieste dall'implementazione del metodo $\mathcal{SDA}_{ls}$. Sono di seguito elencate le operazioni necessarie per inizializzare l'algoritmo se si utilizza una trasformazione affine $\mathcal{A}_\alpha$ corredate di relativo costo computazionale:

- il calcolo della matrice D_α comporta un costo di n operazioni,
- le matrici $B_{1,0}$ e $C_{2,0}$ non necessitano di alcun calcolo, mentre per il calcolo delle matrici $B_{2,0}$ e $C_{1,0}$ occorre risolvere due sistemi del tipo (4.26), con un costo di $2c_\gamma n$ operazioni,
- per il calcolo del termine $t_0 = \|C\|$, con il procedimento descritto nella sezione 4.1.3, occorre realizzare le fattorizzazioni QR delle matrici C_1 e C_2 e di modificare la matrice T, è dunque richiesto un costo computazionale di $8nh^2$ operazioni.

Utilizzando una trasformazione affine, quindi, il costo per inizializzare il metodo $\mathcal{SDA}_{ls}$ è di $(1 + 2c_\gamma + 8h^2)n$ operazioni elementari.

Se si adotta, invece, una trasformazione di Cayley $\mathcal{C}_\gamma$ si rendono necessarie le seguenti operazioni con relativo costo computazionale:

- il calcolo delle matrici A_γ, D_γ ha un costo di $2n$ operazioni elementari,
- per calcolare le matrici $B_{1,0}$ e $C_{1,0}$ occorre risolvere due sistemi (4.26), sono pertanto richieste $c_\gamma n(l + h)$ operazioni; per il calcolo delle matrici $B_{2,0}$ e $C_{2,0}$, sfruttando la (4.6) e la (4.7), occorre
  - risolvere i sistemi $D_\gamma^T Z = B_2$, $A_\gamma^T Z = C_2$ con costo computazionale totale di $c_\gamma(l + h)n$,
  - calcolare due prodotti del tipo MN^T e NM^T con $M \in \mathbb{R}^{l\times n}$ e $N \in \mathbb{R}^{h\times n}$ il cui costo computazionale è di $4lhn$ operazioni,
  - calcolare due prodotti di matrici $l \times l$ per $l \times n$ e $h \times h$ per $h \times n$, con costo computazionale di $2(l^2 + h^2)n$ operazioni,
- per determinare il termine $t_0 = \|C\|$, in analogia con quanto esposto per le trasformazioni affini, si ha un costo computazionale di $8nh^2$ operazioni.

Il costo per inizializzare il metodo $\mathcal{SDA}_{ls}$ a partire da una trasformazione di Cayley è, quindi, di circa $2(1 + c_\gamma(l + h) + 2lh + l^2 + 5h^2)n$ operazioni elementari.

Il calcolo delle operazioni necessarie alla generica iterazione k è abbastanza elaborato, questo perché, come spiegato in precedenza, per non appesantire l'algoritmo si preferisce non calcolare esplicitamente le matrici E_k e F_k, ma applicare ricorsivamente la (4.8). Come appare evidente dalle formule della sezione 4.1.1, infatti, gran parte del costo computazionale all'iterazione k-esima è addebitabile ai prodotti E_kB_{1k}, $F_k^TB_{2k}$, F_kC_{1k}, $E_k^TC_{2k}$. Sfruttando la definizione ricorsiva delle matrici E_k e F_k, il costo computazionale per il calcolo di tali prodotti rimane dell'ordine di $O(n)$ operazioni, ma si complicano, almeno dal punto di vista meramente teorico, i lunghi calcoli da svolgere. Tale impostazione, però, impone la memorizzazione delle matrici B_{ik}, C_{ik}, R_k, T_k per $j \leq k$. Si ha, pertanto, una riduzione del costo computazionale, ma un aggravio dell'impiego di memoria richiesto. Le operazioni da eseguire con il relativo costo computazionale sono riportate nell'elenco sottostante

- per calcolo delle matrici $B_{i,k+1}$, $C_{i,k+1}$ è preferbile, come già spiegato, non calcolare esplicitamente i prodotti E_kB_{1k}, $F_k^TB_{2k}$, F_kC_{1k}, $E_k^TC_{2k}$. Utilizzando ricorsivamente la (4.8), senza entrare nel dettaglio delle singole operazioni, si ha un costo computazionale complessivo di

$$\left(2^{k+2}c_\gamma(l_k + h_k) + 2(6h_k + 1)J_k\right) n,$$

con $J_k := \sum_{i=1}^{k}(l_i + h_i)$;

- per il calcolo di R_{k+1} e T_{k+1} occorre svolgere quattro prodotti matriciali del tipo MN^T e NM^T con $M \in \mathbb{R}^{l_k \times n}$ e $N \in \mathbb{R}^{h_k \times n}$ il cui costo computazionale è di $8l_k h_k n$ operazioni;

- per il calcolo delle matrici $E_{i,k+1}$ e $F_{i,k+1}$, sfruttando i calcoli già svolti, si rendono necessari esclusivamente due prodotti matriciali del tipo MN^T e NM^T con $M \in \mathbb{R}^{l_k \times n}$ e $N \in \mathbb{R}^{h_k \times n}$ con costo computazionale di $4l_k h_k n$ operazioni;

- per il procedimento di compressione e troncamento è necessario calcolare le fattorizzazione QR di due matrici $n \times l_k$ e di due matrici $n \times h_k$, il cui costo computazionale è di $8(l_k^2 + h_k^2)n$ operazioni elementari, la riorganizzazione delle matrici R_{k+1}, T_{k+1} ha costo marginale;

- il calcolo del residuale relativo si compone come segue

  - il termine t_0 è stato calcolato in sede di inizializzazione,
  - il termine t_{1k} necessita del calcolo di due prodotti del tipo (4.25) che, per le assunzioni sopra menzionate, hanno un costo computazionale di $2c_\gamma h_k n$ operazioni, e di due fattorizzazioni QR di matrici $n \times 2h_k$ il cui costo computazionale è di $32h_k^2 n$ operazioni,
  - per il termine t_{2k} occorre svolgere due prodotti di matrici $h_k \times n$ per $n \times l$ con costo di $4h_k ln$ operazioni, e calcolare due fattorizzazioni QR di matrici $n \times h_k$ al costo di $8h_k^2 n$ operazioni,
  - per il termine e_k è sufficiente individuare la solita fattorizzazione QR di matrici $n \times (2h_k + h)$, il cui costo è di $8(2h_k + h)^2 n$ operazioni.

Sommando le varie componenti, si ottiene quindi, un totale di

$$\left(2^{k+2} c_\gamma (l_k + h_k) + 2(6h_k + 1)J_k + 12 l_k h_k + 8l_k^2 + 48h_k^2 + c_\gamma h_k + 4h_k l + 8(2h_k + h)^2\right) n$$

operazioni elementari per passo.

Si osservi che, essendo l_k, h_k maggiorati dai parametri l_{max} e h_{max}, il costo computazionale al passo k-esimo è sostanzialmente dominato dal coefficiente 2^{k+2}. Tale relazione, dunque, implica che ad ogni passo il numero di operazioni richieste è all'incirca il doppio delle operazioni svolte al passo precedente.

Dal momento che si lavora con matrici di grandi dimensioni, oltre al costo computazionale, un altro parametro importante per giudicare la bontà di un algoritmo è il 'volume' di memoria richiesta. Per il metodo $\mathcal{SDA}_{ls}$, stanti le solite ipotesi sulla dimensione delle matrici, non sarebbe possibile registrare tutte le matrici coinvolte, in quanto vi sarebbe una richiesta di memoria eccessiva per le potenzialità di un 'normale' calcolatore. Occorre, dunque, cercare di limitare il più possibile l'impiego di memoria. Per l'implementazione del metodo $\mathcal{SDA}_{ls}$ e per poter correttamente utilizzare le formulazioni ricorsive (4.8), gli autori suggeriscono di registrare al k-esimo esclusivamente le matrici B_{jk}, C_{jk}, R_j, T_j per $j \leq k$, con un consumo totale di $2J_k n$ unità elementari di memoria.

4.2 Implementazioni

Nel presente paragrafo vengono presentati i codici delle parti più significative del metodo $\mathcal{SDA}_{ls}$ e sono illustrati i risultati delle sperimentazioni effettuate. Come ampiamente spiegato la buona riuscita del metodo $\mathcal{SDA}_{ls}$ è sostanzialmente basata sull'ottimizzazione del compromesso tra efficienza dell'algoritmo e accuratezza della soluzione. Nelle implementazioni è dunque sottolineato tale aspetto valutando gli effetti dei parametri τ^B, τ^C e l_{max}, h_{max} su

- errore relativo,

- crescita delle correzioni di rango,

- tempi di utilizzo della CPU.

Per alleggerire la lettura e la comprensione dell'algoritmo si è scelto di riportare lo pesudocodice del metodo $\mathcal{SDA}_{ls}$ in luogo del codice MATLAB (si osservi, tra l'altro, la colorazione differente). Lo pseudocodice ricalca i passi illustrati nel paragrafo precedente:

1. inizializzazione,
2. calcolo ricorsivo delle matrici,
3. troncamento e compressione,
4. calcolo del residuale relativo,
5. verifica della condizione d'arresto.

Listing 4.1: Metodo $\mathcal{SDA}_{ls}$ pseudocodice.

```
Input:      A, D;
            B_1, B_2, R;
            C_1, C_2, T;
            tau;
            l_m, h_m;
            eps;

 % tau = parametro di troncamento
 % l_m, h_m = parametri di compressione
 % eps = soglia per l'errore relativo

 Output:   B_(1n), B_(2n), R_n;
           C_(1n), C_(2n), T_n;

 % X := B_(1n)*R_n*B_(2n)' soluzione della NARE
 % Y := C_(1n)*T_n*C_(2n)' soluzione della NARE duale


 Inizializzare il metodo come espresso dal codice 4.2;
 Calcolare il parametro t_0;
 Porre k=0, rel= 2eps

 % k = numero iterazione
 % rel = errore relativo, parametro d'arresto del metodo

 while rel < eps

     calcolare ricorsivamente le matrici
     B_(1,k+1), B_(2,k+1), R_(k+1);
     C_(1,k+1), C_(2,k+1), T_(k+1);
     G_(1,k+1), G_(2,k+1);
     P_(1,k+1), P_(2,k+1);

     [Q_(Bi), M_(Bi)] = tron_compr(B_(i,k), tau, l_m);    i=1,2
     [Q_(Ci), M_(Ci)] = tron_compr(C_(i,k), tau, h_m);    i=1,2

     Modificare R_k, T_k;

     Calcolare e_0, t_1k, t_2k;

     k = k+1;
 end
```

Il codice sottostante illustra l'inizializzazione del metodo mediante una trasformazione affine. Si osservi che nel codice è calcolata esplicitamente l'inversa della matrice D_α, mentre nelle implementazioni di seguito riportate, si è calcolata tale inversa sfuttando la particolare struttura della matrice D_α.

Listing 4.2: Metodo $\mathcal{SDA}_{ls}$ inizializzazione con trasformazione affine.

```
%  il frammento di codice illustra l'inizializzazione del metodo
%  utilizzando una trasforazione affine

n = size(A,1);
m = size(D,1);
l = size(B,2);
h = size(C,2);

alpha = 1/max(diag(A));

Da = alpha*D + eye(m);

iDa = inv(Da);

E = alpha*A - eye(n) + alpha^2*B1*R*B2'*iDa*C1*T*C2';
P = -alpha*iDa*C1*T*C2';
F = -iDa;
G = alpha*B1*R*B2'*iDa;
```

Il codice 4.3 traduce in 'linguaggio macchina' il procedimento di troncamento e compressione. Alla base del mini-programma vi è la funzione MATLAB `qr` la quale produce la fattorizzazione QR con pivoting delle correzioni di rango. Si osservi che la condizione d'arresto del ciclo `while` interpreta le richieste sulla decomposizione illustrate nella sezione 4.1.2.

Listing 4.3: Processo di troncamento e compressione.

```
function[Q, R] = tron_compr(U, tau, l_m)

% il programma calcola il risultato del procedimento di troncamento e
% compressione della matrice U con
%
%  - tau parametro troncamento,
%  - l_m parametro compressione

[Q,R,E] = qr(U,0);
n = size(U,1);
l = size(U,2);

r = 1;
i = 1;

while (i < min(l,l_m + 1))
    if abs(R(i,i)) > tau
        i = i+1;
    else
        r = i-1;
        i = n+1;
    end
end
Q = Q(:,1:r);
R = R(1:r,:);
R = R*E';
```

Esempio 4.2.1. *Si consideri una $\mathcal{NARE}$ che descrive il moto di particelle lungo un'asta omogenea (cfr. paragrafo 1.2). Si supponga che*

- *i coefficienti della $\mathcal{NARE}$ siano matrici quadrate di dimensione $n = 1024$,*
- *le matrici R e T abbiano rango $l = h = 2$.*

Sono svolte due sperimentazioni numeriche con valori diversi dei parametri che definiscono il procedimento di troncamento e compressione. Nella prima implementazione si è scelto

$$\tau^B, \tau^C \leq 1.e-10 \qquad l_{max}, h_{max} = 110,$$

i risultati ottenuti sono riassunti nella tabella sottostante

iterazione	*errore relativo*	*rango R_k*	*rango T_k*	*tempo CPU*
1	$9.0579e-04$	2	2	$9.133e-02$
2	$6.3235e-04$	4	4	$2.784e-01$
3	$5.4688e-05$	8	8	$3.525e-01$
4	$9.6488e-06$	15	15	$4.576e-01$
5	$3.7059e-06$	23	24	$8.571e-01$
6	$4.8537e-07$	35	36	$1.402e+00$
7	$1.4188e-07$	52	55	$2.217e+00$
8	$4.1573e-08$	81	86	$7.922e+00$
9	$9.5949e-09$	94	98	$1.557e+01$
10	$3.5711e-10$	94	98	$4.491e+01$
11	$9.3399e-12$	94	98	$9.787e+01$
12	$7.5774e-13$	94	98	$1.531e+02$

Nella seconda sperimentazione si scelgono le seguenti soglie

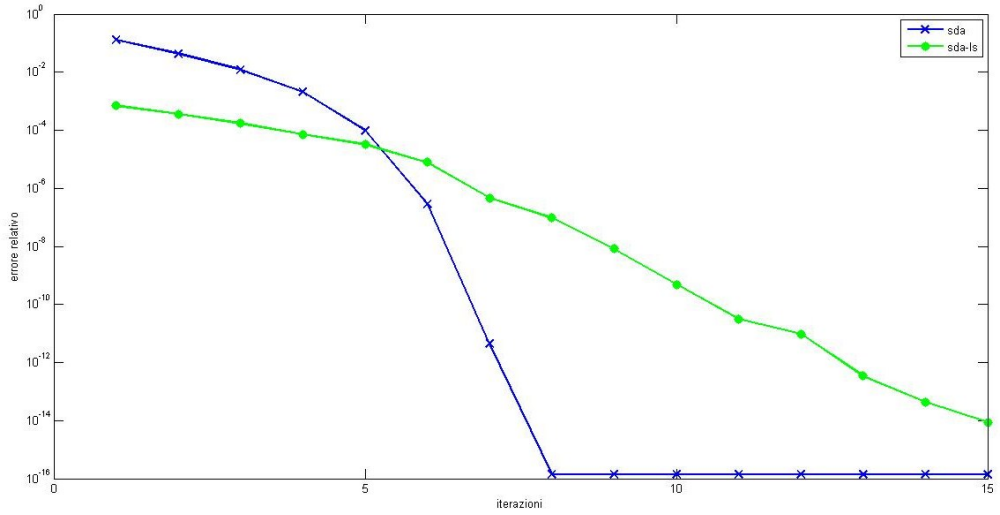
$$\tau^B, \tau^C \leq 1.e-12 \qquad l_{max}, h_{max} = 120,$$

