UNIVERSITÀ DEGLI STUDI DI CAGLIARI

DIPARTIMENTO DI MATEMATICA E INFORMATICA

CORSO DI LAUREA MAGISTRALE IN
MATEMATICA

Tecniche di estrapolazione vettoriale per la scelta del parametro di regolarizzazione nella risoluzione di problemi lineari mal posti

Relatore:
Prof. Giuseppe Rodriguez

Tesi di Laurea di:
Andrea Azzarelli
Matricola 60/65/65104

Anno accademico:
2022/2023

Indice

Prefazione **iii**

1 Introduzione **1**

1.1 Problemi lineari mal posti 1

1.1.1 Problemi ai minimi quadrati 4

1.1.2 Equazioni normali 5

1.2 Richiami sugli operatori lineari 5

1.2.1 Pseudoinversa 7

1.3 Singular Value Decomposition 9

1.3.1 SVD e sistemi lineari 10

1.4 Generalized Singular Value Decomposition 11

1.4.1 GSVD e sistemi lineari 12

1.4.2 Scelta di L 13

1.5 Estrapolazione 14

1.5.1 Metodo di Richardson 15

1.5.2 Il metodo di Aitken 18

2 Metodi di regolarizzazione **19**

2.1 Regolarizzazione 20

2.2 Metodi TSVD e TGSVD 21

2.2.1 Truncated SVD 21

2.2.2 Truncated GSVD 22

2.3 Regole di scelta del parametro 23

2.3.1 Principio della discrepanza 23

2.3.2 Curva-L 24

2.3.3 Generalized Cross Validation 25

3 Tecniche di estrapolazione vettoriale **27**

3.1 Minimal Polynomial Extrapolation 28

3.1.1 Derivazione del metodo MPE 28

3.2 Reduced Rank Extrapolation 30
3.2.1 Derivazione del metodo RRE 31
3.3 Vector Epsilon Algorithm 32

4 Tecniche di estrapolazione per la scelta del parametro 35
4.1 L'algoritmo RESC 36

5 Risultati numerici 39
5.1 Confronto delle varianti del metodo 42
5.2 Confronto col metodo RRE-TSVD 44
5.3 Confronto con GCV e Curva-L 46
5.4 Esempi 48

Conclusioni 53

Ringraziamenti 55

Bibliografia 60

Prefazione

La stesura di questa tesi è stata preceduta da un tirocinio di 75 ore, che ho svolto presso il Dipartimento di Matematica e Informatica dell'Università degli Studi di Cagliari, nei mesi di aprile e maggio 2023. In quest'occasione ho collaborato a un tema di ricerca già avviato ad opera dei professori Claude Brezinski[1], Caterina Fenu[2], Michela Redivo Zaglia[3] e Giuseppe Rodriguez[4].

Il tema della ricerca riguarda lo sviluppo di un metodo di regolarizzazione per problemi lineari mal posti, che sfrutta l'estrapolazione vettoriale per la scelta del parametro di regolarizzazione. Il mio contributo è stato rivolto particolarmente alla scrittura dei codici su Matlab [25] e alla sperimentazione delle diverse varianti possibili.

Questo metodo è stato presentato dal Prof. Giuseppe Rodriguez con un intervento alla 25ª Conferenza dell'*International Linear Algebra Society* (MSC07), che si è tenuta a Madrid, dal 12 al 16 giugno 2023. Un articolo sull'argomento è in corso di preparazione.

Nel Capitolo 1 sono descritti alcuni concetti introduttivi sul problema dei minimi quadrati e sull'estrapolazione. A seguire, nel Capitolo 2, viene descritto cosa è un metodo di regolarizzazione e sono presentati quelli utilizzati nella ricerca. Nel Capitolo 3 è presentata la teoria dell'estrapolazione vettoriale con alcuni metodi. Il Capitolo 4 è dedicato alla descrizione del metodo RESC, ideato nella ricerca a cui ho partecipato. Per concludere, il Capitolo 5 riporta i risultati che il metodo ha ottenuto su diversi problemi test.

[1]Département Mathématiques, Université de Lille, Professor Emeritus
[2]Dipartimento di Matematica e Informatica, Università degli Studi di Cagliari
[3]Dipartimento di Matematica, Università degli Studi di Padova
[4]Dipartimento di Matematica e Informatica, Università degli Studi di Cagliari

Capitolo 1

Introduzione

Il seguente lavoro descrive un algoritmo per la risoluzione di problemi inversi mal posti. Essi si ritrovano in molte applicazioni pratiche.

1.1 Problemi lineari mal posti

L'interesse di questo lavoro è rivolto ai modelli lineari:

$$A\mathbf{x} = \mathbf{b}, \tag{1.1}$$

dove $A \in \mathbb{R}^{m\times n}$ è la matrice che esprime le caratteristiche del problema, $\mathbf{b} \in \mathbb{R}^m$ sono i dati e $\mathbf{x} \in \mathbb{R}^n$ è l'incognita. Questa formulazione è adatta a un enorme numero di applicazioni in cui si ha un sistema fisico che comprende: un fenomeno che si vuole rappresentare, i cui parametri sono le entrate della $\mathbf{x}$; uno strumento di misura le cui caratteristiche sono rappresentate nella matrice A; i dati che effettivamente abbiamo a disposizione, che sono le entrate del vettore $\mathbf{b}$.

Chiamiamo **problema diretto** quello in cui sono noti la matrice A e il vettore $\mathbf{x}$. Si genera il risultato $\mathbf{b}$ calcolando $A\mathbf{x}$ e questo procedimento è, in generale, un calcolo semplice. Viene invece definito **problema inverso** quello per cui si vuole invertire il processo diretto, cioè si cerca di recuperare il vettore $\mathbf{x}$ che ha generato i dati $\mathbf{b}$ osservati.

In Matematica Applicata bisogna tuttavia considerare che i numeri che abbiamo a disposizione per effettuare i calcoli non sono mai esatti, ma frutto di varie approssimazioni. Queste sono dovute principalmente alle limitazioni degli strumenti di misura e alla rappresentazione in virgola mobile nei sistemi di calcolo. Il vettore $\mathbf{b}$, pertanto, non è realmente disponibile e si dispone esclusivamente di una sua

approssimazione $\tilde{\mathbf{b}}$ affetta da errore ignoto $\mathbf{e} \in \mathbb{R}^m$:

$$\tilde{\mathbf{b}} = \mathbf{b} + \mathbf{e}.$$

Affinché si possa risolvere il problema inverso, esso deve essere formulato in modo tale da rispettare le caratteristiche di un problema ben posto [35].

Definizione 1.1. Un problema si dice **ben posto** se possiede le seguenti caratteristiche:

a. esistenza: esiste una soluzione,

b. unicità: la soluzione è unica,

c. stabilità: la soluzione dipende con continuità dai dati.

In caso contrario si dice **mal posto**.

Un problema lineare (1.1) è ben posto se, per esempio, $m = n$, A è invertibile. La buona positura del problema non è sufficiente a garantire che la soluzione del problema sia accettabile, infatti, a causa di piccole variazioni sui dati, la soluzione ottenuta potrebbe essere molto diversa da quella originale. Questo effetto di amplificazione degli errori non dipende dall'algoritmo o dal calcolatore utilizzato per trovare la soluzione, ma dal problema stesso. Per dare una misura quantitativa di tale effetto si definisce il numero di condizionamento: esso misura il massimo fattore di amplificazione dell'errore relativo sulla soluzione rispetto all'errore relativo sui dati. Diamo la definizione di numero di condizionamento per il problema (1.1).

Definizione 1.2. Si definisce **numero di condizionamento** di una matrice A la quantità

$$\kappa(A) = \|A\|\|A^{-1}\|\,.$$

Un problema si dice **ben condizionato** quando ha un numero di condizionamento basso[1], diversamente si dice **mal condizionato**. Se un problema è mal condizionato la soluzione che si ottiene potrebbe essere molto distante da quella esatta.

[1]Non esiste una quantità predefinita che determina se il numero di condizionamento è alto o basso, ma dipende dal livello di precisione che cerchiamo nella soluzione.

Nel caso di una matrice A invertibile con un basso numero di condizionamento, la soluzione ottenuta con

$$\mathbf{x} = A^{-1}\tilde{\mathbf{b}}$$

si può considerare una buona approssimazione della soluzione esatta se l'errore $\mathbf{e}$ è piccolo rispetto ai dati.

Un esempio classico di problema mal posto è un'equazione integrale di Fredholm del primo tipo con nucleo a quadrato integrabile [18]

$$\int_a^b K(s,t)f(t)\mathrm{d}t = g(s), \qquad c < s < d,$$

dove $g(s)$ e il nucleo K sono noti e $f(t)$ è la soluzione incognita.

Osserviamo perché questo problema è mal posto considerando una perturbazione piccola a piacere della soluzione:

$$\Delta f(t) = \varepsilon \sin(2\pi p t), \qquad p \in \mathbb{N}, \qquad \varepsilon = \text{cost.}$$

La perturbazione del termine noto g che ci fa avere questa soluzione è

$$\Delta g(s) = \varepsilon \int_a^b K(s,t)\sin(2\pi p t)\mathrm{d}t, \qquad p \in \mathbb{N}.$$

Dal lemma di Riemann-Lebesgue abbiamo che $\Delta g \to 0$ per $p \to \infty$. Per un ϵ fissato, la perturbazione sul termine noto Δg produce la perturbazione Δf nella soluzione, per la quale il rapporto $\|\Delta f\|/\|\Delta g\|$ può crescere arbitrariamente scegliendo un p abbastanza grande e questo mostra che il problema è mal posto.

I problemi che saranno considerati in questo lavoro sono, per la maggior parte, tratti da un *toolbox* di Hansen [21] e derivano dalla discretizzazione di un integrale di questo tipo; di seguito ne mostriamo un esempio.

Esempio 1.1. L'esempio di Shaw [39] considera $t, s \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. L'equazione integrale è un modello unidimensionale di problema di ricostruzione di immagini

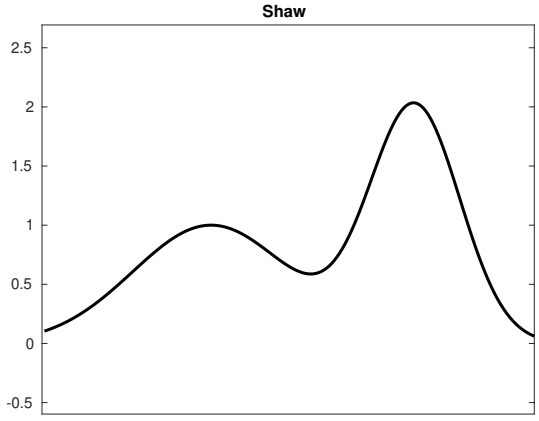
da [2]. Il nucleo K e la soluzione f sono dati da

$$\begin{aligned} K(s,t) &= (\cos(s) + \cos(t))^2 \Big(\frac{\sin(u)}{u}\Big)^2, \\ u &= \pi(\sin(s) + \sin(t)), \\ f(t) &= a_1 \exp\Big(-c_1(t-t_1)^2\Big) + a_2 \exp\Big(-c_2(t-t_2)^2\Big). \end{aligned}$$

Il nucleo K è la *Point Spread Function* per una fessura infinitamente lunga. I parametri a_i, $c_i\, t_i$ determinano la forma della soluzione. Nell'implementazione di Hansen sono scelti in questo modo:

$$\begin{aligned} a_1 = 2, \quad c_1 = 6, \quad t_1 = 0.8, \\ a_2 = 1, \quad c_2 = 2, \quad t_2 = -0.5, \end{aligned}$$

che forniscono una funzione con due picchi.



1.1.1 Problemi ai minimi quadrati

Qualora A sia sovradeterminata, ossia il sistema ha più equazioni che incognite, la soluzione esatta potrebbe non esistere. Invece, se la matrice è sottodeterminata o a rango non pieno vi potrebbero essere infinite soluzioni. Per ovviare a questo problema è possibile utilizzare la sua riformulazione nel senso dei minimi quadrati (vedi [4]):

$$\mathcal{S} := \Big\{\mathbf{x}_{\text{LS}} = \arg\min_{\mathbf{x}\in\mathbb{R}^n} \|A\mathbf{x} - \mathbf{b}\|^2\Big\}; \qquad \mathbf{x}^* := \arg\min_{\mathbf{x}_{\text{LS}}\in\mathcal{S}} \|\mathbf{x}_{\text{LS}}\|^2. \tag{1.2}$$

Un elemento $\mathbf{x}_{\text{LS}} \in \mathcal{S}$ è detto **soluzione dei minimi quadrati** (LS), $\mathbf{x}^*$ è detta **soluzione LS di minima norma**, il vettore $\mathbf{r} = \mathbf{b} - A\mathbf{x}$ è detto **residuo**.

Chiaramente, se l'insieme $\mathcal{S}$ contiene un solo elemento, esso sarà anche quello di minima norma: questo è il caso $m \geq n$ a rango pieno. Nel caso di infinite soluzioni, diventa cruciale la scelta della norma da minimizzare: se la soluzione cercata è continua ha senso utilizzare la norma-2, pertanto, nel resto della discussione, dove non è diversamente specificato, sottointenderemo l'utilizzo della norma euclidea.

1.1.2 Equazioni normali

L'approccio classico per trovare il minimo della funzione $\|A\mathbf{x} - \mathbf{b}\|^2$ consiste nel calcolarne il gradiente e imporlo uguale a zero:

$$0 = \frac{\partial}{\partial \mathbf{x}}(A\mathbf{x}-\mathbf{b})^{\mathsf{T}}(A\mathbf{x}-\mathbf{b}) = \frac{\partial}{\partial \mathbf{x}}(\mathbf{x}^{\mathsf{T}}A^{\mathsf{T}}A\mathbf{x} - 2\mathbf{x}^{\mathsf{T}}A^{\mathsf{T}}\mathbf{b} + \mathbf{b}^{\mathsf{T}}\mathbf{b}) = 2(A^{\mathsf{T}}A\mathbf{x} - A^{\mathsf{T}}\mathbf{b}).$$

Si ottengono così le **equazioni normali** che una soluzione dei minimi quadrati deve soddisfare per realizzare il minimo:

$$A^{\mathsf{T}}A\mathbf{x} = A^{\mathsf{T}}\mathbf{b}.$$

Il problema (1.2), dunque, diventa:

$$\mathcal{S} := \{\mathbf{x} \in \mathbb{R}^n \mid A^{\mathsf{T}}A\mathbf{x} = A^{\mathsf{T}}\mathbf{b}\}; \qquad \mathbf{x}^* := \arg\min_{\mathbf{x}\in\mathcal{S}} \|\mathbf{x}\|. \tag{1.3}$$

Osserviamo che l'argomento del minimo non cambia se consideriamo la norma semplice o al quadrato.

1.2 Richiami sugli operatori lineari

Dato A operatore lineare tra due spazi di Hilbert $\mathbb{R}^n$ e $\mathbb{R}^m$, sappiamo che, fissata la base per ciascuno dei due spazi, A è identificabile con una matrice appartenente a $\mathbb{R}^{m\times n}$. Possiamo definire i seguenti sottospazi vettoriali fondamentali per una matrice $A \in \mathbb{R}^{m\times n}$.

Definizione 1.3.

- Il ***range*** di A è l'insieme di tutti gli elementi del codominio che sono immagine di qualche elemento del dominio tramite A, cioè

$$\mathcal{R}(A) = \Big\{A\mathbf{x} \in \mathbb{R}^m \mid \mathbf{x} \in \mathbb{R}^n\Big\}.$$

- Lo **spazio nullo** o **nucleo** di A è l'insieme di tutti gli elementi del dominio la cui immagine è il vettore nullo del codominio, cioè

$$\mathcal{N}(A) = \Big\{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} = \mathbf{0}_m\Big\}.$$

Ricordiamo alcuni importanti risultati di algebra lineare.

Proposizione 1.4. A è iniettiva se e solo se $\mathcal{N}(A) = \{\mathbf{0}\}$.

Proposizione 1.5.

$$\mathcal{N}(A)^\perp = \mathcal{R}(A^\intercal),$$
$$\mathcal{R}(A)^\perp = \mathcal{N}(A^\intercal).$$

Proposizione 1.6. Sia $\mathbf{b} = \mathbf{b}_1 + \mathbf{b}_2$ con $\mathbf{b}_1 \in \mathcal{R}(A)$ e $\mathbf{b}_2 \in \mathcal{R}(A)^\perp$, allora

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|A\mathbf{x} - \mathbf{b}\|^2 = \|\mathbf{b}_2\|^2.$$

Dimostrazione. Poiché $\mathbf{b}_1 \in \mathcal{R}(A)$, allora esiste $\bar{\mathbf{x}} \in \mathbb{R}^n$ tale che $A\bar{\mathbf{x}} = \mathbf{b}_1$, quindi per ogni $\mathbf{x} \in \mathbb{R}^n$ chiamiamo $\mathbf{z} = \mathbf{x} - \bar{\mathbf{x}}$.

$$\|A\mathbf{x} - \mathbf{b}\|^2 = \|A\mathbf{z} + A\bar{\mathbf{x}} - \mathbf{b}_1 - \mathbf{b}_2\|^2 = \|A\mathbf{z} - \mathbf{b}_2\|^2 = \|A\mathbf{z}\|^2 + \|\mathbf{b}_2\|^2 \geq \|\mathbf{b}_2\|^2,$$

dove abbiamo utilizzato che $\mathbf{b}_2 \perp A\mathbf{z}$. Quindi

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|A\mathbf{x} - \mathbf{b}\|^2 \geq \|\mathbf{b}_2\|^2,$$

e inoltre

$$\min_{\mathbf{x}\in\mathbb{R}^n} \|A\mathbf{x} - \mathbf{b}\|^2 \leq \|A\bar{\mathbf{x}} - \mathbf{b}\| = \|\mathbf{b}_2\|^2.$$

□

Corollario 1.7. Dalla Proposizione 1.6 segue immediatamente che $\mathbf{x} \in \mathbb{R}^n$ è soluzione dei minimi quadrati se e solo se $A\mathbf{x} = \mathbf{b}_1$.

