

CALCOLO NUMERICO - I MODULO

Dispense Prima Parte - [Dispense Prima Parte](#)

Laboratorio - [Laboratorio](#)

Manuale di Sopravvivenza per Calcolo

Elemento	Notazione	Descrizione
Matrice	A	Lettera Latina Maiuscola
Vettore	x	Lettera Latina Minuscola
Vettore colonna di una matrice A	\underline{a}_i	
Scalare	α	Lettera Greca Minuscola
Insieme delle Matrici	$\mathbb{C}^{n \times m}$	Al posto di scrivere $M_{m,n}(\mathbb{C})$
Norma di Matrice	$\ \cdot\ $	Al posto di \cdot ci va una qualsiasi matrice
Matrice Identità	I	
Matrice Trasposta	A^T	
Matrice Trasposta Coniugata	A^* oppure A^H	È la trasposta di A e con i coniugati degli elementi
Autocoppia	(λ, x)	(Autovalore, Autovettore)
Traccia di una matrice	$tr(\cdot)$	Al posto di \cdot c'è una matrice qualsiasi
Valori Singoli di A	$\sigma(A)$	
Matrice Diagonale simile a A	$\Lambda(A)$	
Insieme degli autovalori di A	$Spec(A)$	
Matrice Triangolare Superiore	U	
Matrice Triangolare Inferiore	L	
Matrice Ortogonale	Q	
Matrice di Permutazione	Π	
Matrice Definita Positiva	$A \succ 0$	
Vettore temporaneo o di Lavoro	w	
Tolleranza Fissata	tol	
Vettore di Errore	e	A volte può esserci un pedice per indicare il passo
Vettore Residuo	r	
Norma del Residuo	ρ	
Matrice di Condizionamento	P	

Elemento	Notazione	Descrizione
Raggio Spettrale di una matrice A	$\rho(A)$	
Range di una Matrice A	$Im(A)$	
Nucleo di una Matrice A	$Ker(A) = N(A)$	
Rango di una Matrice A	$R(A)$	
Matrice di Proiezione	P	
Matrice di Riflessione di Householder	$P = I - \beta v^T v$	Dove $\beta = \frac{2}{\ v\ ^2}$

Considerazioni Importanti: Se non sono specificate

- Una matrice ha sempre n autovalori (anche in campo complesso)
- Una matrice è sempre non singolare (invertibile)
- I vettori se non specificato sono vettori colonna
- Con Matlab, mai calcolare inverse di matrici, né determinanti
- Evitare anche di fare cicli `for` e cercare di sfruttare le funzioni già presenti in Matlab
- Per ottimizzare i conti e farli più leggeri, sfruttare le parentesi
- Se non è specificato, si pone $x_0 = 0$
- Se possibile, fare delle sostituzioni di lato, non fare tutti i conti tutte le volte che sono richieste
- Evitare di scrivere temi, l'importante è essere sintetici

A volte alcuni elementi saranno indicati con:

- (\square): Matrice Quadrata
- (\uparrow): Matrice Triangolare Superiore
- (\downarrow): Matrice Triangolare Inferiore
- ($|$): Vettore Colonna
- ($-$): Vettore Riga
- (∇): Matrice Hessenberg Superiore
- (Δ): Matrice Hessenberg Inferiore

Base di Algebra Lineare

Definizione di Norma di una Matrice

Si definisce Norma Matriciale una funzione

$$\|\cdot\| : \mathbb{C}^{n \times n} \rightarrow \mathbb{R}$$

tale che:

1. $\|A\| \geq 0$ e $\|A\| = 0$ se e solo se A è la matrice nulla
2. $\|\alpha A\| = |\alpha| \cdot \|A\|$ con $\alpha \in \mathbb{C}$ e $A \in \mathbb{C}^{n \times n}$
3. $\|A + B\| \leq \|A\| + \|B\|$ con $A, B \in \mathbb{C}^{n \times n}$
4. $\|A \cdot B\| \leq \|A\| \cdot \|B\|$ con $A, B \in \mathbb{C}^{n \times n}$, da cui segue che $\|A^m\| \leq \|A\|^m$

Si parla di Seminorma se la proprietà 4 non vale

Volendo queste definizioni possono essere estese anche a matrici rettangolari, ma noi ci interesseremo principalmente in quelle quadrate

Definizione di Norma di Frobenius

Per $A \in \mathbb{C}^{n \times n}$, si definisce Norma di Frobenius la funzione:

$$\|A\|_F = \|A\|_{\ell_2} = \sqrt{\sum_{i,j=1}^n |a_{i,j}|^2}$$

Questa non è altro che una generalizzazione della norma euclidea per i vettori

Mostriamo alcune proprietà della norma di Frobenius:

Proposizione

$$\|A\|_F^2 = \text{tr}(A^* A)$$

Dimostrazione:

Poniamo $A = (\underline{a}_1, \underline{a}_2, \dots, \underline{a}_n)$, dove gli \underline{a}_i sono i vettori colonna della matrice A .

Allora otteniamo che

$$\begin{aligned} \text{tr}(A^* A) &= \text{tr} \left(\begin{pmatrix} \underline{a}_1^* \\ \vdots \\ \underline{a}_n^* \end{pmatrix} \cdot (\underline{a}_1 \quad \cdots \quad \underline{a}_n) \right) = \text{tr} \left(\begin{array}{ccc} \underline{a}_1^* \underline{a}_1 & & \\ & \ddots & \\ & & \underline{a}_n^* \underline{a}_n \end{array} \right) \xrightarrow{\underline{a}_i^* \underline{a}_i = \sum_{j=1}^n |a_{i,j}|^2 = \|\underline{a}_i\|^2} \text{tr} \left(\begin{array}{ccc} \|\underline{a}_1\|^2 & & \\ & \ddots & \\ & & \|\underline{a}_n\|^2 \end{array} \right) = \\ &= \sum_{j=1}^n \|\underline{a}_j\|^2 = \sum_{i=1}^n \sum_{j=1}^n |a_{i,j}|^2 = \|A\|_F^2 \end{aligned}$$

□

Dimostrazione della Somma:

Facciamo vedere che prese due matrici qualsiasi $A, B \in \mathbb{C}^{n \times n}$ si ha che $\|A + B\|_F \leq \|A\|_F + \|B\|_F$

A tal fine scriviamo la matrice A come $A = (\underline{a}_1, \underline{a}_2, \dots, \underline{a}_n)$ dove \underline{a}_i rappresentano i vettori colonna della matrice.

Per la proposizione precedente si ha che $\|A\|_F = \text{tr}(A^* A)$ e da ciò otteniamo che:

$$\begin{aligned} \|A + B\|^2 &= \text{tr}((A + B)^* \cdot (A + B)) = \text{tr}(A^* A + B^* B + A^* B + B^* A) \xrightarrow{\text{tr lineare}} \\ &= \text{tr}(A^* A) + \text{tr}(B^* B) + 2\text{tr}(A^* B) = \|A\|_F^2 + \|B\|_F^2 + 2\text{tr}(A^* B) \end{aligned}$$

Tuttavia sviluppando $\text{tr}(A^* B)$ si ottiene che

$$\begin{aligned}
tr(A^*B) &= tr \left(\begin{pmatrix} \underline{a}_1^* \\ \vdots \\ \underline{a}_n^* \end{pmatrix} \cdot (\underline{b}_1 \quad \cdots \quad \underline{b}_n) \right) = \left| \sum_{j=1}^n \underline{a}_j^* \underline{b}_j \right| \leq \sum_{j=1}^n |\underline{a}_j^* \underline{b}_j| \leq \sum_{j=1}^n \|\underline{a}_j\| \cdot \|\underline{b}_j\| = \\
&= (\|\underline{a}_1\| \quad \cdots \quad \|\underline{a}_n\|) \cdot \begin{pmatrix} \|\underline{b}_1\| \\ \vdots \\ \|\underline{b}_n\| \end{pmatrix} \leq \|(\|\underline{a}_1\| \quad \cdots \quad \|\underline{a}_n\|)\| \cdot \left\| \begin{pmatrix} \|\underline{b}_1\| \\ \vdots \\ \|\underline{b}_n\| \end{pmatrix} \right\| = \sqrt{\sum_{j=1}^n \|\underline{a}_j\|^2} \cdot \sqrt{\sum_{j=1}^n \|\underline{b}_j\|^2} = \|A\|_F \cdot \|B\|_F
\end{aligned}$$

In sintesi si ottiene che $tr(A^*B) \leq \|A\|_F \cdot \|B\|_F$

Da cui, ritornando all'equazione di prima si ottiene che:

$$\|A + B\|_F^2 = \|A\|_F^2 + \|B\|_F^2 + 2tr(A^*B) \leq \|A\|_F^2 + \|B\|_F^2 + 2\|A\|_F \cdot \|B\|_F = (\|A\|_F + \|B\|_F)^2$$

□

Sulle dispense c'è la dimostrazione per il prodotto, ossia $\|A \cdot B\|_F \leq \|A\|_F \cdot \|B\|_F$

Definizione di Norma Indotta

Si definisce Norma Matriciale Indotta dalla Norma Vettoriale la funzione:

$$\|A\|_p = \max_{0 \neq x \in \mathbb{C}^n} \frac{\|Ax\|_p}{\|x\|_p} = \max_{0 \neq x \in \mathbb{C}^n} \frac{\left\| A \cdot \frac{\|x\|_p}{\|x\|} \right\|_p}{\|x\|_p} = \max_{0 \neq x \in \mathbb{C}^n} \frac{\|x\|_p \cdot \left\| A \frac{x}{\|x\|} \right\|_p}{\|x\|_p} = \max_{0 \neq x \in \mathbb{C}^n} \left\| A \frac{x}{\|x\|_p} \right\|_p = \max_{\|x\|=1, x \in \mathbb{C}^n} \|Ax\|_p$$

Nell'ultima parte si può passare direttamente a vettori di norma 1 per questioni di comodità

Anche in questo caso è possibile applicarlo anche per le matrici rettangolari (con calcoli opportuni e dimensioni adeguate).

Con p nella definizione si intende che si possono fare norme diverse

Ma ci sono tante altre norme come la norma ℓ_1 e la norma ℓ_n

Proprietà delle Norme Indotte

1. $\|Ax\| \leq \|A\| \cdot \|x\|$
2. $\|I\| = 1$

Infatti (per la seconda proprietà) si ha che $\max_x \frac{\|Ix\|}{\|x\|} = \frac{\|x\|}{\|x\|} = 1$

Per le norme matriciali non indotte non sempre è vero, infatti per le proprietà della norma matriciale si ha che:

$$\|I\| = \|I^2\| \leq \|I\| \cdot \|I\| \Rightarrow \|I\| \geq 1$$

Per fare un esempio concreto, $\|I\|_F = \sqrt{n} \neq 1$ con $I \in \mathbb{R}^{n \times n}$ (ma la cosa era analoga anche per le matrici in $\mathbb{C}^{n \times n}$)

Lemma

Sia $\|\cdot\|$ norma matriciale indotta e sia $A \in \mathbb{R}^{n \times n}$ con $\|A\| < 1$.

Allora $I + A$ non è singolare (è invertibile) e vale

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

Dimostrazione:

Poiché $\mathbb{R}^{n \times n} \subseteq \mathbb{C}^{n \times n}$ e \mathbb{C} è un campo chiuso, si ha che sicuramente esistono n autovalori (contati con le loro molteplicità).

Sia quindi $\lambda \in \mathbb{C}$ uno di essi. Poiché esiste un autovalore, sicuramente esiste almeno un autovettore $x \in \mathbb{C}^n$ tale che $Ax = \lambda x$. Definiamo (λ, x) autocoppia di A , costituita da λ autovalore complesso e x autovettore complesso.

Allora abbiamo che vale

$$|\lambda| \cdot \|x\| = \|\lambda x\| = \|Ax\| \leq \|A\| \cdot \|x\|$$

e semplificando $\|x\| \neq 0$ (in quanto condizione necessaria per x affinché sia un autovettore è che sia non nullo) si ottiene che $|\lambda| \leq \|A\| < 1$ da cui si ha che $1 + \lambda \neq 0$ (sempre perché $|\lambda| < 1$)

Abbiamo quindi che $I + A$ è non è singolare, in quanto i suoi autovalori sono del tipo $1 + \lambda_i \neq 0, \forall i$

Per dimostrare la stima possiamo considerare:

$$I = (I + A)(I + A)^{-1} = (I + A)^{-1} + A(I + A)^{-1} \Rightarrow (I + A)^{-1} = I - A(I + A)^{-1}$$

Passando poi per le norme si ottiene che:

$$\|(I + A)^{-1}\| = \|I - A(I + A)^{-1}\| \leq \|I\| + \|A\| \cdot \|(I + A)^{-1}\| = 1 + \|A\| \cdot \|(I + A)^{-1}\|$$

Portando poi a sinistra il termine $\|A\| \cdot \|(I + A)^{-1}\|$ e raccogliendo si ottiene che

$$(1 - \|A\|) \cdot \|(I + A)^{-1}\| \leq 1 \Rightarrow \|(I + A)^{-1}\| \geq \frac{1}{1 - \|A\|}$$

□

Osservazione: La norma dell'inverso è stimata dall'alta da quanto $\|A\|$ è vicino a 1

Bisogna però tenere conto che tutte queste sono stime, quindi bisogna tenere conto che possono essere grossolane

Definizione di Numero di Condizionamento

Sia $\|\cdot\|$ una norma matriciale e sia $A \in \mathbb{C}^{n \times n}$ non singolare, si definisce numero di condizionamento di A la quantità:

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

Se si volesse specificare la norma, basta metterla al pedice $\kappa_F(A) = \|A\|_F \cdot \|A^{-1}\|_F$

Si ha che vale $\kappa(A) > 1$, infatti:

$$I = A \cdot A^{-1} \Rightarrow 1 \leq \|I\| = \|A \cdot A^{-1}\| \leq \|A\| \cdot \|A^{-1}\| = \kappa(A)$$

Se $\kappa(A)$ è molto maggiore di 1 allora si dice che A è mal condizionata

Se $\kappa(A)$ è vicino a 1 allora si dice che A è ben condizionato

Teorema: Prima caratterizzazione di $\kappa(A)$

Sia $A \in \mathbb{R}^{n \times n}$ non singolare e $\|\cdot\|$ norma matriciale indotta (dalla norma euclidea), allora

$$\min \left\{ \frac{\|E\|}{\|A\|} : A + E \text{ è singolare} \right\} = \frac{1}{\kappa(A)}$$

Dimostrazione:

Facciamo la dimostrazione in due passi:

1) Facciamo vedere che $\|E\| \geq \frac{1}{\|A^{-1}\|}$ per E che rende $A + E$ singolare

2) Costruiamo la matrice E tale che $\|E\| = \frac{1}{\|A^{-1}\|}$

Passo 1: Supponiamo per assurdo che $A + E$ sia singolare con $\|E\| < \frac{1}{\|A^{-1}\|} \xrightarrow{\|A^{-1}\| \neq 0} \|E\| \cdot \|A^{-1}\| < 1$

Possiamo scrivere $A + E$ come $A(I + A^{-1}E)$ ma per il lemma precedente, poiché $\|A^{-1} \cdot E\| \leq \|A^{-1}\| \cdot \|E\| < 1$, quindi si ha che $I + A^{-1}E$ è invertibile. Tuttavia per ipotesi abbiamo che A è non singolare, quindi invertibile, ma allora

$A(I + A^{-1}E) = A + E$, poiché è prodotto di matrici invertibili, è invertibile, contro l'ipotesi assunta che fosse singolare.