I risultati ottenuti sono riassunti nella tabella sottostante

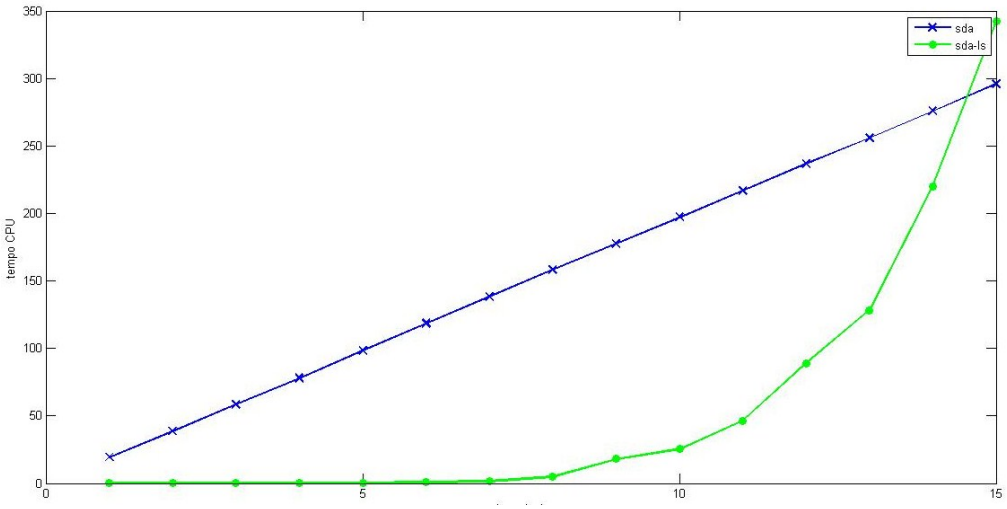
iterazione	*errore relativo*	*rango R_k*	*rango T_k*	*tempo CPU*
1	$6.9222e-04$	2	2	$7.655e-02$
2	$3.5547e-04$	4	4	$1.868e-01$
3	$1.7118e-04$	8	8	$2.597e-01$
4	$7.0604e-05$	16	16	$3.455e-01$
5	$3.1832e-05$	29	29	$5.463e-01$
6	$7.7692e-06$	46	46	$7.093e-01$
7	$4.6171e-07$	72	72	$1.346e+00$
8	$9.7131e-08$	112	106	$4.760e+00$
9	$8.2345e-09$	120	120	$1.797e+01$
10	$4.8286e-10$	120	120	$2.550e+01$
11	$3.1709e-11$	120	120	$4.626e+01$
12	$9.5022e-12$	120	120	$8.889e+01$
13	$3.4446e-13$	120	120	$1.283e+02$
14	$4.3874e-14$	120	120	$2.197e+02$

Le tabelle precedenti mostrano l'importanza della scelta dei parametri del processo di troncamento e compressione: nel primo caso si sono scelte soglie τ^B, τ^C e l_{max}, h_{max} meno restrittive, ottenendo un sostanziale risparmio sui tempi d'esecuzione. Se, dunque, è accettabile un'accuratezza della soluzione di $1e-13$ è di gran lunga preferibile la prima scelta dei parametri piuttosto che la seconda. Viceversa, scegliere parametri più rigidi, comporta un aumento del costo computazionale a fronte di una precisione migliore.

Il grafico sottostante compara l'andamento dell'errore relativo lungo le iterazioni per il metodo $\mathcal{SDA}$ e per la sua versione large-scale (con i parametri adottati nella seconda sperimentazione)



Il seguente grafico, invece, illustra l'evoluzione del tempo di utilizzo della CPU al variare delle iterazioni per i metodi $\mathcal{SDA}$ e $\mathcal{SDA}_{ls}$. Si osservi che, essendo $n = 1024$ un numero non elevatissimo le migliorie offerte nella versione large-scale non sortiscono in tal caso effetti positivi.



4.3 Commenti e conclusioni

Il metodo $\mathcal{SDA}_{ls}$ è di certo un algoritmo ben collaudato e di estrema utilità per risolvere $\mathcal{NARE}$ della forma e delle dimensioni volute. Abbattere il costo computazionale da $O(n^3)$ operazioni elementari per passo a 'sole' $O(n)$ è indubbiamente un risultato notevole, tenuto conto della non eccessiva richiesta in termini di memoria. Illuminante, per la buona riuscita di tale metodo, è l'idea di utilizzare la $\mathcal{SMWF}$, in quanto per le (4.2), parte corposa del costo per il calcolo dei nuovi termini delle successioni $\{ E_k \}_{k\in\mathbb{N}}$, $\{ F_k \}_{k\in\mathbb{N}}$, $\{ P_k \}_{k\in\mathbb{N}}$, $\{ G_k \}_{k\in\mathbb{N}}$ è dovuta alla inversione delle matrici $I_n - G_k P_k$ $I_m - P_k G_k$, operazioni 'alleggerite' di molto proprio per l'astuto utilizzo della $\mathcal{SMWF}$.

Accanto agli indiscutibili aspetti positivi del metodo, vanno, a modesto parere dello scrivente, evidenziate alcune criticità nell'algoritmo. La questione che suscita maggiori perplessità è il procedimento di troncamento e compressione: la scelta dei parametri τ^B, τ^C, l_{max}, h_{max}, non appare suggerita con la dovuta dovizia di particolari. Le soglie dimensionali, soprattutto, paiono quasi parametri da porre a posteriori, a sperimentazione avvenuta, piuttosto che parametri al servizio della stessa. Inoltre, l'esistenza delle

decomposizioni (4.21), empiricamente e sperimentalmente comprovata, è enunciata quasi assiomaticamente, senza porre alcuna condizione sui parametri del procedeimento di compressione e troncamento nè tanto meno abbozzare una traccia di dimostrazione.

La stima sull'amplificazione dell'errore è trattata con superficialità, gli autori si limitano a mostrare che ad ogni passo l'errore si mantiene dello stesso ordine del passo precedente senza entrare ulteriormente nel merito. Si dice sostanzialmente che l'errore verifica una relazione del tipo

$$\varepsilon_{k+1} \leq (1 + c_k)\varepsilon_k$$

con $\lim_{k\to\infty} c_k = 0$ senza però verificare, come fatto nel presente elaborato, se la stima

$$\varepsilon_{k+1} \leq \prod_{i=1}^{k}(1 + c_i)\varepsilon_0,$$

avesse effettivamente senso per ogni valore di k, e dunque se l'amplificazione non portasse a fenomeni di degenerazione dell'errore.

Un altro punto poco chiaro è l'elaborato calcolo del costo computazionale, la scelta di non calcolare esplicitamente le matrici E_k e F_k ma svolgere ricorsivamente i prodotti matrice-vettore è indubbiamente una scelta vincente in quanto si sfrutta la struttura matriciale inizialmente presente, tuttavia impone un aumento esponenziale del numero di prodotti da eseguire, rendendo l'algoritmo meno immediato e di difficile lettura.

Si conclude, quindi, che il metodo $\mathcal{SDA}_{ls}$ è sicuramente un metodo valido per la trattazione di equazione di Riccati di grandi dimensioni e particolari proprietà strutturali, ma andrebbe rivisitato dal punto di vista teorico, approfondendo taluni aspetti non trattati con il formalismo ed il rigore del caso.

Capitolo 5

Metodo $\mathcal{CR}$ per equazioni di Riccati di grandi dimensioni

Indice

Il metodo di riduzione ciclica, come ampiamente descritto nel capitolo 3, presenta innumerevoli caratteristiche comuni al metodo $\mathcal{SDA}$. Entrambi i metodi, infatti, si basano su un procedimento di elevamento al quadrato degli autovalori, per il metodo $\mathcal{CR}$ di autovalori di polinomi matriciali quadratici, per il metodo $\mathcal{SDA}$ di autovalori di matrix pencil. Tra le proprietà comuni dei doubling algorithm vi è l'ordine di convergenza, quadratico per ciascuno dei metodi, ed il costo computazionale che si aggira per entrambi sulle $O(n^3)$ operazioni elementari per iterazione. Tale entità del costo computazionale suggerisce le medesime considerazioni già esposte nel capitolo precendente: per equazioni di Riccati i cui coefficienti hanno 'grandi dimensioni' il metodo $\mathcal{CR}$, basato su operazioni quali il calcolo di prodotti matrice-vettore o la risoluzione di sistemi lineari, risulta inapplicabile in quanto l'implementazione dell'algoritmo richiederebbe tempi d'esecuzione non ragionevoli.

Se, però, la $\mathcal{NARE}$ oggetto di studio ha sì grandi dimensioni ma presenta determinate caratteristiche strutturali, è naturale chiedersi se la modifiche apportate al metodo $\mathcal{SDA}$ dai metematici cinesi Chang-Yi Weng, Tiexiang Li, Eric King-wah Chu e Wen-Wei Lin possono in qualche modo essere adottate dal metodo $\mathcal{CR}$ con, ovviamente, i medesimi benefici in termine di riduzione del costo computazionale. Il presente lavoro di tesi si pone l'obiettivo di rispondere a tale interrogativo, illustrando una proposta di metodo per le $\mathcal{NARE}$ con le suddette caratterisitche dimensionali e strutturali, il *large-scale* $\mathcal{CR}$ ($\mathcal{CR}_{ls}$), che presenta un costo computazionale di $O(n)$ operazioni elementari per iterazione, con un cospicuo miglioramento rispetto alle prestazioni fornite dal metodo $\mathcal{CR}$ standard.

Il metodo $\mathcal{CR}_{ls}$, dunque, reinterpreta le modifiche ideate dagli autori del motodo $\mathcal{SDA}_{ls}$, ottenendo sostanzialmente benefici analoghi. Il presente capitolo descrive la portata di tali correzioni sia in termini teorici che in termini implementativi-applicativi.

Nel paragrafo 5.1 viene descritto il metodo $\mathcal{CR}_{ls}$ per $\mathcal{NARE}$ di grandi dimensioni, sono dapprima delineate le ipotesi sulla strutta dei coefficienti e vengono elencate le linee guida dell'algortimo. Sono, quindi, descritte le successioni definite per ricorrenza alla base del metodo ed è applicato l'analogo procedimento di *troncamento e compressione*

introdotto per il metodo $\mathcal{SDA}_{ls}$ per limitare la crescita delle correzioni di rango nel corso delle iterazioni. È di seguito illustrato un meccanismo per determinare un *controllo di convergenza* del metodo e delle condizioni d'arresto per l'algoritmo senza intaccare il limite massimo delle $O(n)$ operazioni elementari per passo. È poi analizzata l'evoluzione degli errori dovuti al procedimento di troncamento e compressione ed è esplicitamente calcolata una maggiorazione asintotica per gli errori generati da tali arrotondamenti. Sono, quindi, elencate nel dettaglio le stime sul costo computazionale delle singole operazioni che si rendono necessarie per l'inizializzazione e per l'implementazione della generica iterazione del metodo $\mathcal{CR}_{ls}$.

Nel paragrafo 5.2 sono illustrati i codici MATLAB degli algoritmi e sono presentate alcune sperimentazioni di esempi significativi. È particolarmente interessante osservare l'andamento della crescita delle correzioni di rango nel corso delle iterazioni e l'incidenza dei paramentri del procedimento di troncamento e compressione sull'errore relativo e sul tempo d'esecuzione richiesto. Per valori moderati delle dimensioni, vi è anche una comparazione tra i risultati ottenuti dal metodo $\mathcal{CR}$ ed i risultati ottenuti con il metodo $\mathcal{CR}_{ls}$.

Nel paragrafo 5.3 sono esposti alcuni commenti al metodo e sono tratte le conclusioni e le valutazioni sull'intero lavoro di tesi.

5.1 Metodo $\mathcal{CR}$ per $\mathcal{NARE}$ di grandi dimensioni

Si consideri l'equazione algebrica di Riccati non simmetrica

$$C + XA + DX - XBX = 0, \tag{5.1}$$

dove i coefficienti verificano le seguenti ipotesi

- $A \in \mathbb{R}^{n\times n}$ matrice di grandi dimensioni della forma

  $$A = \tilde{A} + A_1 S A_2^T,$$

  con $\tilde{A}$ matrice diagonale e $A_1, A_2 \in \mathbb{R}^{n\times k}$, $S \in \mathbb{R}^{k\times k}$ con $k \ll n$,

- $D \in \mathbb{R}^{m\times m}$ matrice di grandi dimensioni della forma

  $$D = \tilde{D} + D_1 P D_2^T,$$

  con $\tilde{D}$ matrice diagonale e $D_1, D_2 \in \mathbb{R}^{m\times j}$, $P \in \mathbb{R}^{j\times j}$ con $j \ll m$,

- $B \in \mathbb{R}^{n\times m}$ ammettente la seguente fattorizzazione di rango basso

  $$B := B_1 R B_2^T,$$

  dove $B_1 \in \mathbb{R}^{n\times l}$, $B_2 \in \mathbb{R}^{m\times l}$, $R \in \mathbb{R}^{l\times l}$ con $l \ll \max\{n, m\}$,

- $C \in \mathbb{R}^{m\times n}$ ammettente la seguente fattorizzazione di rango basso

  $$C := C_1 T C_2^T,$$

  dove $C_1 \in \mathbb{R}^{m\times h}$, $C_2 \in \mathbb{R}^{n\times h}$, $R \in \mathbb{R}^{h\times h}$ con $h \ll \max\{n, m\}$.

Riprendendo quanto già illustrato nel capitolo precendente, supporre che le matrici in gioco abbiano 'grandi dimensioni' equivale euristicamente a supporre che il calcolo del prodotto matrice-vettore o invertire una matrice di tali dimensioni richieda un tempo non ragionevole.

Come esempio della casisitica descritta dalla $\mathcal{NARE}$ (5.1), si può prendere in considerazione l'equazione che modellizza un problema di moto di particelle lungo un'asta omogenea illustrato nel paragrafo 1.2. Se si suppone, infatti, che il numero degli stati n sia grande, i coefficienti sono dati da

- $A := (I_n - \Phi_1\Phi_2^T)\Delta^+ = \Delta^+ - \Phi_1\left(\Phi_2^T\Delta^+\right)$,
- $D := (I_n - \Phi_1\Phi_2^T)\Delta^- = \Delta^- - \Phi_1\left(\Phi_2^T\Delta^-\right)$,
- $B := -\Gamma_1\Gamma_2^T\Delta^- = -\Gamma_1\left(\Gamma_2^T\Delta^-\right)$,
- $C := -\Gamma_1\Gamma_2^T\Delta^+ = -\Gamma_1\left(\Gamma_2^T\Delta^+\right)$,

con Φ_1, Φ_2 matrici di rango basso sparse, Γ_1, Γ_2 matrici di rango basso, Δ^+, Δ^- matrici diagonali.