Proposizione 1.8. Sia $\mathbf{x} \in \mathcal{S}$ soluzione dei minimi quadrati, allora essa è soluzione di minima norma se e solo se $\mathbf{x} \in \mathcal{N}(A)^\perp$.

Dimostrazione. Sia $\mathbf{x} = \mathbf{x}_0 + \mathbf{x}_1$ con $\mathbf{x}_0 \in \mathcal{N}(A)$ e $\mathbf{x}_1 \in \mathcal{N}(A)^\perp$, allora

$$\|A\mathbf{x}_1 - \mathbf{b}\|^2 = \|A\mathbf{x}_1 + A\mathbf{x}_0 - \mathbf{b}\|^2 = \|A\mathbf{x} - \mathbf{b}\|^2.$$

Quindi $\mathbf{x}_1$ è soluzione LS, inoltre

$$\|\mathbf{x}\|^2 = \|\mathbf{x}_0\|^2 + \|\mathbf{x}_1\|^2 \geq \|\mathbf{x}_1\|^2.$$

Essa sarà minima solo quando $\mathbf{x}_0 = \mathbf{0}$, cioè $\mathbf{x} = \mathbf{x}_1 \in \mathcal{N}(A)^\perp$. □

1.2.1 Pseudoinversa

Definizione 1.9. Sia $A \in \mathbb{R}^{m\times n}$, cioè $A\colon \mathbb{R}^n \to \mathbb{R}^m$ definiamo

$$\bar{A} := A|_{\mathcal{N}(A)^\perp}\colon \mathcal{N}(A)^\perp \to \mathcal{R}(A),$$

Poiché $\mathcal{N}(\bar{A}) = \{\mathbf{0}\}$ e abbiamo ristretto il codominio a $\mathcal{R}(A)$, esiste la sua inversa

$$\bar{A}^{-1}\colon \mathcal{R}(A) \to \mathcal{N}(A)^\perp.$$

La **pseudoinversa di Moore-Penrose** $A^\dagger$ di A è definita come l'unica estensione lineare di $\bar{A}^{-1}$ a $\mathbb{R}^m = \left(\mathcal{R}(A) \oplus \mathcal{R}(A)^\perp\right)$ (che esiste per il teorema di Hahn-Banach) tale che

$$\mathcal{N}(A^\dagger) = \mathcal{R}(A)^\perp. \tag{1.4}$$

Osserviamo che per la linearità di $A^\dagger$ e per la (1.4), se scriviamo $\mathbf{b} = \mathbf{b}_1 + \mathbf{b}_2$ con $\mathbf{b}_1 \in \mathcal{R}(A)$ e $\mathbf{b}_2 \in \mathcal{R}(A)^\perp$, allora

$$A^\dagger\mathbf{b} = A^\dagger\mathbf{b}_1 + A^\dagger\mathbf{b}_2 = \bar{A}^{-1}\mathbf{b}_1 \in \mathcal{N}(A)^\perp.$$

Proprietà 1.10. Siano $\mathcal{P}_{\mathcal{N}(A)}$ e $\mathcal{P}_{\mathcal{R}(A)}$ i proiettori ortogonali sugli spazi vettoriali fondamentali di A, allora

a. $AA^\dagger A = A$;

b. $A^\dagger AA^\dagger = A^\dagger$;

c. $A^\dagger A = I - \mathcal{P}_{\mathcal{N}(A)}$;

d. $AA^\dagger = \mathcal{P}_{\mathcal{R}(A)}$;

e. se U, V sono ortogonali, allora $(UAV^\intercal)^\dagger = (VA^\dagger U^\intercal)$.

Teorema 1.11. Il problema (1.2) ha un'unica soluzione di minima norma:

$$\mathbf{x}^\dagger := A^\dagger \mathbf{b}.$$

Inoltre, l'insieme di tutte le soluzioni dei minimi quadrati è $\mathbf{x}^\dagger + \mathcal{N}(A)$.

Dimostrazione. Per il Corollario 1.7 scriviamo l'insieme delle soluzioni LS come

$$\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} = \mathcal{P}_{\mathcal{R}(A)}\mathbf{b}\}.$$

Sia ora $\mathbf{x}^*$ la soluzione LS di minima norma, allora per la Proposizione 1.8, $\mathbf{x}^* \in \mathcal{N}(A)^\perp$. Questo implica che

$$\mathbf{x}^* = (I - \mathcal{P}_{\mathcal{N}(A)})\mathbf{x}^* = A^\dagger A\mathbf{x}^* = A^\dagger \mathcal{P}_{\mathcal{R}(A)}\mathbf{b} = A^\dagger AA^\dagger \mathbf{x} = A^\dagger \mathbf{b},$$

dove abbiamo utilizzato le Proprietà 1.10. □

Proposizione 1.12. La pseudoinversa di uno scalare σ è

$$\sigma^\dagger = \begin{cases} \frac{1}{\sigma} & \text{se } \sigma \neq 0 \\ 0 & \text{se } \sigma = 0 \end{cases}$$

Dimostrazione. Uno scalare σ visto come operatore non è altro che una applicazione da $\mathbb{R}$ in $\mathbb{R}$ che a x associa σx.

- Se $\sigma \neq 0$, allora la funzione è invertibile e pertanto $\bar{\sigma} = \sigma$ e $\sigma^\dagger = \bar{\sigma}^{-1} = \sigma^{-1}$;
- Se invece $\sigma = 0$, allora $\mathcal{N}(\sigma) = \mathbb{R}$ e $\mathcal{N}(\sigma)^\perp = \{0\}$. Pertanto $\bar{\sigma}^{-1} = \bar{\sigma}$ è l'applicazione banale nulla $0 \longmapsto 0$, per definizione allora la pseudoinversa è la sua estensione lineare a $\mathbb{R}$ che ha come spazio nullo $\mathbb{R}$, ossia la funzione identicamente nulla $\sigma^\dagger = 0$.

□

Proposizione 1.13.

$$\begin{bmatrix} A \\ 0 \end{bmatrix}^\dagger = \begin{bmatrix} A^\dagger \\ 0 \end{bmatrix}; \qquad \begin{bmatrix} A & 0 \end{bmatrix}^\dagger = \begin{bmatrix} A^\dagger & 0 \end{bmatrix}.$$

Proposizione 1.14 (Pseudoinversa di una matrice diagonale)**.**
Sia $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_p)$, allora $\Sigma^\dagger = \mathrm{diag}(\sigma_1^\dagger, \ldots, \sigma_p^\dagger)$.

1.3 Singular Value Decomposition

Teorema 1.15 (Decomposizione a Valori Singolari (SVD) [4])**.** Una qualsiasi matrice $A \in \mathbb{R}^{m\times n}$ di rango p si può scomporre nel seguente modo:

$$A = U\Sigma V^\intercal,$$

dove $U \in \mathbb{R}^{m\times m}$ e $V \in \mathbb{R}^{n\times n}$ sono matrici ortogonali e $\Sigma \in \mathbb{R}^{m\times n}$ è una matrice diagonale, cioè $\sigma_{ij} = 0 \quad \forall i \neq j$.

$$U = \begin{bmatrix} \mathbf{u}_1 \, \mathbf{u}_2 \ldots \mathbf{u}_m \end{bmatrix}, \qquad V = \begin{bmatrix} \mathbf{v}_1 \, \mathbf{v}_2 \ldots \mathbf{v}_n \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} \Sigma_p & 0 \\ 0 & 0 \end{bmatrix}, \qquad \Sigma_p = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_p).$$

I vettori $\mathbf{u}_i$ e $\mathbf{v}_i$ sono detti rispettivamente **vettori singolari sinistri** e **destri** di A. I valori σ_i sono detti **valori singolari** di A. Essi vengono disposti in ordine decrescente $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p > 0$.

Proprietà 1.16. La SVD di una matrice in generale non è unica, tuttavia è possibile dimostrare che i valori singolari di una matrice A sono una proprietà intrinseca e sono indipendenti dalle matrici U e V.

Proprietà 1.17. Siano

$$U = \begin{bmatrix} \underset{p}{U_1} & \underset{(m-p)}{U_2} \end{bmatrix}; \qquad V = \begin{bmatrix} \underset{p}{V_1} & \underset{(n-p)}{V_2} \end{bmatrix}.$$

Allora $A = U_1\Sigma_p V_1^\intercal$ è detta **SVD compatta** e le colonne di

- U_1 sono una base per $\mathcal{R}(A)$,
- U_2 sono una base per $\mathcal{R}(A)^\perp$,
- V_1 sono una base per $\mathcal{N}(A)^\perp$,
- V_2 sono una base per $\mathcal{N}(A)$.

1.3.1 SVD e sistemi lineari

Sostituiamo $A = U\Sigma V^\intercal$ nella quantità che vogliamo minimizzare nel problema (1.2) e moltiplichiamo tutto per la matrice ortogonale $U^\intercal$, che sappiamo avere norma-2 unitaria. Sia inoltre $\hat{\mathbf{b}} = U^\intercal \mathbf{b}$,

$$\|A\mathbf{x} - \mathbf{b}\|^2 = \|U\Sigma V^\intercal \mathbf{x} - \mathbf{b}\|^2 = \|\Sigma V^\intercal \mathbf{x} - U^\intercal \mathbf{b}\|^2 = \left\|\Sigma V^\intercal \mathbf{x} - \hat{\mathbf{b}}\right\|^2.$$

Per le Proposizioni 1.12, 1.14, 1.13, $\Sigma^\dagger = \begin{bmatrix} \Sigma_p^{-1} & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{n\times m}$ allora la soluzione dei minimi quadrati sarà

$$\mathbf{x}^\dagger = V\Sigma^\dagger \hat{\mathbf{b}} = V\Sigma^\dagger U^\intercal \mathbf{b} = \sum_{i=1}^{p} \frac{\mathbf{u}_i^\intercal \mathbf{b}}{\sigma_i} \mathbf{v}_i. \tag{1.5}$$

Per la Proprietà 1.10e., $A^\dagger = V\Sigma^\dagger U^\intercal$ è la pseudoinversa di Moore-Penrose della matrice A secondo la Definizione 1.9.

Utilizzando la SVD è facile ridimostrare che la soluzione $\mathbf{x}^\dagger$ è soluzione del problema ai minimi quadrati.

$$\left\|\Sigma V^\intercal \mathbf{x}^\dagger - \hat{\mathbf{b}}\right\|^2 = \left\|\Sigma V^\intercal V\Sigma^\dagger \hat{\mathbf{b}} - \hat{\mathbf{b}}\right\|^2 = \left\|\left(\begin{bmatrix} \Sigma_p & 0 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} \Sigma_p^{-1} & 0 \\ 0 & 0 \end{bmatrix} - I_m\right)\hat{\mathbf{b}}\right\|^2 =$$

$$= \left\|\left(\begin{bmatrix} I_p & 0 \\ 0 & 0 \end{bmatrix} - I_m\right)\hat{\mathbf{b}}\right\|^2 = \left\|\begin{bmatrix} 0 & 0 \\ 0 & -I_{m-p} \end{bmatrix}\begin{bmatrix} \hat{\mathbf{b}}_1 \in \mathbb{R}^p \\ \hat{\mathbf{b}}_2 \in \mathbb{R}^{m-p} \end{bmatrix}\right\|^2 = \left\|\hat{\mathbf{b}}_2\right\|^2.$$

Osserviamo che, per la Proprietà 1.17, $\hat{\mathbf{b}}_2 = \mathcal{P}_{\mathcal{R}(A)^\perp}\mathbf{b}$ e infatti $\hat{\mathbf{b}}_2$ è combinazione lineare di $\mathbf{u}_{p+1}, \ldots, \mathbf{u}_m$. Questo dimostra, usando la Proposizione 1.6, che $\mathbf{x}^\dagger$ è soluzione LS.

1.4 Generalized Singular Value Decomposition

Sia $L \in \mathbb{R}^{t\times n}$ una matrice tale che $\mathcal{N}(A) \cap \mathcal{N}(L) = \{\mathbf{0}\}$, allora è possibile riformulare il problema (1.3) nel seguente modo:

$$\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^n \mid A^\mathsf{T}A\mathbf{x} = A^\mathsf{T}\mathbf{b}\}; \qquad \mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathcal{S}} \|L\mathbf{x}\|. \tag{1.6}$$

Questo problema, per un'opportuna scelta di L, ci permette, tra tutte le soluzioni possibili, di selezionarne una con determinate caratteristiche desiderate al variare di L. Più avanti descriveremo alcune scelte di L che possono avere senso in determinati contesti.

La **Decomposizione a Valori Singolari Generalizzata** (GSVD) [29] della coppia di matrici (A, L) è la fattorizzazione

$$A = U\Sigma_A Z^{-1}, \qquad L = V\Sigma_L Z^{-1},$$

dove $U \in \mathbb{R}^{m\times m}$ e $V \in \mathbb{R}^{t\times t}$ sono matrici ortogonali, $Z \in \mathbb{R}^{n\times n}$ è invertibile e Σ_A e Σ_L hanno rispettivamente le dimensioni di A e L. Sia $d = n - t \geq 0$. Allora

$$\Sigma_L = \begin{bmatrix} I_{n-p} & 0 & 0 \\ 0 & S & 0 \end{bmatrix},$$

$$\underset{m\geq n>p}{\Sigma_A = \begin{bmatrix} 0_{n-p} & 0 & 0 \\ 0 & C & 0 \\ 0 & 0 & I_d \\ 0 & 0 & 0 \end{bmatrix}}, \qquad \underset{n>m>p}{\Sigma_A = \begin{bmatrix} 0 & 0_{m-p} & 0 & 0 \\ 0 & 0 & C & 0 \\ 0 & 0 & 0 & I_d \end{bmatrix}},$$

dove

$$C = \mathrm{diag}(c_1, \ldots, c_{p-d}), \qquad S = \mathrm{diag}(s_1, \ldots, s_{p-d}),$$

con $c_i^2 + s_i^2 = 1$, $i = 1, \ldots, (p-d)$. Gli elementi diagonali sono ordinati in modo che i **valori singolari generalizzati** $\gamma_i = c_i/s_i$ sono non decrescenti per $i = 1, \ldots, (p-d)$, cioè

$$0 < c_1 \leq c_2 \leq \cdots \leq c_{p-d} < 1 \qquad \text{e} \qquad 1 > s_1 \geq s_2 \geq \cdots \geq s_{p-d} > 0.$$

1.4.1 GSVD e sistemi lineari

Analogamente a quanto fatto per la SVD, sfruttiamo la fattorizzazione per trovare la soluzione al problema dei minimi quadrati. Sia $\hat{\mathbf{b}} = U^\intercal \mathbf{b}$,

$$\|A\mathbf{x} - \mathbf{b}\| = \left\|U\Sigma_A Z^{-1}\mathbf{x} - \mathbf{b}\right\| = \left\|\Sigma_A Z^{-1}\mathbf{x} - U^\intercal\mathbf{b}\right\| = \left\|\Sigma_A Z^{-1}\mathbf{x} - \hat{\mathbf{b}}\right\|.$$

Da cui possiamo ricavare la soluzione $\mathbf{x}^* = Z\Sigma_A^\dagger\hat{\mathbf{b}}$, dove $\Sigma_A^\dagger \in \mathbb{R}^{n\times m}$

$$\underset{m\geq n>p}{\Sigma_A^\dagger = \begin{bmatrix} 0_{n-p} & 0 & 0 & 0 \\ 0 & C^{-1} & 0 & 0 \\ 0 & 0 & I_d & 0 \end{bmatrix}}, \qquad \underset{n>m>p}{\Sigma_A^\dagger = \begin{bmatrix} 0 & 0 & 0 \\ 0_{m-p} & 0 & 0 \\ 0 & C^{-1} & 0 \\ 0 & 0 & I_d \end{bmatrix}}.$$

In entrambi i casi

$$\mathbf{x}^* = Z\Sigma_A^\dagger\hat{\mathbf{b}} = \sum_{i=n-p+1}^{t} \frac{\mathbf{u}_i^\intercal\mathbf{b}}{c_{i-n+p}}\,\mathbf{z}_i + \sum_{i=t+1}^{n} (\mathbf{u}_i^\intercal\mathbf{b})\,\mathbf{z}_i, \tag{1.7}$$

dove $\mathbf{z}_i$ sono le colonne di Z.

Proposizione 1.18. La soluzione $\mathbf{x}^*$ dell'equazione (1.7) è la soluzione del problema (1.6) ed è unica.

Lo schema della dimostrazione è presente in [31].

Dimostrazione. Sia $\mathbf{w} = Z^{-1}\mathbf{x} = [w_1, w_2, \ldots, w_n]^\intercal$, allora $\mathbf{x} = Z\mathbf{w}$.