Passo 2: Costruiamo tale E . Nel fare ciò, definiamo alcuni elementi che ci torneranno utili:

$$x := \arg \max_{0 \neq x \in \mathbb{R}^n} \frac{\|A^{-1} \cdot x\|}{\|x\|} \quad y := \frac{A^{-1}x}{\|A^{-1}x\|} \quad E := -\frac{xy^T}{\|A^{-1}\|}$$

Notiamo che nella definizione di x ritorna la definizione di norma indotta di matrice, quindi x non è altro che il vettore in \mathbb{C}^n tale che viene rispettata quella condizione, per cui si ottiene che $\|A^{-1}x\| = \|A^{-1}\|$

Per l'osservazione appena fatta, possiamo notare che y non è altro che: $y := \frac{A^{-1}x}{\|A^{-1}x\|} = \frac{A^{-1}x}{\|A^{-1}\|}$

Mostriamo adesso che $\|E\| = \frac{1}{\|A^{-1}\|}$ e che $A + E$ è singolare

$$\|E\| \xrightarrow{\text{Norma Indotta}} \max_{z \neq 0} \frac{\|Ez\|}{\|z\|} \xrightarrow{\text{Def } E} \frac{\|xy^T z\|}{\|A^{-1}\| \cdot \|z\|} \xrightarrow{\|y^T z| \in \mathbb{R}} \max_{z \neq 0} \frac{\|x\| \cdot |y^T z|}{\|A^{-1}\| \cdot \|z\|} \xrightarrow{\|A^{-1}\|, \|x\| \in \mathbb{R}^+} \frac{\|x\|}{\|A^{-1}\|} \cdot \max_{z \neq 0} \frac{|y^T z|}{\|z\|}$$

È stato possibile fare questo cambiamento in quanto si ha che né $\|A^{-1}\|$ né $\|x\|$ dipendono da z . Inoltre, sempre per la definizione di Norma Indotta, si ha che $\|x\| = 1$.

E ancora, sempre per la definizione di Norma Indotta e per la definizione di y segue che $\|z\| = \|y\| = 1$. Quindi dipende solamente dalla scelta di z affinché otteniamo il valore massimo.

Per ottenere tale possiamo porre $z = y$, in modo che $\frac{|y^T z|}{\|z\|} = 1$. Da queste considerazioni si ottiene che

$$\|E\| = \max_{z \neq 0} \frac{\|x\| \cdot |y^T z|}{\|A^{-1}\| \cdot \|z\|} = \frac{\|x\|}{\|A^{-1}\|} \cdot \max_{z \neq 0} \frac{|y^T z|}{\|z\|} = \frac{1}{\|A^{-1}\|}$$

Verifichiamo nell'effettivo che $A + E$ sia singolare, calcolando $(A + E)y$:

$$(A + E)y = Ay + Ey \stackrel{\text{Def } y}{=} A \frac{A^{-1}x}{\|A^{-1}\|} - \frac{xy^T}{\|A^{-1}\|}y = \frac{A \cdot A^{-1}x}{\|A^{-1}\|} - \frac{x\|y\|^2}{\|A^{-1}\|} \stackrel{\|y\|^2=1}{=} \frac{x}{\|A^{-1}\|} - \frac{x}{\|A^{-1}\|} = 0$$

Quindi il nucleo di $A + E$ è non banale, in quanto $y \neq 0$, di conseguenza $A + E$ è singolare

□

Un paio di Considerazioni sul teorema: L'insieme presentato nell'enunciato del teorema rappresenta l'insieme delle perturbazioni di A , capaci di rendere A una matrice singolare. Visto che prendiamo una matrice invertibile qualsiasi, si ha che qualunque matrice non singolare può essere perturbata in una singolare. Inoltre, quella singolare più vicina, se normalizzata e con norma minima, "dista" da A solo $\frac{1}{\kappa(A)}$. Quindi mi serve una perturbazione tanto minore quanto è alto $\kappa(A)$. Tutto questo ritornerà assolutamente utile quando si dovranno fare le approssimazioni.

Definizione di Prodotto di Rayleigh

Date $A \in \mathbb{R}^{n \times n}$ e $x \in \mathbb{R}^n$ non nullo, quindi $\|x\| \neq 0$, si definisce prodotto i Rayleigh il prodotto:

$$\frac{x^T Ax}{x^T x}$$

Proposizione

Se A è simmetrica, con $A \subseteq \mathbb{R}^{n \times n}$ non singolare e diagonalizzabile con λ autovalori reali, allora:

$$\kappa(A) = \frac{|\lambda_{\max}|}{|\lambda_{\min}|}$$

Dove

$$\lambda_{\min} = \arg \min_{\lambda \in \text{Spec}(A)} |\lambda| \quad \lambda_{\max} = \arg \max_{\lambda \in \text{Spec}(A)} |\lambda|$$

Piccolo accenno di notazione, con $\text{Spec}(A)$ si intende l'insieme degli autovalori di A e con Λ la matrice diagonale simile a A .

Dimostrazione:

Sia $A = Q\Lambda Q^T$ con Q matrice ortogonale e Λ diagonale

Vogliamo far vedere che $\|A\|_2 = |\lambda_{\max}|$ e che $\|A^{-1}\|_2 = \frac{1}{|\lambda_{\min}|}$. Si ha che:

$$\|A\|_2^2 = \max_{0 \neq x \in \mathbb{R}^n} \frac{\|Ax\|_2^2}{\|x\|_2^2} = \max_{0 \neq x \in \mathbb{R}^n} \frac{\|Q\Lambda Q^T x\|_2^2}{\|x\|_2^2}$$

Poiché Q è ortogonale si ha che $\|Qx\|_2^2 = \|x\|_2^2$, infatti $\|Qx\|_2^2 = (Qx)^T(Qx) = x^T Q^T Qx = x^T x = \|x\|_2^2$

Tornando a sostituire sopra si ha che:

$$\|A\|_2^2 = \max_{0 \neq x \in \mathbb{R}^n} \frac{\|Q\Lambda Q^T x\|_2^2}{\|x\|_2^2} \stackrel{Q^T \text{ Ortogonale}}{=} \max_{0 \neq x \in \mathbb{R}^n} \frac{\|\Lambda Q^T x\|_2^2}{\|Q^T x\|_2^2} \stackrel{y = Q^T x}{=} \max_{0 \neq y \in \mathbb{R}^n} \frac{\|\Lambda y\|_2^2}{\|y\|_2^2}$$

Tuttavia, poiché Λ è una matrice diagonale con gli autovalori di A e y è un vettore colonna, si ha che Λy non è altro che un vettore colonna il cui i -esimo elemento è $\lambda_i y_i$. Quindi la norma di $\|\Lambda y\|_2^2$ è uguale a $\|\Lambda y\|_2^2 = (\Lambda y)^T(\Lambda y) = \sum_{i=1}^n \lambda_i^2 y_i^2$. Quindi riprendendo quanto detto prima si ottiene che

$$\|A\|_2^2 = \max_{0 \neq y \in \mathbb{R}^n} \frac{\|\Lambda y\|_2^2}{\|y\|_2^2} = \max_{0 \neq y \in \mathbb{R}^n} \frac{\sum_{i=1}^n \lambda_i^2 y_i^2}{\|y\|_2^2} \leq \max_{0 \neq y \in \mathbb{R}^n} \frac{\lambda_{\max}^2 \sum_{i=1}^n y_i^2}{\|y\|_2^2} = \lambda_{\max}^2 \max_{0 \neq y \in \mathbb{R}^n} \frac{\|y\|_2^2}{\|y\|_2^2} = \lambda_{\max}^2$$

Ottieniamo quindi che $\|A\|_2 \leq |\lambda_{\max}|$, in particolare abbiamo che questi due valori sono uguali se il vettore x scelto coincide con l'autovettore relativo all'autovalore λ_{\max} , vettore che indicheremo con x_{\max}

Quindi in conclusione se prendiamo $x = x_{\max}$ segue che $\|A\|_2 = |\lambda_{\max}|$

Dimostriamo adesso che $\|A^{-1}\|_2 = \frac{1}{|\lambda_{\min}|}$:

$$\begin{aligned} \|A^{-1}\|_2^2 &= \max_{0 \neq x \in \mathbb{R}^n} \frac{\|A^{-1}x\|_2^2}{\|x\|_2^2} \stackrel{A^{-1}=Q\Lambda^{-1}Q^T}{=} \max_{0 \neq x \in \mathbb{R}^n} \frac{\|Q\Lambda^{-1}Q^Tx\|_2^2}{\|x\|_2^2} = \max_{0 \neq x \in \mathbb{R}^n} \frac{\|\Lambda^{-1}Q^Tx\|_2^2}{\|Q^Tx\|_2^2} \stackrel{y=Q^Tx}{=} \max_{0 \neq y \in \mathbb{R}^n} \frac{\|\Lambda^{-1}y\|_2^2}{\|y\|_2^2} = \\ &= \max_{0 \neq y \in \mathbb{R}^n} \frac{\sum_{i=1}^n \lambda_i^{-2} y_i^2}{\|y\|_2^2} \leq \max_{0 \neq y \in \mathbb{R}^n} \frac{\lambda_{\min}^{-2} \sum_{i=1}^n \lambda_i^2}{\|y\|_2^2} = \max_{0 \neq y \in \mathbb{R}^n} \frac{1}{\lambda_{\min}^2} \frac{\|y\|_2^2}{\|y\|_2^2} = \frac{1}{\lambda_{\min}^2} \end{aligned}$$

Da cui, dalla definizione di $\kappa(A)$ segue che:

$$\kappa(A) = \|A^{-1}\|_2 \cdot \|A\|_2 = \frac{|\lambda_{\max}|}{|\lambda_{\min}|}$$

□

Considerazioni sul teorema: Notiamo che se la matrice è mal condizionata, allora gli autovalori della matrice saranno molto più sparsi sulla retta dei numeri reali, in particolare $|\lambda_{\min}|$ e $|\lambda_{\max}|$ saranno molto distanti fra loro. In questo caso diciamo che lo spettro è molto distribuito sulla retta reale. Viceversa è il caso in cui è ben condizionato

Definizione di Intervallo Spettrale

Si definisce Intervallo Spettrale il più piccolo intervallo contenuto in \mathbb{R} che contiene tutti gli autovalori della matrice non in valore assoluto

Soffermiamoci per un po' sul caso simmetrico reale. Come è fatta la matrice E del teorema di prima caratterizzazione? Dal teorema abbiamo che:

$$E = -\frac{xy^T}{\|A^{-1}\|} \quad \text{dove} \quad x = \arg \max_{x \neq 0} \frac{\|Ax\|}{\|x\|} \quad y = \frac{A^{-1}x}{\|A^{-1}x\|}$$

Notiamo che per avere la matrice $E + A$ più vicina possibile ad A in modo che essa sia singolare, ci basterà sostituire x con x_{\min} , ossia con l'autovettore relativo all'autovalore λ_{\min} . Infatti in questo modo la y diventa:

$$y = \frac{A^{-1}x_{\min}}{\|A^{-1}x_{\min}\|} \stackrel{Ax=\lambda x \Leftrightarrow (1/\lambda)x=A^{-1}x}{=} |\lambda_{\min}| \frac{1}{\lambda_{\min}} x_{\min} = sgn(\lambda_{\min}) x_{\min}$$

Sostituendo la nuova y ottenuta e $\|A^{-1}\|$ con $\frac{1}{|\lambda_{\min}|}$ si ha che la E diventa:

$$E = -\frac{xy^T}{\|A^{-1}\|} = -x_{\min} \cdot sgn(\lambda_{\min}) x_{\min}^T |\lambda_{\min}| = -\lambda_{\min} x_{\min} x_{\min}^T$$

Quindi la matrice E è legata all'autovalore più piccolo e si ottiene che $\|E\| = |\lambda_{\min}|$.

In parole povere, sto togliendo a A il termine associato all'autovalore più piccolo. Infatti possiamo scrivere A come:

$$A = \sum_{i=1}^n x_i \lambda_i x_i^T$$

Infatti, passando dalle autocoppie (λ_i, x_i) e sfruttando la notazione già usata precedentemente $Q = (x_1 \ \cdots \ x_n)$ abbiamo:

$$A = Q\Lambda Q^T = (x_1 \ \cdots \ x_n) \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix} \begin{pmatrix} x_1^T \\ \vdots \\ x_n^T \end{pmatrix} = \sum_{i=1}^n x_i \lambda_i x_i^T = A$$

Quanto scriviamo $A + E$ noi non facciamo altro che togliere uno di questi termini, per cui trasliamo l'autovalore che ha valore assoluto minore di una quantità pari a $-|\lambda_{\min}|$, in modo da ottenere un autovalore nullo, cosicché la matrice diventi singolare

Nota: Chiaramente non è l'unica matrice E che ha questa proprietà (considerando A matrice reale e simmetrica) Infatti, per ogni autovalore λ_j con $j = \{1, \dots, n\}$ si ha che

$$A + (-\lambda_j I) = Q(A - \lambda_j I)Q^T$$

è singolare e con norma $\|E_j\| = |\lambda_j|$, ma chiaramente non è la minima

Possiamo considerare la matrice $E_{\lambda_{min}} = -\lambda_{min}I$ in modo che la norma coincida esattamente con il valore assoluto di λ_{min} , ma ancora non è la minima, perché in questo modo trasliamo tutti gli autovalori di una quantità pari a $|\lambda_{min}|$.

