Dispense del corso di Calcolo Numerico

Modulo di Algebra Lineare Numerica

Davide Palitta e Valeria Simoncini

II edizione, v.7, 9 ottobre 2022

Ringrazio gli studenti del corso di Calcolo Numerico della Laurea Triennale in Matematica, a.a.2014-2015, ed in particolare Roberta Lorenzi, per avermi aiutato nella correzione di queste dispense.

V. Simoncini

Indice

Notazione

In tutte le seguenti dispense la notazione adottata sarà la seguente:

- Le lettere maiuscole ($A, B, \dots$) saranno usate per indicare insiemi.
- Le lettere maiuscole in grassetto ($\mathbf{A}, \mathbf{B}, \dots$) saranno usate per indicare matrici. La componente di posizione (i, j) (i-esima riga e j-esima colonna) della matrice $\mathbf{A}$ sarà indicata con la notazione $\mathbf{A}_{ij}$.
- Le lettere minuscole ($x, y, \dots$) saranno usate per indicare vettori, che sono sempre vettori colonna. La componente i-esima del vettore x sarà indicata con la notazione x_i oppure con $(x)_i$. Talvolta, e senza che venga fatta confusione, x_k indicherà l'iterata k-esima di una successione.

  Le lettere quali $i, j, k, l, n, m, \dots$ verranno usate come indici o per indicare dimensioni.
- Le lettere greche ($\alpha, \beta, \dots$) saranno usate per indicare scalari.

Un altra eccezione è fatta quando il contesto non è il *discreto*, bensì il *continuo*: funzioni, variabili, ecc... verranno indicate nel modo consueto (f,$x, \dots$).

Testi di riferimento:

Metodi numerici per l'algebra lineare, D. Bini, M. Capovani, O. Menchi, Zanichelli 1988.

Analisi Numerica - metodi modelli applicazioni, V. Comincioli, McGraw-Hill 1995.

Applied Numerical Linear Algebra, J. W. Demmel, SIAM 1997.

Matrix computations, G. H. Golub e C. F. Van Loan, The Johns Hopkins University Press, 1996 e succ.

Accuracy and Stability of Numerical Algorithms, N. J. Higham, SIAM 1996.

Matematica Numerica, A. Quarteroni, R. Sacco, F. Saleri, III ed., Springer 2008 e succ.

Introduction to Numerical Analysis, J. Stoer, R. Bulirsch, II ed., Springer 1993 e succ.

Capitolo 1

Fondamenti della Matematica Numerica

1.1 Buona posizione di un problema

Si consideri il seguente problema astratto: dati f funzione di x con $f(x) = y$, e $\widehat{x}$ vicino a x, vogliamo capire quanto varierà $f(\widehat{x})$ al variare di $\widehat{x}$ da x, cioè denotato $\widehat{y} = f(\widehat{x})$, siamo interessati a valutare $\widehat{y} - y$.

Definizione 1.1.1 (Vicinanza (o distanza) in senso assoluto) *Diremo che $\widehat{y}$ è vicino in senso assoluto ad y se*

$$|\widehat{y} - y| \simeq C(x)|\widehat{x} - x|, \tag{1.1}$$

dove $C(x) \in \mathbb{R}$ è detto numero di condizionamento assoluto.

In prima approssimazione, per $\widehat{x}$ in un intorno di x, con $\widehat{x} \neq x$, si ha

$$\widehat{y} - y = f(\widehat{x}) - f(x) = \frac{f(\widehat{x}) - f(x)}{\widehat{x} - x}(\widehat{x} - x) \simeq f'(x)(\widehat{x} - x).$$

Si ha quindi

$$C(x) \simeq |f'(x)|.$$

Definizione 1.1.2 (Vicinanza (o distanza) in senso relativo) *Diremo che $\widehat{y}$ è vicino in senso relativo ad y se*

$$\frac{|\widehat{y} - y|}{|y|} \simeq \kappa(x)\frac{|\widehat{x} - x|}{|x|}, \tag{1.2}$$

dove $k(x) \in \mathbb{R}$ è detto numero di condizionamento relativo.

In prima approssimazione, per $\widehat{x}$ in un intorno di x, con $\widehat{x} \neq x$, si ha

$$\frac{\widehat{y} - y}{y} = \frac{f(\widehat{x}) - f(x)}{f(x)} = \frac{f(\widehat{x}) - f(x)}{\widehat{x} - x}(\widehat{x} - x)\frac{x}{xf(x)} \simeq f'(x)\frac{x}{f(x)}\frac{\widehat{x} - x}{x}.$$

Si ha quindi

$$\kappa(x) \simeq \left| f'(x)\frac{x}{f(x)} \right| = \left| f'(x)\frac{x}{y} \right|.$$

A differenza del numero di condizionamento assoluto, il numero di condizionamento relativo tiene conto dell'ordine di grandezza di x e y.

Un problema, in questo caso la valutazione della funzione f, *dipende con continuità dai dati* se per ogni x, $\widehat{x}$ vicini, $f(x)$ e $f(\widehat{x})$ sono anch'essi vicini (in senso assoluto o relativo), cioè se a piccole variazioni dei dati, corrispondono piccole variazioni del risultato dell'operazione.

Definizione 1.1.3 (Buona posizione di un problema) *Un problema si dice ben posto se dipende con continuità dai dati con un numero di condizionamento* moderato. *Si parlerà di ben posizione (o buon condizionamento) assoluto o relativo a seconda della stima scelta. Un problema si dirà invece mal posto quando non è ben posto.*

Trattare numericamente un problema mal posto è molto difficile poichè la soluzione può variare in modo imprevedibile anche per piccoli cambiamenti nelle osservazioni. Esistono tuttavia metodi che permettono di riformulare un problema mal posto in uno ben posto.

Esempio 1.1.4 È data la funzione $f(x) = \sqrt{x}$. Si ha $f'(x) = \frac{1}{2\sqrt{x}}$, da cui

$$C(x) \simeq \frac{1}{2\sqrt{x}}, \quad \text{e} \quad \kappa(x) \simeq \frac{1}{2}.$$

In questo esempio quindi, la valutazione della funzione f è ben condizionata in senso relativo per ogni x, mentre è mal condizionata in senso assoluto per $x \approx 0$.

Esempio 1.1.5 È data la funzione $f(x) = \sin(x)$. Si ha $f'(x) = \cos(x)$, da cui

$$C(x) \simeq |\cos(x)| \leq 1 \quad \text{e} \quad \kappa(x) \simeq \left| \frac{x\cos(x)}{\sin(x)} \right|.$$

Ne segue che $C(x)$ è limitato ed il problema è ben posto in senso assoluto. D'altra parte $\kappa(x)$ cresce per $x \approx k\pi$, con $0 \neq k \in \mathbb{Z}$. Infatti, per $x = 6.2 \approx 2\pi$ e $\hat{x} = 6.2 + 0.01$ si ha

$$\frac{|\widehat{x} - x|}{|x|} = 1.6 \cdot 10^{-3},$$

mentre

$$\frac{|f(\widehat{x}) - f(x)|}{|f(x)|} = \frac{|\sin(\widehat{x}) - \sin(x)|}{|\sin(x)|} = 0.11998.$$

Quindi ad una variazione dell'ordine di 10^{-3} nei dati, corrisponde una perturbazione dell'ordine di 10^{-1} nel risultato, cioè di . ben due ordini di grandezza superiore. Questo problema è quindi mal condizionato in senso relativo, il che è ulteriormente confermato dal numero di condizionamento relativo, $\kappa(x) \simeq 74.36$.

Esempio 1.1.6 È data la funzione $f(x) = \ln(x)$, in un intorno di 1. Si ha $f'(x) = \frac{1}{x}$, da cui

$$k(x) \simeq \left| \frac{x}{x\ln(x)} \right|.$$

Notiamo che per $x = 1.01$ e $\widehat{x} = 1.015$ si ha

$$\frac{|\widehat{x} - x|}{|x|} = 4.9 \cdot 10^{-3}, \quad \text{mentre} \quad \frac{|f(\widehat{x}) - f(x)|}{|f(x)|} = \frac{|\ln(\widehat{x}) - \ln(x)|}{|\ln(x)|} = 0.4962.$$

Quindi anche in questo caso si perdono due ordini di grandezza e il numero di condizionamento relativo calcolato è $\kappa(x) \simeq 100.5$,. Questo problema risulta mal condizionato in senso relativo.

Il significato di numero di condizionamento *moderato* cambierà da problema a problema e questo concetto tornerà più volte anche in seguito.

1.2 Stabilità di metodi numerici

Consideriamo il problema *ben posto*

$$F(x, d) = 0, \tag{1.3}$$

con x incognita e d che indica i coefficienti del problema (e.g., i dati del problema). Un *metodo numerico* consiste nella risoluzione di una successione di equazioni

$$F_n(x_n, d_n) = 0, \quad n \geq 1, \tag{1.4}$$

con la sottintesa speranza che $x_n \to x$ per $n \to \infty$, ovvero che la soluzione numerica converga alla soluzione esatta. Affinchè questo avvenga, è necessario che $F_n \approx F$ e $d_n \approx d$.

Definizione 1.2.1 (Metodo consistente) *Supponendo che il dato d del problema (1.3) sia ammissibile per F_n, un metodo numerico si dice consistente se*

$$F_n(x, d) = F_n(x, d) - F(x, d) \to 0, \quad \text{per } n \to +\infty,$$

essendo x la soluzione di (1.3) corrispondente al dato d.

Definizione 1.2.2 (Metodo fortemente consistente) *Un metodo numerico è detto fortemente consistente se*

$$F_n(x, d) = 0, \quad \forall n,$$

e non solo per $n \to +\infty$.

In generale quindi, *non* saranno fortemente consistenti tutti quei metodi numerici ottenuti dal troncamento di operazioni di passaggio al limite. Invece, sono fortemente consistenti i metodi che provengono da strategie di punto fisso, cioè del tipo $x_{n+1} = \Phi(x_n)$, in quanto se Φ è una contrazione, la soluzione esatta x soddisfa $x = \Phi(x)$.

Definizione 1.2.3 (Metodo numerico ben posto) *Un metodo numerico si dice ben posto o stabile se, per ogni n fissato,*

1. *Esiste x_n in corrispondenza del dato d_n;*
2. *Il calcolo di x_n in funzione di d_n sia unico (o, più precisamente, riproducibile);*
3. *x_n dipenda con continuità dai dati, cioè*

$$\forall \mu > 0,\ \exists\, C_n(\mu, d_n) \text{ tale che } \forall\, \delta d_n,\ \|\delta d_n\| < \mu \text{ si ha } \|\delta x_n\| \leq C_n(\mu, d_n)\|\delta d_n\|,$$

dove δd_n e δx_n indicano rispettivamente la perturbazione su d_n e x_n.

La grandezza $C_n(\mu, d_n)$ è detta numero di condizionamento del metodo numerico. Nel caso il problema sia posto nella forma $x = f(d)$, cosicchè il metodo numerico possa essere scritto nella forma $x_n = f_n(d_n)$, C_n può essere espresso in termini dei numeri di condizionamento assoluto e relativo (C e κ) visti in precedenza (si veda (1.1)-(1.2)).

Esempio 1.2.4 Si consideri la funzione $f(a, b) = a + b$. Si è visto che a causa degli errori di arrotondamento, sulla macchina il problema della valutazione di questa funzione può essere mal posto per $a \approx -b$. Infatti, il calcolo di f può condurre alla *cancellazione di cifre significative* nella rappresentazione del risultato.

L'obbiettivo ultimo dell'approssimazione numerica è, naturalmente, quello di costruire, attraverso problemi numerici del tipo (1.4), soluzioni x_n che "si avvicinano" tanto più alla soluzione del problema dato (1.3) quanto più n diventa grande. Tale concetto è formalizzato nella definizione che segue.

Definizione 1.2.5 (Metodo convergente) *Un metodo numerico $F_n(x_n, d_n) = 0$, si dice convergente se per ogni $\epsilon > 0$, esiste $n_0(\epsilon)$ ed esiste $\delta(n_0, \epsilon) > 0$ tali che*

$$\forall\, n > n_0(\epsilon), \quad \forall\, d_n \ : \ \|d - d_n\| < \delta(n_0, \epsilon) \quad \textit{si ha} \quad \|x(d) - x_n(d_n)\| \leq \epsilon,$$

dove d e d_n sono dati ammissibili rispettivamente per il problema (1.3) associato alla soluzione $x(d)$ e per il problema (1.4) associato alla soluzione x_n.

La definizione sopra dice che un metodo numerico è convergente se la successione di valori $\{x_n\}$ calcolati risolvendo il problema (1.4) tende a x per $n \to +\infty$. Non è detto che questa convergenza avvenga in modo monotòno. Per il caso scalare, una stima dell'andamento della convergenza viene fornita, per esempio, dall'*errore assoluto*

$$|x - x_n|,$$

e dall'*errore relativo*

$$\frac{|x - x_n|}{|x|}, \quad \text{se } x \neq 0.$$

Siccome la soluzione x non è nota, in pratica bisogna utilizzare altri strumenti per monitorare la convergenza. Vedremo in seguito alcuni esempi a seconda dei problemi specifici.

Teorema 1.2.6 (Teorema di equivalenza di Lax-Richtmyer) *Sia dato un problema ben posto, con $d_n \to d$ per $n \to \infty$. Allora, per approssimazioni numeriche consistenti, stabilità e convergenza sono equivalenti.*

Dim. Limitiamo la dimostrazione al caso lineare, dove quindi approssimiamo il problema $Lx = d$, con x incognita, con il problema numerico $L_n x_n = d_n$.

$\boxed{\Rightarrow}$. Supponiamo che il metodo sia stabile e ne proviamo la convergenza. Si ha, $d_n \to d$ per $n \to \infty$. La consistenza dice che $L_n x - d \to 0$ per $n \to +\infty$, da cui $L_n x - Lx = L_n x - d_n + d_n - d \to 0$ per $n \to \infty$. La stabilità implica che L_n^{-1} rimane uniformemente limitata, infatti da $x_n = L_n^{-1} d_n = f_n(d_n)$ ($x = f(d)$ è la forma esplicita del problema), si ha che $f' = L_n^{-1}$ e $\|f'\| = \|L_n^{-1}\| < \infty$, per quanto visto sul condizionamento di un problema. Quindi

$$\begin{aligned} x - x_n &= L_n^{-1} L_n x - L_n^{-1} Lx + L_n^{-1} Lx - L_n^{-1} L_n x_n \\ &= L_n^{-1}(L_n x - Lx) + L_n^{-1}(Lx - L_n x_n) \\ &= L_n^{-1}(L_n x - Lx) + L_n^{-1}(d - d_n) \to 0 \quad \text{per } n \to +\infty. \end{aligned}$$

Dato che $L_n x - Lx \to 0$ e $d - d_n \to 0$ per $n \to +\infty$, segue $x - x_n \to 0$ per $n \to \infty$, cioè il metodo numerico è convergente.

$\boxed{\Leftarrow}$. Supponiamo ora che il metodo numerico sia convergente e denotiamo con $x_n(d)$ la soluzione di $L_n x = d$, e con $x_n(d + \Delta d)$ quella di $L_n x = d + \Delta d$. Allora

$$\begin{aligned} \|x_n(d + \Delta d) - x_n(d)\| &= \|x_n(d + \Delta d) - x_n(d) + x(d + \Delta d) - x(d + \Delta d) + x(d) - x(d)\| \\ &\leq \|x(d) - x_n(d)\| + \|x(d + \Delta d) - x(d)\| + \|x_n(d + \Delta d) - x(d + \Delta d)\| \\ &\leq \epsilon_n^{(1)} + C_1 \|\Delta d\| + \epsilon_n^{(2)}, \end{aligned}$$

dove la prima e la terza disuguaglianza nell'ultima espressione seguono rispettivamente dalla convergenza di $x_n(d)$ e di $x_n(d + \Delta d)$, mentre la seconda dalla ben posizione del problema $Lx = d$. Quindi la perturbazione $\|x_n(d + \Delta d) - x_n(d)\|$ è controllata per n sufficientemente grande tale che $\epsilon_n^{(i)} < \|\Delta d\|$, per $i = 1, 2$. $\square$.

Quindi per un metodo numerico le parole chiave sono: accuratezza, stabilità ed efficienza. La convergenza deve essere assicurata, infatti non avrebbe senso utilizzare un metodo non convergente, ma il metodo deve essere anche stabile, quindi affidabile, ed efficiente, cioè non deve essere troppo costoso in termini di tempo di macchina.

1.3 Sorgenti di errore nei modelli computazionali

Se consideriamo un problema della forma (1.3) e vogliamo risolverlo con un metodo numerico, allora la soluzione effettivamente calcolata risolvendo

$$F_n(x_n, d_n) = 0,$$

sarà una certa quantità x_n che differirà dalla soluzione x di (1.3) per una certa quantità detta *errore*. Si possono avere vari tipi di errore che contribuiscono ad aumentare la *distanza* tra la soluzione x e la soluzione numerica x_n:

- Errori dovuti al modello matematico (per esempio, il modello rappresenta in modo non fedele il problema fisico);
- Errori nell'insieme dei dati (per esempio, i dati sono stati raccolti in modo non accurato);
- Errori di troncamento nel modello numerico: ad esempio, quando si passa dal continuo al discreto, oppure in presenza di passaggi al limite, questi vengono troncati per avere un numero finito di passi;
- Errori di arrotondamento (legati alla rappresentazione dei numeri nel calcolatore);
- Errore computazionale (di responsabilità dell'analista numerico).

L'analisi dell'errore, e quindi l'analisi di stabilità di un metodo numerico, può essere fatta seguendo diverse prospettive:

1. *Analisi in avanti*, in cui si stima l'errore (relativo o assoluto)

   $$|x - x_n|, \quad \text{oppure} \quad \frac{|x - x_n|}{|x|},$$

   dovuto sia a perturbazioni nei dati, sia ad errori intrinseci al metodo numerico;

2. *Analisi all'indietro*, in cui ci si chiede per quali dati la soluzione calcolata x_n sarebbe la soluzione esatta. Quindi se

   $$x_n \approx x,$$

   ci chiediamo se esiste δd_n tale che

   $$F_n(x_n, d_n + \delta d_n) = 0.$$

   Quindi $|\delta d_n|$ è detto *errore assoluto all'indietro* mentre $|\delta d_n|/|d_n|$ *errore relativo all'indietro*. L'idea di stimare l'errore all'indietro permette di interpretare l'errore stesso come perturbazione nei dati. L'avere un errore relativo più piccolo dell'errore nei dati da affidabilità al risultato ottenuto. In pratica, si dirà che un metodo è stabile secondo l'analisi all'indietro se x_n è soluzione di un problema "vicino", nel senso che δd_n è piccolo, in qualche norma.

L'analisi in avanti e quella all'indietro sono due diverse modalità della cosiddetta *analisi a priori*. Essa può essere applicata per indagare non solo la stabilità di un metodo numerico, ma anche la convergenza di quest'ultimo alla soluzione del problema esatto (1.3). Si parlerà in questo caso di *analisi a priori dell'errore*, e potrà ancora essere realizzata con la tecnica in avanti o con quella all'indietro. Essa si distingue dalla cosiddetta *analisi a posteriori dell'errore*, la quale mira a fornire una stima della bontà dell'approssimazione sulla base di quantità effettivamente calcolate e disponibili, usando uno specifico metodo numerico. Tipicamente, nell'analisi a posteriori si stima l'errore $x - x_n$, che non è in generale disponibile a meno di non conoscere la soluzione esatta, in funzione del *residuo* $r_n = F(x_n, d_n)$.

Esempio 1.3.1 Consideriamo il sistema lineare $\mathbf{A}x = b$, con $\mathbf{A}$ matrice quadrata non singolare e b termine noto. Per il sistema perturbato $\tilde{\mathbf{A}}\tilde{x} = \tilde{b}$, l'analisi in avanti fornisce una stima dell'errore $x - \tilde{x}$ in funzione di $\mathbf{A} - \tilde{\mathbf{A}}$ e di $b - \tilde{b}$. D'altra parte, l'analisi all'indietro stima le più piccole perturbazioni $\delta\mathbf{A}$ e δb che dovrebbero essere imposte ai dati $\mathbf{A}$ e b in modo che risulti

$$(\mathbf{A} + \delta\mathbf{A})\widehat{x} = b + \delta b,$$

dove $\widehat{x}$ è una soluzione calcolata del sistema lineare originario con un metodo qualsiasi. Nell'analisi a posteriori si cerca infine una stima dell'errore $x - \widehat{x}$ in funzione del residuo $r = b - \mathbf{A}\widehat{x}$.

Definizione 1.3.2 (Metodo stabile all'indietro) *Un metodo numerico si dice stabile all'indietro (backward stable, in inglese) se ha un errore all'indietro piccolo.*

Osserviamo quindi che con le definizioni date, si ha

$$\text{errore in avanti} \lesssim \text{n. cond. numerico} \times \text{errore all'indietro.}$$

Introduciamo infine altri due concetti chiave del calcolo numerico: la *precisione* e l'*accuratezza*. Con la parola *precisione* ci si riferisce all'esattezza delle operazioni $+, \times, \ldots$, con il termine *accuratezza* ci si riferisce invece all'errore di una quantità approssimata. Nel caso di grandezze scalari, i due concetti coincidono, mentre per calcoli vettoriali o matriciali l'accuratezza può essere molto peggiore della precisione.

Capitolo 2

Risoluzione di sistemi lineari: metodi diretti

Dato il sistema lineare[1]

$$\mathbf{A}x = b, \tag{2.1}$$

con $\mathbf{A} \in \mathbb{R}^{n\times n}$ non singolare e $b \in \mathbb{R}^n$, esistono, in generale, due tipologie di metodi per la sua risoluzione:

(i) Metodi diretti: prevedono la "trasformazione" del problema in uno più semplice da risolvere in un numero finito di passi;

(ii) Metodi iterativi: costruiscono una successione di iterati $x_k \in \mathbb{R}^n$ che, sotto opportune ipotesi, convergono alla soluzione di (2.1).

In questo capitolo saranno trattati alcuni dei metodi della prima classe, mentre nel capitolo successivo verranno discussi alcuni metodi iterativi.

2.1 Analisi di stabilità

In questa sezione introduciamo alcuni concetti e risultati sulla stabilità nell'approssimazione di sistemi lineari. Questi risultati sono di particolare rilevanza nell'uso di metodi diretti. Nel caso di metodi iterativi, si suppone che gli errori di approssimazione siano molto più grandi della precisione di macchina. Nel seguito, $\mathtt{u}$ indica l'unità di *round-off*, $\mathtt{u} = \frac{1}{2}\beta^{1-t}$ con β base della rappresentazione macchina (in generale, $\beta = 2$) e t numero di cifre della rappresentazione macchina.

Iniziamo con la definizione di norma di matrice, usata in modo sistematico in molti risultati di analisi delle matrici e negli aspetti numerici. La definizione è presentata per matrici quadrate complesse, ma può essere generalizzata al caso di matrici rettangolari, con opportune modifiche.

Definizione 2.1.1 *Una funzione $\|\cdot\| : \mathbb{C}^{n\times n} \to \mathbb{R}$ è una norma di matrice se, per ogni $\mathbf{A}, \mathbf{B} \in \mathbb{C}^{n\times n}$ soddisfa le seguenti proprietà:*

1. *$\|\mathbf{A}\| \geq 0$ e $\|\mathbf{A}\| = 0$ se e solo se $\mathbf{A} = \mathbf{0}$, dove $\mathbf{0}$ indica la matrice nulla;*

[1] Per semplicità consideriamo sistemi lineari in $\mathbb{R}$. Molti degli algoritmi e risultati discussi possono essere generalizzati al caso complesso, con alcune semplici modifiche.

2. $\|\alpha\mathbf{A}\| = |\alpha| \cdot \|\mathbf{A}\| \quad \forall \alpha \in \mathbb{C}$;

3. $\|\mathbf{A} + \mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|$;

4. $\|\mathbf{AB}\| \leq \|\mathbf{A}\| \, \|\mathbf{B}\|$, *da cui discende* $\|\mathbf{A}^m\| \leq \|\mathbf{A}\|^m, \quad \forall m \in \mathbb{N}$.

La definizione può essere generalizzata al caso rettangolare, con le opportune modifiche sulle dimensioni delle matrici. Le norme più usate nell'algebra lineare numerica sono:

Norma di Frobenius: $\|\mathbf{A}\|_F := \left(\sum_{i,j} |\mathbf{A}_{i,j}|^2 \right)^{1/2}$,

Norma ℓ_1 e ℓ_∞: $\|\mathbf{A}\|_{\ell_1} = \sum_{i,j} |\mathbf{A}_{ij}|, \quad \|\mathbf{A}\|_{\ell_\infty} = \max_{i,j} |\mathbf{A}_{ij}|$.

Norma-2: $\|\mathbf{A}\|_2 := \max_{0 \neq x \in \mathbb{R}^n} \dfrac{\|\mathbf{A}x\|_2}{\|x\|_2}$,

Per la norma di Frobenius, notiamo che vale $\|\mathbf{A}\|_F^2 = \text{trace}(\mathbf{A}^H \mathbf{A})$, infatti posto $\mathbf{A} = [a_1, \ldots, a_n]$, si ha che $\|\mathbf{A}\|_F^2 = \sum_{j=1}^n \|a_j\|_2^2$, e la diagonale di $\mathbf{A}^H \mathbf{A}$ è proprio elementi uguali a $\|a_j\|_2^2$, $j = 1, \ldots, n$. Come esercizio, verifichiamo che per la norma di Frobenius vale la quarta proprietà. Scriviamo $\mathbf{AB} = [\mathbf{A}b_1, \ldots, \mathbf{A}b_n]$ e $\mathbf{A} = [\hat{a}_1^H; \ldots; \hat{a}_n^H]$. Quindi $\|\mathbf{AB}\|_F^2 = \sum_{j=1}^n \|\mathbf{A}b_j\|_2^2$, con

$$\|\mathbf{A}b_j\|_2^2 = \sum_{i=1}^n (\hat{a}_i^H b_j)^2 \leq \sum_{i=1}^n \|a_i\|^2 \|b_j\|^2 = \|\mathbf{A}\|_F^2 \|b_j\|^2,$$

dove è stata usata la disuguaglianza di Schwarz. Quindi $\|\mathbf{AB}\|_F^2 = \sum_{j=1}^n \|\mathbf{A}b_j\|_2^2 \leq \sum_{j=1}^n \|\mathbf{A}\|_F^2 \|b_j\|^2 = \|\mathbf{A}\|_F^2 \sum_{j=1}^n \|b_j\|^2 = \|\mathbf{A}\|_F^2 \|\mathbf{B}\|_F^2$.

Per la norma ℓ_1, si ha

$$\|\mathbf{AB}\|_{\ell_1} = \sum_{i,j=1}^n |\sum_{k=1}^n \mathbf{A}_{ik}\mathbf{B}_{kj}| \leq \sum_{i,j,k=1}^n |\mathbf{A}_{ik}\mathbf{B}_{kj}| \tag{2.2}$$

$$\leq \sum_{i,j,k,m=1}^n |\mathbf{A}_{ik}\mathbf{B}_{mj}| = (\sum_{i,k=1}^n |\mathbf{A}_{ik}|)(\sum_{j,m=1}^n |\mathbf{B}_{mj}|) = \|\mathbf{A}\|_{\ell_1} \|\mathbf{B}\|_{\ell_1}. \tag{2.3}$$

Si noti che $\|\mathbf{A}\|_{\ell_\infty}$ non è una norma di matrice secondo la definizione data, in quanto non soddisfa la quarta proprietà (In tal caso si parla di norma generalizzata, in alcuni casi di seminorma). Infatti, sia $J = [1, 1; 1, 1]$. Allora $\|J\|_{\ell_\infty} = 1$, e d'altra parte, $2 = \|J^2\|_{\ell_\infty} \leq \|J\|_{\ell_\infty}^2 = 1$, che è un assurdo.

Oltre alla norma-2 descritta sopra, in generale la norma-p matriciale *indotta* dalla norma-p vettoriale è definita da:

$$\|\mathbf{A}\|_p := \max_{0 \neq x \in \mathbb{C}^n} \frac{\|\mathbf{A}x\|_p}{\|x\|_p} = \max_{\|x\|_p = 1} \|\mathbf{A}x\|_p.$$

Altri esempi di norme matriciali indotte sono quindi la norma-1: $\|\mathbf{A}\|_1 := \max_{j=1,\ldots,n} \sum_{i=1}^m |\mathbf{A}_{i,j}|$, e la norma-$\infty$: $\|\mathbf{A}\|_\infty := \max_{i=1,\ldots,m} \sum_{j=1}^n |\mathbf{A}_{i,j}|$,.

Presentiamo ora alcune delle principali proprietà delle norme di matrice:

1. Se $\|\cdot\|$ è una norma matriciale indotta, allora
$$\|\mathbf{A}x\| \leq \|\mathbf{A}\|\,\|x\|, \quad \forall x \in \mathbb{C}^n;$$
2. Se $\|\cdot\|$ è una norma matriciale indotta, allora $\|\mathbf{I}\| = 1$ dove $\mathbf{I}$ indica la matrice identità. Per una qualsiasi norma matriciale vale $\|\mathbf{I}\| \geq 1$ (infatti , $\|\mathbf{I}\| = \|\mathbf{I}^2\| \leq \|\mathbf{I}\|\,\|\mathbf{I}\|$ da cui segue $1 \leq \|\mathbf{I}\|$);
3. $\|\mathbf{A}^{-1}\| \geq \frac{1}{\|\mathbf{A}\|}$ con $\mathbf{A}$ matrice quadrata invertibile e $\|\cdot\|$ una qualsiasi norma matriciale.

La proprietà 2 mostra che la norma di Frobenius non può essere una norma indotta, infatti se $\mathbf{I} \in \mathbb{R}^{n\times n}$ indica la matrice identità di dimensione n, allora
$$\|\mathbf{I}\|_F = \sqrt{n} \neq 1 \quad \text{per } n > 1.$$
Se $\mathbf{A}$ è una matrice reale $n \times n$ simmetrica ($\mathbf{A} = \mathbf{A}^T$) e definita positiva, cioè soddisfa $x^T\mathbf{A}x > 0$ per ogni $0 \neq x \in \mathbb{R}^n$, allora è possibile definire la norma energia (o norma-$\mathbf{A}$) per un vettore $x \in \mathbb{R}^n$:
$$\|x\|_{\mathbf{A}} := (x^T\mathbf{A}x)^{\frac{1}{2}}.$$
Il seguente Lemma è uno strumento molto utile per le stime degli errori.

Lemma 2.1.2 *Sia $\|\cdot\|$ una norma matriciale indotta e sia $\mathbf{A} \in \mathbb{R}^{n\times n}$ matrice con $\|\mathbf{A}\| < 1$. Allora la matrice $\mathbf{I} + \mathbf{A}$ è non singolare, e vale*
$$\|(\mathbf{I}+\mathbf{A})^{-1}\| \leq \frac{1}{1-\|\mathbf{A}\|}.$$

Dimostrazione. Per $\|\mathbf{A}\| < 1$ si ha che $|\lambda| < 1$ per ogni autovalore λ di $\mathbf{A}$. Infatti
$$\|\mathbf{A}\| = \max_{x\neq 0} \frac{\|\mathbf{A}x\|}{\|x\|} < 1,$$
e se $\widehat{\lambda}$ è autovalore di $\mathbf{A}$ con autovettore $\widehat{x}$ si ha che
$$1 > \frac{\|\mathbf{A}\widehat{x}\|}{\|\widehat{x}\|} = \frac{\|\widehat{\lambda}\widehat{x}\|}{\|\widehat{x}\|} = |\widehat{\lambda}|.$$
Questo significa che $\mathbf{I}+\mathbf{A}$ non ha autovalori nulli e quindi non è singolare. Da $(\mathbf{I}+\mathbf{A})(\mathbf{I}+\mathbf{A})^{-1} = \mathbf{I}$, segue che $(\mathbf{I}+\mathbf{A})^{-1} = \mathbf{I} - \mathbf{A}(\mathbf{I}+\mathbf{A})^{-1}$, da cui, essendo $\|\mathbf{I}\| = 1$, segue
$$\|(\mathbf{I}+\mathbf{A})^{-1}\| = \|\mathbf{I} - \mathbf{A}(\mathbf{I}+\mathbf{A})^{-1}\| \leq \|\mathbf{I}\| + \|\mathbf{A}(\mathbf{I}+\mathbf{A})^{-1}\| \leq 1 + \|\mathbf{A}\|\,\|(\mathbf{I}+\mathbf{A})^{-1}\|.$$
Portando a sinistra il termine $\|\mathbf{A}\|\,\|(\mathbf{I}+\mathbf{A})^{-1}\|$ e raccogliendo, otteniamo
$$(1-\|\mathbf{A}\|)\|(\mathbf{I}+\mathbf{A})^{-1}\| \leq 1,$$
da cui segue il risultato per $1 - \|\mathbf{A}\| > 0$. □

Definizione 2.1.3 (Numero di condizionamento di una matrice) *Sia $\mathbf{A} \in \mathbb{R}^{n\times n}$ non singolare. Si dice numero di condizionamento di $\mathbf{A}$ il numero reale*
$$\kappa(\mathbf{A}) = \|\mathbf{A}\|\,\|\mathbf{A}^{-1}\|,$$
con $\|\cdot\|$ norma di matrice. Inoltre vale $\kappa(\mathbf{A}) \geq 1$.

Nel sequente risultato viene mostrato che $1/\kappa(\mathbf{A})$ è una ottima misura della distanza di $\mathbf{A}$ dall'essere una matrice singolare.

Proposizione 2.1.4 *Sia $A \in \mathbb{R}^{n\times n}$ non singolare e $\|\cdot\|$ norma matriciale indotta. Allora*

$$\min\left\{ \frac{\|\delta\mathbf{A}\|}{\|\mathbf{A}\|} \;/\; \mathbf{A}+\delta\mathbf{A} \text{ è singolare } \right\} = \frac{1}{\kappa(\mathbf{A})}. \tag{2.4}$$

Dimostrazione. Poichè $\kappa(\mathbf{A}) = \|\mathbf{A}\|\,\|\mathbf{A}^{-1}\|$, mostrare l'uguaglianza (2.4) equivale a mostrare che

$$\min\{\|\delta\mathbf{A}\| \;/\; \mathbf{A}+\delta\mathbf{A} \text{ è singolare } \} = \frac{1}{\|\mathbf{A}^{-1}\|},$$

Dal Lemma 2.1.2 segue che se $\|\mathbf{A}^{-1}\|\,\|\delta\mathbf{A}\| < 1$, cioè se $\|\delta\mathbf{A}\| < \dfrac{1}{\|\mathbf{A}^{-1}\|}$, allora $\mathbf{I}+\mathbf{A}^{-1}\delta\mathbf{A}$ è non singolare, da cui anche $\mathbf{A}+\delta\mathbf{A}$ è non singolare. Quindi, affinchè $\mathbf{A}+\delta\mathbf{A}$ sia singolare, deve valere $\|\mathbf{A}^{-1}\|\,\|\delta\mathbf{A}\| \geq 1$, cioè $\|\delta\mathbf{A}\| \geq \frac{1}{\|\mathbf{A}^{-1}\|}$.

È possibile quindi costruire una matrice $\delta\mathbf{A}$ che raggiunga l'uguaglianza: sia $x \in \mathbb{R}^n$ tale che $\|x\| = 1$ e tale che $\|\mathbf{A}^{-1}\| = \|\mathbf{A}^{-1}x\|$, cioè $x = \text{argmax}_{\|x\|=1}\|\mathbf{A}^{-1}x\|$. Inoltre, sia $y := \dfrac{\mathbf{A}^{-1}x}{\|\mathbf{A}^{-1}x\|} = \dfrac{\mathbf{A}^{-1}x}{\|\mathbf{A}^{-1}\|}$, cosicchè $\|y\| = 1$. Quindi

$$\delta\mathbf{A} := -\frac{xy^T}{\|\mathbf{A}^{-1}\|},$$

e si ha

$$\|\delta\mathbf{A}\| = \max_{z\neq 0}\frac{\|\delta\mathbf{A}z\|}{\|z\|} = \max_{z\neq 0}\frac{\|xy^Tz\|}{\|\mathbf{A}^{-1}\|\,\|z\|} = \frac{\|x\|}{\|\mathbf{A}^{-1}\|}\max_{z\neq 0}\frac{|y^Tz|}{\|z\|}. \tag{2.5}$$

Si ha $\frac{|y^Tz|}{\|z\|} = |\cos\theta|$, dove θ è l'angolo compreso tra y e z. Quindi il massimo, cioè 1, si ottiene per $z = \alpha y$, con $\alpha \neq 0$. Quindi, siccome $\|x\| = 1$, vale $\|\delta\mathbf{A}\| = 1/\|\mathbf{A}^{-1}\|$. Rimane da mostrare che $\mathbf{A}+\delta\mathbf{A}$ è singolare. Infatti si ha: $(\mathbf{A}+\delta\mathbf{A})y = \mathbf{A}y + \delta\mathbf{A}y = x/\|\mathbf{A}^{-1}\| - x(y^Ty)/\|\mathbf{A}^{-1}\| = 0$. □

Per il calcolo quantitativo ed anche qualitativo di $\kappa_2(\mathbf{A})$, per $\mathbf{A}$ simmetrica vale il seguente risultato,

$$\kappa_2(\mathbf{A}) = \frac{|\lambda_{\max}|}{|\lambda_{\min}|},$$

dove $\lambda_{\max} = \arg\max_{\lambda\in\text{spec}(\mathbf{A})}|\lambda|$ e $\lambda_{\min} = \arg\min_{\lambda\in\text{spec}(\mathbf{A})}|\lambda|$ indicano rispettivamente l'autovalore più grande e quello più piccolo in modulo di $\mathbf{A}$. Per dimostrare tale proprietà, mostriamo innanzi tutto che $\|\mathbf{A}\| = \lambda_{\max}$. Sia $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$ la decomposizione spettrale di $\mathbf{A}$ simmetrica, con $\mathbf{Q}$ ortogonale e $\mathbf{\Lambda} = \text{diag}(\lambda_1,\ldots,\lambda_n)$, matrice diagonale contenente gli autovalori di $\mathbf{A}$. Si ha

$$\|\mathbf{A}\|_2 = \max_{x\neq 0}\frac{\|\mathbf{A}x\|_2}{\|x\|_2} = \max_{\|x\|_2=1}\|\mathbf{A}x\|_2 = \max_{\|x\|_2=1}\|\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^Tx\|_2,$$

Ricordando che una matrice ortogonale è un'isometria (cosicchè $\|\mathbf{Q}x\|_2 = \|x\|_2$), si ha

$$\|\mathbf{A}\|_2 = \max_{\|x\|_2=1}\|\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^Tx\|_2 = \max_{\|\mathbf{Q}^Tx\|_2=1}\|\mathbf{\Lambda}\mathbf{Q}^Tx\|_2 = \max_{\|y\|_2=1}\|\mathbf{\Lambda}y\|_2.$$

Essendo $\mathbf{\Lambda}$ diagonale, si ha direttamente

$$\|\mathbf{\Lambda}y\|_2^2 = \sum_{i=1}^n\lambda_i^2y_i^2 \leq \max_{\lambda\in\text{spec}(\mathbf{A})}|\lambda|^2\sum_{i=1}^n y_i^2 = \max_{\lambda\in\text{spec}(\mathbf{A})}|\lambda|^2\|y\|_2^2.$$

Dunque

$$\|\mathbf{A}\|_2 \leq \max_{\lambda \in \text{spec}(\mathbf{A})} |\lambda| = |\lambda_{\max}|,$$

e il massimo viene raggiunto considerando l'autovettore $\widehat{x}$ di norma unitaria, $\|\widehat{x}\|_2 = 1$, associato a $\lambda_{\max}$, poichè $\|\mathbf{A}\widehat{x}\|_2 = \|\lambda_{\max}\widehat{x}\|_2 = |\lambda_{\max}|$. Abbiamo quindi mostrato che, per $\mathbf{A}$ simmetrica, si ha $\|\mathbf{A}\|_2 = |\lambda_{max}|$.

Avendosi inoltre che gli autovalori di $\mathbf{A}^{-1}$ sono i reciproci degli autovalori di $\mathbf{A}$, si procede come sopra per ottenere $\|\mathbf{A}^{-1}\|_2 = \frac{1}{|\lambda_{\min}|}$. In conclusione, $\kappa_2(\mathbf{A}) = \|\mathbf{A}\|_2 \|\mathbf{A}^{-1}\|_2 = |\lambda_{\max}| \frac{1}{|\lambda_{\min}|} = \frac{|\lambda_{\max}|}{|\lambda_{\min}|}$.

In particolare, questa proprietà mostra che $\kappa_2(\mathbf{A})$ dipende da come sono raggruppati gli autovalori di $\mathbf{A}$: più sono raggruppati, più si riduce la distanza relativa tra $\lambda_{\max}$ e $\lambda_{\min}$ che implicherà un più basso numero di condizionamento.

Come già detto, un metodo numerico per la risoluzione del sistema lineare (2.1) genera una soluzione $x + \delta x$ tale che

$$(\mathbf{A} + \delta\mathbf{A})(x + \delta x) = b + \delta b, \tag{2.6}$$

per qualche perturbazione $\delta\mathbf{A}, \delta b$. Il prossimo risultato fornisce una stima dell'errore relativo atteso, $\|\delta x\|/\|x\|$, (quindi un errore in avanti) in funzione della perturbazione dei dati. Il risultato evidenzia il ruolo di $\kappa(A)$ nella perturbazione della soluzione.

Teorema 2.1.5 *Sia $\mathbf{A} \in \mathbb{R}^{n\times n}$ non singolare e sia $\delta\mathbf{A} \in \mathbb{R}^{n\times n}$ tale che $\|\mathbf{A}^{-1}\| \|\delta\mathbf{A}\| < 1$ con $\|\cdot\|$ norma indotta. Se $x \in \mathbb{R}^n$ è soluzione del sistema lineare (2.1) e $\delta x \in \mathbb{R}^n$ è tale che valga l'equazione (2.6), allora*

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{\kappa(\mathbf{A})}{1 - \kappa(\mathbf{A})\frac{\|\delta\mathbf{A}\|}{\|\mathbf{A}\|}} \left(\frac{\|\delta b\|}{\|b\|} + \frac{\|\delta\mathbf{A}\|}{\|\mathbf{A}\|} \right). \tag{2.7}$$

Dimostrazione. La condizione $\|\mathbf{A}^{-1}\| \|\delta\mathbf{A}\| < 1$ implica che $\mathbf{A} + \delta\mathbf{A}$ sia non singolare. Questo si deduce dalla Proposizione 2.1.4, oppure, direttamente, dal seguente ragionamento:

$$\mathbf{A} + \delta\mathbf{A} = \mathbf{A}(\mathbf{I} + \mathbf{A}^{-1}\delta\mathbf{A}),$$

dove $\|\mathbf{A}^{-1}\delta\mathbf{A}\| \leq \|\mathbf{A}^{-1}\| \|\delta\mathbf{A}\| < 1$. Quindi il Lemma 2.1.2 implica che $\mathbf{I} + \mathbf{A}^{-1}\delta\mathbf{A}$ è non singolare. Dato che $\mathbf{A}$ è non singolare per ipotesi, si ha che anche $\mathbf{A} + \delta\mathbf{A}$ è non singolare.