La funzione che dobbiamo minimizzare diventa:

$$\|A\mathbf{x} - \mathbf{b}\| = \left\|U\Sigma_A Z^{-1}\mathbf{x} - \mathbf{b}\right\| = \left\|\Sigma_A\mathbf{w} - \hat{\mathbf{b}}\right\|. \tag{1.8}$$

Per la particolare struttura di Σ_A possiamo scrivere

$$\left(\Sigma_A\mathbf{w}\right)_i = \begin{cases} 0 & i = 1,\ldots,n-p \\ c_{i-n+p}w_i & i = n-p+1,\ldots,t \\ w_i & i = t,\ldots,n \\ \big(0 & i = m-n,\ldots,m\big) \qquad \text{solo nel caso } m>n. \end{cases}$$

Posso minimizzare la norma della (1.8) ponendo

$$w_i = \begin{cases} \dfrac{\hat{b}_i}{c_{i-n+p}} & i = n-p+1, \dots, t \\ \hat{b}_i & i = t, \dots, n \end{cases}. \tag{1.9}$$

Tutti i vettori $\mathbf{w} \in \mathbb{R}^n$ le cui ultime p entrate sono scelte in questo modo sono soluzioni LS. Restano dunque $n-p$ gradi di libertà per minimizzare $\|L\mathbf{x}\|$, che scriviamo utilizzando il fatto che $\mathbf{x} = Z\mathbf{w}$:

$$\|L\mathbf{x}\|^2 = \left\|V\Sigma_L Z^{-1} Z\mathbf{w}\right\|^2 = \|\Sigma_L \mathbf{w}\|^2 = \sum_{i=1}^{n-p} w_i^2 + \sum_{n-p+1}^{t} s_{i-n+p}^2 w_i^2.$$

Gli elementi della seconda sommatoria sono vincolati da (1.9), per minimizzare la norma di $L\mathbf{x}$ si deve imporre $w_i = 0$, $i = 1, \dots, n-p$. Questa scelta corrisponde esattamente a porre $\mathbf{w} = \Sigma_A^\dagger \hat{\mathbf{b}}$ che porta alla soluzione $\mathbf{x}^*$ per definizione di $\mathbf{w}$. □

Si può osservare da questa dimostrazione perché è importante che i nuclei di A e di L siano disgiunti, questo infatti implica che $\mathcal{N}(A)^\perp \cap \mathcal{N}(L)^\perp \neq \{\mathbf{0}\}$. La matrice Z, vista come trasformazione di $\mathbb{R}^n$, permette di separare le componenti di $\mathbf{x}$ in tre parti:

$$\begin{cases} w_1, \dots, w_{n-p} & \to \quad \mathcal{N}(L)^\perp \setminus \mathcal{N}(A)^\perp \text{ influiscono su } L\mathbf{x}, \\ w_{n-p+1}, \dots, w_t & \to \quad \mathcal{N}(A)^\perp \cap \mathcal{N}(L)^\perp \text{ influiscono sia su } A\mathbf{x} \text{ che su } L\mathbf{x}, \\ w_{t+1}, \dots, w_n & \to \quad \mathcal{N}(A)^\perp \setminus \mathcal{N}(L)^\perp \text{ influiscono su } A\mathbf{x}. \end{cases}$$

Se ci fossero vettori in $\mathcal{N}(A) \cap \mathcal{N}(L)$, le loro componenti tramite Z non sarebbero né vincolate da $\|A\mathbf{x} - \mathbf{b}\|$, né influirebbero su $\|L\mathbf{x}\|$, pertanto resterebbero libere e il problema (1.6) diventerebbe mal posto.

1.4.2 Scelta di *L*

Per come è definito il problema, la matrice L determina quale viene scelta tra le soluzioni LS. Costruiamo dunque la matrice L in modo tale che, se $\mathbf{x}$ è una soluzione desiderabile, $\|L\mathbf{x}\|$ è vicina a 0, se, al contrario, $\mathbf{x}$ è indesiderabile, $\|L\mathbf{x}\|$ è grande.

Chiaramente, se si sceglie $L = L_0 = I_n$, allora si ritrova la SVD. Due scelte che solitamente vengono fatte per la L sono

$$L_1 = \underbrace{\begin{bmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{bmatrix}}_{\in \mathbb{R}^{(n-1)\times n}}, \qquad L_2 = \underbrace{\begin{bmatrix} 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \end{bmatrix}}_{\in \mathbb{R}^{(n-2)\times n}}.$$

Esse, applicate a un vettore, sono le discretizzazioni rispettivamente delle derivate prima e seconda. Di conseguenza, se si sceglie L_1 si troveranno delle soluzioni senza brusche variazioni, se si sceglie L_2 si eviteranno le soluzioni che presentano grande curvatura, cioè quelle in cui vi sono improvvisi cambi di direzione crescente/decrescente.

1.5 Estrapolazione

Questa sezione è una breve introduzione ai metodi di estrapolazione classici e alle loro applicazioni, la trattazione è presa da [13, 7]. L'idea dell'estrapolazione nasce dalla teoria dell'approssimazione di funzioni e integrali.

Come si può intuire dal nome, la sua formulazione è legata a quella dell'interpolazione, un procedimento di approssimazione che affonda le sue radici nell'antichità e si è sviluppato nei secoli come si può leggere in [26].

Definizione 1.19. Dati i valori x_i di una funzione $F(t)$ nei punti $t_1, \ldots, t_{n+1} \in \mathbb{R}$,

$$x_i = F(t_i) \qquad i = 1 \ldots n+1$$

diremo che la funzione $P(t)$ **interpola** la funzione $F(t)$ nei punti (o nodi) se

$$P(t_i) = x_i, \qquad i = 1, \ldots, n+1.$$

Spesso le funzioni P che si scelgono come funzioni interpolanti sono polinomi, vantaggiosi per la loro regolarità e facilità di calcolo, inoltre sull'interpolazione polinomiale esistono numerosi risultati notevoli che giustificano questa scelta.

Definizione 1.20. Quando la funzione di interpolazione $P(t)$ della Definizione 1.19 è utilizzata per stimare il valore della funzione $F(t)$ al di fuori dell'intervallo che comprende i t_i, si parla di una operazione di **estrapolazione**.

Sono risolvibili mediante metodi di estrapolazione molti problemi numerici che si possono formulare nel seguente modo.

Sia $F(t)$ una funzione definita in $\mathbb{R}^+$, siamo interessati ad approssimare

$$x = \lim_{t \to 0} F(t). \tag{1.10}$$

I passi di un generico metodo di estrapolazione sono i seguenti:

1. si scelgono dei nodi $t_i > 0$,
2. si calcolano $x_i = F(t_i)$,
3. si trova il polinomio interpolante $P(t)$,
4. si estrapola stimando $x \simeq P(0)$.

1.5.1 Metodo di Richardson

Un metodo classico di estrapolazione è quello di Richardson [34]. Fissati due numeri reali $t_0 > 0$ e $0 < r < 1$ scegliamo i nodi $t_i = r^i t_0$. Osserviamo che

$$t_i \overset{i \to \infty}{\longrightarrow} 0 \quad \Rightarrow \quad F(t_i) \overset{i \to \infty}{\longrightarrow} x.$$

Il metodo di Richardson consiste nel costruire k successioni secondo l'algoritmo seguente

$$\begin{aligned} x_i^{(0)} &= F(r^i t_0), && i = 0, 1 \ldots, n; \\ x_i^{(k+1)} &= \frac{x_i^{(k)} - r^{k+1} x_{i-1}^{(k)}}{1 + r^{k+1}}, \qquad k = 0, 1, \ldots, n-1, && i = k+1, \ldots, n. \end{aligned}$$

Questi passi corrispondono al calcolo di $F(t_i)$ e alle operazioni di interpolazione ed estrapolazione descritte in precedenza. In forma tabellare si ha

$$\begin{array}{ccccccc}
x_0^{(0)} & & & & & & \\
\downarrow & & & & & & \\
x_1^{(0)} & \to & x_1^{(1)} & & & & \\
\downarrow & & \downarrow & & & & \\
x_2^{(0)} & \to & x_2^{(1)} & \to & x_2^{(2)} & & \\
\downarrow & & \downarrow & & \downarrow & & \\
x_3^{(0)} & \to & x_3^{(1)} & \to & x_3^{(2)} & \to & x_3^{(3)} \\
\vdots & & \vdots & & \vdots & & \vdots
\end{array}$$

Proposizione 1.21. Per ogni $k \in \mathbb{N}$ si ha

$$x_i^{(k)} = x + o\Big((r^i t_0)^{k+1}\Big).$$

Si ha, quindi, la convergenza verso x di ciascuna delle colonne della tavola. La convergenza della colonna k-ma è asintoticamente k volte più rapida della prima colonna, perciò considerando i valori dell'ultima colonna possiamo ottenere una stima accurata con un costo computazionale inferiore.

Esempio 1.2 (Romberg - Formula di quadratura dei trapezi)**.** Consideriamo il problema del calcolo numerico di integrali definiti

$$I = \int_a^b f(u)\mathrm{d}u.$$

La formula di quadratura dei trapezi consiste nell'approssimare $f(u)$ con una funzione lineare a tratti che coincide con essa in $n+1$ punti equidistanti $u_i = a + hi$, $h = \frac{b-a}{n}$ come si vede in Figura 1.1.

$$I \simeq \mathcal{T}(h) = h \sum_{i=1}^{n-1} f(u_i) + \frac{h}{2}\Big(f(a) + f(b)\Big). \tag{1.11}$$

Se la funzione f è sufficientemente regolare

$$\lim_{h \to 0} \mathcal{T}(h) = I,$$

questo problema è quindi del tipo (1.10).

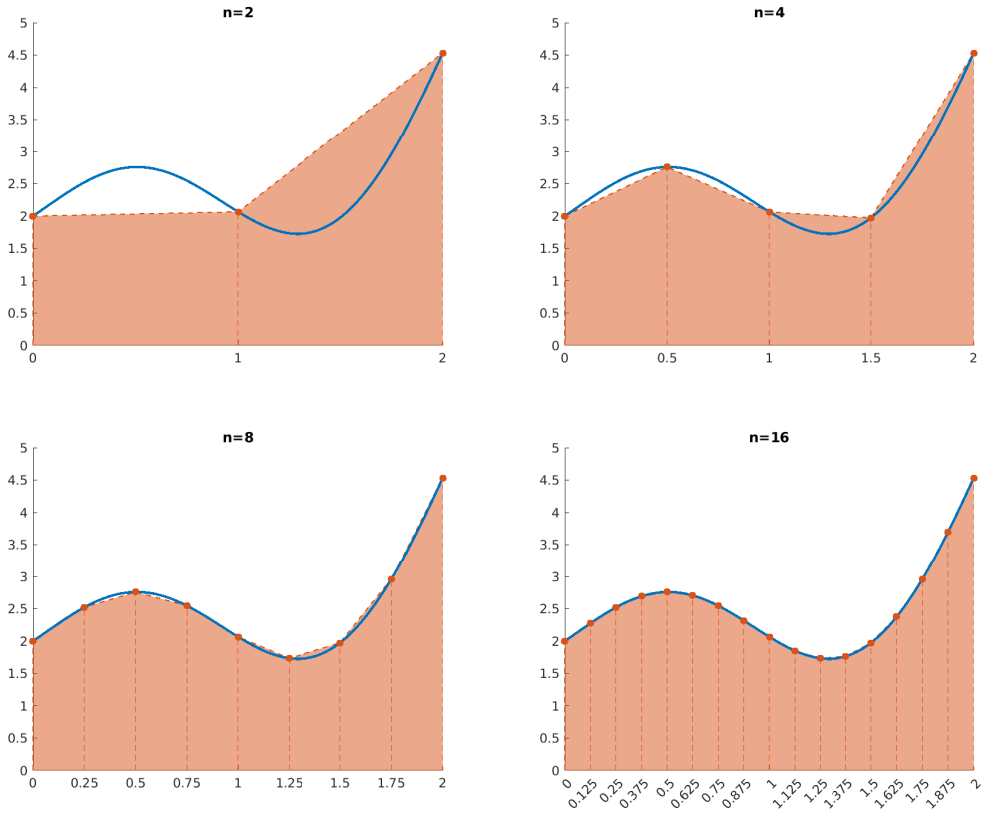


Figura 1.1: Quadratura con i trapezi per numero di nodi crescenti.

Il metodo di Richardson applicato alla successione di integrali approssimati ottenuti con la formula (1.11) prende il nome di metodo di **Romberg** [37]. Si fissa $t = h^2$, $t_0 = (b-a)^2$, $r = 1/4$ che corrisponde a dimezzare h ogni volta, questo è vantaggioso perché in questo modo è possibile, ad ogni passo, valutare $f(u)$ solo nei nuovi nodi, cioè quelli dispari. Il metodo allora diventa

$$x_i^{(0)} = \mathcal{T}\left(\frac{(b-a)^2}{4^i}\right), \qquad i = 0, 1 \ldots, n-1,$$
$$x_i^{(k+1)} = \frac{4^{k+1} x_i^{(k)} - x_{i-1}^{(k)}}{4^{k+1} - 1}, \qquad k = 0, 1, \ldots, n-1, \qquad i = k+1, \ldots, n.$$

Il metodo di Richardson è un particolare esempio di metodo di accelerazione per la convergenza di successioni, di cui ora diamo una definizione più precisa.

Definizione 1.22. Sia $\{x_i\}_{i \in \mathbb{N}}$ una successione di numeri reali che converge al limite x.

Una trasformazione T che mappa la successione $\{x_i\}$ in un'altra $\{y_i\} = T(\{x_i\})$

convergente al limite y si dice **regolare** se $y = x$ e **accelera la convergenza** se

$$\lim_{i\to\infty} \frac{y_i - x}{x_i - x} = 0.$$

In generale una trasformazione T non possiede queste caratteristiche per ogni successione $\{x_i\}$ e, in particolare, è stato dimostrato in [14] che una trasformazione T universale che accelera tutte le successioni non può esistere. Questo implica che sarà sempre interessante ricercare e studiare nuove trasformazioni di successioni, dal momento che ciascuna di esse sarà in grado di accelerare la convergenza solo di una certa classe di successioni.

1.5.2 Il metodo di Aitken

Il metodo Δ^2 di Aitken [1] consiste nella seguente trasformazione di successioni

$$y_i = \frac{x_i x_{i+2} - x_{i+1}^2}{x_{i+2} - 2x_{i+1} + x_i}, \quad i = 0, 1, \ldots$$

Si può facilmente dimostrare che questa trasformazione accelera la convergenza di tutte le sequenze per le quali esiste $\lambda \in [-1, 1)$ tale che

$$\lim_{i\to\infty} \frac{x_{i+1} - x}{x_i - x} = \lambda.$$

Il metodo di Aitkin, al contrario di quello di Richardson, è una trasformazione non lineare. In genere i metodi non lineari accelerano una classe di successioni più ampia, ma non sempre trasformano una successione convergente in un un'altra convergente e anche quando questo accade il limite potrebbe essere differente.

Capitolo 2

Metodi di regolarizzazione

Come descritto in Sezione 1.1, nei problemi che si incontrano nelle applicazioni, i dati a disposizione sono affetti da errore. Se il problema è instabile o mal condizionato, allora una minima variazione nei dati comporta una grande variazione nella soluzione. Pertanto, si rende necessario l'utilizzo di un **metodo di regolarizzazione**, ossia un sistema per ottenere una soluzione al problema inverso che non risenta troppo dell'errore, che sia più vicina possibile a quella cercata.

In Figura 2.1 sono rappresentate la soluzione esatta del problema (1.2) $A^{\dagger}\mathbf{b}$ e la soluzione che si ottiene con dati affetti da errore. In Figura 2.1a le due soluzioni sono raffigurate interamente, così da poter osservare le grandi oscillazioni della soluzione $A^{\dagger}\tilde{\mathbf{b}}$. In Figura 2.1b sono rappresentate le stesse due soluzioni, ma l'asse y è stato limitato per poter osservare la forma della soluzione esatta. Le linee quasi verticali che si vedono in tale grafico sono in gergo definite "spaghetti", sono la prova visiva che la soluzione che abbiamo ottenuto è priva di significato, in quanto la differenza tra le due soluzioni è enorme. Questo giustifica l'esigenza di utilizzare i metodi di regolarizzazione che di seguito presentiamo.

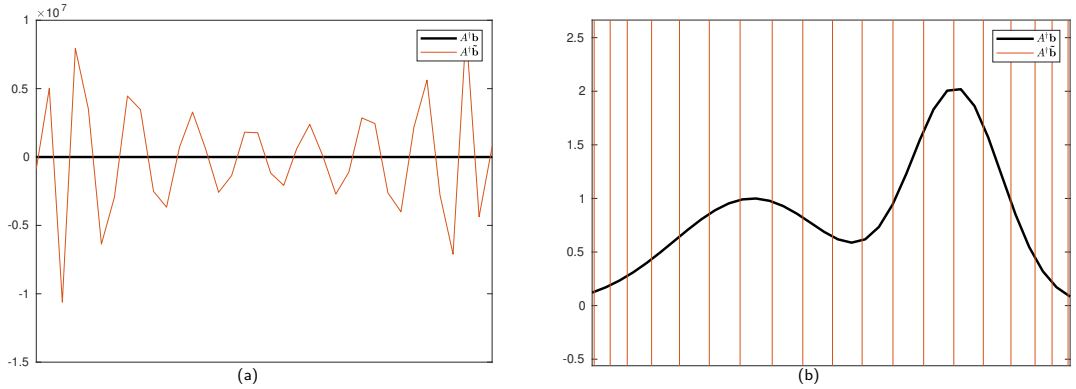


Figura 2.1: Confronto tra soluzione esatta e soluzione inversa per un problema di Shaw con $A \in \mathbb{R}^{36\times 36}$, livello dell'errore $\|\mathbf{b} - \tilde{\mathbf{b}}\| < \delta = 10^{-6}$.

2.1 Regolarizzazione

In generale il processo di regolarizzazione di un problema mal posto può essere definito per gli operatori lineari su spazi di Hilbert (vedi [16]), noi daremo la definizione per spazi a dimensione finita dove possiamo identificare ogni spazio di Hilbert con $\mathbb{R}^n$ e ogni operatore lineare con una matrice.

La regolarizzazione consiste sostanzialmente nell'approssimazione di un problema mal posto per mezzo di una famiglia di problemi ben posti "vicini" ad esso.