Sfruttando la matrice trovata nel teorema troviamo la matrice minima E che rende $A + E$ una matrice singolare, ma che lascia intatti gli altri autovalori.

Risoluzione di Sistemi Lineari - Metodi Diretti

Il numero di condizionamento è fortemente considerato nell'analisi di stabilità di risoluzione di sistemi lineari del tipo $Ax = b$ dove, dati $A \in \mathbb{R}^{n \times n}$ invertibile e $b \in \mathbb{R}^n$, si cerca di trovare $x \in \mathbb{R}^n$ soluzione.

Il metodo numerico che la risolve è

$$(A + \delta A)(x + \delta x) = b + \delta b$$

Dove δA e δb (la prima matrice e la seconda vettore) sono le perturbazioni dei dati.

Potremmo usare altre lettere per indicarle, ma poi risulterebbe scomodo con i calcoli. Idealmente a δA corrisponderebbe a E_A mentre a δb corrisponderebbe E_b . In maniera analoga sarebbe per $\delta x = E_x$

$x + \delta x$ è la soluzione del nostro sistema, con la sua perturbazione.

Teorema

Sia $A \in \mathbb{R}^{n \times n}$ non singolare e sia $\delta A \in \mathbb{R}^{n \times n}$ la sua perturbazione tale che $\|\delta A\| \cdot \|A\| < 1$ con $\|\cdot\|$ norma matriciale indotta dalla norma vettoriale euclidea e sia $b \in \mathbb{R}^n$ un vettore con la sua perturbazione $\delta b \in \mathbb{R}^n$.

Se $x \in \mathbb{R}^n$ è soluzione del sistema lineare $Ax = b$ e δx è tale che $x + \delta x$ è soluzione di $(A + \delta A)(x + \delta x) = b + \delta b$, allora vale:

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

Considerazioni del Teorema: Quando risolviamo un sistema di questo tipo, con le perturbazioni involte, in realtà stiamo cercando un'approssimazione della soluzione, in quanto non è possibile rappresentare (e quindi dare alla macchina) i valori completi di A e di b . Proprio per questo motivo cerchiamo dei valori che siano effettivamente vicini alla realtà. E da come si vede dalla formula, la stima dipende da $\kappa(A)$, quindi è sensibile alle perturbazioni

Dimostrazione:

Cerchiamo di trovare una soluzione per $(A + \delta A)(x + \delta x) = b + \delta b$.

La prima cosa da fare è cercare di capire se ha senso quello scritto, cioè se $A + \delta A$ è singolare oppure no.

Possiamo raccogliere a sinistra A , quindi diventa: $A + \delta A = A(I + A^{-1}\delta A)$

Se passiamo per la norma di $A^{-1}\delta A$ possiamo stabilire che: $\|A^{-1}\delta A\| \leq \|A^{-1}\| \cdot \|\delta A\| < 1 \xrightarrow{\text{Lemma}} I + A^{-1}\delta A$ è invertibile.

Poiché anche A è invertibile per ipotesi, si ha che A è invertibile, quindi $A(I + A^{-1}\delta A)$ è invertibile, in quanto prodotto di matrici invertibili, quindi $A + \delta A$ è non singolare.

Tutta questa parte poteva volendo essere saltata, in quanto l'ipotesi che $\|\delta A\| \cdot \|A\| < 1$ è equivalente a $\|\delta A\| < \frac{1}{\|A^{-1}\|}$, che, per quanto detto in precedenza, implica che è invertibile

Quindi l'espressione $(A + \delta A)(x + \delta x) = b + \delta b$ ha senso. Sviluppiamola:

$$(A + \delta A)(x + \delta x) = b + \delta b \Rightarrow (A + \delta A)x + (A + \delta A)\delta x = b + \delta b \Rightarrow (A + \delta A)\delta x = -Ax - \delta Ax + b + \delta b \Rightarrow \\ \xrightarrow{Ax = b} A(I + A^{-1}\delta A)\delta x = -\delta Ax + \delta b \Rightarrow \delta x = (I + A^{-1}\delta A)^{-1}A^{-1}(-\delta Ax + \delta b)$$

Passando ora per le norme si ottiene che:

$$\|\delta x\| \leq \|(I + A^{-1}\delta A)^{-1}\| \cdot \|A^{-1}\| \cdot (\|\delta A\| \cdot \|x\| + \|\delta b\|)$$

Per il lemma si ha che $\|(I + A^{-1}\delta A)^{-1}\| \leq \frac{1}{1 - \|A^{-1}\delta A\|}$:

$$\|\delta x\| \leq \|(I + A^{-1}\delta A)^{-1}\| \cdot \|A^{-1}\| \cdot (\|\delta A\| \cdot \|x\| + \|\delta b\|) \leq \frac{\|A^{-1}\|}{1 - \|A^{-1}\delta A\|} (\|\delta A\| \cdot \|x\| + \|\delta b\|)$$

Sapendo che $\|A^{-1}\delta A\| \leq \|A^{-1}\| \cdot \|\delta A\|$ e moltiplicando sopra e sotto per $\|A\|$ si ottiene che:

$$\begin{aligned} \|\delta x\| &\leq \frac{\|A^{-1}\|}{1 - \|A^{-1}\delta A\|} (\|\delta A\| \cdot \|x\| + \|\delta b\|) \leq \\ &\leq \frac{\|A^{-1}\|}{1 - \|A^{-1}\| \cdot \|\delta A\|} (\|\delta A\| \cdot \|x\| + \|\delta b\|) \cdot \frac{\|A\|}{\|A\|} = \frac{\|A^{-1}\| \cdot \|A\|}{1 - \|A^{-1}\| \cdot \|\delta A\|} \left(\frac{\|\delta A\| \cdot \|x\|}{\|A\|} + \frac{\|\delta b\|}{\|A\|} \right) \end{aligned}$$

Adesso dividiamo tutto per $\|x\|$ e moltiplichiamo e dividiamo il denominatore della prima frazione per $\|A\|$. Si ottiene:

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

Infine, sapendo che $Ax = b$, segue che $\|b\| \leq \|Ax\| \leq \|A\| \cdot \|x\|$ da cui $\frac{1}{\|A\| \cdot \|x\|} \leq \frac{1}{\|b\|}$. Quindi si ha che:

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{\kappa(A)}{1 - \kappa(A) \frac{\|\delta A\|}{\|A\|}} \left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|A\| \cdot \|x\|} \right) \leq \frac{\kappa(A)}{1 - \kappa(A) \frac{\|\delta A\|}{\|A\|}} \left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|b\|} \right)$$

□

Osservazione: Nella formula presentata nel teorema abbiamo tutte quantità relative (nella forma di $\frac{\|\delta x\|}{\|x\|}$), ossia non tiene conto della grandezza delle misure. Infatti la grandezza delle perturbazioni deve essere analizzata in maniera assoluta (confrontata con l'usuale unità), ma va guardata in relazione dell'unità di grandezza dell'elemento stesso. Proprio per questo la si divide per la norma dei dati stessi, in questo modo la si "relativizza" rispetto alla misura stessa.

È in particolare soffermarsi su queste quantità in quanto svolgono un ruolo fondamentale nel calcolo dell'errore (che in questo caso prende il nome di *errore in avanti*).

Definizione di Errore in Avanti

Si definisce Errore in Avanti la differenza tra la soluzione esatta e la soluzione approssimata / perturbata:

$$\delta x = x - (x + \delta x)$$

Bisogna sottolineare la differenza tra approssimata e perturbata, in quanto a seconda del termine utilizzato si vanno ad intendere due scopi diversi. Il primo ha lo scopo di trovare una soluzione più semplice ad un problema complesso, il secondo è "spingersi in avanti" con le modifiche, in modo che sia ancora lo stesso problema

Definizione di Errore all'Indietro

Si definisce Errore all'Indietro la perturbazione dei dati δA e δb per cui

$$(A + \delta A)(x + \delta x) = b + \delta b$$

Il nostro compito sarà quindi quello di trovare soluzioni ad un problema semplificato ottenuto dal problema originale, mediamente più complesso. Se $\frac{\|\delta A\|}{\|A\|}$ e $\frac{\|\delta b\|}{\|b\|}$ sono piccoli, l'errore all'indietro è piccolo, quindi $x + \delta x$ è soluzione ad un problema più vicino (quindi il problema è perturbato in maniera minima).

Osservazione: Secondo l'analisi dell'errore all'indietro, l'errore di $x + \delta x$ è piccolo se $x + \delta x$ risolve un problema vicino, cioè $\frac{\|\delta A\|}{\|A\|}$ e $\frac{\|\delta b\|}{\|b\|}$ sono piccoli.

Dal precedente teorema segue che l'errore in avanti è circa uguale (è minore o uguale) all'errore all'indietro amplificato dall'ordine di grandezza del numero di condizionamento $\kappa(A)$. Quindi non sempre è vero che se $\frac{\|\delta A\|}{\|A\|}$ e $\frac{\|\delta b\|}{\|b\|}$ sono piccoli, allora $\frac{\|\delta x\|}{\|x\|}$ è piccolo, dipende infatti da $\kappa(A)$. Infatti, se il numero di condizionamento $\kappa(A)$ è grande, ad un errore all'indietro piccolo corrisponde un errore in avanti grande.

Esempio della Matrice di Hilbert

La matrice di Hilbert è un esempio di una matrice malcondizionata.

Definita come:

$$H_n \in \mathbb{R}^{n \times n} \quad (H_n)_{i,j} = \frac{1}{i+j-1} \quad \text{con } n \in \mathbb{N}$$

Abbiamo che la matrice H_n è della forma:

$$\begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

Seppur definita in maniera così semplice, si ha che ha un numero di condizionamento incredibilmente alto, infatti:

- con $n = 4$, si ha che $\kappa(H_4) = 1,55 \cdot 10^6$
- con $n = 8$, si ha che $\kappa(H_8) = 1,52 \cdot 10^{10}$
- generalmente per un qualsiasi n , $\kappa(H_n)$ cresce come $e^{3,5n}$

Quindi dati $b, b + \delta b$ vettori delle soluzioni desiderate (la prima reale, la seconda perturbata), si ha che c'è uno sbalzo di errore di grandezza dell'ordine di 10^9 tra queste e $x, x + \delta x$ soluzioni dei sistemi reale / perturbato.

Infatti se $b, n + \delta b$ sono dell'ordine di 10^{-12} si ha che le soluzioni sono dell'ordine di 10^{-4} .

Per un esempio più concreto, Virtuale oppure dispense

Esistono due classi distinte di algoritmi per la risoluzione di sistemi lineari del tipo $Ax = b$, con $A \in \mathbb{R}^{n \times n}$ non singolare e $b \in \mathbb{R}^n$. Questi prendono il nome di Metodi Diretti e Metodi Iterativi:

Definizione di Metodo Diretto

Un metodo per la risoluzione di sistemi lineari si definisce Diretto se in un numero finito di passi si ottiene la soluzione

Definizione di Metodo Iterativo

Un metodo per la risoluzione di sistemi lineari si definisce Iterativo se si ottiene una soluzione dopo un numero "infinito" di passi. (Per semplificare l'idea, è un concetto simile a quello delle successioni, dove ogni indice n rappresenta un'approssimazione)

Ecco una serie di metodi diretti che dovranno essere utilizzati in maniera diversa a seconda dei casi.

Per i metodi iterativi ne vedremo pochi, in generale quelli più efficienti che hanno dato nel tempo a partire dal secolo scorso i migliori risultati.

Caso 0 - Matrice Diagonale

Il caso della matrice diagonale è il caso più semplice, in quanto la matrice diagonale è della forma:

$$D = \begin{pmatrix} D_{1,1} & & & \\ & D_{2,2} & & \\ & & \ddots & \\ & & & D_{n,n} \end{pmatrix} = diag(D_{1,1}, D_{2,2}, \dots, D_{n,n})$$

Dove ogni elemento non rappresentato è 0

Banalmente il sistema lineare rappresentato da $Dx = b$ è rappresentabile come:

$$\begin{pmatrix} D_{1,1} & & & \\ & D_{2,2} & & \\ & & \ddots & \\ & & & D_{n,n} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \Leftrightarrow \begin{cases} D_{1,1}x_1 = b_1 \\ D_{2,2}x_2 = b_2 \\ \dots \\ D_{n,n}x_n = b_n \end{cases}$$

Dove per ogni i la soluzione è della forma $x_i = b_i/D_{i,i}$

NON È CICLO E NON DEVE ESSERE FATTO COME CICLO: Infatti in Matlab c'è l'opzione di trasformare la matrice diagonale in un vettore con `Dfull = diag(D);` e poi fare la divisione componente per componente `x = b ./ Dfull`.

Naturalmente questo è possibile se le dimensioni dei vettori lo consentono.

Cerchiamo di capirne il costo computazionale: poiché ci sono n flop in quanto nell'effettivo vengono fatte n divisioni.

Caso 1 - Matrice Triangolare

Possiamo distinguere le matrici triangolari superiori U e inferiori L

Facciamo prima il caso delle matrici triangolari inferiori.