Ora sia x soluzione di (2.1), cioè $\mathbf{A}x = b$. Sostituendo questa relazione nell'equazione (2.6) si ottiene

$$\mathbf{A}\delta x + \delta\mathbf{A}(x + \delta x) = \delta b, \quad \Rightarrow \quad (\mathbf{A} + \delta\mathbf{A})\delta x = \delta b - \delta\mathbf{A}x,$$

da cui, moltiplicando per $\mathbf{A}^{-1}$,

$$(\mathbf{I} + \mathbf{A}^{-1}\delta\mathbf{A})\delta x = \mathbf{A}^{-1}(\delta b - \delta\mathbf{A}x).$$

Grazie al Lemma 2.1.2, vale

$$\|(\mathbf{I} + \mathbf{A}^{-1}\delta\mathbf{A})^{-1}\| \leq \frac{1}{1 - \|\mathbf{A}^{-1}\delta\mathbf{A}\|} \leq \frac{1}{1 - \|\mathbf{A}^{-1}\| \|\delta\mathbf{A}\|}.$$

(la seconda disugaglianza segue da $\|\mathbf{A}^{-1}\delta\mathbf{A}\| \leq \|\mathbf{A}^{-1}\| \|\delta\mathbf{A}\|$). Quindi,

$$\|\delta x\| = \|(\mathbf{I} + \mathbf{A}^{-1}\delta\mathbf{A})^{-1}\mathbf{A}^{-1}(\delta b - \delta\mathbf{A}x)\| \leq \frac{\|\mathbf{A}^{-1}\|}{1 - \|\mathbf{A}^{-1}\| \|\delta\mathbf{A}\|} \left(\|\delta\mathbf{A}\| \dot{\|}x\| + \|\delta b\| \right). \tag{2.8}$$

Ricordando che $b = \mathbf{A}x$, si ha $\|b\| \leq \|\mathbf{A}\|\,\|x\|$ per qualsiasi norma indotta, cioè $\dfrac{1}{\|x\|} \leq \dfrac{\|\mathbf{A}\|}{\|b\|}$. Dividendo quindi la disuguaglianza (2.8) per $\|x\|$ e raccogliendo $\|\mathbf{A}\|$ si ottiene

$$\frac{\|\delta x\|}{\|x\|} =\leq \frac{\|\mathbf{A}^{-1}\|}{1-\|\mathbf{A}^{-1}\|\,\|\delta\mathbf{A}\|}\left(\|\delta\mathbf{A}\| + \frac{\|\delta b\|}{\|x\|}\right) \leq \frac{\|\mathbf{A}^{-1}\|\,\|\mathbf{A}\|}{1-\|\mathbf{A}^{-1}\|\,\|\delta\mathbf{A}\|}\left(\frac{\|\delta\mathbf{A}\|}{\|\mathbf{A}\|} + \frac{\|\delta b\|}{\|b\|}\right).$$

Il risultato è ottenuto osservando che $\|\mathbf{A}^{-1}\|\,\|\delta\mathbf{A}\| = \kappa(\mathbf{A})\dfrac{\|\delta\mathbf{A}\|}{\|\mathbf{A}\|}$. □

Il risultato fornisce un esempio di relazione tra l'errore in avanti e l'errore all'indietro, come descritto nel capitolo precedente. Infatti si verifica che l'errore relativo in avanti, $\|\delta x\|/\|x\|$ è maggiorato dall'errore all'indietro, moltiplicato per il numero di condizionamento della matrice.

In generale il Teorema 2.1.5 può fornire stime pessimistiche di $\|\delta x\|$, anche se ci sono esempi di matrici, come il seguente, per i quali la stima è raggiunta, almeno qualitativamente.

Esempio 2.1.6 La matrice di Hilbert è definita come

$$\mathbf{H}_{i,j} = \frac{1}{i+j-1}, \qquad \mathbf{H}_n = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \ddots \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{pmatrix} \in \mathbb{R}^{n\times n}.$$

La matrice di Hilbert è molto mal condizionata al crescere della dimensione. Ad esempio, se consideriamo la norma-2, per $n = 4$ si ha $\kappa_2(\mathbf{H}_4) = 1.551 \cdot 10^4$, e già per $n = 8$, si ha $\kappa_2(\mathbf{H}_8) = 1.526 \cdot 10^{10}$. In generale si ha $\kappa_2(\mathbf{H}_n) \approx \mathrm{e}^{3.5n}$ per n sufficientemente grande.

Con la matrice $\mathbf{H}_8$, consideriamo $\mathbf{H}_8 x = b$ con $b = (1,\ldots,1)^T \in \mathbb{R}^8$. Supponiamo che sia la matrice che il termine noto vengano perturbati,

$$\widehat{\mathbf{H}}_8 = \mathbf{H}_8 + \delta\mathbf{H}_8, \qquad \widehat{b} = b + \delta b,$$

dove $\delta\mathbf{H}_8 = 10^{-11}\cdot\mathbf{E}$, $\delta b = 10^{-11}e$ con $\mathbf{E}$ ed e matrice e vettore di numeri casuali presi da una distribuzione normale (funzione `randn` in Matlab). Un calcolo diretto mostra che $\|\delta\mathbf{H}_8\| \approx 6.6\cdot 10^{-11}$, e $\|\delta b\| \approx 2.4\cdot 10^{-11}$, quindi entrambe sono perturbazioni piuttosto piccole. Siccome $\|\delta\mathbf{H}_8\|\,\|\mathbf{H}_8^{-1}\| = 6\cdot 10^{-1} < 1$ è possibile applicare il Teorema 2.1.5, il quale prevede una perturbazione nella soluzione finale del tipo $\frac{\|\delta x\|}{\|x\|} < 4.72$. In effetti il calcolo diretto mostra che $\frac{\|\delta x\|}{\|x\|} = 0.0708\ldots$, quindi leggermente minore della stima fornita dal teorema, ma comunque di quasi 10 ordini di grandezza superiore alla perturbazione dai dati.

Corollario 2.1.7 *Nelle ipotesi del Teorema 2.1.5, se $\delta\mathbf{A} = 0$, allora*

$$\frac{1}{\kappa(\mathbf{A})}\frac{\|\delta b\|}{\|b\|} \leq \frac{\|\delta x\|}{\|x\|} \leq \kappa(\mathbf{A})\frac{\|\delta b\|}{\|b\|}.$$

Dimostrazione. Dimostriamo solo la prima disuguaglianza in quanto la seconda segue direttamente dalla (2.7) per $\delta\mathbf{A} = 0$. Dalla relazione $\mathbf{A}\delta x = \delta b$ discende

$$\|\delta b\| \leq \|\mathbf{A}\|\,\|\delta x\|.$$

Moltiplicando ambo i membri per $\|x\|$ e ricordando che $\|x\| \leq \|\mathbf{A}^{-1}\|\cdot\|b\|$, si ha

$$\|x\|\,\|\delta b\| \leq \kappa(\mathbf{A})\|b\|\,\|\delta x\|,$$

e quindi la disuguaglianza desiderata. □

Nell'interesse algoritmico, in generale $\|\delta b\|$ e $\|\delta \mathbf{A}\|$ dipendono dal troncamento avvenuto durante la rappresentazione dei dati sulla macchina e dovranno essere stimati in funzione delle caratteristiche dell'aritmetica finita utilizzata. È quindi ragionevole supporre che le perturbazioni relative sui dati soddisfino

$$\frac{\|\delta b\|}{\|b\|} < \gamma, \qquad \frac{\|\delta \mathbf{A}\|}{\|\mathbf{A}\|} < \gamma,$$

con γ costante "piccola", cioè si suppone che la perturbazione nel dato sia significativamente più piccola della norma del dato stesso. Si considera spesso $\gamma = \beta^{1-t}$ dove β è la base della rappresentazione dei numeri macchina (in generale, $\beta = 2$) e t il numero di cifre significative della rappresentazione stessa.

2.2 Risoluzione di sistemi triangolari

È dato il sistema lineare

$$\mathbf{L}x = b, \tag{2.9}$$

dove la matrice dei coefficienti $\mathbf{L}$ è *triangolare inferiore* (dove "L" sta per *Lower triangular*) cioè della forma

$$\mathbf{L} = \begin{pmatrix} \mathbf{L}_{11} & & & \\ \mathbf{L}_{21} & \mathbf{L}_{22} & & \\ \vdots & \dots & \ddots & \\ \mathbf{L}_{n1} & \dots & \dots & \mathbf{L}_{nn} \end{pmatrix}.$$

La condizione di non singolarità di $\mathbf{L}$ è assicurata da $\mathbf{L}_{ii} \neq 0$, $\forall i$. Per risolvere il sistema lineare (2.9), è possibile calcolare le componenti del vettore x in modo sequenziale, con il *metodo delle sostituzioni in avanti*:

$$\begin{aligned} x_1 &= \frac{b_1}{\mathbf{L}_{11}}, \\ x_2 &= \frac{1}{\mathbf{L}_{22}}\left(b_2 - \mathbf{L}_{21}x_1\right), \\ &\vdots \\ x_i &= \frac{1}{\mathbf{L}_{ii}}\left(b_i - \sum_{j=1}^{i-1} \mathbf{L}_{ij}x_j\right), \qquad i = 3, \dots, n. \end{aligned} \tag{2.10}$$

Considerazioni analoghe valgono per un sistema lineare $\mathbf{U}x = b$, con $\mathbf{U}$ matrice triangolare superiore ("U" sta per *Upper triangular*). In questo caso, l'algoritmo per determinare le componenti del vettore incognito x prende il nome di *metodo delle sostituzioni all'indietro* che, nel caso generale, assume la seguente forma:

$$\begin{aligned} x_n &= \frac{b_n}{\mathbf{U}_{nn}}, \\ &\vdots \\ x_i &= \frac{1}{\mathbf{U}_{ii}}\left(b_i - \sum_{j=i+1}^{n} \mathbf{U}_{ij}x_j\right), \qquad i = n-1, \dots, 1. \end{aligned} \tag{2.11}$$

In entrambi i casi, l'i-esima componente del vettore x viene calcolata con $(2(i-1)-1)+1+1 = 2i-1$ operazioni floating point (flops). Questo significa che per la risoluzione di un sistema triangolare $n \times n$ sono necessarie

$$\sum_{i=1}^{n}(2i-1) = 2\frac{n(n+1)}{2} - n = n^2 \text{ flops.}$$

Si può dimostrare che la soluzione trovata risolve il problema triangolare (assumendo $\delta b = 0$)

$$(\mathbf{L} + \delta\mathbf{L})x = b, \quad \text{con} \quad \|\delta\mathbf{L}\|_F \leq \frac{n\mathtt{u}}{1-n\mathtt{u}}\|\mathbf{L}\|_F, \tag{2.12}$$

o in alternativa, $\|\delta\mathbf{L}\|_F \leq n\mathtt{u}\|\mathbf{L}\|_F + O(\mathtt{u}^2)$. Quindi se $\|\delta\mathbf{L}\|_F$ è piccola, la soluzione calcolata risolve un problema vicino, e quindi il metodo è stabile per l'analisi dell'errore all'indietro. Se $n\mathtt{u}\kappa_F(\mathbf{L}) < 1$, allora

$$\frac{\|\delta x\|_F}{\|x\|_F} \leq \frac{n\mathtt{u}\kappa_F(\mathbf{L})}{1-n\mathtt{u}\kappa_F(\mathbf{L})}.$$

Tale risultato mostra che la perturbazione nella soluzione è piccola (quindi un piccolo errore in avanti), se il problema è ben posto.

2.2.1 Calcolo dell'inversa di una matrice triangolare

L'algoritmo descritto in (2.10) può essere adattato per il calcolo esplicito dell'inversa di una matrice triangolare inferiore $\mathbf{L}$. Infatti, osserviamo che se v_i indica l'i-esimo vettore colonna dell'inversa, $\mathbf{L}^{-1} = [v_1, \ldots, v_n]$, allora ogni colonna i dell'inversa è soluzione del seguente sistema lineare

$$\mathbf{L}v_i = e_i, \qquad i = 1, \ldots, n, \tag{2.13}$$

essendo $\{e_i\}$ la base canonica di $\mathbb{R}^n$. A priori, questo comporta la risoluzione di n sistemi lineari triangolari di dimensioni $n \times n$. D'altra parte, per ogni i, le prime $i-1$ componenti del vettore $v_i = [v(1:i-1); v(i:n)]$ sono nulle. Questo può essere visto confrontando i due lati dell'uguaglianza,

$$\begin{pmatrix} \mathbf{L}_{1:i-1,1:i-1} & 0 \\ \mathbf{L}_{i:n,1:i-1} & \mathbf{L}_{i:n,i:n} \end{pmatrix} \begin{pmatrix} v(1:i-1) \\ v(i:n) \end{pmatrix} = \begin{pmatrix} 0_{1:i-1} \\ e_1 \end{pmatrix},$$

con $e_i = [0_{1:i-1}; e_1]$, dove e_1 qui ha dimensione $n-i+1$. I sistemi rimanenti, $\mathbf{L}_{i:n,i:n}v(i:n) = e_1$, sono ancora triangolari inferiori ma di dimensioni k, $k = i, \ldots, n$. Il costo computazionale per il calcolo dell'inversa di una matrice triangolare inferiore è quindi

$$\sum_{j=1}^{n} j^2 = \frac{n(n+1)(2n+1)}{6} = \frac{n^3}{3} + \mathcal{O}\left(n^2\right).$$

Questa procedura può essere applicata anche per il calcolo dell'inversa di una matrice triangolare superiore e, anche in questo caso, il numero di operazioni floating point previsto dall'algoritmo è un $\frac{n^3}{3} + \mathcal{O}(n^2)$.

2.3 Fattorizzazioni

È dato il sistema lineare $\mathbf{A}x = b$, con $\mathbf{A} \in \mathbb{R}^{n\times n}$ matrice invertibile e $b \in \mathbb{R}^n$. Se $\mathbf{A}$ non ha una struttura particolare, una procedura di risoluzione del sistema consiste nella fattorizzazione delle matrice nel prodotto di due matrici, per le quali la risoluzione dei sistemi associati abbia un costo

computazionale più basso che per l'intera matrice. Supponendo che sia possibile determinare una fattorizzazione del tipo $\mathbf{A} = \mathbf{BC}$, si risolveranno in sequenza i seguenti sistemi lineari:

$$i)\ \mathbf{B}y = b, \qquad ii)\ \mathbf{C}x = y. \tag{2.14}$$

Alcuni esempi di fattorizzazioni sono:

- Fattorizzazione LU:
$$\mathbf{A} = \mathbf{LU},$$
dove $\mathbf{L}$ è triangolare inferiore mentre $\mathbf{U}$ è triangolare superiore. Questa fattorizzazione, quando esiste, è ottenuta applicando l'algoritmo di eliminazione di Gauss;

- Fattorizzazione di Cholesky:
$$\mathbf{A} = \mathbf{L}\mathbf{L}^T,$$
dove $\mathbf{L}$ è triangolare inferiore e $\mathbf{A}$ è simmetrica e definita positiva. La fattorizzazione è ottenuta applicando il metodo di Cholesky;

- Fattorizzazione QR:
$$\mathbf{A} = \mathbf{QR},$$
dove $\mathbf{Q}$ è unitaria e $\mathbf{R}$ è triangolare superiore. Questa fattorizzazione esiste per ogni matrice $\mathbf{A}$, anche rettangolare, ed è ottenuta mediante trasformazioni ortogonali, per esempio di Householder, o mediante processi di ortogonalizzazione di Gram-Schmidt.

Il costo di queste fattorizzazioni è un $\mathcal{O}(n^3)$ a cui va aggiunto il costo della risoluzione dei sistemi in (2.14). In tutte le fattorizzazioni elencate, tale costo è $\mathcal{O}(n^2)$ (nel caso della fattorizzazione QR, la risoluzione del sistema con $\mathbf{Q}$ corrisponde all'operazione $y = \mathbf{Q}^T b$, il cui costo è ancora $\mathcal{O}(n^2)$).

Nel seguito di questo capitolo saranno trattate in dettaglio le fattorizzazioni LU e di Cholesky, mentre la fattorizzazione QR verrà descritta nell'ambito dei problemi agli autovalori ed ai minimi quadrati.

2.3.1 Metodo di eliminazione di Gauss

Consideriamo il sistema lineare

$$\mathbf{A}x = b, \tag{2.15}$$

con $\mathbf{A} \in \mathbb{R}^{n\times n}$ e $b \in \mathbb{R}^n$. Questa procedura determina una fattorizzazione di $\mathbf{A}$,

$$\mathbf{A} = \mathbf{LU},$$

con $\mathbf{L} \in \mathbb{R}^{n\times n}$ triangolare inferiore e $\mathbf{U} \in \mathbb{R}^{n\times n}$ triangolare superiore, chiamata fattorizzazione LU. In particolare la procedura calcola direttamente la matrice $\mathbf{U}$ mentre solo implicitamente la matrice $\mathbf{L}$. Durante la riduzione verrà sfruttata in modo essenziale la proprietà per la quale sostituendo un'equazione del sistema con la differenza tra l'equazione stessa ed un'altra, moltiplicata per una costante non nulla, si ottiene un sistema equivalente (cioè con la stessa soluzione) a quello di partenza.

Sia dunque la matrice $\mathbf{A}^{(1)} := \mathbf{A}$ ed indichiamo il sistema originario come

$$\mathbf{A}^{(1)}x = b^{(1)}.$$

Supponendo che $\mathbf{A}^{(1)}_{11} \neq 0$, definiamo i seguenti *moltiplicatori*

$$m_{i1} = \frac{\mathbf{A}^{(1)}_{i1}}{\mathbf{A}^{(1)}_{11}}, \quad i = 2, 3, \ldots, n,$$

L'applicazione della strategia di eliminazione per gli elementi della prima colonna viene fatta mediante la seguente operazione:

$$\begin{aligned} \mathbf{A}_{ij}^{(2)} &= \mathbf{A}_{ij}^{(1)} - m_{i1}\mathbf{A}_{1j}^{(1)}, \quad i = 2, \ldots, n, j = 1, \ldots, n \\ b_i^{(2)} &= b_i^{(1)} - m_{i1}b_1^{(1)}, \quad i = 2, \ldots, n, \end{aligned} \tag{2.16}$$

che determina un nuovo sistema, equivalente a quello di partenza, della forma

$$\begin{pmatrix} \mathbf{A}_{11}^{(1)} & \mathbf{A}_{12}^{(1)} & \ldots & \mathbf{A}_{1n}^{(1)} \\ 0 & \mathbf{A}_{22}^{(2)} & \ldots & \mathbf{A}_{2n}^{(2)} \\ \vdots & \vdots & & \vdots \\ 0 & \mathbf{A}_{n2}^{(2)} & \ldots & \mathbf{A}_{nn}^{(2)} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1^{(1)} \\ b_2^{(2)} \\ \vdots \\ b_n^{(2)} \end{pmatrix} \quad \Leftrightarrow \quad \mathbf{A}^{(2)}x = b^{(2)}. \tag{2.17}$$

La procedura quindi ha eliminato gli elementi della prima colonna, sotto la diagonale principale, nella matrice risultante $\mathbf{A}^{(2)}$, mantenendo il sistema equivalente al precedente. Si noti che l'operazione in (2.16) non è necessaria per $i = 2$ e $j = 1$, in quanto si sa a priori che gli elementi sotto la diagonale, della prima colonna, saranno zero. Quindi in una tipica implementazione, entrambi gli indici sono definiti da 2 in poi, cioè $i, j = 2, \ldots n$. Useremo questa notazione nel seguito. Questo risparmio computazionale richiede comunque che al termine della procedura di eliminazione di Gauss, gli elementi sotto la diagonale principale siano comunque scartati, perchè "logicamente" zero.

Proseguendo in modo analogo, e supponendo $\mathbf{A}_{kk}^{(k)} \neq 0$ per $k = 1, \ldots, i-1$, viene definito il moltiplicatore

$$m_{ik} = \frac{\mathbf{A}_{ik}^{(k)}}{\mathbf{A}_{kk}^{(k)}}, \quad i = k+1, \ldots, n,$$

e le nuove componenti

$$\mathbf{A}_{ij}^{(k+1)} = \mathbf{A}_{ij}^{(k)} - m_{ik}\mathbf{A}_{kj}^{(k)}, \quad i, j = k+1, \ldots, n,$$

$$b_i^{(k+1)} = b_i^{(k)} - m_{ik}b_k^{(k)}, \quad i = k+1, \ldots, n.$$

Si otterrà una successione finita di sistemi lineari $\mathbf{A}^{(k)}x = b^{(k)}$, $k = 1, \ldots, n$, dove per $k \geq 2$, la matrice $\mathbf{A}^{(k)}$ ha la forma seguente

$$\mathbf{A}^{(k)} = \begin{pmatrix} \mathbf{A}_{11}^{(1)} & \mathbf{A}_{12}^{(1)} & \ldots & \ldots & \ldots & \mathbf{A}_{1n}^{(1)} \\ 0 & \mathbf{A}_{22}^{(2)} & & & & \mathbf{A}_{2n}^{(2)} \\ \vdots & & \ddots & & & \vdots \\ 0 & \ldots & 0 & \mathbf{A}_{kk}^{(k)} & \ldots & \mathbf{A}_{kn}^{(k)} \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & \ldots & 0 & \mathbf{A}_{nk}^{(k)} & \ldots & \mathbf{A}_{nn}^{(k)} \end{pmatrix}.$$

Al termine delle $n-1$ iterazioni, cioè per $k = n$, si ottiene un sistema triangolare superiore

$$\mathbf{A}^{(n)}x = b^{(n)}, \tag{2.18}$$

con $\mathbf{U} := \mathbf{A}^{(n)}$ triangolare superiore. Per risolvere il sistema (2.18) è possibile utilizzare il metodo delle sostituzioni all'indietro con un costo computazionale di $\mathcal{O}(n^2)$ flops, a cui va aggiunto il

costo della fattorizzazione. Il numero di operazioni floating point per portare a termine la k-esima iterazione dell'eliminazione di Gauss è di $n-k$ operazioni per il calcolo di m_{ik}, poi $2(n-k)(n-k)$ operazioni per l'aggiornamento di $\mathbf{A}^{(k+1)}$, e $2(n-k)$ operazioni per l'aggiornamento di $b^{(k)}$, per un totale di $(n-k)+2(n-k)(n-k)+2(n-k) = 2(n-k)^2+3(n-k)$ operazioni al passo k. Il costo totale della procedura di eliminazione è quindi:

$$\sum_{k=1}^{n-1}[2(n-k)^2+3(n-k)] = 2\left(\sum_{k=1}^{n-1}n^2+\sum_{k=1}^{n-1}k^2-2\sum_{k=1}^{n-1}nk\right)+3\sum_{k=1}^{n-1}(n-k) = \frac{2}{3}n^3+\mathcal{O}(n^2).$$

Il processo di eliminazione di Gauss è equivalente ad una fattorizzazione del tipo $\mathbf{A} = \mathbf{LU}$. La matrice $\mathbf{U}$ corrisponde a $\mathbf{A}^{(n)}$, mentre $\mathbf{L}$ non viene esplicitamente costruita. In effetti $\mathbf{L}$ viene costruita implicitamente mediante i moltiplicatori m_{ik} definiti durante l'iterazione. Infatti, sia

$$\mathbf{M}^{(1)} = \begin{pmatrix} 1 & & & & \\ -m_{21} & 1 & & & \\ -m_{31} & 0 & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ -m_{n1} & 0 & \dots & 0 & 1 \end{pmatrix} = \mathbf{I} - m_1 e_1^T, \quad \text{con} \quad m_1 = \begin{pmatrix} 0 \\ m_{21} \\ m_{31} \\ \vdots \\ m_{n1} \end{pmatrix} \in \mathbb{R}^n.$$

Analogamente, al k-esimo ciclo di eliminazione sia

$$\mathbf{M}^{(k)} = \begin{pmatrix} 1 & & & & & & \\ 0 & \ddots & & & & & \\ \vdots & \ddots & \ddots & & & & \\ 0 & \dots & 0 & 1 & & & \\ 0 & \dots & 0 & -m_{k+1,k} & 1 & & \\ \vdots & & \vdots & \vdots & \vdots & \ddots & \\ 0 & \dots & 0 & -m_{nk} & 0 & \dots & 1 \end{pmatrix} = \mathbf{I} - m_k e_k^T, \qquad \text{con} \quad m_k = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ m_{k+1,k} \\ \vdots \\ m_{nk} \end{pmatrix} \in \mathbb{R}^n.$$

Si verifica che

$$(\mathbf{M}^{(k)})^{-1} = \begin{pmatrix} 1 & & & & & & \\ 0 & \ddots & & & & & \\ \vdots & \ddots & \ddots & & & & \\ 0 & \dots & 0 & 1 & & & \\ 0 & \dots & 0 & m_{k+1,k} & 1 & & \\ \vdots & & \vdots & \vdots & \vdots & \ddots & \\ 0 & \dots & 0 & m_{nk} & 0 & \dots & 1 \end{pmatrix} = \mathbf{I} + m_k e_k^T,$$

infatti

$$(\mathbf{I} - m_k e_k^T)(\mathbf{I} + m_k e_k^T) = \mathbf{I} + m_k e_k^T - m_k e_k^T - (m_k e_k^T)(m_k e_k^T) = \mathbf{I} - m_k \underbrace{(e_k^T m_k)}_{=0} e_k^T = \mathbf{I},$$

dove è stato sfruttato il fatto che m_k ha componenti zero fino alla k-esima inclusa.

Scrivendo esplicitamente il prodotto $\mathbf{M}^{(1)}\mathbf{A}$, si verifica che questo coincide con il calcolo in (2.17), ed analogamente per $\mathbf{M}^{(1)}b$. Quindi si ha

$$\mathbf{M}^{(1)}\mathbf{A}x = \mathbf{M}^{(1)}b \quad \Leftrightarrow \quad \mathbf{A}^{(2)}x = b^{(2)}.$$

In modo analogo si verifica che

$$\mathbf{M}^{(n-1)}\mathbf{M}^{(n-2)}\cdots\mathbf{M}^{(2)}\mathbf{M}^{(1)}\mathbf{A} = \mathbf{A}^{(n)} \tag{2.19}$$

con $\mathbf{A}^{(n)} = \mathbf{U}$, e

$$\mathbf{M}^{(n-1)}\mathbf{M}^{(n-2)}\cdots\mathbf{M}^{(2)}\mathbf{M}^{(1)}b = b^{(n)}. \tag{2.20}$$

Sia $\mathbf{L}^{-1} := \mathbf{M}^{(n-1)}\mathbf{M}^{(n-2)}\cdots\mathbf{M}^{(2)}\mathbf{M}^{(1)}$; questa matrice è triangolare inferiore, in quanto prodotto di matrici triangolari inferiori, e anche non singolare, poichè tutti i suoi elementi diagonali sono diversi da zero (sono tutti 1). La matrice $\mathbf{L}$ cercata è quindi data da

$$\begin{aligned}\mathbf{L} &= (\mathbf{M}^{(1)})^{-1}(\mathbf{M}^{(2)})^{-1}\cdots(\mathbf{M}^{(n-2)})^{-1}(\mathbf{M}^{(n-1)})^{-1} \\ &= (\mathbf{I}+m_1e_1^T)(\mathbf{I}+m_2e_2^T)\cdots(\mathbf{I}+m_{n-1}e_{n-1}^T) = \mathbf{I}+\sum_{i=1}^{n-1} m_ie_i^T,\end{aligned}$$

dove l'ultima uguaglianza è ottenuta osservando che $(m_ie_i^T)(m_je_j^T) = 0$, $\forall\, i < j$. Dunque, sotto la diagonale principale, la matrice $\mathbf{L}$ contiene tutti i moltiplicatori della procedura di eliminazione. Dalla (2.19) si ottiene $\mathbf{L}^{-1}\mathbf{A} = \mathbf{A}^{(n)} = \mathbf{U}$, cioè $\mathbf{A} = \mathbf{LU}$.

La procedura vista fino ad ora può essere applicata solo sotto l'ipotesi restrittiva che $\mathbf{A}_{kk}^{(k)} \neq 0$, come appare evidente nell'esempio che segue.

Esempio 2.3.1 Consideriamo la matrice

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 7 & 8 & 9 \end{pmatrix}.$$

Dopo il primo ciclo di eliminazione,

$$\mathbf{A}^{(2)} = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 0 & -1 \\ 0 & -6 & -12 \end{pmatrix},$$

e la procedura si ferma poichè $\mathbf{A}_{22}^{(2)} = 0$.

Il seguente teorema dà condizioni sufficienti per l'esistenza ed unicità della fattorizzazione LU di $\mathbf{A}$.

Teorema 2.3.2 *Sia $\mathbf{A} \in \mathbb{R}^{n\times n}$ e siano $\mathbf{A}_k$ le sue sottomatrici principali dominanti di dimensione $k \times k$. Se $\mathbf{A}_k$ è non singolare per $k = 1, \ldots, n-1$ allora esiste ed è unica la fattorizzazione LU di $\mathbf{A}$ in cui gli elementi diagonali della matrice $\mathbf{L}$ sono tutti 1.*

Dimostrazione. Si procede per induzione su n.
Sia $n = 1$. Allora $\mathbf{A} = (\mathbf{A}_{11})$ e quindi $\mathbf{L} = 1$, e $\mathbf{U} = \mathbf{A}_{11}$, univocamente determinati.
Sia ora $n = k > 1$. Allora

$$\mathbf{A}_k = \begin{pmatrix} \mathbf{A}_{k-1} & d \\ c^T & \alpha \end{pmatrix}, \qquad d, c \in \mathbb{R}^{k-1}, \quad \alpha \in \mathbb{R}. \tag{2.21}$$

Per ipotesi induttiva si ha $\mathbf{A}_{k-1} = \mathbf{L}_{k-1}\mathbf{U}_{k-1}$ non singolare. Siano quindi

$$\mathbf{L}_k := \begin{pmatrix} \mathbf{L}_{k-1} & 0 \\ u^T & 1 \end{pmatrix}, \qquad \mathbf{U}_k := \begin{pmatrix} \mathbf{U}_{k-1} & v \\ 0 & \beta \end{pmatrix},$$

con $u, v \in \mathbb{R}^{k-1}$ e $\beta \in \mathbb{R}$ da determinare. Dal confronto tra $\mathbf{A}_k$ e

$$\mathbf{L}_k\mathbf{U}_k = \begin{pmatrix} \mathbf{A}_{k-1} & \mathbf{L}_{k-1}v \\ u^T\mathbf{U}_{k-1} & u^Tv + \beta \end{pmatrix},$$

si ottiene

$$\mathbf{L}_{k-1}v = d, \quad u^T\mathbf{U}_{k-1} = c^T, \quad u^Tv + \beta = \alpha.$$

Essendo $\mathbf{L}_{k-1}$ e $\mathbf{U}_{k-1}$ non singolari per ipotesi, u, v, e β sono univocamente determinati. □

Si noti che la fattorizzazione LU può esistere anche se $\mathbf{A}$ è singolare. La singolarità di $\mathbf{A}$ si rifletterà nella singolarità di $\mathbf{U}$.

Nonostante i fattori della decomposizione LU siano strettamente legati alla matrice che determinano, alcune buone proprietà di $\mathbf{A}$ possono non essere trasmesse ai fattori, come mostrato nel seguente esempio.

Esempio 2.3.3 Sia

$$\mathbf{A} = \begin{pmatrix} 10^{-4} & 1 \\ 1 & 1 \end{pmatrix}.$$

Un calcolo diretto mostra che $\kappa(\mathbf{A}) \approx 4$, e quindi la matrice è ben condizionata. L'eliminazione di Gauss determina la matrice

$$\mathbf{U} = \begin{pmatrix} 10^{-4} & 1 \\ 0 & 1 - 10^4 \end{pmatrix},$$

il cui numero di condizionamento è $\kappa(\mathbf{U}) \approx 10^8$, cioè la matrice $\mathbf{U}$ è molto mal condizionata. Siccome $\mathbf{A} = \mathbf{LU}$, sarebbe auspicabile che questa relazione si riflettesse anche sui numeri di condizionamento, cioè che si avesse qualcosa del tipo $\kappa(\mathbf{A}) \approx \kappa(\mathbf{L}), \kappa(\mathbf{U})$, cosa che in questo caso non avviene.

Per poter applicare l'eliminazione di Gauss ad una classe di matrici molto più ampia, e per migliorarne le proprietà di condizionamento, è necessario introdurre operazioni di permutazione di righe e/o colonne della matrice di partenza.

Definizione 2.3.4 (Matrice di permutazione) *Si dice matrice di permutazione una matrice ottenuta permutando le righe o le colonne della matrice identità.*

Esempio 2.3.5 Data la matrice identità $\mathbf{I} \in \mathbb{R}^{3\times 3}$, esempi di matrice di permutazione sono dati dalle matrici

$$\mathbf{\Pi} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad \mathbf{\Pi} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}.$$

Il seguente risultato mostra che è sempre possibile determinare una fattorizzazione LU di una matrice permutata di $\mathbf{A}$.

Teorema 2.3.6 *Sia $\mathbf{A} \in \mathbb{R}^{n\times n}$. Allora esiste una matrice di permutazione $\mathbf{\Pi}$ per cui si può ottenere la fattorizzazione LU di $\mathbf{\Pi A}$, cioè $\mathbf{\Pi A} = \mathbf{LU}$.*

Dimostrazione. La dimostrazione procede per induzione su n.

Sia $n = 1$. Come nella dimostrazione del Teorema 2.3.2, si ha $\mathbf{L} = 1$ e $\mathbf{U} = \mathbf{A}_{11}$, da cui $\mathbf{\Pi} = 1$.

Sia ora $n = k > 1$, e si hanno due possibilità: i) La matrice $\mathbf{A}$ ha tutta la prima colonna nulla, cioè è della forma

$$\mathbf{A} = \begin{pmatrix} 0 & c^T \\ \underline{0} & \mathbf{A}_{k-1} \end{pmatrix},$$

dove $\underline{0} \in \mathbb{R}^{k-1}$ indica il vettore nullo e $\mathbf{A}_{k-1}$ è la sottomatrice $(k-1) \times (k-1)$ di $\mathbf{A}$. Allora, per ipotesi induttiva, esistono $\mathbf{\Pi}_{k-1}, \mathbf{L}_{k-1}$ e $\mathbf{U}_{k-1}$ tali che

$$\mathbf{\Pi}_{k-1}\mathbf{A}_{k-1} = \mathbf{L}_{k-1}\mathbf{U}_{k-1}.$$

Quindi

$$\underbrace{\begin{pmatrix} 1 & \underline{0}^T \\ \underline{0} & \mathbf{\Pi}_{k-1} \end{pmatrix}}_{\mathbf{\Pi}} \mathbf{A} = \begin{pmatrix} 0 & c^T \\ \underline{0} & \mathbf{\Pi}_{k-1}\mathbf{A}_{k-1} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & \\ \underline{0} & \mathbf{L}_{k-1} \end{pmatrix}}_{\mathbf{L}} \underbrace{\begin{pmatrix} 0 & c^T \\ & \mathbf{U}_{k-1} \end{pmatrix}}_{\mathbf{U}}.$$

ii) La seconda possibilità è che non tutta la prima colonna di $\mathbf{A}$ sia nulla. Sia quindi i tale che $\alpha := \mathbf{A}_{i1} \neq 0$ e sia $\mathbf{\Pi}'$ la matrice di permutazione definita permutando la prima e la i-esima riga della matrice identità. Quindi

$$\mathbf{\Pi}'\mathbf{A} = \begin{pmatrix} \alpha & c^T \\ d & \mathbf{A}_{k-1} \end{pmatrix} = \begin{pmatrix} 1 & \underline{0}^T \\ g & \mathbf{I}_{k-1} \end{pmatrix} \begin{pmatrix} \alpha & c^T \\ \underline{0} & \mathbf{B}_{k-1} \end{pmatrix},$$

dove $c, d, g, \underline{0} \in \mathbb{R}^{k-1}$ e $\mathbf{B}_{k-1} \in \mathbb{R}^{(k-1)\times(k-1)}$, e g e $\mathbf{B}_{k-1}$ sono da determinare. Perchè la fattorizzazione sia corretta, dev'essere (calcolando il prodotto esplicitamente), $g\alpha = d$ e $gc^T + \mathbf{B}_{k-1} = \mathbf{A}_{k-1}$, da cui segue

$$g = \frac{1}{\alpha}d, \qquad \mathbf{B}_{k-1} = \mathbf{A}_{k-1} - \frac{1}{\alpha}dc^T.$$

Per ipotesi induttiva, essendo $\mathbf{B}_{k-1}$ di dimensioni $(k-1) \times (k-1)$, esiste $\mathbf{\Pi}_{k-1}$ tale che

$$\mathbf{\Pi}_{k-1}\mathbf{B}_{k-1} = \mathbf{L}_{k-1}\mathbf{U}_{k-1} \;\Rightarrow\; \mathbf{B}_{k-1} = \mathbf{\Pi}_{k-1}^T\mathbf{L}_{k-1}\mathbf{U}_{k-1},$$

($\mathbf{\Pi}_{k-1}$ è una matrice ortogonale, quindi $\mathbf{\Pi}_{k-1}^{-1} = \mathbf{\Pi}_{k-1}^T$). Si ha quindi

$$\begin{aligned}
\mathbf{\Pi}'\mathbf{A} &= \begin{pmatrix} 1 & \underline{0}^T \\ g & \mathbf{I}_{k-1} \end{pmatrix} \begin{pmatrix} 1 & \underline{0}^T \\ \underline{0} & \mathbf{\Pi}_{k-1}^T\mathbf{L}_{k-1} \end{pmatrix} \begin{pmatrix} \alpha & c^T \\ \underline{0} & \mathbf{U}_{k-1} \end{pmatrix} \\
&= \begin{pmatrix} 1 & \underline{0}^T \\ g & \mathbf{\Pi}_{k-1}^T\mathbf{L}_{k-1} \end{pmatrix} \begin{pmatrix} \alpha & c^T \\ \underline{0} & \mathbf{U}_{k-1} \end{pmatrix} \\
&= \begin{pmatrix} 1 & \underline{0}^T \\ \underline{0} & \mathbf{\Pi}_{k-1}^T \end{pmatrix} \begin{pmatrix} 1 & \underline{0}^T \\ \mathbf{\Pi}_{k-1}g & \mathbf{L}_{k-1} \end{pmatrix} \begin{pmatrix} \alpha & c^T \\ \underline{0} & \mathbf{U}_{k-1} \end{pmatrix},
\end{aligned}$$

da cui

$$\underbrace{\begin{pmatrix} 1 & \underline{0}^T \\ \underline{0} & \mathbf{\Pi}_{k-1} \end{pmatrix}}_{\mathbf{\Pi}_k} \mathbf{\Pi}'\mathbf{A} = \underbrace{\begin{pmatrix} 1 & \underline{0}^T \\ \mathbf{\Pi}_{k-1}g & \mathbf{L}_{k-1} \end{pmatrix}}_{\mathbf{L}_k} \underbrace{\begin{pmatrix} \alpha & c^T \\ \underline{0} & \mathbf{U}_{k-1} \end{pmatrix}}_{\mathbf{U}_k}. \quad \square$$

La matrice $\mathbf{\Pi}$ non è unica (vedi Esempio 2.3.7). La scelta delle permutazioni da applicare è guidata dalla ricerca dell'elemento più grande della colonna, per assicurare la migliore stabilità (si veda la sezione 2.3.3). Il coefficiente così trovato, che verrà posizionato sulla diagonale della matrice permutata, si chiama *pivot*, e la procedura prende il nome di *pivoting parziale*. Se la ricerca dell'elemento più grande non si limita alla colonna k-esima in esame, ma spazia tra tutte le colonne e righe dalla k-esima alla n-esima, allora si parla di *pivoting completo* (o totale). Nel caso il pivot sia determinato in una colonna diversa dalla k-esima, allora è necessario scambiare le due colonne della matrice, con conseguente permutazione della soluzione finale.

Il pivoting parziale comporta un costo addizionale di circa n^2 *flops* da aggiungere al costo dell'eliminazione di Gauss standard ($2n^3/3+\mathcal{O}(n^2)$). Il pivoting completo ha un costo molto più elevato, cioè di $2n^3/3$, che comporta un considerevole aggravio del costo computazionale dell'eliminazione di Gauss.

Esempio 2.3.7 Sia

$$\mathbf{A} = \left(\begin{array}{cc|c} 1 & 2 & -1 \\ -1 & -2 & 0 \\ \hline 1 & 1 & 2 \end{array}\right).$$

La sottomatrice principale 2×2 è singolare, per cui la fattorizzazione LU si bloccherà dopo la prima iterazione con uno zero nel secondo elemento diagonale. D'altra parte, definendo

$$\mathbf{\Pi} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad \text{si ottiene} \quad \mathbf{\Pi A} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 2 & -1 \\ 0 & -1 & 3 \\ 0 & 0 & -1 \end{pmatrix},$$

oppure,

$$\mathbf{\Pi}' = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad \text{con cui} \quad \mathbf{\Pi}'\mathbf{A} = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 1 & -1 & 1 \end{pmatrix}\begin{pmatrix} 1 & 1 & 2 \\ 0 & -1 & 2 \\ 0 & 0 & -1 \end{pmatrix}.$$

Esempio 2.3.8 Richiamando l'Esempio 2.3.3, sia

$$\mathbf{A} = \begin{pmatrix} 10^{-4} & 1 \\ 1 & 1 \end{pmatrix},$$

e un calcolo diretto mostra che $\kappa(\mathbf{A}) \approx 4$. Applicando l'eliminazione di Gauss con pivoting, otteniamo la matrice

$$\mathbf{U} = \begin{pmatrix} 1 & 1 \\ 0 & 1-10^{-4} \end{pmatrix},$$

il cui numero di condizionamento $\kappa(\mathbf{U}) \approx 4$. Abbiamo quindi mostrato come la strategia di pivoting riesca a ridurre gli errori di arrotondamento: nell'Esempio 2.3.3, applicando l'eliminazione di Gauss senza pivoting, ottenevamo una matrice $\mathbf{U}$ con un grande numero di condizionamento.

2.3.2 Fattorizzazione di Cholesky

Definizione 2.3.9 (Matrice simmetrica definita positiva, s.d.p.) *Sia $\mathbf{A} \in \mathbb{R}^{n\times n}$ una matrice simmetrica. Si dice che $\mathbf{A}$ è definita positiva se $x^T\mathbf{A}x > 0$ per ogni $0 \neq x \in \mathbb{R}^n$.*

La notazione usata per indicare una matrice simmetrica e definita positiva è anche $\mathbf{A} > 0$.

Lemma 2.3.10 *Una matrice simmetrica $\mathbf{A}$ è definita positiva se e solo se i determinanti di tutte le sue sottomatrici principali dominanti ("di testa") sono positivi.*

Se $\mathbf{A}$ è s.d.p., allora i suoi elementi diagonali siano tutti positivi: $(\mathbf{A})_{ii} > 0\ \forall i$, dato che $\mathbf{A}_{ii} = e_i^T\mathbf{A}e_i$ dove e_i è lo i-esimo elemento della base canonica. Con un ragionamento analogo si dimostra che $\mathbf{A}$ è definita positiva se e solo se i suoi autovalori (tutti reali) sono positivi.

Nel caso in cui la matrice dei coefficienti $\mathbf{A}$ di (2.15) sia simmetrica e definita positiva, la fattorizzazione LU si semplifica, dando luogo ad una fattorizzazione simmetrica $\mathbf{A} = \widehat{\mathbf{L}}\widehat{\mathbf{L}}^T$ con $\widehat{\mathbf{L}}$ triangolare inferiore (con diagonale non necessariamente unitaria), detta fattorizzazione di Cholesky.

Teorema 2.3.11 [**Fattorizzazione di Cholesky**] *Sia* $\mathbf{A} \in \mathbb{R}^{n\times n}$ *simmetrica e definita positiva. Allora esiste ed è unica la fattorizzazione* $\mathbf{A} = \widehat{\mathbf{L}}\widehat{\mathbf{L}}^T$, *dove* $\widehat{\mathbf{L}}$ *è triangolare inferiore, non singolare con* $\widehat{\mathbf{L}}_{ii} > 0$.