Dato il problema

$$A\mathbf{x} = \mathbf{b},$$

con $A\colon \mathbb{R}^n \to \mathbb{R}^m$, supponiamo di conoscere un'approssimazione dei dati $\tilde{\mathbf{b}} = \mathbf{b}+\mathbf{e}$, sia $\delta \in (0, +\infty)$ il **livello dell'errore**, ossia

$$\|\mathbf{b} - \tilde{\mathbf{b}}\| \leq \delta. \tag{2.1}$$

Definizione 2.1 (Metodo di regolarizzazione)**.** Sia $A : \mathbb{R}^n \to \mathbb{R}^m$ un operatore lineare limitato tra spazi di Hilbert, $\alpha_0 \in (0, +\infty)$. Per ogni $\alpha \in (0, \alpha_0)$, sia

$$R_\alpha : \mathbb{R}^m \to \mathbb{R}^n$$

un operatore continuo non necessariamente lineare.

La famiglia $\{R_\alpha\}$ è un **operatore di regolarizzazione** per $A^\dagger$, se per ogni $\mathbf{b} \in \mathbb{R}^m$ esiste una **regola di scelta del parametro** $\alpha = \alpha(\delta, \tilde{\mathbf{b}})$ tale che valga

$$\lim_{\delta\to 0} \sup_{\tilde{\mathbf{b}}\in\mathbb{R}^m} \left\{ \|R_\alpha \tilde{\mathbf{b}} - A^\dagger \mathbf{b}\| \mid \|\mathbf{b} - \tilde{\mathbf{b}}\| \leq \delta \right\} = 0.$$

Inoltre la funzione

$$\alpha\colon \mathbb{R}^+ \times \mathbb{R}^n \to (0, \alpha_0)$$

è tale che

$$\lim_{\delta\to 0} \sup_{\tilde{\mathbf{b}}\in\mathbb{R}^m} \left\{ \alpha(\delta, \tilde{\mathbf{b}}) \mid \|\mathbf{b} - \tilde{\mathbf{b}}\| \leq \delta \right\} = 0.$$

La coppia (R_α, α) è detta **metodo di regolarizzazione** per $A\mathbf{x} = \mathbf{b}$.

Nei metodi trattati in questa discussione il parametro di regolarizzazione varia in

un insieme discreto, tuttavia esistono (e sono molto utilizzati) metodi continui di cui il più famoso è quello di Tikhonov [42].

2.2 Metodi TSVD e TGSVD

Osservando la Proposizione 1.12 si nota che la pseudoinversa $A^\dagger$ non è una funzione continua di A, a meno che non consideriamo perturbazioni di A che non ne modificano il rango. Per via degli errori di arrotondamento, un qualsiasi algoritmo che calcoli la pseudoinversa A otterrà come risultato la pseudoinversa di $A + E$, dove E è una matrice di perturbazione. Dunque, per via della discontinuità, il risultato ottenuto potrà essere anche molto lontano da quello desiderato. Se una matrice A_{esatta} ha rango $\ell < \min\{n, m\}$, molto probabilmente la matrice perturbata avrà invece rango pieno. In [41] viene mostrato che in questo caso i valori singolari della matrice A sono diversi da 0, ma molto piccoli da un certo punto in poi. Questo giustifica l'idea dei metodi di troncamento della SVD e GSVD, che considerano nulli i valori singolari più piccoli di una certa soglia.

Definizione 2.2. Fissata una soglia $\varepsilon > 0$ si dice che una matrice A ha ε-**rango numerico** uguale a k se

$$k = \min\{\operatorname{rank}(B) \mid \|A - B\| \leq \varepsilon\}.$$

In altre parole l'idea di questi metodi è quella di considerare il rango numerico di A al posto di quello classico.

2.2.1 Truncated SVD

Se nella soluzione (1.5) consideriamo l'errore sui dati $\tilde{\mathbf{b}} = \mathbf{b} + \mathbf{e}$ otteniamo [19]

$$\tilde{\mathbf{x}} = \sum_{i=1}^{p} \frac{\mathbf{u}_i^\intercal(\mathbf{b} + \mathbf{e})}{\sigma_i}\mathbf{v}_i = \sum_{i=1}^{p} \frac{\mathbf{u}_i^\intercal \mathbf{b}}{\sigma_i}\mathbf{v}_i + \sum_{i=1}^{p} \frac{\mathbf{u}_i^\intercal \mathbf{e}}{\sigma_i}\mathbf{v}_i = \mathbf{x}^* + \sum_{i=1}^{p} \frac{\mathbf{u}_i^\intercal \mathbf{e}}{\sigma_i}\mathbf{v}_i.$$

L'effetto dell'errore sarà molto grande quando $\frac{\|\mathbf{e}\|}{\sigma_i} \gg 1$, cioè per i crescente. Costruiamo un metodo di regolarizzazione troncando la sommatoria prima di arrivare a p, in modo tale da conservare solo i termini per cui $\frac{\|\mathbf{e}\|}{\sigma_i} \ll 1$.

Sia dunque $1 \leq \ell \leq p$ il parametro di regolarizzazione,

$$\mathbf{x}_\ell = \sum_{i=1}^{\ell} \frac{\mathbf{u}_i^\intercal \mathbf{b}}{\sigma_i} \mathbf{v}_i. \tag{2.2}$$

In forma matriciale possiamo scrivere

$$\mathbf{x}_\ell = A_\ell^\dagger \tilde{\mathbf{b}} = V \Sigma_\ell^\dagger U^\intercal \mathbf{b},$$

dove $\Sigma_\ell^\dagger = \operatorname{diag}(\sigma_1^{-1}, \sigma_2^{-1}, \ldots, \sigma_\ell^{-1}, 0, \ldots, 0) \in \mathbb{R}^{n \times m}$.

La soluzione (2.2) risolve, al posto del problema originale, il problema

$$\min_{\mathbf{x}} \|A_\ell \mathbf{x} - \mathbf{b}\|,$$

dove A_ℓ è la migliore approssimazione di A tra tutte le matrici di rango ℓ, cioè

$$A_\ell := \sum_{i=1}^{\ell} \sigma_i \mathbf{u}_i \mathbf{v}_i^\intercal = \arg \min_{\operatorname{rank} B \leq \ell} \|B - A\|.$$

2.2.2 Truncated GSVD

Ricordiamo la soluzione della GSVD equazione (1.7)

$$\mathbf{x}^* = Z \Sigma_A^\dagger \tilde{\mathbf{b}} = \sum_{i=t-(p-d)+1}^{t} \frac{\mathbf{u}_i^\intercal \mathbf{b}}{c_{i-n+p}} \mathbf{z}_i + \sum_{i=t+1}^{n} (\mathbf{u}_i^\intercal \mathbf{b})\, \mathbf{z}_i,$$

in cui abbiamo sostituito $n = t + d$ nell'indice di partenza della sommatoria. Per ottenere un metodo di regolarizzazione vogliamo una successione di soluzioni $\mathbf{x}_\ell$, $\ell = 1, \ldots, (p-d)$ tale che vengano considerati solo i termini della sommatoria più significativi corrispondenti ai valori di c_i più grandi:

$$\mathbf{x}_\ell = \sum_{i=t-\ell+1}^{t} \frac{\mathbf{u}_i^\intercal \mathbf{b}}{c_{i-n+p}} \mathbf{z}_i + \sum_{i=t+1}^{n} (\mathbf{u}_i^\intercal \mathbf{b})\, \mathbf{z}_i,$$

2.3 Regole di scelta del parametro

Finora abbiamo discusso di diversi operatori di regolarizzazione, ma non ci siamo ancora chiesti come scegliere il parametro ottimale ℓ. Esistono diversi tipi di regole di scelta del parametro.

Definizione 2.3. Sia α una regola di scelta del parametro secondo la Definizione 2.1, allora si dice che α è una regola di scelta del parametro **a priori** se α non dipende da $\tilde{\mathbf{b}}$, ma solo da δ, e si scrive $\alpha = \alpha(\delta)$.

In caso contrario si dice regola di scelta del parametro **a posteriori**.

Si possono considerare anche regole di scelta del parametro che dipendono solo da $\tilde{\mathbf{b}}$ e non da δ, esse sono dette **euristiche**. Un risultato presentato in [3] mostra che queste non possono soddisfare strettamente la Definizione 2.1 se l'operatore $A^\dagger$ è illimitato, tuttavia vi sono esempi in cui questo genere di regole portano a ricostruzioni migliori delle altre. Inoltre, il loro utilizzo è molto vasto poiché nelle applicazioni raramente si dispone di un valore affidabile del livello dell'errore, pertanto, in questi casi, i metodi euristici sono l'unica scelta possibile.

Iniziamo descrivendo alcune regole tra le più usate, per poi descrivere, nel Capitolo 4, una nuova regola di scelta basata sull'estrapolazione vettoriale. Le regole [32] che vogliamo descrivere sono:

- Principio di Discrepanza,
- Curva-L,
- Generalized Cross Validation.

Ogni metodo di regolarizzazione discreto fornisce un insieme di soluzioni $\mathbf{x}_\ell$ e siamo interessati a conoscere quale di queste è la più vicina a $\mathbf{x}^* = A^\dagger\mathbf{b}$. Sappiamo che $A^\dagger\tilde{\mathbf{b}}$ non è quasi mai una buona soluzione, in quanto, salvo rari casi, differisce molto dalla soluzione cercata (vedi Figura 2.1). Per i primi valori di ℓ, $\|\mathbf{x}_\ell - \mathbf{x}^*\|$ diminuisce, mentre da un certo punto in poi inizia a crescere sempre di più.

2.3.1 Principio della discrepanza

Il principio della discrepanza o *Discrepancy Principle* (DP), dovuto a Morozov [28], è una regola di scelta del parametro a posteriori.

Definizione 2.4. Il valore di ℓ_{DP} è

$$\ell_{\text{DP}} = \max_{\ell}\{\ell \mid \|A\mathbf{x}_\ell - \tilde{\mathbf{b}}\| \leq \tau\delta\}.$$

Dove $\tau > 1$ è una costante indipendente da δ, da stabilirsi a priori.

In [16] è fornita una dimostrazione del fatto che per $\delta \to 0$, $\mathbf{x}_{\ell_{\text{DP}}} \to \mathbf{x}^*$. Quando la norma dell'errore è nota, il DP fornisce spesso valori ottimali del parametro (se si sceglie correttamente τ).

2.3.2 Curva-L

La trattazione di questo metodo inizialmente presentato in [20, 22] è presa da [36], articolo in cui è descritto anche un algoritmo di implementazione del metodo. Il metodo consiste nell'analisi della poligonale che unisce i punti del piano

$$(x_\ell, y_\ell) = (\log_{10}\|A\mathbf{x}_\ell - \tilde{\mathbf{b}}\|, \log_{10}\|L\mathbf{x}_\ell\|), \qquad \ell = 1, \ldots, t,$$

dove t è il numero di righe di L.

Questa curva, in molti casi, è a forma di 'L'. Il metodo prescrive di scegliere come parametro di regolarizzazione quello che corrisponde allo spigolo della 'L'.

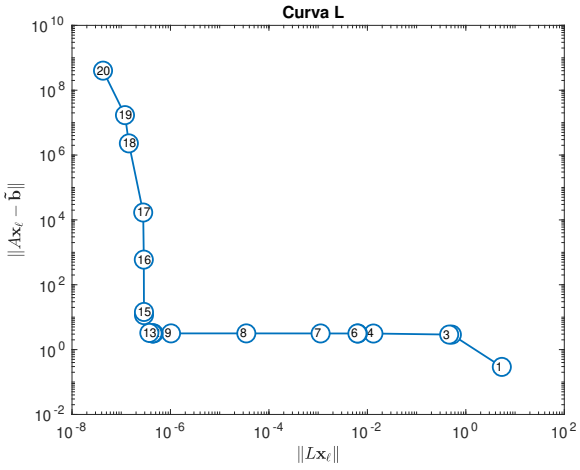


Figura 2.2: Esempio di curva 'L'.

Questa scelta è giustificata dal fatto che, all'aumentare di ℓ, la norma di $\|A\mathbf{x}_\ell - \tilde{\mathbf{b}}\|$ diminuisce e contemporaneamente, per via del mal condizionamento della matrice A, la seminorma $\|L\mathbf{x}_\ell\|$ aumenta improvvisamente nel momento in cui ℓ supera

una determinata soglia che corrisponde all'ε-rango numerico di A, per un determinato ε scelto correttamente. Lo spigolo della curva L corrisponde a questo passaggio e permette un compromesso tra la minimizzazione della norma del residuo e della seminorma-L della soluzione. Questo è molto evidente nella Figura 2.2: nell'esempio mostrato, fino al passo 13 la norma del residuo decresce senza che la seminorma della soluzione aumenti significativamente, poi si nota la crescita improvvisa di $\|L\mathbf{x}_\ell\|$.

2.3.3 Generalized Cross Validation

Questo metodo euristico deriva da considerazioni statistiche. Approssimativamente potremmo dire che si vuole trovare il parametro di regolarizzazione tale che se una delle entrate del vettore $\tilde{\mathbf{b}}$ dovesse mancare, la soluzione regolarizzata deve essere in grado di ritrovarla accuratamente calcolando $A\mathbf{x}_\ell$. Si suppone che l'errore sia una variabile casuale con media 0 e matrice di varianza $\sigma^2 I$ con σ ignoto.

Il metodo GCV prescrive di scegliere come indice ℓ_{GCV} quello che minimizza la funzione

$$G(\ell) = \frac{\|A\mathbf{x}_\ell - \tilde{\mathbf{b}}\|^2}{\mathrm{tr}\{I - AA_\ell^\dagger\}}$$

che nel caso della SVD diventa

$$G(\ell) = \frac{\|A\mathbf{x}_\ell - \tilde{\mathbf{b}}\|^2}{(m-\ell)^2}.$$

Il problema numerico della GCV è che la funzione G risulta essere molto piatta intorno al minimo, perciò la scelta del valore minimo è molto sensibile a piccole variazioni del problema ed errori di calcolo.

Capitolo 3

Tecniche di estrapolazione vettoriale

L'estrapolazione vettoriale è la generalizzazione alle successioni di vettori di $\mathbb{R}^n$ delle tecniche di estrapolazione scalare descritte nella Sezione 1.5. Una panoramica dei metodi presentati in questa tesi è presente in [8]. Gli algoritmi che descriveremo sono tutti metodi per l'accelerazione della convergenza di successioni costruite nel seguente modo.

Supponiamo che, scelto un vettore iniziale $\mathbf{x}_0 \in \mathbb{R}^n$, si generi una successione di vettori secondo la seguente regola:

$$\mathbf{x}_{i+1} = M\mathbf{x}_i + \mathbf{g}, \qquad i = 0, 1, \ldots, \tag{3.1}$$

con $\mathbf{x}_i \in \mathbb{R}^n$, M matrice quadrata e $\mathbf{g}$ un vettore fissati. Indipendentemente dalla convergenza della successione, assumendo che $I - M$ sia invertibile, il limite (o antilimite nel caso la successione non converga [38]) $\mathbf{x}$ della successione è il punto fisso della trasformazione (3.1), cioè

$$\mathbf{x} = M\mathbf{x} + \mathbf{g}. \tag{3.2}$$

Definiamo

$$\begin{aligned} \Delta\mathbf{x} &= \mathbf{x} - \mathbf{x}_0, \\ \Delta\mathbf{x}_i &= \mathbf{x}_{i+1} - \mathbf{x}_i, \\ \Delta^2\mathbf{x}_i &= \Delta\mathbf{x}_{i+1} - \Delta\mathbf{x}_i. \end{aligned}$$

Nelle sezioni che seguono descriveremo diversi metodi utili per trovare un'approssimazione del punto fisso di una successione senza conoscere la matrice M e il vettore $\mathbf{g}$ che l'hanno generata.