Per quanto esposto nel capitolo 3, il metodo $\mathcal{CR}$ consiste nel generare la successione di polinomi matriciali quadratici

$$\mathcal{A}_k(X) := A_0^{(k)} + A_1^{(k)}X + A_2^{(k)}X^2,$$

i cui coefficienti sono definiti per ricorrenza dalle equazioni

$$\begin{aligned} A_0^{(k+1)} &= -A_0^{(k)}\left(A_1^{(k)}\right)^{-1}A_0^{(k)}, \\ A_1^{(k+1)} &= A_1^{(k)} - A_0^{(k)}\left(A_1^{(k)}\right)^{-1}A_2^{(k)} - A_2^{(k)}\left(A_1^{(k)}\right)^{-1}A_0^{(k)}, \\ A_2^{(k+1)} &= -A_2^{(k)}\left(A_1^{(k)}\right)^{-1}A_2^{(k)}. \end{aligned} \tag{5.2}$$

Se il metodo è applicato per risolvere una $\mathcal{NARE}$ tali matrici risultano avere la seguente struttura

$$A_0^{(k)} := \begin{pmatrix} R_1^{(k)} & 0 \\ R_2^{(k)} & 0 \end{pmatrix}, \quad A_1^{(k)} := \begin{pmatrix} -I_n & R_3^{(k)} \\ R_4^{(k)} & R_5^{(k)} \end{pmatrix}, \quad A_2^{(k)} := \begin{pmatrix} 0 & 0 \\ 0 & R_6^{(k)} \end{pmatrix}.$$

I valori dei termini iniziali delle successioni variano a seconda che si utilizzi una trasformazione affine $\mathcal{A}_\alpha$ o una trasformazione di Cayley $\mathcal{C}_\gamma$. Nel primo caso, si hanno le relazioni

$$\begin{aligned} R_1^{(0)} = R_1^{(\alpha)} &:= \alpha A - I_n, & R_2^{(0)} = R_2^{(\alpha)} &:= -\alpha C, \\ R_3^{(0)} = R_3^{(\alpha)} &:= -\alpha B, & R_4^{(0)} = R_4^{(\alpha)} &:= 0, \\ R_5^{(0)} = R_5^{(\alpha)} &:= -\alpha D - I_m, & R_6^{(0)} = R_6^{(\alpha)} &:= -I_m. \end{aligned} \tag{5.3}$$

Se si utilizza, invece, una trasformazione di Cayley $\mathcal{C}_\gamma$ di parametro $\gamma > 0$, le matrici sono inizializzate come segue

$$\begin{aligned} R_1^{(0)} = R_1^{(\gamma)} &:= I_n - 2\gamma W^{-1}, & R_2^{(0)} = R_2^{(\gamma)} &:= 2\gamma D_\gamma^{-1}CW^{-1}, \\ R_3^{(0)} = R_3^{(\gamma)} &:= -2\gamma A_\gamma^{-1}BV^{-1}, & R_4^{(0)} = R_4^{(\gamma)} &:= 0, \\ R_5^{(0)} = R_5^{(\gamma)} &:= I_m - 2\gamma V^{-1}, & R_6^{(0)} = R_6^{(\gamma)} &:= -I_m, \end{aligned} \tag{5.4}$$

dove

$$\begin{aligned} W &:= A_\gamma + BD_\gamma^{-1}C \\ V &:= -D_\gamma - CA_\gamma^{-1}B, \end{aligned}$$

con

$$A_\gamma := A + \gamma I_n \qquad \text{e} \qquad D_\gamma := D - \gamma I_m.$$

Nella sezione 3.2.3 si è poi mostrato che la successione di matrici $\{ X_k \}_{k\in\mathbb{N}}$ dove

$$X_k := -(R_5^{(0)} + R_4^{(k)}R_3^{(0)})^{-1}(R_2^{(0)} + R_4^{(k)}R_1^{(0)}) \tag{5.5}$$

è ben definita ed ha come limite la soluzione minimale non negativa $X_{\min}$ della $\mathcal{NARE}$ (5.1).

La seguente formulazione del metodo $\mathcal{CR}$, esposta nel dettaglio nel capitolo 3, rappresenta la versione standard dell'algoritmo. Tale versione, avendo carattere generali, non sfrutta le particolari proprietà di struttura del problema (5.1), e risulta nella pratica inapplicabile in quanto richederebbe $O(n^3)$ operazioni elementari per passo, costo non sostenibile per le ipotesi sulla dimensione dei coefficienti della $\mathcal{NARE}$. Nel presente capitolo, sulla scorta di quanto proposto dai matematici Chang-Yi Weng, Tiexiang Li, Eric King-wah Chu e Wen-Wei Lin, è stata proposta una nuova versione del metodo di riduzione ciclica, il **large-scale** $\mathcal{CR}$ ($\mathcal{CR}_{ls}$), che meglio si adatta alla risoluzione di $\mathcal{NARE}$ con le suddette caratteristiche strutturali e dimensionali. È doveroso sottolineare che per una corretta implementazione di tale veriante del metodo $\mathcal{CR}$ è fondamentale che i coefficienti dell'equazione di Riccati abbiano le proprietà di struttura richieste, in caso contrario, infatti, il metodo $\mathcal{CR}_{ls}$ risulterebbe ben più oneroso della sua versione standard.

Le modifiche apportate al metodo $\mathcal{CR}$ di seguito presentate si propongono sostanzialmente due obiettivi: il primo è quello di abbattere il costo computazionale per iterazione portandolo a circa $O(n)$ operazioni per passo, il secondo è quello di preservare, nel corso delle iterazioni, le proprietà strutturali delle matrici $A_i^{(k)}$ per $i = 0, 1, 2$, rendendo semplificata l'implementazione dell'algoritmo e la risoluzione del sistema (5.5). Le idee fondamentali che hanno portato alla formulazione del metodo $\mathcal{CR}_{ls}$ ricalcano quelli che sono i punti cardine delle modifiche apportate al metodo $\mathcal{SDA}_{ls}$. In fase di elaborazione dell'algoritmo, quindi, si sono seguite le seguenti linee guida

- utilizzare la formula di Sherman-Morrison-Woodbury per calcolare efficientemente l'inversa di una matrice della forma matrice diagonale o sparsa più matrice di rango basso,
- introdurre nuove successioni definite per ricorrenza per le $A_i^{(k)}$ con $i = 0, 1, 2$ in modo da preservare le proprietà di struttura presenti nei termini iniziali,
- utilizzare il procedimento di troncamento e compressione per controllare la crescita potenzialmente esponenziale delle correzioni di rango cercando di raggiungere un 'compromesso' tra migliore efficienza dell'algortirmo e perdita di accuratezza della soluzione,
- determinare condizioni d'arresto ed un controllo di convergenza che mantengano il costo computazionale dell'algoritmo entro la soglia delle $O(n)$ operazioni per iterazione.

Sia $\bar{n} := \max\{n, m\}$, e si osservi che alla luce delle ipotesi sulla $\mathcal{NARE}$ (5.1), per inizializzare l'algoritmo sono necessarie le seguenti operazioni:

- se si utilizza una trasformazione affine $\mathcal{A}_\alpha$, il calcolo dei termini iniziali $A_i^{(0)}$ per $i = 0, 1, 2$, richiede $O(\bar{n})$ operazioni per definire le matrici $R_1^{(\alpha)}$ e $R_5^{(\alpha)}$, ed un costo trascurabile per il calcolo delle matrici $R_2^{(\alpha)}$ e $R_3^{(\alpha)}$. Si ha, quindi, un costo complessivo di $O(\bar{n})$ operazioni elementari come voluto;
- se si utilizza, invece, una trasformazione di Cayley C_γ, il calcolo dei termini iniziali risulta più elaborato, ma con un costo analogo al caso precedentemente trattato. Adoperando opportunamente la $\mathcal{SMWF}$, è possibile calcolare efficientemente le inverse che compaiono nelle (5.4), si ha infatti

$$\begin{aligned}(A_\gamma)^{-1} &= (A+\gamma I_n)^{-1} = \left((\tilde{A}+\gamma I_n) + A_1 S A_2^T\right)^{-1} \\ &= \tilde{A}_\gamma^{-1} - \tilde{A}_\gamma^{-1} A_1 \left(I_k + S A_2^T \tilde{A}_\gamma^{-1} A_1\right)^{-1} S A_2^T \tilde{A}_\gamma^{-1},\end{aligned} \tag{5.6}$$

$$\begin{aligned}(D_\gamma)^{-1} &= (D-\gamma I_m)^{-1} = \left((\tilde{D}-\gamma I_n) + D_1 P D_2^T\right)^{-1} \\ &= \tilde{D}_\gamma^{-1} - \tilde{D}_\gamma^{-1} D_1 \left(I_j + P D_2^T \tilde{D}_\gamma^{-1} D_1\right)^{-1} P D_2^T \tilde{D}_\gamma^{-1},\end{aligned} \tag{5.7}$$

dunque tali inverse si calcolano in $O(\bar{n})$ operazioni elementari, allo stesso modo

$$\begin{aligned} W^{-1} &= (A_\gamma + BD_\gamma^{-1}C)^{-1} = (A_\gamma + B_1RB_2^TD_\gamma^{-1}C)^{-1} \\ &= A_\gamma^{-1} - A_\gamma^{-1}B_1\left(I_l + RB_2^TD_\gamma^{-1}CA_\gamma^{-1}B_1\right)^{-1}RB_2^TD_\gamma^{-1}CA_\gamma^{-1}, \\ &= (A_\gamma + BD_\gamma^{-1}C)^{-1} = (A_\gamma + BD_\gamma^{-1}C_1TC_2^T)^{-1} \\ &= A_\gamma^{-1} - A_\gamma^{-1}BD_\gamma^{-1}C_1T\left(I_h + C_2^TA_\gamma^{-1}BD_\gamma^{-1}C_1T\right)^{-1}C_2^TA_\gamma^{-1}, \end{aligned} \tag{5.8}$$

$$\begin{aligned} V^{-1} &= -(D_\gamma + CA_\gamma^{-1}B)^{-1} = -(D_\gamma + C_1TC_2^TA_\gamma^{-1}B)^{-1} \\ &= -D_\gamma^{-1} + D_\gamma^{-1}C_1\left(I_h + TC_2^TA_\gamma^{-1}BD_\gamma^{-1}C_1\right)^{-1}TC_2^TA_\gamma^{-1}BD_\gamma^{-1}, \\ &= -(D_\gamma + CA_\gamma^{-1}B)^{-1} = -(D_\gamma + CA_\gamma^{-1}B_1RB_2^T)^{-1} \\ &= -D_\gamma^{-1} + D_\gamma^{-1}CA_\gamma^{-1}B_1R\left(I_l + B_2^TD_\gamma^{-1}CA_\gamma^{-1}B_1R\right)^{-1}B_2^TD_\gamma^{-1}. \end{aligned} \tag{5.9}$$

Utilizzando le formule (5.6), è possibile, quindi, calcolare le precedenti inversioni matriciali con $O(\bar{n})$ operazioni elementari. Si osservi, dunque, che, almeno per quanto riguarda la definizione dei primi termini delle successioni $\{ A_i^{(k)} \}$, sia adoperando una trasformazione affine, che una trasformazione di Cayley, si è raggiunto l'obiettivo di limitare ad $O(n)$ operazioni il costo computazionale. Per una trattazione dettagliata di tale costo si rimanda alla sezione 5.1.5.

5.1.1 Descrizione dell'algoritmo

Come ampiamente illustrato, la $\mathcal{SMWF}$ risulta uno strumento utilissimo per calcolare in modo efficiente l'inversa di una matrice della forma matrice diagonale (o quanto meno sparsa) più una matrice di rango basso. Nell'elaborazione dell'algoritmo si è dunque cercato di presevare tale struttuara in modo da adoperare nel modo più appropriato la $\mathcal{SMWF}$. Alla luce delle suddette osservazioni, il metodo $\mathcal{CR}_{ls}$ introduce delle successioni definite per ricorrenza da affiancare alle (5.2), che realizzino l'obiettivo di preservare la particolare struttura di matrice diagonale più rango basso presente nei termini iniziali.

Si ponga dunque

$$\begin{aligned} A_0^{(k)} &:= \tilde{A_0}^{(k)} + U_0^{(k)}T_0^{(k)}V_0^{(k)^T}, \\ A_1^{(k)} &:= \tilde{A_1}^{(k)} + U_1^{(k)}T_1^{(k)}V_1^{(k)^T}, \\ A_2^{(k)} &:= \tilde{A_2}^{(k)} + U_2^{(k)}T_2^{(k)}V_2^{(k)^T}, \end{aligned} \tag{5.10}$$

con

- $U_i^{(k)}, V_i^{(k)} \in \mathbb{R}^{(m+n)\times l_i^k}$,
- $\tilde{A_i}^{(k)} \in \mathbb{R}^{(m+n)\times(m+n)}$ matrici diagonali,

per $i = 0, 1, 2$.

Si osservi ora cha la matrice $A_1^{(k)}$, posta nella forma indicata dalla (5.10), risulta facilmente invertibile adoperando la $\mathcal{SMWF}$, infatti, ponendo $\tilde{B}^{(k)} := \left(\tilde{A_1}^{(k)}\right)^{-1}$, si ha

$$\begin{aligned} \left(A_1^{(k)}\right)^{-1} &= \left(\tilde{A_1}^{(k)} + U_1^{(k)}T_1^{(k)}V_1^{(k)^T}\right)^{-1} \\ &= \tilde{B}^{(k)} - \tilde{B}^{(k)}U_1^{(k)}\left(I_{l_1^k} + T_1^{(k)}V_1^{(k)^T}\tilde{B}^{(k)}U_1^{(k)}\right)^{-1}T_1^{(k)}V_1^{(k)^T}\tilde{B}^{(k)}, \\ &=: \tilde{B}^{(k)} - Z^{(k)}Y^{(k)}W^{(k)^T}, \end{aligned} \tag{5.11}$$

con

$$Z^{(k)} := \tilde{B}^{(k)}U_1^{(k)}, \qquad W^{(k)} := \tilde{B}^{(k)}V_1^{(k)}T_1^{(k)^T}, \qquad Y^{(k)} := \left(I_{l_1^k} + W^{(k)^T}U_1^{(k)}\right)^{-1}.$$

Si conclude, quindi, che adoperando la $\mathcal{SMWF}$ e la particolare espressione della matrice $A_1^{(k)}$, è possibile calcolare l'inversa (5.11) con sole $O(\bar{n})$ operazioni elementari, a patto che le dimensioni delle matrici $U_1^{(k)}$, $V_1^{(k)}$, $T_1^{(k)}$ siano trascurabili.