Dimostrazione. Il risultato del Lemma 2.3.10 assicura che esiste ed è unica la fattorizzazione LU di $\mathbf{A}$, $\mathbf{A} = \mathbf{LU}$. Sia $\mathbf{D}$ la matrice diagonale avente sulla diagonale gli elementi diagonali della matrice $\mathbf{U}$, cioè $\mathbf{D} = \mathrm{diag}(\mathbf{U})$. Quindi

$$\mathbf{A} = \mathbf{LD}(\mathbf{D}^{-1}\mathbf{U}) = \mathbf{LDR}, \quad \text{dove} \quad \mathbf{R} = \mathbf{D}^{-1}\mathbf{U}.$$

Si noti che $\mathbf{R}$ così definita è una matrice triangolare superiore avente diagonale unitaria. Essendo $\mathbf{A}$ simmetrica, vale

$$\mathbf{LDR} = \mathbf{A} = \mathbf{A}^T = (\mathbf{LDR})^T = \mathbf{R}^T\mathbf{DL}^T,$$

e per l'unicità della decomposizione deve valere $\mathbf{R}^T = \mathbf{L}$ e $\mathbf{DR} = \mathbf{DL}^T$, quindi dev'essere $\mathbf{R} = \mathbf{L}^T$, da cui $\mathbf{A} = \mathbf{LDL}^T$, con $\mathbf{D}$ diagonale. Mostriamo infine che $\mathbf{D} > 0$. Sia $x \neq 0$, allora per $\mathbf{A} > 0$ vale $0 < x^T\mathbf{A}x = x^T\mathbf{LDL}^Tx = y^T\mathbf{D}y$, dove è stato posto $y = \mathbf{L}^Tx \neq 0$. Dunque

$$\mathbf{A} = \mathbf{LDL}^T = \mathbf{LD}^{1/2}\mathbf{D}^{1/2}\mathbf{L}^T = \widehat{\mathbf{L}}\widehat{\mathbf{L}}^T, \quad \text{con} \quad \widehat{\mathbf{L}} := \mathbf{LD}^{1/2}. \quad \square$$

In effetti il Teorema 2.3.11 è una doppia implicazione: se è possibile determinare $\widehat{\mathbf{L}}$ non singolare tale che $\mathbf{A} = \widehat{\mathbf{L}}\widehat{\mathbf{L}}^T$, allora la matrice simmetrica $\mathbf{A}$ è anche definita positiva. Infatti, per $x \neq 0$ si ha $x^T\mathbf{A}x = x^T\widehat{\mathbf{L}}\widehat{\mathbf{L}}^Tx = \|\widehat{\mathbf{L}}^Tx\|^2 \geq 0$. Siccome $\widehat{\mathbf{L}}$ è non singolare, segue che $\|\widehat{\mathbf{L}}^Tx\|^2 > 0$, da cui segue che $\mathbf{A} > 0$.

Il fattore $\widehat{\mathbf{L}}$ può essere determinato in modo costruttivo, da cui segue l'effettivo algoritmo. Nel seguito riportiamo il primo passo: sia $n = 1$, allora $\widehat{\mathbf{L}} \equiv \widehat{\mathbf{L}}_{11} = \sqrt{\mathbf{A}_{11}}$. Sia allora $n > 1$. Scriviamo

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_{11} & v^T \\ v & \mathbf{A}_2 \end{pmatrix} = \begin{pmatrix} \sqrt{\mathbf{A}_{11}} & 0 \\ \frac{1}{\sqrt{\mathbf{A}_{11}}}v & \mathbf{I} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \tilde{\mathbf{A}}_2 \end{pmatrix} \begin{pmatrix} \sqrt{\mathbf{A}_{11}} & \frac{1}{\sqrt{\mathbf{A}_{11}}}v^T \\ 0 & \mathbf{I} \end{pmatrix},$$

con $v \in \mathbb{R}^{n-1}$, e $\tilde{\mathbf{A}}_2 = \mathbf{A}_2 - \dfrac{1}{\mathbf{A}_{11}}vv^T \in \mathbb{R}^{(n-1)\times(n-1)}$. Si noti che $\tilde{\mathbf{A}}_2$ è ancora s.d.p.. Infatti, scrivendo il prodotto sopra come $\mathbf{A} = \mathbf{LDL}^T$ con $\mathbf{D} = \mathrm{diag}(1, \tilde{\mathbf{A}}_2)$, si ha che per $x \neq 0$ e $y = \mathbf{L}^Tx \neq 0$ vale $0 < x^T\mathbf{A}x = (x^T\mathbf{L})\mathbf{D}(\mathbf{L}^Tx) = y^T\mathbf{D}y$, per cui $\mathbf{D}$ è definita positiva, da cui segue che anche $\tilde{\mathbf{A}}_2$ è definita positiva (si noti che $\mathbf{L}$ è non singolare perchè ha diagonale con elementi tutti non nulli). Per ipotesi induttiva si può quindi scrivere $\tilde{\mathbf{A}}_2 = \tilde{\mathbf{L}}\tilde{\mathbf{L}}^T$. Si ha dunque che

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_{11} & v^T \\ v & \mathbf{A}_2 \end{pmatrix} = \begin{pmatrix} \sqrt{\mathbf{A}_{11}} & 0 \\ \frac{1}{\sqrt{\mathbf{A}_{11}}}v & \mathbf{I} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \tilde{\mathbf{L}} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & \tilde{\mathbf{L}}^T \end{pmatrix} \begin{pmatrix} \sqrt{\mathbf{A}_{11}} & \frac{1}{\sqrt{\mathbf{A}_{11}}}v^T \\ 0 & \mathbf{I} \end{pmatrix}.$$

La matrice $\widehat{\mathbf{L}}$ cercata è

$$\widehat{\mathbf{L}} \equiv \begin{pmatrix} \sqrt{\mathbf{A}_{11}} & 0 \\ \frac{1}{\sqrt{\mathbf{A}_{11}}}v & \tilde{\mathbf{L}} \end{pmatrix}.$$

Usando la procedura appena descritta, si può ricavare il seguente algoritmo di Cholesky,

Algoritmo di Cholesky

For $j = 1, \ldots, n$,
  $\widehat{\mathbf{L}}_{jj} = (\mathbf{A}_{jj} - \sum_{k=1}^{j-1} \widehat{\mathbf{L}}_{jk}^2)^{1/2}$,
  For $i = j+1, \ldots, n$,
    $\widehat{\mathbf{L}}_{ij} = (\mathbf{A}_{ij} - \sum_{k=1}^{j-1} \widehat{\mathbf{L}}_{ik}\widehat{\mathbf{L}}_{jk})/\widehat{\mathbf{L}}_{jj}$,
  end

end

Il costo computazionale dell'algoritmo è di $\frac{1}{3}n^3 + \mathcal{O}(n^2)$ operazioni floating point. Questo costo va confrontato con quello della fattorizzazione LU: la fattorizzazione di Cholesky costa la metà sia in termini di operazioni, che in termini di memoria, in quanto viene memorizzata una sola matrice triangolare. L'algoritmo di Cholesky non richiede alcuna strategia di pivoting, in quanto si dimostra che la fattorizzazione risulta essere già stabile.

Se la matrice $\mathbf{A}$ è simmetrica ma non definita positiva, l'algoritmo si interrompe, in quanto non è possibile determinare $\sqrt{\widehat{\mathbf{L}}_{ii}}$ per qualche $i = 1, \ldots, n-1$.

L'algoritmo di Cholesky può essere usato come procedura poco costosa per verificare se una matrice simmetrica $\mathbf{A}$ è definita positiva: se la fattorizzazione si blocca per un elemento diagonale non positivo, significa che $\mathbf{A}$ non è definita positiva.

Esiste una fattorizzazione di "tipo Cholesky" anche per matrici simmetriche ma non definite positive: $\Pi\mathbf{A}\Pi^T = \mathbf{L}\mathbf{D}\mathbf{L}^T$, dove $\mathbf{D}$ in generale è una matrice diagonale a blocchi, con blocchi simmetrici 1×1 o 2×2, ma non necessariamente definiti positivi.

2.3.3 Analisi dell'errore

In questa sezione analizziamo le proprietà di stabilità della fattorizzazione LU.

Teorema 2.3.12 *Sia $\mathbf{A} \in \mathbb{R}^{n\times n}$ e siano $\tilde{\mathbf{L}} = \mathbf{L} + \delta\mathbf{L}$, e $\tilde{\mathbf{U}} = \mathbf{U} + \delta\mathbf{U}$, i fattori della decomposizione LU di $\mathbf{A}$ effettivamente calcolati. Allora*

$$\tilde{\mathbf{L}}\tilde{\mathbf{U}} = \mathbf{A} + \mathbf{E}, \quad \text{con} \quad \|\mathbf{E}\|_F \leq 2n\mathtt{u}(\|\mathbf{A}\|_F + \|\tilde{\mathbf{L}}\|_F \, \|\tilde{\mathbf{U}}\|_F) + \mathcal{O}(\mathtt{u}^2 n^2).$$

Il risultato del Teorema, che non dimostriamo, mostra che i fattori calcolati rappresentano la fattorizzazione esatta di una matrice "vicina" ad $\mathbf{A}$, cioè $\mathbf{A} + \mathbf{E}$; viene inoltre fornita una stima di tale vicinanza, in termini della norma di $\mathbf{E}$. La fattorizzazione è quindi stabile, nel senso dell'analisi all'indietro, se $\|\mathbf{E}\|_F$ è piccola rispetto a $\|\mathbf{A}\|_F$.

Teorema 2.3.13 *Siano $\tilde{\mathbf{L}}$ e $\tilde{\mathbf{U}}$ come nel Teorema 2.3.12, e sia $\tilde{x}$ la soluzione del sistema lineare $\tilde{\mathbf{L}}\tilde{\mathbf{U}}\tilde{x} = b$. Allora $\tilde{x}$ è anche soluzione del sistema*

$$(\mathbf{A} + \delta\mathbf{A})\tilde{x} = b, \quad \text{dove} \quad \|\delta\mathbf{A}\|_F \leq 4n\mathtt{u}(\|\mathbf{A}\|_F + \|\tilde{\mathbf{L}}\|_F \, \|\tilde{\mathbf{U}}\|_F) + \mathcal{O}(\mathtt{u}^2 n^2).$$

Dimostrazione. La soluzione $\tilde{x}$ è ottenuta mediante la risoluzione a cascata di $\tilde{\mathbf{L}}y = b$, e $\tilde{\mathbf{U}}x = \tilde{y}$. Grazie a (2.12), $\tilde{y}$ risolve il sistema

$$(\tilde{\mathbf{L}} + \mathbf{E}_{\tilde{\mathbf{L}}})\tilde{y} = b, \quad \text{dove} \quad \|\mathbf{E}_{\tilde{\mathbf{L}}}\|_F \leq n\mathtt{u}\|\tilde{\mathbf{L}}\|_F, \tag{2.22}$$

e $\tilde{x}$ risolve il sistema

$$(\tilde{\mathbf{U}} + \mathbf{E}_{\tilde{\mathbf{U}}})\tilde{x} = \tilde{y}, \quad \text{dove} \quad \|\mathbf{E}_{\tilde{\mathbf{U}}}\|_F \leq n\mathtt{u}\|\tilde{\mathbf{U}}\|_F. \tag{2.23}$$

Combinando (2.22) e (2.23) si ottiene $(\tilde{\mathbf{L}} + \mathbf{E}_{\tilde{\mathbf{L}}})(\tilde{\mathbf{U}} + \mathbf{E}_{\tilde{\mathbf{U}}})\tilde{x} = b$. Svolgendo il prodotto matriciale si ottiene

$$(\tilde{\mathbf{L}}\tilde{\mathbf{U}} + \tilde{\mathbf{L}}\mathbf{E}_{\tilde{\mathbf{U}}} + \mathbf{E}_{\tilde{\mathbf{L}}}\tilde{\mathbf{U}} + \mathbf{E}_{\tilde{\mathbf{L}}}\mathbf{E}_{\tilde{\mathbf{U}}})\tilde{x} = b.$$

Dal Teorema 2.3.12, si ha che $\tilde{\mathbf{L}}\tilde{\mathbf{U}} = \mathbf{A} + \mathbf{E}$ e, definendo

$$\delta\mathbf{A} := \mathbf{E} + \tilde{\mathbf{L}}\mathbf{E}_{\tilde{\mathbf{U}}} + \mathbf{E}_{\tilde{\mathbf{L}}}\tilde{\mathbf{U}} + \mathbf{E}_{\tilde{\mathbf{L}}}\mathbf{E}_{\tilde{\mathbf{U}}},$$

si ottiene che $\tilde{x}$ è soluzione del sistema $(\mathbf{A}+\delta\mathbf{A})\tilde{x}=b$, con

$$\begin{aligned}\|\delta\mathbf{A}\|_F &\leq \|\mathbf{E}\|_F + \|\tilde{\mathbf{L}}\|_F\,\|\mathbf{E}_{\tilde{\mathbf{U}}}\|_F + \|\mathbf{E}_{\tilde{\mathbf{L}}}\|_F\,\|\tilde{\mathbf{U}}\|_F + \|\mathbf{E}_{\tilde{\mathbf{L}}}\|_F\,\|\mathbf{E}_{\tilde{\mathbf{U}}}\|_F \\ &\leq \|\mathbf{E}\|_F + n\mathtt{u}\|\tilde{\mathbf{L}}\|_F\,\|\tilde{\mathbf{U}}\|_F + n\mathtt{u}\|\tilde{\mathbf{L}}\|_F\,\|\tilde{\mathbf{U}}\|_F + \mathcal{O}(\mathtt{u}^2n^2) \\ &\leq 2n\mathtt{u}(\|\mathbf{A}\| + \|\tilde{\mathbf{L}}\|_F\,\|\tilde{\mathbf{U}}\|_F) + 2n\mathtt{u}\|\tilde{\mathbf{L}}\|_F\,\|\tilde{\mathbf{U}}\|_F + \mathcal{O}(\mathtt{u}^2n^2) \\ &\leq 4n\mathtt{u}(\|\mathbf{A}\| + \|\tilde{\mathbf{L}}\|_F\,\|\tilde{\mathbf{U}}\|_F) + \mathcal{O}(\mathtt{u}^2n^2). \quad \square\end{aligned}$$

Il metodo di eliminazione di Gauss è stabile all'indietro se $\frac{\|\delta\mathbf{A}\|_F}{\|\mathbf{A}\|_F} = \mathcal{O}(\mathtt{u})$, cioè se

$$4n\mathtt{u}(\|\mathbf{A}\| + \|\tilde{\mathbf{L}}\|_F\,\|\tilde{\mathbf{U}}\|_F) = \mathcal{O}(\mathtt{u})\|\mathbf{A}\|_F. \tag{2.24}$$

Se il metodo viene applicato con la procedura di pivoting, si ha che $|\tilde{\mathbf{L}}_{ij}| \approx 1$, e quindi $\|\tilde{\mathbf{L}}\|_F \leq n$. Perchè (2.24) sia soddisfatta, è dunque sufficiente controllare la crescita di $\|\tilde{\mathbf{U}}\|_F$.

Sia

$$g_{pp} := \frac{\|\tilde{\mathbf{U}}\|_{\max}}{\|\mathbf{A}\|_{\max}}$$

il fattore di crescita della fattorizzazione LU con pivoting parziale, dove $\|\mathbf{A}\|_{\max} = \max_{i,j}|\mathbf{A}_{ij}|$.

Proposizione 2.3.14 *Il fattore di crescita per il metodo di elimininazione di Gauss con pivot parziale verifica* $g_{pp} \leq 2^{n-1}$.

Dim. Alla prima iterazione dell'eliminazione di Gauss con pivot parziale si ha $|a^{(2)}_{i,j}| = |a^{(1)}_{i,j} - m_{i,1}a^{(1)}_{1,j}| \leq |a^{(1)}_{i,j}| + |m_{i,1}|\,|a^{(1)}_{1,j}| \leq 2\|A\|_{\max}$, dove abbiamo usato il fatto che con il pivot parziale, $|m_{i,1}| \leq 1$ per ogni i. Quindi ad ogni passo si ha un possibile raddoppio del valore degli elementi rimanenti. Dopo $n-1$ passi si avrà quindi che l'ultimo elemento può essere in effetti 2^{n-1} volte l'elemento più grande di A. $\square$

Tale stima è estremamente pessimistica, anche se è raggiungibile per certe matrici considerate "patologiche".

Esempio. Un esempio che mostra che tale stima è raggiungibile è dato dalla matrice

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 & 1 \\ -1 & 1 & 0 & 0 & \dots & 0 & 1 \\ -1 & -1 & 1 & 0 & \dots & 0 & 1 \\ -1 & -1 & -1 & 1 & 0\dots & 0 & 1 \\ \vdots & \ddots & \ddots & \ddots\ddots & \ddots & \ddots & \vdots \\ -1 & -1 & -1 & -1\dots & -1 & -1 & 1 \end{bmatrix} \in \mathbb{R}^{n\times n}.$$

Il metodo di eliminazione di Gauss non fa pivoting, e determina $U_{n,n} = 2^{n-1}$.

Con questa stima, usando la stima[2] $\|\tilde{\mathbf{U}}\|_F \leq n g_{pp}\|\mathbf{A}\|_F$, si ottiene $\|\delta\mathbf{A}\|_F \leq (4n\mathtt{u}+4n^3\mathtt{u}g_{pp})\|\mathbf{A}\|_F$, da cui

$$\|\delta\mathbf{A}\|_F = \mathcal{O}(4n^3\mathtt{u}g_{pp}\|\mathbf{A}\|_F).$$

2.4 Sistemi con matrice a banda

In questa sezione vengono trattati sistemi la cui matrice dei coefficienti ha una struttura "a banda", che permette una implementazione efficiente della fattorizzazione LU.

[2]Si ha $\|U\|_F^2 = |u_{ij}|^2_{\max}\sum_{i,j} u_{ij}^2/|u_{ij}|^2_{\max} \leq \|U\|^2_{\max}n^2$ e $\|A\|_{\max} \leq \|A\|_F$, da cui $g_{pp} \geq \|U\|_F/(\|A\|_F n)$.

Definizione 2.4.1 (Matrice a banda) *Una matrice $\mathbf{A} \in \mathbb{R}^{n\times m}$ si dice a banda con banda inferiore b_L e banda superiore b_U se*

$$\mathbf{A}_{ij} = 0, \quad \textit{per } i > j + b_L \textit{ e } i < j - b_U.$$

Una matrice $\mathbf{A}$ a banda risulta quindi avere la forma

$$\mathbf{A} = \begin{pmatrix} \times & \circ & 0 & \dots & \dots & 0 \\ * & \times & \circ & \ddots & & \vdots \\ * & \ddots & \times & \circ & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \circ \\ 0 & \dots & 0 & * & * & \times \end{pmatrix},$$

dove gli elementi "*" costituiscono la banda inferiore (nell'esempio $b_L = 2$) e gli elementi $\circ$ la banda superiore (nell'esempio $b_U = 1$) di $\mathbf{A}$. La seguente proposizione, che non dimostriamo, descrive il vantaggio - computazionale e di memoria - di avere a disposizione una matrice a banda.

Proposizione 2.4.2 *Sia $\mathbf{A}$ una matrice a banda con banda inferiore b_L e banda superiore b_U, e supponiamo che esista la sua fattorizzazione LU (senza pivoting), $\mathbf{A} = \mathbf{LU}$. Allora $\mathbf{L}$ e $\mathbf{U}$ sono matrici a banda di ampiezza rispettivamente b_L e b_U. Inoltre $\mathbf{L}$ e $\mathbf{U}$ possono essere costruite con $\mathcal{O}(nb_Ub_L)$ operazioni floating points.*

2.4.1 Caso tridiagonale. L'algoritmo di Thomas

Un caso particolarmente interessante è quello in cui $\mathbf{A}$ è tridiagonale, cioè con $b_U = b_L = 1$. Matrici di questo tipo si ottengono in svariate applicazioni, per esempio durante la discretizzazione di certi problemi differenziali ai limiti uno-dimensionali, e nella costruzione di alcune *spline* nell'approssimazione di funzioni.

Esempio 2.4.3 *A titolo di semplice esempio, mostriamo che la risoluzione numerica mediante discretizzazione con differenze finite del problema differenziale uno-dimensionale $u''(x) = f(x)$ con $x \in (0,1)$ e $u(0) = u(1) = 0$, determina un sistema lineare con matrice dei coefficienti tridiagonale. Infatti, consideriamo una suddivisione equispaziata $0 = x_0 < x_1 < \cdots < x_{n+1} = 1$ dell'intervallo $[0,1]$, con $h = x_j - x_{j-1}$. Poniamo $u_j := u(x_j)$. Allora per ogni nodo interno si ha*

$$u_{j+1} = u_j + hu'(x_j) + \frac{1}{2}h^2u''(x_j) + \mathcal{O}(h^3), \qquad u_{j-1} = u_j - hu'(x_j) + \frac{1}{2}h^2u''(x_j) + \mathcal{O}(h^3).$$

Sommando le due relazioni si ottiene la stima $u''(x_j) \approx \frac{1}{h^2}(u_{j-1} - 2u_j + u_{j+1})$ per $h \to 0$. In ogni punto interno, l'equazione differenziale è quindi approssimata come $\frac{1}{h^2}(u_{j-1} - 2u_j + u_{j+1}) = f(x_j)$. Per tutti i nodi interni, e tenendo conto che $u(x_0) = 0 = u(x_n)$, si ha dunque

$$\begin{array}{rcl} \frac{1}{h^2}(-2u_1 + u_2) & = & f(x_1), \\ \frac{1}{h^2}(u_1 - 2u_2 + u_3) & = & f(x_2), \\ \vdots & & \vdots \\ \frac{1}{h^2}(u_{n-1} - 2u_n) & = & f(x_n), \end{array} \quad \text{cioe}' \quad \frac{1}{h^2}\begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & & -2 & 1 \\ & & & 1 & -2 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \end{bmatrix} \quad (2.25)$$

Il sistema a destra, $\mathbf{A}\underline{u} = \underline{f}$, ha matrice tridiagonale e simmetrica, ed il termine noto contiene i valori della funzione f nei nodi interni. Le componenti del vettore soluzione del sistema lineare sono le approssimazioni nei nodi interni dei valori della funzione u soluzione del problema differenziale.

È dato il sistema con matrice tridiagonale

$$\mathbf{A}x = f, \quad \text{con} \quad \mathbf{A} = \begin{pmatrix} a_1 & c_1 & 0 & \dots & 0 \\ b_2 & a_2 & c_2 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & c_{n-1} \\ 0 & \dots & 0 & b_n & a_n \end{pmatrix}. \tag{2.26}$$

L'algoritmo per l'implementazione della fattorizzazione LU di $\mathbf{A}$ può essere estremamente semplificato e reso meno costoso, e prende il nome di *algoritmo di Thomas*. Dal teorema precedente segue che le matrici $\mathbf{L}$ e $\mathbf{U}$ sono bidiagonali, ed hanno la forma

$$\mathbf{L} = \begin{pmatrix} 1 & & & & \\ \beta_2 & 1 & & & \\ & \beta_3 & \ddots & & \\ & & \ddots & \ddots & \\ & & & \beta_n & 1 \end{pmatrix}, \qquad \mathbf{U} = \begin{pmatrix} \alpha_1 & \gamma_1 & & & \\ & \alpha_2 & \ddots & & \\ & & \ddots & \ddots & \\ & & & \alpha_{n-1} & \gamma_{n-1} \\ & & & & \alpha_n \end{pmatrix}.$$

Moltiplicando in modo esplicito $\mathbf{L}$ ed $\mathbf{U}$ e confrontando i valori con quelli di $\mathbf{A}$, si ottiene $\gamma_i = c_i$, $i = 1, \dots, n-1$, e

$$\alpha_1 = a_1, \quad \beta_i = \frac{b_i}{\alpha_{i-1}}, \quad \alpha_i = a_i - \beta_i c_{i-1}, \quad i = 2, \dots, n.$$

Il numero di operazioni floating point necessarie per questa iterazione è dato da $n-1$ divisioni per generare i coefficienti β_i, e $2(n-1)$ operazioni per generare α_i. In totale l'algoritmo di Thomas richiede $(n-1) + 2(n-1) = 3n-3$ flops, da confrontare con i $\frac{2}{3}n^3 + \mathcal{O}(n^2)$ flops richiesti per l'eliminazione di Gauss con matrice generica. Il costo per matrici tridiagonali è quindi inferiore di due ordini di grandezza.

Una volta determinati i coefficienti di $\mathbf{L}$ ed $\mathbf{U}$, la determinazione della soluzione x richiede la risoluzione a cascata dei due sistemi

$$\mathbf{L}y = f, \qquad \mathbf{U}x = y. \tag{2.27}$$

La struttura bidiagonale di $\mathbf{L}$ e $\mathbf{U}$ può essere sfruttata anche nella risoluzione di questi sistemi. Infatti per $\mathbf{L}y = f$ si ha

$$y_1 = f_1, \quad y_i = f_i - \beta_i y_{i-1}, \quad i = 2, \dots, n,$$

al costo di $2(n-1)$ flops. Applicando il metodo di sostituzione all'indietro al secondo sistema, $\mathbf{U}x = y$, si ha

$$x_n = \frac{y_n}{\alpha_n}, \quad x_i = \frac{1}{\alpha_i}(y_i - c_i x_{i+1}), \quad i = n-1, \dots, 1,$$

al costo di $3(n-1) + 1\, flops$. Il costo della risoluzione dei sistemi bidiagonali (2.27) è quindi di $2(n-1) + 3(n-1) + 1 = 5n - 4\, flops$. Ciò significa che il costo totale della risoluzione del sistema tridiagonale (2.26), che tiene conto sia della fattorizzazione della matrice dei coefficienti sia della risoluzione dei sistemi bidiagonali (2.27), è di

$$3n - 3 + 5n - 4 = 8n + \mathcal{O}(1) \text{ flops}.$$

Si noti che per la risoluzione dei due sistemi bidiagonali, non è necessario creare esplicitamente $\mathbf{L}$ e $\mathbf{U}$, ma solo memorizzare i termini $\{\beta_i\}$ e $\{\alpha_i\}$.

2.5 Applicazioni

In questa sezione sono proposte alcune applicazioni degli algoritmi e delle proprietà proposte. Verranno anche introdotti ulteriori risultati sulle proprietà delle matrici, che risulteranno utili per lo svolgimento di alcuni esercizi.

Esercizio 2.5.1 *Determinare $|det(\mathbf{A})|$ mediante la fattorizzazione LU e stimarne il costo computazionale.*

Risoluzione. *Supponiamo dapprima che sia possibile fattorizzare $\mathbf{A}$ come $\mathbf{A} = \mathbf{LU}$. Si ha $det(\mathbf{A}) = det(\mathbf{LU}) = det(\mathbf{L})det(\mathbf{U})$, dove nell'ultima uguaglianza è stata applicata la formula di Binet. Le matrici $\mathbf{L}$ e $\mathbf{U}$ sono triangolari, quindi il loro determinante è il prodotto dei loro elementi diagonali. Dato che tutti gli elementi diagonali di $\mathbf{L}$ sono uguali ad 1, si ha $det(\mathbf{L}) = 1$, mentre $det(\mathbf{U}) = \prod_{i=1}^{n} \mathbf{U}_{ii}$. Quindi $det(\mathbf{A}) = \prod_{i=1}^{n} \mathbf{U}_{ii}$. Il costo complessivo di questa operazione è di $\frac{2}{3}n^3 + n$ flops, poichè sono necessari $2n^3/3$ flops per la fattorizzazione, e altri n per la prodottoria.*

Nel caso in cui $\mathbf{\Pi A} = \mathbf{LU}$,, allora possiamo solo dire che $|det(\mathbf{A})| = |\prod_{i=1}^{n} \mathbf{U}_{ii}|$, poichè $|det(\mathbf{\Pi})| = 1$ essendo $\mathbf{\Pi}$ una matrice ortogonale.

Esercizio 2.5.2 *Sia*

$$\mathbf{A} = \begin{pmatrix} \alpha & 3\alpha \\ 1 & 1 \end{pmatrix},$$

e $\|\mathbf{A}\|_\infty = \max\limits_{i=1,\dots,n} \sum\limits_{j=1}^{n} |\mathbf{A}_{ij}|$. Determinare α in modo che $\kappa_\infty(\mathbf{A})$ sia minimo.

Risoluzione. *Si ha direttamente $\|\mathbf{A}\|_\infty = \max\{4\alpha, 2\}$. Inoltre,*

$$\mathbf{A}^{-1} = \begin{pmatrix} -\dfrac{1}{2\alpha} & \dfrac{3}{2} \\ \\ \dfrac{1}{2\alpha} & -\dfrac{1}{2} \end{pmatrix},$$

per cui

$$\|\mathbf{A}^{-1}\|_\infty = \max\left\{\frac{1}{2\alpha} + \frac{3}{2}, \frac{1}{2\alpha} + \frac{1}{2}\right\} = \frac{1}{2\alpha} + \frac{3}{2}.$$

Quindi

$$\kappa_\infty(\mathbf{A}) = \max\{4\alpha, 2\} \cdot \left(\frac{1}{2\alpha} + \frac{3}{2}\right).$$

Per $4\alpha \geq 2$, cioè $\alpha \geq \frac{1}{2}$, si ha $\kappa_\infty(\mathbf{A}) = 4\alpha\left(\frac{1}{2\alpha} + \frac{3}{2}\right) = 2 + 6\alpha \geq 2 + \frac{6}{2} = 5$.

D'altra parte, per $4\alpha < 2$, cioè per $\alpha < \frac{1}{2}$, si ha $\kappa_\infty(\mathbf{A}) = 2\left(\frac{1}{2\alpha} + \frac{3}{2}\right) = \frac{1}{\alpha} + 3 > 5$.

Per $\alpha = \dfrac{1}{2}$ si ottiene quindi il minimo, $\kappa_\infty(\mathbf{A}) = 5$.

Esercizio 2.5.3 *È dato il sistema lineare $Ax = b$, con A non singolare,*

$$A = \begin{bmatrix} S & uv^T \\ 0 & S^T \end{bmatrix} \in \mathbb{R}^{2n \times 2n},\ u, v \in \mathbb{R}^n,$$

Proponi un algoritmo completo che risolva il sistema in modo efficiente.

Risoluzione. *Scriviamo $x = [x_1; x_2]$ e $b = [b_1; b_2]$ con blocchi conformi con quelli di A. Quindi $Sx_1 + u(v^T x_2) = b_1$ e $S^T x_2 = b_2$. Da A non singolare segue che S e S^T sono non singolari. Sia $S^T = \Pi^T LU$ la fattorizzazione LU di S^T con pivoting. Quindi $x_2 = U^{-1}(L^{-1}(\Pi b_2))$. In effetti x_2 si puo' ottenere mediante eliminazione di Gauss, cosi da ottenere direttamente $L^{-1}(\Pi b_2)$, quindi il calcolo di x_2 richiede la sola risoluzione con U, supponendo di aver già calcolato la fattorizzazione. Le matrici L e Π servono per determinare x_1. Infatti, si ha $Sx_1 = b_1 - u(v^T x_2)$, da cui $x_1 = (U^T L^T \Pi)^{-1}(b_1 - u(v^T x_2))$, cioè $x_1 = \Pi^T(L^{-T}((U^{-T}(b_1 - u(v^T x_2)))))$. Quindi, per determinare x_1 occorre 1 prodotto scalare ed una daxpy per ottenere il termine noto, e poi la risoluzione di un sistema triangolare inferiore e superiore, infine la permutazione delle componenti (zero costo computazionale, se fatto mediante un vettore).*

Per il proseguo, anticipiamo alcune proprietà di autovalori e di norme.

Proposizione 2.5.4 *Sia $\mathbf{A} \in \mathbb{R}^{n\times n}$ simmetrica e definita positiva. Allora*

$$\|\mathbf{A}\|_2 = \lambda_{\max} = \max_{0\neq x\in\mathbb{R}^n} \frac{x^T\mathbf{A}x}{x^T x}.$$

Il rapporto $\frac{x^T\mathbf{A}x}{x^T x}$ si chiama rapporto (o quoziente) di Rayleigh. Il risultato vale anche per $\mathbf{A}$ simmetrica ma non necessariamente definita positiva.

Dimostrazione. Il fatto che $\|\mathbf{A}\|_2 = \lambda_{\max}$ è già stato dimostrato in precedenza. Rimane da dimostrare la seconda uguaglianza. Sia $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$ la decomposizione spettrale di $\mathbf{A}$, con $\mathbf{Q}$ ortogonale e $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$, dove $\lambda_i > 0$ sono gli autovalori di $\mathbf{A}$. Allora

$$\frac{x^T\mathbf{A}x}{x^T x} = \frac{x^T\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T x}{x^T\mathbf{Q}\mathbf{Q}^T x} = \frac{y^T\mathbf{\Lambda}y}{y^T y} = \frac{\sum_i \lambda_i y_i^2}{y^T y} \leq \lambda_{\max}\frac{\sum_i y_i^2}{y^T y} = \lambda_{\max}.$$

Tale maggiorazione vale per ogni x, e quindi anche per quel vettore x che dà il massimo del quoziente di Rayleigh. Il valore $\lambda_{\max}$ è raggiunto per x uguale all'autovettore corrispondente a $\lambda_{\max}$. □

Proposizione 2.5.5 *Sia $\mathbf{A} \in \mathbb{R}^{n\times n}$. Allora $\|\mathbf{A}\|_2 = \|\mathbf{A}^T\|_2$.*

Dimostrazione. Questa proprietà può essere dimostrata in modo elementare mediante la scomposizione di $\mathbf{A}$ in valori singolari, e vale anche per $\mathbf{A}$ rettangolare. Qui verrà dimostrata per $\mathbf{A}$ non singolare.

Ricordiamo che $\mathbf{A}^T\mathbf{A} > 0$. Sfruttando la Proposizione 2.5.4 Si ha

$$\|\mathbf{A}\|_2^2 = \max_{\|x\|=1} \|\mathbf{A}x\|_2^2 = \max_{\|x\|=1} x^T\mathbf{A}^T\mathbf{A}x = \lambda_{\max}(\mathbf{A}^T\mathbf{A}),$$

dove $\lambda_{\max}(\mathbf{A}^T\mathbf{A})$ indica l'autovalore massimo di $\mathbf{A}^T\mathbf{A}$. Analogamente,

$$\|\mathbf{A}^T\|_2^2 = \max_{\|x\|=1} \|\mathbf{A}^T x\|_2^2 = \max_{\|x\|=1} x^T\mathbf{A}\mathbf{A}^T x = \lambda_{\max}(\mathbf{A}\mathbf{A}^T).$$

Dato che $\mathbf{A}^T\mathbf{A} = \mathbf{A}^T\mathbf{A}\mathbf{A}^T(\mathbf{A}^T)^{-1}$, si ha che $\mathbf{A}^T\mathbf{A}$ e $\mathbf{A}\mathbf{A}^T$ sono simili, e quindi hanno gli stessi autovalori. In particolare sarà uguale il loro autovalore massimo. □

Esercizio 2.5.6 *Sia $\mathbf{M} \in \mathbb{R}^{n\times n}$ simmetrica e definita positiva, e sia $\mathbf{M} = \mathbf{L}\mathbf{L}^T$ la sua fattorizzazione di Cholesky. Mostrare che $\|\mathbf{M}\|_2 = \|\mathbf{L}\|_2^2$, e $\kappa_2(\mathbf{M}) = \kappa_2(\mathbf{L})^2$.*

Risoluzione. *Usando la fattorizzazione di Cholesky, e sfruttando i risultati delle Proposizioni 2.5.4 e 2.5.5 si ha*

$$\|\mathbf{M}\|_2 = \max_{x\neq 0} \frac{x^T\mathbf{M}x}{x^T x} = \max_{x\neq 0} \frac{x^T\mathbf{L}\mathbf{L}^T x}{x^T x} = \max_{x\neq 0} \frac{\|\mathbf{L}^T x\|_2^2}{\|x\|_2^2} = \left(\max_{x\neq 0} \frac{\|\mathbf{L}^T x\|_2}{\|x\|_2}\right)^2 = \|\mathbf{L}^T\|_2^2 = \|\mathbf{L}\|_2^2.$$

Analogamente si mostra che $\|\mathbf{M}^{-1}\|_2 = \|\mathbf{L}^{-1}\|_2^2$. *Infine,* $\kappa_2(\mathbf{M}) = \|\mathbf{M}\|_2 \|\mathbf{M}^{-1}\|_2 = \|\mathbf{L}\|_2^2 \|\mathbf{L}^{-1}\|_2^2 = \kappa_2(\mathbf{L})^2$.

Esercizio 2.5.7 *Sia* $\mathbf{A} \in \mathbb{R}^{n\times n}$ *simmetrica e definita positiva. Dimostrare che*

$$|\mathbf{A}_{ij}| < (\mathbf{A}_{ii}\mathbf{A}_{jj})^{1/2}.$$

Risoluzione. *Dimostriamo la disuguaglianza per induzione su* n.

Sia $n = 2$. *Allora, essendo* $\mathbf{A}$ *simmetrica e definita positiva, i suoi autovalori sono tutti positivi e si ha*

$$0 < det(\mathbf{A}) = det \begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{pmatrix} = \mathbf{A}_{11}\mathbf{A}_{22} - \mathbf{A}_{12}\mathbf{A}_{21} = \mathbf{A}_{11}\mathbf{A}_{22} - \mathbf{A}_{12}^2.$$

Sia ora $n > 2$. *Siano* e_i *ed* e_j *rispettivamente l'*i*-esimo e il* j*-esimo vettore della base canonica di* $\mathbb{R}^n$. *Allora la matrice* 2×2, $[e_i, e_j]^T\mathbf{A}[e_i, e_j]$ *è definita positiva, infatti per ogni* $0 \neq x \in \mathbb{R}^2$ *si ha* $x^T[e_i, e_j]^T\mathbf{A}[e_i, e_j]x = y^T\mathbf{A}y > 0$. *Quindi*

$$[e_i, e_j]^T\mathbf{A}[e_i, e_j] = \begin{pmatrix} \mathbf{A}_{ii} & \mathbf{A}_{ij} \\ \mathbf{A}_{ij} & \mathbf{A}_{jj} \end{pmatrix}.$$

Per ipotesi induttiva abbiamo allora che $|\mathbf{A}_{ij}| < (\mathbf{A}_{ii}\mathbf{A}_{jj})^{1/2}$. *Lo stesso argomento può essere utilizzato per ogni* i, j, *con* $i, j = 1, \ldots, n$.

La seguente formula, nota col nome di Formula di Sherman-Morrison, permette di scrivere l'inversa di una matrice con una certa struttura, come modifica di rango uno di una matrice non singolare. La dimostrazione della formula è una semplice verifica.

Proposizione 2.5.8 *Sia* $\mathbf{A} \in \mathbb{R}^{n\times n}$ *e siano* $u, v \in \mathbb{R}^n$. *Se* $\mathbf{A}$ *è non singolare e* $1 + v^T\mathbf{A}^{-1}u \neq 0$ *vale la seguente formula,*

$$(\mathbf{A} + uv^T)^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}u(1 + v^T\mathbf{A}^{-1}u)^{-1}v^T\mathbf{A}^{-1}.$$

Nel caso in cui la modifica sia di rango maggiore di uno, cioè $\mathbf{A} + \mathbf{U}\mathbf{V}^T$ con $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{n\times r}$, $r \geq 1$, la formula diventa

$$(\mathbf{A} + \mathbf{U}\mathbf{V}^T)^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}(\mathbf{I} + \mathbf{V}^T\mathbf{A}^{-1}\mathbf{U})^{-1}\mathbf{V}^T\mathbf{A}^{-1},$$

dove ora $(\mathbf{I} + \mathbf{V}^T\mathbf{A}^{-1}\mathbf{U}) \in \mathbb{R}^{r\times r}$ e si suppone che sia non singolare. Tale generalizzazione è nota con il nome di Formula di Sherman-Morrison-Woodbury.

Esercizio 2.5.9 *Sia* $\mathbf{B} = \mathbf{A} + uv^T$ *con* $\mathbf{A} \in \mathbb{R}^{n\times n}$ *e* $u, v \in \mathbb{R}^n$. *Supponendo di avere un metodo con basso costo computazionale per risolvere il sistema* $\mathbf{A}x = b$, *con* $b \in \mathbb{R}^n$, *mostrare come risolvere il sistema* $\mathbf{B}x = b$, *dando opportune condizioni di esistenza per la procedura.*

Risoluzione. *Applicando la formula di Sherman-Morrison-Woodbury, si ha che*

$$\begin{aligned} x &= (\mathbf{A} + uv^T)^{-1}b = (\mathbf{A}^{-1} - \mathbf{A}^{-1}u(1 + v^T\mathbf{A}^{-1}u)^{-1}v^T\mathbf{A}^{-1})b \\ &= \mathbf{A}^{-1}b - \mathbf{A}^{-1}u(1 + v^T\mathbf{A}^{-1}u)^{-1}v^T\mathbf{A}^{-1}b. \end{aligned}$$

Siano y_1 *e* y_2, *rispettivamente, le soluzioni dei sistemi lineari* $\mathbf{A}y_1 = b$, *e* $\mathbf{A}y_2 = u$, *ottenute con il metodo "veloce", come da ipotesi. Sostituendo* y_1 *e* y_2 *nella formula, si ottiene*

$$x = y_1 - y_2(1 + v^Ty_2)^{-1}v^Ty_1 = y_1 - y_2\frac{v^Ty_1}{1 + v^Ty_2}.$$

Una volta calcolati y_1 e y_2, il costo computazionale quindi è dato dal calcolo dei due prodotti scalari $v^T y_1$ e $v^T y_2$, cioè $2(2n-1)$ flops, e della DAXPY*, che costa $2n$ flops. Il costo complessivo per risolvere il sistema con $\mathbf{B}$ è quindi dato dal costo della risoluzione di* **due** *sistemi con $\mathbf{A}$, a cui vanno aggiunte circa $6n$ operazioni floating point. Se il costo per risolvere sistemi con $\mathbf{A}$ è molto più basso che per $\mathbf{B}$, per esempio nel caso di $\mathbf{A}$ a banda, la procedura è quindi molto vantaggiosa.*

Esercizio 2.5.10 *Determinare un metodo efficiente per risolvere il sistema*

$$\mathbf{A}x = b, \quad \text{con} \quad \mathbf{A} = \begin{pmatrix} \mathbf{A}_{11} & 0 & \dots & 0 & \mathbf{A}_{1n} \\ \mathbf{A}_{21} & \mathbf{A}_{22} & 0 & \dots & 0 \\ 0 & \mathbf{A}_{32} & \mathbf{A}_{33} & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \mathbf{A}_{n,n-1} & \mathbf{A}_{nn} \end{pmatrix},$$

e valutarne il costo computazionale.