Sia quindi $Lx = b$ il sistema lineare con $L \in \mathbb{R}^{n \times n}$ matrice triangolare inferiore. Allora è equivalente a:

$$\begin{pmatrix} L_{1,1} & & & \\ L_{2,1} & L_{2,2} & & \\ \vdots & & \ddots & \\ L_{n,1} & \dots & \dots & L_{n,n} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \Leftrightarrow \begin{cases} L_{1,1}x_1 = b_1 \\ L_{2,1}x_1 + L_{2,2}x_2 = b_2 \\ \dots \\ \dots \end{cases}$$

In generale otterremo che $\forall i \in \{1, \dots, n\}$ la soluzione è

$$x_i = \frac{b_i - \sum_{j=1}^{i-1} (L_{i,j}x_j)}{L_{i,i}}$$

In questo caso si parla di **Sostituzione in Avanti**

Facciamo il costo computazionale: per tutti gli i abbiamo:

$$\left. \begin{array}{l} i-1 \text{ prodotti} \\ i-2 \text{ somme} \\ 1 \text{ somma algebrica} \\ 1 \text{ divisione} \end{array} \right\} \Rightarrow (i-1) + (i-2) + 1 + 1 = 2i-2$$

Quindi in totale avremo:

$$\sum_{i=1}^n 2(i-1) = \sum_{i=1}^n (2i) - \sum_{i=1}^n 2 = 2 \left(\frac{n(n+1)}{2} \right) - n = n^2$$

Quindi ha un costo computazionale pari a n^2 .

Se n è piccolo, è molto efficiente, altrimenti non ci sono altri metodi di risoluzione

Con la matrice triangolare superiore è sostanzialmente la stessa cosa, però si parla di **Sostituzione all'indietro**. Quindi:

$$Ux = b \Rightarrow \begin{pmatrix} U_{1,1} & U_{1,2} & \dots & U_{1,n} \\ & U_{2,2} & \dots & U_{2,n} \\ & & \ddots & \vdots \\ & & & U_{n,n} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

Proprio come nel caso precedente si avrà che la soluzione generale sarà:

$$x_i = \frac{b_i - \sum_{j=1}^{i+1} (U_{i,j}x_j)}{U_{i,i}}$$

Il calcolo computazionale è sempre lo stesso, quindi è pari a n^2

È possibile anche dimostrare che la soluzione x ottenuta mediante l'algoritmo con sostituzione in avanti è la soluzione esatta del problema $(L + \delta L)x = b$, cioè ponendo $\delta b = 0$, e che:

$$\|\delta L\|_F \leq \frac{nu}{1-nu} \|L\|_F$$

Dove u è l'**eps** della macchina e n è la dimensione dei dati

Notiamo anche che per un n molto piccolo abbiamo che la cosa è approssimabile a:

$$\|\delta L\|_F \leq \frac{nu}{1-nu} \|L\|_F \approx \Theta((nu)^2)$$

Inoltre, per l'errore in avanti, vale che se $1 - nu\kappa_F(L) > 0$, allora vale:

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{n\kappa_F(L)}{1 - n\mu\kappa_F(L)}$$

Questo che abbiamo appena visto non è altro che un'applicazione esplicita di quanto visto nella parte iniziale del corso

Caso 2 - Matrice Qualsiasi

Quando abbiamo una matrice $A \in \mathbb{R}^{n \times n}$ e un sistema lineare $Ab = b$ con $b \in \mathbb{R}^n$, la procedura standard è quella di trasformare la matrice A in due matrici più semplici, cioè si cercano due matrici B, C tali che

$$A = BC \quad \Rightarrow \quad BCx = b$$

In questo modo abbiamo due sistemi lineari da risolvere, prima $By = b$ e poi $Cx = y$

Chiaramente se non porta ad una semplificazione del problema non serve a niente.

Ma come possiamo scegliere B, C ?

Esistono diverse tipologie di fattorizzazioni:

- **Metodo di Fattorizzazione LU:** Cerchiamo L e U con L matrice triangolare inferiore e U matrice triangolare superiore tali:

$$A = LU \quad \Rightarrow \quad LUx = B$$

In questo modo abbiamo che dovremmo risolvere i sistemi lineari $Ly = b$ e $Ux = y$. Sappiamo

Visto che si tratta di risolvere sistemi lineari con matrici triangolari si sa che questo procedimento avrà un costo computazionale pari a $2n^2$ flops.

- **Metodo di Fattorizzazione di Cholesky:** Questo è per il caso particolare in cui $A \in \mathbb{R}^{n \times n}$ è simmetrica definita positiva.

In questo caso abbiamo che A può essere scomposta come fattorizzata come

$$A = \hat{L}\hat{L}^T$$

Con \hat{L} matrice triangolare inferiore (*con il cappellino per indicare che è sempre la stessa*)

- **Metodo di Fattorizzazione QR:** Possiamo trovare due matrici $Q, R \in \mathbb{R}^{n \times n}$, la prima ortogonale, la seconda triangolare superiore, tali che:

$$A = QR$$

In questo caso abbiamo che dobbiamo risolvere i sistemi $Qy = b \Leftrightarrow y = Q^Tb$ e $Rx = y$, quindi siamo nell'ordine di $\Theta(n^2)$

Questo metodo esiste per tutte le matrici, sia singolari, sia rettangolari

Esiste poi un metodo molto *naive* e molto inefficiente che riguarda se una matrice è diagonalizzabile. In tal caso se

$$A = H\Lambda H^{-1} \quad \Rightarrow \quad x = H\Lambda H^{-1}b$$

Ma è estremamente scomodo in quanto dovremmo prima trovare gli autovalori e poi determinare la matrice di cambio di base e computazionalmente parlando è estremamente costoso.

Metodo di Riduzione di Gauss - Fattorizzazione LU

Il metodo di riduzione di Gauss permette ci permette, a partire da una matrice qualsiasi, di ottenere una matrice triangolare superiore, in particolare, posto $n = 4$ otterremo che:

$$A = \begin{pmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{pmatrix} \quad \xrightarrow{\text{Gauss}} \quad A' = \begin{pmatrix} * & * & * & * \\ * & * & * & * \\ * & * & * & * \\ * & * & * & * \end{pmatrix} \in \mathbb{K}^4$$

Quindi il nostro sistema lineare diventerà

$$A^{n \times n}x = b^{1 \times n}$$

con A matrice triangolare superiore.

Ci permetterà di trovare una fattorizzazione LU per la matrice A

Per la scrittura del metodo di Gauss, definiamo $A^{(1)} = A$ e $b^{(1)} = b$

Per la prima colonna, per $i \in \{2, \dots, n\}$, andiamo a definire $m_{i,1}$ come

$$m_{i,1} = \frac{A_{i,1}^{(1)}}{A_{1,1}^{(1)}}$$

Ora andiamo a definire gli elementi della matrice $A^{(2)}$ e $b^{(2)}$:

$$\begin{aligned} A_{i,j}^{(2)} &= A_{i,j}^{(1)} - m_{i,1} A_{1,j}^{(1)} & i \in \{2, \dots, n\} \quad j \in \{1, \dots, n\} \\ b_i^{(2)} &= b_i^{(1)} - m_{i,1} b_i^{(1)} & i \in \{2, \dots, n\} \end{aligned}$$

In questo modo otteniamo un nuovo sistema lineare con le stesse soluzioni di quello precedente:

$$A^{(2)}x = b^{(2)} = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,n} \\ 0 & a_{2,2} - a_{2,1} & a_{2,3} - a_{2,1} & \cdots & a_{2,n} - a_{2,1} \\ 0 & a_{3,2} - a_{3,1} & a_{3,3} - a_{3,1} & \cdots & a_{3,n} - a_{3,1} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & a_{n,2} - a_{n,1} & a_{n,3} - a_{n,1} & \cdots & a_{n,n} - a_{n,1} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 - a_{2,1} \\ b_3 - a_{3,1} \\ \vdots \\ b_n - a_{n,1} \end{pmatrix}$$

Questo procedimento possiamo estenderlo per tutte le colonne, in particolare per la k -esima colonna si avrà che:

$$\begin{aligned} m_{i,k} &= A_{i,k}^{(k)} / A_{k,k}^{(k)} & i \in \{k+1, \dots, n\} \\ A_{i,j}^{(k+1)} &= A_{i,j}^{(k)} - m_{i,k} A_{k,j}^{(k)} & i \in \{k+1, \dots, n\} \quad j \in \{k, \dots, n\} \\ b_i^{(k+1)} &= b_i^{(k)} - m_{i,k} b_k^{(k)} & i \in \{k+1, \dots, n\} \end{aligned}$$

La matrice $A^{(k+1)}$ sarà quindi:

$$A^{(k+1)} = \begin{pmatrix} * & * & \cdots & * & *_{1,k} & *_{1,k+1} & \cdots & * & * \\ 0 & * & \cdots & * & *_{2,k} & *_{2,k+1} & \cdots & * & * \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & * & *_{k-1,k} & *_{k-1,k+1} & \cdots & * & * \\ 0 & 0 & \cdots & 0 & *_{k,k} & *_{k,k+1} & \cdots & * & * \\ 0 & 0 & \cdots & 0 & 0_{k+1,k} & *_{k+1,k+1} & \cdots & * & * \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0_{n-1,k} & *_{n-1,k+1} & \cdots & * & * \\ 0 & 0 & \cdots & 0 & 0_{n,k} & *_{n,k+1} & \cdots & * & * \end{pmatrix}$$

Quindi dopo n passi, $A^{(n)}$ sarà triangolare superiore

Vediamo ora il costo computazionale: per ogni k abbiamo:

$$\left. \begin{aligned} &n-k \text{ divisioni per la creazione di } m_{i,k} \\ &2(n-k)(n-k) \text{ somme per la creazione di } A^{(k+1)} \\ &2(n-k) \text{ somme per la creazione di } b^{(k+1)} \end{aligned} \right\} \Rightarrow 2(n-k)^2 + 3(n-k)$$

Sommando ora per tutti i k otteniamo che:

$$\begin{aligned} \sum_{k=1}^{n-1} (2(n-k)^2 + 3(n-k)) &= 2 \sum_{k=1}^{n-1} (n-k)^2 + 3 \sum_{k=1}^{n-1} (n-k) - 3 \sum_{k=1}^{n-1} k \\ &= 2n^2(n-1) + 2 \frac{(n-1)n(2n-1)}{6} - 4n \frac{(n-1)n}{2} + 3n(n-1) - 3 \frac{(n-1)n}{2} \\ &= 2n^3 + \frac{2}{3}n^3 - 2n^3 + \Theta(n^2) = \frac{2}{3}n^3 + \Theta(n^2) \end{aligned}$$

In questo modo abbiamo quindi costruito una matrice triangolare superiore.

Come possiamo costruire quella inferiore?

In un certo senso l'abbiamo già costruita. Definiamo per $k \in \{1, \dots, n\}$:

$$M_k = I - m_k e_k^T = \begin{pmatrix} 1_{1,1} & & & & \\ & \ddots & & & \\ & & 1_{k-1,k-1} & & \\ & & & 1_{k,k} & \\ & & & -m_{k+1,k} & 1_{k+1,k+1} \\ & & & \vdots & \\ & & & -m_{n,k} & \\ & & & & \ddots & \\ & & & & & 1_{n,n} \end{pmatrix} \quad \text{dove } m_k = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ m_{k+1,k} \\ \vdots \\ m_{n,k} \end{pmatrix}_{n-k}$$

Facciamo il caso per $k = 1$ esplicito:

$$M_1 = \begin{pmatrix} 1 & & & \\ -m_{2,1} & 1 & & \\ \vdots & & \ddots & \\ -m_{n,1} & & & 1 \end{pmatrix} = I - m_1 e_1^T$$

Notiamo quindi che vale:

$$M_1 A^{(1)} = \begin{pmatrix} 1 & & & \\ -m_{2,1} & 1 & & \\ \vdots & & \ddots & \\ -m_{n,1} & & & 1 \end{pmatrix} \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & & \vdots \\ a_{1,n} & a_{2,n} & \cdots & a_{n,n} \end{pmatrix} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ 0 & a_{2,2} - a_{2,1} & \cdots & a_{2,n} - a_{2,1} \\ 0 & a_{3,2} - a_{3,1} & \cdots & a_{3,n} - a_{3,1} \\ \vdots & \vdots & & \vdots \\ 0 & a_{n,2} - a_{n,1} & \cdots & a_{n,n} - a_{n,1} \end{pmatrix} = A^{(2)}$$

Notiamo quindi che per ogni $k \in \{1, \dots, n\}$ vale la seguente uguaglianza:

$$A^{(k+1)} = M_k A^{(k)}$$

Da cui segue direttamente che:

$$A^{(n)} = M_{n-1} M_{n-2} \cdots M_2 M_1 A = U$$

Quindi:

$$A = (M_{n-1} M_{n-2} \cdots M_2 M_1)^{-1} U = M_1^{-1} M_2^{-1} \cdots M_{n-2}^{-1} M_{n-1}^{-1} U$$

La matrice $M_1^{-1} M_2^{-1} \cdots M_{n-2}^{-1} M_{n-1}^{-1}$ è la matrice triangolare inferiore che stiamo cercando.