Utilizzando la (5.11), è possibile esplicitare una formulazione ricorsiva delle matrici diagonali $\tilde{A_i}^{(k)}$ e delle correzioni di rango $U_i^{(k)}$ e $V_i^{(k)}$ per $i = 0, 1, 2$. Si ha, infatti,

$$\begin{aligned}
A_0^{(k+1)} =& - A_0^{(k)} \left(A_1^{(k)}\right)^{-1} A_0^{(k)} \\
=& - \left(\tilde{A_0}^{(k)} + U_0^{(k)} T_0^{(k)} {V_0^{(k)}}^T\right) \left(\tilde{B}^{(k)} - Z^{(k)} Y^{(k)} {W^{(k)}}^T\right) \left(\tilde{A_0}^{(k)} + U_0^{(k)} T_0^{(k)} {V_0^{(k)}}^T\right) \\
=& - \tilde{A_0}^{(k)} \tilde{B}^{(k)} \tilde{A_0}^{(k)} + \tilde{A_0}^{(k)} Z^{(k)} Y^{(k)} {W^{(k)}}^T \tilde{A_0}^{(k)} - \tilde{A_0}^{(k)} \left(A_1^{(k)}\right)^{-1} U_0^{(k)} T_0^{(k)} {V_0^{(k)}}^T \\
& - U_0^{(k)} T_0^{(k)} {V_0^{(k)}}^T \left(A_1^{(k)}\right)^{-1} \tilde{A_0}^{(k)} - U_0^{(k)} T_0^{(k)} {V_0^{(k)}}^T \left(A_1^{(k)}\right)^{-1} U_0^{(k)} T_0^{(k)} {V_0^{(k)}}^T,
\end{aligned}$$

da cui si ottiene

$$\tilde{A_0}^{(k+1)} := -\tilde{A_0}^{(k)} \tilde{B}^{(k)} \tilde{A_0}^{(k)}, \tag{5.12}$$

$$\begin{aligned}
U_0^{(k+1)} &:= \left(\tilde{A_0}^{(k)} Z^{(k)} \quad U_{00}^{(k)} \quad U_0^{(k)}\right), \\
V_0^{(k+1)} &:= \left(\tilde{A_0}^{(k)} W^{(k)} \quad V_{00}^{(k)} \quad V_0^{(k)}\right),
\end{aligned} \tag{5.13}$$

e

$$T_0^{(k+1)} := \begin{pmatrix} Y^{(k)} & 0 & 0 \\ 0 & 0 & -T_0^{(k)} \\ 0 & -T_0^{(k)} & T_{00}^{(k)} \end{pmatrix}, \tag{5.14}$$

con

$$U_{00}^{(k)} := \tilde{A_0}^{(k)} \left(A_1^{(k)}\right)^{-1} U_0^{(k)}, \qquad V_{00}^{(k)} := \tilde{A_0}^{(k)} \left(A_1^{(k)}\right)^{-T} V_0^{(k)}$$
$$T_{00}^{(k)} := -T_0^{(k)} {V_0^{(k)}}^T \left(A_1^{(k)}\right)^{-1} U_0^{(k)} T_0^{(k)}.$$

Per la matrice $A_2^{(k)}$, si svolgono calcoli analoghi

$$\begin{aligned}
A_2^{(k+1)} =& - A_2^{(k)} \left(A_1^{(k)}\right)^{-1} A_2^{(k)} \\
=& - \left(\tilde{A_2}^{(k)} + U_2^{(k)} T_2^{(k)} {V_2^{(k)}}^T\right) \left(\tilde{B}^{(k)} - Z^{(k)} Y^{(k)} {W^{(k)}}^T\right) \left(\tilde{A_2}^{(k)} + U_2^{(k)} T_2^{(k)} {V_2^{(k)}}^T\right) \\
=& - \tilde{A_2}^{(k)} \tilde{B}^{(k)} \tilde{A_2}^{(k)} + \tilde{A_2}^{(k)} Z^{(k)} Y^{(k)} {W^{(k)}}^T \tilde{A_2}^{(k)} - \tilde{A_2}^{(k)} \left(A_1^{(k)}\right)^{-1} U_2^{(k)} T_2^{(k)} {V_2^{(k)}}^T \\
& - U_2^{(k)} T_2^{(k)} {V_2^{(k)}}^T \left(A_1^{(k)}\right)^{-1} \tilde{A_2}^{(k)} - U_2^{(k)} T_2^{(k)} {V_2^{(k)}}^T \left(A_1^{(k)}\right)^{-1} U_2^{(k)} T_2^{(k)} {V_2^{(k)}}^T,
\end{aligned}$$

da cui si ottiene

$$\tilde{A_2}^{(k+1)} := -\tilde{A_2}^{(k)} \tilde{B}^{(k)} \tilde{A_2}^{(k)}, \tag{5.15}$$

$$\begin{aligned}
U_2^{(k+1)} &:= \left(\tilde{A_2}^{(k)} Z^{(k)} \quad U_{22}^{(k)} \quad U_2^{(k)}\right), \\
V_2^{(k+1)} &:= \left(\tilde{A_2}^{(k)} W^{(k)} \quad V_{22}^{(k)} \quad V_2^{(k)}\right),
\end{aligned} \tag{5.16}$$

e

$$T_2^{(k+1)} := \begin{pmatrix} Y^{(k)} & 0 & 0 \\ 0 & 0 & -T_2^{(k)} \\ 0 & -T_2^{(k)} & T_{22}^{(k)} \end{pmatrix} \tag{5.17}$$

con

$$U_{22}^{(k)} := \tilde{A_2}^{(k)} \left(A_1^{(k)}\right)^{-1} U_2^{(k)}, \qquad V_{22}^{(k)} := \tilde{A_2}^{(k)} \left(A_1^{(k)}\right)^{-T} V_2^{(k)},$$
$$T_{22}^{(k)} := -T_2^{(k)} {V_2^{(k)}}^T \left(A_1^{(k)}\right)^{-1} U_2^{(k)} T_2^{(k)}.$$

Allo stesso modo per la matrice $A_1^{(k)}$ si ha

$$\begin{aligned}
A_1^{(k+1)} =& A_1^{(k)} - A_0^{(k)} \left(A_1^{(k)}\right)^{-1} A_2^{(k)} - A_2^{(k)} \left(A_1^{(k)}\right)^{-1} A_0^{(k)} \\
=& \tilde{A_1}^{(k)} + U_1^{(k)} T_1^{(k)} {V_1^{(k)}}^T + \\
& - \left(\tilde{A_0}^{(k)} + U_0^{(k)} T_0^{(k)} {V_0^{(k)}}^T\right) \left(\tilde{B}^{(k)} - Z^{(k)} Y^{(k)} {W^{(k)}}^T\right) \left(\tilde{A_2}^{(k)} + U_2^{(k)} T_2^{(k)} {V_2^{(k)}}^T\right) + \\
& - \left(\tilde{A_2}^{(k)} + U_2^{(k)} T_2^{(k)} {V_2^{(k)}}^T\right) \left(\tilde{B}^{(k)} - Z^{(k)} Y^{(k)} {W^{(k)}}^T\right) \left(\tilde{A_0}^{(k)} + U_0^{(k)} T_0^{(k)} {V_0^{(k)}}^T\right) \\
=& \tilde{A_1}^{(k)} - \tilde{A_0}^{(k)} \tilde{B}^{(k)} \tilde{A_2}^{(k)} - \tilde{A_2}^{(k)} \tilde{B}^{(k)} \tilde{A_0}^{(k)} + U_1^{(k)} T_1^{(k)} {V_1^{(k)}}^T \\
& + \tilde{A_0}^{(k)} Z^{(k)} Y^{(k)} {W^{(k)}}^T \tilde{A_2}^{(k)} - \tilde{A_0}^{(k)} \left(A_1^{(k)}\right)^{-1} U_2^{(k)} T_2^{(k)} {V_2^{(k)}}^T + \\
& - U_0^{(k)} T_0^{(k)} {V_0^{(k)}}^T \left(A_1^{(k)}\right)^{-1} \tilde{A_2}^{(k)} - U_0^{(k)} T_0^{(k)} {V_0^{(k)}}^T \left(A_1^{(k)}\right)^{-1} U_2^{(k)} T_2^{(k)} {V_2^{(k)}}^T + \\
& + \tilde{A_2}^{(k)} Z^{(k)} Y^{(k)} {W^{(k)}}^T \tilde{A_0}^{(k)} - \tilde{A_2}^{(k)} \left(A_1^{(k)}\right)^{-1} U_0^{(k)} T_0^{(k)} {V_0^{(k)}}^T + \\
& - U_2^{(k)} T_2^{(k)} {V_2^{(k)}}^T \left(A_1^{(k)}\right)^{-1} \tilde{A_0}^{(k)} - U_2^{(k)} T_2^{(k)} {V_2^{(k)}}^T \left(A_1^{(k)}\right)^{-1} U_0^{(k)} T_0^{(k)} {V_0^{(k)}}^T,
\end{aligned}$$

si pone quindi

$$\tilde{A_1}^{(k+1)} := \tilde{A_1}^{(k)} - \tilde{A_0}^{(k)} \tilde{B}^{(k)} \tilde{A_2}^{(k)} - \tilde{A_2}^{(k)} \tilde{B}^{(k)} \tilde{A_0}^{(k)}, \tag{5.18}$$

$$\begin{aligned}
U_1^{(k+1)} &:= \begin{pmatrix} U_1^{(k)} & \tilde{A_0}^{(k)} Z^{(k)} & U_{02}^{(k)} & U_0^{(k)} & \tilde{A_2}^{(k)} Z^{(k)} & U_{20}^{(k)} & U_2^{(k)} \end{pmatrix}, \\
V_1^{(k+1)} &:= \begin{pmatrix} V_1^{(k)} & \tilde{A_2}^{(k)} W^{(k)} & V_{20}^{(k)} & V_2^{(k)} & \tilde{A_0}^{(k)} W^{(k)} & V_{02}^{(k)} & V_0^{(k)} \end{pmatrix},
\end{aligned} \tag{5.19}$$

e

$$T_1^{(k+1)} := T_1^{(k)} \oplus \begin{pmatrix} Y^{(k)} & 0 & 0 \\ 0 & 0 & -T_2^{(k)} \\ 0 & -T_0^{(k)} & T_{02}^{(k)} \end{pmatrix} \oplus \begin{pmatrix} Y^{(k)} & 0 & 0 \\ 0 & 0 & -T_0^{(k)} \\ 0 & -T_2^{(k)} & T_{20}^{(k)} \end{pmatrix} \tag{5.20}$$

con

$$\begin{aligned}
U_{02}^{(k)} &:= \tilde{A_0}^{(k)} \left(A_1^{(k)}\right)^{-1} U_2^{(k)}, \qquad U_{20} := \tilde{A_2}^{(k)} \left(A_1^{(k)}\right)^{-1} U_0^{(k)}, \\
V_{02}^{(k)} &:= \tilde{A_0}^{(k)} \left(A_1^{(k)}\right)^{-T} V_2^{(k)}, \qquad V_{20}^{(k)} := \tilde{A_2}^{(k)} \left(A_1^{(k)}\right)^{-T} V_0^{(k)}, \\
T_{02}^{(k)} &:= -T_0^{(k)} {V_0^{(k)}}^T \left(A_1^{(k)}\right)^{-1} U_2^{(k)} T_2^{(k)}, \\
T_{20}^{(k)} &:= -T_2^{(k)} {V_2^{(k)}}^T \left(A_1^{(k)}\right)^{-1} U_0^{(k)} T_0^{(k)}
\end{aligned}$$

Per completare la formulazione dell'algoritmo, occorre dare un'espressione coerente con le (5.10) dei termini iniziali $A_i^{(0)}$ per $i = 0, 1, 2$. Al solito, si hanno due differenti inizializzazioni a seconda che si usi una trasformazione affine $\mathcal{A}_\alpha$ o una trasformazione di Cayley $\mathcal{C}_\gamma$. Nel primo caso, per la (5.3), si ha

$$\begin{aligned}
A_0^{(0)} &= \begin{pmatrix} R_1^{(\alpha)} & 0 \\ R_2^{(\alpha)} & 0 \end{pmatrix} = \begin{pmatrix} \alpha A - I_n & 0 \\ -\alpha C & 0 \end{pmatrix} = \begin{pmatrix} \alpha \tilde{A} - I_n & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} \alpha A_1 S A_2^T & 0 \\ -\alpha C_1 T C_2^T & 0 \end{pmatrix} \\
&= \begin{pmatrix} \alpha \tilde{A} - I_n & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} A_1 & 0 \\ 0 & C_1 \end{pmatrix} \begin{pmatrix} \alpha S & 0 \\ 0 & -\alpha T \end{pmatrix} \begin{pmatrix} A_2^T & 0 \\ C_2^T & 0 \end{pmatrix} =: \tilde{A_0}^{(0)} + U_0^{(0)} T_0^{(0)} {V_0^{(0)}}^T,
\end{aligned}$$

$$\begin{aligned}
A_1^{(0)} &= \begin{pmatrix} -I_n & R_3^{(\alpha)} \\ R_4^{(\alpha)} & R_5^{(\alpha)} \end{pmatrix} = \begin{pmatrix} -I_n & -\alpha B \\ 0 & -\alpha D - I_m \end{pmatrix} = \begin{pmatrix} -I_n & 0 \\ 0 & -\alpha \tilde{D} - I_m \end{pmatrix} + \begin{pmatrix} 0 & -\alpha B_1 R B_2^T \\ 0 & -\alpha D_1 P D_2^T \end{pmatrix} \\
&= \begin{pmatrix} -I_n & 0 \\ 0 & -\alpha \tilde{D} - I_m \end{pmatrix} + \begin{pmatrix} B_1 & 0 \\ 0 & D_1 \end{pmatrix} \begin{pmatrix} -\alpha R & 0 \\ 0 & -\alpha P \end{pmatrix} \begin{pmatrix} 0 & B_2^T \\ 0 & D_2^T \end{pmatrix} =: \tilde{A_1}^{(0)} + U_1^{(0)} T_1^{(0)} {V_1^{(0)}}^T,
\end{aligned}$$

$$A_2^{(0)} = \begin{pmatrix} 0 & 0 \\ 0 & R_6^{(\alpha)} \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & -I_m \end{pmatrix} =: \tilde{A}_2{}^{(0)} + U_2^{(0)} T_2^{(0)} V_2^{(0)^T}.$$

Adoperando una trasformazione di Cayley, per la (5.6) e la (5.8), i primi termini della successione sono inizializzati come segue

$$\begin{aligned}
A_0^{(0)} &= \begin{pmatrix} R_1^{(\gamma)} & 0 \\ R_2^{(\gamma)} & 0 \end{pmatrix} = \begin{pmatrix} I_n - 2\gamma W^{-1} & 0 \\ 2\gamma D_\gamma^{-1} C W^{-1} & 0 \end{pmatrix} \\
&= \begin{pmatrix} I_n - 2\gamma A_\gamma^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 2\gamma A_\gamma^{-1} B_1 \Lambda B_2^T D_\gamma^{-1} C A_\gamma^{-1} & 0 \\ 2\gamma D_\gamma^{-1} C_1 T C_2 W^{-1} & 0 \end{pmatrix} \\
&= \begin{pmatrix} I_n - 2\gamma A_\gamma^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} A_\gamma^{-1} B_1 & 0 \\ 0 & D_\gamma^{-1} C_1 \end{pmatrix} \begin{pmatrix} 2\gamma\Lambda & 0 \\ 0 & 2\gamma T \end{pmatrix} \begin{pmatrix} B_2^T D_\gamma^{-1} C A_\gamma^{-1} & 0 \\ C_2^T W^{-1} & 0 \end{pmatrix} \\
&=: \tilde{A}_0{}^{(0)} + U_0^{(0)} T_0^{(0)} V_0^{(0)^T}, \\
A_1^{(0)} &= \begin{pmatrix} -I_n & R_3^{(\gamma)} \\ R_4^{(\gamma)} & R_5^{(\gamma)} \end{pmatrix} = \begin{pmatrix} -I_n & -2\gamma A_\gamma^{-1} B V^{-1} \\ 0 & I_m - 2\gamma V^{-1} \end{pmatrix} \\
&= \begin{pmatrix} I_n & 0 \\ 0 & I_m + 2\gamma D_\gamma^{-1} \end{pmatrix} + \begin{pmatrix} 0 & -2\gamma A_\gamma^{-1} B_1 R B_2^T V^{-1} \\ 0 & -2\gamma D_\gamma^{-1} C A_\gamma^{-1} B_1 \Gamma B_2^T D_\gamma^{-1} \end{pmatrix} \\
&= \begin{pmatrix} I_n & 0 \\ 0 & I_m + 2\gamma D_\gamma^{-1} \end{pmatrix} + \begin{pmatrix} A_\gamma^{-1} B_1 & 0 \\ 0 & D_\gamma^{-1} C A_\gamma^{-1} B_1 \end{pmatrix} \begin{pmatrix} -2\gamma R & 0 \\ 0 & -2\gamma\Gamma \end{pmatrix} \begin{pmatrix} 0 & B_2^T V^{-1} \\ 0 & B_2^T D_\gamma^{-1} \end{pmatrix} \\
&=: \tilde{A}_1{}^{(0)} + U_1^{(0)} T_1^{(0)} V_1^{(0)^T}, \\
A_2^{(0)} &= \begin{pmatrix} 0 & 0 \\ 0 & R_6^{(\gamma)} \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & -I_m \end{pmatrix} =: \tilde{A}_2{}^{(0)} + U_2^{(0)} T_2^{(0)} V_2^{(0)^T}.
\end{aligned}$$