Risoluzione. *È possibile scrivere $\mathbf{A} = \mathbf{L} + \mathbf{A}_{1n} e_1 e_n^T$, dove $\mathbf{L}$ indica la parte triangolare inferiore di $\mathbf{A}$. Usando la formula di Sherman-Morrison, la soluzione x può essere scritta come*

$$x = \mathbf{A}^{-1}b = (\mathbf{L} + \mathbf{A}_{1n}e_1e_n^T)^{-1}b = \mathbf{L}^{-1}b - \mathbf{L}^{-1}e_1(1 - \mathbf{A}_{1n}e_n^T\mathbf{L}^{-1}e_1)\mathbf{A}_{1n}e_n^T\mathbf{L}^{-1}b,$$

che richiede la risoluzione dei due sistemi $\mathbf{L}y_1 = b$ ($3n$ flops) e $\mathbf{L}y_2 = e_1$ ($2n$ flops), con $\mathbf{L}$ bidiagonale. I prodotti scalari con e_n non comportano alcun costo computazionale (corrispondono all'estrazione della corrispondente componente del vettore), quindi l'unico costo aggiuntivo è l'operazione di DAXPY*, che costa $2n$ flops. Il costo complessivo del calcolo di x è quindi di $7n$ flops.*

Esercizio 2.5.11 *Sia $\kappa(\mathbf{A})$ il numero di condizionamento di $\mathbf{A}$ con la norma indotta Euclidea. Sia x la soluzione esatta del sistema $\mathbf{A}x = b$, e sia invece $\tilde{x}$ una soluzione approssimata (es. in aritmetica finita). Sia $r = b - \mathbf{A}\tilde{x}$ il residuo associato a tale approssimazione. Mostrare che*

$$\frac{\|x - \tilde{x}\|}{\|x\|} \leq \kappa(\mathbf{A}) \frac{\|r\|}{\|b\|}.$$

Risoluzione. *Dalla definizione di residuo si ha che $\tilde{x} = \mathbf{A}^{-1}(b - r) = x - \mathbf{A}^{-1}r$. Quindi*

$$\frac{\|x - \tilde{x}\|}{\|x\|} = \frac{\|\mathbf{A}^{-1}r\|}{\|x\|} \leq \frac{\|\mathbf{A}^{-1}\|\,\|r\|}{\|x\|} = \frac{\|\mathbf{A}\|}{\|\mathbf{A}\|}\frac{\|\mathbf{A}^{-1}\|\,\|r\|}{\|x\|} = \kappa(\mathbf{A})\frac{\|r\|}{\|\mathbf{A}\|\,\|x\|} \leq \kappa(\mathbf{A})\frac{\|r\|}{\|\mathbf{A}x\|} = \kappa(\mathbf{A})\frac{\|r\|}{\|b\|}.$$

Questo esercizio mostra che il residuo, nonostante sia una grandezza molto utilizzata per valutare la bontà della approssimazione $\tilde{x}$, può non essere molto informativo sull'errore nel caso di un numero di condizionamento elevato della matrice dei coefficienti $\mathbf{A}$. In particolare, la maggiorazione mostra la relazione che intercorre tra una stima a posteriori (il residuo) ed una stima a priori (l'errore). Questo esercizio corrisponde al Corollario 2.1.7, dove il residuo gioca il ruolo di δb.

Esercizio 2.5.12 *Determinare il costo computazionale per il calcolo dell'inversa di una matrice $\mathbf{A}$ tridiagonale applicando il metodo di eliminazione di Gauss senza pivoting.*

Risoluzione. *Il calcolo dei fattori $\mathbf{L}$ e $\mathbf{U}$ mediante l'algoritmo di Thomas costa $3n-3$ flops. Come nel caso triangolare (Sezione 2.2.1) è quindi necessario risolvere gli n sistemi $\mathbf{L}y = e_k$, $\mathbf{U}x = y$, per*

$k = 1, \ldots, n$. *Notiamo che il costo per la risoluzione del primo sistema, quello bidiagonale inferiore, può essere ridotto seguendo lo stesso argomento utilizzato nella Sezione 2.2.1, e ottenendo*

$$y_k = f_k, \quad y_i = f_i - \beta_i y_{i-1}, \quad i = k, \ldots, n,$$

con un costo di $n - k$ flops per $k = 1, \ldots, n$ invece di $2n - 2$ flops. Il costo totale dell'inversione è quindi di

$$3n - 3 + \sum_{k=1}^{n}(n-k) + n(3n-2) = \frac{7}{2}n^2 + \frac{1}{2}n - 3 = \frac{7}{2}n^2 + \mathcal{O}(n).$$

Esercizio 2.5.13 *Siano $\mathbf{A} \in \mathbb{R}^{n\times m}$ con $n \geq m$ e*

$$\mathbf{B} = \begin{pmatrix} \mathbf{I}_n & \mathbf{A} \\ \mathbf{A}^T & \mathbf{O}_m \end{pmatrix} \in \mathbb{R}^{(n+m)\times(n+m)},$$

dove $\mathbf{I}_n \in \mathbb{R}^{n\times n}$ e $\mathbf{O}_m \in \mathbb{R}^{m\times m}$ indicano rispettivamente la matrice identità e la matrice nulla.

1. *Mostrare che $\mathbf{B}$ è non singolare se e solo se $\mathbf{A}^T\mathbf{A}$ è non singolare.*
2. *Dedurre dal punto precedente un metodo risolutivo per il sistema lineare*

$$\mathbf{B}x = b, \quad b \in \mathbb{R}^{n+m},$$

che abbia un costo inferiore a $\mathcal{O}((n+m)^3)$.

Risoluzione. *Si verifica facendo il prodotto esplicito che*

$$\mathbf{B} = \begin{pmatrix} \mathbf{I}_n & \\ \mathbf{A}^T & \mathbf{I}_m \end{pmatrix} \begin{pmatrix} \mathbf{I}_n & \\ & -\mathbf{A}^T\mathbf{A} \end{pmatrix} \begin{pmatrix} \mathbf{I}_n & \mathbf{A} \\ & \mathbf{I}_m \end{pmatrix} =: \mathbf{L}\mathbf{D}\mathbf{L}^T.$$

La matrice $\mathbf{L}$ è necessariamente non singolare in quanto triangolare inferiore con diagonale unitaria. Si ha quindi che

$$det\,(\mathbf{B}) = det\,(\mathbf{L}\mathbf{D}\mathbf{L}^T) = det\,(\mathbf{L})\,det\,(\mathbf{D})\,det\,(\mathbf{L}^T) = det\,(\mathbf{D}) = det\,(\mathbf{I}_n)\,det\,(-\mathbf{A}^T\mathbf{A}) = det\,(-\mathbf{A}^T\mathbf{A}).$$

Quindi $\mathbf{B}$ è non singolare se e solo se $\mathbf{A}^T\mathbf{A}$ è non singolare. (abbiamo utilizzato il fatto che il determinante di una matrice diagonale a blocchi è il prodotto dei determinanti dei blocchi). La risoluzione del sistema $\mathbf{B}x = b$, equivale a risolvere $\mathbf{L}\mathbf{D}\mathbf{L}^T x = b$, da cui $x = \mathbf{L}^{-T}\mathbf{D}^{-1}\mathbf{L}^{-1}b$. Il calcolo di x quindi prevede lo svolgimento delle seguenti operazioni

1) Risoluzione di $\mathbf{L}v = b$

2) Risoluzione di $\mathbf{D}w = v$

3) Risoluzione di $\mathbf{L}^T x = w$

Partizioniamo il vettore b come $b = [b_1; b_2]$ in modo conforme ai blocchi di righe di $\mathbf{B}$, cosicchè $b_1 \in \mathbb{R}^n$ e $b_2 \in \mathbb{R}^m$. Si verifica che

$$\mathbf{L}^{-1} = \begin{pmatrix} \mathbf{I}_n & \\ -\mathbf{A}^T & \mathbf{I}_m \end{pmatrix}$$

per cui il punto 1) corrisponde al prodotto $\mathbf{L}^{-1}b$, il cui unico costo computazionale è dato dal prodotto $-\mathbf{A}^T b_1$, ed è di $m(2n-1)$, a cui deve essere aggiunto il costo della somma di due vettori di dimensione m.

Analogamente, il costo del punto 3) corrisponde al prodotto $-\mathbf{A}w_2$, che è di $n(2m-1)$, a cui va aggiunto il costo della somma di due vettori di dimensione n.

La risoluzione del sistema a blocchi nel punto 2) richiede la risoluzione del sistema $\mathbf{A}^T\mathbf{A}w_2 = v_2$. *Siccome* $\mathbf{A}^T\mathbf{A}$ *è* $m \times m$, *simmetrica e definita positiva, è possibile usare la fattorizzazione di Cholesky, con cui la risoluzione di questo sistema ha un costo di* $\frac{1}{3}m^3 + \mathcal{O}(m^2)$.

Il costo complessivo dell'intera procedura è quindi di $4nm + \frac{1}{3}m^3 + \mathcal{O}(m^2) + \mathcal{O}(n)$. *Nel caso* $\mathbf{A}^T\mathbf{A}$ *debba essere calcolata esplicitamente, allora occorre aggiungere* $(2n-1)m^2$ *operazioni aggiuntive. Si noti che è possibile ottenere il fattore di Cholesky* $\mathbf{L}^T$ *direttamente da* $\mathbf{A}$ *(cioè senza il calcolo di* $\mathbf{A}^T\mathbf{A}$*) mediante la fattorizzazione QR (si veda il Capitolo 4).*

Capitolo 3

Risoluzione di sistemi lineari: metodi iterativi

Si vuole risolvere il sistema

$$\mathbf{A}x = b, \tag{3.1}$$

con $\mathbf{A} \in \mathbb{R}^{n\times n}$ e $b \in \mathbb{R}^n$, mediante la costruzione di una successione di soluzioni approssimate $\{x_k\}$, $x_k \in \mathbb{R}^n \; \forall k$, tali che[1]

$$x_k \to x_*, \quad k \to +\infty,$$

dove x_* indica la soluzione esatta di (3.1). Ovviamente la successione dev'essere costruita in modo tale che dopo un certo numero di passi $\widehat{k}$, "piccolo" in termini di costi computazionali, sia soddisfatta la relazione

$$\|x_{\widehat{k}} - x_*\| < \texttt{tol}$$

per qualche norma vettoriale, e per una tolleranza (`tol`) fissata, per esempio $\texttt{tol} = 10^{-6}$. Esistono metodi che, almeno in aritmetica esatta, generano una successione finita, $x_k \to x_*$, per $k \to n$, cioè dopo al piú n passi viene ottenuta la soluzione x_*.

3.1 Metodi iterativi classici (stazionari)

In questa sezione introduciamo alcuni metodi classici per la risoluzione iterativa di sistemi lineari. Nella maggior parte dei casi sono stati ormai soppiantati da metodi molto più potenti, in termini di costo computazionale e velocità di convergenza. D'altra parte rappresentano un ottimo punto di partenza per investigazioni sia teoriche che applicative, ed in alcune applicazioni, per esempio metodi multi-livello, vengono ancora usati con successo.

Una strategia generale pr costruire metodi iterativi stazionari è basata su una decomposizione additiva, detta *splitting* della matrice dei coefficienti $\mathbf{A}$,

$$\mathbf{A} = \mathbf{P} - \mathbf{N}, \tag{3.2}$$

[1] Una notazione alternativa molto usata per x_k è $x^{(k)}$, in modo che l'indice k di iterazione non venga confuso con l'indice di componente del vettore. Per non appesantire la notazione useremo x_k, ed useremo una diversa notazione per l'indice di componente, usato molto più di rado in questo contesto.

con $\mathbf{P}$ e $\mathbf{N}$ matrici opportune e $\mathbf{P}$ non singolare. Allora

$$\mathbf{A}x = b, \quad \Leftrightarrow \quad \mathbf{P}x - \mathbf{N}x = b, \quad \Leftrightarrow \quad \mathbf{P}x = \mathbf{N}x + b.$$

Per $\mathbf{P}$ non singolare, questa equazione ha la forma $x = \Phi(x)$, e quindi è possibile pensare ad una iterazione di punto fisso, che convergerà sotto opportune ipotesi su Φ. Più precisamente, fissato x_0, può essere definita la seguente iterazione del tipo $x_{k+1} = \Phi(x_k)$:

$$\mathbf{P}x_{k+1} = \mathbf{N}x_k + b, \quad k \geq 0 \quad \Leftrightarrow \quad x_{k+1} = \mathbf{P}^{-1}\mathbf{N}x_k + \mathbf{P}^{-1}b. \tag{3.3}$$

L'iterata successiva x_{k+1} è ottenuta risolvendo un sistema lineare con $\mathbf{P}$.

Osservazione: L'iterazione (3.3) può essere scritta come $x_{k+1} = x_k + \mathbf{P}^{-1}r_k$, dove $r_k = b - \mathbf{A}x_k$ indica il residuo al passo k. Infatti, inserendo la definizione del residuo e ricordando che $\mathbf{A} = \mathbf{P} - \mathbf{N}$ si ha

$$x_{k+1} = x_k + \mathbf{P}^{-1}b - \mathbf{P}^{-1}\mathbf{A}x_k = x_k + \mathbf{P}^{-1}b - \mathbf{P}^{-1}\mathbf{P}x_k + \mathbf{P}^{-1}\mathbf{N}x_k = \mathbf{P}^{-1}b + \mathbf{P}^{-1}\mathbf{N}x_k. \tag{3.4}$$

Per costruzione dell'iterazione di punto fisso, la formulazione (3.3) del metodo iterativo risulta essere fortemente consistente, in quanto la soluzione esatta x soddisfa l'iterazione per ogni k.

Ogni scelta di $\mathbf{P}$ darà luogo ad un distinto metodo, con diverse proprietà di convergenza e diversi costi computazionali. Alcune proprietà sono comunque comuni, e derivanti dall'iterazione generale.

Definizione 3.1.1 (Metodo convergente) *L'iterazione (3.4) definisce un metodo* convergente *se l'errore $e_k := x_* - x_k$, soddisfa*

$$\lim_{k\to+\infty} e_k = 0, \quad \textit{per ogni} \quad x_0 \textit{ iterata iniziale.}$$

Per introdurre il concetto di convergenza dell'iterazione, si richiama la definizione di raggio spettrale $\rho(\mathbf{B})$ definito come $\rho(\mathbf{B}) = \max_{\lambda\in\text{spec}(\mathbf{B})} |\lambda|$.

Teorema 3.1.2 *Il metodo (3.3) converge alla soluzione esatta x_* per ogni iterato iniziale x_0 se e solo se la* matrice di iterazione

$$\mathbf{B} := \mathbf{P}^{-1}\mathbf{N},$$

soddisfa $\rho(\mathbf{B}) < 1$.

Dimostrazione. Come relazione preliminare, notiamo che sottraendo $x_* = \mathbf{P}^{-1}\mathbf{N}x_* + \mathbf{P}^{-1}b$ da (3.4), si ottiene

$$x_* - x_{k+1} = \mathbf{P}^{-1}\mathbf{N}(x_* - x_k), \qquad \Leftrightarrow \qquad e_{k+1} = \mathbf{B}e_k.$$

Iterando la relazione,

$$e_{k+1} = \mathbf{B}e_k = \mathbf{B}(\mathbf{B}e_{k-1}) = \cdots = \mathbf{B}^{k+1}e_0.$$

$\boxed{\Leftarrow}$ Supponiamo che $\rho(\mathbf{B}) < 1$. Allora

$$\lim_{k\to+\infty} \mathbf{B}^k = 0. \tag{3.5}$$

Dimostriamo questo fatto nel solo caso diagonalizzabile, sebbene sia dimostrabile anche in presenza di blocchi di Jordan: si ha

$$\lim_{k\to+\infty} \mathbf{B}^k = \lim_{k\to+\infty} (\mathbf{X}\mathbf{\Lambda}\mathbf{X}^{-1})^k = \mathbf{X}(\lim_{k\to+\infty} \mathbf{\Lambda}^k)\mathbf{X}^{-1} = 0.$$

Dalla (3.5) segue che $\lim_{k\to+\infty} e_k = \lim_{k\to+\infty} \mathbf{B}^k e_0 = 0$, per ogni $e_0 \in \mathbb{R}^n$, cioè la successione in (3.3) converge per ogni scelta dell'iterato iniziale x_0.

$\boxed{\Rightarrow}$ Supponiamo che il metodo sia convergente e supponiamo per assurdo che $\rho(\mathbf{B}) \geq 1$. Allora esiste almeno un'autocoppia di $\mathbf{B}$, (λ, v) con $|\lambda| \geq 1$. Scegliamo quindi l'iterato iniziale x_0 tale che l'errore iniziale e_0 sia proprio uguale all'autovettore v, $e_0 = v$. Dunque, $e_k = \mathbf{B}^k e_0 = \lambda^k e_0 \nrightarrow 0$ per $k \to +\infty$. Tale risultato è un assurdo, poichè il metodo è convergente. $\square$

Corollario 3.1.3 *Se $\|\mathbf{B}\| < 1$ per qualche norma matriciale indotta, allora il metodo (3.3) è convergente.*

Dim. Per le osservazioni fatte nella dimostrazione del Teorema 3.1.2, per ogni e_0 e per la norma per cui vale $\|\mathbf{B}\| < 1$ si ha

$$\|e_k\| = \|\mathbf{B}^k e_0\| \leq \|\mathbf{B}^k\| \, \|e_0\| \leq \|\mathbf{B}\|^k \, \|e_0\| \to 0, \quad k \to +\infty. \quad \square$$

Più il raggio spettrale della matrice di iterazione $\rho(\mathbf{B})$, è piccolo, cioè lontano da 1, più la convergenza del metodo (3.3) sarà rapida. Condizioni **necessarie** per la convergenza del metodo, legate alla distribuzione degli autovalori della matrice di iterazione, sono:

i) $|\det(\mathbf{B})| = \left| \prod_{i=1}^{n} \lambda_i \right| < 1,$

ii) $|\mathrm{tr}(\mathbf{B})| = \left| \sum_{i=1}^{n} \mathbf{B}_{ii} \right| = \left| \sum_{i=1}^{n} \lambda_i \right| \leq n.$

Per lo studio della convergenza è interessante studiare quale sia il decremento medio dell'errore ad ogni iterazione. A tal fine, consideriamo la media geometrica del decremento dell'errore, in norma, ad ogni iterazione, cioè

$$\left(\frac{\|e_k\|}{\|e_{k-1}\|} \frac{\|e_{k-1}\|}{\|e_{k-2}\|} \cdots \frac{\|e_1\|}{\|e_0\|} \right)^{1/k} = \left(\frac{\|e_k\|}{\|e_0\|} \right)^{1/k} = \left(\frac{\|\mathbf{B}^k e_0\|}{\|e_0\|} \right)^{1/k} \leq \|\mathbf{B}^k\|^{1/k}$$

da cui si deduce che $\|\mathbf{B}^k\|^{1/k}$ è il fattore medio di convergenza del metodo dopo k iterazioni. Tale termine può essere usato per dare indicazioni su quanto calerà l'errore, in media, per ogni iterazione. Infatti, supponiamo che si desideri una relazione del tipo $\|e_j\| \leq 10^{-\alpha}\|e_{j-1}\|$ ad ogni iterazione j fino a k. Allora è sufficiente che si abbia $10^{-\alpha} = \|\mathbf{B}^k\|^{1/k}$, cioè

$$\log_{10} 10^{-\alpha} = \log_{10} \|\mathbf{B}^k\|^{1/k}, \quad \text{cioè} \quad \alpha = -\frac{1}{k} \log_{10} \|\mathbf{B}^k\| =: R_k(\mathbf{B}),$$

dove $R_k(\mathbf{B})$ è la velocità media di convergenza. Quindi $R_k(\mathbf{B})$ indica il decremento medio ad ogni iterazione del metodo.

Grazie al seguente risultato, è possibile sostituire il calcolo di $R_k(\mathbf{B})$ ad ogni k, con il calcolo del solo raggio spettrale.

Teorema 3.1.4 *(Formula di Gelfand). Sia $\mathbf{A} \in \mathbb{R}^{n \times n}$ e sia $\|\cdot\|$ una norma matriciale. Si ha allora che*

$$\lim_{k \to +\infty} \sqrt[k]{\|\mathbf{A}^k\|} = \rho(\mathbf{A}).$$

Dim. Consideriamo solo il caso di norma indotta Euclidea ed $\mathbf{A}$ diagonalizzabile. Per $\mathbf{A}$ simmetrica, il risultato è vero per ogni k, senza andare al limite. Per $\mathbf{A}$ diagonalizzabile, la dimostrazione segue dalla continuità della norma matriciale:

Notiamo innanzi tutto che $\rho(A)^k = \rho(A^k) \leq \|A^k\|$ e quindi $\rho(A) \leq \|A^k\|^{1/k}$ per ogni $k = 1, 2, ...$ Sia dato $\varepsilon > 0$. La marice $\tilde{A} = (\rho(A) + \varepsilon)^{-1} A$ ha raggio spettale strettamente minore di uno e quindi $\|\tilde{A}^k\| \to 0$ per $k \to +\infty$ ed esiste un $n = n(\varepsilon, A)$ tale che $\|\tilde{A}^k\| \leq 1$ per $k \geq n$. Questa relazione equivale a dire che $\|A^k\| \leq (\rho(A) + \varepsilon)^k$ per $k \geq n$, o meglio ancora, $\|A^k\|^{1/k} \leq (\rho(A) + \varepsilon)$ per $k \geq n$. Dato che $\varepsilon > 0$ è arbitrario e $\rho(A) \leq \|A^k\|^{1/k}$ per ogni k, si ottiene che $\lim_{k \to +\infty} \sqrt[k]{\|\mathbf{A}^k\|}$ esiste ed è uguale a $\rho(A)$. $\square$

Quindi, per una norma indotta $\|\cdot\|$, definendo con $R(\mathbf{B}) = -\log_{10} \rho(\mathbf{B})$ la velocità *asintotica* di convergenza, si ha

$$R(\mathbf{B}) = \lim_{k \to +\infty} R_k(\mathbf{B}).$$

3.1.1 Metodo di Jacobi

Consideriamo la seguente decomposizione additiva (o splitting) della matrice $\mathbf{A}$ in (3.1):

$$\mathbf{A} = -\mathbf{E} + \mathbf{D} - \mathbf{F}, \tag{3.6}$$

con $\mathbf{E}$ triangolare strettamente inferiore, $\mathbf{F}$ triangolare strettamente superiore e $\mathbf{D}$ diagonale. Riscrivendo lo splitting nella notazione di (3.2), il metodo di Jacobi corrisponde a scegliere $\mathbf{P} = \mathbf{D}$, quindi

$$\mathbf{A} = \mathbf{D} - (\mathbf{E} + \mathbf{F}), \quad \Rightarrow \quad \mathbf{P} := \mathbf{D}, \quad \mathbf{N} := \mathbf{E} + \mathbf{F}.$$

Tale scelta di $\mathbf{P}$ è possibile sotto l'ipotesi che la diagonale di $\mathbf{A}$ sia non singolare, cioè $\mathbf{A}_{ii} \neq 0$ per $i = 1, \ldots, n$. La matrice di iterazione per il metodo di Jacobi si può quindi scrivere come:

$$\mathbf{B}_J := \mathbf{P}^{-1}\mathbf{N} = \mathbf{D}^{-1}(\mathbf{E} + \mathbf{F}) = \mathbf{D}^{-1}(\mathbf{D} - \mathbf{A}) = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A},$$

mentre il corrispondente algoritmo (di Jacobi) è quindi dato da:

Fissato $x_0 \in \mathbb{R}^n$ e $r_0 = b - Ax_0$
  Per $k = 0, 1, 2, \ldots,$
    $x_{k+1} = x_k + \mathbf{D}^{-1}r_k,$ $2n$ flops
    $r_{k+1} = b - \mathbf{A}x_{k+1},$ n flops + Mxv
    Se convergente stop $2n$ (norma del residuo)
  end

A fianco di ogni operazione è stato segnalato il costo computazionale: in totale il costo è di $5n$ flops per iterazione, a cui va aggiunto il costo dell'operazione matrice-per-vettore, che dipende dalla sparsità di A. Si noti che l'operazione $\mathbf{D}^{-1}r_k$ è svolta dividendo ogni componente di r_k per il corrispondente elemento diagonale di $\mathbf{D}$, cosicchè la matrice $\mathbf{D}$ non viene in effetti memorizzata. Il costo complessivo del metodo dipende dal numero di iterazioni effettuato dal metodo, e quindi non è possibile stimarlo a priori in modo affidabile, in quanto dipende molto dal dato iniziale x_0. Per completare l'algoritmo bisogna fornire un criterio d'arresto, che, senza altre informazioni, può essere basato sulla norma del residuo. In particolare, l'iterazione si arresta se vale la seguente disuguaglianza,

$$\frac{\|r_{k+1}\|}{\|r_0\|} \leq \texttt{tol},$$

dove `tol` è una tolleranza scelta dall'utente, e dipende dall'accuratezza richiesta dal problema; tipici valori di `tol` nella pratica sono dell'ordine di $10^{-6}, 10^{-8}$. Si noti che è stato usato il residuo relativo, quindi l'iterazione termina quando la norma del residuo *è diminuita rispetto a quella iniziale* di `tol`. Se $\|r_0\|$ è già molto piccolo, allora richiedere una ulteriore diminuizione di `tol` potrebbe essere eccessivo. In questi casi, un test anche sul residuo assoluto può essere opportuno. Il criterio d'arresto descritto sopra sarà usato anche per i metodi iterativi presentati nel seguito.

3.1.2 Metodo di Gauss-Seidel

Nel metodo di Gauss-Seidel la matrice $\mathbf{P}$ viene scelta come la parte triangolare inferiore di $\mathbf{A}$ nello splitting (3.6), cioè

$$\mathbf{P} = \mathbf{D} - \mathbf{E}, \qquad \mathbf{N} = \mathbf{F}.$$

Anche in questo caso, $\mathbf{P}$ è invertibile se e solo se $\mathbf{A}_{ii} \neq 0$ per $i = 1, \ldots, n$ poichè $\mathbf{P}$ è triangolare inferiore con diagonale coincidente con quella di $\mathbf{A}$. La matrice di iterazione del metodo di Gauss-Seidel è quindi data da

$$\mathbf{B}_{GS} = \mathbf{P}^{-1}\mathbf{N} = (\mathbf{D} - \mathbf{E})^{-1}\mathbf{F}.$$

L'iterazione del metodo di Gauss-Seidel sarà quindi della forma

Fissato $x_0 \in \mathbb{R}^n$ e $r_0 = b - Ax_0$
  Per $k = 0, 1, 2, \ldots,$
    $x_{k+1} = x_k + (\mathbf{D} - \mathbf{E})^{-1} r_k,$ n flops + risoluz.triang.inf.
    $r_{k+1} = b - \mathbf{A}x_{k+1},$ n flops + Mxv
    Se convergente stop $2n$ (norma del residuo)
  end

Il costo computazionale per iterazione differisce da quello del metodo di Jacobi solo per il costo della risoluzione con la matrice $\mathbf{D} - \mathbf{E}$. Nel caso di matrice triangolare inferiore piena, il costo sarà quello visto nel Paragrafo 2.2, cioè n^2 operazioni floating point.

Osservazione. L'iterazione $(\mathbf{D}-\mathbf{E})x_{k+1} = \mathbf{F}x_k + b$ può essere riscritta come $\mathbf{D}x_{k+1} = \mathbf{E}x_{k+1} + \mathbf{F}x_k + b$, cioè $x_{k+1} = \mathbf{D}^{-1}\mathbf{E}x_{k+1} + \mathbf{D}^{-1}\mathbf{F}x_k + \mathbf{D}^{-1}b$. Sfruttando la struttura strettamente triangolare inferiore di $\mathbf{E}$, si nota che le componenti di x_{k+1} possono essere determinate in modo da utilizzare subito le componenti appena calcolate dello stesso vettore, cioè

$$x_{k+1}(i) = \frac{1}{\mathbf{A}_{ii}} \left(b(i) - \sum_{j=1}^{i-1} \mathbf{A}_{ij} x_{k+1}(j) - \sum_{j=i+1}^{n} \mathbf{A}_{ij} x_k(j) \right), \qquad i = 1, \ldots, n$$

Per questo motivo il metodo veniva anche chiamato in passato il "metodo degli spostamenti successivi".

Esempio 3.1.5 È dato il sistema lineare $\mathbf{A}x = b$ con matrice

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 0 \\ 4 & 5 & 5 \\ 4 & 0 & 11 \end{pmatrix} = \mathbf{D} - \mathbf{E} - \mathbf{F} = \begin{pmatrix} 1 & & \\ & 5 & \\ & & 11 \end{pmatrix} + \begin{pmatrix} 0 & & \\ 4 & 0 & \\ 4 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 2 & 0 \\ & 0 & 5 \\ & & 0 \end{pmatrix},$$

da cui, la matrice di iterazione dei metodi di Jacobi e di Gauss-Seidel è:

$$\mathbf{B}_J = \mathbf{D}^{-1}(\mathbf{E} + \mathbf{F}) = -\begin{pmatrix} 0 & 2 & 0 \\ 4/5 & 0 & 1 \\ 4/11 & 0 & 0 \end{pmatrix}, \qquad \mathbf{B}_{GS} = (\mathbf{D} - \mathbf{E})^{-1}\mathbf{F} = \begin{pmatrix} 0 & -2 & 0 \\ 0 & 1.6 & -1 \\ 0 & 0.7 & 0 \end{pmatrix}.$$

Gli autovalori di queste matrici sono (accurati ai primi due decimali) $\Lambda(\mathbf{B}_J) = \{-1.44, 0.87, 0.57\}$ e $\Lambda(\mathbf{B}_{GS}) = \{0, 0.8 \pm 0.29i\}$, per cui

$$\rho(\mathbf{B}_J) = 1.44 > 1, \qquad \rho(\mathbf{B}_{GS}) = 0.8 < 1.$$

Il Teorema 3.1.2 assicura quindi che il metodo di Gauss-Seidel converge mentre quello di Jacobi diverge.

Esempio 3.1.6 È dato il sistema lineare

$$\mathbf{A}x = b, \qquad \mathbf{A} = \begin{pmatrix} 3 & 0 & 4 \\ 7 & 4 & 2 \\ -1 & -1 & -2 \end{pmatrix}, \quad b = \begin{pmatrix} 7 \\ 13 \\ -4 \end{pmatrix}.$$

Si ha quindi che $\rho(\mathbf{B}_J) \approx 1.33$, e $\rho(\mathbf{B}_{GS}) \approx 0.25$, e come mostrato nel primo grafico di Figura 3.1, la norma-2 del residuo ottenuto applicando il metodo di Gauss-Seidel decresce mentre quella del residuo ottenuto applicando Jacobi cresce, cioè il metodo di Jacobi non converge come assicurato dal Teorema 3.1.2.

Esempio 3.1.7 È dato il sistema lineare

$$\mathbf{A}x = b, \qquad \mathbf{A} = \begin{pmatrix} -3 & 3 & -6 \\ -4 & 7 & -8 \\ 5 & 7 & -9 \end{pmatrix}, \quad b = \begin{pmatrix} -6 \\ -5 \\ 3 \end{pmatrix}.$$

Si ha quindi che $\rho(\mathbf{B}_J) \approx 0.813$, e $\rho(\mathbf{B}_{GS}) \approx 1.11$, e come mostrato nel secondo grafico in Figura 3.1, la situazione è del tutto ribaltata rispetto a quella dell'Esempio 3.1.6: il metodo di Jacobi converge mentre quello di Gauss-Seidel no.

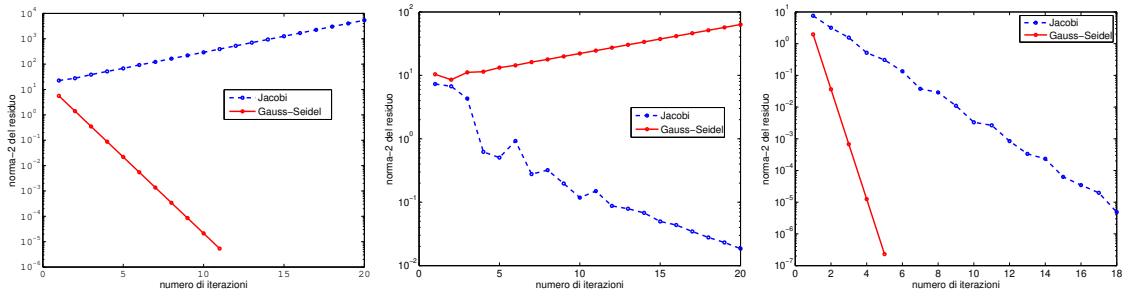

Figura 3.1: Metodi di Jacobi e di Gauss-Seidel. Norma-2 del residuo all'aumentare del numero di iterazioni.

Esempio 3.1.8 È dato il sistema lineare

$$\mathbf{A}x = b, \qquad \mathbf{A} = \begin{pmatrix} 4 & 1 & 1 \\ 2 & -9 & 0 \\ 0 & -8 & -6 \end{pmatrix}, \quad b = \begin{pmatrix} 6 \\ -7 \\ -14 \end{pmatrix}.$$

Si ha quindi che $\rho(\mathbf{B}_J) \approx 0.44$, e $\rho(\mathbf{B}_{GS}) \approx 0.018$, e il Teorema 3.1.2 assicura che entrambi i metodi, Jacobi e Gauss-Seidel, convergono. Come mostrato nel grafico di destra in Figura 3.1 però, il metodo di Gauss-Seidel converge più velocemente di quello di Jacobi. Infatti, minore è il raggio spettrale della matrice di iterazione, maggiore sarà la velocità di convergenza del metodo.

3.1.3 Risultati di convergenza per i metodi di Jacobi e Gauss-Seidel

In questa sezione sono riportati alcuni risultati sulla convergenza dei metodi stazionari proposti fino ad ora. Questi risultati si basano su diverse ipotesi imposte sulla matrice dei coefficienti $\mathbf{A}$. La letteratura in materia è estremamente vasta, e la presentazione qui si limita a pochissimi risultati di interesse. Per esempio, esistono molti altri risultati di convergenza per matrici irriducibili, che non riportiamo. Rimandiamo in questo caso alla letteratura di riferimento ed ai testi in bibliografia.

Teorema 3.1.9 *Sia* $\mathbf{A} \in \mathbb{R}^{n\times n}$ *una matrice a* dominanza diagonale stretta per righe*, cioè*

$$|\mathbf{A}_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{n} |\mathbf{A}_{ij}|, \quad i = 1, \ldots, n.$$

allora sia il metodo di Jacobi che quello di Gauss-Seidel convergono e si ha $\|\mathbf{B}_{GS}\|_\infty \leq \|\mathbf{B}_J\|_\infty < 1$.

Dimostrazione. Dimostriamo solo la convergenza dei due metodi.
La dominanza diagonale di $\mathbf{A}$ assicura che la sua diagonale, cioè $\mathbf{D}$ sia non singolare. Inoltre,

$$\|\mathbf{B}_J\|_\infty = \|\mathbf{D}^{-1}(\mathbf{E}+\mathbf{F})\|_\infty = \max_{i=1,\dots,n} \frac{1}{|\mathbf{A}_{ii}|} \sum_{\substack{j=1 \\ j\neq i}}^{n} |\mathbf{A}_{ij}| < 1,$$

e quindi il metodo di Jacobi converge.

Dimostrazione per il metodo di Gauss-Seidel. Sia (λ, v) una autocoppia di $\mathbf{B}_{GS} = (\mathbf{D}-\mathbf{E})^{-1}\mathbf{F}$ tale che $|\lambda| = \rho(\mathbf{B}_{GS})$. Supponiamo che $\|v\|_\infty = 1 = |v_k|$ per qualche $k \in \{1, \dots n\}$. Allora

$$\lambda v = \mathbf{B}_{GS} v \qquad \Leftrightarrow \qquad \lambda \mathbf{D} v = \lambda \mathbf{E} v + \mathbf{F} v.$$

Supponiamo per assurdo che $|\lambda| \geq 1$. Quindi $\mathbf{D}v = \mathbf{E}v + \frac{1}{\lambda}\mathbf{F}v$ e, per la componente k-esima, si ha

$$A_{kk} v_k = \sum_{j=1}^{k-1} A_{k,j} v_j + \frac{1}{\lambda} \sum_{j=k+1}^{n} A_{k,j} v_j.$$

Usando i moduli, e ricordando che $|v_k| = \|v\|_\infty$,

$$|A_{kk}|\,|v_k| \leq \left(\sum_{j=1}^{k-1} |A_{k,j}|\,|v_j| + \frac{1}{|\lambda|} \sum_{j=k+1}^{n} |A_{k,j}|\,|v_j| \right) \leq \left(\sum_{j=1}^{k-1} |A_{k,j}| + \frac{1}{|\lambda|} \sum_{j=k+1}^{n} |A_{k,j}| \right) |v_k|.$$

Siccome $\frac{1}{|\lambda|} \leq 1$, vale quindi

$$|A_{kk}| \leq \sum_{j=1, j\neq k}^{n} |A_{k,j}|,$$

che contraddice l'ipotesi di dominanza diagonale stretta. □

Teorema 3.1.10 *Sia $\mathbf{A} \in \mathbb{R}^{n\times n}$ una matrice simmetrica, non singolare e avente la parte diagonale definita positiva, cioè $\mathbf{D} > 0$. Allora il metodo di Gauss-Seidel converge se e solo se $\mathbf{A}$ è s.d.p.*

Dimostrazione. Essendo $\mathbf{A}$ simmetrica, è possibile scrivere lo splitting come $\mathbf{A} = -\mathbf{E} + \mathbf{D} - \mathbf{E}^T$. Da $\mathbf{E}^T = -\mathbf{E} + \mathbf{D} - \mathbf{A}$ segue

$$\mathbf{B}_{GS} = (-\mathbf{E}+\mathbf{D})^{-1}\mathbf{E}^T = \mathbf{I} - (-\mathbf{E}+\mathbf{D})^{-1}\mathbf{A} = \mathbf{I} - \mathbf{H},$$

dove è stato posto $\mathbf{H} := (-\mathbf{E}+\mathbf{D})^{-1}\mathbf{A}$. Si noti che $\mathbf{H}$ è non singolare. Dimostriamo innanzi tutto che $\mathbf{A} - \mathbf{B}_{GS}^T \mathbf{A} \mathbf{B}_{GS} > 0$. Infatti, notando che $\mathbf{A}\mathbf{H}^{-1} = (-\mathbf{E}+\mathbf{D})$, si ha

$$\begin{aligned}
\mathbf{A} - \mathbf{B}_{GS}^T \mathbf{A} \mathbf{B}_{GS} &= \mathbf{A} - (\mathbf{I}-\mathbf{H})^T \mathbf{A} (\mathbf{I}-\mathbf{H}) = \mathbf{A} + \mathbf{H}^T\mathbf{A} - \mathbf{A} - \mathbf{H}^T\mathbf{A}\mathbf{H} + \mathbf{A}\mathbf{H} \\
&= \mathbf{H}^T\mathbf{A} - \mathbf{H}^T\mathbf{A}\mathbf{H} + \mathbf{A}\mathbf{H} = \mathbf{H}^T(\mathbf{A}\mathbf{H}^{-1} - \mathbf{A} + \mathbf{H}^{-T}\mathbf{A})\mathbf{H} \\
&= \mathbf{H}^T((\mathbf{D}-\mathbf{E}) - \mathbf{A} + (\mathbf{D}-\mathbf{E})^T)\mathbf{H} = \mathbf{H}^T\mathbf{D}\mathbf{H}.
\end{aligned}$$

Dall'ipotesi che $\mathbf{D} > 0$, segue che per ogni $x \neq 0$, e $y = \mathbf{H}x \neq 0$ vale $x^T\mathbf{H}^T\mathbf{D}\mathbf{H}x = y^T\mathbf{D}y > 0$, da cui possiamo concludere che $\mathbf{A} - \mathbf{B}_{GS}^T\mathbf{A}\mathbf{B}_{GS} > 0$.

$\boxed{\Leftarrow}$ Supponiamo che $\mathbf{A}$ sia s.d.p. e sia $(\widehat{\lambda}, \widehat{x})$ un'autocoppia di $\mathbf{B}_{GS}$ tale che $\widehat{\lambda}$ sia il più grande autovalore in modulo, cosicchè $|\widehat{\lambda}| = \rho(\mathbf{B}_{GS})$. Dunque

$$0 < \widehat{x}^*\mathbf{A}\widehat{x} - \widehat{x}^*\mathbf{B}^T\mathbf{A}\mathbf{B}\widehat{x} = \widehat{x}^*\mathbf{A}\widehat{x} - |\widehat{\lambda}|^2\widehat{x}^*\mathbf{A}\widehat{x} = (1-|\widehat{\lambda}|^2)\widehat{x}^*\mathbf{A}\widehat{x}.$$

Avendo supposto $\mathbf{A}$ s.d.p., si ha quindi $1 - |\widehat{\lambda}|^2 > 0$, cioè $\rho(\mathbf{B}_{GS}) < 1$. La convergenza del metodo segue dal Teorema 3.1.2.

$\boxed{\Rightarrow}$ Supponiamo che il metodo sia convergente. Consideriamo la relazione per i vettori errore, $e_k = \mathbf{B}_{GS} e_{k-1}$. Dalla positività di $\mathbf{A} - \mathbf{B}_{GS}^T \mathbf{A} \mathbf{B}_{GS}$ segue

$$0 < e_{k-1}^T (\mathbf{A} - \mathbf{B}_{GS}^T \mathbf{A} \mathbf{B}_{GS}) e_{k-1} = e_{k-1}^T \mathbf{A} e_{k-1} - e_{k-1}^T \mathbf{B}_{GS}^T \mathbf{A} \mathbf{B}_{GS} e_{k-1} = e_{k-1}^T \mathbf{A} e_{k-1} - e_k^T \mathbf{A} e_k,$$

cioè la successione $\{e_k^T \mathbf{A} e_k\}_{k \in \mathbb{N}}$ è monotona strettamente decrescente. Se per assurdo $\mathbf{A}$ non è s.d.p., allora è possibile scegliere un dato iniziale x_0 in modo che l'errore iniziale $e_0 \in \mathbb{R}^n$ soddisfi $e_0^T \mathbf{A} e_0 < 0$, ed i termini della successione decrescente saranno ancora più negativi, per cui si avrà $e_k^T \mathbf{A} e_k \nrightarrow 0$, cioè il metodo non converge, arrivando ad una contraddizione. $\square$

Segue dalla dimostrazione precedente che se $\mathbf{A}$ è simmetrica e definita positiva, allora la convergenza del metodo di Gauss-Seidel è monotòna in $\|\cdot\|_{\mathbf{A}}$. Infine, vale il seguente risultato.

Teorema 3.1.11 *Se $\mathbf{A}$ è tridiagonale, allora $\rho(\mathbf{B}_{GS}) = \rho^2(\mathbf{B}_J)$.*

Dim. Iniziamo con una osservazione che sarà usata in seguito: date le parametrizzazioni

$$A(\mu) := \begin{bmatrix} a_1 & c_1/\mu & & & \\ b_2\mu & a_2 & c_2/\mu & & \\ & b_3\mu & a_3 & c_3/\mu & \\ \ddots & \ddots & \ddots & \ddots & \ddots \\ & & & b_n\mu & a_n \end{bmatrix} \text{ e } Q(\mu) := \mathrm{diag}(\mu, \mu^2, \ldots, \mu^n),$$

valgono $A(1) = A$ e $A(\mu) = Q(\mu) A(1) (Q(\mu))^{-1}$. Quindi

$$\det(A(\mu)) = \det(A(1)), \quad \forall \mu \neq 0. \tag{3.7}$$

Venendo alla dimostrazione, notiamo che gli autovalori di $B_J = D^{-1}(E+F)$ sono le radici del polinomio caratteristico $p_J(\lambda) = \det(B_J - \lambda I)$. Siccome $D^{-1}(E+F) - \lambda I = -D^{-1}(\lambda D - E - F)$, questi sono anche le radici del polinomio $q_J(\lambda) = \det(\lambda D - E - U)$.

In modo analogo, scrivendo $(D-E)^{-1}F - \lambda I = -(D-E)^{-1}(\lambda D - \lambda E - F)$, segue che le radici di $p_{GS}(\lambda) = \det(B_{GS} - \lambda I)$ sono anche le radici di $q_{GS}(\lambda) = \det(\lambda D - \lambda E - F)$. Dalla (3.7) con $\mu = 1/\lambda$ segue, per $\lambda \neq 0$, che

$$q_{GS}(\lambda^2) \quad = \quad \det(\lambda^2 D - \lambda^2 E - F) = \det(\lambda^2 D - \lambda E - \lambda F) = \lambda^n \det(\lambda D - E - F) = \lambda^n q_J(\lambda).$$

Per continuità di q_{GS}, la relazione vale anche per $\lambda = 0$. Quindi $q_{GS}(\lambda^2) = \lambda^n q_J(\lambda)$, da cui segue il risultato. $\square$

Il Teorema 3.1.11 mostra che, nel caso di matrici tridiagonali, i metodi di Jacobi e Gauss-Seidel convergono o divergono entrambi e, nel caso in cui convergano, il metodo di Gauss-Seidel lo farà a velocità doppia rispetto al quella del metodo di Jacobi.