Notiamo poi che per ogni $k \in \{1, \dots, n-1\}$ vale che $M_k^{-1} = I + m_k e_k^T$, infatti:

$$M_k M_k^{-1} = (I - m_k e_k^T)(I + m_k e_k^T) = I + m_k e_k^T - m_k e_k^T - \underbrace{m_k e_k^T m_k e_k^T}_{=0} = I$$

Segue quindi che la matrice L che stiamo cercando è della forma:

$$L = (I + m_1 e_1^T)(I + m_2 e_2^T) \cdots (I + m_{n-2} e_{n-2}^T)(I + m_{n-1} e_{n-1}^T)$$

Come è fatto questo prodotto?

$$\begin{aligned} L &= (I + m_1 e_1^T)(I + m_2 e_2^T)(I + m_3 e_3^T) \cdots = (I + m_2 e_2^T + m_1 e_1^T + \underbrace{m_2 e_2^T m_1 e_1^T}_0)(I + m_3 e_3^T) \cdots = \\ &= (I + m_2 e_2^T + m_1 e_1^T)(I + m_3 e_3^T) \cdots = ((I + m_2 e_2^T + m_1 e_1^T) + (m_3 e_3^T + \underbrace{m_2 e_2^T m_3 e_3^T}_0 + \underbrace{m_1 e_1^T m_3 e_3^T}_0)) \cdots = \\ &= (I + m_2 e_2^T + m_1 e_1^T + m_3 e_3^T) \cdots = [\dots] = I + \sum_{i=1}^{n-1} m_i e_i^T \end{aligned}$$

Quindi la matrice L sarà:

$$L = \begin{pmatrix} 1 & & & & \\ m_{2,1} & 1 & & & \\ m_{3,1} & m_{3,2} & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ m_{n,1} & m_{n,2} & \cdots & m_{n,n-1} & 1 \end{pmatrix}$$

Esempio in cui le cose non sempre vanno bene

Prendiamo per esempio la matrice A come:

$$A \equiv A^{(1)} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 7 & 8 & 9 \end{pmatrix} \Rightarrow A^{(2)} = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 0 & -1 \\ 0 & -6 & -12 \end{pmatrix}$$

Abbiamo che lo 0 in posizione rompe l'algoritmo che abbiamo definito nella pagina precedente, in quanto non abbiamo definito la possibilità di cambiare righe

Per questo motivo, enunciato questo teorema:

Teorema

Sia $A \in \mathbb{R}^{n \times n}$ e siano A_k le sottomatrici dominanti principali (*minori principali o minori di nord ovest, per come li abbiamo definiti nel corso di geometria 1B*) di A di dimensione $k \times k$ (cioè $\forall k \in \{1, \dots, n-1\}, A_k \in \mathbb{R}^{k \times k}$). Se A_k è non singolare per tutti i $k \in \{1, \dots, n-1\}$, allora esiste ed è unica la fattorizzazione $A = LU$ con $L \in \mathbb{R}^{n \times n}$ triangolare inferiore con diagonale principale composta da elementi uguali a 1 e $U \in \mathbb{R}^{n \times n}$ triangolare superiore

Considerazioni del Teorema: A_k è non singolare per ogni $k \in \{1, \dots, n-1\}$. **NON** sto ipotizzando che A sia invertibile. Quindi anche una matrice singolare che rispecchia questa condizione ha una fattorizzazione unica. Se A è singolare, questo si riflette su U , sicuro non su L , in quanto continuerebbe ad avere sempre 1 sulla diagonale principale

Dimostrazione:

Mostriamo per induzione sulla dimensione della sottomatrice di A di ordine k .

Base Induttiva: Se $k = 1$, allora segue che $U = A$ e $L = (1)$

Passo Induttivo: Diamo per vero il risultato per k e mostriamolo per $k+1$

Siano quindi $d, c \in \mathbb{R}^k$ e $\alpha \in \mathbb{R}$ tale che

$$A_{k+1} = \begin{pmatrix} A_k & d \\ c^T & \alpha \end{pmatrix}$$

È vero che esistono L_{k+1} e U_{k+1} tali che $A_{k+1} = L_{k+1}U_{k+1}$?

Cioè è vero che esistono $L_k, U_k \in \mathbb{R}^{k \times k}$, $u, v \in \mathbb{R}^k$, $\beta \in \mathbb{R}$ tale che:

$$A_{k+1} = \begin{pmatrix} A_k & d \\ c^T & \alpha \end{pmatrix} = \begin{pmatrix} L & 0 \\ u^T & 1 \end{pmatrix} \begin{pmatrix} U_k & v \\ 0 & \beta \end{pmatrix} = \begin{pmatrix} L_k U_k & L_k v \\ u^T U_k & u^T v + \beta \end{pmatrix} = L_{k+1} U_{k+1}$$

Sappiamo, per ipotesi induttiva, che $A_k = L_k U_k$

Ora resta da scegliere u, b, β in modo che $A_{k+1} = L_{k+1} U_{k+1}$

- $L_k v = d$ è vero se v è soluzione di tale sistema, che esiste in quanto L_k è invertibile - non singolare

- $u^T U_k = c^T \Leftrightarrow U_k^T u = c$ è vero se u è soluzione di tale sistema, soluzione che esiste in quanto U_k è non singolare, dal momento che $A_k = L_k U_k$ e A_k è non singolare per ipotesi

- $u^T v + \beta = \alpha \Leftrightarrow \beta = \alpha - u^T v$

Quindi abbiamo dimostrato che se la proposizione è vera per k , allora è vera per $k+1$, quindi segue che la scomposizione esiste.

L'unicità della scomposizione segue dal fatto che per ogni $k \in \{1, \dots, n-1\}$, U_k e L_k sono non singolari

□

Osservazione: Se $A_{k,k}^{(k)} = 0$, allora l'algoritmo di eliminazione di Gauss si blocca.

Numericamente anche se $|A_{k,k}^{(k)}|$ è molto piccolo può dare problemi

Esempio in cui c'è problema nell'eliminazione di Gauss

Prendiamo per esempio la matrice

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

Tramite conti diretti si può dimostrare che $\kappa(A) \approx 4$, quindi la matrice è ben condizionata.

Facciamo la fattorizzazione $A = LU$ e otteniamo la matrice

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

Segue immediatamente che $\|U\|_F \geq 10^4$. Inoltre sapendo che:

$$U^{-1} = \begin{pmatrix} 10^4 & * \\ 0 & (1 - 10^{-4})^{-1} \end{pmatrix} \quad \Rightarrow \quad \kappa(U^{-1}) \approx 10^4$$

Da cui segue direttamente che $\kappa(U) \geq 10^8$, quindi U è molto mal condizionata.

Invece vogliamo che i fattori non peggiorino le proprietà di A , ossia vorremmo che

$$\kappa(A) \approx \kappa(L) \quad \text{e} \quad \kappa(A) \approx \kappa(U)$$

Per poter applicare il metodo di eliminazione di Gauss a più matrici, diamo la seguente definizione.

Definizione di Matrice Π di Permutazione

Si definisce $\Pi \in \mathbb{R}^{n \times n}$ la matrice che si ottiene a partire dall'identità scambiando righe e colonne

Osservazione: Il prodotto di matrici di permutazioni è a sua volta una matrice di permutazione

Esempi di Matrici di Permutazione

Sono esempi di Matrici di permutazione le matrici:

$$\Pi = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad \Pi' = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad \Pi'' = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

Il seguente teorema dimostra che, data una matrice A , è sempre possibile trovare una fattorizzazione $A = LU$

Teorema

Sia $A \in \mathbb{R}^{n \times n}$. Allora esiste una matrice di permutazione $\Pi \in \mathbb{R}^{n \times n}$ per cui è possibile trovare una fattorizzazione LU di ΠA , cioè si riesce a trovare una matrice L triangolare inferiore avente tutti 1 sulla diagonale principale e una matrice U triangolare superiore tali che $\Pi A = LU$

Considerazione del Teorema: La matrice di permutazione non è unica e a diverse matrici di permutazione corrispondono diverse fattorizzazioni in LU

Dimostrazione:

Procediamo per induzione su $k < n$

Base Induttiva: Per $k = 1$, abbiamo $A = (A_{1,1})$. Quindi è facilmente dimostrabile in quanto basta prendere:

$$L = (1) \quad U = (A_{1,1}) \quad \Rightarrow \quad A = LU = (A_{1,1})$$

Passo Induttivo: Supponiamo quindi che esista una fattorizzazione LU di una matrice in $\mathbb{R}^{k \times k}$, previa permutazione, e lo dimostriamo per $k + 1$. Possiamo distinguere due casi

- Il primo caso è che abbia la prima colonna nulla, ossia la matrice $A \in \mathbb{R}^{k+1 \times k+1}$ è della forma:

$$A_{k+1} = \begin{pmatrix} 0 & c^T \\ 0 & A_k \end{pmatrix} \quad \underline{0}, c \in \mathbb{R}^k, \quad A_k \in \mathbb{R}^{k \times k}$$

Per l'ipotesi induttiva abbiamo che $\exists \Pi_k : \Pi_k A = L_k U_k$, quindi vale che:

$$A_k = \Pi_k^T L_k U_k \quad \Rightarrow \quad A_{k+1} = \begin{pmatrix} 0 & c^T \\ 0 & \Pi_k^T L_k U_k \end{pmatrix}$$

In particolare otteniamo che:

$$A_{k+1} = \begin{pmatrix} 0 & c^T \\ 0 & \Pi_k^T L_k U_k \end{pmatrix} = \begin{pmatrix} 1 & \underline{0}^T \\ 0 & \Pi_k^T L_k \end{pmatrix} \begin{pmatrix} 0 & c^T \\ 0 & U_k \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & \underline{0}^T \\ 0 & \Pi_k^T \end{pmatrix}}_{\Pi_{k+1}^T} \underbrace{\begin{pmatrix} 1 & \underline{0}^T \\ 0 & L_k \end{pmatrix}}_{L_{k+1}} \underbrace{\begin{pmatrix} 0 & c^T \\ 0 & U_k \end{pmatrix}}_{U_{k+1}}$$

Da cui segue che $\Pi_{k+1} A_{k+1} = L_{k+1} U_{k+1}$

- Il secondo caso è che $\exists i \in \{1, \dots, n\}$ tale che $A_{i,1} \neq 0$ e poniamo tale valore come α

Sia $\hat{\Pi}_1$ la matrice di permutazione che scambia la prima riga di A_{k+1} con la i -esima riga di A_{k+1} , quindi:

$$\hat{\Pi}_1 A_{k+1} = \begin{pmatrix} \alpha & c^T \\ d & A_k \end{pmatrix} = \begin{pmatrix} 1 & \underline{0}^T \\ g & I_k \end{pmatrix} \begin{pmatrix} \alpha & c^T \\ 0 & B_k \end{pmatrix} \quad g, d, c, \underline{0} \in \mathbb{R}^k, \quad I_k, A_k, B_k \in \mathbb{R}^{k \times k}$$

Infatti vale che:

$$\begin{cases} 1 \cdot \alpha = \alpha \\ \alpha \cdot g = d \\ c^T \cdot 1 = c^T \\ g \cdot c^T + B_k = A_k \end{cases} \Rightarrow \begin{cases} g = \frac{d}{\alpha} \\ B_k = A_k - gc^T \end{cases}$$

Per Ipotesi induttiva abbiamo che $B_k = \Pi_k^T L_k U_k$. Abbiamo quindi che:

$$\begin{aligned} \hat{\Pi}_1 A_{k+1} &= \begin{pmatrix} 1 & \underline{0}^T \\ g & I_k \end{pmatrix} \begin{pmatrix} \alpha & c^T \\ 0 & \Pi_k^T L_k U_k \end{pmatrix} = \begin{pmatrix} 1 & \underline{0}^T \\ g & I_k \end{pmatrix} \begin{pmatrix} 1 & \underline{0}^T \\ 0 & \Pi_k^T L_k \end{pmatrix} \begin{pmatrix} \alpha & c^T \\ 0 & U_k \end{pmatrix} = \\ &= \begin{pmatrix} 1 & \underline{0}^T \\ g & \Pi_k^T L_k \end{pmatrix} \begin{pmatrix} \alpha & c^T \\ 0 & U_k \end{pmatrix} = \begin{pmatrix} 1 & \underline{0}^T \\ 0 & \Pi_k^T \end{pmatrix} \begin{pmatrix} 1 & \underline{0}^T \\ \Pi_k g & L_k \end{pmatrix} \begin{pmatrix} \alpha & c^T \\ 0 & U_k \end{pmatrix} = \tilde{\Pi}_{k+1}^T L_{k+1} U_{k+1} \end{aligned}$$

Abbiamo quindi che

$$\tilde{\Pi}_{k+1} \hat{\Pi}_1 A_{k+1} = L_{k+1} U_{k+1} \quad \Rightarrow \quad \Pi_{k+1} A_{k+1} = L_{k+1} U_{k+1}$$

□

Osservazioni: Numericamente parlando, la scelta di i tale che $A_{i,1}$ è legata alla grandezza di $A_{i,1}$, cioè prendo i tale che

$$|A_{i,1}| = \max\{|A_{j,i}| : j \in \{1, \dots, n\}\}$$

Tale processo prende il nome di *processo pivoting* e l'elemento preso prende il nome di *pivot*

Se lavoriamo solo sulla riga/colonna presa in considerazione, abbiamo un processo che prende il nome di *pivoting parziale*, mentre se facciamo una permutazione che va ad interessare anche le righe e le colonne successive si parla di *pivoting totale o completo*. Chiaramente abbiamo che è migliore fare il *pivoting totale*, tuttavia è molto più costoso in quantità di tempo e di ricerca, quindi è più consigliato quello parziale

Esempio di Fattorizzazioni Diverse in base alla matrice di Permutazione

Riprendiamo la matrice

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

Se permuto le righe della matrice ottengo:

$$\Pi A = \begin{pmatrix} 1 & 1 \\ 10^{-4} & 1 \end{pmatrix} \quad \Rightarrow \quad U = \begin{pmatrix} 1 & 1 \\ 0 & 10^{-4} \end{pmatrix}$$

Da cui si ottiene che

$$\kappa(A) \approx \kappa(U) \approx 4$$

Analizziamo un Caso Particolare - A simmetrica e definita positiva

Prima di andare avanti diamo una definizione

Definizione Definita Positiva

Sia $A \in \mathbb{R}^{n \times n}$ simmetrica. Si dice che A è definita positiva se

$$\forall x \in \mathbb{R}^n, x \neq 0 \quad x^T A x > 0$$

In particolare $A_{i,i} > 0$ per ogni $i \in \{1, \dots, n\}$ e i valori autovalori sono maggiori di 0

Lemma

Sia $A \in \mathbb{R}^{n \times n}$ simmetrica. A è definita positiva se e solo se i determinanti delle sottomatrici principali dominanti (di testa) sono positivi

Questo lemma ci sarà fondamentale per la dimostrazione della fattorizzazione di Cholesky

Fattorizzazione di Cholesky

Sia $A \in \mathbb{R}^{n \times n}$ una matrice simmetrica definita positiva, allora esiste un'unica fattorizzazione $A = \hat{L}\hat{L}^T$, con \hat{L} triangolare inferiore e con $\hat{L}_{i,i} > 0$ per $i \in \{1, \dots, n\}$

Dimostrazione:

Grazie al lemma precedente, sappiamo che i determinanti dei minori principali sono tutti positivi, quindi per il teorema precedente esiste un'unica fattorizzazione $A = LU$ con L matrice triangolare inferiore con tutti 1 sulla diagonale principale e con U triangolare superiore non singolare.