Si osservi che le espressioni precedenti, sebbene tutt'altro che attraenti, inizializzano il metodo $\mathcal{CR}_{ls}$ con un costo computazionale di $O(\bar{n})$ operazioni aritmetiche. Per uno studio più puntuale del costo computazionale per implementare un'iterazione dell'algoritmo e per inizializzare i termini delle successioni si rimanda alla sezione 5.1.5

5.1.2 Troncamento e compressione

L'algoritmo illustrato nella sezione precedente si propone di conservare nelle matrici $A_i^{(k)}$ la particolare struttura 'diagonale più rango basso' per utilizzare al meglio la $\mathcal{SMWF}$. È evidente, però, dalle formule (5.13), (5.14), (5.19), (5.20), (5.16), (5.17), che le correzioni di rango si compongono di 3 o 7 blocchi matriciali. Tali relazioni, quindi, scoraggerebbero l'utilizzo dell'algoritmo in quanto le dimensioni delle correzioni di rango avrebbero crescita potenzialmente esponenziale, vanificando in poche iterazioni qualsiasi vantaggio derivante dalle proprietà di struttura iniziali. Come per l'algoritmo $\mathcal{SDA}_{ls}$, però, è possibile utilizzare le proprietà asintotiche delle matrici 'in gioco' per definire un meccanismo che limiti la crescita esponenziale delle dimensioni. A tal proposito si ricordi che, per quanto esposto nel teorema 3.2.11 valgono le relazioni

$$\lim_{k\to\infty} A_0^{(k)} = 0 \qquad \lim_{k\to\infty} A_2^{(k)} = 0,$$

e sono verificate le stime

$$\begin{aligned}
\limsup_{k\to\infty} \sqrt[2^k]{\|A_0^{(k)}\|} \leq \sigma, \\
\limsup_{k\to\infty} \sqrt[2^k]{\|A_2^{(k)}\|} \leq \sigma,
\end{aligned} \tag{5.21}$$

con $\sigma < 1$.

Alla luce delle stime precedenti, è sperabile che, applicando un procedimento di **troncamento e compressione** analogo a quanto ideato da Chang-Yi Weng, Tiexiang Li, Eric King-wah Chu e Wen-Wei Lin, si riesca a limitare la crescita esponenziale delle correzioni di rango. Siano, quindi,

- τ_0, τ_1, τ_2 numeri reali 'abbastanza piccoli' che definiscono la soglia di tolleranza del troncamento,
- $l_{max} \ll n, m$ il quale esprime una maggiorazione per la dimensione delle correzioni di rango e definisce il parametro del processo di compressione.

In analogia con quanto fatto per il metodo $\mathcal{SDA}_{ls}$, adoperando la fattorizzazione QR con pivoting, è possibile ottenere la seguenti decomposizioni

$$U_i^{(k)} = Q_{U_i}^{(k)} M_{U_i}^{(k)} + W_{U_i}^{(k)}, \qquad V_i^{(k)} = Q_{V_i}^{(k)} M_{V_i}^{(k)} + W_{V_i}^{(k)}, \tag{5.22}$$

$$\tilde{A_0}^{(k)} = \tilde{B_0}^{(k)} + Z_0^{(k)}, \qquad \tilde{A_2}^{(k)} = \tilde{B_2}^{(k)} + Z_2^{(k)}, \tag{5.23}$$

con

- $Q_{U_i}^{(k)}, Q_{V_i}^{(k)} \in \mathbb{R}^{(m+n)\times r_i^k}$ matrici unitarie,
- $M_{U_i}^{(k)}, M_{V_i}^{(k)} \in \mathbb{R}^{r_i^k \times l_i^k}$ matrici triangolari superiori,
- $\tilde{B_0}^{(k)}$, $\tilde{B_2}^{(k)}$ matrici diagonali,

tali che

$$\|W_{U_i}^{(k)}\| \leq \tau_i, \qquad \|W_{V_i}^{(k)}\| \leq \tau_i, \qquad \|Z_0^{(k)}\| \leq \tau_0, \qquad \|Z_2^{(k)}\| \leq \tau_2,$$

e

$$r_i^k = \texttt{rank} U_i^{(k)} \leq l_i^k \leq l_{max} \ll n, m$$

per $i = 0, 1, 2$.

Sull'esistenza della decomposizione (5.22) valgono le medesime osservazioni fatte per l'analoga decomposizione illustrata nella sezione 4.1.2: stanti le stime (5.21), le matrici $A_0^{(k)}$ e $A_2^{(k)}$ hanno norma che decresce molto rapidamente, dunque, scegliendo opportunamente i parametri τ_i e l_{max}, è evidente che tali scomposizioni siano corrette. Tuttavia non è possibile dimostrare formalmente l'esistenza delle suddette decomposizioni o individuare delle restrizioni sui parametri che definiscono il processo di troncamento e compressione.

Utilizzando le decomposizioni (5.22) si hanno allora le relazioni

$$\begin{aligned}
A_0^{(k)} &= \tilde{A_0}^{(k)} + U_0^{(k)} T_0^{(k)} {V_0^{(k)}}^T = \tilde{B_0}^{(k)} + Q_{U_0}^{(k)} M_{U_0}^{(k)} T_0^{(k)} (Q_{V_0}^{(k)} M_{V_0}^{(k)})^T + O(\tau_0), \\
A_1^{(k)} &= \tilde{A_1}^{(k)} + U_1^{(k)} T_1^{(k)} {V_1^{(k)}}^T = \tilde{A_1}^{(k)} + Q_{U_1}^{(k)} M_{U_1}^{(k)} T_1^{(k)} (Q_{V_1}^{(k)} M_{V_1}^{(k)})^T + O(\tau_1), \\
A_2^{(k)} &= \tilde{A_2}^{(k)} + U_2^{(k)} T_2^{(k)} {V_2^{(k)}}^T = \tilde{B_2}^{(k)} + Q_{U_2}^{(k)} M_{U_2}^{(k)} T_2^{(k)} (Q_{V_2}^{(k)} M_{V_2}^{(k)})^T + O(\tau_2),
\end{aligned}$$

pertanto, trascurando i termini di norma minore delle soglie di troncamento τ_i, si può porre

$$\begin{aligned}
U_i^{(k)} &:= Q_{U_i}^{(k)}, \\
V_i^{(k)} &:= Q_{V_i}^{(k)}, \\
T_i^{(k)} &:= M_{U_i}^{(k)} T_i^{(k)} (M_{V_i}^{(k)})^T, \\
\tilde{A_0}^{(k)} &:= \tilde{B_0}^{(k)}, \\
\tilde{A_2}^{(k)} &:= \tilde{B_2}^{(k)}.
\end{aligned} \tag{5.24}$$

Il procedimento di compressione e troncamento sopra illustrato è indubbiamente uno strumento utilissimo per evitare la crescita fuori controllo delle dimensioni delle correzioni di rango, comporta tuttavia una perdita di accuratezza delle soluzioni. Tale perdita dipende ovviamente dalla scelta dei parametri τ_i e l_{max}. La buona riuscita del metodo $\mathcal{CR}_{ls}$ dipende sostanzialmente, quindi, dalla scelta dei suddetti parametri: scegliendo

soglie di troncamento τ_i molto piccole o margini dimensionali l_{max} molto larghi si ottiene indubbiamente una notevole precisione nella soluzione a scapito però delle prestazioni computazionali, di contro, scegliendo meno rigidamente i suddetti parametri, si velocizza l'algoritmo senza perdere necessariamente in accuratezza. Occorre, pertanto, cercare un compromesso tra efficienza e precisione, compromesso espresso della scelta dei parametri τ_i e l_{max}.

5.1.3 Condizione d'arresto e controllo di convergenza

Le stime (5.21), come osservato nel capitolo 3, suggeriscono una valida condizione per arrestare le iterazione del metodo $\mathcal{CR}$ (si confronti, a tal proposito, il codice 3.4). Poiché le stime (5.21) rimangono valide anche per il metodo $\mathcal{CR}_{ls}$, è possibile adottare tale criterio d'arresto anche per la versione large-scale del metodo $\mathcal{CR}$. Sia, dunque, ε la soglia di tolleranza scelta per l'implementazione, allora le condizioni richieste per terminare l'algoritmo sono

$$\|A_0^{(k)}\| \leq \|\tilde{A_0}^{(k)}\| + \|U_0^{(k)} T_0^{(k)} {V_0^{(k)}}^T\| \leq \varepsilon,$$
$$\|A_2^{(k)}\| \leq \|\tilde{A_2}^{(k)}\| + \|U_2^{(k)} T_2^{(k)} {V_2^{(k)}}^T\| \leq \varepsilon.$$

Si applichi ora il procedimento di compressione (senza troncamento) alle matrici $U_0^{(k)}$, $V_0^{(k)}$, $U_2^{(k)}$, $V_2^{(k)}$, allora, modificando opportunamente i nuclei $T_0^{(k)}$ e $T_2^{(k)}$, si possono riscrivere le relazioni precedenti nella nuova formulazione

$$\|A_0^{(k)}\| \leq \|\tilde{A_0}^{(k)}\| + \|T_0^{(k)}\| \leq \varepsilon,$$
$$\|A_2^{(k)}\| \leq \|\tilde{A_2}^{(k)}\| + \|T_2^{(k)}\| \leq \varepsilon.$$

Si osservi ora che, essendo le matrici $\tilde{A_0}^{(k)}$ e $\tilde{A_2}^{(k)}$ diagonali e poiché le matrici $T_0^{(k)}$ e $T_2^{(k)}$ hanno dimensione limitata dal parametro l_{max}, le condizioni d'arresto precedenti hanno un costo computazionale di $O(\bar{n})$ operazioni elementari per passo, sono pertanto conformi al target dell'algoritmo di non superare tale soglia del numero di operazioni.

Per controllare l'andamento della convergenza durante l'algoritmo, occorre, invece, calcolare le norme

$$\delta_i^k := \|A_i^k - A_i^{(k-1)}\|,$$

per $i = 0, 1, 2$. Utilizzando le (5.10) si ha

$$\begin{aligned} A_i^{(k)} - A_i^{(k-1)} &= \tilde{A_i}^{(k)} - \tilde{A_i}^{(k-1)} + U_i^{(k)} T_i^{(k)} {V_i^{(k)}}^T - U_i^{(k-1)} T_i^{(k-1)} {V_i^{(k-1)}}^T \\ &= \tilde{A_i}^{(k)} - \tilde{A_i}^{(k-1)} + \bar{U_i}^{(k)} \bar{T_i}^{(k)} {\bar{V_i}^{(k)}}^T \end{aligned}$$

con

$$\bar{U_i}^{(k)} := \begin{pmatrix} U_i^{(k)} & U_i^{(k-1)} \end{pmatrix} \qquad \bar{T_i}^{(k)} := \begin{pmatrix} T_i^{(k)} & 0 \\ 0 & -T_i^{(k-1)} \end{pmatrix} \qquad \bar{V_i}^{(k)} := \begin{pmatrix} V_i^{(k)} & V_i^{(k-1)} \end{pmatrix}.$$

Se si applica alle matrici $\bar{U_i}^{(k)}$ e $\bar{V_i}^{(k)}$ il solito procedimento di compressione, ricomponendo i nuclei $\bar{T_i}^{(k)}$, si ottengono le stime

$$\delta_i^k := \|A_i^{(k)} - A_i^{(k-1)}\| \leq \|\tilde{A_i}^{(k)} - \tilde{A_i}^{(k-1)}\| + \|\bar{T_i}^{(k)}\|.$$

Mediante tale procedimento si ottiene una maggiorazione dei parametri δ_i^k semplicemente calcolando la norma di una matrice diagonale e la norma di una matrice di rango basso, raggiungendo l'obiettivo di ottenere un valido controllo di convergenza entro le $O(\bar{n})$ operazioni elementari per passo prefissate.

Si osservi ora che, in analogia con quanto accade per il metodo $\mathcal{SDA}_{ls}$, il metodo $\mathcal{CR}_{ls}$ genera approssimazioni della soluzione numericamente di rango basso. Tale proprietà può essere immediatamente verificata applicando alla (5.5) la $\mathcal{SMWF}$. È, dunque possibile, rappresentando l'approssimazione X_k nella sua decomposizione di rango basso, individuare una formulazione del residuale relativo analoga a quella calcolata nel paragrafo 4.1.3. Tale formulazione, è però eccessivamente pesante e ricca di termini, pertanto si omette in tale contesto una trattazione dettgliata in quanto risulterebbe noiosa e si ridurrebbe ad una lunga serie di conti. È però opportuno sottolineare che anche per il metodo $\mathcal{CR}_{ls}$ è possibile calcolare il residuo relativo con un costo computazionale di $O(\bar{n})$ operazioni elementari per iterazione.

5.1.4 Propagazione dell'errore

Il procedimento troncamento e compressione è uno strumento fondamentale per l'algoritmo: lasciare incontrollata la crescita delle correzioni di rango 'brucerebbe' in pochissime iterazioni la struttura di rango basso. Di contro, però, tale meccanismo comporta una perdita nella precisione del metodo, perdita che andrebbe quantificata e se, possibile, mantenuta sotto controllo per evitare che la soluzione calcolata sia del tutto inattendibile. In tale sezione si studia l'andamento dell'errore generato dal procedimento di troncamento e compressione dando delle stime per il comportamento asintotico dell'errore stesso. Il risultato ottenuto è abbastanza confortante: gli errori, infatti, rimangono ad ogni iterazione dello stesso ordine del passo precedente, e sfruttando le relazioni (5.21), è possibile addirittura concludere che, anche al tendere all'infinito del numero delle iterazioni, non si verificano fenomeni di 'esplosione' dell'errore.