3.1.4 Il metodo di rilassamento successivo (SOR)

A partire dal metodo di Gauss-Seidel, è possibile definire un metodo più generale, chiamato metodo di rilassamento successivo (SOR), introducendo un parametro $\omega \neq 0$ detto *parametro di rilassamento*. Aggiungendo e sottraendo $\frac{1}{\omega}\mathbf{D}$, è possibile introdurre il seguente splitting:

$$\mathbf{A} = \mathbf{P} - \mathbf{N} = \frac{1}{\omega}\mathbf{D} - \mathbf{E} - \left(\left(\frac{1}{\omega} - 1 \right) \mathbf{D} + \mathbf{F} \right),$$

dove ancora una volta, $\mathbf{P}$ è non singolare se e solo se $\mathbf{D}$ è non singolare. La matrice di iterazione del metodo SOR risulta essere

$$\begin{aligned} \mathbf{B}_{SOR}(\omega) &= \left(\frac{1}{\omega}\mathbf{D} - \mathbf{E} \right)^{-1} \left(\left(\frac{1}{\omega} - 1 \right) \mathbf{D} + \mathbf{F} \right) = \left(\frac{1}{\omega}\mathbf{I} - \mathbf{D}^{-1}\mathbf{E} \right)^{-1} \left(\left(\frac{1}{\omega} - 1 \right) \mathbf{I} + \mathbf{D}^{-1}\mathbf{F} \right) \\ &= (\mathbf{I} - \omega \mathbf{D}^{-1}\mathbf{E})^{-1} \left((1-\omega)\,\mathbf{I} + \omega \mathbf{D}^{-1}\mathbf{F} \right), \end{aligned}$$

mentre l'algoritmo è dato da:

Fissato $x_0 \in \mathbb{R}^n$ e $r_0 = b - Ax_0$
Per $k = 0, 1, 2, \ldots,$
$x_{k+1} = x_k + \left(\frac{1}{\omega}\mathbf{D} - \mathbf{E}\right)^{-1} r_k,,$ n flops + risoluz.triang.inf.
$r_{k+1} = b - \mathbf{A}x_{k+1},$ n flops + Mxv
Se convergente stop $2n$ (norma del residuo)
end

La matrice di iterazione del metodo dipende dalla scelta del parametro ω, e vale $\mathbf{B}_{SOR}(1) \equiv \mathbf{B}_{GS}$. Si dice che il metodo è *sottorilassato* se $\omega \in (0, 1)$, mentre che il metodo è *sovrarilassato* se $\omega > 1$.

Osservazione. L'algoritmo può anche essere interpretato da un punto di vista geometrico. Infatti, supponendo che $x_{k+1} = x_k + \mathbf{P}_{GS}^{-1} r_k$ sia l'iterazione di Gauss-Seidel, l'iterata x_{k+1} rappresenta un punto sulla retta passante per x_k e nella direzione di $\mathbf{P}_{GS}^{-1} r_k$. Mediante la modifica $x_{k+1} = x_k + \omega \mathbf{P}_{GS}^{-1} r_k$, il metodo SOR modifica la scelta del punto sulla stessa retta, prendendo un punto più vicino o più lontano ad x_k, a seconda che ω sia minore o maggiore di 1. Riscrivendo esplicitamente $\mathbf{P}_{GS}$ ed r_k si arriva all'iterazione SOR (si veda p.261 del testo "Metodi Numerici per l'Algebra Lineare", Bini, Capovani, Menchi, Zanichelli, 1988). Infatti, sia $x_{k+1} = x_k + \mathbf{P}_{GS}^{-1} r_k =: x_k + v_k$. Dato che per Gauss-Seidel si ha $(\mathbf{D} - \mathbf{E})x_{k+1} = \mathbf{F}x_k + b$, cioè $\mathbf{D}x_{k+1} = \mathbf{E}x_{k+1} + \mathbf{F}x_k + b$, per cui $x_{k+1} = \mathbf{D}^{-1}(\mathbf{E}x_{k+1} + \mathbf{F}x_k + b)$, allora per v_k vale

$$v_k = x_{k+1} - x_k = \mathbf{D}^{-1}(\mathbf{E}x_{k+1} + \mathbf{F}x_k + b) - x_k.$$

Per ottenere SOR, si sceglie ω e si definisce $x_{k+1} = x_k + \omega v_k$. Sostituendo v_k si ottiene

$$\begin{aligned} x_{k+1} &= x_k + \omega\mathbf{D}^{-1}(\mathbf{E}x_{k+1} + \mathbf{F}x_k + b) - \omega x_k \\ &= (1-\omega)x_k + \omega\mathbf{D}^{-1}(\mathbf{E}x_{k+1} + \mathbf{F}x_k + b). \end{aligned}$$

Portando a sinistra il termine in x_{k+1} si ha
$(I - \omega\mathbf{D}^{-1}\mathbf{E})x_{k+1} = (1-\omega)x_k + \omega\mathbf{D}^{-1}(\mathbf{F}x_k + b)$ cioè, moltiplicando per $\mathbf{D}$ da sinistra,

$$(\mathbf{D} - \omega\mathbf{E})x_{k+1} = (1-\omega)\mathbf{D}x_k + \omega\mathbf{F}x_k + \omega b.$$

Dividendo per ω si ottiene l'iterazione SOR:

$$(1/\omega\mathbf{D} - \mathbf{E})x_{k+1} = ((1/\omega - 1)\mathbf{D} + \mathbf{F})x_k + b.$$

3.1.5 Risultati di convergenza per il metodo SOR

Riportiamo di seguito alcuni risultati di convergenza per il metodo SOR. Rimandiamo alla ricca letteratura per ulteriori approfondimenti.

Teorema 3.1.12 (di Kahan) *La condizione $\omega \in (0, 2)$ è necessaria per la convergenza del metodo SOR in quanto*

$$\rho(\mathbf{B}_{SOR}(\omega)) \geq |\omega - 1|.$$

Dimostrazione. Siano $\{\lambda_i\}_{i=1,\ldots,n}$ gli autovalori della matrice di iterazione $\mathbf{B}_{SOR}(\omega)$. Allora

$$\begin{aligned} \prod_{i=1}^{n} \lambda_i &= \det\left(\mathbf{B}_{SOR}(\omega)\right) = \det\left(\left(\mathbf{I} - \omega\mathbf{D}^{-1}\mathbf{E}\right)^{-1}\left((1-\omega)\,\mathbf{I} + \omega\mathbf{D}^{-1}\mathbf{F}\right)\right) \\ &= \det\left(\left(\mathbf{I} - \omega\mathbf{D}^{-1}\mathbf{E}\right)^{-1}\right) \det\left((1-\omega)\,\mathbf{I} + \omega\mathbf{D}^{-1}\mathbf{F}\right). \end{aligned}$$

Grazie alla struttura strettamente triangolare risp. inferiore e superiore di $\mathbf{D}^{-1}\mathbf{E}$, $\mathbf{D}^{-1}\mathbf{F}$, si ha

$$\det(\left(\left(\mathbf{I}-\omega\mathbf{D}^{-1}\mathbf{E}\right)^{-1}\right)=1, \qquad \det\left((1-\omega)\,\mathbf{I}+\omega\mathbf{D}^{-1}\mathbf{F}\right)=\prod_{i=1}^{n}(1-\omega)=(1-\omega)^n.$$

Si ottiene quindi $\prod_{i=1}^{n}\lambda_i=(1-\omega)^n$, che implica

$$\rho(\mathbf{B}_{SOR}(\omega))\geq\left|\sqrt[n]{\prod_{i=1}^{n}\lambda_i}\right|=|1-\omega|,$$

da cui segue la tesi. □

Teorema 3.1.13 (di Ostrowski) *Sia* $\mathbf{A}$ *s.d.p., allora il metodo SOR converge se e solo se* $\omega\in(0,2)$. *In più la convergenza è monotona in* $\|\cdot\|_{\mathbf{A}}$.

La scelta del parametro di rilassamento ω è cruciale per una buona velocità di convergenza del metodo SOR. Il problema della determinazione del valore ω_{opt} in corrispondenza del quale la velocità di convergenza sia la più elevata possibile (cioè ω_{opt} per cui $\rho(B_{SOR})$ sia minimo) è assai complesso e se ne conoscono soluzioni soddisfacenti solo in casi particolari.

Teorema 3.1.14 *Sia* $\mathbf{A}$ *s.d.p. e tridiagonale. Supponiamo che* $\rho(\mathbf{B}_J)<1$, *dove* $\mathbf{B}_J$ *indica la matrice di iterazione del metodo di Jacobi. Allora il metodo SOR converge per ogni iterato iniziale* $x_0\in\mathbb{R}^n$ *se* $\omega\in(0,2)$ *e in tal caso*

$$\omega_{opt}=\frac{2}{1+\sqrt{1-\rho^2(\mathbf{B}_J)}}.$$

Per tale ω_{opt} *si ha inoltre che*

$$\rho(\mathbf{B}_{SOR}(\omega_{opt}))=\frac{1-\sqrt{1-\rho^2(\mathbf{B}_J)}}{1+\sqrt{1-\rho^2(\mathbf{B}_J)}}=\omega_{opt}-1.$$

Anche se $\mathbf{A}$ è una matrice simmetrica, i metodi di Gauss-Seidel e SOR generano matrici di iterazione non necessariamente simmetriche. Esistono tuttavia varianti di questi metodi (es. il metodo SSOR) che generano matrici di iterazione simmetriche, partendo da matrici dei coefficienti simmetriche.

3.2 Il metodo dei gradienti coniugati

Il metodo dei gradienti coniugati (CG) è un esempio di metodo iterativo non stazionario per la risoluzione del sistema lineare $\mathbf{A}x=b$, con $\mathbf{A}\in\mathbb{R}^{n\times n}$ simmetrica e definita positiva, e $b\in\mathbb{R}^n$, in quanto determina una successione di approssimazioni $\{x_k\}_{k\geq0}$ mediante una iterazione del tipo $x_{k+1}=x_k+\alpha_k p_k$, dove il coefficiente α_k viene ricalcolato ad ogni iterazione. L'algoritmo può essere visto come un metodo per la risoluzione di un problema di minimo di un funzionale. Consideriamo la seguente funzione convessa,

$$\Phi:\mathbb{R}^n\to\mathbb{R}, \qquad \Phi:\,x\mapsto\frac{1}{2}x^T\mathbf{A}x-b^Tx, \tag{3.8}$$

ed il problema di minimo:

$$\min_{x\in\mathbb{R}^n}\Phi(x).$$

Questo problema ha come unica soluzione $x_* = \mathbf{A}^{-1}b = \arg\min_{x\in\mathbb{R}^n} \Phi(x)$, poichè vale $\nabla\Phi(x_*) = 0$ dove $\nabla\Phi(x) = -b + \mathbf{A}x$; inoltre, se x_* è soluzione del sistema, si verifica che per ogni altro $y \in \mathbb{R}^n$ si ha $\Phi(x_* + y) = \Phi(x_*) + \frac{1}{2}y^T Ay > \Phi(x_*)$ e quindi $\Phi(x_*)$ è l'unico minimo.

Definiamo quindi una successione di soluzioni approssimate $\{x_k\}_{k\in\mathbb{N}}$ tale che

$$x_{k+1} = x_k + \alpha_k p_k, \tag{3.9}$$

con $\alpha_k \in \mathbb{R}$ e $p_k \in \mathbb{R}^n$, dove il vettore p_k è un vettore lungo la direzione di discesa di Φ, cioè tale che

$$p_k^T \nabla\Phi(x_k) < 0, \quad \forall\, k.$$

Con tale scelta, x_{k+1} è sulla retta per x_k e nella direzione di p_k. Dall'iterazione per x_{k+1} si ottiene anche l'iterazione per il residuo $r_{k+1} = b - \mathbf{A}x_{k+1}$, cioè sostituendo,

$$r_{k+1} = r_k - \alpha_k \mathbf{A}p_k, \qquad r_0 = b - \mathbf{A}x_0.$$

Rimangono quindi da determinare α_k e p_k. Ogni scelta di p_k definirà un diverso metodo di discesa.

Proposizione 3.2.1 *Per $x_k, p_k \in \mathbb{R}^n$ fissati, Sia $x_{k+1} = x_k + \alpha_k p_k$. La scelta $\alpha_k = \frac{r_k^T p_k}{p_k^T \mathbf{A} p_k}$, soddisfa*

$$\Phi(x_{k+1}) = \min_{\alpha\in\mathbb{R}} \Phi(x_k + \alpha p_k).$$

Dimostrazione. Si ha $\Phi(x_k + \alpha p_k) = \dfrac{1}{2}(x_k + \alpha p_k)^T \mathbf{A}(x_k + \alpha p_k) - b^T(x_k + \alpha p_k)$. Inoltre,

$$\frac{\mathrm{d}\Phi}{\mathrm{d}\alpha}(x_k + \alpha p_k) = \frac{1}{2}p_k^T \mathbf{A}(x_k + \alpha p_k) + \frac{1}{2}(x_k + \alpha p_k)^T \mathbf{A}p_k - b^T p_k = (x_k + \alpha p_k)^T \mathbf{A}p_k - b^T p_k,$$

dove è stata sfruttata la simmetria di $\mathbf{A}$. Imponendo $\frac{\mathrm{d}\Phi}{\mathrm{d}\alpha} = 0$, si ha $\alpha_k = \frac{(b^T - x_k^T\mathbf{A})p_k}{p_k^T\mathbf{A}p_k} = \frac{r_k^T p_k}{p_k^T \mathbf{A} p_k}$. Il fatto che $\mathbf{A}$ sa simmetrica e definita positiva assicura che α_k sia un minimo, valutando la derivata seconda. □

Questa scelta di α_k nella proposizione determina il minimo di Φ sulla retta per x_k e nella direzione di p_k, e quindi è ottima, almeno localmente.

Si noti che α_k è ben definito, in quanto il denominatore è sempre diverso da zero (è strettamente positivo), grazie all'ipotesi su $\mathbf{A}$ di essere simmetrica e definita positiva. Inoltre vale

$$r_k^T p_k = -\nabla\Phi(x_k)^T p_k > 0 \qquad \Rightarrow \quad \alpha_k > 0,$$

poichè p_k è un vettore nella direzione di discesa di Φ. Infine, per come è stato scelto α_k, si ha

$$p_k^T r_{k+1} = p_k^T r_k - \alpha_k p_k^T \mathbf{A} p_k = 0, \tag{3.10}$$

cioè il nuovo residuo è ortogonale alla direzione precedente.

Come già detto, i metodi di discesa differiscono nella scelta della direzione p_k; ad esempio, nel metodo di "discesa ripida", p_k corrisponde alla direzione di massima pendenza, quindi $p_k = -\nabla\Phi(x_k) = r_k$. Per la proprietà (3.10), le direzioni successive saranno quindi ortogonali tra loro.

Nel metodo dei gradienti coniugati, invece, p_k è scelto in modo da soddisfare una condizione di $\mathbf{A}$-coniugazione: le direzioni vengono determinate mediante l'iterazione

$$p_0 = r_0, \qquad p_{k+1} = r_{k+1} + \beta_k p_k, \quad \text{per } k \geq 0, \tag{3.11}$$

dove β_k è ottenuta in modo che p_k e p_{k+1} siano $\mathbf{A}$-coniugati, cioè

$$p_{k+1}^T \mathbf{A} p_k = 0. \tag{3.12}$$

Sostituendo p_{k+1} in (3.12) si ottiene

$$0 = p_{k+1}^T \mathbf{A} p_k = (r_{k+1} + \beta_k p_k)^T \mathbf{A} p_k = r_{k+1}^T \mathbf{A} p_k + \beta_k p_k^T \mathbf{A} p_k, \qquad \Rightarrow \beta_k = -\frac{r_{k+1}^T \mathbf{A} p_k}{p_k^T \mathbf{A} p_k}. \quad (3.13)$$

Con questa relazione l'algoritmo dei gradienti coniugati è formalmente completato, in quanto tutte le quantità sono state definite. Nel seguito, comunque, vengono evidenziate alcune proprietà che permettono un calcolo più conveniente dei coefficienti α_k e β_k.

Le direzioni p_k definite in (3.11) sono effettivamente direzioni di discesa per la funzione Φ definita in (3.8). Infatti, fintanto che x_{k+1} non è la soluzione esatta,

$$\begin{aligned} p_{k+1}^T \nabla\Phi(x_{k+1}) &= -p_{k+1}^T r_{k+1} = -(r_{k+1} + \beta_k p_k)^T r_{k+1} \\ &= -(r_{k+1}^T r_{k+1} + \beta_k \underbrace{p_k^T r_{k+1}}_{\substack{=0 \\ \text{da (3.10)}}}) = -r_{k+1}^T r_{k+1} = -\|r_{k+1}\|^2 < 0, \end{aligned}$$

Con questa proprietà è possibile ottenere una diversa espressione per α_k, che risulterà più vantaggiosa da un punto di vista algoritmico:

$$\alpha_k = \frac{p_k^T r_k}{p_k^T \mathbf{A} p_k} = \frac{r_k^T r_k}{p_k^T \mathbf{A} p_k}.$$

Per i residui vale inoltre $r_k^T r_{k-1} = 0$, infatti

$$\begin{aligned} r_k^T r_{k-1} &= r_k^T (p_{k-1} - \beta_{k-2} p_{k-2}) = -\beta_{k-2} r_k^T p_{k-2} \\ &= -\beta_{k-2} (r_{k-1}^T p_{k-2} - \alpha_{k-1} (\mathbf{A} p_{k-1})^T p_{k-2}) = 0, \end{aligned}$$

dove sono state usate le proprietà $r_{k-1}^T p_{k-2} = 0$ in (3.10) e $(\mathbf{A} p_{k-1})^T p_{k-2} = 0$ in (3.12).

Le relazioni di ortogonalità determinate qui sopra e in (3.12) sono in effetti più generali, ed infatti vale il seguente risultato.

Teorema 3.2.2 *Sia $r_0 = b - \mathbf{A} x_0$ il residuo iniziale e supponiamo che $r_k \neq 0$ per ogni $k \leq \widehat{k}$. Allora*

$$r_k^T r_j = 0, \quad \forall\, k \neq j, \quad k, j = 0, \ldots, \widehat{k},$$

e

$$p_k^T \mathbf{A} p_j = 0, \quad \forall\, k \neq j, \quad k, j = 0, \ldots, \widehat{k}.$$

Il teorema precedente mostra che tutte le direzioni sono tra loro $\mathbf{A}$-coniugate, e che i residui sono tra loro ortogonali, o coniugati. Dato che i residui hanno la stessa direzione dei gradienti, questa proprietà giustifica il nome del metodo "dei gradienti coniugati".

Queste importanti proprietà permettono di dare una diversa espressione per β_k: da $r_{k+1} = r_k - \alpha_k \mathbf{A} p_k$ si ottiene

$$\beta_k = -\frac{r_{k+1}^T \mathbf{A} p_k}{p_k^T \mathbf{A} p_k} = \frac{r_{k+1}^T (\frac{1}{\alpha_k}(r_{k+1} - r_k))}{p_k^T \mathbf{A} p_k} = \frac{1}{\alpha_k} \frac{r_{k+1}^T r_{k+1}}{p_k^T \mathbf{A} p_k} = \frac{p_k^T \mathbf{A} p_k}{r_k^T r_k} \frac{r_{k+1}^T r_{k+1}}{p_k^T \mathbf{A} p_k} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}.$$

La versione finale dell'algoritmo dei gradienti coniugati è quindi data dalla seguente iterazione:

Algoritmo dei Gradienti Coniugati.
Fissato $x_0 \in \mathbb{R}^n$, e posto $r_0 = b - \mathbf{A} x_0$, $p_0 = r_0$ e $\beta_0 = 0$, $\rho_0 = r_0^T r_0$
Per $k = 0, 1, \ldots,$

$\alpha_k = \frac{\rho_k}{p_k^T \mathbf{A} p_k}$, $\quad 2n$ flops + 1 Mxv
$x_{k+1} = x_k + \alpha_k p_k$, $\quad 2n$ flops
$r_{k+1} = r_k - \alpha_k \mathbf{A} p_k$, $\quad 2n$ flops
$\rho_{k+1} = r_{k+1}^T r_{k+1}$, $\quad \beta_k = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k} = \frac{\rho_{k+1}}{\rho_k}$, $\quad 2n$ flops
$p_{k+1} = r_{k+1} + \beta_k p_k$, $\quad 2n$ flops
end

L'algoritmo ha un costo computazionale di $10n$ operazioni floating point per iterazione, a cui va aggiunto il costo di una moltiplicazione matrice-per-vettore (il prodotto compare due volte, ma includendo un ulteriore vettore di "lavoro" $w = \mathbf{A}p_k$, è sufficiente richiamare w nel calcolo del residuo r_{k+1} invece di ricalcolare il prodotto).

Il seguente risultato assicura che in aritmetica esatta, il metodo termina in al più n passi.

Teorema 3.2.3 *Sia $K_k = span\{p_0, \ldots, p_{k-1}\}$. Allora $\Phi(x_k) = \min_{x \in K_k} \Phi(x)$.*

Dopo n iterazioni, il sottospazio K_k coincide con tutto $\mathbb{R}^n$, quindi la soluzione approssimata corrente coincide con il minimo in tutto lo spazio, e quindi con la soluzione esatta. In pratica l'algoritmo permette di determinare una soluzione accurata molto prima di n passi, e quindi viene solitamente inserito un criterio d'arresto, dopo aver calcolato il residuo. Come per gli altri metodi iterativi, questo criterio può essere basato sulla norma del residuo, cioè

$$\frac{\|r_{k+1}\|}{\|r_0\|} < \texttt{tol}$$

dove `tol` è una tolleranza fissata a priori. In generale, è anche opportuno prevedere un massimo numero di iterazioni, per terminare l'iterazione nel caso di convergenza troppo lenta.

Osservazione. Guardando attentamente la definizione delle successioni $\{r_k\}$, $\{p_k\}$, si ottiene la seguente relazione:

$$\text{span}\{p_0, p_1, \ldots, p_{k-1}\} = \text{span}\{r_0, r_1, \ldots, r_{k-1}\} = \text{span}\{r_0, \mathbf{A}r_0, \ldots, \mathbf{A}^{k-1} r_0\},$$

dove lo spazio $K_k(\mathbf{A}, r_0) = \text{span}\{r_0, \mathbf{A}r_0, \ldots, \mathbf{A}^{k-1} r_0\}$ è detto "sottospazio di Krylov" di dimensione k associato ad $\mathbf{A}$ e generato da r_0. La dimostrazione esplicita delle due uguaglianze può essere fatta per induzione.

Il seguente risultato mostra una condizione sufficiente perchè il metodo converga in meno di n passi, in aritmetica esatta.

Osservazione. Se $K_k(\mathbf{A}, r_0)$ è invariante per A, allora $x_* \in K_k(\mathbf{A}, r_0)$, cioè la soluzione esatta è contenuta nello spazio, cosicchè $x_k = x_*$.

Dim. Per semplicità supponiamo $x_0 = 0$, per cui $r_0 = b$. L'ipotesi assicura che $AK_k(\mathbf{A}, r_0) \subseteq K_k(\mathbf{A}, r_0)$. Per il residuo $r_k = b - Ax_k$ vale quindi $r_k \in K_k(\mathbf{A}, r_0)$, dato che $b = r_0 \in K_k(\mathbf{A}, r_0)$ e $Ax_k \in AK_k(\mathbf{A}, r_0) \subseteq K_k(\mathbf{A}, r_0)$. D'altra parte, $r_k \perp r_j$ per ogni $j < k$, e $\text{span}\{r_0, r_1, \ldots, r_{k-1}\} = K_k(\mathbf{A}, r_0)$, quindi dev'essere necessariamente $r_k = 0$, da cui segue la tesi.

Riportiamo un importante risultato di convergenza molto generale, che si applica a qualsiasi matrice $\mathbf{A}$ simmetrica e definita positiva, e per qualsiasi dato iniziale x_0.

Teorema 3.2.4 *Sia $\mathbf{A} \in \mathbb{R}^{n \times n}$ s.d.p. e sia x_* la soluzione esatta del sistema $\mathbf{A}x = b$. Allora il metodo dei gradienti coniugati soddisfa*

$$\|x_* - x_k\|_{\mathbf{A}} = \min_{x \in K_k} \|x_* - x\|_{\mathbf{A}},$$

cioè minimizza l'errore in norma $\mathbf{A}$ *(norma energia) nel sottospazio di Krylov* K_k*. Inoltre*

$$\|x_* - x_k\|_{\mathbf{A}} \leq 2 \left(\frac{\sqrt{\kappa_2(\mathbf{A})} - 1}{\sqrt{\kappa_2(\mathbf{A})} + 1} \right)^k \|x_* - x_0\|_{\mathbf{A}}.$$

Il primo risultato del teorema mostra il legame tra l'algoritmo ed il problema di minimo di partenza, in quanto la minimizzazione del funzionale è equivalente alla minimizzazione dell'errore in norma energia. Infatti si ha

$$\|x_* - x\|_{\mathbf{A}}^2 = (x_* - x)^T \mathbf{A}(x_* - x) = \ldots = 2(\frac{1}{2} x^T \mathbf{A} x - x^T b) + x_*^T \mathbf{A} x_* = 2\Phi(x) + x_*^T \mathbf{A} x_*.$$

Visto che $x_*^T \mathbf{A} x_*$ è positivo e non dipende da x, minimizzare $\|x_* - x\|_{\mathbf{A}}$ con $x \in K_k$ corrisponde a minimizzare $\Phi(x)$ per $x \in K_k$.

Il teorema propone inoltre una stima dell'errore dopo k iterazioni, che dipende dal numero di condizionamento della matrice dei coefficienti, $\kappa_2(\mathbf{A}) = \text{cond}_2(\mathbf{A}) = \lambda_{\max}/\lambda_{\min}$. Se la matrice è ben condizionata, allora il metodo converge in poche iterazioni, mentre potrebbe convergere molto lentamente nel caso di una matrice $\mathbf{A}$ molto mal condizionata, cioè con $\kappa_2(\mathbf{A}) \gg 1$.

Si noti che per il metodo di discesa ripida vale il seguente risultato

$$\|x_* - x_k\|_{\mathbf{A}} \leq 2 \left(\frac{\kappa_2(\mathbf{A}) - 1}{\kappa_2(\mathbf{A}) + 1} \right)^k \|x_* - x_0\|_{\mathbf{A}},$$

che mostra che tale metodo può essere molto più lento del metodo dei gradienti coniugati.

Nel caso in cui la matrice dei coefficienti $\mathbf{A}$ non sia s.d.p., esistono altri metodi basati sul sottospazio di Krylov K_k. Per esempio

- Nel caso in cui $\mathbf{A}$ non sia simmetrica definita positiva, possiamo utilizzare i gradienti coniugati applicandoli al sistema di *equazioni normali* $\mathbf{A}^T \mathbf{A} x = \mathbf{A}^T b$. In questo modo la matrice dei coefficienti risulta simmetrica e definita positiva. D'altra parte, il suo numero di condizionamento cresce. Per esempio, per $\mathbf{A}$ simmetrica ma indefinita, vale $\kappa(\mathbf{A}^T \mathbf{A}) = \kappa(\mathbf{A})^2$.
- MINRES, SYMLQ, per sistemi simmetrici ma non definiti positivi;
- BICG, GMRES, ecc: per sistemi non simmetrici.

Capitolo 4

La fattorizzazione QR

4.1 Introduzione

In questo capitolo trattiamo la fattorizzazione QR, che è uno strumento di fondamentale importanza in moltissimi ambiti dell'analisi numerica. Più precisamente, data una generica matrice $\mathbf{A} \in \mathbb{R}^{n\times m}$ con $n \geq m$, allora $\mathbf{A}$ può essere scritta come prodotto di due matrici come segue:

$$\mathbf{A} = \mathbf{QR} = [\mathbf{Q}_1, \mathbf{Q}_2] \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix} = \mathbf{Q}_1\mathbf{R}_1, \qquad \mathbf{R}_1 \in \mathbb{R}^{m\times m},$$

dove $\mathbf{Q}$ è una matrice $n \times n$ reale ortogonale, $\mathbf{Q} = [\mathbf{Q}_1, \mathbf{Q}_2]$ con $\mathbf{Q}_1 \in \mathbb{R}^{n\times m}$ e $\mathbf{Q}_2 \in \mathbb{R}^{n\times(n-m)}$, dove le colonne di $\mathbf{Q}_1$, $\mathbf{Q}_2$ sono ortonormali. Inoltre, $\mathbf{R}_1$ è triangolare superiore ed $\mathbf{R}$ ha le stesse dimensioni di $\mathbf{A}$. La versione $\mathbf{A} = \mathbf{Q}_1\mathbf{R}_1$ prende il nome di fattorizzazione QR ridotta (o *skinny*, o *economy-size*), perchè non genera nè memorizza[1] la matrice $\mathbf{Q}_2$.

La fattorizzazione permette di evidenziare se $\mathbf{A}$ ha rango massimo. Infatti, se $\mathbf{R}_1$ è singolare, cioè ha elementi diagonali nulli, allora ci sono colonne di $\mathbf{A}$ che sono linearmente dipendenti. D'altra parte, se $\mathbf{R}_1$ è non singolare, allora $\mathbf{A}$ ha rango massimo e Range($\mathbf{Q}_1$) = Range($\mathbf{A}$). In seguito supporremo che $\mathbf{R}_1$ sia non singolare.

La fattorizzazione QR permette di determinare una base ortogonale per lo spazio immagine di $\mathbf{A}$, dato che vale Range($\mathbf{Q}_1$) = Range($\mathbf{A}$). Vedremo nel paragrafo 4.2 che è possibile determinare questa base con il metodo di Gram-Schmidt, e che quest'ultimo in effetti fornisce una fattorizzazione "ridotta", $\mathbf{A} = \mathbf{Q}_1\mathbf{R}_1$.

Dalla fattorizzazione QR si ha anche $\mathbf{Q}^T\mathbf{A} = \mathbf{R}$, cioè $\mathbf{Q}$ è la matrice che trasforma $\mathbf{A}$ in una matrice triangolare superiore. Questo modo di procedere verrà usato nei metodi numerici per i problemi agli autovalori, in cui $n = m$. Algoritmi per generare la matrice completa $\mathbf{Q}^T$ verranno discussi nel paragrafo 4.3.

Nel caso in cui $\mathbf{A}$ sia invece "larga", cioè $n < m$, la fattorizzazione QR è ancora possibile, e le dimensioni delle matrici risultanti sono conformi. Più precisamente,

$$\mathbf{A} = \mathbf{QR} = \mathbf{Q}[\mathbf{R}_1, \star],$$

[1] si noti che $\mathbf{Q}_2^T\mathbf{A} = 0$, quindi le colonne di $\mathbf{Q}_2$ generano una base per lo spazio nullo (kernel) di $\mathbf{A}^T$ se $\mathbf{R}_1$ è non singolare.

in cui $\mathbf{Q}$ è $n \times n$ ortogonale, $\mathbf{R}_1$ è triangolare superiore, e $\star$ è una matrice piena. Le colonne di $\mathbf{Q}$ generano lo spazio immagine di $\mathbf{A}$, come in precedenza, ma in questo caso Range($\mathbf{A}$) ha al più dimensione n.

La fattorizzazione QR può essere usata per la risoluzione di sistemi lineari con matrice $\mathbf{A} \in \mathbb{R}^{n\times n}$. Infatti, scrivendo $\mathbf{A} = \mathbf{QR}$ con $\mathbf{Q}$ ortogonale e $\mathbf{R} \in \mathbb{R}^{n\times n}$ triangolare superiore, si ha

$$\mathbf{A}x = b \quad \Leftrightarrow \quad \mathbf{QR}x = b \quad \Leftrightarrow \quad \mathbf{R}x = (\mathbf{Q}^T b) \quad \Leftrightarrow \quad x = \mathbf{R}^{-1}(\mathbf{Q}^T b).$$

Oltre al calcolo effettivo della fattorizzazione, questo modo di procedere richiede di effettuare il prodotto $\mathbf{Q}^T b$ (ma questo può essere fatto durante la fattorizzazione stessa, si veda la discussione nel Capitolo 5) e la risoluzione del sistema triangolare superiore. Il costo complessivo è alto, più alto dell'eliminazione Gaussiana, anche se sempre $\mathcal{O}(n^3)$ (si veda il paragrafo 4.3.1 per maggiori dettagli sul costo della fattorizzazione). D'altra parte, il metodo è più stabile dell'eliminazione di Gauss, e può essere preferibile per matrici per cui l'eliminazione di Gauss mostra problemi di accuratezza perchè $\mathbf{A}$ è mal condizionata.

4.2 Ortogonalizzazione di Gram-Schmidt

La ortogonalizzazione di Gram-Schmidt è un algoritmo per trasformare una base di un sottospazio vettoriale di $\mathbb{R}^n$ di dimensione m in una base ortonormale. In pratica, si tratta in effetti di una fattorizzazione QR ridotta, in cui gli elementi di $\mathbf{R}_1$ coincidono con i coefficienti dell'ortogonalizzazione. Più precisamente, sia $\mathbf{A} = [a_1, a_2, \ldots, a_m] \in \mathbb{R}^{n\times m}$ con $n \geq m$. Allora l'algoritmo di Gram-Schmidt per determinare la base ortonormale $\{q_1, \ldots, q_m\}$ procede come segue

$$\begin{aligned}
q_1 &= a_1/\|a_1\| \\
\widehat{q}_2 &= a_2 - q_1(q_1^T a_2) \\
q_2 &= \widehat{q}_2/\|\widehat{q}_2\| \\
\widehat{q}_3 &= a_3 - q_1(q_1^T a_3) - q_2(q_2^T a_3) \\
q_3 &= \widehat{q}_3/\|\widehat{q}_3\| \\
\vdots &\quad \vdots
\end{aligned}$$

Si noti che dato un vettore v, l'operazione $\widehat{v} = v - q_j(q_j^T v)$ con q_j di norma unitaria, "elimina" dal vettore v la direzione del vettore q_j; il vettore risultante $\widehat{v}$ soddisfa quindi $q_j^T \widehat{v} = 0$.

Riscrivendo le espressioni sopra in modo da avere tutti i termini in q_j a destra, si ha

$$\begin{aligned}
a_1 &= q_1\|a_1\| \\
a_2 &= q_1(q_1^T a_2) + q_2\|\widehat{q}_2\| \\
a_3 &= q_1(q_1^T a_3) + q_2(q_2^T a_3) + q_3\|\widehat{q}_3\| \\
\vdots &\quad \vdots
\end{aligned}$$

In forma compatta matriciale, queste relazioni possono essere quindi scritte come

$$[a_1, a_2, a_3, \ldots] = [q_1, q_2, q_3, \ldots] \begin{bmatrix} \|a_1\| & q_1^T a_2 & q_1^T a_3 & \ldots \\ 0 & \|\widehat{q}_2\| & q_2^T a_3 & \ldots \\ 0 & 0 & \|\widehat{q}_3\| & \ldots \\ 0 & 0 & 0 & \ddots \end{bmatrix} = \mathbf{Q}_1\mathbf{R}_1.$$

Dunque nella fattorizzazione QR ridotta risultante, gli elementi di $\mathbf{R}_1$ sono in effetti i coefficienti della ortonormalizzazione. Nel caso in cui si abbia $\widehat{q}_j = 0$ per qualche j, allora, $\|\widehat{q}_j\| = 0$ e $\mathbf{R}_1$ sarà singolare, evidenziando il fatto che le colonne $a_1, \ldots, a_j$ sono linearmente dipendenti. L'algoritmo di Gram-Schmidt quindi termina, non essendo in grado di costruire il prossimo elemento della base ortonormale, q_j.

Da un punto di vista numerico, anche quantità $\|\widehat{q}_j\|$ molto piccole sono di enorme interesse, perchè evidenziano che la matrice $\mathbf{A}$ ha colonne "quasi" linearmente dipendenti. In particolare, se $\|\widehat{q}_j\| = \mathcal{O}(\mathbf{u})$, allora la matrice $\mathbf{A}$ numericamente ha rango inferiore ad m, anche se matematicamente $\mathbf{A}$ ha rango pieno.

Anche se in aritmetica esatta la procedura "classica" di Gram-Schmidt descritta sopra determina una base ortogonale, in aritmetica floating point i vettori così determinati spesso non sono ortogonali (al livello della precisione macchina). È possibile farne una modifica che risulta essere più accurata, anche da un punto di vista numerico, dando luogo alla procedura di Gram-Schmidt *modificata*.

4.3 Matrici ortogonali di trasformazione

Le matrici di trasformazione hanno un ruolo chiave sia nei problemi agli autovalori che in quelli dei minimi quadrati. Nel seguito vengono introdotte le *matrici di riflessione di Householder* e le *matrici di rotazione di Givens*: entrambe queste classi di matrici permettono di annullare specifici elementi, o gruppi di elementi di vettori, e rappresentano uno strumento semplice ma molto potente per la costruzione di altri metodi.

4.3.1 Matrici di riflessione di Householder

Dato un vettore $v \in \mathbb{R}^n$, $v \neq 0$, detto *vettore di Householder*, la *matrice di riflessione di Householder* è definita come segue:

$$\mathbf{P} := \mathbf{I} - \beta v v^T \in \mathbb{R}^{n \times n}, \qquad \text{con } \beta = \frac{2}{\|v\|_2^2}.$$

Si noti che $\mathbf{P}$ è una modifica di rango uno della matrice identità. Nel seguito vengono illustrate alcune sue importanti proprietà.

i) $\mathbf{P}$ *è simmetrica e ortogonale.* La simmetria è evidente. Per l'ortogonalità si ha:

$$\mathbf{P}\mathbf{P}^T = \left(\mathbf{I} - \beta v v^T\right)\left(\mathbf{I} - \beta v v^T\right)^T = \mathbf{I} - \beta v v^T - \beta v v^T + \beta^2 v (v^T v) v^T = \mathbf{I} - 2\beta v v^T + 2\beta v v^T = \mathbf{I},$$

dove si è utilizzato il fatto che $\beta^2 v(v^T v)v^T = (4/\|v\|^4) v \|v\|^2 v^T = 2\beta v v^T$.

ii) $\mathbf{P}$ *è una riflessione.* Sia $x \in \mathbb{R}^n$, allora il vettore $y = \mathbf{P}x$ è il riflesso di x rispetto all'iperpiano $\operatorname{span}\{v\}^\perp$. Infatti ogni vettore $x \in \mathbb{R}^n$ può essere scritto come $x = \gamma v + x_2$ dove $\gamma \in \mathbb{R}$ e $x_2 \perp v$. Quindi

$$\mathbf{P}x = \left(\mathbf{I} - \beta v v^T\right) x = \left(\mathbf{I} - \beta v v^T\right)(\gamma v + x_2) = \gamma v + x_2 - \gamma\beta v \underbrace{v^T v}_{=2/\beta} - \beta v \underbrace{v^T x_2}_{=0} = -\gamma v + x_2.$$

iii) $\mathbf{P}$ *può annullare componenti di un vettore.* Dato x, è possibile determinare una matrice di riflessione $\mathbf{P}$ tale che $\mathbf{P}x = \alpha e_1 = [\alpha, 0, \ldots, 0]^T$ dove $|\alpha| = \|\mathbf{P}x\|_2 = \|x\|_2$; e_1 è il primo vettore della base canonica di $\mathbb{R}^n$. A tal fine, è sufficiente scegliere v come segue (entrambi i segni + o − possono essere usati per α):

$$v := x - \alpha e_1, \quad \text{con } \alpha = \pm\|x\|_2\theta, \quad \text{dove } \theta = \begin{cases} \operatorname{sgn}(x_1), & \text{se } x_1 \neq 0, \\ 1, & \text{se } x_1 = 0. \end{cases}$$

Quindi, $\mathbf{P}x = (\mathbf{I} - \beta vv^T)x = x - \frac{2}{v^T v} vv^T x$. Siccome

$$v^T x = (x - \alpha e_1)^T x = \left(x^T - \alpha e_1^T\right) x = x^T x - \alpha e_1^T x = \|x\|_2^2 - \alpha x_1 = \alpha^2 - \alpha x_1,$$

e

$$\begin{aligned} v^T v &= \left(x^T - \alpha e_1^T\right)(x - \alpha e_1) = x^T x - \alpha x^T e_1 - \alpha e_1^T x + \alpha^2 e_1^T e_1 \\ &= \underbrace{\|x\|_2^2}_{=\alpha^2} - 2\alpha x_1 + \alpha^2 = 2\left(\alpha^2 - \alpha x_1\right), \end{aligned}$$

si ottiene

$$\mathbf{P}x = x - \frac{2}{v^T v} vv^T x = x - \frac{2}{2\left(\alpha^2 - \alpha x_1\right)} v(\alpha^2 - \alpha x_1) = x - v = \alpha e_1.$$

Le matrici di Householder possono essere usate per determinare la fattorizzazione QR di una matrice $\mathbf{A} \in \mathbb{R}^{n \times m}$ sfruttando la proprietà (*iii*) appena descritta. Se a_1 denota la prima colonna di $\mathbf{A}$, allora si determina $\mathbf{P}_1 = \mathbf{I} - \beta^{(1)} vv^T$ con $v = a_1 - \alpha^{(1)} e_1$ e $\beta^{(1)}, \alpha^{(1)}$ definite di conseguenza. Quindi posto $\mathbf{R}^{(0)} = \mathbf{A}$ si ha

$$\mathbf{P}_1 \mathbf{R}^{(0)} = \begin{bmatrix} \alpha^{(1)} & * & \cdots & * \\ 0 & * & \cdots & * \\ \vdots & \vdots & \vdots & \vdots \\ 0 & * & \cdots & * \end{bmatrix} =: \mathbf{R}^{(1)}.$$

Il costo computazionale dell'applicazione di $\mathbf{P}_1$ all'intera matrice $\mathbf{R}^{(0)}$ si calcola come segue: il vettore v ha n componenti, quindi l'applicazione di $\mathbf{P}_1$ ad ogni colonna x, $\mathbf{P}_1 x = x - \beta v(v^T x)$, richiede due prodotti scalari (per β e per $v^T x$), ed una `daxpy`, per un totale di $(2n-1)+(2n-1)+2n = 6n-2$ flops. L'applicazione a tutte le m colonne richiede quindi un costo totale di $(6n-2)m$ flops. **Si noti comunque che β dev'essere calcolata una sola volta, quindi il costo effettivo è di $(4n-2)m + 2n$ flops.**

Si definisce quindi una seconda matrice di Householder per azzerare gli elementi della seconda colonna, sotto la diagonale principale,

$$\mathbf{P}_2 = \begin{bmatrix} 1 & 0 \\ 0 & \widehat{\mathbf{P}}_2 \end{bmatrix}, \quad \widehat{\mathbf{P}}_2 = \mathbf{I} - \beta^{(2)} vv^T \in \mathbb{R}^{(n-1)\times(n-1)}, \quad v = (\mathbf{R}^{(1)})_{2:n,2} - \alpha^{(2)} e_1,$$

con $e_1 \in \mathbb{R}^{n-1}$ e $\beta^{(2)}, \alpha^{(2)}$ di nuovo definite di conseguenza. Quindi

$$\mathbf{P}_2 \mathbf{P}_1 \mathbf{R}^{(0)} = \mathbf{P}_2 \mathbf{R}^{(1)} = \begin{bmatrix} \alpha^{(1)} & * & * & \cdots & * \\ 0 & \alpha^{(2)} & * & \cdots & * \\ 0 & 0 & * & \cdots & * \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & * & \cdots & * \end{bmatrix} =: \mathbf{R}^{(2)}.$$

Si noti che ad ogni passo, la matrice di Householder $\widehat{\mathbf{P}}_j$ ha dimensioni decrescenti, $(n-j+1)\times(n-j+1)$, quindi la matrice di trasformazione usata, $\mathbf{P}_j$, deve essere completata nella parte diagonale in alto con una porzione della matrice identità. Anche la sua applicazione viene limitata ad una

porzione sempre più piccola della matrice $n \times n$. Il costo computazionale associato diminuisce quindi ad ogni passo, ed è di $(4(n-j+1)-2)(m-j)$ flops al passo j. Dopo m passi, si otterrà

$$\mathbf{P}_m \cdots \mathbf{P}_2 \mathbf{P}_1 \mathbf{R}^{(0)} = \begin{bmatrix} \alpha^{(1)} & * & * & \cdots & * \\ 0 & \alpha^{(2)} & * & \cdots & * \\ 0 & 0 & \alpha^{(3)} & \cdots & * \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \alpha^{(m)} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix} =: \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix} =: \mathbf{R}.$$

Posto $\mathbf{Q}^T = \mathbf{P}_m \cdots \mathbf{P}_2 \mathbf{P}_1$, si ottiene quindi $\mathbf{Q}^T \mathbf{A} = \mathbf{R}$. La matrice $\mathbf{Q}^T$ è ortogonale perchè prodotto di matrici ortogonali, per cui $\mathbf{A} = \mathbf{QR}$. Il costo computazionale complessivo per generare $\mathbf{Q}$ e $\mathbf{R}$ risulta quindi essere di $\sum_{j=1}^{m}(4(n-j+1)-2)(m-j) \approx 4\sum_{j=1}^{m}(n-j)(m-j) = 2nm^2 - 2/3m^3 + ...$ flops.