3.1 Minimal Polynomial Extrapolation

In questa sezione descriviamo il metodo *Minimal Polynomial Extrapolation* (MPE) [12]. Per ottenere la formula di estrapolazione MPE (3.4) si definiscono i polinomi

$$p(t) = \sum_{j=0}^{s+1} p_j t^j,$$

di grado $s+1$, con $p_{s+1} = 1$, e

$$S_p(t) = \sum_{i=0}^{s} q_i t^i,$$

di grado s, con

$$q_s = p_{s+1}, \qquad q_{i-1} = q_i + p_i, \qquad \text{per } i = s, (s-1), \ldots, 1.$$

È necessario che $p(1) \neq 0$; come vedremo in seguito questo vale se imponiamo che la matrice $I - M$ sia invertibile. Se $p(t)$ è il polinomio monico tale che "annulla" $\Delta\mathbf{x}_0$, cioè

$$p(M)\Delta\mathbf{x}_0 = 0, \tag{3.3}$$

allora la formula di estrapolazione MPE è la seguente:

$$\mathbf{x} \simeq \mathbf{y}^{(s)} := \sum_{j=0}^{s+1} \frac{p_j}{p(1)} \mathbf{x}_j. \tag{3.4}$$

3.1.1 Derivazione del metodo MPE

Mostriamo come si ottiene questa formula. Dalle (3.1), (3.2), si ha

$$\Delta\mathbf{x}_i = M\Delta\mathbf{x}_{i-1} = M^i\Delta\mathbf{x}_0; \tag{3.5}$$

$$\Delta\mathbf{x}_0 = (I - M)\Delta\mathbf{x}. \tag{3.6}$$

Infatti

$$\begin{aligned}\Delta\mathbf{x}_i &= \mathbf{x}_{i+1} - \mathbf{x}_i = \\ &= M\mathbf{x}_i + \mathbf{g} - M\mathbf{x}_{i-1} - \mathbf{g} = M\Delta\mathbf{x}_{i-1};\end{aligned}$$

$$\begin{aligned}(I - M)\Delta\mathbf{x} &= \mathbf{x} - M\mathbf{x} - \mathbf{x}_0 + M\mathbf{x}_0 = \\ &= \mathbf{g} - \mathbf{x}_0 + \mathbf{x}_1 - \mathbf{g} = \Delta\mathbf{x}_0.\end{aligned}$$

Lemma 3.1.

$$(I - M)S_p(M) = p(I) - p(M).$$

Dimostrazione. Per definizione, i q_i valgono

$$q_i = \sum_{j=i+1}^{s+1} p_j, \qquad \text{e in particolare } q_0 = p(1) - p_0.$$

Allora

$$\begin{aligned}(I - M)S_p(M) &= \sum_{i=0}^{s} q_i M^i - \sum_{i=0}^{s} q_i M^{i+1} = \\ &= q_0 I + \sum_{i=1}^{s}(q_i - q_{i-1})M^i - q_s M^{s+1} = \\ &= p(I) - p_0 I - \sum_{i=1}^{s} p_i M^i - p_{s+1}M^{s+1} = p(I) - p(M).\end{aligned}$$

□

Imponendo la (3.3) si ottiene

$$p(I)\Delta\mathbf{x}_0 = (I - M)S_p(M)\Delta\mathbf{x}_0.$$

Da questa equazione si vede che se $p(1) = 0$, allora $(I - M)S_p(M)\Delta\mathbf{x}_0 = p(I) = p(1)I = 0$, quindi M avrebbe un autovalore uguale a 1 e $(I-M)^{-1}$ non esisterebbe. Usando la (3.5) e la (3.6),

$$p(1)\Delta\mathbf{x} = S_p(M)\Delta\mathbf{x}_0 = \sum_{i=0}^{s} q_i M^i \Delta\mathbf{x}_0 = \sum_{i=0}^{s} q_i \Delta\mathbf{x}_i.$$

Infine

$$\begin{aligned}
\mathbf{x} = \mathbf{x}_0 + \Delta\mathbf{x} &= \mathbf{x}_0 + \tfrac{1}{p(1)} \sum_{i=0}^{s} q_i \Delta\mathbf{x}_i = \\
&= \mathbf{x}_0 + \tfrac{1}{p(1)} \Big[\sum_{i=0}^{s} q_i \mathbf{x}_{i+1} - \sum_{i=0}^{s} q_i \mathbf{x}_i \Big] = \\
&= \mathbf{x}_0 + \tfrac{1}{p(1)} \Big[q_s \mathbf{x}_{s+1} + \sum_{i=1}^{s} (q_{i-1} - q_i)\mathbf{x}_i - q_0 \mathbf{x}_0 \Big] = \\
&= \mathbf{x}_0 + \tfrac{1}{p(1)} \Big[p_{s+1} \mathbf{x}_{s+1} + \sum_{i=1}^{s} p_i \mathbf{x}_i + p_0 \mathbf{x}_0 - p(1)\mathbf{x}_0 \Big] = \sum_{i=1}^{s+1} \frac{p_i}{p(1)} \mathbf{x}_i.
\end{aligned}$$

La formula di estrapolazione (3.4) è esatta solo se M ha rango s, altrimenti fornisce solo un'approssimazione del limite della successione di vettori che chiamiamo $\mathbf{y}^{(s)}$. Nel caso in cui M ha un piccolo numero di autovalori dominanti si può dimostrare [12] che l'errore di approssimazione è piccolo. Un'altra possibilità di applicazione della formula è quella di estrapolare per valori crescenti di s allo scopo di accelerare la convergenza della successione. Chiameremo $\mathbf{y}_r^{(s)}$ la soluzione del metodo MPE applicata alla sequenza $\{\mathbf{x}_r, \mathbf{x}_{r+1}, \ldots, \mathbf{x}_{r+s+2}\}$.

3.2 Reduced Rank Extrapolation

Descriviamo ora un altro metodo per l'estrapolazione vettoriale: il metodo *Reduced Rank Extrapolation* (RRE). [15, 27]. Si tratta di un'estensione al caso vettoriale della formula Δ^2 di Aitken (Sottosezione 1.5.2). Definiamo le seguenti matrici:

$$\begin{aligned}
K_{r,s} &= [\Delta\mathbf{x}_r \ \ldots \ \Delta\mathbf{x}_{r+s}] \in \mathbb{R}^{n\times(s+1)}, \\
H_{r,s} &= [\Delta^2\mathbf{x}_r \ \ldots \ \Delta^2\mathbf{x}_{r+s}] \in \mathbb{R}^{n\times(s+1)}.
\end{aligned}$$

Il metodo RRE può essere definito in diversi modi (vedi [24, 40]), noi scriveremo il vettore estrapolato come:

$$\mathbf{y}_r^{(s)} = \mathbf{x}_r + \sum_{i=0}^{s} \xi_i \Delta\mathbf{x}_{r+i}, \tag{3.7}$$

i cui coefficienti ξ_i si ottengono risolvendo il problema ai minimi quadrati

$$\min_{\boldsymbol{\xi}\in\mathbb{R}^{s+1}} \|H_{r,s}\boldsymbol{\xi} + \Delta\mathbf{x}_r\|, \tag{3.8}$$

dove le incognite sono $\boldsymbol{\xi} = (\xi_0, \ldots, \xi_s)^\intercal$.

3.2.1 Derivazione del metodo RRE

Le differenze $\Delta^2\mathbf{x}_i$ per la (3.5) soddisfano la seguente relazione:

$$\Delta^2\mathbf{x}_i = (A - I)\Delta\mathbf{x}_i.$$

Per brevità indichiamo con $K_i = K_{0,i}$, e $H_i = H_{0,i}$, le quali soddisfano le seguenti relazioni:

$$\begin{aligned} K_{1,i} &= MK_i, \\ H_i &= K_{1,i} - K_i = (M - I)K_i. \end{aligned} \tag{3.9}$$

Dall'equazione (3.6) possiamo ricavare

$$\mathbf{x} = \mathbf{x}_0 + (I - M)^{-1}\Delta\mathbf{x}_0. \tag{3.10}$$

A questo punto, se scegliamo $i = n - 1$, le matrici H_i e K_i sono quadrate e, se H_i è non singolare, si può ottenere $(I - M)^{-1}$ dalla (3.9)

$$(I - M)^{-1} = -K_{n-1}H_{n-1}^{-1},$$

e la (3.10) diventa

$$\mathbf{x} = \mathbf{x}_0 - K_{n-1}H_{n-1}^{-1}\Delta\mathbf{x}_0,$$

o, analogamente

$$\begin{cases} \mathbf{x} = \mathbf{x}_0 + K_{n-1}\boldsymbol{\xi}, & \boldsymbol{\xi} \in \mathbb{R}^n \\ \mathbf{0} = \Delta\mathbf{x}_0 + H_{n-1}\boldsymbol{\xi} \end{cases}.$$

Queste equazioni rappresentano l'**estrapolazione *full rank***. Teoricamente esse conducono al risultato esatto $\mathbf{x}$, tuttavia non hanno un'utilità pratica per la risoluzione di sistemi lineari, dal momento che richiedono la soluzione di un sistema

dello stesso ordine dell'originale. Sono comunque interessanti poiché forniscono uno schema per formulare l'estrapolazione a rango ridotto (RRE).

Se ci troviamo nella situazione in cui solo i primi $s+1 < n$ vettori $\mathbf{x}_i$ sono (numericamente) linearmente indipendenti, allora non abbiamo sufficienti informazioni per ricavare esattamente $(I-M)^{-1}$ dalla (3.9), dal momento che H_{n-1} e K_{n-1} hanno solo $s+1$ colonne linearmente indipendenti. Il massimo che si può fare è approssimare $(I-M)^{-1}$ con una matrice di rango s. Quindi supponiamo che $(I-M)^{-1}\Delta\mathbf{x}_0$ si possa esprimere come una combinazione lineare delle colonne di K_s,

$$(I-M)^{-1}\Delta\mathbf{x}_0 = K_s\boldsymbol{\xi}, \qquad \boldsymbol{\xi} \in \mathbb{R}^{s+1}.$$

Allora per la (3.9) vale

$$\Delta\mathbf{x}_0 = (I-M)K_s\boldsymbol{\xi} = -H_s\boldsymbol{\xi}.$$

Il metodo che si ottiene è il seguente:

$$\begin{cases} \mathbf{y}^{(s)} = \mathbf{x}_0 + K_s\boldsymbol{\xi}, & \boldsymbol{\xi} \in \mathbb{R}^{s+1} \\ \mathbf{0} = \Delta\mathbf{x}_0 + H_s\boldsymbol{\xi} \end{cases}.$$

Se osserviamo la seconda equazione notiamo che è un sistema lineare sottodeterminato con matrice $H_s \in \mathbb{R}^{n\times(s+1)}$, che possiamo risolvere con i minimi quadrati:

$$\min_{\boldsymbol{\xi}\in\mathbb{R}^{s+1}} \|H_s\boldsymbol{\xi} + \Delta\mathbf{x}_0\|.$$

A questo punto abbiamo ottenuto il metodo RRE per una sequenza di vettori che parte da $\mathbf{x}_0$; se consideriamo come vettore di partenza della successione il vettore $\mathbf{x}_r$ otteniamo esattamente le formule (3.7) e (3.8).

3.3 Vector Epsilon Algorithm

L'algoritmo-ε vettoriale (VEA) è stato il primo algoritmo di estrapolazione vettoriale ad essere ideato, ad opera di P. Wynn [43]. Una descrizione più recente dell'algoritmo è presente in [17]. La sua formulazione è esattamente la generalizzazione al caso vettoriale dell'algoritmo-ε di Shanks [38].

Inizializzazione

$$\boldsymbol{\varepsilon}_{-1}^{(i)} = \mathbf{0}, \quad \boldsymbol{\varepsilon}_{0}^{(i)} = \mathbf{x}_i, \qquad i = 0, 1, \ldots$$

Iterazione

$$\boldsymbol{\varepsilon}_{k+1}^{(i)} = \boldsymbol{\varepsilon}_{k-1}^{(i+1)} + [\boldsymbol{\varepsilon}_{k}^{(i+1)} - \boldsymbol{\varepsilon}_{k}^{(i)}]^{-1}, \qquad i, k = 0, 1, \ldots,$$

dove si è posta come generalizzazione dell'operazione di inversione al caso vettoriale:

$$\mathbf{x}^{-1} := \mathbf{x}^{\dagger} = \frac{\mathbf{x}}{\mathbf{x}^{\intercal}\mathbf{x}}.$$

Come per il metodo di Richardson descritto in Sottosezione 1.5.1, le soluzioni $\boldsymbol{\varepsilon}_k^{(i)}$ possono essere disposte in una tabella

$$\begin{array}{cccccc}
 & \mathbf{x}_0 & & & & \\
0 & & \boldsymbol{\varepsilon}_1^{(0)} & & & \\
 & \mathbf{x}_1 & & \boldsymbol{\varepsilon}_2^{(0)} & & \\
0 & & \boldsymbol{\varepsilon}_1^{(1)} & & \boldsymbol{\varepsilon}_3^{(0)} & \\
 & \mathbf{x}_2 & & \boldsymbol{\varepsilon}_2^{(1)} & & \boldsymbol{\varepsilon}_4^{(0)} \\
0 & & \boldsymbol{\varepsilon}_1^{(2)} & & \boldsymbol{\varepsilon}_3^{(1)} & \\
 & \mathbf{x}_3 & & \boldsymbol{\varepsilon}_2^{(2)} & & \\
0 & & \vdots & & \ddots & \\
 & \vdots & & \vdots & &
\end{array}$$

Se si applica l'algoritmo a una successione del tipo (3.1), esistono numerosi risultati di convergenza, descritti in [6]. Sotto determinate condizioni sugli autovalori e autovettori della matrice M vale il seguente risultato

$$\lim_{i \to \infty} \boldsymbol{\varepsilon}_{2k}^{(i)} = \mathbf{x}.$$

In [11] è stato dimostrato che l'algoritmo ε accelera una classe di successioni molto più ampia rispetto ad altri metodi come RRE e MPE.

Per uniformità di notazione chiameremo $\mathbf{y}_r^{(s)} := \boldsymbol{\varepsilon}_{s+2}^{(r)}$ con $s \geq 0$ pari, cioè i vettori che estrapolano $\{\mathbf{x}_r, \ldots, \mathbf{x}_{r+s+2}\}$.

Capitolo 4

Tecniche di estrapolazione per la scelta del parametro

L'idea di utilizzare metodi di estrapolazione scalare per la risoluzione di problemi lineari mal posti è stata presentata inizialmente in [9, 10, 33]. In questi articoli si utilizza un metodo di estrapolazione per trovare una stima dell'errore compiuto dal metodo di regolarizzazione in ogni passo e si sceglie come parametro ottimale quello che minimizza la funzione stima trovata. Dalle sperimentazioni numeriche è emerso che spesso, anche quando la stima non è vicina all'errore reale, essa ne segue l'andamento, pertanto il suo minimo coincide con quello cercato.

L'idea di utilizzare un metodo di estrapolazione vettoriale è stata presentata per la prima volta poco dopo in [5].

Nella regolarizzazione T(G)SVD, descritta nella Sezione 2.2, si costruisce una sequenza di soluzioni $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_\ell$. Questa sequenza, all'aumentare di ℓ, tende a $A^\dagger \tilde{\mathbf{b}}$, che è fortemente contaminata da errore, noi, al contrario, vorremmo ottenere $\mathbf{x}^* = A^\dagger \mathbf{b}$; per questo motivo l'utilizzo dei metodi di accelerazione della convergenza all'intera successione è sconsigliato.

Nei problemi discreti mal posti succede che la sequenza di soluzioni regolarizzate segue un andamento semiconvergente. Inizialmente sono aggiunte alla soluzione le basse frequenze che sono associate ai valori singolari più grandi. In questa fase, la soluzione regolarizzata si avvicina a $\mathbf{x}^*$ se questo è un vettore "regolare". In seguito, quando sono aggiunte le frequenze più alte che corrispondono ai valori singolari più piccoli, la soluzione viene contaminata irrimediabilmente dagli errori.

Il primo processo è generalmente lento e mostra un andamento convergente, mentre il secondo produce un cambiamento improvviso nell'elemento successivo della sequenza.

Inoltre, poiché le matrici che generano un problema lineare mal posto sono caratterizzate nella maggior parte dei casi da un piccolo numero di valori singolari dominanti, la soluzione $\mathbf{x}^*$ può essere approssimata con un'accuratezza ragionevole da un vettore appartenente a uno spazio di dimensione piccola rispetto alla dimensione del problema n. Per questo motivo può essere pensata come il limite di una sequenza del tipo (3.1), per una matrice M di rango piccolo.

4.1 L'algoritmo RESC

Il metodo che è stato ideato sceglie il parametro di regolarizzazione facendo un confronto tra le soluzioni regolarizzate e quelle estrapolate usando un piccolo sottoinsieme di esse pertanto è stato denominato ***Regularized and Extrapolated Solution Comparison*** *(RESC)*. L'idea che ha portato all'algoritmo RESC è di prevedere il limite $\mathbf{x} := \mathbf{t}_\ell$ della sequenza $\mathbf{x}_0, \mathbf{x}_1, \ldots \mathbf{x}_{\ell-1}$, per poi confrontarlo con la successiva soluzione regolarizzata $\mathbf{x}_\ell$. Finché i due vettori sono vicini accettiamo la soluzione $\mathbf{x}_\ell$; nel momento in cui iniziano a essere molto distanti significa che la soluzione ha cominciato ad essere contaminata eccessivamente dagli errori e pertanto fermiamo il processo. Il risultato che si ottiene è una regola di scelta del parametro che può essere utilizzata per qualunque metodo discreto, come per esempio il metodo iterativo LSQR [30].

In [5] è presentato un algoritmo che applica questa idea a successioni di vettori soluzioni della TSVD o LSQR e utilizza come metodo di estrapolazione RRE con $r = 0$ fissato e s crescente; gli algoritmi sono chiamati rispettivamente RRE-TSVD e RRE-LSQR.

In questo modo si calcolano i vettori estrapolati $\mathbf{t}_\ell := \mathbf{y}_0^{(\ell)}$ usando le soluzioni $\{\mathbf{x}_i\}_{0\le i\le \ell+2}$ e si sceglie come parametro di regolarizzazione il primo indice k per cui vale

$$\frac{\|\mathbf{t}_k - \mathbf{t}_{k-1}\|}{\|\mathbf{t}_{k-1}\|} < tol,$$

dove $tol > 0$ è una soglia fissata a priori.

Il nuovo algoritmo consiste in una variante di questo approccio in cui si calcolano le soluzioni estrapolate $\mathbf{y}_r^{(s)}$ dapprima con $r = 0$ e facendo crescere s fino a un certo valore fissato $\bar{s}$, successivamente incrementando r e tenendo fissato $\bar{s}$. Chiamiamo $\mathbf{t}_\ell$ la soluzione che estrapola $\{\mathbf{x}_r, \ldots, \mathbf{x}_{\ell-1}\}$, cioè

$$\mathbf{t}_\ell = \begin{cases} \mathbf{y}_0^{(\ell-3)} & \ell = 3, \dots, \bar{s}+1 \\ \mathbf{y}_{\ell-\bar{s}-3}^{(\bar{s})} & \ell = \bar{s}+2, \dots, n-1 \end{cases}$$

Per la scelta del parametro ottimale si cerca il primo valore che minimizza lo scostamento relativo tra le due soluzioni $\mathbf{t}_\ell$ e $\mathbf{x}_\ell$, cioè

$$S_\ell = \frac{\|\mathbf{t}_\ell - \mathbf{x}_\ell\|_\infty}{\|\mathbf{t}_\ell + \mathbf{x}_\ell\|_\infty} \qquad \ell = 3, \dots, n-1. \tag{4.1}$$

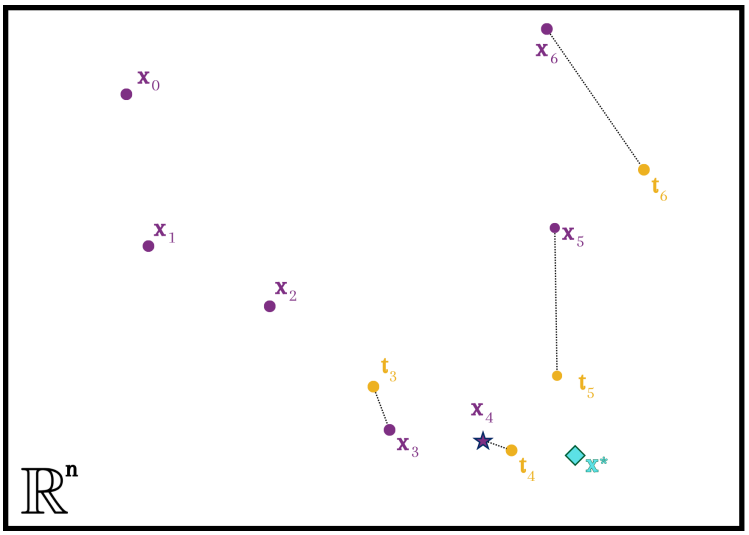


Figura 4.1: Rappresentazione grafica dell'algoritmo.