Poniamo $D \in \mathbb{R}^{n \times n}$ come:

$$D = \text{diag}(U) = \begin{pmatrix} U_{1,1} & & & \\ & \ddots & & \\ & & U_{n,n} & \end{pmatrix}$$

Abbiamo anche che non è nulla perché U è non singolare, quindi $U_{i,i} > 0, \forall i \in \{1, \dots, n\}$

Allora, otteniamo che:

$$A = LU = LDD^{-1}U = LDR \quad \text{con } R = D^{-1}U$$

Notiamo che R è una matrice triangolare superiore, in quanto prodotto di una matrice diagonale D^{-1} con una triangolare superiore U . Inoltre R ha tutti 1 sulla diagonale, in quanto:

$$R = D^{-1}U = \begin{pmatrix} U_{1,1}^{-1} & & & \\ & U_{2,2}^{-1} & & \\ & & \ddots & \\ & & & U_{n,n}^{-1} \end{pmatrix} \begin{pmatrix} U_{1,1} & U_{1,2} & \cdots & U_{1,n} \\ U_{2,2} & \cdots & U_{2,n} & \\ \vdots & & \ddots & \\ & & & U_{n,n} \end{pmatrix} = \begin{pmatrix} 1 & \frac{U_{1,2}}{U_{1,1}} & \cdots & \frac{U_{1,n}}{U_{1,1}} \\ 1 & \cdots & \frac{U_{2,n}}{U_{2,2}} & \\ \vdots & & \ddots & \\ & & & 1 \end{pmatrix}$$

Abbiamo inoltre, poiché A è simmetrica, che $A = A^T$, di conseguenza segue che:

$$A = A^T = (LDR)^T = R^T D L^T$$

Notiamo che, visto che R è triangolare superiore, R^T è triangolare inferiore, mentre, visto che L è una matrice triangolare, L^T è una matrice triangolare superiore. Quindi abbiamo ancora ottenuto una fattorizzazione LU

Tuttavia, per il teorema, segue che è unica, quindi segue necessariamente che $R^T = L$

Segue quindi che $A = LDL^T$.

Notiamo quindi che D è definita positiva, ossia $D_{i,i} > 0$, inatti:

- 1) la segnatura di A è la stessa di D , in quanto sono congruenti (tramite la matrice L)
- 2) Sia $x \in \mathbb{R}^n, x \neq 0$ tale che $x^T A x > 0$, allora segue che:

$$x^T A x > 0 \Rightarrow x^T (L D L^T) x > 0 \xrightarrow{y = L^T x} y^T D y > 0$$

Inoltre, poiché $y \neq 0$ in quanto L è non singolare ($y \neq 0 \Leftrightarrow L^T x \neq 0$)

Quindi ha senso scrivere $D = D^{1/2} \cdot D^{1/2}$, dove $D^{1/2}$ è la matrice avente sulla diagonale le radici quadrate degli elementi sulla diagonale principale di U , (o le radici quadrate degli elementi di D , è uguale)

Otteniamo quindi che:

$$A = L D L^T = L D^{1/2} \cdot D^{1/2} L^T \xrightleftharpoons{\hat{L}=LD^{1/2}} \hat{L} \hat{L}^T$$

Quindi abbiamo trovato una matrice che rispecchia la condizione enunciata nel teorema.

Da dove viene l'unicità?

L'unicità segue proprio dal fatto che $\hat{L}_{i,i} > 0 \Leftrightarrow \sqrt{L_{i,i}} > 0$

Infatti, volendo potevo prendere anche una matrice

$$\mathcal{J} = \begin{pmatrix} \pm 1 & & \\ & \ddots & \\ & & \pm 1 \end{pmatrix} \quad \text{tale che} \quad \hat{L} \hat{L}^T = \hat{L} \mathcal{J} \mathcal{J}^T \hat{L}^T$$

Quindi avrei un'altra matrice tale che svolge il ruolo richiesto dal teorema, però non è verificata la condizione che $\hat{L}_{i,i} > 0$

□

Considerazione del Teorema: La fattorizzazione è molto conveniente per quanto riguarda il costo computazionale, infatti vale $\frac{1}{3}n^3 + \Theta(n^2)$ rispetto alla fattorizzazione LU che era $\frac{2}{3}n^3 + \Theta(n^2)$.

Un'altra considerazione da fare è che questo teorema in realtà è un se e solo se. Infatti è anche vero che se esiste una matrice $M \in \mathbb{R}^{n \times n}$ tale che $A = MM^T$, allora A è definita positiva. Infatti, possiamo notare subito che $A = A^T$, infatti

$$A^T = (MM^T)^T = (M^T)^T M^T = MM^T = A$$

Prendiamo un $x \in \mathbb{R}^n$ non nullo, allora

$$x^T A x = x^T MM^T x \xrightarrow{y=M^T x} y^T y = \|y\|^2$$

$E \|y\|^2 \geq 0$, inoltre è uguale a 0 se e solo se $y = 0$ ovvero se e solo se $x = 0$

Rifacciamo la dimostrazione con un Approccio Algoritmico:

Proseguiamo per induzione.

Base Induttiva: Sia $k = 1$, allora banalmente si ha che $A_{1,1} > 0$ e abbiamo che $\hat{L}_{1,1} = \sqrt{A_{1,1}}$

Passo Induttivo: Supponiamo che sia vero per k e mostriamo che sia vero per $k+1$

$$A_{k+1} = \begin{pmatrix} A_{1,1} & v^T \\ v & A^{(2)} \end{pmatrix} \xrightarrow{?} \underbrace{\begin{pmatrix} \sqrt{A_{1,1}} & 0^T \\ \frac{1}{\sqrt{A_{1,1}}}v & I \end{pmatrix}}_L \underbrace{\begin{pmatrix} 1 & 0^T \\ 0 & \tilde{A}^{(2)} \end{pmatrix}}_{L^T} \underbrace{\begin{pmatrix} \sqrt{A_{1,1}} & \frac{1}{\sqrt{A_{1,1}}}v^T \\ 0 & I \end{pmatrix}}_{L^T} \quad 0, v \in \mathbb{R}^k, \quad A^{(2)} \in \mathbb{R}^{k \times k}, \quad A_{1,1} \in \mathbb{R}$$

È vera l'uguaglianza? Facciamo i prodotti:

$$\begin{pmatrix} \sqrt{A_{1,1}} & 0^T \\ \frac{v}{\sqrt{A_{1,1}}} & \tilde{A}^{(2)} \end{pmatrix} \begin{pmatrix} \sqrt{A_{1,1}} & \frac{1}{\sqrt{A_{1,1}}}v^T \\ 0 & I \end{pmatrix} = \begin{pmatrix} A_{1,1} & v^T \\ v & \frac{vv^T}{A_{1,1}} + \tilde{A}^{(2)} \end{pmatrix}$$

Se poniamo $\tilde{A}^{(2)} = A^{(2)} - \frac{vv^T}{A_{1,1}}$, allora è verificata, in quanto se A_{k+1} è definita positiva, allora anche $\tilde{A}^{(2)}$ è definita positiva, in quanto hanno la stessa segnatura.

Osservazione: $\tilde{A}^{(2)} \succ 0$, allora può essere scomposta per la scomposizione di Cholesky

$$\tilde{A}^{(2)} = \tilde{L}_k \cdot \tilde{L}_k^T \quad \Leftrightarrow \quad A_{k+1} = \begin{pmatrix} \sqrt{A_{1,1}} & 0^T \\ \frac{v}{\sqrt{A_{1,1}}} & \tilde{L}_k \end{pmatrix} \begin{pmatrix} \sqrt{A_{1,1}} & \frac{v^T}{\sqrt{A_{1,1}}} \\ 0 & \tilde{L}_k^T \end{pmatrix} = \tilde{L}_{k+1} \hat{L}_{k+1}^T$$

Lo stesso posso poi continuare con $\tilde{A}^{(2)}$

Vediamo l'Algoritmo di Cholesky scritto in Pseudocodice:

Per $j = 1, \dots, n$

$$\hat{L}_{j,j} = \left(A_{j,j} - \sum_{k=1}^{j-1} \hat{L}_{j,k}^2 \right)^{1/2}$$

Per $i = j + 1, \dots, n$

$$\hat{L}_{i,j} = \left(A_{i,j} - \sum_{k=1}^{j-1} \hat{L}_{i,k}^2 \right) / (\hat{L}_{j,j})$$

In Matlab esiste direttamente un comando che permette la fattorizzazione di Cholesky:

- `R = chol(A)`, da cui abbiamo che $R = \hat{L}^T$
- `R = chol(A, 'lower')` da cui otteniamo che $R = \hat{L}$

Osservazione: L'algoritmo richiede radici quadrate, quindi se c'è un numero negativo c'è un problema (non precisamente in matlab, in quanto la radice viene automaticamente convertita in numero complesso, però generalmente può creare problemi). In generale, l'algoritmo termina con successo se $A \succ 0$, in aritmetica e precisione finita, in particolare, l'algoritmo di Cholesky può essere usato per verificare se A è definita positiva

Ha un costo computazionale pari a $\frac{1}{3}n^3 + \Theta(n^2)$

Osservazione: Nella precedente dimostrazione abbiamo scritto che $A = LDL^T$. In questo contesto abbiamo utilizzato pesantemente il fatto che A era definita positiva ($A \succ 0$), questo perché abbiamo potuto, con certezza, calcolare le radici quadrate degli elementi sulla diagonale principale, per poi scrivere $A = \hat{L}\hat{L}^T$. Se A invece fosse stata simmetrica non definita, allora il massimo a cui ci si poteva ispirare era $A = LDL^T$

Tutto questo tenendo conto del fatto che non stiamo applicando pivoting

Inoltre, se A non è definita, allora anche D diagonale non lo è e sulla diagonale ci sono dei valori ≤ 0 .

Ci sono poi delle fattorizzazioni chiamate di "Tipo Cholesky" per matrici simmetriche ma non definite positive tali che

$$\Pi A \Pi^T = LDL^T$$

dove D in generale è una matrice diagonale a blocchi simmetrici con blocchi 1×1 o 2×2 non necessariamente definiti positivi.

Analisi dell'Errore

Teorema

Sia $A \in \mathbb{R}^{n \times n}$ e siano $\tilde{L} = L + \delta L$ e $\tilde{U} = U + \delta U$ le matrici effettivamente calcolate nella fattorizzazione LU di A . Allora:

$$\tilde{L}\tilde{U} = A + E \quad \text{con} \quad \|E\|_F \leq 2nu(\|A\|_F + \|\tilde{L}\|_F + \|\tilde{U}\|_F) + \Theta(u^2n^2)$$

Considerazioni del Teorema: $\tilde{L}\tilde{U}$ assomiglia ad A se $\|E\|_F$ è piccola e dipende dal valore enunciato nel teorema, con u il valore epsilon della macchina e n la dimensione dei dati. Notiamo che nella scrittura di tale valore c'è sempre $\|A\|_F$ perché serve per vedere la vicinanza da A

Teorema

Siano \tilde{L}, \tilde{U} i fattori come precedentemente annunciati nel teorema precedente e sia \tilde{x} la soluzione ottenuta dalla risoluzione di

$$\tilde{L}\tilde{U}\tilde{x} = b$$

Allora \tilde{x} è anche soluzione di

$$(A + \delta A)\tilde{x} = b \quad \text{con} \quad \|\delta A\|_F \leq 4nu(\|A\|_F + \|\tilde{L}\|_F\|\tilde{U}\|_F) + \Theta(u^2n^2)$$

Considerazioni del Teorema: Qui abbiamo che la nostra soluzione è soluzione di un problema perturbato la cui norma è controllata dal valore nell'enunciato del teorema. Quindi \tilde{x} è soluzione del sistema tanto più vicino quanto $\frac{\|\delta A\|_F}{\|A\|_F}$ è piccolo

Dimostrazione:

Abbiamo per ipotesi che \tilde{x} risolve il sistema

$$\tilde{L}\tilde{U}\tilde{x} = b$$

Per cui posto $y = \tilde{U}\tilde{x}$ otteniamo che dobbiamo risolvere i sistemi

$$\tilde{L}y = b \quad \text{e} \quad \tilde{U}\tilde{x} = y$$

In particolare, in precisione finita, che:

$$\begin{cases} \text{Otteniamo } \tilde{y} : (\tilde{L} + E_L)\tilde{y} = b \quad \text{con} \quad \|E_L\| \leq mu\|\tilde{L}\|_F + \Theta(n^2u^2) \\ \text{Otteniamo } \tilde{x} : (\tilde{U} + E_U)\tilde{x} = \tilde{y} \quad \text{con} \quad \|E_U\| \leq mu\|\tilde{U}\|_F + \Theta(n^2u^2) \end{cases}$$

In particolare, combinando tutto insieme otteniamo che:

$$(\tilde{L} + E_L)(\tilde{U} + E_U)\tilde{x} = b \Rightarrow (\underbrace{\tilde{L}\tilde{U}}_{A+E} + \underbrace{\tilde{L}E_U + E_L\tilde{U} + E_L E_U}_{\delta A})\tilde{x} = b \Rightarrow (A + E + \delta A)\tilde{x} = b$$

Da cui, utilizzando le proprietà delle norme, segue che

$$\|\delta A\|_F \leq \|E\|_F + \|\tilde{L}\|_F\|E_U\|_F + \|E_L\|_F\|\tilde{U}\|_F + \|E_L\|_F\|E_U\|_F$$

Ma abbiamo che, per il teorema precedente:

$$\left. \begin{array}{l} \|E\|_F \leq 2nu(\|A\|_F + \|\tilde{L}\|_F\|\tilde{U}\|_F) + \Theta(u^2n^2) \\ \|E_U\|_F \leq 2nu(\|\tilde{U}\|_F) + \Theta(u^2n^2) \\ \|E_L\|_F \leq 2nu(\|\tilde{L}\|_F) + \Theta(u^2n^2) \\ \|E_l\|_F\|E_U\|_F \leq \Theta(u^2n^2) \end{array} \right\} \|\delta A\|_F \leq 4nu(\|A\|_F + \|\tilde{L}\|_F\|\tilde{U}\|_F) + \Theta(u^2n^2)$$

□

Rimane ora da fare una simila per $\|\tilde{U}\|_F$ e $\|\tilde{L}\|_F$:

Consideriamo il caso di pivoting parziale, in cui mettiamo ad ogni iterazione dell'eliminazione di Gauss cerchiamo il coefficiente più grande e lo spostiamo sulla diagonale. Sapendo però che sulla diagonale principale di \tilde{L} ci sono solo 1, abbiamo che:

$$\forall i, j \in \{1, \dots, b\} \quad |\tilde{L}_{i,j}| \leq 1$$

Segue quindi che

$$\|\tilde{L}\|_F = \sqrt{\sum_{i,j=1}^n \tilde{L}_{i,j}^2} \leq \sqrt{\sum_{i,j=1}^n 1} = \sqrt{n^2} = n$$

Ci resta ora da maggiorare $\|\tilde{U}\|_F$.