Siano $B_i^{(k)}$ i valori effettivamente calcolati dal metodo $\mathcal{CR}_{ls}$ a seguito del procedimento di troncamento e compressioni in luogo delle matrici 'esatte' $A_i^{(k)}$ per $i = 0, 1, 2$ e siano

$$\Sigma_i^k := B_i^{(k)} - A_i^{(k)} \qquad \text{per } i = 0, 1, 2,$$

gli errori generati al passo k-esimo da tale procedimento. Si ponga inoltre

$$\varepsilon_k := \max_{i=0,1,2} \|\Sigma_i^k\|, \qquad \tau := \max_{i=0,1,2} \tau_i.$$

Si osservi allora che $\varepsilon_0 = O(\tau)$. L'obiettivo ora è quello valutare l'evoluzione del parametro ε_k al crescere dell'iterazione k. A tal proposito si hanno le seguenti relazioni

$$\begin{aligned}
\Sigma_0^{k+1} &= B_0^{(k+1)} - A_0^{(k+1)} = -B_0^{(k)} \left(B_1^{(k)}\right)^{-1} B_0^{(k)} + A_0^{(k)} \left(A_1^{(k)}\right)^{-1} A_0^{(k)} \\
&= -(A_0^{(k)} + \Sigma_0^k) \left(A_1^{(k)} + \Sigma_1^k\right)^{-1} (A_0^{(k)} + \Sigma_0^k) + A_0^{(k)} \left(A_1^{(k)}\right)^{-1} A_0^{(k)} \\
&= -(A_0^{(k)} + \Sigma_0^k) \left(A_1^{(k)}\right)^{-1} \left(I + \Sigma_1^k \left(A_1^{(k)}\right)^{-1}\right)^{-1} (A_0^{(k)} + \Sigma_0^k) + A_0^{(k)} \left(A_1^{(k)}\right)^{-1} A_0^{(k)} \\
&= -(A_0^{(k)} + \Sigma_0^k) \left(A_1^{(k)}\right)^{-1} \left(I - \Sigma_1^k \left(A_1^{(k)}\right)^{-1}\right) (A_0^{(k)} + \Sigma_0^k) + A_0^{(k)} \left(A_1^{(k)}\right)^{-1} A_0^{(k)} + o(\varepsilon_k) \\
&= -A_0^{(k)} \left(A_1^{(k)}\right)^{-1} \Sigma_0^k - \Sigma_0^k \left(A_1^{(k)}\right)^{-1} A_0^{(k)} + A_0^{(k)} \left(A_1^{(k)}\right)^{-1} \Sigma_1^k \left(A_1^{(k)}\right)^{-1} A_0^{(k)} + o(\varepsilon_k);
\end{aligned}$$

$$\begin{aligned}
\Sigma_2^{k+1} &= B_2^{(k+1)} - A_2^{(k+1)} = -B_2^{(k)} \left(B_1^{(k)}\right)^{-1} B_2^{(k)} + A_2^{(k)} \left(A_1^{(k)}\right)^{-1} A_2^{(k)} \\
&= -(A_2^{(k)} + \Sigma_2^k) \left(A_1^{(k)} + \Sigma_1^k\right)^{-1} (A_2^{(k)} + \Sigma_2^k) + A_2^{(k)} \left(A_1^{(k)}\right)^{-1} A_2^{(k)} \\
&= -(A_2^{(k)} + \Sigma_2^k) \left(A_1^{(k)}\right)^{-1} \left(I + \Sigma_1^k \left(A_1^{(k)}\right)^{-1}\right)^{-1} (A_2^{(k)} + \Sigma_2^k) + A_2^{(k)} \left(A_1^{(k)}\right)^{-1} A_2^{(k)} \\
&= -(A_2^{(k)} + \Sigma_2^k) \left(A_1^{(k)}\right)^{-1} \left(I - \Sigma_1^k \left(A_1^{(k)}\right)^{-1}\right) (A_2^{(k)} + \Sigma_2^k) + A_2^{(k)} \left(A_1^{(k)}\right)^{-1} A_2^{(k)} + o(\varepsilon_k) \\
&= -A_2^{(k)} \left(A_1^{(k)}\right)^{-1} \Sigma_2^k - \Sigma_2^k \left(A_1^{(k)}\right)^{-1} A_2^{(k)} + A_2^{(k)} \left(A_1^{(k)}\right)^{-1} \Sigma_1^k \left(A_1^{(k)}\right)^{-1} A_2^{(k)} + o(\varepsilon_k);
\end{aligned}$$

$$
\begin{aligned}
\Sigma_1^{k+1} =& B_1^{(k+1)} - A_1^{(k+1)} \\
=& B_1^{(k)} - B_0^{(k)} \left(B_1^{(k)}\right)^{-1} B_2^{(k)} - B_2^{(k)} \left(B_1^{(k)}\right)^{-1} B_0^{(k)} + \\
& - A_1^{(k)} + A_0^{(k)} \left(A_1^{(k)}\right)^{-1} A_2^{(k)} + A_2^{(k)} \left(A_1^{(k)}\right)^{-1} A_0^{(k)} \\
=& \Sigma_1^k - (A_0^{(k)} + \Sigma_0^k) \left(A_1^{(k)} + \Sigma_1^k\right)^{-1} (A_2^{(k)} + \Sigma_2^k) + A_0^{(k)} \left(A_1^{(k)}\right)^{-1} A_2^{(k)} + \\
& - (A_2^{(k)} + \Sigma_2^k) \left(A_1^{(k)} + \Sigma_1^k\right)^{-1} (A_0^{(k)} + \Sigma_0^k) + A_2^{(k)} \left(A_1^{(k)}\right)^{-1} A_0^{(k)} \\
=& \Sigma_1^k - (A_0^{(k)} + \Sigma_0^k) \left(A_1^{(k)}\right)^{-1} \left(I + \Sigma_1^k \left(A_1^{(k)}\right)^{-1}\right)^{-1} (A_2^{(k)} + \Sigma_2^k) + A_0^{(k)} \left(A_1^{(k)}\right)^{-1} A_2^{(k)} + \\
& - (A_2^{(k)} + \Sigma_2^k) \left(A_1^{(k)}\right)^{-1} \left(I + \Sigma_1^k \left(A_1^{(k)}\right)^{-1}\right)^{-1} (A_0^{(k)} + \Sigma_0^k) + A_2^{(k)} \left(A_1^{(k)}\right)^{-1} A_0^{(k)} \\
=& \Sigma_1^k - (A_0^{(k)} + \Sigma_0^k) \left(A_1^{(k)}\right)^{-1} \left(I - \Sigma_1^k \left(A_1^{(k)}\right)^{-1}\right) (A_2^{(k)} + \Sigma_2^k) + A_0^{(k)} \left(A_1^{(k)}\right)^{-1} A_2^{(k)} + \\
& - (A_2^{(k)} + \Sigma_2^k) \left(A_1^{(k)}\right)^{-1} \left(I - \Sigma_1^k \left(A_1^{(k)}\right)^{-1}\right) (A_0^{(k)} + \Sigma_0^k) + A_2^{(k)} \left(A_1^{(k)}\right)^{-1} A_0^{(k)} + o(\varepsilon_k) \\
=& \Sigma_1^k - A_0^{(k)} \left(A_1^{(k)}\right)^{-1} \Sigma_2^k - \Sigma_0^k \left(A_1^{(k)}\right)^{-1} A_2^{(k)} + A_0^{(k)} \left(A_1^{(k)}\right)^{-1} \Sigma_1^k \left(A_1^{(k)}\right)^{-1} A_2^{(k)} + \\
& - A_2^{(k)} \left(A_1^{(k)}\right)^{-1} \Sigma_0^k - \Sigma_2^k \left(A_1^{(k)}\right)^{-1} A_0^{(k)} + A_2^{(k)} \left(A_1^{(k)}\right)^{-1} \Sigma_1^k \left(A_1^{(k)}\right)^{-1} A_0^{(k)} + o(\varepsilon_k).
\end{aligned}
$$

Le equazioni precedenti permettono di ottenere le stime volute. Siano, infatti

- $\delta_k := \max\{ \|A_0^{(k)}\|, \|A_2^{(k)}\| \}$,
- $\gamma_k := \left\| \left(A_1^{(k)}\right)^{-1} \right\|$,

allora

$$\varepsilon_{k+1} \leq \{ \delta_k \gamma_k (2 + \delta_k \gamma_k), 1 + 2\delta_k \gamma_k (2 + \delta_k \gamma_k) \} \varepsilon_k + o(\varepsilon_k). \tag{5.25}$$

Alla luce del teorema 3.2.11, inoltre, valgono le relazioni

$$\gamma_k \leq \gamma, \qquad \delta_k \leq \sigma^{2^k}$$

dove γ è una costante limitata e $\sigma < 1$. Mettendo insieme le precedenti stime si ottiene

$$\varepsilon_{k+1} \leq (1 + \xi \sigma^{2^k}) \varepsilon_k,$$

si conclude, quindi, che l'errore al passo $(k+1)$-esimo si mantiene dello stesso ordine dell'errore maturato al passo k-esimo. Svolgendo calcoli analoghi a quelli svolti nella sezione 4.1.4, è possibile esplicitare la costante che esprime il fattore di amplificazione dell'errore generato dal procedimento di troncamento e compressione. Si ha quindi

$$\lim_{k \to \infty} \varepsilon_k \leq \frac{1}{1 - \xi \sigma^2} \tau + O(\tau),$$

si deduce dunque che l'evoluzione dell'errore generato dal metodo $\mathcal{CR}_{ls}$ si mantiene sotto controllo senza dare luogo a fenomeni di degenerazione.

5.1.5 Costo computazionale

Il metodo $\mathcal{CR}_{ls}$ formulato nelle precedenti sezioni si compone sostanzialmente di quattro 'componenti'

- inizializzazione,

- iterazione dell'algoritmo,
- prcedimento di troncamento e compressione,
- condizioni d'arresto e controllo di convergenza,
- risoluzione del sistema (5.5).

Ciascuna di tali componeni è stata elaborata con il preciso intento di tenere il costo computazionale nella soglia delle $O(n)$ (nel caso $n = m$) operazioni aritmetiche limitando al minimo l'impiego di memoria richiesto. La presente sezione dà un quadro dettagliato del numero di operazioni e delle unità di memoria di cui necessita l'algoritmo.

Si assuma, come già fatto nella sezione 4.1.5, che il calcolo della fattorizzazione QR con pivoting di una matrice $n \times s$ abbia un costo computazionale di $4ns^2$ operazioni elementari.

Per quanto rigurada l'inizializzazione si hanno al solito due varianti, se si adopera una trasformazione affine $\mathcal{A}_\alpha$ il calcolo dei dati iniziali si riduce alle computazioni delle matrici $\alpha\tilde{A} - I_n$ e $-\alpha\tilde{D} - I_m$, il cui costo è di $2n$ operazioni aritmetiche. Più complessa è l'inizializzazione a seguito di una trasformazione di Cayley:

- per il calcolo delle matrici $\tilde{A_0}^{(0)}$, $\tilde{A_1}^{(0)}$ si rendono necessarie $4n$ operazioni, mentre la matrice $\tilde{A_2}^{(0)}$ non necessita di alcun calcolo,
- il calcolo della matrice $U_0^{(0)}$ necessita di $n(l+h)$ operazioni, mentre il calcolo della matrice $V_0^{(0)}$ richiede lo svolgimento di $8n(l^2+h^2)$ operazioni, trascurabile è invece il calcolo della matrice Λ,
- simmetrico è il calcolo delle matrici $U_1^{(0)}$ e $V_1^{(1)}$ i quali richiedono rispettivamente $8n(l^2+h^2)$ e $n(l+h)$ operazioni, marginale è la computazione della matrice Γ.

Si conclude, quindi, che per inizializzare il metodo adoperando una trasformazione di Cayley si rendono necessarie $2(16l^2 + 16h^2 + l + h + 2)n$ operazioni aritmetiche. Per ciascuna delle trasformazioni, dunque, si raggiunge il target delle $O(n)$ operazioni come limite massimale.

Il corpo dell'iterazione del metodo $\mathcal{CR}_{ls}$ è sviluppato in modo differente dal metodo $\mathcal{SDA}_{ls}$. La struttura di matrice diagonale più matrice di rango basso data alle matrici che definiscono il metodo rende particolarmente semplificate le operazioni di prodotto matrice-vettore o le risoluzioni di sistemi lineari, dunque non è più necessario come per il metodo $\mathcal{SDA}_{ls}$ svolgere ricorsivamente i calcoli in quanto le proprietà desiderate sono immediatamente spendibili nell'iterazione corrente. Tale caratteristica dell'algoritmo consente, come meglio specificato in seguito, di avere un dispendio di memoria minore rispetto al metodo $\mathcal{SDA}_{ls}$.

Il corpo centrale dell'algoritmo richiede

- $4(2l_1^k + 1)l_1^k n + 2n$ operazioni elementari per l'inversione della matrice $A_1^{(k)}$ così composte
  - $2n$ operazioni per il calcolo della matrice $\tilde{B}^{(k)}$;
  - $2l_1^k n$ operazioni per il calcolo di $Z^{(k)}$,
  - $2(2l_1^k + 1)l_1^k n$ operazioni per il calcolo di $W^{(k)}$,
  - $4{l_1^k}^2$ operazioni per il calcolo dela matrice $Y^{(k)}$;
- $16n$ operazioni per il calcolo delle matrici diagonali $\tilde{A_i}^{(k+1)}$, per $i = 0, 1, 2$, così ripartite
  - $4n$ operazioni per $\tilde{A_0}^{(k+1)}$,
  - $8n$ operazioni per $\tilde{A_1}^{(k+1)}$,

  - $4n$ operazioni per $\tilde{A_2}^{(k+1)}$;

- $4l_1^k n + 4(4l_i^k l_0^k + 3)l_0^k n + 4(4l_i^k l_2^k + 3)l_2^k n$ operazioni aritmetiche dovute al calcolo della correzione di rango $U_i^{(k+1)}$ suddivise come segue

  - $2l_1^k n$ operazioni per la computazione di $\tilde{A_j}^{(k)} Z^{(k)}$, per $j = 0, 2$,
  - $2(4l_i^k l_{j_2}^k + 3)l_{j_2}^k n$ per la computazione di $U_{j_1 j_2}^{(k)}$, per $j_1, j_2 = 0, 2$,

- $4l_1^k n + 4(4l_i^k l_0^k + 3)l_0^k n + 4(4l_i^k l_2^k + 3)l_2^k n$ operazioni aritmetiche dovute al calcolo della correzione di rango $V_i^{(k+1)}$ (calcoli analoghi a quelli svolti per $U_i^{(k+1)}$);

- all'incirca
$$8n + 8(l_0^k + l_2^k)(1 + l_1^k)n + 8(l_0^k + l_2^k)^2 n,$$
operazioni per il calcolo delle matrici $T_i^{(k)}$, in quanto il calcolo delle matrici $T_{j_1 j_2}^{(k)}$ comporta
$$2n + 2l_{j_1}^k n + 4l_{j_1}^k l_{j_2}^k n + 4(l_{j_1}^k + l_{j_2}^k)l_1^k n$$
operazioni.

Il procedimento di troncamento e compressione richiede in tutto $2(8l_0^k + 8l_1^k + 8l_2^k + 1)n$ operazioni aritmetiche così ripartite

- il calcolo delle fattorizzazioni QR delle matrici $U_i^{(k)}$ e $V_i^{(k)}$ per $i = 0, 1, 2$ comporta un costo di $16({l_0^k}^2 + {l_1^k}^2 + {l_2^k}^2)n$ operazioni elementari,

- il troncamento delle matrici $\tilde{A_0}^{(k)}$ e $\tilde{A_2}^{(k)}$ ha un costo di $4n$ operazioni aritmetiche.

Infine, per il calcolo delle condizioni d'arresto e del controllo di convergenza si hanno le seguenti stime dei costi

- le condizioni d'arresto risultano gratuite in quanto già calcolate in sede di troncamento e compressione,

- il calcolo del controllo di convergenza richiede

  - $12n$ operazioni elementari per il calcolo delle norme $\|\tilde{A_i}^{(k)} - \tilde{A_i}^{(k-1)}\|$,
  - $32({l_0^k}^2 + {l_1^k}^2 + {l_2^k}^2)n$ per il calcolo delle fattorizzazioni QR con pivoting delle matrici $\bar{U_i}^{(k)}$ e $\bar{V_i}^{(k)}$ per $i = 0, 1, 2$.