Le matrici di Householder possono essere anche usate per trasformare una matrice $n \times n$ in forma di Hessenberg superiore, mantenendo invariati gli autovalori, mediante applicazione da sinistra e da destra delle stesse trasformazioni. Questa procedura verrà usata nel metodo QR per approssimare la decomposizione di Schur di una matrice. Più precisamente, nel seguito mostriamo che è possibile determinare $\mathbf{Q}$ ortogonale tale che la matrice $\mathbf{Q}^T \mathbf{A} \mathbf{Q} = \widetilde{\mathbf{A}}$ soddisfa $\widetilde{\mathbf{A}}_{ij} = 0$ per $i > j+1$. Infatti, siano

$$\mathbf{P}_j = \begin{bmatrix} I_j & 0 \\ 0 & \widehat{\mathbf{P}}_j \end{bmatrix}$$

le matrici di trasformazione con $\widehat{\mathbf{P}}_j$ matrici di Householder, tali che gli elementi sotto al $(j+1)$-esimo elemento diagonale della colonna j-esima vengano annullati. Per $j = 1$, sia $\mathbf{H}^{(2)} = \mathbf{P}_1 \mathbf{A}$. Notiamo poi che l'applicazione di $\mathbf{P}_1^T = \mathbf{P}_1$ da destra non distrugge la forma di Hessenberg. Infatti

$$\mathbf{H}^{(2)} \mathbf{P}_1 = \begin{bmatrix} \times & \times & \cdots & \times \\ \times & \times & \cdots & \times \\ 0 & \times & \cdots & \times \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \times & \cdots & \times \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & \widehat{\mathbf{P}}_1 \end{bmatrix} = \begin{bmatrix} \times & \star & \cdots & \star \\ \times & \star & \cdots & \star \\ 0 & \star & \cdots & \star \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \star & \cdots & \star \end{bmatrix}$$

dove con $\star$ sono indicati gli elementi che sono stati modificati. Sia quindi $\mathbf{P}_2 = \mathrm{diag}(I_2, \widehat{\mathbf{P}}_2)$ la matrice di trasformazione che azzera gli elementi della seconda colonna, dal quarto elemento in poi. La matrice risultante $\mathbf{H}^{(3)} = \mathbf{P}_2(\mathbf{P}_1 \mathbf{A} \mathbf{P}_1)$ avrà le prime due colonne in forma di Hessenberg superiore, cioè

$$\mathbf{H}^{(3)} = \begin{bmatrix} \times & \times & \cdots & \times & \times \\ \times & \times & \cdots & \times & \times \\ 0 & \times & \cdots & \times & \times \\ 0 & 0 & \cdots & \times & \times \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \times & \times \end{bmatrix}.$$

La moltiplicazione da destra per $\mathbf{P}_2$ non coinvolgerà le prime due colonne, per cui la forma di Hessenberg sarà preservata. Quindi la procedura continua nel modo seguente:

$$\widetilde{\mathbf{A}} = \mathbf{P}^{(n-2)}(\cdots(\mathbf{P}^{(2)}(\mathbf{P}^{(1)} \mathbf{A} \mathbf{P}^{(1)})\mathbf{P}^{(2)})\cdots)\mathbf{P}^{(n-2)},$$

che costituisce la forma finale desiderata. Il costo computazionale di questa operazione è dell'ordine di quello per la fattorizzazione QR, dato che il costo dell'applicazione delle matrici di Householder da destra è lo stesso: $\mathcal{O}(n^3)$ flops.

Nonostante la trattazione delle matrici di Householder sia stata fatta in $\mathbb{R}$, non ci sono modifiche sostanziali nel caso di $\mathbb{C}$. Le matrici risultanti saranno ovviamente unitarie (ortogonali in $\mathbb{C}$).

4.3.2 Rotazioni di Givens

Da un punto di vista geometrico, le "rotazioni" di Givens sono matrici ortogonali 2×2 che effettuano una rotazione di un vettore nel piano, in modo da portare il vettore sull'asse delle ascisse. Quindi, dato $x = (x_1, x_2)^T \in \mathbb{R}^2$, si tratta di determinare una matrice (di rotazione) $\mathbf{G} \in \mathbb{R}^{2\times 2}$ tale che $y = \mathbf{G}x = (y_1, 0)^T$. La rotazione è data da

$$\mathbf{G} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix},$$

dove[2] $\sin\theta = \frac{x_2}{\sqrt{x_1^2+x_2^2}}$ e $\cos\theta = \frac{x_1}{\sqrt{x_1^2+x_2^2}}$. Quindi $y = \mathbf{G}x = (\|x\|_2, 0)^T$. La matrice $\mathbf{G}$ così costruita è ortogonale, infatti

$$\mathbf{G}\mathbf{G}^T = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} = \begin{bmatrix} \cos^2\theta + \sin^2\theta & -\cos\theta\,\sin\theta + \sin\theta\,\cos\theta \\ -\cos\theta\,\sin\theta + \sin\theta\,\cos\theta & \sin^2\theta + \cos^2\theta \end{bmatrix} = \mathbf{I}.$$

Il costo computazionale dell'applicazione di $\mathbf{G}$ ad un vettore è di 6 flops.

Come per le matrici di Householder, le matrici di Givens possono essere usate per azzerare, questa volta *specifici*, elementi di una matrice data, uno alla volta. La procedura può quindi essere usata per trasformare una matrice $\mathbf{A}$ in una matrice triangolare superiore, anullando tutti gli elementi sotto la diagonale principale, dalla prima all'ultima colonna. La procedura è simile alla eliminazione di Gauss: per la prima colonna, per esempio, si procede così:

$$\begin{bmatrix} I_{n-2} & 0 \\ 0 & G_1^{(n-1)} \end{bmatrix} \begin{bmatrix} \times & \times & \dots & \times \\ \times & \times & \dots & \times \\ \vdots & \vdots & \vdots & \vdots \\ \times & \times & \dots & \times \\ \times & \times & \dots & \times \end{bmatrix} = \begin{bmatrix} \times & \times & \dots & \times \\ \times & \times & \dots & \times \\ \vdots & \vdots & \vdots & \vdots \\ \star & \star & \star & \star \\ 0 & \star & \star & \star \end{bmatrix}$$

$$\begin{bmatrix} I_{n-3} & 0 & 0 \\ 0 & G_1^{(n-2)} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \times & \times & \dots & \times \\ \vdots & \vdots & \vdots & \vdots \\ \times & \times & \dots & \times \\ \times & \times & \dots & \times \\ 0 & \times & \dots & \times \end{bmatrix} = \begin{bmatrix} \times & \times & \dots & \times \\ \vdots & \vdots & \vdots & \vdots \\ \star & \star & \star & \star \\ 0 & \star & \star & \star \\ 0 & \times & \dots & \times \end{bmatrix} \dots$$

Ogni applicazione costa $6m$ flops, perchè la rotazione viene applicata a tutti gli elementi delle corrispondenti due righe della matrice. Per azzerare tutti gli elementi sotto la diagonale della prima colonna sono necessarie $n-1$ applicazioni e quindi $6m(n-1)$ flops. L'azzeramento degli elementi della j-esima colonna (agendo solo sul blocco $(n-j)\times(m-j)$ rimanente), richiede $6(n-j)(m-j)$ flops. Il costo complessivo per una fattorizzazione QR mediante rotazioni di Givens è quindi di $\sum_{j=1}^{m} 6(n-j)(m-j) = \mathcal{O}(nm^2 + m^3)$.

Agendo su specifici elementi di una matrice, le rotazioni di Givens sono particolarmente adatte a ridurre a forma triangolare superiore matrici aventi già una struttura con pochi elementi non zero sotto la diagonale principale, come ad esempio matrici di tipo Hessenberg superiore. In questo ultimo caso, ciò implica che nella matrice originaria solo un elemento per colonna deve essere annullato. Il costo computazionale si riduce sensibilmente, avendosi quindi $\sum_{j=1}^{m} 6(m-j) = \mathcal{O}(m^2)$.

[2] In $\mathbb{C}$ si avrà $\cos\theta = |x_1|/\sqrt{|x_1|^2+|x_2|^2}$ e $\sin\theta = (x_1/|x_1|)\bar{x}_2/\sqrt{|x_1|^2+|x_2|^2}$.

Come per le trasformazioni di Householder, notiamo infine che se applichiamo le rotazioni di Givens per ottenere una matrice di Hessenberg superiore, la successiva applicazione a destra delle stesse rotazioni (trasposte) in sequenza non distrugge la forma di Hessenberg. Questa proprietà rende le trasformazioni di Givens molto interessanti all'interno dell'iterazione QR per il calcolo degli autovalori (si veda il paragrafo 6.5), quando la matrice di iterazione ha forma di Hessenberg.

Capitolo 5

Problema dei minimi quadrati

Il problema dei minimi quadrati trova applicazione in molti ambiti del calcolo scientifico. Un esempio molto comune è quello della *regressione lineare* nella statistica applicata, ma ci sono molte altre applicazioni dove il problema dei minimi quadrati è usato per approssimare la soluzione di un sistema $Ax = b$ sovradeterminato, cioè con un numero di equazioni superiore al numero di incognite. Dati $\mathbf{A} \in \mathbb{R}^{n\times m}$, con $n \geq m$ e $b \in \mathbb{R}^n$, il problema dei minimi quadrati consistente nel determinare il vettore $x \in \mathbb{R}^m$ che minimizzi il residuo $r := b - \mathbf{A}x$, tipicamente in norma Euclidea, cioè

$$\min_{x\in\mathbb{R}^m} \|b - \mathbf{A}x\|_2, \tag{5.1}$$

Nel seguito vengono proposti tre diversi modi per la risoluzione di questo problema, con proprietà di accuratezza e stabilità piuttosto diverse, almeno il primo rispetto agli altri due. In tutti i casi, verrà supposto che $\mathbf{A}$ abbia rango massimo, uguale ad m.

5.1 L'equazione normale

Il primo metodo risolutivo proposto trasforma il problema di minimo in un sistema lineare, detto sistema normale o equazione normale, avente una sola soluzione. A tal fine diamo il seguente risultato generale, che stabilisce condizioni di esistenza ed unicità della soluzione di (5.1), e mostra il suo legame con l'equazione normale. Premettiamo un lemma che verrà usato nella dimostrazione del teorema.

Lemma 5.1.1 *Una matrice $\mathbf{A} \in \mathbb{R}^{n\times m}$ con $n \geq m$ ha rango massimo se e solo se la matrice $\mathbf{A}^T\mathbf{A}$ è non singolare.*

Dimostrazione. Sia $\mathbf{A} = Q_1R_1$ la fattorizzazione QR di $\mathbf{A}$, con R_1 quadrata $m \times m$ e Q_1 avente colonne ortonormali. Dalla procedura di Gram-Schmidt, per es., segue che le colonne di $\mathbf{A}$ sono linearmente indipendenti se e solo se R_1 è non singolare. Si ha inoltre che $\mathbf{A}^T\mathbf{A} = R_1^TQ_1^TQ_1R_1 = R_1^TR_1 \geq 0$. Segue quindi che $\mathbf{A}^T\mathbf{A}$ è non singolare, ed in particolare definita positiva, se e solo se $R_1^TR_1$ è non singolare, e cioè definita positiva, cioè se R_1 è non singolare. □

È quindi possibile procedere con il risultato generale.

Teorema 5.1.2 *Sia X l'insieme dei vettori $x \in \mathbb{R}^m$ che risolvono (5.1). Allora valgono le seguenti affermazioni:*

1. $x \in X$ se e solo se x è soluzione dell' equazione normale

$$\mathbf{A}^T\mathbf{A}x = \mathbf{A}^T b; \tag{5.2}$$

2. X ha un solo elemento se e solo se $\mathbf{A}$ ha rango massimo.

Dimostrazione. Siano $R = \text{range}(\mathbf{A}) = \{y \in \mathbb{R}^n \text{ taleche } y = \mathbf{A}x,\ x \in \mathbb{R}^m\}$, e il suo ortogonale $R^\perp = \{z \in \mathbb{R}^n \text{ taleche } z^T y = 0,\ y \in R\}$. Si può quindi scrivere $b = b_1 + b_2$, con $b_1 \in R$ e $b_2 \in R^\perp$.

1) Si ha $r = b - \mathbf{A}x = b_1 - \mathbf{A}x + b_2$ con $b_1 - \mathbf{A}x \in R$ e $b_2 \in R^\perp$. Quindi, dato che $b_2^T(b_1 - \mathbf{A}x) = 0$, si ha

$$\|r\|_2^2 = \|b_1 - \mathbf{A}x\|_2^2 + \|b_2\|_2^2.$$

Per ottenere il minimo di $\|r\|$, è possibile operare solo sul primo termine, visto che il secondo termine non contiene x. Quindi il minimo di $\|r\|_2$ sarà raggiunto se e solo se esiste x tale che $b_1 = \mathbf{A}x$. Seguirà inoltre che $r = b_2$, cosicchè $r \in R^\perp$. In particolare, il residuo r è così ortogonale alle colonne della matrice $\mathbf{A}$:

$$0 = \mathbf{A}^T r = \mathbf{A}^T(b - \mathbf{A}x) \quad \Leftrightarrow \quad \mathbf{A}^T\mathbf{A}x = \mathbf{A}^T b.$$

2) Per il Lemma 5.1.1, la matrice $\mathbf{A}$ ha rango massimo se e solo se la matrice $\mathbf{A}^T\mathbf{A}$ è non singolare, cioè se e solo se l'equazione normale (5.2) ha una e una sola soluzione x. □

Nel seguito supporremo $\mathbf{A}$ di rango massimo, cioè m. Il Teorema 5.1.2 suggerisce un primo metodo per la risoluzione del problema ai minimi quadrati. Infatti, per $\mathbf{A}$ di rango massimo, la matrice $\mathbf{A}^T\mathbf{A}$ è s.d.p. e se ne può quindi calcolare la fattorizzazione di Cholesky, $\mathbf{A}^T\mathbf{A} = \mathbf{L}\mathbf{L}^T$, e risolvere in ordine i sistemi triangolari

$$\mathbf{L}y = \mathbf{A}^T b,$$
$$\mathbf{L}^T x = y.$$

Queto metodo risulta essere abbastanza costoso, infatti la moltiplicazione $\mathbf{A}^T\mathbf{A}$ con $\mathbf{A} \in \mathbb{R}^{n\times m}$ costa $\frac{1}{2}(2n-1)m^2$ flops (la divisione per due è dovuta al fatto che $\mathbf{A}^T\mathbf{A}$ è simmetrica e quindi non è necessario calcolare esplicitamente sia gli elementi sopra che sotto la diagonale); il prodotto $\mathbf{A}^T b$ costa $(2n-1)m$ flops, e la fattorizzazione di Cholesky di $\mathbf{A}^T\mathbf{A}$ costa $\frac{m^3}{3} + \mathcal{O}(m^2)$ flops, per un totale di $(m+2)mn + \frac{m^3}{3} + \mathcal{O}(m^2)\ flops$.

Da un punto di vista numerico, non c'è purtroppo una separazione precisa tra matrici di rango massimo e radici non di rango massimo. Più precisamente, colonne "quasi" linearmente dipendenti possono causare seri problemi in aritmetica con precisione finita, anche se matematicamente le colonne non sono combinazioni lineari. In tal caso, il metodo dell'equazione normale non può essere usato per risolvere il problema ai minimi quadrati. Rendiamo più precisa questa affermazione mediante un semplice esempio.

Esempio 5.1.3 Siano

$$\mathbf{A} = \begin{pmatrix} 3 & 3 \\ 4 & 4 \\ 0 & \alpha \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

dove $\alpha \in \mathbb{R}$ è tale che con $\mathtt{u} \le \alpha < 1$ e $\alpha^2 < \mathtt{u}$, dove $\mathtt{u}$ indica la precisione di macchina. Calcolando esplicitamente i prodotti $\mathbf{A}^T\mathbf{A}$ e $\mathbf{A}^T b$ si ottiene

$$\mathbf{A}^T\mathbf{A} = \begin{pmatrix} 25 & 25 \\ 25 & 25+\alpha^2 \end{pmatrix}, \quad \mathbf{A}^T b = \begin{pmatrix} 7 \\ 7+\alpha \end{pmatrix},$$

da cui la soluzione del sistema all'equazioni normali $\mathbf{A}^T\mathbf{A}x = \mathbf{A}^T b$, risulta essere $x = (\frac{7}{25} - \frac{1}{\alpha}, \frac{1}{\alpha})^T$. Essendo $\alpha^2 < \mathtt{u}$, la rappresentazione di $\mathbf{A}^T\mathbf{A}$ sul calcolatore è data da

$$fl(\mathbf{A}^T\mathbf{A}) = \begin{pmatrix} 25 & 25 \\ 25 & 25 \end{pmatrix},$$

con le due colonne uguali, cioè $fl(\mathbf{A}^T\mathbf{A})$ è singolare ! In questa circostanza l'equazione normale in aritmetica con precisione finita non ha soluzioni, e quindi il metodo dell'equazione normale non risolve il problema ai minimi quadrati.

5.2 Fattorizzazione QR

Un metodo alternativo per la risoluzione del problema ai minimi quadrati (5.1) prevede l'utilizzo della fattorizzazione QR della matrice $\mathbf{A} \in \mathbb{R}^{n\times m}$. Scriviamo innanzi tutto

$$\mathbf{A} = \mathbf{QR} = [\mathbf{Q}_1, \mathbf{Q}_2] \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix} = \mathbf{Q}_1\mathbf{R}_1,$$

con $\mathbf{Q} \in \mathbb{R}^{n\times n}$ ortogonale e $\mathbf{R}_1 \in \mathbb{R}^{m\times m}$ triangolare superiore e non singolare (si veda la dimostrazione del Lemma 5.1.1). Notiamo in particolare che Range($\mathbf{Q}_1$) = Range($\mathbf{A}$) e Range($\mathbf{Q}_2$) = Range($\mathbf{A}$)$^\perp$, dato che $\mathbf{Q}_2^T\mathbf{Q}_1 = 0$. Si ha dunque

$$\begin{aligned} \|b - \mathbf{A}x\|_2^2 &= \|b - \mathbf{QR}x\|_2^2 = \|\mathbf{Q}(\mathbf{Q}^T b - \mathbf{R}x)\|_2^2 \\ &= \left\| \mathbf{Q}^T b - \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix} x \right\|_2^2 = \left\| \begin{bmatrix} \mathbf{Q}_1^T b - \mathbf{R}_1 x \\ \mathbf{Q}_2^T b \end{bmatrix} \right\|_2^2 = \|\mathbf{Q}_1^T b - \mathbf{R}_1 x\|_2^2 + \|\mathbf{Q}_2^T b\|_2^2. \end{aligned}$$

Per minimizzare la norma del residuo, è possibile agire solo sul primo termine della somma che appare nell'ultima espressione sopra, cioè dev'essere $\mathbf{Q}_1^T b - \mathbf{R}_1 x = 0$. Siccome la matrice $\mathbf{R}_1$ è non singolare, il problema ai minimi quadrati (5.1) può essere risolto per $x = \mathbf{R}_1^{-1}\mathbf{Q}_1^T b$. In tal caso si ha quindi

$$\min_{x\in\mathbb{R}^m} \|b - \mathbf{A}x\|_2^2 \equiv \|\mathbf{Q}_2^T b\|_2^2,$$

dove $\|\mathbf{Q}_2^T b\| = \|\mathbf{Q}_2\mathbf{Q}_2^T b\|$, e $\mathbf{Q}_2\mathbf{Q}_2^T b$ consiste proprio nella proiezione di b nello spazio ortogonale alle colonne di $\mathbf{A}$ (si confrontino queste considerazioni con quelle nella dimostrazione del Teorema 5.1.2). L'uso della fattorizzazione QR permette di risolvere accuratamente anche quei problemi ai minimi quadrati in cui la matrice dei coefficienti ha colonne "quasi" linearmente dipendenti come quello visto nell'Esempio 5.1.3. Infatti, la fattorizzazione di $\mathbf{A}$ è

$$\mathbf{A} = \mathbf{Q}_1\mathbf{R}_1 = \begin{bmatrix} \frac{3}{5} & 0 \\ \frac{4}{5} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 5 & 5 \\ 0 & \alpha \end{bmatrix};$$

la matrice $\mathbf{R}_1$ ha condizionamento $\mathrm{cond}_F(\mathbf{R}_1) = \|\mathbf{R}_1\|_F\|\mathbf{R}_1^{-1}\|_F \approx \mathcal{O}(1/\alpha)$, che è significativamente sotto il valore dell'inversa della precisione di macchina. Il sistema associato può quindi essere risolto con un buon numero di cifre significative.

È importante notare a fini implementativi che il vettore $\hat{b} := \mathbf{Q}^T b$ può essere costruito durante la fattorizzazione QR, applicando al vettore b le matrici ortogonali di trasformazione (di Householder o di Givens) via via generate, come fatto per le colonne di $\mathbf{A}$. In tal modo, non è necessario memorizzare esplicitamente $\mathbf{Q}$ ed il costo computazionale può essere di molto inferiore.

Riassumendo, la procedura per la determinazione di x mediante fattorizzazione QR è data dai seguenti passi:

1. Applica le trasformazioni ad $\mathbf{A}$ e b contemporaneamente:

$$[\mathbf{R}_1, \hat{b}] := \mathbf{P}_m \mathbf{P}_{m-1} \cdots \mathbf{P}_1 [\mathbf{A}, b]$$

2. Determina $x = (\mathbf{R}_1)^{-1}(\hat{b})_{1:m}$ risolvendo il sistema triangolare superiore.

Il seguente teorema mostra come la fattorizzazione QR permetta di risolvere, in pratica, un problema "vicino" al problema originario (5.1), e quindi è stabile secondo l'analisi all'indietro.

Teorema 5.2.1 *Il vettore $\widehat{x} \in \mathbb{R}^m$, approssimazione della soluzione del problema (5.1) ottenuta mediante fattorizzazione QR, è tale che*

$$\widehat{x} = \underset{x \in \mathbb{R}^m}{\operatorname{argmin}} \|(\mathbf{A} + \delta\mathbf{A})x - (b + \delta b)\|_2,$$

dove

$$\|\delta\mathbf{A}\|_F \leq (6n - 3m + 40)m\mathtt{u}\|\mathbf{A}\|_F + \mathcal{O}(\mathtt{u}^2), \quad e \quad \|\delta b\|_2 \leq (6n - 3m + 40)m\mathtt{u}\|b\|_2 + \mathcal{O}(\mathtt{u}^2).$$

5.3 Decomposizione in valori singolari

Nota: *Argomento complementare, non fa parte del programma d'esame.*

Per matrici rettangolari non è possibile ottenere una diagonalizzazione come, per es., per matrici quadrate e diagonalizzabili, od a maggior ragione per matrici simmetriche, del tipo $\mathbf{X}\mathbf{\Lambda}\mathbf{X}^T$ con $\mathbf{X}$ ortogonale (nel caso reale simmetrico). D'altra parte, è in effetti possibile estendere il concetto di *diagonalizzazione*, permettendo di prendere due matrici ortogonali, seppure *diverse*, come matrici di trasformazione. Questo è ciò che determina la decomposizione in valori singolari (in inglese, *Singular Values Decomposition*, SVD). Questa decomposizione matriciale è dunque molto generale e ricopre una posizione importante all'interno del calcolo numerico ed in svariati campi applicativi. In questa sezione ne sarà data la definizione e saranno fornite le prime proprietà fondamentali; inoltre, verrà mostrata la sua applicazione al problema ai minimi quadrati. Vengono inoltre dati alcuni cenni su come affrontare il suo calcolo da un punto di vista numerico.

Sia $\mathbf{A} \in \mathbb{R}^{n \times m}$ (o $\mathbb{C}^{n \times m}$) una generica matrice e si supponga per semplicità di esposizione, ma senza perdere generalità, che la matrice sia "alta" cioè $n \geq m$ (nel caso $n \leq m$, si può scrivere la decomposizione per $\mathbf{A}^T$ e poi trasporla). Il seguente teorema assicura l'esistenza della decomposizione.

Teorema 5.3.1 *Sia $\mathbf{A} \in \mathbb{R}^{n \times m}$, con $n \geq m$, allora esistono $\mathbf{U} = [u_1, \ldots, u_n] \in \mathbb{R}^{n \times n}$, $\mathbf{V} = [v_1, \ldots, v_m] \in \mathbf{R}^{m \times m}$ ortogonali e $\mathbf{\Sigma} \in \mathbb{R}^{n \times m}$, tali che*

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \quad \textit{con } \mathbf{\Sigma} = \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_m \\ \mathbf{0} & \ldots & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \mathbf{\Sigma}_1 \\ \mathbf{0} \end{pmatrix},$$

dove $\mathbf{\Sigma}_1$ è diagonale e $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m \geq 0$.

Le colonne $\{u_i\}_{i=1,\ldots,n}$ della matrice $\mathbf{U}$ sono dette *vettori singolari sinistri*, le colonne $\{v_j\}_{j=1,\ldots,m}$ di $\mathbf{V}$ sono dette *vettori singolari destri*, e gli scalari $\{\sigma_k\}_{k=1,\ldots,m}$ sono detti *valori singolari*. In particolare, la decomposizione mostra che

$$\mathbf{A}v_i = u_i\sigma_i, \qquad \mathbf{A}^T u_i = v_i\sigma_i, \quad i = 1, \ldots, m.$$

Ricordiamo che, anche per matrici rettangolari, la norma-2 è definita come

$$\|\mathbf{A}\|_2^2 := \max_{\|x\|_2=1} \|\mathbf{A}x\|_2^2 = \max_{\|x\|_2=1} x^T\mathbf{A}^T\mathbf{A}x = \lambda_{max}(\mathbf{A}^T\mathbf{A}),$$

dove nell'ultima uguaglianza sono state utilizzate le proprietà del quoziente di Rayleigh. Allora, per qualsiasi $x \in \mathbb{R}^m$ con $\|x\|_2 = 1$, si può scrivere

$$\begin{aligned}\|\mathbf{A}x\|_2 &= \|\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T x\|_2 = \|\mathbf{\Sigma} \underbrace{\mathbf{V}^T x}_{\substack{=:y,\\ \|y\|_2=1}} \|_2 = \|\mathbf{\Sigma}y\|_2 \\ &= \left\| \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_m \\ \mathbf{0} & \dots & \mathbf{0} \end{pmatrix} y \right\|_2 = \left\| \begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_m \end{pmatrix} y \right\|_2 \leq \sigma_1,\end{aligned}$$

dove σ_1 indica il più grande valore singolare di $\mathbf{A}$. Inoltre, tale valore è raggiunto per $x = v_1$, infatti si ha $\|\mathbf{A}v_1\|_2 = \|u_1\sigma_1\|_2 = \sigma_1$. Quindi effettivamente si ha

$$\|A\|_2 = \max_{\|x\|_2=1} \|Ax\|_2 = \sigma_1.$$

In modo del tutto analogo si mostra che per ogni $x \in \mathbb{R}^m$ con $\|x\|_2 = 1$ si ha $\|\mathbf{A}x\|_2 \geq \sigma_m$, e $\|\mathbf{A}v_m\|_2 = \sigma_m$, quindi $\min_{\|x\|_2=1} \|\mathbf{A}x\|_2 = \sigma_m$. Se $\sigma_m \neq 0$, si definisce il numero di condizionamento di una generica matrice rettangolare $\mathbf{B} \in \mathbb{R}^{n\times m}$ come

$$\kappa_2(\mathbf{B}) := \frac{\sigma_1(\mathbf{B})}{\sigma_{\min\{m,n\}}(\mathbf{B})}, \quad \text{se } \sigma_{\min\{m,n\}}(\mathbf{B}) \neq 0.$$

Si noti che questo concetto di condizionamento generalizza quanto visto nel caso di matrici quadrate non singolari. Infatti, se $\mathbf{B} \in \mathbb{R}^{n\times n}$, allora $\mathbf{B} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ con $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_n)$ e $\|\mathbf{B}\|_2 = \sigma_1(\mathbf{B})$; inoltre $\mathbf{B}^{-1} = \mathbf{V}\mathbf{\Sigma}^{-\mathbf{1}}\mathbf{U}^T$ con $\Sigma^{-1} = \text{diag}(1/\sigma_1, \dots, 1/\sigma_n)$ (in questo caso in ordine *crescente*), cosicchè $\|\mathbf{B}^{-1}\|_2 = \frac{1}{\sigma_n(\mathbf{B})}$.

La decomposizione in valori singolari e quella in autovalori sono collegate, nonostante la prima sia molto più generale. Infatti, per $\mathbf{A} \in \mathbb{R}^{n\times m}$ e $n \geq m$ si ha

$$\mathbf{A}^T\mathbf{A} = (\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T)^T\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \mathbf{V}\mathbf{\Sigma}^T\mathbf{U}^T\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \mathbf{V}\mathbf{\Sigma}_1^2\mathbf{V}^T,$$

ottenendo così la decomposizione spettrale di $\mathbf{A}^T\mathbf{A}$. In generale vale quindi che $\sigma_i(\mathbf{A}) = \sqrt{\lambda_i(\mathbf{A}^T\mathbf{A})}$, $i = 1, \dots, m$, e, nel caso di $\mathbf{A}$ normale, $\sigma_i(\mathbf{A}) = |\lambda_i(\mathbf{A})|$.

Esistono diverse relazioni che legano decomposizione in valori singolari e norme matriciali, ad esempio,

$$\|\mathbf{A}\|_F^2 = \text{tr}\,(\mathbf{A}^T\mathbf{A}) = \text{tr}\,(\mathbf{V}\mathbf{\Sigma}^T\mathbf{U}^T\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T) = \text{tr}\,(\mathbf{\Sigma}^T\mathbf{\Sigma}\mathbf{V}^T\mathbf{V}) = \text{tr}\,(\begin{bmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_m^2 \end{bmatrix}) = \sum_{k=1}^m \sigma_k^2.$$

Si noti come la decomposizione in valori singolari possa essere scritta come

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_{k=1}^m u_i\sigma_i v_i^T,$$

e se quindi esiste $p \in \{1, \ldots, m-1\}$ tale che $\sigma_{p+1} = \cdots = \sigma_m = 0$, allora la decomposizione si riduce al termine p-esimo cioè

$$\mathbf{A} = \sum_{k=1}^{p} u_i \sigma_i v_i^T = [u_1, \ldots, u_p] \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_p \end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_p^T \end{bmatrix},$$

e il rango della matrice $\mathbf{A}$ risulta essere proprio p.

Alcune fondamentali proprietà della decomposizione in valori singolari sono riportate nella proposizione seguente.

Proposizione 5.3.2 *Sia $\mathbf{A} \in \mathbf{R}^{n \times m}$ e sia $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ la sua decomposizione in valori singolari. Sia $p \in \{1, \ldots, m\}$ tale che $\sigma_p = \min\{\sigma_i, \text{con} \sigma_i > 0\}$. Definendo $\mathbf{U}_1 := [u_1, \ldots, u_p]$, $\mathbf{U}_2 = [u_{p+1}, \ldots, u_n]$, $\mathbf{V}_1 = [v_1, \ldots, v_p]$ e $\mathbf{V}_2 = [v_{p+1}, \ldots, v_m]$, valgono le seguenti affermazioni:*

1. *$\mathbf{A}$ ha rango massimo se e solo se $p = m$.*
2. *$\text{Range}(\mathbf{A}) = \text{Range}(\mathbf{U}_1)$;*
3. *$\text{Range}(\mathbf{A}^T) = \text{Range}(\mathbf{V}_1)$;*
4. *$\text{Ker}(\mathbf{A}) = \text{Range}(\mathbf{V}_2)$;*
5. *$\text{Ker}(\mathbf{A}^T) = \text{Range}(\mathbf{U}_2)$.*

Per dimostrare tali proprietà basta osservare che, data $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, si ha

$$\mathbf{A}\mathbf{V} = \mathbf{U}\mathbf{\Sigma} = [\mathbf{U}_1, \mathbf{U}_2] \begin{bmatrix} \mathbf{\Sigma}_1 \\ \mathbf{0} \end{bmatrix} = \mathbf{U}_1 \mathbf{\Sigma}_1, \tag{5.3}$$

cioè lo spazio generato dalle colonne di $\mathbf{A}$ coincide con lo spazio generato dalle colonne di $\mathbf{U}_1$. Analogamente, $\mathbf{A}^T = \mathbf{V}\mathbf{\Sigma}^T\mathbf{U}^T$, da cui

$$\mathbf{A}^T\mathbf{U} = \mathbf{V}\mathbf{\Sigma}^T = [\mathbf{V}_1, \mathbf{V}_2][\mathbf{\Sigma}_1, \mathbf{0}] = \mathbf{V}_1\mathbf{\Sigma}_1.$$

La SVD ed il problema dei minimi quadrati

La decomposizione in valori singolari può essere sfruttata per la risoluzione numerica del problema dei minimi quadrati (5.1). Supponendo $\mathbf{A}$ di rango massimo ($p = m$) ed usando le relazioni in (5.3), si osserva che

$$\begin{aligned} \|b - \mathbf{A}x\|_2^2 &= \|b - \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T x\|_2^2 = \|\mathbf{U}(\mathbf{U}^T b - \mathbf{\Sigma}\mathbf{V}^T x)\|_2^2 = \|\mathbf{U}^T b - \mathbf{\Sigma}\underbrace{\mathbf{V}^T x}_{=:y}\|_2^2 \\ &= \left\| \begin{pmatrix} \mathbf{U}_1^T b \\ \mathbf{U}_2^T b \end{pmatrix} - \begin{pmatrix} \mathbf{\Sigma}_1 \\ \mathbf{0} \end{pmatrix} y \right\|_2^2 = \|\mathbf{U}_1^T b - \mathbf{\Sigma}_1 y\|_2^2 + \|\mathbf{U}_2^T b\|_2^2. \end{aligned}$$

Ancora una volta, per minimizzare la norma del residuo si può solo agire sul primo dei due termini. Per $y = \mathbf{\Sigma}_1^{-1}\mathbf{U}_1^T b$ tale termine viene annullato, da cui $x = \mathbf{V}y = \mathbf{V}\mathbf{\Sigma}_1^{-1}\mathbf{U}_1^T b$, che corrisponde alla soluzione cercata. Per il residuo si ha $\min \|b - \mathbf{A}x\|_2 = \|\mathbf{U}_2^T b\|_2$. Si noti la similarità con la fattorizzazione QR, dove però ora è stata usata una decomposizione diversa. La soluzione x può essere espressa come combinazione lineare dei vettori singolari destri come $x = \sum_{k=1}^{m} \left(\frac{u_k^T b}{\sigma_1}\right) v_i$,

Il seguente teorema mostra le potenzialità della decomposzione in valori singolari e spiega, almeno in parte, il perchè tale decomposizione venga applicata in gran parte delle aree del calcolo numerico.

Teorema 5.3.3 *Sia $\mathbf{A} \in \mathbb{R}^{n\times m}$ una matrice di rango massimo e sia $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ la sua decomposizione in valori singolari. Sia $r < m$ e si definisca $\mathbf{A}_r := \sum_{k=1}^{r} u_k\sigma_k v_k^T$. Allora*

$$\min_{\substack{\mathbf{B}\in\mathbb{R}^{n\times m}\\ \text{rango}\,(\mathbf{B})=r}} \|\mathbf{A}-\mathbf{B}\|_2 = \|\mathbf{A}-\mathbf{A}_r\|_2 = \sigma_{r+1}.$$

Il teorema mostra che tra tutte le matrice di rango r, quella più vicina alla matrice $\mathbf{A}$, in norma indotta Euclidea, è quella data dalla sua decomposizione in valori singolari troncata al termine r-esimo. Questo risultato ha innumerevoli utilizzi in ambito applicativo, dalla compressione di immagini, all'analisi statistica di dati, al pattern recognition e cosi via.

Il Teorema 5.3.3 permette anche di introdurre il concetto di *rango numerico* di una matrice che consiste nel numero s di valori singolari maggiori di una tolleranza prefissata `tol` (ES: 10^{-12}). La matrice $\mathbf{A}_s := \sum_{k=1}^{s} u_k\sigma_k v_k^T$ è quindi tale che $\|\mathbf{A}-\mathbf{A}_s\|_2 = \sigma_{s+1}$, dove $\sigma_{s+1} \approx 0$ può essere considerato numericamente zero. Una volta calcolato il rango numerico, si può preferire lavorare con $\mathbf{A}_s$ piuttosto che con $\mathbf{A}$.

Riportiamo infine un risultato che mostra l'errore ottenuto nella soluzione del problema dei minimi quadrati quando viene usato l'approccio della SVD.

Teorema 5.3.4 *Sia $\mathbf{A} \in \mathbb{R}^{n\times m}$, $n \geq m$, una matrice di rango massimo e sia $\delta\mathbf{A} \in \mathbb{R}^{n\times m}$ una sua perturbazione tale che $\|\delta\mathbf{A}\|_2 < \sigma_m(\mathbf{A})$. Siano poi $b \in \mathbb{R}^n$, $b \neq 0$ e $\delta b \in \mathbb{R}^n$. Allora si ha che*

1. *$\mathbf{A} + \delta\mathbf{A}$ ha rango massimo;*
2. *Detta $x + \delta x$ la soluzione del problema dei minimi quadrati perturbato*

$$\min_{y\in\mathbb{R}^m} \|(b+\delta b) - (\mathbf{A}+\delta\mathbf{A})y\|_2,$$

risulta

$$\frac{\|\delta x\|_2}{\|x\|_2} \leq \frac{\kappa_2(\mathbf{A})}{1-\varepsilon_{\mathbf{A}}\kappa_2(\mathbf{A})}\left(\left(1+\kappa_2(\mathbf{A})\frac{\|r\|_2}{\|\mathbf{A}\|_2\|x\|_2}\right)\varepsilon_{\mathbf{A}} + \frac{\|b\|_2}{\|\mathbf{A}\|_2\|x\|_2}\varepsilon_b\right),$$

dove $r = b - \mathbf{A}x$ è il vettore residuo,

$$\varepsilon_{\mathbf{A}} = \frac{\|\delta\mathbf{A}\|_2}{\|\mathbf{A}\|_2}, \quad e \quad \varepsilon_b = \frac{\|\delta b\|_2}{\|b\|_2}.$$

Calcolo della SVD

Per il calcolo effettivo delle matrici $\mathbf{U}, \mathbf{V}, \mathbf{\Sigma}$ è possibile seguire varie strade; ne mostreremo due, la seconda delle quali è preferibile da un punto di vista della stabilità. Entrambi i metodi sfruttano algoritmi visti per il calcolo degli autovalori di matrice. La dimostrazione del Teorema 5.3.1 è costruttiva, nel senso che mostra un altro metodo per effettivamente costruire le matrici della decomposizione, ma non viene riportata in questi appunti.

1) Come mostrato in precedenza, se $\mathbf{A} = \mathbf{U}\Sigma V^T$, allora si ha $\mathbf{A}^T\mathbf{A} = \mathbf{V}\mathbf{\Sigma}_1^2\mathbf{V}^T$. Quindi è possibile determinare i vettori singolari destri v_i ed i valori singolari risolvendo il problema agli autovalori $\mathbf{A}^T\mathbf{A}v = \lambda v$ con le tecniche note, dove vale $\lambda = \sigma_i^2$. Questa strategia può dare difficoltà per il calcolo di valori singolari piccoli, con $\sigma_i < \sqrt{\mathbf{u}}$, in quando gli autovalori $\lambda_i = \sigma_i^2$ saranno caratterizzati dall'essere più piccoli della precisione della macchina. In pratica, quindi sono zero in aritmetica finita, mentre i corrispondenti σ_i hanno valori ancora apprezzabili.

2) Un approccio che non soffre di questi problemi sfrutta il fatto che la matrice simmetrica

$$\boldsymbol{\mathcal{A}} = \begin{bmatrix} 0 & \mathbf{A}^T \\ \mathbf{A} & 0 \end{bmatrix}$$

ammette la decomposizione spettrale $\boldsymbol{\mathcal{A}} = \mathbf{Q}\Lambda Q^T$ con $\mathbf{Q}$ ortogonale, che può essere scritta esplicitamente in termini della SVD di $\mathbf{A}$:

$$\mathbf{Q} = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{V} & \mathbf{V} & 0 \\ \mathbf{U}_1 & -\mathbf{U}_1 & \mathbf{U}_2 \end{bmatrix}, \qquad \mathbf{U} = [\mathbf{U}_1, \mathbf{U}_2].$$

Il prodotto esplicito mostra che

$$\mathbf{Q}^T \boldsymbol{\mathcal{A}} \mathbf{Q} = \operatorname{diag}(\sigma_1, \sigma_2, \ldots, \sigma_m, -\sigma_1, -\sigma_2, \ldots, -\sigma_m, 0, \ldots, 0).$$

Gli autovalori non zero di $\boldsymbol{\mathcal{A}}$ coincidono con i valori singolari di $\mathbf{A}$ e compaiono in coppie $\pm\sigma_i$. Il calcolo degli vettori singolari e dei valori singolari associati può essere quindi effettuato con maggiore accuratezza, al costo di una maggiore dimensione della matrice.

Capitolo 6

Problema agli autovalori

6.1 Introduzione

Data $\mathbf{A} \in \mathbb{R}^{n\times n}$ (o $\mathbb{C}^{n\times n}$), il problema agli autovalori consiste nel calcolo approssimato della coppia (λ, x) (detta autocoppia) con $\lambda \in \mathbb{C}$ e $0 \neq x \in \mathbb{C}^n$ (risp. autovalore e autovettore) tale che

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

Possono essere di interesse sia tutte le autocoppie della matrice $\mathbf{A}$, oppure solo alcune di esse, quale per l'esempio quella corrispondente all'autovalore più grande in modulo, così come può essere necessario il solo calcolo degli autovalori, senza i corrispondenti autovettori.

Questo tipo di problema ha grande interesse applicativo, per esempio nell'ingegneria civile, dove le autocoppie sono in relazione alle vibrazioni critiche delle strutture, oppure nell'analisi di circuiti elettrici, oppure ancora nello studio di sistemi dinamici (esempi tipici nella dinamica delle popolazioni), ma anche nella ricerca di parole chiave dei motori di ricerca (Google, Yahoo, ecc.), reti complesse in generale e social networks in particolare.

Questo problema è non lineare, quindi non ci sono in generale algoritmi "diretti" per la sua risoluzione. Una delle maggiori difficoltà che si riscontrano è data dal fatto che la soluzione del problema può essere molto sensibile a perturbazioni: anche a piccole perturbazioni della matrice $\mathbf{A}$ possono corrispondere grandi perturbazioni dei suoi autovalori e autovettori. Il prossimo esempio mostra la perturbazione dell'autovalore, dovuta ad una piccola perturbazione del dato (la matrice), quando questa è un blocco di Jordan. Questo può essere considerato un caso limite, in quanto è proprio con queste matrici che solitamente si vedono le maggiori amplificazioni degli errori nei dati.