In particolare ci resta da maggiorare il tasso di crescita $\frac{\|\tilde{U}\|_F}{\|A\|_F}$

Definizione di $\|A\|_{max}$

Data una matrice A si definisce norma massima di A la norma:

$$\|A\|_{max} = \max_{i,j} |A_{i,j}|$$

Definizione di Fattore di Crescita in Pivoting Parziale

Sia A una matrice e sia \tilde{U} una matrice triangolare superiore ottenuta dall'eliminazione di Gauss con pivoting parziale. Si definisce Fattore di Crescita in Pivoting Parziale o $g_{p,p}$ (*Growth factor in partial pivoting*) la quantità

$$g_{p,p} = \frac{\|\tilde{U}\|_F}{\|A\|_F}$$

Sapendo che $\|A\|_F \leq \|A\|_{max}$, abbiamo che:

$$\frac{\|\tilde{U}\|_F}{\|A\|_F} \leq \frac{\|\tilde{U}\|_F}{\|A\|_{max}} = \frac{\sqrt{\sum_{i,j}^n \tilde{U}_{i,j}^2}}{\|A\|_{max}} \leq \frac{\|\tilde{U}\|_{max} \sqrt{\sum_{i,j}^n (\tilde{U}_{i,j}^2) / (\|\tilde{U}\|_{max}^2)}}{\|A\|_{max}}$$

Tuttavia, sapendo che:

$$\forall i, j \in \{1, \dots, n\} \quad \frac{\tilde{U}_{i,j}^2}{\|U\|_{max}} \leq 1$$

Otteniamo che:

$$\frac{\|\tilde{U}\|_F}{\|A\|_F} \leq \frac{\|\tilde{U}\|_{max} \sqrt{\sum_{i,j}^n (\tilde{U}_{i,j}^2) / (\|\tilde{U}\|_{max}^2)}}{\|A\|_{max}} \leq n \frac{\|\tilde{U}\|_{max}}{\|A\|_{max}} = n g_{p,p}$$

Teorema

Il fattore di crescita per il metodo di eliminazione di Gauss con pivot parziale verifica

$$g_{p,p} \leq 2^{n-1}$$

Dimostrazione:

Riprendiamo i passi della dimostrazione di Gauss

Sapendo che tutti i passaggi valgono per ogni elemento $A_{i,j}^{(2)}$, sicuramente varranno anche per $\|A^{(2)}\|_{max}$, quindi partendo da:

$$A_{i,j}^{(2)} = A_{i,j}^{(1)} - m_{i,1} A_{1,j}^{(1)} \Rightarrow |A_{i,j}^{(2)}| \leq |A_{i,j}^{(1)}| + 1 |A_{i,j}^{(1)}| \leq 2 \|A^{(1)}\|_{max}$$

Continuando con il ragionamento seguente si ottiene che:

$$|A_{i,j}^{(3)}| \leq |A_{i,j}^{(2)}| - m_{i,2} A_{2,j}^{(2)} \leq |A_{i,j}^{(2)}| + |A_{i,j}^{(2)}| \leq 2 \|A^{(2)}\|_{max} \leq 4 \|A^{(1)}\|_{max}$$

Proseguendo con tutte le iterazioni si ottiene che:

$$\|A^{(n)}\|_{max} \leq 2^{n-1} \|A^{(1)}\|_{max} = 2^{n-1} \|A\|_{max}$$

Tuttavia, sapendo che $\tilde{U} = A^{(n)}$ otteniamo che:

$$\|\tilde{U}\|_{max} \leq 2^{n-1} \|A\|_{max} \Rightarrow \frac{\|\tilde{U}\|_{max}}{\|A\|_{max}} = g_{p,p} \leq 2^{n-1}$$

□

Quindi abbiamo che $\|\tilde{U}\|_F \leq n g_{p,p} \|A\|_F \leq n 2^{n-1} \|A\|_F$. Quindi, per i teoremi precedenti, otteniamo che:

$$\|\delta A\|_F \leq 4nu(\|A\|_F + n \cdot n \cdot 2^{n-1} \|A\|_F) + \Theta(n^2 u^2) \approx 4n^3 2^{n-1} u \|A\|_F$$

Se siamo sfortunati abbiamo che la fattorizzazione LU fornisce una \bar{x} che è soluzione di un sistema molto lontano da quello originale.

Nel 99% dei casi va bene, però bisogna anche analizzare il caso in cui vada male

Esempio della Matrice di Moler

Consideriamo la matrice di Moler definita come

$$A = \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 & 1 \\ -1 & 1 & 0 & \cdots & 0 & 1 \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ -1 & \cdots & -1 & 1 & 0 & 1 \\ -1 & \cdots & \cdots & -1 & 1 & 1 \\ -1 & \cdots & \cdots & \cdots & -1 & 1 \end{pmatrix}$$

In questa matrice, la cui eliminazione di Gauss non fa pivoting, abbiamo che $U_{n,n} = 2^{n-1}$ mentre

$$\frac{\|U\|_{max}}{\|A\|_{max}} = 2^{n-1}$$

In quanto $\|A\|_{max} = 1$

Caso 2.5 - Matrici a Banda

Definizione di Matrice a Banda

Sia $A \in \mathbb{R}^{n \times n}$. Si dice che A è una matrice a banda con banda inferiore inferiore b_L e banda superiore b_U una matrice che ha 0 nelle posizioni i, j tali che $i > j + b_L$ oppure $j > i + b_U$

Esempio: Una matrice $n \times n$ a banda con $b_U = 1$ e $b_L = 2$ è una matrice della forma:

$$A = \begin{pmatrix} \circ & \times & 0 & \cdots & \cdots & 0 \\ \times & \circ & \times & \ddots & & \vdots \\ \times & \ddots & \circ & \times & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \times \\ 0 & \cdots & 0 & \times & \times & \circ \end{pmatrix}$$

Definizione di Matrici Tridiagonale

Sia $A \in \mathbb{R}^{n \times n}$ una matrice a bande. Si dice che A è tridiagonale se $b_L = b_U = 1$

Definizione di Matrice Hessenberg

Si definisce Matrice di Hessenberg una matrice $A \in \mathbb{R}^{n \times n}$ a banda con $b_L = 1$ e $b_U = n - 1$

Se la matrice è a banda, allora abbiamo un sacco di zeri messi nella matrice, quindi possiamo usare il comando `sparse` in matlab per ottimizzare lo spazio utilizzato per la matrice.

Comunque sia abbiamo questo risultato.

Proposizione

Sia $A \in \mathbb{R}^{n \times n}$ una matrice a banda con banda inferiore b_L e banda superiore b_U .

Se esiste la fattorizzazione LU senza pivoting, cioè $A = LU$ con $L = (\Delta)$ con tutti 1 sulla diagonale principale e $U = (\nabla)$, allora L ha banda b_L e U ha banda b_U

Inoltre, il costo computazionale dell'eliminazione di Gauss è pari a $\Theta(b_L b_U n)$

Tutto questo per far vedere che è fondamentale sfruttare la struttura della matrice

Osservazione: Se noi abbiamo una matrice a bande con $b_L = b_U = 2$, allora per costruire il vettore m_i possiamo fermarci prima:

$$m_{i,1} = \frac{A_{i,1}^{(1)}}{A_{1,1}^{(1)}} \quad i \in \{2, \dots, 1 + b_L\}$$

Infatti non c'è assolutamente bisogno di dover continuare e registrare tutti zero, occupa solo memoria.

In maniera analoga anche per

$$A_{i,j}^{(2)} = A_{i,j}^{(1)} - m_{i,1} A_{i,j}^{(1)} \quad i \in \{2, \dots, n+1\} \quad j \in \{2, \dots, i + b_U - 1\}$$

Algoritmo di Thomas per il caso Tridiagonale

Ci sofferriamo su questo caso particolare perché le matrici tridiagonali le vedremo spesso per i polinomi e per la risoluzione di sistemi lineari, in particolare per le equazioni del moto.

Consideriamo quindi la seguente matrice tridiagonale:

$$A = \begin{pmatrix} a_1 & c_1 & 0 & \cdots & \cdots & 0 \\ b_2 & a_2 & c_2 & \ddots & & \vdots \\ 0 & b_3 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & c_{n-1} \\ 0 & \cdots & \cdots & 0 & b_n & a_n \end{pmatrix}$$

Vogliamo risolvere il sistema lineare $Ax = f$

Supponiamo esistano $L, U \in \mathbb{R}^{n \times n}$ tali che $A = LU$ come nell'enunciato della proposizione precedente, ossia:

$$L = \begin{pmatrix} 1 & & & & \\ \beta_2 & 1 & & & \\ & \ddots & \ddots & & \\ & & \beta_n & 1 & \end{pmatrix} \quad U = \begin{pmatrix} \alpha_1 & \gamma_1 & & & \\ & \alpha_2 & \ddots & & \\ & & \ddots & & \gamma_{n-1} \\ & & & \alpha_n & \end{pmatrix}$$

Vogliamo trovare i vari $\alpha_i, \beta_i, \gamma_i$

Iniziamo a moltiplicare $L_{:,i}$ con $U_{:,j}$ per ottenere $A_{i,j}$ attraverso la moltiplicazione riga per colonna, otteniamo che:

$$\begin{aligned} i = 1, j = 1 : \alpha_1 &= a_1 & i = 1, j = 2 : \gamma_1 &= c_1 \\ i = 2, j = 1 : \beta_2 \cdot \alpha_1 &= b_2 & i = 2, j = 2 : \beta_2 \cdot \gamma_1 + \alpha_2 &= a_2 & i = 2, j = 3 : \gamma_2 &= c_2 \end{aligned}$$

In maniera del tutto analoga per le altre colonne.

Andando avanti con i calcoli otteniamo che:

$$\boxed{\forall i \in \{1, \dots, n-1\} \ \forall j \in \{2, \dots, n\} \quad \alpha_1 = a_1, \ \gamma_i = c_i, \ \beta_j = \frac{b_j}{\alpha_{j-1}}, \ \alpha_j = a_j - \beta_j c_{j-1}}$$

Facciamo il calcolo computazionale: per il calcolo dei β_i ci si impiegano $n-1$ flops, mentre per il calcolo dei α_i ci vogliono $2(n-1)$ flops, da cui segue effettivamente che il costo computazionale è dell'ordine di $\Theta(3n-3)$

Tuttavia abbiamo ancora da risolvere il sistema lineare in sé e nel fare ciò possiamo fare il solito metodo di scomporre il sistema lineare in due sistemi diversi, ossia:

$$\underbrace{LUx}_y = f \quad \Leftrightarrow \quad Ly = f \quad Ux = y$$

Facendo i calcoli esplicativi segue che:

$$Ly = f \quad \Leftrightarrow \quad \begin{pmatrix} 1 & & & & \\ \beta_2 & 1 & & & \\ & \ddots & \ddots & & \\ & & \beta_n & 1 & \end{pmatrix} \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} f_1 \\ \vdots \\ f_n \end{pmatrix} \quad \Rightarrow \quad \begin{cases} y_1 = f_1 \\ \beta_i y_{i-1} + y_i = f_i \quad i \neq 1 \end{cases} \Rightarrow \quad \boxed{\begin{cases} y_1 = f_1 & i = 1 \\ y_i = f_i + \beta_i y_{i-1} & i \neq 1 \end{cases}}$$

$$Ux = y \Leftrightarrow \begin{pmatrix} \alpha_1 & \gamma_1 & & \\ & \alpha_2 & \ddots & \\ & & \ddots & \gamma_{n-1} \\ & & & \alpha_n \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \Rightarrow \begin{cases} x_n = \frac{y_n}{\alpha_n} & i = n \\ \alpha_i x_i + c_i x_{i+1} = y_i & i \neq n \end{cases} \Rightarrow \begin{cases} x_n = \frac{y_n}{\alpha_n} & i = n \\ x_i = \frac{y_i - c_i x_{i+1}}{\alpha_i} & i \neq n \end{cases}$$

Mettiamo adesso tutto insieme per vedere il costo computazionale dell'algoritmo.

Se prendiamo tutto quello che è stato quadrettato, abbiamo $3(n-1)$ flops per la fattorizzazione LU della matrice, $2(n-1)$ flops per la risoluzione del primo sistema lineare e $3(n-1)$ per la risoluzione del secondo, arrivando ad avere un costo computazionale pari a

$$8n + \Theta(1)$$

Notevole considerando il fatto che una normalissima scomposizione avrebbe un costo pari a $\Theta(n^3)$

Appendice - Formula di Sherman - Morrison

Questa formula risulta estremamente comoda per il calcolo dell'inversa di una matrice modificata da una'altra matrice di rango 1

Formula di Sherman - Morrison

Sia $A \in \mathbb{R}^{n \times n}$ invertibile e siano $u, v \in \mathbb{R}^n$ tale che $A + uv^T$ sia ancora invertibile, allora vale l'uguaglianza:

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

In particolare possiamo notare che il membro di destra è sempre una matrice, infatti, usando la struttura otteniamo che:

$$((\square) + (\square))^{-1} = (\square)^{-1} - (\square)^{-1} \cdot (|) \cdot (1 + (-)(\square)(|))^{-1} \cdot (-) \cdot (\square)^{-1}$$

Chiaramente quest'uguaglianza vale se $a + v^T A^{-1}u \neq 0$

Con questa formula stiamo dicendo che l'inversa di una matrice di una modifica di rango 1 è ancora una modifica di rango 1
A cosa ci potrebbe servire?