A condizione d'arresto raggiunta, andrebbe inoltre risollto il sistema lineare (5.5). sfruttando opportunamente la struttura delle matrici e la $\mathcal{SMWF}$ è possibile calcolare l'approssimazione X_k in $O(n)$ operazioni aritmetiche. Tali operazioni aggiuntive, verrebbero, ovviamente ammortizzate nel corso dell'algoritmo.

Per quanto rigurada l'impiego di memoria richiesto, contrariamente a quanto accade nel metodo $\mathcal{SDA}_{ls}$, il metodo $\mathcal{CR}_{ls}$ non necessita di alcun calcolo ricorsivo dei prodotti matrice-vettore, ciò comporta una diminuzione del numero di prodotti da eseguire e un impiego decisamente minore di memoria Si osservi, infatti, che per implementare il metodo, è sufficiente registrare ad ogni iterazione i valori delle matrici $A_i^{(k)}$ per $i = 0, 1, 2$. È dunque necessario memorizzare le matrici diagonali $\tilde{A_i}^{(k)}$ e le matrici di rango basso $U_i^{(k)}$, $V_i^{(k)}$, $T_i^{(k)}$, quindi l'impiego di memoria richiesto è di

$$2(3 + 2l_0^k + 2l_1^k + 2l_2^k)n$$

unità elementari.

Si conclude, quindi, che il metodo $\mathcal{CR}_{ls}$ ha un costo computazionele sostenibile e non necessita di un uso spropositato di memoria, presenta dunque delle caratteristiche numeriche idonee per la risoluzione di problemi di grandi dimensioni.

5.2 Implementazioni

Il presente paragrafo illustra i codici MATLAB relativi al metodo $\mathcal{CR}_{ls}$. Anche in tal caso si è scelto di riportare i frammenti dei codici delle parti più significative e dare solo lo pseudocodice per la descrizione globale dell'algoritmo. Sono quindi riportati risultati delle implementazioni, sottolineando ancora una volta la centralità della scelta dei parametri τ e l_{max} per la buona riuscita delle sperimentazioni. Al solito, è valutata l'incidenza dei parametri del processo di troncamento e compressione su

- errore relativo,
- crescita delle correzioni di rango,
- tempi di utilizzo della CPU.

Il codice sottostante genera casualmente, mediante il comando `rand` e `sprand`, i problemi di grandi dimensioni necessari per le sperimentazioni. Il programma ha come input la dimensione n delle matrici e come output i coefficienti di una $\mathcal{NARE}$ compatibile con le ipotesi (5.1). La $\mathcal{NARE}$ viene costruita prendendo come modello l'equazione che definisce il moto di particelle lungo un'asta illustrato nel paragrafo 1.2.

Listing 5.1: Problema di grandi dimensioni.

```
function[At, A1,A2,S, Dt, D1,D2,P, B1, B2,R, C1, C2,T] = large(n)

%  il programma prende in input la dimensione delle matrici e
%  genera i coefficienti di una NARE che descrive il moto di
%  particelle lungo un asta come illustrato nella sezione 1.2.
%  si osservi che le matrici diagonali At e Dt sono definite
%  come vettori.


Del1 = rand(n,1);
Del2 = rand(n,1);


Bet1 = sprand(n,2,1/n);
Bet2 = rand(n,2);

Phi1 = sprand(n,2,1/n);
Phi2 = rand(n,2);

Bet = Bet1*Bet2';
Phi = Phi1*Phi2';

%  il ciclo for riscala le matrici Phi e Bet in modo che la matrice
%  dei coefficienti risulti una M-matrice

for j= 1:n
    s=0;
    for i=1:2
        s= s + Phi(j,i) + Bet(j,i);
    end
    if s>= 1
        for i=1:2
            Phi1(j,i)= Phi1(j,i)/(4*s);
            Phi2(j,i)= Phi2(j,i)/(4*s);
            Bet1(j,i)= Bet1(j,i)/(4*s);
            Bet2(j,i)= Bet2(j,i)/(4*s);
        end
    end
```

```
end


At = Del1;
Dt = Del2;

A1 = Phi1;
D1 = Phi1;

B1 = Bet1;
C1 = Bet1;

A2 = zeros(n,2);
D2 = zeros(n,2);

for i=1:n
    for j=1:2
        A2(i,j) = Del1(i)*Phi2(i,j);
        D2(i,j) = Del2(i)*Phi2(i,j);
    end
end

for i=1:n
    for j=1:2
        B2(i,j) = Del2(i)*Bet2(i,j);
        C2(i,j) = Del1(i)*Bet2(i,j);
    end
end

S = -eye(2);
P = -eye(2);
R = -eye(2);
T = -eye(2);
```

Di seguito si riporta lo pseudocodice del metodo $\mathcal{CR}_{ls}$. L'algoritmo prende in input le matrici generate dal codice precedente e restituisce la soluzione X della $\mathcal{NARE}$. I passi da seguire lungo l'esecuzione del codice sono:

1. inizializzazione,
2. calcolo ricorsivo delle matrici,
3. troncamento e compressione delle correzioni di rango,
4. troncamento delle matrici diagonali,
5. calcolo della norma delle matrici $A_i^{(k)}$ per $i = 0, 2$,
6. verifica delle condizioni d'arresto.

Listing 5.2: Metodo $\mathcal{CR}_{ls}$ pseudocodice.

```
Input:      At, A1, A2, S;
            Dt, D1, D2, P;
            B_1, B_2, R;
            C_1, C_2, T;
            tau;
            l_m,
            tol;

 % tau = parametro di troncamento
 % l_m = parametro di compressione
```

```
% tol = soglia tolleranza per l'errore

Output:  X

% X soluzione delle NARE

Inizializzare il metodo come espresso dal codice 5.2;
Porre k=0, tol= 2eps

% k = numero iterazione
% rel =  parametro d'arresto del metodo

while tol < eps

    [B, Z, W, Y]= smwf(At1,U1,V1,T1)

    calcolare ricorsivamente le matrici
    At0, U_0^(k+1), V_0^(k+1), T_0^(k+1);
    At1, U_1^(k+1), V_1^(k+1), T_1^(k+1);
    At2, U_2^(k+1), V_2^(k+1), T_2^(k+1);

    [Q_(Ui), M_(Vi)] = tron_compr(U_i^(k+1), tau, l_m);    i=0,1,2
    [Q_(Vi), M_(Vi)] = tron_compr(V_i^(k+1), tau, l_m);    i=0,1,2

    At0 = tron(At0, tau);
    At2 = tron(At2, tau);

    Modificare T_i^(k+1);  i=0,1,2

    tol = min( max(Ati), norm(T_i^(k+1)));  i=0,2

    k = k+1;
end

Calcolare X a partire dalle matrici U1, T1, V1
```

Il codice seguente illustra l'inizializzazione del metodo a partire da una trasformazione affine. È importante osservare che le matrici diagonali utilizzate dal metodo $\mathcal{CR}_{ls}$ sono per il calcolatore semplici vettori: tale scelta rende meno leggibili i calcoli, ma comporta un notevole risparmio di unità di memoria.

Listing 5.3: Metodo $\mathcal{CR}_{ls}$ inizializzazione con trasformazione affine.

```
%  il frammento di codice illustra l'inizializzazione del metodo
%  utilizzando una trasforazione affine

n = size(At);
k = size(S,1);
l = size(T,1);

alpha = 1/max(At + diag(A1*S*A2'));

At0 = zeros(2*n,1);
At0(1:n) = alpha*At - ones(n);


U0 = [A1, zeros(n,l); zeros(n,k), C1];
V0 = [A2, C2; zeros(n,l+k)];
T0 = [alpha*S, zeros(k,l); zeros(l,k), -alpha*T];
```

```
At1 = -ones(2*n,1);
At0(1:n) = -alpha*Dt - ones(n);

U1 = [B1, zeros(n,k); zeros(n,l), D1];
V1 = [zeros(n,l+k); B2, D2];
T1 = [-alpha*R, zeros(l,k); zeros(k,l), -alpha*P];


At2 = zeros(2*n,1);
At2(n+1:2*n) = -eye(n);

U2 = [];
V2 = [];
T2 = [];
```

Il codice di seguito riportato mostra come calcolare efficientemente l'inversa di matrici del tipo 'diagonale più rango basso' adoperando la $\mathcal{SMWF}$. Si osservi che anche in tal caso le matrici diagonali sono in realtà dei vettori.

Listing 5.4: Formula di Scherman-Morrison-Woodbury.

```
function[B, Z, W, Y]= smwf(A,U,V,T)

%  il programma calcola l'inversa della matrice
%                 diag(A) + U*T*V'
%  utilizzando la smwf.
%  L'output è definito dalla relazione
%       (diag(A) + U*T*V')^(-1) = B + Z*Y*W



n = size(A,1);
l = size(U,2);

B = zeros(n,1);
for i=1:n
    B(i) = 1/A(i,1);
end

Z = zeros(n,l);
W = V*T';

for i=1:n
    for j=1:l
        Z(i,j) = B(i)*U(i,j);
        W(i,j) = B(i)*W(i,j);
    end
end

Y = - inv(eye(l) + W'*U);
```

I codici che seguono attuano il procedimento di troncamento e compressione per le correzioni di rango ed il procedimento di troncamento per le matrici diagonali.

Listing 5.5: Processo di troncamento e compressione.

```
function[Q, R] = tron_compr(U, tau, l_m)

% il programma calcola il risultato del procedimento di troncamento e
% compressione della matrice U con
%
%  - tau parametro  troncamento,
```

```
%  - l_m parametro compressione

[Q,R,E] = qr(U,0);
n = size(U,1);
l = size(U,2);

r = 1;
i = 1;

while (i < min(l,l_m + 1))
    if abs(R(i,i)) > tau
        i = i+1;
    else
        r = i-1;
        i = n+1;
    end
end
Q = Q(:,1:r);
R = R(1:r,:);
R = R*E';
```

Listing 5.6: Processo di troncamento di matrici diagonali.

```
function[B] = tron(A, tau)

%  il programma calcola il troncamento della matrice diagonale A
%  utilizzando come soglia il parametro di troncamento tau


n = size(A);

B = A;

for i=1:n
    if (A(i) < tau)
        B(i) = 0;
    end
end
```

Esempio 5.2.1. *Si consideri una $\mathcal{NARE}$ che descrive il moto di particelle lungo un'asta omogenea (cfr. paragrafo 1.2) generato dal codice 5.1 ponendo $n = 2000$. Sono svolte due sperimentazioni numeriche con valori diversi dei parametri che definiscono il procedimento di troncamento e compressione. Nella prima implementazione si è scelto*

$$\tau_0, \tau_1, \tau_2 \leq 1.e-10 \qquad l_{max} = 200,$$

i risultati ottenuti sono riassunti nella tabella sottostante

iterazione	*errore relativo*	*rango* $T_0^{(k)}$	*rango* $T_1^{(k)}$	*rango* $T_1^{(k)}$	*tempo CPU*
1	$9.5371e-04$	4	4	0	$8.344e-02$
2	$7.9171e-04$	12	20	2	$2.731e-01$
3	$4.9427e-05$	44	88	24	$4.707e-01$
4	$2.4021e-05$	96	200	54	$1.452e-00$
5	$8.8797e-06$	124	200	82	$9.834e-00$
6	$6.2365e-06$	88	200	76	$2.625e+01$
7	$3.6264e-07$	52	200	41	$6.433e+01$
8	$4.6642e-08$	45	200	27	$9.542e+01$
9	$5.4672e-09$	36	200	18	$1.116e+02$
10	$1.8459e-09$	20	200	8	$3.162e+02$
11	$4.4564e-10$	8	200	1	$5.938e+02$
12	$3.9574e-11$	2	200	1	$8.230e+02$

Nella seconda sperimentazione si è scelto

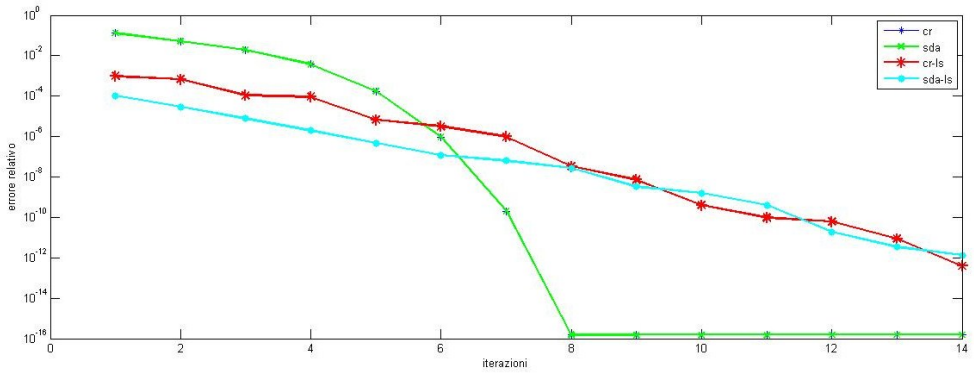
$$\tau^B, \tau^C \leq 1.e-12 \qquad l_{max}, h_{max} = 240,$$

e si sono conseguiti i risultati illustrati in tabella

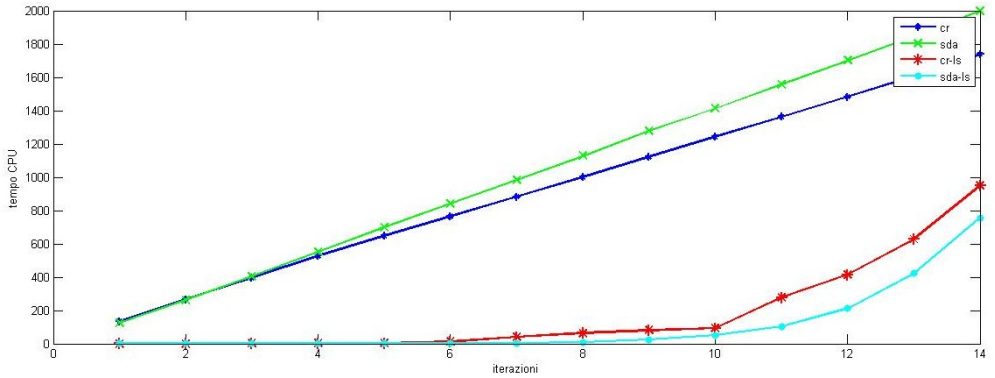
iterazione	*errore relativo*	*rango* $T_0^{(k)}$	*rango* $T_1^{(k)}$	*rango* $T_1^{(k)}$	*tempo CPU*
1	$9.7017e-04$	4	4	0	$9.333e-02$
2	$6.6935e-04$	12	20	2	$3.231e-01$
3	$1.0965e-04$	44	88	24	$1.917e-00$
4	$9.2542e-05$	96	220	54	$2.960e-00$
5	$6.7213e-06$	143	220	87	$4.279e-00$
6	$3.1882e-06$	182	220	153	$1.302e+01$
7	$9.9178e-07$	97	220	72	$3.970e+01$
8	$3.2555e-08$	56	220	34	$6.410e+01$
9	$7.3235e-09$	32	220	21	$8.136e+01$
10	$4.0995e-10$	24	220	13	$9.448e+01$
11	$9.7200e-11$	14	220	5	$2.787e+02$
12	$6.1365e-11$	6	220	2	$4.141e+02$
13	$8.5615e-12$	3	220	1	$6.277e+02$
14	$3.8489e-13$	2	220	1	$9.491e+02$

Le tabelle precedenti illustrano l'incidenza dei parametri che definiscono il processo di troncamento e compressione per la buona riuscita dell'algoritmo: nella prima sperimentazione si 'rilassano' i valori di τ *e* l_{max} *ottenendo soluzioni con precisione dell'ordine* $1e-11$ *in circa* 820 *secondi, nella seconda, invece, i parametri sono più 'restrittivi', si ha, dunque, un'accuratezza migliore dell'ordine* $1e-13$ *a fronte di un utilizzo della CPU di circa* 950 *secondi. Al solito, spetta allo sperimentatore individuare il miglior compromesso tra precisione ed efficienza.*

Il grafico sottostante compara l'andamento dell'errore relativo lungo le iterazioni per i metodi $\mathcal{SDA}$ *e* $\mathcal{CR}$ *e per le rispettive versioni large-scale (con i parametri adottati nella seconda sperimentazione)*



Il grafico seguente mostra, invece, l'evoluzione al variare delle iterazioni del tempo di utilizzo della CPU per ciascun metodo.