Esempio 6.1.1 Sia

$$\mathbf{J}(0) = \begin{pmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & 0 \end{pmatrix} \in \mathbb{R}^{n\times n},$$

il blocco di Jordan di dimensione n relativo all'autovalore nullo, $\lambda = 0$. Si consideri poi la matrice perturbata di ϵ nel solo elemento di posto $(n, 1)$:

$$\mathbf{J}_\epsilon = \mathbf{J}(0) + \epsilon e_n e_1^T = \begin{pmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ \epsilon & & & 0 \end{pmatrix} \in \mathbb{R}^{n\times n}.$$

che consiste in una perturbazione "ϵ" della matrice $\mathbf{J}$. Il suo polinomio caratteristico è dato da:

$$\begin{aligned}
\det\left(\mathbf{J}_\epsilon - \lambda\mathbf{I}\right) &= \det\begin{pmatrix} -\lambda & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ \epsilon & & & -\lambda \end{pmatrix} \\
&= -\lambda \det \underbrace{\begin{pmatrix} -\lambda & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & -\lambda \end{pmatrix}}_{\in\mathbb{R}^{(n-1)\times(n-1)}} + (-1)^{n+1}\epsilon \det \underbrace{\begin{pmatrix} 1 & & & \\ -\lambda & \ddots & & \\ & \ddots & \ddots & \\ & & -\lambda & 1 \end{pmatrix}}_{\in\mathbb{R}^{(n-1)\times(n-1)}} \\
&= (-\lambda)^n + (-1)^{n+1}\epsilon,
\end{aligned}$$

dove si è sviluppato il determinante per la prima colonna, e si è tenuto conto che entrambe le matrici $(n-1)\times(n-1)$ sono triangolari e quindi il loro determinante è il prodotto degli elementi sulla diagonale. Si ha quindi che gli autovalori della matrice $\mathbf{J}_\epsilon$ sono le radici dell'equazione complessa

$$(-\lambda)^n + (-1)^{n+1}\epsilon = 0.$$

Si ottengono dunque n radici distinte λ_i, $i = 1, \ldots, n$ distribuite sulla circonferenza di raggio

$$|\lambda_i| = \sqrt[n]{\epsilon}, \quad \forall\, i = 1, \ldots, n.$$

Quindi, mediante una perturbazione di ϵ, l'autovalore nullo di molteplicità n non solo si sposta lontano dallo zero, ma diventa un autovalore semplice, quindi le proprietà spettrali della matrice cambiano in modo molto significativo. Ad esempio, per $n = 6$ e $\epsilon = 10^{-8}$, si ha $|\lambda| = 10^{-8/6} \approx 0.046$: ad una piccola perturbazione della matrice ($\epsilon = 10^{-8}$) corrispondono grandi perturbazioni degli autovalori.

Ci sono due grandi classi di metodi per il problema agli autovalori:

- Metodi basati su trasformazioni della matrice, che mantengano lo spettro;
- Metodi iterativi per l'approssimazione di alcune autocoppie.

Nel seguito faremo riferimento alle sequenti decomposizioni:

- Se $\mathbf{A} \in \mathbb{R}^{n\times n}$ è simmetrica, si può scrivere $\mathbf{A} = \mathbf{X}\mathbf{\Lambda}\mathbf{X}^T$, con $X \in \mathbb{R}^{n\times n}$ ortogonale e $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ reale. Analogamente, se $\mathbf{A} \in \mathbb{C}^{n\times n}$ è Hermitiana, si può scrivere $\mathbf{A} = \mathbf{X}\mathbf{\Lambda}\mathbf{X}^H$, con $X \in \mathbb{C}^{n\times n}$ unitaria e $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$ reale.
- Se $\mathbf{A} \in \mathbb{C}^{n\times n}$ è non Hermitiana ma ammette n autovettori linearmente indipendenti allora è diagonalizzabile, cioè $\mathbf{A} = \mathbf{X}\mathbf{\Lambda}\mathbf{X}^{-1}$, con $\mathbf{X} = [x_1, \ldots, x_n]$ invertibile e x_i autovettore destro di $\mathbf{A}$ per ogni i: $\mathbf{A}x_i = \lambda_i x_i \ \forall\, i = 1, \ldots, n$. Sia $\mathbf{Y} = [y_1, \ldots, y_n]$ la matrice avente sulle colonne gli autovettori sinistri di $\mathbf{A}$, cioè $y_i^H\mathbf{A} = \lambda_i y_i^H \ \forall\, i = 1, \ldots, n$. Si può dunque scrivere

$$\mathbf{Y}^H\mathbf{A} = \mathbf{\Lambda}\mathbf{Y}^H.$$

Dividendo per $\mathbf{Y}^H$ si ottiene $\mathbf{A}\mathbf{Y}^{-H} = \mathbf{Y}^{-H}\mathbf{\Lambda}$, cioè $\mathbf{Y}^{-H}$ è una matrice di autovettori destri. Quindi, ponendo $\mathbf{X} = \mathbf{Y}^{-H}$ si ha

$$\mathbf{Y}^H\mathbf{X} = \mathbf{I}.$$

È quindi possibile scrivere la decomposizione in autovalori ed autovettori sfruttando entrambi gli autovettori, cioè

$$\mathbf{A} = \mathbf{X}\mathbf{\Lambda}\mathbf{Y}^H = \sum_{i=1}^{n} x_i \lambda_i y_i^H,$$

e questa prende il nome di *decomposizione spettrale*. Per $\mathbf{A}$ Hermitiana, $\mathbf{Y} = \mathbf{X}$ e si ritrova la nota relazione.

- Se $\mathbf{A}$ non è diagonalizzabile, è sempre possibile scrivere la decomposizione di Jordan,

$$\mathbf{A} = \mathbf{X}\mathbf{J}\mathbf{X}^{-1}.$$

Da un punto di vista computazionale questa decomposizione è molto difficile da ottenere - come visto nell'Esempio 6.1.1 - in quanto la decomposizione è numericamente instabile: piccole perturbazioni dei dati transformano un autovalore multiplo con blocco di Jordan non banale in un gruppo di autovalori tutti semplici, per cui il blocco di Jordan scompare.

- Se $\mathbf{A}$ non è diagonalizzabile, è sempre possibile scrivere la decomposizione di Schur

$$\mathbf{A} = \mathbf{Q}\mathbf{R}\mathbf{Q}^H,$$

con $\mathbf{Q}$ unitaria e $\mathbf{R}$ triangolare superiore avente sulla diagonale gli autovalori di $\mathbf{A}$. È possibile approssimare questa decomposizione in modo stabile, e questo è ciò che fa l'iterazione QR.

Riassumendo, le decomposizioni citate mostrano che se una matrice è reale simmetrica (o Hermitiana in $\mathbb{C}$) è sempre possibile trasformarla in una matrice diagonale mediante trasformazioni ortogonali. Così non è per matrici non simmetriche, per le quali non è deppure detto che la matrice sia trasformabile in una matrice diagonale mediante trasformazioni per similitudine. Quindi le matrici non simmetriche sono molto più difficili da trattare, da un punto di vista spettrale. Queste difficoltà si ripercuotono in modo naturale ai metodi computazionali.

Infine, introduciamo una classe di matrici più ampia di quella delle matrici simmetriche, ma con buone proprietà come quelle delle matrici simmetriche.

Definizione 6.1.2 (Matrice normale) *Una matrice* $\mathbf{A} \in \mathbb{C}^{n\times n}$ *si dice normale se* $\mathbf{A}\mathbf{A}^H = \mathbf{A}^H\mathbf{A}$.

La seguente proprietà è per le matrici normali così importante che viene spesso usata come definizione. Infatti le matrici normali sono quelle matrici diagonalizzabili mediante matrici di trasformazioni unitarie; diversamente dal caso Hermitiano, la matrice diagonale ha autovalori complessi, in generale.

Teorema 6.1.3 *Una matrice* $\mathbf{A} \in \mathbb{C}^{n\times n}$ *è normale se e solo se è diagonalizzabile mediante una matrice unitaria, cioè* $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^H$, *con* $\mathbf{Q} \in \mathbb{C}^{n\times n}$ *unitaria e* $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n) \in \mathbb{C}^{n\times n}$.

Dimostrazione. $\boxed{\Leftarrow}$. Si ha $AA^H = Q\Lambda Q^H Q\bar{\Lambda}Q^H = Q|\Lambda|^2Q^H$ dove $|\Lambda|^2 = \Lambda\bar{\Lambda}$ è matrice diagonale con gli elementi $|\lambda_i|^2$ sulla diagonale. Analogamente si ha $A^HA = Q|\Lambda|^2Q^H$ e quindi A è normale.

$\boxed{\Rightarrow}$. Facciamo prima vedere che se R è triangolare (superiore) e normale, allora dev'essere diagonale. Sia $R = [R_{11}, R_{12}; 0, R_{22}]$. Si ha $RR^H = [R_{11}R_{11}^H + R_{12}R_{12}^H, *; *, *]$ e $R^HR = [R_{11}^HR_{11}, *; *, *]$. Siccome R è normale, dev'essere $RR^H = R^HR$ e per il blocco (1,1) dev'essere $R_{11}R_{11}^H + R_{12}R_{12}^H = R_{11}^HR_{11}$. In particolare, la traccia

dev'essere la stessa, da cui $\mathrm{tr}(R_{11}R_{11}^H) + \mathrm{tr}(R_{12}R_{12}^H) = \mathrm{tr}(R_{11}^H R_{11})$. Un calcolo diretto mostra che $\mathrm{tr}(R_{11}R_{11}^H) = \mathrm{tr}(R_{11}^H R_{11})$, quindi $\mathrm{tr}(R_{12}R_{12}^H) = 0$. Ma siccome $\mathrm{tr}(R_{12}R_{12}^H) = \|R_{12}\|_F^2$, si ha che $R_{12} = 0$. Iterando l'argomento ai due singoli blocchi diagonali, si trova che ogni elemento non diagonale dev'essere zero. La dimostrazione si conclude scrivendo la decomposizione di Schur $A = QRQ^H$ e notando che $AA^H = QRR^HQ^H$, $A^HA = QR^HRQ^H$, per cui $AA^H = A^HA$ se e solo se $RR^H = R^HR$, cioè R è triangolare superiore e normale, e quindi diagonale. □

6.2 Localizzazione e perturbazione di autovalori

In molti ambiti può essere richiesto di localizzare gli autovalori, nel piano complesso o sulla retta reale. In particolare, nel seguito vengono mostrate alcune tecniche per individuare regioni del piano dove ci saranno autovalori della matrice data. Indichiamo con $\mathrm{spec}(\mathbf{A}) = \{\lambda_1, \ldots, \lambda_n\} \subset \mathbb{C}$ l'insieme degli autovalori (lo spettro) della matrice $\mathbf{A}$.

Proposizione 6.2.1 (di Hirsch) *Sia $\mathbf{A} \in \mathbb{C}^{n\times n}$ e $\|\cdot\|$ norma matriciale indotta. Allora*

$$\mathrm{spec}(\mathbf{A}) \subset \{z \in \mathbb{C} \ t.c. \ |z| \le \|\mathbf{A}\|\},$$

cioè $|\lambda| \le \|\mathbf{A}\|$, $\forall\, \lambda \in \mathrm{spec}(\mathbf{A})$.

Dimostrazione. Sia (λ, x) con $\|x\| = 1$ una autocoppia di $\mathbf{A}$. Quindi $\lambda x = \mathbf{A}x$, da cui $\|\lambda x\| = \|\mathbf{A}x\|$. Usando $\|\lambda x\| = |\lambda|\|x\| = |\lambda|$ e $\|\mathbf{A}x\| \le \|\mathbf{A}\|\|x\| = \|\mathbf{A}\|$, si ottiene $|\lambda| \le \|\mathbf{A}\|$. □

Definizione 6.2.2 (Cerchi di Gerschgorin) *Sia $\mathbf{A} \in \mathbb{C}^{n\times n}$. I cerchi del piano complesso*

$$\mathcal{G}_i^{(r)} := \left\{ z \in \mathbb{C} \ t.c. \ |z - \mathbf{A}_{ii}| \le \sum_{\substack{j=1 \\ j\neq i}}^{n} |\mathbf{A}_{ij}| \right\}, \quad i = 1, \ldots, n,$$

di centro $\mathbf{A}_{ii}$ e raggio $\displaystyle\sum_{\substack{j=1 \\ j\neq i}}^{n} |\mathbf{A}_{ij}|$ sono detti cerchi di Gerschgorin per righe (da cui l'apice (r)).

Teorema 6.2.3 (Primo teorema di Gerschgorin) *Sia $\mathbf{A} \in \mathbb{C}^{n\times n}$. Allora*

$$\mathrm{spec}(\mathbf{A}) \subset \bigcup_{i=1}^{n} \mathcal{G}_i^{(r)},$$

cioè gli autovalori di $\mathbf{A}$ sono contenuti nell'unione dei cerchi di Gerschgorin per righe.

Dimostrazione. Sia (λ, x), con $x \neq 0$ un'autocoppia di $\mathbf{A}$, cioè tale che $\mathbf{A}x = \lambda x$. Quindi, scrivendo questa uguaglianza per righe, si ha

$$\sum_{j=1}^{n} \mathbf{A}_{ij}x_j = \lambda x_i, \quad \Leftrightarrow \quad (\mathbf{A}_{ii} - \lambda)x_i = -\sum_{\substack{j=1 \\ j\neq i}}^{n} \mathbf{A}_{ij}x_j. \tag{6.1}$$

Sia ora $\widehat{i} \in \{1, \ldots, n\}$ tale che $|x_{\widehat{i}}| = \max_{k=1,\ldots,n} |x_k|$. Dall'equazione (6.1), per $i = \widehat{i}$, si ha

$$(\mathbf{A}_{\widehat{i}\widehat{i}} - \lambda)x_{\widehat{i}} = -\sum_{\substack{j=1 \\ j\neq \widehat{i}}}^{n} \mathbf{A}_{\widehat{i}j}x_j,$$

da cui

$$|\mathbf{A}_{\widehat{i}\widehat{i}} - \lambda| \leq \sum_{\substack{j=1 \\ j\neq\widehat{i}}}^{n} \frac{|\mathbf{A}_{\widehat{i}j}|\,|x_j|}{|x_{\widehat{i}}|} \leq \sum_{\substack{j=1 \\ j\neq\widehat{i}}}^{n} |\mathbf{A}_{\widehat{i}j}|, \quad \Rightarrow \quad \lambda \in \mathcal{G}_{\widehat{i}}^{(r)}.$$

Siccome non è noto a priori quale sia $\widehat{i}$, si ha $\lambda \in \bigcup_{i=1}^{n} \mathcal{G}_i^{(r)}$. $\square$

Il Teorema 6.2.3 vale anche per $\mathbf{A}^T$ (attenzione all'uso di 'T', non 'H'), e quindi per le colonne di $\mathbf{A}$:

$$\mathrm{spec}(\mathbf{A}^T) \subset \bigcup_{i=1}^{n} \mathcal{G}_i^{(c)},$$

dove $\mathcal{G}_j^{(c)} := \left\{ z \in \mathbb{C} \text{ t.c. } |z - \mathbf{A}_{jj}| \leq \sum_{\substack{i=1 \\ i\neq j}}^{n} |\mathbf{A}_{ij}| \right\}$. Si ha quindi che

$$\mathrm{spec}(\mathbf{A}) \subset \left(\bigcup_{i=1}^{n} \mathcal{G}_i^{(r)} \right) \cap \left(\bigcup_{i=1}^{n} \mathcal{G}_i^{(c)} \right).$$

Teorema 6.2.4 (Secondo teorema di Gerschgorin) *Sia $\mathcal{I} \subset \{1, \ldots, n\}$ un insieme di indici. Se*

$$\left(\bigcup_{i\in\mathcal{I}} \mathcal{G}_i^{(r)} \right) \cap \left(\bigcup_{i\in\{1,\ldots,n\}\setminus\mathcal{I}} \mathcal{G}_i^{(r)} \right) = \emptyset,$$

allora ci sono esattamente $|\mathcal{I}|$ autovalori in $\bigcup_{i\in\mathcal{I}} \mathcal{G}_i^{(r)}$ e $n - |\mathcal{I}|$ in $\bigcup_{i\in\{1,\ldots,n\}\setminus\mathcal{I}} \mathcal{G}_i^{(r)}$, dove $|\mathcal{I}|$ indica la cardinalità dell'insieme $\mathcal{I}$.

Dimostrazione. Sia $A = D + M$ con D la diagonale di A. Per $t \in [0, 1]$, sia $A_t = D + tM$. Allora $A_0 = D$ e $A_1 = A$, e gli autovalori di A_t sono funzioni continue di t. Applicando il primo teorema di Gerschgorin ad A_t, troviamo che per $t = 0$, $|\mathcal{I}|$ autovalori di A_0 si trovano nel primo insieme e $n - |\mathcal{I}|$ sono nel secondo insieme (sono i centri dei dischi), con le loro molteplicità algebriche. Dato che per $0 \leq t \leq 1$ analogamente tutti gli autovalori di A_t devono stare in questi dischi, segue per continuità che anche $|\mathcal{I}|$ autovalori di A stanno nel primo insieme, ed i rimanenti nel secondo. $\square$.

Esempio 6.2.5 Sia $A = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1/2 & 1 \\ 0 & 1 & 5 \end{bmatrix}$. Localizzare l'autovalore piú grande di A, senza calcolare gli autovalori e stimare dal basso $\min \lambda$ (in altre parole, stimare l'intervallo spettrale di A).

Sol. I dischi di Gerschgorin per righe sono (la matrice è simmetrica per cui i dischi per colonne coincidono con quelli per righe) $\mathcal{G}_1^{(r)} = \{|z - 2| \leq 1\}$, $\mathcal{G}_2^{(r)} = \{|z - 1/2| \leq 2\}$, $\mathcal{G}_3^{(r)} = \{|z - 5| \leq 1\}$. Dunque l'autovalore massimo in modulo è contenuto in $\mathcal{G}_3^{(r)}$, per cui si avrà $4 \leq \max \lambda \leq 6$. Il secondo disco fornisce una stima inferiore per l'autovalore minimo, e dunque $-3/2 < \lambda_{\min}$.

Esempio 6.2.6 Sia $A = \begin{bmatrix} 15 & 1 & 2 \\ -1 & 6 & -2 \\ 1 & 0 & -2 \end{bmatrix}$. Stabilire se tutti gli autovalori di A sono reali senza calcolare gli autovalori stessi, e localizzare $\rho(A)$.

Sol. I dischi di Gerschgorin per righe sono $\mathcal{G}_1^{(r)} = \{|z-15| \leq 3\}$, $\mathcal{G}_2^{(r)} = \{|z-6| \leq 3\}$, $\mathcal{G}_3^{(r)} = \{|z+2| \leq 1\}$ *e sono tutti disgiunti. Quindi per il secondo Teorema di Gerschgorin ognuno di essi contiene un solo autovalore. Dato che la matrice è reale, se ci fossero autovalori complessi, anche i loro coniugati sarebbero autovalori ed apparterrebbero allo stesso disco centrato sull'asse dei numeri reali, fatto che non è possibile. Quindi tutti gli autovalori sono reali. Si ha* $\rho(A) = \max|\lambda|$. *Dal primo disco si ottiene* $12 \leq \rho(A) \leq 18$.

I teoremi di Gerschgorin possono essere utili per valutare la bontà della decomposizione numerica. Infatti, in generale, otterremo qualcosa della forma

$$\mathbf{Q}^H \mathbf{A} \mathbf{Q} = \mathbf{\Lambda} + \mathbf{E},$$

dove la matrice $\mathbf{E}$ è intrepretabile come l'errore che intercorre tra la decomposizione numerica ottenuta (e quindi approssimata) e quella esatta. I Teoremi 6.2.3 e 6.2.4 possono anche aiutare capire quanto gli autovalori di $\mathbf{Q}^H\mathbf{A}\mathbf{Q}$ siano vicini a $\mathbf{\Lambda}$, valutando i dischi di Gerschgorin per la matrice $\mathbf{\Lambda} + \mathbf{E}$.

Il prossimo teorema offre una stima della minima distanza tra un autovalore di $\mathbf{A}$ e quelli di una sua perturbazione $\mathbf{A} + \mathbf{E}$, sotto l'ipotesi che $\mathbf{A}$ sia diagonalizzabile.

Teorema 6.2.7 (di Bauer-Fike) *Sia* $\mathbf{A} \in \mathbb{C}^{n\times n}$ *diagonalizzabile cioè* $\mathbf{A} = \mathbf{X}\mathbf{\Lambda}\mathbf{X}^{-1}$ *con* $\mathbf{\Lambda}$ *diagonale, e sia* $\lambda(\mathbf{A})$ *un generico autovalore di* $\mathbf{A}$. *Se* $\mathbf{A} + \mathbf{E}$ *è una perturbazione di* $\mathbf{A}$, *allora per ogni autovalore* $\lambda(\mathbf{A} + \mathbf{E})$ *esiste almeno un autovalore di* $\mathbf{A}$, $\lambda(\mathbf{A})$, *tale che*

$$|\lambda(\mathbf{A} + \mathbf{E}) - \lambda(\mathbf{A})| \leq \kappa_2(\mathbf{X})\|\mathbf{E}\|_2.$$

dove $\kappa_2(\mathbf{X})$ *è il numero di condizionamento della matrice degli autovettori, con la norma indotta Euclidea.*

Dimostrazione. Sia (ξ, y) una autocoppia di $\mathbf{A}+\mathbf{E}$ con $\|y\|_2 = 1$. Si ha quindi che $(\mathbf{A}+\mathbf{E})y = \xi y$, cioè $\mathbf{E}y = (\xi\mathbf{I} - \mathbf{A})y$. Se $\xi \in \text{spec}(\mathbf{A})$ allora non c'è niente da dimostrare. Se invece $\xi \notin \text{spec}(\mathbf{A})$, allora la matrice $\xi\mathbf{I} - \mathbf{A}$ è non singolare e si ha

$$y = (\xi\mathbf{I} - \mathbf{A})^{-1}\mathbf{E}y,$$

da cui

$$\begin{aligned} 1 &= \|y\|_2 \leq \|(\xi\mathbf{I} - \mathbf{A})^{-1}\mathbf{E}\|_2 \, \|y\|_2 = \|\mathbf{X}^{-1}(\xi\mathbf{I} - \mathbf{\Lambda})^{-1}\mathbf{X}\mathbf{E}\|_2 \\ &\leq \|\mathbf{X}\|_2 \, \|\mathbf{X}^{-1}\|_2 \, \|(\xi\mathbf{I} - \mathbf{\Lambda})^{-1}\|_2 \, \|\mathbf{E}\|_2 = \kappa_2(\mathbf{X}) \frac{1}{\min\limits_{\lambda\in\text{spec}(\mathbf{A})} |\xi - \lambda|} \|\mathbf{E}\|_2, \end{aligned} \tag{6.2}$$

dove nell'ultima uguaglianza si è utilizzato il fatto che, essendo $(\xi\mathbf{I} - \mathbf{\Lambda})^{-1} = \text{diag}(\frac{1}{\xi-\lambda_1}, \ldots, \frac{1}{\xi-\lambda_n})$, si ha

$$\|(\xi\mathbf{I} - \mathbf{\Lambda})^{-1}\|_2 = \max_{\lambda_i\in\text{spec}(\mathbf{A})} \frac{1}{|\xi - \lambda_i|} = \frac{1}{\min\limits_{\lambda_i\in\text{spec}(\mathbf{A})} |\xi - \lambda_i|}.$$

Dalla disuguaglianza (6.2) segue il risultato. □

Si osserva quindi che se $\mathbf{A}$ è diagonalizzabile, allora i suoi autovalori variano in funzione di $\|\mathbf{E}\|_2$, ma che la perturbazione di $\mathbf{A}$ può essere enormemente amplificata dal fattore $\kappa_2(\mathbf{X})$, se la matrice degli autovettori è mal condizionata. Questo può avere luogo se alcuni degli autovettori sono quasi linearmente dipendenti.

Per matrici normali, ed in particolari simmetriche, la perturbazione ha in generale effetti molto meno devastanti. Infatti si ha il seguente risultato.

Corollario 6.2.8 *Se* $\mathbf{A} \in \mathbb{C}^{n\times n}$ *è normale allora* $\kappa_2(\mathbf{X}) = 1$ *da cui segue che*

$$|\lambda(\mathbf{A}+\mathbf{E}) - \lambda(\mathbf{A})| \leq \|\mathbf{E}\|_2.$$

Quindi se $\mathbf{A}$ è normale, allora il problema del calcolo dei suoi autovalori è ben condizionato rispetto all'errore assoluto.

Per matrici non normali si preferisce analizzare la sensibilità di ogni singolo autovalore, distinguendo il caso di un autovalore di molteplicità uno dal caso di un autovalore di molteplicità maggiore di uno. Il seguente risultato, di grande importanza per la teoria della perturbazione spettrale di matrici non normali, descrive la perturbazione di un autovalore mediante l'uso dello sviluppo di Taylor della perturbazione ϵ per $\epsilon \to 0$ ($\epsilon = 0$ equivale ad una perturbazione nulla della matrice $\mathbf{A}$), corrispondente all'autovalore originario.

Teorema 6.2.9 *Sia* $\mathbf{A} \in \mathbb{C}^{n\times n}$ *(in generale non normale). Siano* λ *un suo autovalore semplice e* x *e* y *i relativi autovettori destro e sinistro di norma unitaria. Allora esiste un intorno dell'origine in cui sono definite le funzioni* $\lambda(\varepsilon)$ *e* $x(\varepsilon)$ *tali che*

1. $(\mathbf{A}+\varepsilon\mathbf{E})x(\varepsilon) = \lambda(\varepsilon)x(\varepsilon)$, *con* $\mathbf{E} \in \mathbb{C}^{n\times n}$, $\|\mathbf{E}\|_2 = 1$ *e* $\lambda(\varepsilon)$ *semplice;*
2. $\lambda(0) = \lambda$ *e* $x(0) = x$*;*
3. $\lambda'(0) = \dfrac{y^H\mathbf{E}x}{y^Hx}$ *da cui* $\lambda(\varepsilon) = \lambda + \varepsilon\dfrac{y^H\mathbf{E}x}{y^Hx} + \mathcal{O}(\|\varepsilon\mathbf{E}\|_2^2)$ *per* $\varepsilon \to 0$.

Il Teorema 6.2.9 mostra che in prima approssimazione (cioè al primo ordine) la variazione nell'autovalore λ dovuta alla perturbazione $\varepsilon\mathbf{E}$ di $\mathbf{A}$ è data da $\varepsilon\frac{y^H\mathbf{E}x}{y^Hx}\varepsilon$. In particolare, si ha

$$|\lambda'(0)| = \left|\frac{y^H\mathbf{E}x}{y^Hx}\right| \leq \frac{1}{|y^Hx|},$$

e la quantità $\frac{1}{|y^Hx|}$ prende il nome di *numero di condizionamento* dell'autovalore λ. Maggiore è l'angolo tra gli autovettori destro e sinistro di $\mathbf{A}$, più grande sarà il numero di condizionamento dell'autovalore associato. Quindi per la stessa matrice, ci possono essere autovalori semplici ben condizionati, ed altri mal condizionati.

Se $\mathbf{A}$ è normale, allora $y^Hx = 1$ in quanto $\mathbf{X}$ è unitaria e $\mathbf{Y}^H = \mathbf{X}^H$. Quindi gli autovalori di una tale matrice sono tutti ben condizionati, ed ad una perturbazione di norma ε della matrice $\mathbf{A}$, tutti i suoi autovalori vengono perturbati di $c\varepsilon$ con $c = \mathcal{O}(1)$.

D'altra parte, se $\mathbf{A}$ ha blocchi di Jordan, si ha che per ogni autovalore con blocco di Jordan di dimensioni maggiori di uno vale $y^Hx = 0$. Infatti, consideriamo per semplicità $\mathbf{A} = \mathbf{J}(\lambda) \in \mathbb{R}^{n\times n}$, $n > 1$. Allora gli autovettori destro e sinistro sono $x = e_1$ e $y = e_n$, e soddisfano $y^Hx = 0$. Se più in generale $\mathbf{A} = \mathbf{Q}\mathbf{J}(\lambda)\mathbf{Q}^{-1}$, si ha che $\hat{x} = \mathbf{Q}x$, $\hat{y} = \mathbf{Q}^{-H}y$ sono autovettori destro e sinistro di λ. Vale quindi $\hat{y}^H\hat{x} = y^H\mathbf{Q}^{-1}\mathbf{Q}x = y^Hx = 0$. Per una matrice con più blocchi di Jordan si procede in modo analogo. Si noti che il teorema precedente non si applica a blocchi di Jordan (vale per autovalori semplici!), ma siccome esistono matrici diagonalizzabili con autovalori distinti vicini quanto si vuole a blocchi di Jordan, si può pensare al caso del blocco di Jordan come ad un caso limite.

Ne segue che la quantità $1/|y^Hx|$ è una misura di quanto l'autovalore ed i suoi autovettori si avvicinino ad essere parte di un blocco di Jordan, cioè di quanto quell'autovalore contribuisca a rendere la matrice da normale a quasi non diagonalizzabile.

Nel caso in cui $\mathbf{A}$ non sia diagonalizzabile e $\lambda \in \text{spec}(\mathbf{A})$ sia un autovalore avente blocchi di Jordan di dimensione al più p, allora si può dimostrare che

$$|\lambda(\mathbf{A}+\varepsilon\mathbf{E}) - \lambda(\mathbf{A})| \leq \gamma\,\varepsilon^{1/p},$$

dove γ dipende dagli autovettori e dai vettori principali corrispondenti.

6.3 Il metodo delle potenze

Il *metodo delle potenze* è un metodo iterativo particolarmente adatto per il calcolo dell'autovalore *dominante* della matrice $\mathbf{A} \in \mathbb{R}^{n\times n}$, ovvero quello di modulo massimo, λ_1, ed il relativo autovettore associato. La soluzione di tale problema è di grande interesse in numerose applicazioni reali dove il calcolo di tale autovalore, e del corrispondente autovettore, è collegato alla determinazione di certe grandezze fisiche.

Sia quindi $\mathbf{A} \in \mathbb{R}^{n\times n}$ e, fissato un vettore iniziale $x^{(0)}$ di norma unitaria, l'iterazione del metodo delle potenze consiste in

Per $i = 0, 1, 2, \ldots$, fino a convergenza,
$y = \mathbf{A}x^{(i)}$
$x^{(i+1)} = y/\|y\|$
$\lambda^{(i+1)} = (x^{(i+1)})^H \mathbf{A} x^{(i+1)}$
End

dove $\lambda^{(i+1)}$ è calcolato applicando il quoziente di Rayleigh in cui non è presente il denominatore, in quanto il vettore $x^{(i+1)}$ ha norma unitaria. Indicata con (λ_1, x_1) l'autocoppia dominante di $\mathbf{A}$, il metodo delle potenze costruisce due successioni $\{x^{(k)}\}_{k\in\mathbb{N}}$ e $\{\lambda^{(k)}\}_{k\in\mathbb{N}}$ tali che

$$x^{(k)} \to x_1, \quad \lambda^{(k)} \to \lambda_1, \qquad k \to +\infty.$$

Perchè il metodo converga, λ_1 deve essere un autovalore semplice in modulo.

Nel caso in cui $\mathbf{A}$ sia diagonalizzabile, sia $\mathbf{A} = \mathbf{X}\mathbf{\Lambda}\mathbf{X}^{-1}$ con $\mathbf{X} = [x_1, \ldots, x_n]$ matrice avente come colonne gli autovettori di $\mathbf{A}$ e $|\lambda_i|$ in ordine decrescente. Allora è possible scrivere l'i-esimo iterato $x^{(i)}$ del metodo delle potenze, opportunamente scalato, come combinazione lineare degli autovettori come segue[1]:

$$\begin{aligned}\frac{x^{(i)}}{\lambda_1^i} &= \frac{\mathbf{A}^i x^{(0)}}{\lambda_1^i} = \frac{1}{\lambda_1^i}\mathbf{X}\begin{bmatrix}\lambda_1^i & & \\ & \ddots & \\ & & \lambda_n^i\end{bmatrix}\underbrace{\mathbf{X}^{-1}x_0}_{=:\xi} = \mathbf{X}\begin{bmatrix}1 & & & \\ & \lambda_2^i/\lambda_1^i & & \\ & & \ddots & \\ & & & \lambda_n^i/\lambda_1^i\end{bmatrix}\xi \\ &= (\xi)_1 x_1 + \sum_{j=2}^{n}\frac{\lambda_j^i}{\lambda_1^i}(\xi)_j x_j \to (\xi)_1 x_1 \quad \text{per} \quad i \to \infty,\end{aligned}$$

dove si è supposto che $(\xi)_1 \neq 0$. Tale ipotesi è importante ai fini della convergenza. In assenza di informazioni specifiche, un vettore iniziale $x^{(0)}$ scelto da una distribuzione casuale (in Matlab `rand` o `randn`, per esempio), solitamente verifica tale ipotesi.

Questa relazione mostra che le iterate tendono alla direzione dell'autovettore x_1.

L'ipotesi che λ_1 sia autovalore semplice in modulo risulta cruciale per la convergenza. D'altra parte, se λ_1 è autovalore con molteplicità algebrica (e geometrica) maggiore di 1, e non ci sono altri autovalori con lo stesso modulo ma distinti da λ_1, allora la convergenza all'autovalore si ha ancora, e l'iterazione converge ad un qualsiasi vettore dell'autospazio associato.

L'ipotesi prevede che λ_1 sia semplice in modulo, e quindi reale se $\mathbf{A} \in \mathbb{R}^{n\times n}$ (il suo complesso coniugato sarebbe distinto da λ_1 ma avrebbe lo stesso modulo).

Abbiamo in pratica dimostrato il seguente risultato.

[1] Qui e nel seguito, $(x)_i$ denota la i-esima componente del vettore x.

Teorema 6.3.1 *Siano $|\lambda_i|$, $i = 1, \ldots, n$ ordinati in modo decrescente, e $|\lambda_1| > |\lambda_2|$, con $|\lambda_1|$ semplice e sia x_1 l'autovettore corrispondente. Sia $x^{(0)} = X\xi$. Se $(\xi)_1 \neq 0$ allora esiste una costante $C > 0$, tale che*

$$\|x^{(i)} - x_1\|_2 \leq C \left|\frac{\lambda_2}{\lambda_1}\right|^i, \quad i \geq 1.$$

dove $x^{(i)}$ è normalizzato in modo opportuno.

Dim. Sia $v^{(i)} := A^i x^{(0)}/((\xi)_1 \lambda_1^i)$, dove $v^{(i)}$ differisce da $x^{(i)}$ per un fattore di normalizzazione. Allora, sfruttando $A = X\Lambda X^{-1}$ e le espressioni scritte sopra, si ha $v^{(i)} = x_1 + \sum_{j=2}^n ((\xi)_j/(\xi)_1)(\lambda_j/\lambda_1)^i x_j$. Quindi

$$\|v^{(i)} - x_1\| \leq \sum_{j=2}^n \left|\frac{(\xi)_j}{(\xi)_1}\right| \left|\frac{\lambda_j}{\lambda_1}\right|^i \|x_j\| \leq \left|\frac{\lambda_2}{\lambda_1}\right|^i \sum_{j=2}^n \left|\frac{(\xi)_j}{(\xi)_1}\right| \|x_j\| =: C \left|\frac{\lambda_2}{\lambda_1}\right|^i \qquad \square.$$

Il teorema precedente mostra che la convergenza della successione $\{x^{(k)}\}_{k\in\mathbb{N}}$ all'autovettore x_1 è lineare rispetto al rapporto $|\lambda_2/\lambda_1|$. Quindi, più l'autovalore λ_1 è separato dal resto dello spettro, maggiore sarà la velocità di convergenza del metodo delle potenze.

Se $\mathbf{A}$ è reale simmetrica (ma anche complessa Hermitiana), $\mathbf{A} = \mathbf{X}\mathbf{\Lambda}\mathbf{X}^H$, con $\mathbf{X}$ unitaria, la velocità di convergenza di $\lambda^{(i)}$ all'autovalore cercato è quadratica, cioè dipende da $|\lambda_2/\lambda_1|^2$. Infatti, poichè $x^{(k)} = \alpha_k \mathbf{A}^k x^{(0)}$ per qualche α_k, si ha

$$\begin{aligned}
\lambda^{(k)} &= \frac{(x^{(k)})^H \mathbf{A} x^{(k)}}{(x^{(k)})^H x^{(k)}} = \frac{(x^{(0)})^H \mathbf{A}^{2k+1} x^{(0)}}{(x^{(0)})^H \mathbf{A}^{2k} x^{(0)}} \overset{(y^{(0)} = \mathbf{X}^H x^{(0)})}{=} \frac{(y^{(0)})^H \mathbf{\Lambda}^{2k+1} y^{(0)}}{(y^{(0)})^H \mathbf{\Lambda}^{2k} y^{(0)}} \\
&= \frac{(y^{(0)})_1^2 \lambda_1^{2k+1} + \sum_{i=2}^n (y^{(0)})_i^2 \lambda_i^{2k+1}}{\lambda_1^{2k} (y^{(0)})_1^2 + \sum_{i=2}^n (y^{(0)})_i^2 \lambda_i^{2k}} = \lambda_1 \frac{|(y^{(0)})_1|^2 + \sum_{i=2}^n |(y^{(0)})_i|^2 (\frac{\lambda_i}{\lambda_1})^{2k+1}}{|(y^{(0)})_1|^2 + \sum_{i=2}^n |(y^{(0)})_i|^2 (\frac{\lambda_i}{\lambda_1})^{2k}} \\
&\leq \lambda_1 \left(1 + \sum_{i=2}^n \frac{|(y^{(0)})_i|^2}{|(y^{(0)})_1|^2} \left(\frac{\lambda_i}{\lambda_1}\right)^{2k+1}\right) \\
&= \lambda_1 \left(1 + \left(\frac{\lambda_2}{\lambda_1}\right)^{2k+1} \frac{|(y^{(0)})_2|^2}{|(y^{(0)})_1|^2} + \sum_{i=3}^n \frac{|(y^{(0)})_i|^2}{|(y^{(0)})_1|^2} \left(\frac{\lambda_i}{\lambda_1}\right)^{2k+1}\right) = \lambda_1 \left(1 + \mathcal{O}\left(\frac{\lambda_2}{\lambda_1}\right)^{2k+1}\right),
\end{aligned}$$

dove, per ottenere la disuguaglianza, il denominatore è stato minorato con il solo primo termine, essendo tutte quantità non negative. Quindi abbiamo dimostrato che esiste una costante $C > 0$ tale che $|\lambda^{(k)} - \lambda_1| \leq C \left|\frac{\lambda_2}{\lambda_1}\right|^{2k}$, cioè la convergenza è quadratica in $|\lambda_2/\lambda_1|$. Si noti che solo l'autovalore approssimato converge in modo quadratico, mentre l'autovettore approssimato converge ancora in modo lineare.

Nel caso di $\mathbf{A}$ normale la convergenza risulterà sempre quadratica rispetto al rapporto $|\lambda_2/\lambda_1|$, dove $\lambda_i \in \mathbb{C}$, $i = 1, 2$; la dimostrazione è del tutto analoga a quella sopra.

Esempio 6.3.2 Si consideri la matrice

$$\mathbf{A} = \begin{pmatrix} 8 & 0 & -1 \\ 1 & 5 & 0 \\ 1 & -1 & -1 \end{pmatrix}.$$

I cerchi di Gerschgorin,

$$\mathcal{G}_1^{(r)} = \{|z - 8| \leq 1\}, \quad \mathcal{G}_2^{(r)} = \{|z - 5| \leq 1\}, \quad \mathcal{G}_3^{(r)} = \{|z + 1| \leq 2\},$$

dicono che l'autovalore dominante $\lambda_1 \approx 8$ è isolato e quindi semplice. Ciò significa che il metodo delle potenze convergerà.

Criterio d'arresto. Il metodo delle potenze ha bisogno di un criterio di arresto per la sua terminazione. Non avendo a disposizione l'errore, la norma euclidea del vettore residuo può essere una buona quantità calcolabile, dove il residuo è definito in questo caso come $r^{(i)} = \mathbf{A}x^{(i)} - \lambda^{(i)}x^{(i)}$. Usando un criterio relativo, il test può essere scritto come

$$\text{Se } \frac{\|r^{(i)}\|}{|\lambda^{(i)}|} < \texttt{tol} \text{ Stop}$$

Il costo computazionale dell'intera procedura deve far si che il prodotto matrice-vettore non debba essere fatto due volte. L'algoritmo originale può essere così modificato:

Fissato $x^{(0)} \in \mathbb{C}^n$ di norma unitaria, e $y = \mathbf{A}x^{(0)}$
Per $i = 0, 1, 2, \ldots,$ fino a convergenza,
  $\lambda^{(i)} = (x^{(i)})^H y$ $2n-1$ flops
  Se $\frac{\|r^{(i)}\|}{|\lambda^{(i)}|} < \texttt{tol}$ Stop $2n+1$ flops
  $x^{(i+1)} = y/\|y\|$ $3n$ flops
  $y = \mathbf{A}x^{(i+1)}$ 1 Mxv
End

Per ogni iterazione, il costo computazionale è: $(2n-1) + (2n+1) + 3n +$ 1 Mxv, cioè $\mathcal{O}(7n)$ flops + 1 Mxv.

Il criterio d'arresto utilizzato, basato sul residuo e quindi un criterio a-posteriori, può essere messo in relazione con l'effettivo errore, per valutare l'accuratezza delle quantità ottenute. In altre parole ci si chiede: Ad un residuo piccolo, corrisponde un errore piccolo? Per questo scopo possono essere utilizzati i seguenti risultati.

Teorema 6.3.3 *Sia* $\mathbf{A}$ *una matrice normale e sia* $(\widehat{\lambda}, \widehat{x})$ *l'approssimazione di una sua autocoppia. Allora* $\exists\, \lambda \in \text{spec}(\mathbf{A})$ *tale che*

$$|\lambda - \widehat{\lambda}| \leq \frac{\|\mathbf{A}\widehat{x} - \widehat{\lambda}\widehat{x}\|_2}{\|\widehat{x}\|_2},$$

Dimostrazione. Per $\mathbf{A}$ normale si ha $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^H$ con $\mathbf{Q}$ unitaria. Sia $r = (\mathbf{A} - \widehat{\lambda}\mathbf{I})\widehat{x}$ il residuo. Per $\widehat{\lambda} \notin \text{spec}(\mathbf{A})$ si ha $\widehat{x} = (\mathbf{A} - \widehat{\lambda}\mathbf{I})^{-1}r$, quindi $\|\widehat{x}\| = \|(\mathbf{\Lambda} - \widehat{\lambda}\mathbf{I})^{-1}\mathbf{Q}^H r\| \leq (1/\min_{\lambda\in\Lambda(\mathbf{A})} |\lambda - \widehat{\lambda}|)\|r\|$, da cui segue il risultato. □

Dal Teorema 6.3.3 si ha inoltre, sempre supponendo $\mathbf{A}$ normale ed anche $\|\widehat{x}\| = 1$, che

$$\frac{|\lambda - \widehat{\lambda}|}{|\widehat{\lambda}|} \leq \frac{\|\mathbf{A}\widehat{x} - \widehat{\lambda}\widehat{x}\|_2}{|\widehat{\lambda}|},$$

ed il termine di destra è proprio la quantità usata nel criterio d'arresto dell'algoritmo. In generale, per $\mathbf{A}$ non normale, la norma $\|\mathbf{A}\widehat{x} - \widehat{\lambda}\widehat{x}\|_2$ è influenzata anche dal numero di condizionamento dell'autovalore cercato, come visto nel Teorema 6.2.9. Infatti, posto $r := \mathbf{A}\widehat{x} - \widehat{\lambda}\widehat{x}$, è possibile trasformare l'equazione del residuo, $\mathbf{A}\widehat{x} - \widehat{\lambda}\widehat{x} = r$, in una relazione utile per il Teorema 6.2.9, cioè del tipo $(\mathbf{A} + \mathbf{E})x = \lambda x$ per una matrice di perturbazione $\mathbf{E}$ scelta in modo opportuno. Ponendo $\mathbf{E} := -r\widehat{x}^H$ la relazione è verificata. Per il Teorema 6.2.9 si ha quindi[2]

$$|\lambda - \widehat{\lambda}| \approx \frac{\|\mathbf{E}\|_2}{|y^H x|} = \frac{\|r\|_2}{|y^H x|},$$

[2]La relazione $\|\mathbf{E}\| = \|r\|$ segue per esempio dal fatto che $\mathbf{E}\hat{x} = r$ e che per ogni altro vettore $w \perp \hat{x}$ si ha $\mathbf{E}w = 0$, quindi $\|\mathbf{E}\| = \max_x \|\mathbf{E}x\|/\|x\| = \|r\|$.

con y e x rispettivamente autovettore sinistro e destro di $\mathbf{A}$ relativi a λ. Quindi, ad un piccolo residuo in norma non corrisponde necessariamente un piccolo errore nell'approssimazione dell'autovalore, se l'autovalore è mal condizionato. Si noti che l'errore relativo soddisfa

$$\frac{|\lambda - \widehat{\lambda}|}{|\widehat{\lambda}|} \approx \frac{1}{|y^H x|} \frac{\|r\|_2}{|\widehat{\lambda}|},$$

evidenziando ancora una volta che l'errore relativo (in avanti) è circa uguale al residuo relativo (errore all'indietro), moltiplicato per il numero di condizionamento del problema (vedi Capitolo 1).