Uno schema di come viene scelta la soluzione ottimale è presente in Figura 4.1; in questo esempio la soluzione ottimale $\mathbf{x}_4$, indicata con $\star$, è la più vicina alla corrispondente $\mathbf{t}_4$.

Con *primo* valore che minimizza S_ℓ si intende il primo k per cui

$$S_k \leq S_i \qquad i = 3, 4, \dots k + \bar{s} + 2,$$

ottenuto mediante una ricerca del minimo che si arresta quando il valore minimo non viene più aggiornato per $\bar{s}+2$ iterazioni successive.

Un'ulteriore innovazione dell'algoritmo è stata la sperimentazione di altre tecniche di estrapolazione vettoriale oltre a RRE: MPE e VEA, descritte nel Capitolo 3.

Dall'osservazione dei risultati numerici è stata successivamente formulata una variante di questo metodo che si spinge oltre la scelta del parametro.

Questa variante, che chiameremo $\mathbf{t}_k$-RESC, consiste nello scegliere come soluzione regolarizzata la $\mathbf{t}_k$ ottenuta nel calcolo. Tale scelta è giustificata dal fatto che, se k è il parametro ottimale, le soluzioni $\mathbf{x}_\ell$ fino a k stanno convergendo alla soluzione esatta; pertanto, la soluzione $\mathbf{t}_k$ ottenuta accelerando la convergenza di questa successione è più vicina al limite $\mathbf{x}^*$. Inoltre, quando il parametro k non è stato scelto correttamente, ma "ci si è andati vicino", la soluzione estrapolata potrebbe comunque fornire una buona approssimazione della soluzione perché sfrutta le proprietà di semiconvergenza della sequenza $\mathbf{x}_\ell$.

Nota 1. Con questo metodo il parametro di regolarizzazione scelto k è sempre maggiore o uguale a 3. Ciò significa che se il valore ottimo della soluzione è 0, 1 o 2, non sarà *mai* possibile trovarlo. Questo è uno dei principali difetti di questo algoritmo e ne limita l'utilizzo a problemi in cui non c'è un grande divario tra i primissimi valori singolari e i successivi.

Capitolo 5

Risultati numerici

La sperimentazione numerica si è basata su un grande numero di sistemi lineari quadrati. Per ciascun esempio sono stati scelti una matrice test $A \in \mathbb{R}^{n \times n}$ mal condizionata, e un vettore soluzione esatta $x* \in \mathbb{R}^n$. Si è generato il termine noto esatto $b = A\mathbf{x}$ a cui è stato aggiunto un errore casuale $\mathbf{e}$ per ottenere $\tilde{\mathbf{b}} = \mathbf{b} + \mathbf{e}$. L'errore è stato scalato in modo tale che valesse l'equazione (2.1) per un valore di δ fissato. Sono stati ottenuti in questo modo 23040 esempi combinando le seguenti variabili.

- 8 matrici test mal condizionate: vedi Tabella 5.1 e Figura 5.1;
- 8 soluzioni esatte: vedi Figura 5.2;
- 2 dimensioni: 36×36 e 100×100;
- 3 matrici di regolarizzazione $L = I$, L_1, L_2;
- 6 livelli di errore: $\delta = 10^{-1}, 10^{-2}, \ldots, 10^{-6}$;
- 10 realizzazioni casuali dell'errore.

	$\kappa(A)$	
	$n = 36$	$n = 100$
Hilbert[1]	2.93×10^{18}	3.45×10^{19}
Heat[2]	4.07×10^{23}	2.52×10^{37}
Shaw[2]	1.14×10^{18}	6.83×10^{18}
Lotkin[3]	1.08×10^{19}	3.18×10^{19}
Baart[2]	2.84×10^{17}	1.52×10^{18}
Phillips[2]	4.35×10^{04}	2.64×10^{06}
Foxgood[2]	3.26×10^{18}	7.63×10^{18}
Gravity[2]	7.28×10^{10}	1.99×10^{18}

Tabella 5.1: Matrici test: [1] funzione *built-in* di Matlab, [2] dal *toolbox* [21], [3] dal pacchetto *gallery* [23].

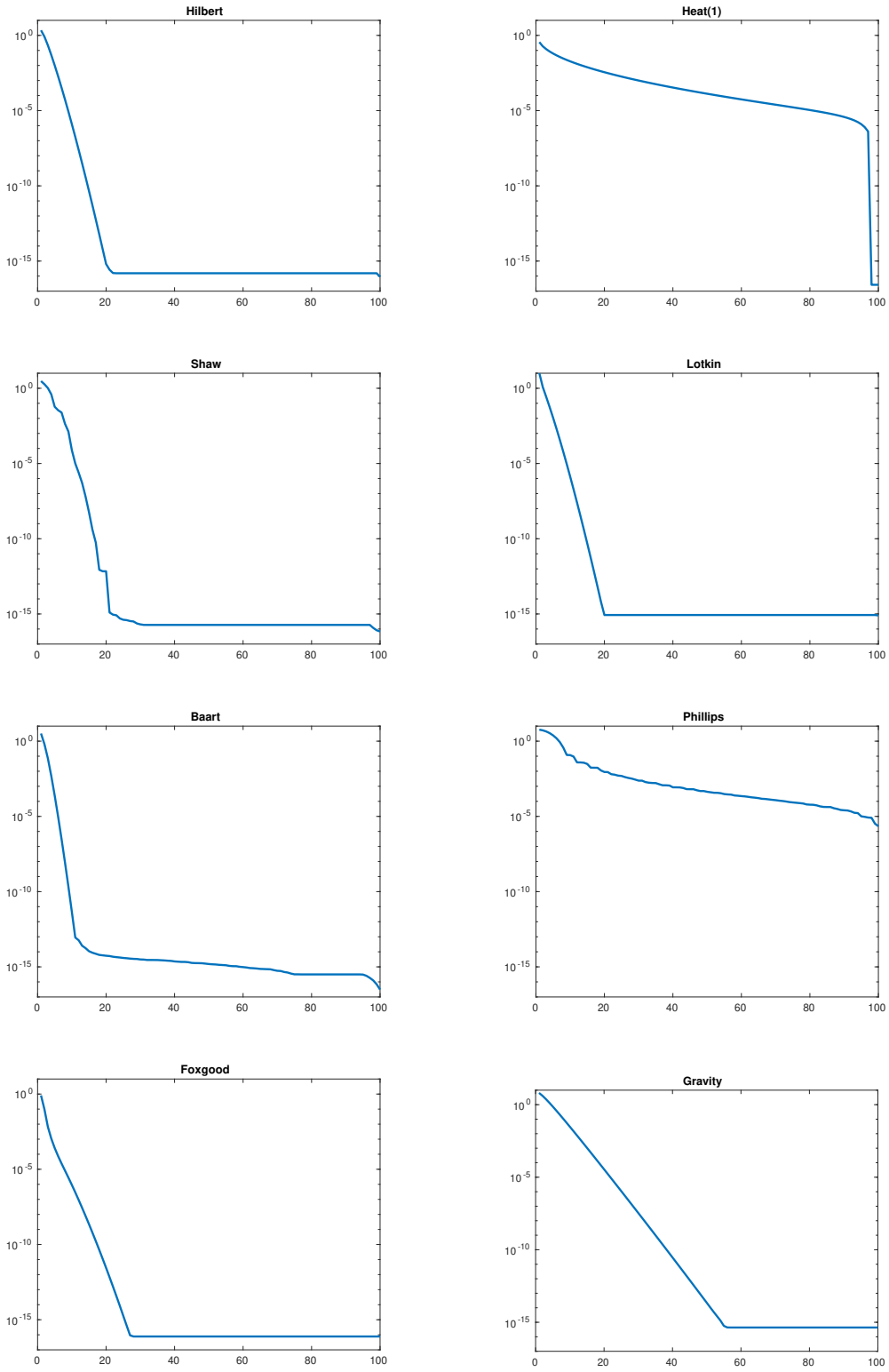
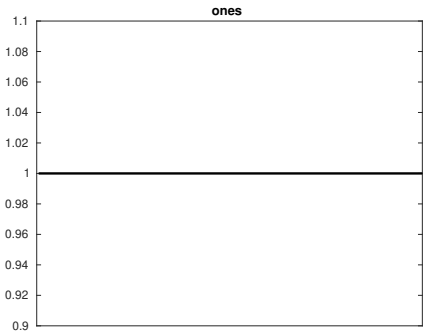
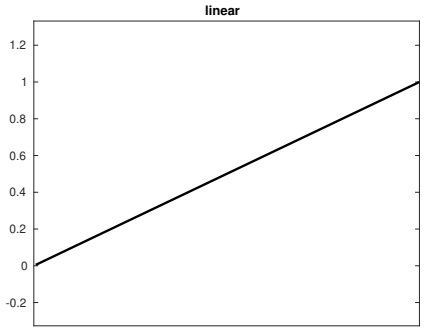


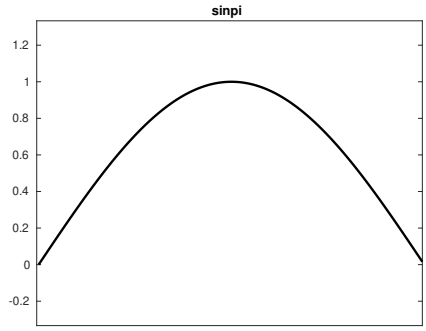
Figura 5.1: Valori singolari delle matrici test 100×100.



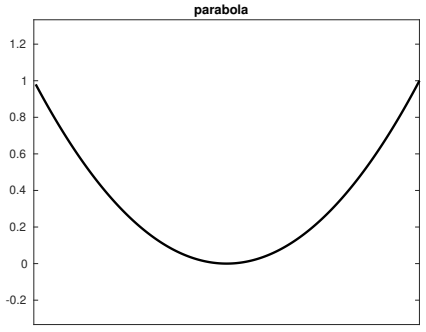
(a) $y = 1$



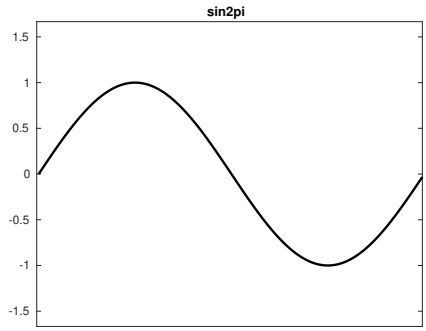
(b) $y = x$, $x \in [0, 1]$



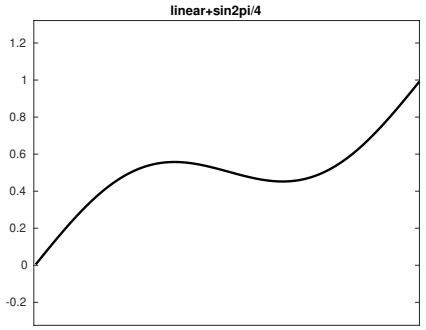
(c) $y = \sin(\pi x)$, $x \in [0, 1]$



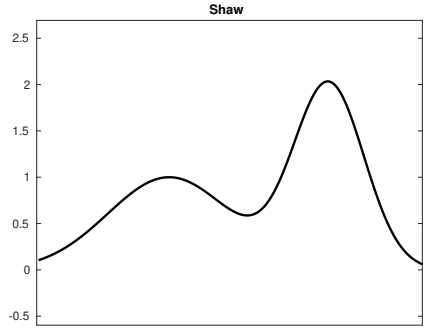
(d) $y = x^2$, $x \in [-1, 1]$



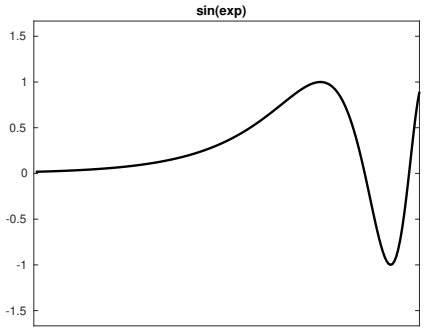
(e) $y = \sin(2\pi x)$, $x \in [0, 1]$



(f) $y = x + \frac{\sin(2\pi x)}{4}$, $x \in [0, 1]$



(g) $y = 2e^{-6(x-0.8)^2} + e^{-2(x+0.5)^2}$, $x \in [-\frac{\pi}{2}, \frac{\pi}{2}]$



(h) $y = \sin(e^x)$, $x \in [-4, 2]$

Figura 5.2: Soluzioni test.

5.1 Confronto delle varianti del metodo

Per ogni esempio è stata calcolata la successione di soluzioni regolarizzate con la T(G)SVD a cui sono stati applicati i diversi metodi di scelta del parametro ed estrapolazione. Per confrontare i diversi metodi è stato considerato l'indice di qualità

$$Q(\mathbf{s}) = \frac{\|\mathbf{x} - \mathbf{s}\|}{\|\mathbf{x} - \mathbf{x}_{k_{\text{opt}}}\|}.$$

Al numeratore c'è la norma dell'errore commesso dalla soluzione stimata dal metodo: $\mathbf{s} = \mathbf{x}_k$ o $\mathbf{s} = \mathbf{t}_k$. Al denominatore c'è la norma dell'errore della soluzione ottima della T(G)SVD, che corrisponde a

$$k_{\text{opt}} = \arg\min_{\ell} \|\mathbf{x} - \mathbf{x}_\ell\|_\infty.$$

Considereremo un test

- un *fallimento* se $Q(\mathbf{s}) > 10$;
- un *grave fallimento* se $Q(\mathbf{s}) > 10^2$.

Varrà dunque sempre $Q \geq 1$ se $\mathbf{s} = \mathbf{x}_\ell$, mentre potrebbe accadere $Q < 1$ se $\mathbf{s} = \mathbf{t}_\ell$; in questo caso l'estrapolazione porterebbe ad ottenere una soluzione più vicina a quella esatta di tutte le soluzioni $\mathbf{x}_\ell$ utilizzate per generarla. Un possibile sviluppo della ricerca potrebbe essere quello di eseguire un controllo sui dati per determinare se questo sia effettivamente accaduto nel corso di questa sperimentazione. Ad ogni modo si considera un successo per l'algoritmo se la norma dell'errore compiuto dalla soluzione trovata non è più di dieci volte superiore a quella di $\mathbf{x}_{k_{\text{opt}}}$.

Nella Tabella 5.2 si possono vedere i risultati ottenuti con il metodo RESC, sia dalle soluzioni regolarizzate della T(G)SVD $\mathbf{x}_k$, sia da quelle estrapolate $\mathbf{t}_k$. Le varianti del metodo considerano i tre algoritmi di estrapolazione MPE, RRE, VEA e 6 possibili valori di $\bar{s}$. Si è scelto di non aumentare ulteriormente il valore di $\bar{s}$ poiché ciò avrebbe portato all'aumento della complessità computazionale a fronte di un esigua differenza nei risultati. Infatti, nei problemi scelti, specialmente nel caso $n = 36$, la soluzione scelta k raramente va oltre 12, il ché rende una scelta di $\bar{s} > 10$ ininfluente per l'algoritmo.

Si possono fare le seguenti considerazioni:

- Il metodo di estrapolazione più efficace è VEA, a seguire RRE. MPE ha fornito i risultati peggiori;
- All'aumentare di $\bar{s}$ si ottengono risultati migliori, tuttavia con miglioramenti sempre più leggeri.
- Le soluzioni estrapolate $\mathbf{t}_k$ danno risultati migliori delle $\mathbf{x}_k$.

	RESC-MPE				RESC-RRE				RESC-VEA			
	$\mathbf{x}_k$		$\mathbf{t}_k$		$\mathbf{x}_k$		$\mathbf{t}_k$		$\mathbf{x}_k$		$\mathbf{t}_k$	
	>10	$>10^2$	>10	$>10^2$	>10	$>10^2$	>10	$>10^2$	>10	$>10^2$	>10	$>10^2$
$\bar{s}=0$	31%	10%	**31%**	**7%**	27%	9%	**21%**	**4%**	27%	9%	**22%**	**4%**
$\bar{s}=2$	25%	9%	**20%**	**3%**	21%	9%	**10%**	**2%**	21%	9%	**10%**	**2%**
$\bar{s}=4$	22%	9%	**15%**	**2%**	18%	8%	**7%**	**1%**	19%	8%	**7%**	**1%**
$\bar{s}=6$	19%	9%	**11%**	**2%**	18%	8%	**7%**	**1%**	17%	8%	**5%**	**1%**
$\bar{s}=8$	18%	9%	**9%**	**2%**	18%	8%	**7%**	**1%**	17%	8%	**5%**	**0%**
$\bar{s}=10$	17%	9%	**8%**	**3%**	17%	8%	**6%**	**1%**	16%	8%	**4%**	**0%**

Tabella 5.2: Percentuale di fallimenti nelle diverse varianti del metodo RESC.