5.3 Commenti e conclusioni

L'obiettivo di tale elaborato è descrivere nel dettaglio le modifiche apportate da Chang-Yi Weng, Tiexiang Li, Eric King-wah Chu e Wen-Wei Lin al metodo $\mathcal{SDA}$, capirne la filosofia e ideare una analoga strategia per tentare di ottimizzare il metodo $\mathcal{CR}$. Lo scrivente, dunque, ha cercato di realizzare un algoritmo che ripondesse alle esigenze di costo computazionale ed impiego di memoria imposte dall'ordine di grandezza del problema.

Ovviamente, come in ogni lavoro di sperimentazione, sono state innumerevoli le versioni elaborate prima di arrivare alla definitiva. L'ultima stesura è però apparsa la migliore in quanto interpreta al meglio le potenzialità della $\mathcal{SMWF}$: si è dunque scelto di adottare per le matrici $A_i^{(k)}$ una struttura di matrice diagonale più matrice di rango basso. Tale scelta comporta però, come evidente dalla lunga serie di calcoli esposti nella sezione 5.1.2, una corposa crescita delle correzioni di rango, ad ogni iterazione, infatti, le nuove correzioni sono composte da tre o addirittura da sette blocchi matriciali. Tale situazione rappresenta indubbiamente un punto debole del metodo, in quanto potenzialmente le dimensioni diverrebbero insostenibili dopo poche iterazioni dell'algoritmo. A sostegno del metodo viene in 'soccorso' il meccanismo di troncamento e compressione: tale procedimento mostra le stesse falle teoriche già diffusamente trattate nel capitolo 4, ma risulta uno strumento fondamentale per un corretto funzionamento dell'algoritmo. Intuitivamente le stime asintotiche (5.21) dovrebbero rassicurare lo sperimentatore sull'esistenza delle decomposizioni volute, e l'intuizione è tra l'altro confermata dalle sperimentazioni, tuttavia non è possibile approfondire con il rigore del caso la natura dei paramteri τ_i e l_{max}.

Per quanto riguarda la stima dell'amplificazione dell'errore dovuto al procedimento di troncamento e compressione, si è raggiunto un risultato analogo a quello ottenuto per il metodo $\mathcal{SDA}_{ls}$. Questo è dovuto alle proprietà di convergenza sostanzialmente

equivalenti in entrambi i metodi (non per altro il metodi $\mathcal{SDA}$ e $\mathcal{CR}$ sono chiamati doubling algorithm).

Da ultimo l'autore della presente tesi vuole manifestare la propria soddisfazione per quanto svolto, non tanto per il valore del lavoro (la cui valutazione spetta ovviamente ad altri), quanto per la passione e per il piacere con cui è stato redatto l'elaborato. È stato davvero stimolante ingegnarsi nella creazione di nuovi algoritmi sperimentando immediatamente al calcolatore le buona riuscita o meno di determinate intuizioni. Anche un settore apparentemente rigido e schematico come l'analisi numerica richiede in realtà un duro sforzo di creatività. È doveroso, quindi, ringraziare il Prof. Bini per aver, da relatore della presente tesi, assecondato tale 'entusiasmo' mostrando grande disponibilità ed attenzione, e per aver dato il supporto teorico ed umano basilare per le la riuscita dell'elaborato.

Bibliografia

[1] E. Arias, V. Hernández, J. J. Ibáñez, and J. Peinado. A fixed point-based BDF method for solving differential Riccati equations. *Appl. Math. Comput.*, 188(2):1319–1333, 2007.

[2] U. M. Ascher and L. R. Petzold. *Computer methods for ordinary differential equations and differential-algebraic equations*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1998.

[3] R. Bartels and G. Stewart. Solution of the matrix equation $AX + XB = C$. *Commun. ACM*, 15(9):820–826, 1972.

[4] P. Benner and R. Byers. Evaluating products of matrix pencils and collapsing matrix products. *Numer. Linear Algebra Appl.*, 8(6-7):357–380, 2001. Numerical linear algebra techniques for control and signal processing.

[5] A. Berman and R. J. Plemmons. *Nonnegative matrices in the mathematical sciences*, volume 9 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1994. Revised reprint of the 1979 original.

[6] D. P. Bertsekas. *Dynamic programming and optimal control. Vol. I*. Athena Scientific, Belmont, MA, third edition, 2005.

[7] R. Bevilacqua, D. A. Bini, M. Capovani, and O. Menchi. *Metodi numerici*. Zanichelli, 1992.

[8] D. Bini, M. Capovani, and O. Menchi. *Metodi numerici per l'algebra lineare*. Nicola Zanichelli Editore S.p.A., Bologna, 1988.

[9] D. Bini and B. Meini. On the solution of a nonlinear matrix equation arising in queueing problems. *SIAM J. Matrix Anal. Appl.*, 17(4):906–926, 1996.

[10] D. A. Bini, B. Iannazzo, and B. Meini. *Numerical solution of algebraic Riccati equations*, volume 9 of *Fundamentals of Algorithms*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2012.

[11] D. A. Bini, G. Latouche, and B. Meini. *Numerical methods for structured Markov chains*. Numerical Mathematics and Scientific Computation. Oxford University Press, New York, 2005. Oxford Science Publications.

[12] D. A. Bini and B. Meini. The cyclic reduction algorithm: from Poisson equation to stochastic processes and beyond. In memoriam of Gene H. Golub. *Numer. Algorithms*, 51(1):23–60, 2009.

[13] D. A. Bini, B. Meini, and F. Poloni. Transforming algebraic Riccati equations into unilateral quadratic matrix equations. *Numer. Math.*, 116(4):553–578, 2010.

[14] B. L. Buzbee, G. H. Golub, and C. W. Nielson. On direct methods for solving Poisson's equations. *SIAM J. Numer. Anal.*, 7:627–656, 1970.

[15] C.-Y. Chiang, E. K.-W. Chu, C.-H. Guo, T.-M. Huang, W.-W. Lin, and S.-F. Xu. Convergence analysis of the doubling algorithm for several nonlinear matrix equations in the critical case. *SIAM J. Matrix Anal. Appl.*, 31(2):227–247, 2009.

[16] C. H. Choi and A. J. Laub. Constructing Riccati differential equations with known analytic solutions for numerical experiments. *IEEE Trans. Automat. Control*, 35(4):437–439, 1990.

[17] C. H. Choi and A. J. Laub. Efficient matrix-valued algorithms for solving stiff Riccati differential equations. *IEEE Trans. Automat. Control*, 35(7):770–776, 1990.

[18] E. K.-W. Chu, H.-Y. Fan, and W.-W. Lin. A structure-preserving doubling algorithm for continuous-time algebraic Riccati equations. *Linear Algebra Appl.*, 396:55–80, 2005.

[19] E. K.-W. Chu, H.-Y. Fan, W.-W. Lin, and C.-S. Wang. Structure-preserving algorithms for periodic discrete-time algebraic Riccati equations. *Internat. J. Control*, 77(8):767–788, 2004.

[20] J. C. Engwerda, A. C. M. Ran, and A. L. Rijkeboer. Necessary and sufficient conditions for the existence of a positive definite solution of the matrix equation $X + A^*X^{-1}A = Q$. *Linear Algebra Appl.*, 186:255–275, 1993.

[21] G. H. Golub and C. F. Van Loan. *Matrix computations*, volume 3 of *Johns Hopkins Series in the Mathematical Sciences*. Johns Hopkins University Press, Baltimore, MD, second edition, 1989.

[22] C.-H. Guo. Nonsymmetric algebraic Riccati equations and Wiener-Hopf factorization for M-matrices. *SIAM J. Matrix Anal. Appl.*, 23(1):225–242 (electronic), 2001.

[23] C.-H. Guo. Convergence analysis of the Latouche-Ramaswami algorithm for null recurrent quasi-birth-death processes. *SIAM J. Matrix Anal. Appl.*, 23(3):744–760 (electronic), 2001/02.

[24] C.-H. Guo. A note on the minimal nonnegative solution of a nonsymmetric algebraic Riccati equation. *Linear Algebra Appl.*, 357:299–302, 2002.

[25] C.-H. Guo. On a quadratic matrix equation associated with an M-matrix. *IMA J. Numer. Anal.*, 23(1):11–27, 2003.

[26] C.-H. Guo. Efficient methods for solving a nonsymmetric algebraic Riccati equation arising in stochastic fluid models. *J. Comput. Appl. Math.*, 192(2):353–373, 2006.

[27] C.-H. Guo and N. J. Higham. Iterative solution of a nonsymmetric algebraic Riccati equation. *SIAM J. Matrix Anal. Appl.*, 29(2):396–412, 2007.

[28] C.-H. Guo, B. Iannazzo, and B. Meini. On the doubling algorithm for a (shifted) nonsymmetric algebraic Riccati equation. *SIAM J. Matrix Anal. Appl.*, 29(4):1083–1100, 2007.

[29] C.-H. Guo and A. J. Laub. On the iterative solution of a class of nonsymmetric algebraic Riccati equations. *SIAM J. Matrix Anal. Appl.*, 22(2):376–391 (electronic), 2000.

[30] X.-X. Guo, W.-W. Lin, and S.-F. Xu. A structure-preserving doubling algorithm for nonsymmetric algebraic Riccati equation. *Numer. Math.*, 103(3):393–412, 2006.

[31] V. Hernández, J. J. Ibáñez, J. Peinado, and E. Arias. A GMRES-based BDF method for solving differential Riccati equations. *Appl. Math. Comput.*, 196(2):613–626, 2008.

[32] N. J. Higham. *Functions of matrices*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2008. Theory and computation.

[33] R. W. Hockney. A fast direct solution of Poisson's equation using Fourier analysis. *J. Assoc. Comput. Mach.*, 12:95–113, 1965.

[34] R. A. Horn and C. R. Johnson. Topics in matrix analysis. pages viii+607, 1994. Corrected reprint of the 1991 original.

[35] A. S. Householder. *The theory of matrices in numerical analysis*. Blaisdell Publishing Co. Ginn and Co. New York-Toronto-London, 1964.

[36] T.-M. Hwang, E. K.-W. Chu, and W.-W. Lin. A generalized structure-preserving doubling algorithm for generalized discrete-time algebraic Riccati equations. *Internat. J. Control*, 78(14):1063–1075, 2005.

[37] V. Kučera. Algebraic Riccati equation: Hermitian and definite solutions. In *The Riccati equation*, Comm. Control Engrg. Ser., pages 53–88. Springer, Berlin, 1991.

[38] P. Lancaster and L. Rodman. *Algebraic Riccati equations*. Oxford Science Publications. The Clarendon Press Oxford University Press, New York, 1995.

[39] G. Latouche and V. Ramaswami. A logarithmic reduction algorithm for quasi-birth-death processes. *J. Appl. Probab.*, 30(3):650–674, 1993.

[40] A. J. Laub. A Schur method for solving algebraic Riccati equations. *IEEE Trans. Automat. Control*, 24(6):913–921, 1979.

[41] A. J. Laub. Schur techniques for Riccati differential equations. In *Feedback control of linear and nonlinear systems (Bielefeld/Rome, 1981)*, volume 39 of *Lecture Notes in Control and Inform. Sci.*, pages 165–174. Springer, Berlin, 1982.

[42] T. Li, P. C.-Y. Weng, E. K.-W. Chu, and W.-W. Lin. Solving large-scale Stein and Lyapunov equations by doubling. *J. Comput. Appl. Math.*, 192(2):353–373, 2011.

[43] T. Li, P. C.-Y. Weng, E. K.-W. Chu, and W.-W. Lin. Solving large-scale Stein and Lyapunov equations by doubling. *J. Comput. Appl. Math.*, 192(2):353–373, 2011.

[44] W.-W. Lin and S.-F. Xu. Convergence analysis of structure-preserving doubling algorithms for Riccati-type matrix equations. *SIAM J. Matrix Anal. Appl.*, 28(1):26–39, 2006.

[45] G. Loria. *Storia delle matematiche dall'alba della civiltà al secolo XIX*. Ulrico Hoepli, Milano, 1950. 2d ed.

[46] R. Mattheij and J. Molenaar. *Ordinary differential equations in theory and practice*, volume 43 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2002. Reprint of the 1996 original.

[47] V. L. Mehrmann. *The autonomous linear quadratic control problem*, volume 163 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag, Berlin, 1991. Theory and numerical solution.

[48] B. Meini. Efficient computation of the extreme solutions of $X + A^*X^{-1}A = Q$ and $X - A^*X^{-1}A = Q$. *Math. Comp.*, 71(239):1189–1204 (electronic), 2002.

[49] F. Poloni. *Algorithms for quadratic matrix and vector equations*, volume 16 of *Tesi. Scuola Normale Superiore di Pisa (Nuova Series) [Theses of Scuola Normale Superiore di Pisa (New Series)]*. Edizioni della Normale, Pisa, 2011. Dissertation, Scuola Normale Superiore, Pisa, 2011.

[50] A. S. Poznyak. *Advanced mathematical tools for automatic control engineers. Vol. 1*. Elsevier B. V., Amsterdam, 2008. Deterministic techniques.

[51] V. Ramaswami. Matrix analytic methods: a tutorial overview with some extensions and new results. In *Matrix-analytic methods in stochastic models (Flint, MI)*, volume 183 of *Lecture Notes in Pure and Appl. Math.*, pages 261–296. Dekker, New York, 1997.

[52] J. Sherman and W. J. Morrison. Adjustment of an inverse matrix corresponding to a change in one element of a given matrix. *Ann. Math. Statistics*, 21:124–127, 1950.

[53] V. Sima. *Algorithms for linear-quadratic optimization*, volume 200 of *Monographs and Textbooks in Pure and Applied Mathematics*. Marcel Dekker Inc., New York, 1996.

[54] W. C. Su and Z. Gajic. Reduced-order solution to the finite-time optimal-control problems of linear weakly coupled systems. *IEEE Trans. Automat. Control*, 36(4):498–501, 1991.

[55] P. C.-Y. Weng, H.-Y. Fan, and E. K.-w. Chu. Low-rank approximation to the solution of a nonsymmetric algebraic Riccati equation from transport theory. *Appl. Math. Comput.*, 219(2):729–740, 2012.

[56] J. H. Wilkinson. *The algebraic eigenvalue problem*. Monographs on Numerical Analysis. The Clarendon Press Oxford University Press, New York, 1988. Oxford Science Publications.