Come ci aspettavamo dai risultati di perturbazione, l'approssimazione di un autovalore di una matrice lontana dall'essere normale potrebbe non essere facile, in quando anche a fronte di un piccolo residuo, l'autovalore approssimato potrebbe non essere una approssimazione soddisfacente dell'autovalore vero.

Concludiamo questa sezione con un esempio riguardo una possibile strategia di accelerazione in un caso particolare.

Esempio 6.3.4 *Sia $\mathbf{A} \in \mathbb{R}^{3\times 3}$ con autovalori $\lambda \in \{-0.99, 0, 1\}$. Allora il metodo delle potenze applicato ad $\mathbf{A}$ converge molto lentamente, con velocità $(0.99/1)^k = 0.99^k$. Applichiamo invece il metodo delle potenze alla matrice $\mathbf{A}+0.5\mathbf{I}$, avente autovalori $\theta = \lambda + 0.5$, cioè $\theta \in \{-0.49, 0.5, 1.5\}$. La velocità di convergenza all'autovalore dominante $\theta = 1.5$ sarà $(0.5/1.5)^k = 0.33^k$, decisamente più soddisfacente. L'autovalore λ può essere recuperato a posteriori come $\lambda = \theta - 0.5$.*

6.4 Metodo delle potenze inverse shiftate

Per approssimare autovalori diversi da quello dominante esistono metodi derivanti dal metodo delle potenze. Ad esempio, se si vuole approssimare l'autocoppia (λ_n, x_n) con λ_n autovalore più piccolo in modulo di $\mathbf{A} \in \mathbb{C}^{n\times n}$ può essere utilizzato il metodo delle potenze dove la moltiplicazione per $\mathbf{A}$ viene "sostituita" con la moltiplicazione per $\mathbf{A}^{-1}$; più correttamente, all'i-esima iterazione del metodo delle potenze si risolverà il sistema lineare

$$\mathbf{A}y = x^{(i)},$$

Questa rappresenta l'unica modifica formale all'algoritmo; Questo metodo, delle potenze inverse, è anche chiamato *metodo di Wielandt*.

Più in generale, dato $\sigma \in \mathbb{C}$ con $\sigma \notin \text{spec}(\mathbf{A})$, è possibile approssimare l'autovalore di $\mathbf{A}$ più vicino a σ in modulo. Più precisamente, sia σ tale che esiste $\lambda_m \in \text{spec}(\mathbf{A})$ che soddisfa

$$|\lambda_m - \sigma| < |\lambda_i - \sigma|, \quad \forall\, i = 1, \ldots, n,\ i \neq m. \tag{6.3}$$

(per $\sigma = 0$ si ottiene il metodo delle potenze inverse per approssimare l'autovalore più vicino a zero). In particolare, la disuguaglianza stretta in (6.3) implica che l'autovalore λ_m più vicino al parametro σ abbia molteplicità uno in modulo. Allora è possibile usare il metodo delle potenze dove al prodotto $\mathbf{A}x^{(i)}$ viene sostituita l'operazione $(\mathbf{A} - \sigma\mathbf{I})^{-1}x^{(i)}$. Questo metodo è detto *metodo delle potenze inverse traslate* (o shiftate): dato $x^{(0)} \in \mathbb{C}^n$, la sua iterazione è data dal seguente algoritmo

Per $i = 1, 2, \ldots,$ fino a convergenza,
$y = (\mathbf{A} - \sigma\mathbf{I})^{-1}x^{(i)}$
$x^{(i+1)} = y/\|y\|$
$\lambda^{(i+1)} = (x^{(i+1)})^H \mathbf{A} x^{(i+1)}$
End

L'algoritmo proposto sfrutta il fatto che gli autovettori di $\mathbf{A}$ e di $(\mathbf{A}-\sigma I)^{-1}$ rimangono invariati. Più precisamente, (λ, x) è una autocoppia di $\mathbf{A}$ se e solo se $(1/(\lambda-\sigma), x)$ è una autocoppia di $(\mathbf{A}-\sigma I)^{-1}$. Nell'algoritmo, una stima di λ si potrebbe quindi ottenere anche come $\lambda^{(i+1)} = \frac{1}{\theta} + \sigma$, dove $\theta = (x^{(i+1)})^H(\mathbf{A}-\sigma\mathbf{I})^{-1}x^{(i+1)}$ (stando ben accorti a non risolvere due sistemi lineari ad ogni iterazione!). Per motivi di accuratezza, preferiamo il calcolo di $\lambda^{(i+1)}$ come proposto nell'algoritmo, anche se questo richiede una moltiplicazione per $\mathbf{A}$.

La velocità di convergenza dipende da quanto σ è vicino all'autovalore cercato, rispetto agli altri autovalori. Se gli autovalori λ_j sono ordinati in modo che $|\lambda_1-\sigma| < |\lambda_2-\sigma| \leq \ldots \leq |\lambda_n-\sigma|$ allora la velocità di convergenza è del tipo $\mathcal{O}((|\lambda_1-\sigma|/|\lambda_2-\sigma|)^k)$.

Il costo computazionale è come quello del metodo delle potenze, con l'aggiunta del costo della risoluzione del sistema lineare $(\mathbf{A}-\sigma\mathbf{I})y = x^{(i)}$. Dato che tale risoluzione deve essere fatta ad ogni iterazione, e se le dimensioni non sono troppo elevate, è opportuno fattorizzare la matrice $\mathbf{A}-\sigma\mathbf{I}$ una volta sola, prima del ciclo iterativo, ad esempio mediante la fattorizzazione LU. Ad ogni iterazione del metodo basterà quindi risolvere i due sistemi triangolari, che richiedono un costo computazionale di $\mathcal{O}(n^2)$ se la matrice è piena. Nel caso in cui le dimensioni della matrice siano molto elevate, il sistema lineare può essere risolto in modo inesatto mediante un metodo iterativo.

6.5 L'iterazione QR

L'iterazione QR è un metodo molto sofisticato per il calcolo di tutti gli autovalori di una matrice $\mathbf{A} \in \mathbb{R}^{n\times n}$ e, se quest'ultima è normale, anche dei suoi autovettori. Questo metodo è l'algoritmo fondante del comando `eig` di MATLAB, e la sua descrizione in questo paragrafo è organizzata nel modo seguente:

- Algoritmo di base (costoso, e convergente sotto particolari condizioni);
- Considerazioni computazionali;
- Algoritmo più avanzato (riduzione in Hessenberg superiore, traslazioni) che converge sotto ipotesi più generali.

L'algoritmo determina una successione di matrici $\{\mathbf{T}_k\}_{k\in\mathbb{N}}$ tale che

$$\mathbf{T}_k := \mathbf{U}_k^H\mathbf{A}\mathbf{U}_k \to \mathbf{T}, \quad k \to +\infty,$$

con $\mathbf{T}$ matrice triangolare superiore (diagonale se $\mathbf{A}$ è normale) avente come elementi diagonali gli autovalori di $\mathbf{A}$. Come vedremo, questa iterazione tende ad ottenere la decomposizione di Schur della matrice $\mathbf{A}$, cioè

$$\mathbf{A} = \mathbf{U}_\infty\mathbf{T}\mathbf{U}_\infty^H. \tag{6.4}$$

Intuitivamente, questa procedura può essere vista come una generalizzazione alle matrici del metodo delle potenze: invece di applicare il metodo ad un vettore $x^{(k)}$, lo si applica ad una intera matrice $\mathbf{U}_k$. La normalizzazione del vettore è sostituita dalla ortogonalizzazione delle colonne della matrice. L'algoritmo risultante, che può essere applicato anche a matrici $\mathbf{U}_k$ rettangolari alte (con un numero di colonne inferiore al numero di righe), prende il nome di *iterazione del sottospazio*.

Tornando alla successione $\{\mathbf{T}_k\}_{k\in\mathbb{N}}$, questa può essere determinata come segue:

$\mathbf{T}_0 = \mathbf{A}$
Per $k = 0, 1, \ldots,$
    $\mathbf{T}_k = \mathbf{Q}_k\mathbf{R}_k$ (fattorizzazione QR di $\mathbf{T}_k$)

$\mathbf{T}_{k+1} := \mathbf{R}_k\mathbf{Q}_k$
end

Nella prima operazione viene effettuata una fattorizzazione QR della matrice quadrata $\mathbf{T}_k$. Nella seconda operazione, viene creata la nuova matrice $\mathbf{T}_{k+1}$ come prodotto dei due fattori $\mathbf{R}_k$, $\mathbf{Q}_k$ in ordine scambiato. In questo modo, dato che dalla fattorizzazione segue $\mathbf{Q}_k^H\mathbf{T}_k = \mathbf{R}_k$, si ha $\mathbf{T}_{k+1} = \mathbf{Q}_k^H\mathbf{T}_k\mathbf{Q}_k$. Essendo le matrici $\mathbf{Q}_k$ unitarie per ogni k, si ha che tutte le matrici $\mathbf{T}_k$ sono simili tra loro. Inoltre,

$$\mathbf{T}_{k+1} = \mathbf{Q}_k^H\mathbf{Q}_{k-1}^H\mathbf{T}_{k-1}\mathbf{Q}_{k-1}\mathbf{Q}_k = \ldots = \underbrace{\mathbf{Q}_k^H\mathbf{Q}_{k-1}^H\cdots\mathbf{Q}_0^H}_{\mathbf{U}_k^H}\mathbf{T}_0\mathbf{Q}_0\cdots\mathbf{Q}_{k-1}\mathbf{Q}_k =: \mathbf{U}_k^H\mathbf{A}\mathbf{U}_k,$$

con $\mathbf{U}_k$ unitaria, quindi gli autovalori di $\mathbf{A}$ vengono mantenuti.

Teorema 6.5.1 (con ipotesi restrittive) *Sia $\mathbf{A} \in \mathbb{C}^{n\times n}$ con $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n| > 0$ dove λ_i, $\forall\, i = 1, \ldots, n$ indica un autovalore di $\mathbf{A}$ (e quindi in particolare $\mathbf{A}$ è diagonalizzabile), allora*

$$\lim_{k\to+\infty} \mathbf{T}_k = \mathbf{T},$$

dove $\mathbf{T}$ è triangolare superiore (diagonale se $\mathbf{A}$ è normale) avente come elementi diagonali gli autovalori di $\mathbf{A}$, non necessariamente ordinati.

L'ipotesi $|\lambda_i| > |\lambda_{i+1}|$, $\forall i$, è molto restrittiva. D'altra parte, se ad esempio $|\lambda_r| = |\lambda_{r+1}|$ per qualche r, allora il blocco di $\mathbf{A}$

$$\begin{bmatrix} \mathbf{A}_{rr} & \mathbf{A}_{r,r+1} \\ \mathbf{A}_{r+1,r} & \mathbf{A}_{r+1,r+1} \end{bmatrix},$$

non convergerà ad una matrice della forma

$$\begin{bmatrix} \lambda_r & * \\ 0 & \lambda_{r+1} \end{bmatrix}, \quad \text{bensì a} \quad \mathbf{\Gamma} = \begin{bmatrix} \gamma_1 & \gamma_2 \\ \gamma_3 & \gamma_4 \end{bmatrix},$$

dove $\gamma_i \neq 0$ $\forall\, i = 1, \ldots, 4$ e $\operatorname{spec}(\mathbf{\Gamma}) = \{\lambda_r, \lambda_{r+1}\}$. Quindi la matrice $\mathbf{T}$ non risulterà triangolare superiore ma "quasi-triangolare superiore" nel senso che avrà qualche elemento non zero nella prima sottodiagonale. Più precisamente, la matrice $\mathbf{T}$ sarà *triangolare a blocchi*, con blocchi diagonali 1×1 o 2×2.

Il costo computazionale di questo algoritmo di base è alquanto elevato infatti la fattorizzazione QR costa un $\mathcal{O}(n^3)$ flops se $\mathbf{T}_k$ è piena (si veda il Capitolo 4), così come il costo del prodotto per $\mathbf{T}_k$ costa $\mathcal{O}(n^3)$ flops. In più la convergenza ottenuta è "solo" lineare. Nel seguito vengono descritte varie procedure per limitare il costo computazionale, e per velocizzare la convergenza.

Supponiamo che $\mathbf{T}_k$ abbia forma di Hessenberg superiore ($(\mathbf{T}_k)_{ij} = 0$ per $i > j+1$), cioè della forma

$$\mathbf{T}_k = \begin{bmatrix} * & * & \ldots & \ldots & * \\ * & * & \ddots & \ldots & * \\ & * & \ddots & \ldots & * \\ & & \ddots & & \vdots \\ & & & * & * \end{bmatrix}.$$

In tal caso, il costo della sua fattorizzazione QR sarà molto inferiore, in quanto devono essere azzerati solo gli $n-1$ elementi della sottodiagonale. Questo può essere fatto applicando le rotazioni di Givens

viste nel Capitolo 4, che annullano elementi selezioni di una matrice a basso costo computazionale. Supponendo quindi $\mathbf{A} \equiv \mathbf{A}_1$ Hessenberg superiore, definiamo la matrice

$$\mathbf{G_1} = \begin{pmatrix} \mathbf{G}^{(1)} & \\ & \mathbf{I} \end{pmatrix} \in \mathbb{R}^{n \times n},$$

dove $\mathbf{G}^{(1)} \in \mathbb{R}^{2\times 2}$ è tale che $\mathbf{G}^{(1)}(\mathbf{A}_{11}, \mathbf{A}_{21})^T = (\widehat{\mathbf{A}}_{11}, 0)^T$. Si ha dunque che

$$\mathbf{G}_1\mathbf{A}_1 = \begin{pmatrix} \widehat{\mathbf{A}}_{11} & \widehat{\mathbf{A}}_{12} & \dots & \dots & \widehat{\mathbf{A}}_{1n} \\ 0 & \widehat{\mathbf{A}}_{22} & \ddots & \dots & \widehat{\mathbf{A}}_{2n} \\ & \mathbf{A}_{32} & \ddots & \dots & \mathbf{A}_{3n} \\ & & \ddots & & \vdots \\ & & & \mathbf{A}_{n,n-1} & \mathbf{A}_{nn} \end{pmatrix} =: \mathbf{A}_2,$$

al costo di $6(n-k)$ *flops*. Il secondo passo consiste nel prodotto

$$\mathbf{G}_2\mathbf{A}_2 = \begin{pmatrix} 1 & & \\ & \mathbf{G}^{(2)} & \\ & & \mathbf{I} \end{pmatrix} \mathbf{A}_2 = \begin{pmatrix} \widehat{\mathbf{A}}_{11} & \widehat{\mathbf{A}}_{12} & \dots & & \dots & \widehat{\mathbf{A}}_{1n} \\ 0 & \widehat{\widehat{\mathbf{A}}}_{22} & \dots & & \dots & \widehat{\widehat{\mathbf{A}}}_{2n} \\ & 0 & \widehat{\mathbf{A}}_{33} & & \dots & \widehat{\mathbf{A}}_{3n} \\ & & \mathbf{A}_{43} & \ddots & \dots & \mathbf{A}_{4n} \\ & & & \ddots & \ddots & \vdots \\ & & & & \mathbf{A}_{n,n-1} & \mathbf{A}_{nn} \end{pmatrix} =: \mathbf{A}_3.$$

Dopo $n-1$ rotazioni si ottiene

$$\mathbf{Q}_0^H\mathbf{A} \equiv \mathbf{G}_{n-1}\mathbf{G}_{n-2}\cdots\mathbf{G}_1\mathbf{A} = \begin{pmatrix} * & * & \dots & \dots & * \\ & * & * & \dots & * \\ & & \ddots & & \vdots \\ & & & \ddots & \vdots \\ & & & & * \end{pmatrix} =: \mathbf{R}_1,$$

ed essendo $\mathbf{Q}_0$ unitaria (è il prodotto di $n-1$ trasformazioni unitarie), si ha $\mathbf{A} = \mathbf{Q}_0\mathbf{R}_0$, e il costo della fattorizzazione è di $\sum_{k=1}^{n-1} 6(n-k) = 6n(n-1) - 6\sum_{k=1}^{n-1} k = 3n^2 - 3n$ flops.

Il costo della moltiplicazione $\mathbf{T}_1 := \mathbf{R}_0\mathbf{Q}_0 = \mathbf{R}_0\mathbf{G}_1^H\cdots\mathbf{G}_{n-1}^H$ è di $\mathcal{O}(3n^2)$ flops. Si può verificare che $\mathbf{T}_1$ è ancora Hessenberg superiore. Si noti che non è necessario creare $\mathbf{Q}_0$ esplicitamente, ma è sufficiente memorizzare le rotazioni man mano generate (2 allocazioni di memoria per ogni rotazione).

All'iterazione successiva verranno poi applicate da sinistra $n-1$ nuove rotazioni di Givens per ottenere la fattorizzazione di $\mathbf{T}_1$, per ottenere nuovamente una matrice di tipo Hessenberg dopo l'applicazione delle rotazioni a destra, per ottenere $\mathbf{T}_2$, e così via. Questo procedimento si applica fino a convergenza, cioè fino a che gli elementi sotto la diagonale principale non saranno abbastanza piccoli, solitamente sotto l'epsilon macchina.

Il costo di ogni iterazione QR, con $\mathbf{A}$ Hessenberg superiore (e quindi tutte le successive $\mathbf{T}_k$), è di $\mathcal{O}(6n^2)$ flops, quindi di un'ordine di grandezza inferiore al caso di matrice generica $\mathbf{A}$.

Il passo che rimane da fare è quindi quello di ottenere una matrice $\mathbf{T}_0$ Hessenberg superiore partendo da una matrice $\mathbf{A}$ qualunque. Questa trasformazione può essere ottenuta applicando per

esempio le trasformazioni di Householder: $\widehat{\mathbf{T}}_0 = \mathbf{Q}^H\mathbf{A}$, dove $\mathbf{Q}^H$ è il prodotto di transformazioni di Householder, e $\widehat{\mathbf{T}}_0$ è in forma di Hessenberg superiore. Inoltre, si verifica che l'applicazione di $\mathbf{Q}$ a destra non distrugge la forma di Hessenberg, per cui $\mathbf{T}_0 = \widehat{\mathbf{T}}_0\mathbf{Q} = \mathbf{Q}^H\mathbf{A}\mathbf{Q}$ ha forma di Hessenberg superiore. Il costo computazione per ottenere $\mathbf{T}_0$ è di $\mathcal{O}(n^3)$, ma è importante sottolineare che questa operazione viene fatta una sola volta.

Riassumendo quindi, l'iterazione QR è della forma

Data $\mathbf{T}_0 = \texttt{Hess}(\mathbf{A})$,
Per $k = 0, 1, \ldots,$
$\mathbf{T}_k = \mathbf{Q}_k\mathbf{R}_k$ (QR con rotazioni di Givens)
$\mathbf{T}_{k+1} = \mathbf{R}_k\mathbf{Q}_k$
se $\max_j \frac{|(\mathbf{T}_{k+1})_{j+1,j}|}{|(\mathbf{T}_{k+1})_{j+1,j+1}|+|(\mathbf{T}_{k+1})_{j,j}|} < \epsilon$ Stop, altrimenti Continua
end

dove con `Hess` si indica la procedura per trasformare $\mathbf{A}$ in forma di Hessenberg superiore.

L'intera procedura può essere molto lenta e per accelerarla viene utilizzato uno "shift" che viene aggiornato man mano che si converge:

Data $\mathbf{T}_0 = \texttt{Hess}(\mathbf{A})$ e μ shift,
Per $k = 0, 1, \ldots,$
$\mathbf{T}_k - \mu\mathbf{I} = \mathbf{Q}_k\mathbf{R}_k$ (QR con rotazioni di Givens)
$\mathbf{T}_{k+1} = \mathbf{R}_k\mathbf{Q}_k + \mu\mathbf{I}$
se $\max_j \frac{|(\mathbf{T}_{k+1})_{j+1,j}|}{|(\mathbf{T}_{k+1})_{j+1,j+1}|+|(\mathbf{T}_{k+1})_{j,j}|} < \epsilon$ Stop, altrimenti Continua
end

La convergenza sarà più rapida per l'autovalore più vicino a μ in modulo, come previsto dal metodo delle potenze inverse traslate.

Nella pratica, si sceglie $\mu = \mathbf{T}_{k-1}(n, n)$ nelle prime iterazioni, fino a che, ad un certo $\widehat{k}$, $\mathbf{T}_{\widehat{k}-1}(n, n-1) \approx 0$. A questo punto si sceglie $\mu = \mathbf{T}_{\widehat{k}-1}(n-1, n-1)$ e la parte sottostante di $\mathbf{T}$ non viene più modificata. L'aggiornamento di μ continua poi allo stesso modo: quando all'iterazione $\widehat{\widehat{k}}$, $\mathbf{T}_{\widehat{\widehat{k}}}(n-1, n-2) \approx 0$, si porrà $\mu = \mathbf{T}_{\widehat{\widehat{k}}}(n-2, n-2)$.

È importante notare che, grazie al doppio uso dello shift ad ogni iterazione, le matrici $\mathbf{T}_k$ ottenute mediante l'iterazione QR con shift sono ancora tutte simili tra loro, infatti

$$\mathbf{T}_{k+1} = \mathbf{R}_k\mathbf{Q}_k + \mu\mathbf{I} = \mathbf{Q}_k^H(\mathbf{T}_k - \mu\mathbf{I})\mathbf{Q}_k + \mu\mathbf{I} = \mathbf{Q}_k^H\mathbf{T}_k\mathbf{Q}_k,$$

dove nell'ultima uguaglianza si è utilizzato il fatto che $\mathbf{Q}_k$ è unitaria ($\mathbf{Q}_k^H\mathbf{Q}_k = \mathbf{I}$).

L'uso del parametro μ permette di accelerare notevolmente la convergenza dell'iterazione. Infatti, se supponiamo di scegliere uno shift μ fisso, e $|\lambda_1 - \mu| \geq \cdots \geq |\lambda_n - \mu|$, allora

$$(\mathbf{T}_{k+1})_{p+1,p} \to 0, \quad k \to +\infty,$$

con la stessa velocità di $\left|\frac{\lambda_{p+1}-\mu}{\lambda_p-\mu}\right|^k$, con convergenza anche quadratica, in generale.

La relazione con $\left|\frac{\lambda_{p+1}-\mu}{\lambda_p-\mu}\right|$ mostra d'altra parte che se $\lambda_p = \lambda_{p+1}$ non si ha convergenza. Esistono comunque ulteriori varianti dell'iterazione QR con shift ("doppio shift") che permettono di gestire matrici aventi autovalori di stesso modulo.

La presente implementazione non è in grado di gestire autovalori complessi. Si può dimostrare che l'uso di un doppio shift complesso, μ e $\bar{\mu}$ in sequenza, permette di determinare anche gli autovalori complessi. Se la matrice $\mathbf{A}$ è a valori reali, questa procedura permette anche di mantenere l'iterazione con matrici reali; la matrice reale risultante sarà triangolare a blocchi con blocchi

diagonali 1×1 o 2×2. Vista la complessità del metodo nella sua interezza, ci limitiamo solo ad accennare questi sofisticati ed importanti dettagli tecnici. Approssimativamente, l'algoritmo finale con shift impiega una o due iterazioni per la convergenza di ogni autovalore, permettendo il calcolo della forma di Schur in $\mathcal{O}(n^3)$ operazioni floating point.

L'iterazione QR permette di ottenere tutti gli autovalori della matrice con una ottima accuratezza. Se richiesto, ogni autovettore può essere ottenuto mediante 1 o al massimo due iterazioni del metodo delle potenze inverse traslate, scegliendo come parametro σ di shift proprio l'autovalore approssimato trovato, $\sigma = (\mathbf{T}_k)_{j,j}$, per $j \in \{1, \ldots, n\}$. Nonostante la matrice $\mathbf{A} - \sigma I$ diventi fortemente mal condizionata, il metodo delle potenze riesce a determinare facilmente la direzione dell'autovettore cercato. Strategie opportune sono usate nel caso di autovalori non semplici in modulo.

Concludiamo con un teorema sulla stabilità dell'iterazione QR. Il seguente risultato assicura che le matrici ottenute con l'iterazione QR in aritmetica finita $\mathbf{A} \approx \widehat{\mathbf{U}}_{\hat{k}} \widehat{\mathbf{T}}_{\hat{k}} \widehat{\mathbf{U}}_{\hat{k}}^H$ siano approssimazioni stabili, nel senso dell'analisi all'indietro, dei fattori della decomposizione di Schur della matrice $\mathbf{A}$ (si veda (6.4)).

Teorema 6.5.2 *Supponiamo che l'iterazione QR converga dopo $\hat{k}$ iterazioni. Allora le matrici $\widehat{\mathbf{T}}_{\hat{k}}$ e $\widehat{\mathbf{U}}_{\hat{k}}$ effettivamente calcolate sono tali che*

$$\widehat{\mathbf{T}}_{\hat{k}} = \mathbf{Q}^H(\mathbf{A} + \mathbf{E})\mathbf{Q}, \quad \textit{con } \|\mathbf{E}\|_2 = \mathcal{O}(\mathtt{u}\|\mathbf{A}\|_2),$$

e $\mathbf{Q}$ unitaria, e

$$\widehat{\mathbf{U}}_{\hat{k}}^H \widehat{\mathbf{U}}_{\hat{k}} = \mathbf{I} + \mathbf{F}, \quad \textit{con } \|\mathbf{F}\|_2 = \mathcal{O}(\mathtt{u}).$$

In altre parole, il teorema mostra che $\widehat{\mathbf{T}}_k$ è l'esatta matrice della decomposizione di Schur per una matrice "vicina" ad $\mathbf{A}$, e che $\widehat{\mathbf{U}}_{\hat{k}}$ è vicina all'essere ortogonale.

6.6 Autovalori ed il calcolo di radici di polinomi

Un polinomio $p_n(x) = a_0 + a_1 x + \ldots + a_n x^n$ $(a_n \neq 0)$ è in generale una funzione non lineare. Le sue radici si possono determinare imponendo $p_n(x) = 0$, e possono essere quindi approssimate - singolarmente - mediante il metodo di Newton per determinare gli zeri di una funzione non lineare.

D'altra parte, è interessante notare che c'è uno stretto legame tra i polinomi e le matrici, e questo legame permette di calcolare *tutte* le radici di p_n mediante il calcolo di autovalori. A tal fine, definiamo la *matrice Companion* di p_n:

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & \cdots & 0 & -a_0/a_n \\ 1 & 0 & \cdots & 0 & -a_1/a_n \\ 0 & 1 & \cdots & 0 & -a_2/a_n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -a_{n-1}/a_n \end{bmatrix}.$$

Allora un calcolo esplicito mostra che il polinomio caratteristico di $\mathbf{C}$ è strettamente legato a p_n, cioè vale il seguente risultato

$$\det(\mathbf{C} - \lambda I) = \frac{1}{a_n}(-1)^n p_n(\lambda). \tag{6.5}$$

Dunque, gli autovalori di $\mathbf{C}$ coincidono con le radici di p_n.

Esempio 6.6.1 È dato il polinomio $p_3(x) = x^3 - 2x^2 + x - 3$, con matrice Companion

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 3 \\ 1 & 0 & -1 \\ 0 & 1 & 2 \end{bmatrix}$$

Con il calcolo esplicito del determinante si ottiene $\det(\mathbf{C} - \lambda I) = -\lambda^3 + 2\lambda^2 - \lambda + 3$, che corrisponde al polinomio dato, a meno del segno, come in (6.5).

Con il metodo QR è possibile determinare tutti gli autovalori di $\mathbf{C}$, e quindi le radici di p_n; il metodo risulta veloce specie se le radici sono distinte, non necessariamente reali.

Capitolo 7

Radici di polinomi

Questo capitolo è un'appendice all'argomento dei metodi numerici per equazioni non lineari.

7.1 Introduzione

I polinomi di grado maggiore o uguale ad uno sono funzioni non lineari algebriche, quindi il metodo di Newton è applicabile per determinare alcuni zeri. Sia quindi $p_n = p_n(x)$ un polinomio di grado $n \geq 1$ e sia ξ una sua radice. Fissato x_0, il metodo di Newton applicato a p_n costruisce la successione di iterate $\{x_k\}$ con

$$x_{k+1} = x_k - \frac{p_n(x_k)}{p_n'(x_k)}, \quad k = 0, 1, \ldots.$$

Per x_0 preso in modo opportuno, $x_k \to \xi$ per $k \to \infty$. Il calcolo di x_{k+1} richiede la valutazione di p_n e p_n' nel punto x_k, che può essere ottenuta a basso costo computazionale nel modo seguente.

Innanzi tutto notiamo che il calcolo di $p_n(x_k)$ in modo "diretto", per esempio in codice Matlab:

```
p =  3 x.^3 - 2 x.^2 + x.^1 -4
```

oltre ad essere molto costoso (si calcolano le potenze in modo separato, ripetendo lo stesso conto più volte), può dare seri problemi per l'amplificazione degli errori di arrotondamento. Un modo meno oneroso e meno instabile, ma comunque ugualmente immediato, è dato dal seguente ciclo, dove definiamo $p_n(x) = a_0 + a_1 x + \ldots + a_n x^n$:

$$\begin{array}{rcl} s & = & 1 \\ p & = & a_0 \\ \text{Per} \quad k & = & 1, \ldots, n \\ & & s = s \cdot x \\ & & p = p + a_k s \end{array}$$

Al termine, p contiene il valore di $p_n(x)$ per x fissato. Il costo computazionale comunque può essere ulteriormente abbassato, usando la *regola di Horner*. Scriviamo il polinomio come

$$p_n(x) = (\cdots(((a_n x + a_{n-1})x + a_{n-2})x + a_{n-3})x + \cdots)x + a_0,$$

e per $x = x_k$ definiamo l'iterazione

$$\begin{array}{rcl} b_0 & = & a_n \\ b_j & = & b_{j-1} x_k + a_{n-j}, \quad j = 1, \ldots, n. \end{array}$$

Al termine, si ottiene $b_n = p_n(x_k)$.

Osservazione. L'iterazione che determina i b_j può essere vista come la procedura di risoluzione di un sistema con matrice bidiagonale inferiore:

$$\begin{bmatrix} 1 & -x_k & & & & \\ & 1 & -x_k & & & \\ & & 1 & -x_k & & \\ & & & \ddots & \ddots & \\ & & & & 1 & -x_k \\ & & & & & 1 \end{bmatrix} \begin{bmatrix} b_n \\ \vdots \\ b_1 \\ b_0 \end{bmatrix} = \begin{bmatrix} a_0 \\ \vdots \\ a_{n-1} \\ a_n \end{bmatrix}.$$

Con i coefficienti b_j è possibile definire il polinomio $\widehat{p}_{n-1}(x) = b_0 x^{n-1} + b_1 x^{n-2} + \ldots + b_{n-1}$. Allora si ha

$$p_n(x) = (x - x_k)\widehat{p}_{n-1}(x) + b_n. \tag{7.1}$$

Tale relazione può essere verificata confrontando i coefficienti delle potenze del polinomio a sinistra con quelli delle potenze a destra. Infatti, per la potenza $j$esima a sinistra avremo $a_j x^j$, mentre a destra avremo il coefficiente $(b_{n-j} - x_k b_{n-j-1})x^j$. Quindi l'uguaglianza è verificata se

$$a_j = b_{n-j} - x_k b_{n-j-1}, \quad \forall j = 0, \ldots, n$$

che è vera per costruzione dei b_k.

La relazione (7.1) mostra che

$$p_n'(x) = \widehat{p}_{n-1}(x) + (x - x_k)p_{n-1}'(x), \quad \Rightarrow \quad p_n'(x_k) \equiv \widehat{p}_{n-1}(x_k).$$

Quindi, i b_j sono i coefficienti di un polinomio che in x_k coincide con il valore del polinomio derivata calcolato in x_k. Mediante nuovamente l'uso della regola di Horner è possibile valutare quindi $\widehat{p}_{n-1}(x_k)$. Riassumendo possiamo quindi definire la doppia iterazione:

$$\begin{aligned} b_0 &= a_n, \ c_0 = b_0 \\ b_j &= b_{j-1}x_k + a_{n-j}, \quad j = 1, \ldots, n \\ c_j &= c_{j-1}x_k + b_j, \quad j = 1, \ldots, n-1, \end{aligned}$$

da cui otteniamo $p_n(x_k) = b_n$ e $p_n'(x_k) = c_{n-1}$. Ad ogni iterazione di Newton vengono quindi calcolati i nuovi b_n e c_{n-1}, per determinare x_{k+1}.

Come noto, in generale la convergenza del metodo di Newton è di tipo locale. Questo vale anche per i polinomi, quindi il metodo potrebbe divergere se x_0 non è scelto in modo opportuno. In particolare, se p_n non ha radici reali, sicuramente il metodo di Newton con $x_0 \in \mathbb{R}$ divergerà, comunque sia preso x_0. D'altra parte, se le radici sono complesse, è possibile inizializzare il metodo di Newton con $x_0 \in \mathbb{C}$. I risultati di convergenza visti in $\mathbb{R}$ non saranno direttamente applicabili.

La situazione cambia significativamente se il polinomio è reale ed ha *tutte* le radici reali. Vale infatti il seguente risultato.

Teorema 7.1.1 *Sia p_n un polinomio a coefficienti reali con radici $\xi_1 \geq \xi_2 \geq \ldots \geq \xi_n$ tutte reali. Allora il metodo di Newton genera una successione $\{x_k\}$ convergente in modo monotono strettamente decrescente per ogni $x_0 > \xi_1$.*

Il problema si sposta quindi alla scelta di x_0, visto che la radice più grande ξ_1 in generale non è nota. Il seguente risultato fornisce stime dall'alto per le radici del polinomio, che possono essere usate per inizializzare il metodo di Newton. La dimostrazione segue da risultati noti sugli autovalori, come vedremo tra breve.

Teorema 7.1.2 *Le radici ξ_j, $j = 1, \ldots, n$ di un polinomio $p_n(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_0$ con $a_n \neq 0$ soddisfano*

$$\begin{aligned} |\xi_j| &\leq \max\left\{1, \sum_{k=0}^{n-1} \frac{|a_k|}{|a_n|}\right\}, \\ |\xi_j| &\leq \max\left\{\left|\frac{a_0}{a_n}\right|, 1 + \left|\frac{a_1}{a_n}\right|, \ldots, 1 + \left|\frac{a_{n-1}}{a_n}\right|\right\}. \end{aligned}$$

Il teorema assicura che se x_0 è preso almeno grande quanto il più piccolo tra i due massimi ottenuti nelle stime, allora si ha convergenza del metodo di Newton.

La convergenza può essere molto lenta, specie all'inizio. Ci sono procedure che possono essere adottate, per esempio il metodo del doppio passo, che però potrebbero portare ad un "overshooting", cioè nella perdita di convergenza per superamento della radice.

Dopo aver trovato una radice con il metodo di Newton, è possibile determinare la radice successiva mediante la procedura di *deflazione*. Se $\tilde{\xi}$ è la prima radice determinata, allora si definisce il polinomio di grado $n-1$

$$p_{n-1}(x) = \frac{p_n(x)}{(x - \tilde{\xi})}$$

e si ripete la procedura con p_{n-1} per trovare le radici successive. Se è noto che tutte le radici sono reali e $\tilde{\xi}$ approssima la radice più a destra, allora tale approssimazione può essere presa come valore iniziale x_0 della nuova iterazione di Newton. La procedura di deflazione è pericolosa, in quanto la radice $\tilde{\xi}$ è solo una approssimazione della vera radice ξ_1, e quindi le radici del polinomio p_{n-1} non saranno ξ_2, ξ_3, ecc., bensì $\tilde{\xi}_2, \tilde{\xi}_3$, ecc., che potrebbero essere anche molto diverse da quelle del polinomio originario (vedi il successivo paragrafo sulla perturbazioni di radici). È quindi necessario determinare $\tilde{\xi}$ con grande accuratezza per procedere con la deflazione in modo preciso.

È possibile determinare il numero di radici reali di un polinomio in una certa regione mediante la generazione della cosiddetta "sequenza di Sturm", che studia il cambio di segno in un punto $x = a$ di una successione di polinomi.

7.2 Radici di polinomi ed autovalori

Dato il polinomio $p_n(x) = a_n x^n + a_{n-1}x^{n-1} + \ldots + a_0$ è possibile definire la *matrice companion*

$$C = \begin{bmatrix} 0 & & & & -\frac{a_0}{a_n} \\ 1 & 0 & & & -\frac{a_1}{a_n} \\ 0 & 1 & 0 & & -\frac{a_2}{a_n} \\ & & \ddots & \ddots & \vdots \\ & & & 1 & -\frac{a_{n-1}}{a_n} \end{bmatrix} \in \mathbb{R}^{n \times n}$$

il cui polinomio caratteristico è $\phi_C(\lambda) = \det(C - \lambda I)$. Allora si può dimostrare che

$$\phi_C(\lambda) = \frac{(-1)^n}{a_n} p_n(\lambda).$$

La relazione mostra che, a meno di un fattore di scala, il polinomio generatore di C coincide con il polinomio caratteristico di C. Ne segue che le radici di p_n coincidono con gli autovalori di C. Di conseguenza è possibile calcolare tutte le radici di p_n mediante il calcolo degli autovalori di C. A tal fine, è possibile per esempio utilizzare l'iterazione QR, che è molto efficiente ed è adatta anche nel caso di autovalori complessi. Questo approccio evita tutti i problemi visti in precedenza associati all'uso del metodo di Newton. Chiaramente, se solo alcune radici del polinomio sono richieste, allora il metodo di Newton può ancora essere di interesse o, in alternativa, il metodo delle potenze con le sue varianti.

Esempio 7.2.1 *Sia* $p_3(x) = x^3 - 2x^2 + x - 3$. *Allora*

$$C = \begin{bmatrix} 0 & 0 & 3 \\ 1 & 0 & -1 \\ 0 & 1 & 2 \end{bmatrix}.$$

Con un paio di semplici calcoli si trova che $det(C - \lambda I) = -(\lambda^3 - 2\lambda^2 + \lambda - 3)$, *che coincide col polinomio* $p_3(\lambda)$, *a meno del fattore* $(-1)^3$ *(qui* $a_n = 1$*).*

Sfruttando i dischi di Gerschgorin, è possibile localizzare gli autovalori di C, e quindi le radici di p_n mediante l'uso dei suoi coefficienti. applicando i risultati dei dischi alle righe e alle colonne di C si trovano le due stime del Teorema 7.1.2.

7.3 Stabilità delle radici

In pratica, i coefficienti di un polinomio sono raramente calcolati in modo accurato. Ci chiediamo quindi come cambiano le radici del polinomio, se i suoi cofficienti vengono perturbati. Dai risultati sugli autovalori e dal legame tra autovalori e radici mediante la matrice companion, sappiamo che la perturbazione può essere molto significativa. In particolare, se ξ è una radice semplice di p_n, la teoria degli autovalori assicura che per perturbazioni ε piccole, il polinomio perturbato avrà una radice semplice $\xi(\varepsilon)$ che tende a $\xi = \xi(0)$ per $\varepsilon \to 0$. Formalizziamo questa argomentazione, e cerchiamo di precisare il tipo di perturbazione ottenibile. A tal fine, sia (eliminiamo la dipendenza da n per alleggerire la notazione)

$$p_\varepsilon(x) = p(x) + \varepsilon g(x),$$

il polinomio perturbato, dove g è un polinomio generico di grado al più n. Questo corrisponde ad imporre una perturbazione sui coefficienti di p. Sia $\xi(\varepsilon)$ una radice del polinomio perturbato, cioè

$$p_\varepsilon(\xi(\varepsilon)) \equiv p(\xi(\varepsilon)) + \varepsilon g(\xi(\varepsilon)) = 0.$$

Derivando rispetto ad ε,

$$p'(\xi(\varepsilon))\frac{d\xi(\varepsilon)}{d\varepsilon} + g(\xi(\varepsilon)) + \varepsilon(g(\xi(\varepsilon)))' = 0$$

e per $\varepsilon = 0$ si ha

$$p'(\xi(0))\frac{d\xi(0)}{d\varepsilon} + g(\xi(0)) = 0 \quad \Rightarrow \quad \frac{d\xi(0)}{d\varepsilon} = -\frac{g(\xi(0))}{p'(\xi(0))},$$

dove ricordiamo che $\xi(0) = \xi$ è la radice del polinomio non perturbato. Quindi, usando lo sviluppo di Taylor rispetto a ε con troncamento al prim'ordine, la radice perturbata si comporta come

$$\xi(\varepsilon) \approx \xi - \varepsilon\frac{g(\xi)}{p'(\xi)}.$$

Quindi al prim'ordine, la radice perturbata dista da quella originaria per un multiplo di ε, che è la perturbazione imposta ai coefficienti. Tale perturbazione ε viene amplificata se per esempio la derivata del polinomio p è piccola in ξ, cioè se p cresce molto lentamente in un intorno di ξ, o se il valore di p' è molto più piccolo della perturbazione $\varepsilon g(\xi)$. Il prossimo esempio evidenzia questa seconda situazione.

Esempio 7.3.1 *(Wilkinson) Sia* $p(x) = (x-1)(x-2)\cdots(x-20)$ *e consideriamo la radice* $\xi = 20$. *Si ha che* $p'(20) = 19!$ *e* $a_{19} = -210$. *Consideriamo il polinomio di perturbazione* $g(x) = a_{19}x^{19}$, *cosicchè il coefficiente di* x^{19} *del polinomio perturbato è* $a_{19} + \varepsilon a_{19}$. *Dal risultato precedente segue che*

$$\xi(\varepsilon) - \xi \approx \varepsilon \frac{210 \cdot 20^{19}}{19!} \approx \varepsilon \cdot 10^{10}.$$

Dunque ad una piccolissima perturbazione del coefficiente del polinomio di $\varepsilon = 10^{-16}$, *per esempio, corrisponde una radice perturbata che differisce dalla radice originaria già nella settima cifra, cioè solo le prime 6 cifre significative sono le stesse! Per una perturbazione un pò più grande, per esempio di* $\varepsilon = 10^{-10}$, *non ci saranno cifre significative uguali, cioè le due radici saranno molto diverse.*

Come per gli autovalori, il problema diventa ancora più sensibile se la radice è multipla: la perturbazione avrà un comportamento che può dipendere non più da ε ma da $\varepsilon^{1/m}$, dove m è la molteplicità della radice.