5.2 Confronto col metodo RRE-TSVD

La versione migliore del metodo RESC-VEA$_{10}$, che sta per il metodo di estrapolazione VEA e $\bar{s} = 10$, è stata confrontata con RRE-TSVD [5]. In Tabella 5.3 è possibile osservare i risultati ottenuti: per questo confronto si è utilizzato solo $L = I$ perché l'algoritmo non prevede l'utilizzo della TGSVD. Sono presenti dunque solo 7680 esempi.

La tabella è composta da tre parti: nella prima i dati sono suddivisi per matrici, nella seconda sono suddivisi per livello di errore e nella terza per soluzione esatta utilizzata.

Si può affermare, osservando i dati ottenuti in questo esperimento, che L'algoritmo RRE-TSVD come metodo di scelta del parametro posto nella prima colonna non è efficace, lo è invece se consideriamo le soluzioni estrapolate $\mathbf{t}_k$ (seconda colonna).

MATRIX Su questi problemi test l'algoritmo RESC come metodo di scelta del parametro produce risultati analoghi a $\mathbf{t}_k$-RRE-TSVD mentre l'algoritmo $\mathbf{t}_k$-RESC ottiene risultati sensibilmente migliori negli esempi *Heat*, *Lotkin*, *Phillips*, *Foxgood*, *Gravity* e peggiori in *Shaw*. Dai dati si evince che la matrice *Shaw* è quella che dà maggiori difficoltà all'algoritmo RESC avendo generato la quasi totalità dei fallimenti in questo esperimento; ulteriori esperimenti serviranno a capire per quale motivo questo avvenga.

NOISE LEVEL Osservando la seconda parte della tabella si vede che il metodo funziona meglio per livelli di errore più alti. Sottolineiamo che questo non significa che le soluzioni ottenute con *noise* più alto sono più vicine alla soluzione esatta, ma che l'errore compiuto da RESC è quasi sempre dello stesso ordine di grandezza di quello compiuto da $\mathbf{x}_{k_{\text{opt}}}$.

SOLUTION Per quanto riguarda le diverse soluzioni si vede che la soluzione *ones* è quella più problematica per entrambi gli algoritmi ma RESC in questo caso ottiene circa un quarto dei fallimenti rispetto all'altro metodo. Tuttavia si vede che gli errori sono quasi omogeneamente distribuiti su tutte le soluzioni, fatta eccezione per *Shaw* e *sin(exp)*, per le quali l'algoritmo seleziona quasi sempre una soluzione che commette un errore vicino a quello ottimale.

MATRIX A	$\mathbf{x}_k$ RRE-TSVD >10	$>10^2$	$\mathbf{t}_k$ RRE-TSVD >10	$>10^2$	$\mathbf{x}_k$ RESC-VEA$_{10}$ >10	$>10^2$	$\mathbf{t}_k$ RESC-VEA$_{10}$ >10	$>10^2$
Hilbert	956	954	4	0	8	0	5	0
Heat(1)	416	250	10	1	0	0	0	0
Shaw	953	907	207	39	217	8	277	22
Lotkin	956	949	50	0	79	19	10	0
Baart	948	934	9	0	100	4	6	0
Phillips	833	635	149	42	0	0	0	0
Foxgood	956	953	17	0	136	41	8	1
Gravity	946	897	87	27	10	0	8	0
TOTAL	6964	6479	533	109	550	72	314	23
(7680)	91%	84%	7%	1%	7%	1%	4%	0%

NOISE LEVEL δ	$\mathbf{x}_k$ RRE-TSVD >10	$>10^2$	$\mathbf{t}_k$ RRE-TSVD >10	$>10^2$	$\mathbf{x}_k$ RESC-VEA$_{10}$ >10	$>10^2$	$\mathbf{t}_k$ RESC-VEA$_{10}$ >10	$>10^2$
10^{-6}	1034	978	138	60	87	8	110	18
10^{-5}	1103	1010	138	45	79	0	103	4
10^{-4}	1155	1006	120	1	74	0	80	0
10^{-3}	1192	1101	74	0	33	0	13	0
10^{-2}	1209	1180	24	0	55	8	1	0
10^{-1}	1271	1204	39	3	222	56	7	1
TOTAL	6964	6479	533	109	550	72	314	23

SOLUTION $\mathbf{x}$	$\mathbf{x}_k$ RRE-TSVD >10	$>10^2$	$\mathbf{t}_k$ RRE-TSVD >10	$>10^2$	$\mathbf{x}_k$ RESC-VEA$_{10}$ >10	$>10^2$	$\mathbf{t}_k$ RESC-VEA$_{10}$ >10	$>10^2$
ones	878	728	257	106	184	33	65	21
linear	863	816	24	1	68	16	46	1
sinpi	890	851	45	0	66	5	38	0
parabola	851	815	57	0	90	4	64	1
sin2pi	869	818	54	0	44	1	45	0
lin+sin2pi/4	866	822	59	1	59	7	46	0
Shaw	891	836	15	1	27	3	3	0
sin(exp)	856	793	22	0	12	3	7	0
TOTAL	6964	6479	533	109	550	72	314	23

Tabella 5.3: Confronto tra RRE-TSVD e RESC-VEA$_{10}$.

5.3 Confronto con GCV e Curva-L

Passiamo adesso al confronto dell'algoritmo RESC in tutti i 23040 esempi con due classici metodi euristici di scelta del parametro presentati in Sezione 2.3: GCV e Curva-L, implementati da Hansen nelle funzioni *gcv* e *corner* di [21]. I risultati di questo esperimento sono presentati in Tabella 5.4. In questa tabella, oltre alle prime tre parti, è presente anche una quarta che suddivide i risultati a seconda della matrice di regolarizzazione utilizzata.

Si può osservare dai risultati generali che il metodo che RESC ottiene molti meno fallimenti rispetto ai metodi precedenti, sia nella versione $\mathbf{x}_k$, che nella versione $\mathbf{t}_k$ su tutti gli esempi.

MATRIX Per quanto riguarda le diverse matrici di test, si osserva che in diversi esempi l'utilizzo della GSVD ha portato a compiere più errori, per esempio le matrici *Phillips* e *Foxgood* hanno generato più di 100 fallimenti in più e 20 fallimenti gravi in più rispetto ai risultati ottenuti solo con la TSVD.

NOISE LEVEL Anche in questo caso il numero di fallimenti di $\mathbf{t}_k$-RESC decresce all'aumentare del livello dell'errore, questa tendenza come già accadeva nella Tabella 5.3 sembra essere invertita per il metodo $\mathbf{x}_k$-RESC.

SOLUTION Le soluzioni più problematiche per il metodo sono *ones*, *linear* e *sin(exp)*. Le prime due sono difficili da trovare per ogni metodo che utilizzi la GSVD poiché *ones* appartiene a $\mathcal{N}(L_1)$ e $\mathcal{N}(L_2)$, e *linear* appartiene a $\mathcal{N}(L_2)$. Questo significa che il metodo di regolarizzazione, che cerca un compromesso tra la minimizzazione del residuo e quella della seminorma della soluzione, sbaglia perché in questo caso la soluzione ottima è quella che minimizza totalmente la seminorma, azzerandola. Per quanto riguarda *sin(exp)*, essa ha in alcuni punti curvature elevate, per questo l'utilizzo della GSVD con la matrice L_2 compie più errori.

REGULARIZATION MATRIX A conseguenza di quanto già osservato in precedenza, la matrice che totalizza più fallimenti è la L_2, mentre la matrice L_1 è quella che riesce a ottenere nel complesso meno errori approssimando in generale meglio le soluzioni testate (diverse da *ones*).

MATRIX A	$\mathbf{x}_k$ GCV >10	$>10^2$	$\mathbf{t}_k$ Curva-L >10	$>10^2$	$\mathbf{x}_k$ RESC-VEA$_{10}$ >10	$>10^2$	$\mathbf{t}_k$ RESC-VEA$_{10}$ >10	$>10^2$
Hilbert	1245	1174	491	438	484	312	54	3
Heat(1)	956	812	638	465	202	6	88	0
Shaw	1515	1425	502	375	567	54	355	26
Lotkin	1047	963	724	669	655	400	63	10
Baart	1138	1039	545	466	780	504	93	14
Phillips	1410	1080	856	324	110	21	103	21
Foxgood	1230	1151	811	694	802	517	174	20
Gravity	1256	1188	809	414	127	6	55	0
TOTAL	9797	8832	5376	3845	3727	1820	985	94
(23040)	43%	38%	23%	17%	16%	8%	4%	0%

NOISE LEVEL δ	$\mathbf{x}_k$ GCV >10	$>10^2$	$\mathbf{t}_k$ Curva-L >10	$>10^2$	$\mathbf{x}_k$ RESC-VEA$_{10}$ >10	$>10^2$	$\mathbf{t}_k$ RESC-VEA$_{10}$ >10	$>10^2$
10^{-6}	1583	1424	960	586	483	237	237	36
10^{-5}	1594	1429	846	533	454	214	212	7
10^{-4}	1619	1446	773	457	475	214	205	3
10^{-3}	1664	1512	659	451	471	218	100	3
10^{-2}	1681	1518	725	592	665	347	89	16
10^{-1}	1656	1503	1413	1226	1179	590	142	29
TOTAL	9797	8832	5376	3845	3727	1820	985	94

SOLUTION $\mathbf{x}$	$\mathbf{x}_k$ GCV >10	$>10^2$	$\mathbf{t}_k$ Curva-L >10	$>10^2$	$\mathbf{x}_k$ RESC-VEA$_{10}$ >10	$>10^2$	$\mathbf{t}_k$ RESC-VEA$_{10}$ >10	$>10^2$
ones	1240	1172	2192	1805	1505	871	313	45
linear	1262	1154	1174	910	784	480	174	21
sinpi	1268	1108	436	249	291	101	60	2
parabola	1252	1102	280	147	265	92	80	5
sin2pi	1226	1096	243	84	151	36	70	2
lin+sin2pi/4	1219	1098	390	232	253	95	63	2
Shaw	1194	1084	423	262	213	74	57	1
sin(exp)	1136	1018	238	156	265	71	168	16
TOTAL	9797	8832	5376	3845	3727	1820	985	94

REG. MATRIX L	$\mathbf{x}_k$ GCV >10	$>10^2$	$\mathbf{t}_k$ Curva-L >10	$>10^2$	$\mathbf{x}_k$ RESC-VEA$_{10}$ >10	$>10^2$	$\mathbf{t}_k$ RESC-VEA$_{10}$ >10	$>10^2$
I	2087	1796	1320	328	550	72	314	23
L_1	3207	2936	1375	1171	945	412	203	9
L_2	4503	4100	2681	2346	2232	1336	468	62
TOTAL	9797	8832	5376	3845	3727	1820	985	94

Tabella 5.4: Confronto tra i metodi preesistenti e RESC-VEA$_{10}$.

5.4 Esempi

Esempio 5.1. In questo esempio è stato applicato il metodo RESC-VEA$_{10}$ a un problema con i seguenti parametri:

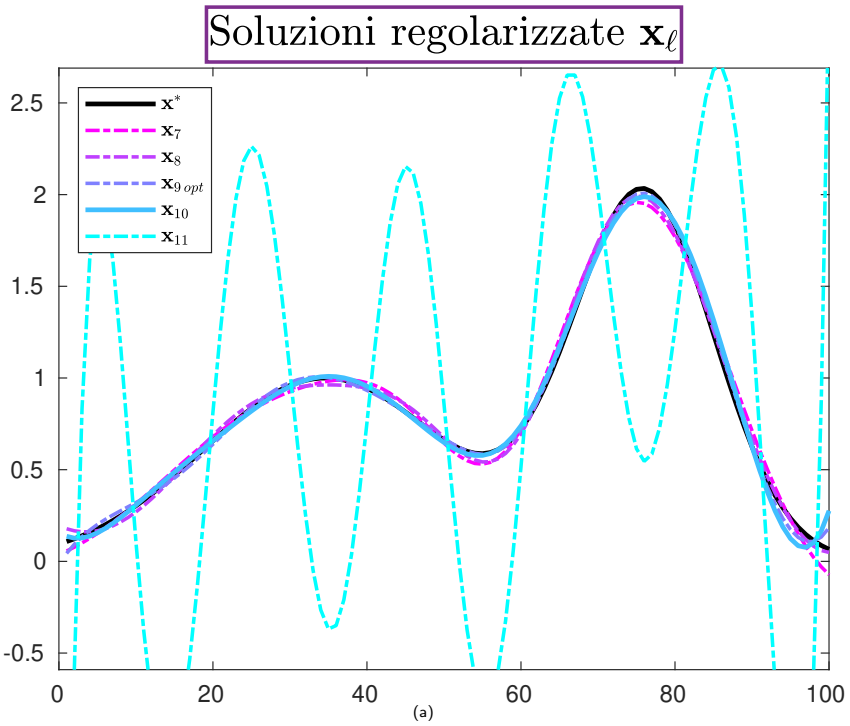
n	A	δ	$\mathbf{x}^*$	L
100	*Shaw*	10^{-5}	*Shaw*	I

In Figura 5.3 sono confrontate le soluzioni $\mathbf{x}_\ell$ e $\mathbf{t}_\ell$, $\ell = 7, \ldots, 11$. La soluzione ottima corrisponde a $k_{\text{opt}} = 9$, mentre l'algoritmo sceglie $k = 10$; infatti come si può vedere nella Figura 5.3a, dopo una sequenza di soluzioni simili tra loro, l'undicesima soluzione presenta quel salto improvviso che segna il punto in cui l'algoritmo di scelta del parametro si ferma.

La Figura 5.4 mostra in alto il grafico dello scostamento relativo S_ℓ definito nell'equazione (4.1). Esso viene utilizzato per trovare il punto di minimo k, in basso sono riportati gli errori relativi rispetto alla soluzione esatta delle due sequenze di soluzioni, mentre sono evidenziate rispettivamente in verde e rosso le zone che corrispondono a un fallimento e a un grave fallimento.

$$\text{err}(\mathbf{s}) = \frac{\|\mathbf{x} - \mathbf{s}\|}{\|\mathbf{x}\|}.$$

Possiamo osservare che, nonostante in questo caso il k selezionato non sia quello ottimale, sia la soluzione $\mathbf{x}_k$, che $\mathbf{t}_k$, restano all'interno della fascia bianca che corrisponde alla zona che abbiamo fissato come successo negli esperimenti precedenti. Notiamo, inoltre, che, mentre all'aumentare di ℓ la soluzione $\mathbf{x}_\ell$ si allontana molto da quella esatta, la $\mathbf{t}_\ell$ resta accettabile a lungo. Questo in parte spiega i migliori risultati ottenuti dalle soluzioni estrapolate, rispetto a quelle iniziali.



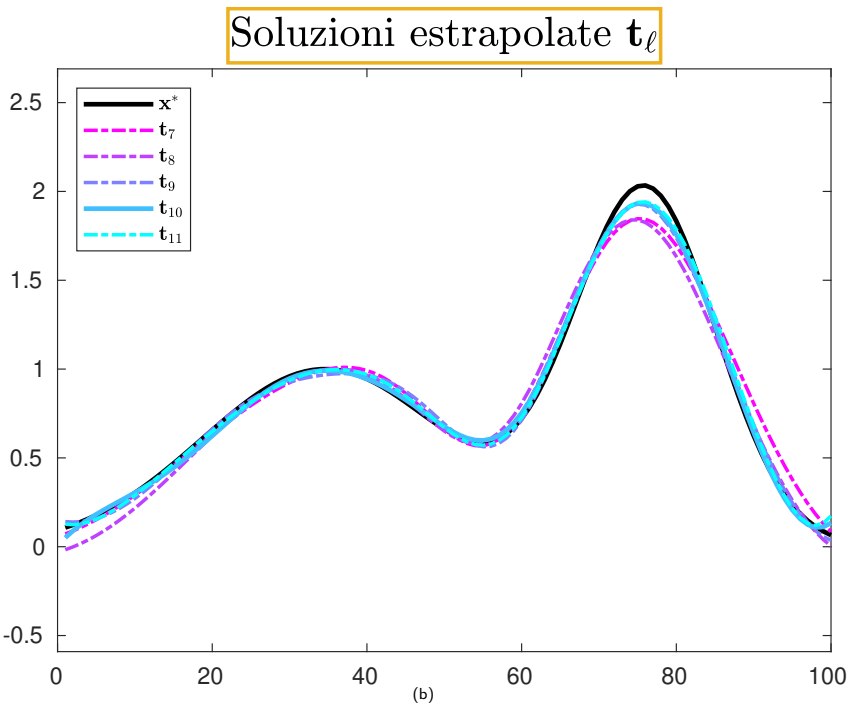


Figura 5.3: Soluzioni dell'Esempio 5.1.

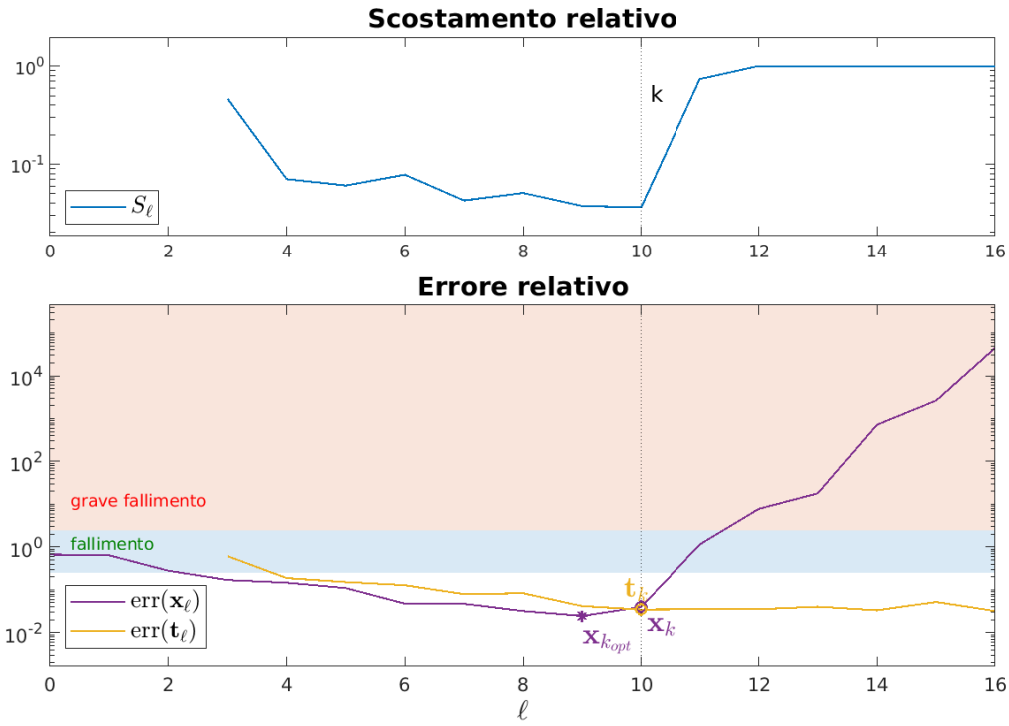


Figura 5.4: Risultati dell'Esempio 5.1.

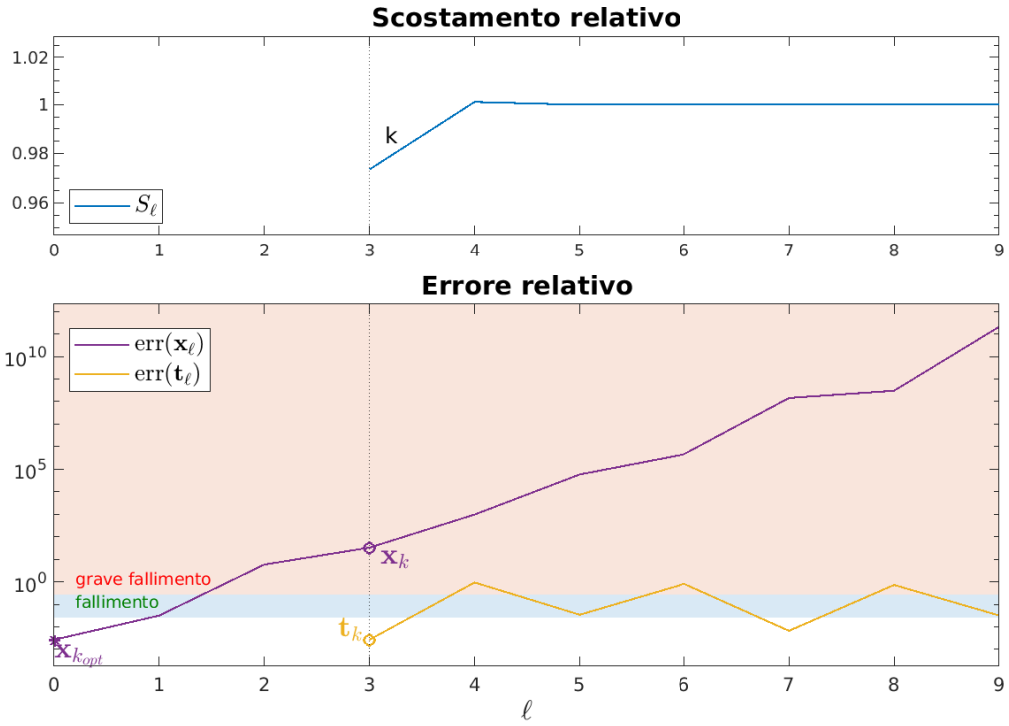


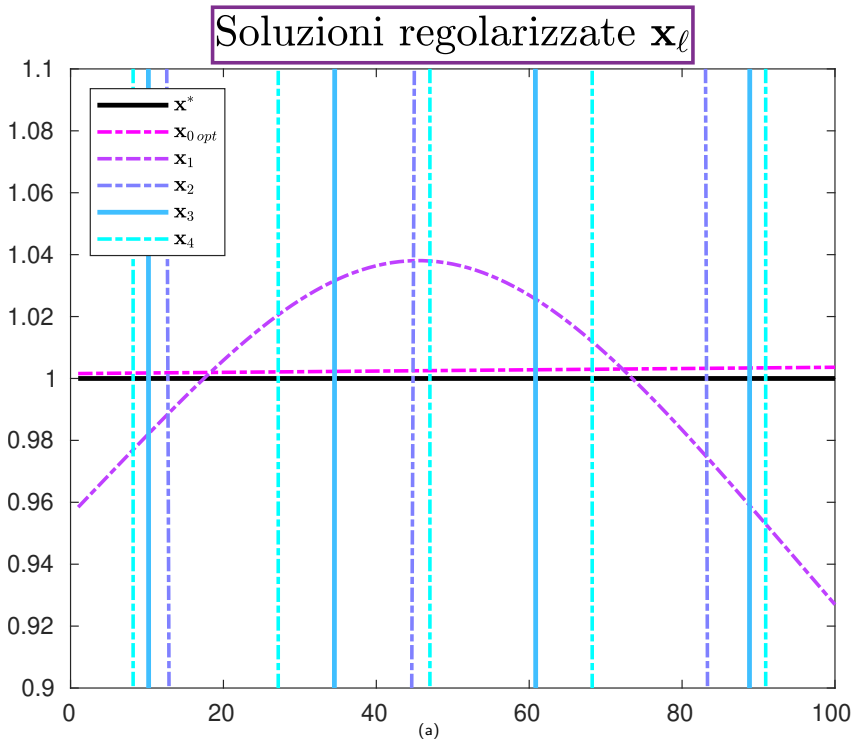
Figura 5.5: Risultati dell'Esempio 5.2.

Esempio 5.2. In questo esempio è stato applicato il metodo RESC-VEA$_{10}$ a un problema con i seguenti parametri:

n	A	δ	$\mathbf{x}^*$	L
100	*Baart*	10^{-1}	*ones*	L_1

Osserviamo dalla Figura 5.5 che in questo caso la soluzione ottima è $\mathbf{x}_0$. Questa eventualità, come già osservato nella Nota 1, impedisce al metodo di selezionarla perché la prima possibilità che abbiamo è $k = 3$, che è effettivamente quella scelta dall'algoritmo.

L'aspetto notevole di questo esempio è che, nonostante la scelta di $\mathbf{x}_k$ porti a un grave fallimento, quella di $\mathbf{t}_k$ permette di ottenere una soluzione di pari livello di quella ottima. Come è possibile osservare anche a occhio nudo dalla Figura 5.6b la soluzione che troviamo è un'approssimazione ottima di quella cercata.



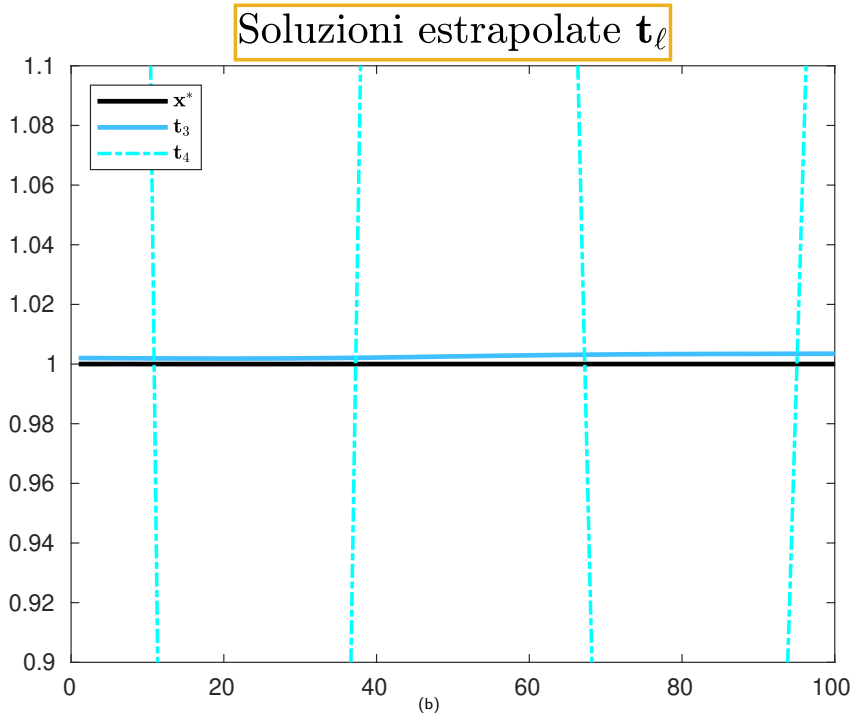


Figura 5.6: Soluzioni dell'Esempio 5.2.

Conclusioni

Dai risultati numerici osservati nel Capitolo 5, è evidente che il metodo RESC sia efficace; esso ottiene, grazie all'estrapolazione vettoriale, risultati migliori dei metodi di regolarizzazione standard usati finora e in molti esempi si osservano risultati migliori del metodo RRE-TSVD che lo ha ispirato. L'analisi dei fallimenti che abbiamo eseguito garantisce che l'errore che si ottiene è confrontabile con quello della soluzione ottima del metodo T(G)SVD nella quasi totalità dei casi.

Lo studio proseguirà verificando cosa succede per valori di $\bar{s}$ maggiori di 10 nel caso in cui la soluzione ottima non sia tra le prime, per verificare come l'aumento di questo parametro influisca sui risultati in presenza di problemi differenti. In futuro si prevede di osservare il caso in cui la soluzione estrapolata risulti essere migliore della soluzione ottima della T(G)SVD, ciò significherebbe un successo particolare per l'algoritmo RESC. Inoltre si cercherà di capire quale sia la motivazione per cui l'algoritmo trova maggiori difficoltà nel problema di Shaw. Ulteriori sviluppi potrebbero portare ad applicare il metodo a sistemi di dimensione grande utilizzando algoritmi come LSQR per generare la successione di vettori $\mathbf{x}_\ell$ e a problemi inversi nonlineari, risolti con metodi iterativi tipo-Newton.

Ringraziamenti

Voglio ringraziare particolarmente professor Rodriguez per la sua rara infinita disponibilità, ma soprattutto per avermi trasmesso la sua passione per questo lavoro. Grazie a lui ho scoperto la bellezza che c'è dietro allo studio di calcoli e numeri e ho provato la soddisfazione di vederli messi a servizio di problemi reali.

L'elenco delle persone che dovrei ringraziare ora sarebbe molto lungo, se sono riuscito a portare a termine questo percorso è in gran parte grazie a tutte le persone che mi vogliono bene e che mi hanno supportato. Non voglio, tuttavia, spendere parole per nessuno ma spero vivamente di riuscire ad esprimere la mia gratitudine negli anni che verranno, ricambiando con tutto l'affetto che mi è stato donato.

Bibliografia

[1] A. C. Aitken, *XXV.-On Bernoulli's numerical solution of algebraic equations*, Proceedings of the Royal Society of Edinburgh, 46 (1926), pp. 289–305.

[2] R. C. Allen, W. R. Boland, V. Faber, and G. M. Wing, *Singular values and condition numbers of galerkin matrices arising from linear integral equations of the first kind*, Journal of Mathematical Analysis and Applications, 109 (1985), pp. 564–590.

[3] A. Bakushinskii, *Remarks on choosing a regularization parameter using the quasi-optimality and ratio criterion*, USSR Computational Mathematics and Mathematical Physics, 24 (1984), pp. 181–182.

[4] A. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, PA, 1996.

[5] A. Bouhamidi, K. Jbilou, L. Reichel, H. Sadok, and Z. Wang, *Vector extrapolation applied to truncated singular value decomposition and truncated iteration*, J. Eng. Math., 93 (2015), pp. 99–112.

[6] C. Brezinski, *Some results in the theory of the vector ε-algorithm*, Linear Algebra and its Applications, 8 (1974), pp. 77–86.

[7] C. Brezinski and M. Redivo Zaglia, *Extrapolation Methods: Theory and Practice*, ISSN, Elsevier Science, 1991.

[8] C. Brezinski, M. Redivo Zaglia, and Y. Saad, *Shanks sequence transformations and anderson acceleration*, SIAM Review, 60 (2018), pp. 646–669.

[9] C. Brezinski, G. Rodriguez, and S. Seatzu, *Error estimates for linear systems with applications to regularization*, Numerical Algorithms, 49 (2008), pp. 85–104.

[10] ——, *Error estimates for the regularization of least squares problems*, Numerical Algorithms, 51 (2009), pp. 61–76.

[11] C. Brezinski, M. R. Zaglia, and A. Salam, *On the kernel of vector ϵ-algorithm and related topics*, Numer. Algorithms, 92 (2023), pp. 207–221.

[12] S. Cabay and L. W. Jackson, *A polynomial extrapolation method for finding limits and antilimits of vector sequences*, SIAM J. Numer. Anal., 13 (1976), pp. 734–752.

[13] V. Comincioli, *Analisi numerica: metodi, modelli, applicazioni*, Apogeo, Milano, 2005.

[14] J. P. Delahaye and B. Germain Bonne, *Résultats négatifs en accélération de la convergence*, Numerische Mathematik, 35 (1980), pp. 443–457.

[15] R. P. Eddy, *Extrapolating to the limit of a vector sequence*, in Information Linkage between Applied Mathematics and Industry, P. C. C. Wang, ed., Academic Press, New York, 1979, pp. 387–396.

[16] H. Engl, M. Hanke, and A. Neubauer, *Regularization of Inverse Problems*, Mathematics and Its Applications, Springer Netherlands, 1996.

[17] P. R. Graves Morris, D. E. Roberts, and A. Salam, *The epsilon algorithm and related topics*, J. Comp. Appl. Math., 122 (2000), pp. 51–80.

[18] C. Groetsch, *The theory of Tikhonov regularization for Fredholm equations of the first kind*, Pitman Advanced Publishing Program, 01 1984.

[19] P. C. Hansen, *The truncated SVD as a method for regularization*, BIT, 27 (1987), pp. 543–553.

[20] ——, *Analysis of discrete ill-posed problems by means of the L-curve*, SIAM Review, 34 (1992), pp. 561–580.

[21] ——, *Regularization tools: A Matlab package for analysis and solution of discrete ill-posed problems*, Numerical Algorithms, 6 (1994), pp. 1–35.

[22] P. C. Hansen and D. P. O'Leary, *The use of the L-curve in the regularization of discrete ill-posed problems*, SIAM J. Sci. Comput., 14 (1993), pp. 1487–1503.

[23] D. J. Higham and N. J. Higham, *MATLAB Guide*, SIAM, Philadelphia, PA, USA, third ed., 2017.

[24] K. Jbilou, L. Reichel, and H. Sadok, *Vector extrapolation enhanced TSVD for linear discrete ill-posed problems*, Numer. Algorithms, 51 (2009), pp. 195–208.

[25] Matlab, *version: 9.14.0 (R2023a)*, The MathWorks Inc., Natick, Massachusetts, 2023.

[26] E. Meijering, *A chronology of interpolation: From ancient astronomy to modern signal and image processing*, Proceedings of the IEEE, 90 (2002), pp. 319–342.

[27] M. Mešina, *Convergence acceleration for the iterative solution of the equations* $x = ax + f$, Comput. Meth. Appl. Mech. Eng., 10 (1977), pp. 165–173.

[28] V. A. Morozov, *On the solution of functional equations by the method of regularization*, Doklady Mathematics, 7 (1966), pp. 414–417.

[29] C. C. Paige and M. A. Saunders, *Towards a generalized singular value decomposition*, SIAM J. Numer. Anal., 18 (1981), pp. 398–405.

[30] ——, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Softw., 8 (1982), pp. 43–71.

[31] F. Pes and G. Rodriguez, *The minimal-norm Gauss-Newton method and some of its regularized variants*, Electron. Trans. Numer. Anal., 53 (2020), pp. 459–480.

[32] L. Reichel and G. Rodriguez, *Old and new parameter choice rules for discrete ill-posed problems*, Numerical Algorithms, 63 (2013), pp. 65–87.

[33] L. Reichel, G. Rodriguez, and S. Seatzu, *Error estimates for large-scale ill-posed problems*, Numerical Algorithms, 51 (2009), pp. 341–361.

[34] L. F. Richardson, *IX. The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam*, Philosophical Transactions of the Royal Society of London, 210 (1911), pp. 307–357.

[35] G. Rodriguez, *Algoritmi numerici*, Pitagora Editrice Bologna, Cagliari, 2008.

[36] G. Rodriguez and D. Theis, *An algorithm for estimating the optimal regularization parameter by the L-curve*, Rendiconti di Matematica, 25 (2005), pp. 69–84.

[37] W. Romberg, *Vereinfachte numerische integration*, Det Kongelige Norske Videnskabers Selskab Forhandlinger, 28 (1955), pp. 30–36.

[38] D. Shanks, *Non-linear transformations of divergent and slowly convergent sequences*, J. Math. and Phys., 34 (1955), pp. 1–42.

[39] C. B. Shaw, *Improvement of the resolution of an instrument by numerical solution of an integral equation*, Journal of Mathematical Analysis and Applications, 37 (1972), pp. 83–112.

[40] A. Sidi, *Efficient implementation of minimal polynomial and reduced rank extrapolation methods*, J. Comp. Appl. Math., 36 (1991), pp. 305–337.

[41] G. W. Stewart, *Rank degeneracy*, SIAM Journal on Scientific and Statistical Computing, 5 (1984), pp. 403–413.

[42] A. N. Tikhonov, *Regularization of incorrectly posed problems*, Soviet Math. Dokl., 4 (1963), pp. 1624–1627.

[43] P. Wynn, *Acceleration techniqes for iterated vector and matrix problems*, Math. Comp., 16 (1962), pp. 301–322.