Supponiamo di avere A diagonale e avere dei vettori $u, v \in \mathbb{R}^n$ e consideriamo

$$A + uv^T = (|) + (-)(|) = (|) + (\square) = (\square)$$

Conoscendo la formula possiamo minimizzare i conti per calcolare l'inversa ed in particolare per minimizzare il sistema lineare:

$$(A + uv^T)x = b$$

In particolare, andando a risolverlo avremmo che:

$$x = (A + uv^T)^{-1}b \xrightarrow{SM} A^{-1}b - A^{-1}u(1 - v^T A^{-1}u)^{-1}v^T A^{-1}b$$

Andiamo ora a fare tutta una serie di sostituzioni, in particolare poniamo

$$(1) : Aw_1 = b \Rightarrow A^{-1}b = w_1 \quad \text{e} \quad (2) : Aw_2 = u \Rightarrow w_2 = A^{-1}u$$

Andiamo poi a sostituire

$$(3) : \gamma_1 = v^T w_1 \quad (4) : \gamma_2 = v^T w_2 \quad (5) : \gamma_3 = \frac{1}{1 + \gamma_2}$$

Facendo queste sostituzioni otteniamo che:

$$\begin{aligned} x &= A^{-1}b - A^{-1}u(1 - v^T A^{-1}u)^{-1}v^T A^{-1}b \xrightarrow{(1)} w_1 - A^{-1}u(1 - v^T A^{-1}u)^{-1}v^T w_1 \xrightarrow{(2)} w_1 - w_2(1 - v^T w_2)^{-1}v^T w_1 \xrightarrow{(3)} \\ &= w_1 - w_2(1 - v^T w_2)^{-1} \cdot \gamma_1 \xrightarrow{(4)} w_1 - w_2(1 - \gamma_2)^{-1} \cdot \gamma_1 \xrightarrow{(5)} w_1 - w_2(\gamma_3 \cdot \gamma_1) \end{aligned}$$

E adesso si può risolvere facilmente nei metodi usuali:

Il suo costo computazionale è pari a: costo di due risoluzioni di sistemi lineari + $2(n-1)$ flops per γ_1 + $2(n-1)$ per γ_2 + 3

per γ_3 , quindi il costo computazionale è pari a 2 soluzioni di $A + 6n + \Theta(1)$

Chiaramente tutto questo ha senso se la struttura di A è ottimale

Osservazione Importante: $w_2(\gamma_3 \cdot \gamma_1)$ e $(w_2 \cdot \gamma_3)\gamma_1$ non hanno lo stesso costo computazionale, infatti il primo costa $n+1$ flops, mentre il secondo costa $2n$ flops. La stessa cosa vale anche per uv^Tb . Infatti il costo di $(uv^T)b$ è molto molto maggiore di $u(v^Tb)$, perché mentre nel primo caso bisogna prima costruire una matrice e poi fare un prodotto tra una matrice e un vettore (uv^T è una matrice), nel secondo basta risolvere un prodotto scalare e moltiplicare un vettore per uno scalare.

TUTTO QUESTO SARÀ FONDAMENTALE PER LA PROVA DI LABORATORIO

Primo Esempio di Utilizzao della Formula S.M.

Vogliamo risolvere il sistema lineare $\hat{A}x = b$ con A matrice di questa forma:

$$\hat{A} = \begin{pmatrix} \times & 0 & \cdots & \cdots & 0 & \times \\ \times & \times & \ddots & & \vdots & \vdots \\ 0 & \times & \times & \ddots & \vdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 & \times \\ \vdots & & \ddots & \times & \times & \times \\ 0 & \cdots & \cdots & 0 & \times & \times \end{pmatrix}$$

Cioè A è bidiagonale con l'ultima colonna non nulla.

Possiamo trovare dei vettori $u, v \in \mathbb{R}^n$ tali che $\hat{A} = A + uv^T$?

Visto che sappiamo facilmente risolvere un sistema lineare con una matrice a banda, matrice che chiameremo A , ci servono dei vettori che creino l'ultima colonna. Per tale scelta possiamo prendere:

$$u = \begin{pmatrix} U_{1,n} \\ \vdots \\ U_{n-1,n} \\ 0 \end{pmatrix} \quad \text{e} \quad v = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} = e_n$$

Abbiamo quindi che

$$\hat{A} = (\backslash\backslash) + \hat{A}_{j,n}e_n^T$$

Quindi il nostro sistema lineare diventerà:

$$\hat{A}x = b \quad \Leftrightarrow \quad (A + ue_n^T)x = b$$

Osservazione La formula di Sherman - Morrison può essere estesa anche al caso di più vettori, ossia date $A \in \mathbb{R}^{n \times n}$ e $U, V \in \mathbb{R}^{n \times s}$ con s piccola, allora, se la somma è invertibile si ha che

$$(A + UV^T)^{-1} = A^{-1} - A^{-1}U(I + V^TA^{-1}U)V^TA^{-1}$$

Secondo Esempio di Utilizzao della Formula S.M.

Consideriamo la matrice

$$\hat{A} = \begin{pmatrix} \times & & \times \\ & \ddots & \vdots \\ \times & \cdots & \times \end{pmatrix}$$

Esistono delle matrici U, V tali che $\hat{A} = A + UV^T$ con A diagonale?

Possiamo applicare la formula prima in due passi, ossia:

$$\begin{pmatrix} \times & & \times \\ & \ddots & \vdots \\ \times & \cdots & \times \end{pmatrix} = \begin{pmatrix} \times & & \times \\ & \ddots & \\ \times & \cdots & \times \end{pmatrix} + \begin{pmatrix} 0 & \times \\ & \vdots \\ & \times \end{pmatrix} = \begin{pmatrix} \times & & \times \\ & \ddots & \\ \times & \cdots & \times \end{pmatrix} + \begin{pmatrix} 0 & & \\ & \cdots & \\ & & \times \end{pmatrix} + \begin{pmatrix} 0 & \times \\ & \vdots \\ & \times \end{pmatrix}$$

In particolare tutto questo è uguale a:

$$\hat{A} = A_{diag} + e_n v^T + u e_n^T = A_{diag} + (e_n \quad u) \begin{pmatrix} v^T \\ e_n^T \end{pmatrix} = A_{diag} + U V^T$$

Dove U, V sono delle matrici in $\mathbb{R}^{n \times 2}$

Metodi Iterativi per Sistemi Lineari

Si utilizzano questi metodi quando non è possibile calcolare la soluzione in forma chiusa. In questo caso parliamo di successioni di valori che tendono ad essere la soluzione del sistema lineare. Ci troviamo quindi nel caso in cui abbiamo un sistema lineare:

$$Ax = b$$

di cui però non possiamo trovare la soluzione direttamente

Sono tutti metodi che possono tornare comodi comodi per matrici dalle dimensioni di $10^6 \times 10^6$

In generale i sistemi iterativi si segue una stessa procedura.

Fissiamo un vettore $x_0 \in \mathbb{R}^n$, determiniamo poi una successione $(x_k)_{k \geq 0}$ tale che sotto particolari condizioni soddisfi:

$$x_k \xrightarrow{k \rightarrow +\infty} x^*$$

dove x^* è la soluzione del sistema lineare $Ax = b$

Prima di poter presentare i metodi iterativi per sistemi lineari, diamo la definizione di convergenza.

Definizione di Convergenza per Metodi Iterativi

Un metodo si dice convergente se $\forall x_0 \in \mathbb{R}^n$ so ha che l'errore al passo k -esimo, definito come

$$e_k = x_k - x^*$$

segue la seguente relazione:

$$\|e_k\| \xrightarrow{k \rightarrow +\infty} 0$$

Il metodo chiaramente deve essere valido per ogni vettore di partenza x_0 scelto e per ogni soluzione.

Chiaramente non deve dipendere da dalla scelta di nessuno dei due

Osservazione: In pratica ci accontentiamo del fatto che l'errore in norma sia più piccolo di una tolleranza fissata (che potremo indicare con tol). Inoltre, se dopo k passi, otteniamo che:

$$\|x_k\| - \|x^*\| < tol$$

Allora ci fermiamo. Chiaramente la scelta di tol deve dipendere dal contesto in cui ci troviamo. Purché non sia troppo vicino al valore `eps` della macchina.

Osservazione: Fare un controllo dell'errore è un concetto molto particolare, in quanto non abbiamo nell'effettivo x^* , quindi non è possibile calcolare esplicitamente $x_k - x^*$, appunto perché non abbiamo x^* .

Possiamo però calcolare il vettore residuo r_k al k -esimo passo:

$$r_k = b - Ax_k$$

Facciamo un confronto tra i due vettori che abbiamo trovato e il loro utilizzo:

- e_k serve per vedere quanto siamo vicini dalla soluzione (*l'errore deve scendere a 0 per trovare la soluzione*)
- r_k serve per vedere quanto siamo lontani dalla soluzione (*lo scarto tra il vettore e la soluzione deve scendere a 0*)

La vera grande differenza è che il secondo possiamo calcolarlo, perché abbiamo tutto l'occorrente in quanto abbiamo sia b sia A .

Quindi ci basta creare un algoritmo che finisce quando

$$\|r_k\| < tol$$

Siamo quasi vicini. L'unica cosa che manca è l'utilizzo dei valori relativi:

$$\frac{\|r_k\|}{\|r_0\|} < tol \quad \Leftrightarrow \quad \|r_k\| < tol \cdot \|r_0\|$$

Dove $r_0 = b - Ax_0$

Attenzione: Se otteniamo che $\|r_0\| << 1$ allora possiamo già fermarci, perché avremmo trovato una soluzione molto vicina alla soluzione effettiva.

Confronto tra Residuo ed Errore

Dalle definizioni che abbiamo dato di Residuo e di Errore abbiamo che:

$$r_k = b - Ax_k = Ax^* - Ax_k = A(x^* - x_k) = Ae_k$$

Da cui passando per le norme otteniamo che:

$$\|r_k\| \leq \|A\| \cdot \|e_k\| \Leftrightarrow \frac{\|r_k\|}{\|r_0\|} \leq \frac{\|A\| \cdot \|e_k\|}{\|r_0\|}$$

Dall'altra parte, passando per l'uguaglianza $e_k = A^{-1}r_k$ otteniamo che:

$$\|e_k\| \leq \|A^{-1}\| \cdot \|r_k\| \Leftrightarrow \frac{\|e_k\|}{\|e_0\|} \leq \frac{\|A^{-1}\| \cdot \|r_k\|}{\|e_0\|}$$

Mettendo insieme tutto quello che abbiamo trovato, otteniamo che:

$$\frac{\|e_k\|}{\|e_0\|} \leq \frac{\|A^{-1}\| \cdot \|r_k\|}{\|e_0\|} \leq \|A^{-1}\| \cdot \|A\| \cdot \frac{\|r_k\|}{\|r_0\|} = \kappa(A) \cdot \frac{\|r_k\|}{\|r_0\|}$$

Otteniamo quindi che ad un residuo piccolo potremmo ottenere un errore più o meno piccolo a seconda della grandezza di $\kappa(A)$, cioè a seconda se la matrice è ben condizionata oppure no.

Notiamo che una cosa del genere l'avevamo già incontrata, quando nel capitolo precedente parlavamo di perturbazione dei dati.

Infatti quando abbiamo un sistema perturbato (assunto $\delta A = 0$) otteniamo che:

$$A(x + \delta x) = b + \delta b$$

Quello che abbiamo fatto nelle righe precedenti non era altro che dare un nome a queste perturbazioni.

Infatti x_k non è altro che la soluzione esatta di un sistema più o meno vicino rispetto a quello di partenza (in questo modo possiamo identificare $\delta b = r_k$)

Quindi se la perturbazione è piccola otteniamo che $e_k = x_k - x^* = \delta x$

Quindi otteniamo che $\|r_k\|$ è l'errore all'indietro mentre $\|e_k\|$ è l'errore in avanti.

Il nostro sistema lineare perturbato diventerà quindi:

$$A(x + e_k) = b + r_k$$

Classi di Metodi Iterativi

Esistono due classi di metodi Iterativi:

- Metodi Iterativi Classici o Stazionari (*come quelli di Jacobi e di Gauss-Seidel*)
- Metodi Iterativi Non stazionari (*Come quello dei Gradienti Coniugati*)

Metodi Stazionari

I metodi stazionari funzionano generalmente tutti seguendo uno stesso scherma.

Dato un sistema lineare $Ax = b$ vogliamo dividere (*Splitting*) la matrice A in due parti, in particolare vogliamo trovare due matrici N e P con P non singolare tale che

$$A = P - N$$

Da questo vogliamo arrivare a

$$Ax = b \Rightarrow (P - N)x = b \Rightarrow Px - Nx = b \Rightarrow Px = b + Nx$$

Inoltre, la richiesta della non singolarità di P voene dal fatto che possiamo invertire P per dare un'espressione esplicita della x , ossia:

$$x = P^{-1}Nx + P^{-1}b$$

Possiamo chiamare quest'equazione $\Phi(x)$, equazione che può essere pensata come equazione di punto fisso.

Questo perché la x che risolve tale equazione è esattamente x^* , cioè

$$x^* = \Phi(x^*)$$

e questo punto prende il nome di punto fisso.

Per risolvere equazioni di questo genere, di norma si innescano innesca un'iterazione, dato un vettore $x_0 \in \mathbb{R}^n$ calcoliamo:

$$x_{k+1} = \Phi(x_k) = P^{-1}N x_k + P^{-1}b$$

Ha senso fare tutto questo se P è invertibile, che è vero per come abbiamo posto P , e se Φ è poco costosa

Osservazione: Scrivendo il residuo come $r_k = b - Ax_k$, possiamo scrivere $x_{k+1} = x_k + P^{-1}r_k$

Infatti notiamo che:

$$\begin{aligned} x_{k+1} &= x_k + P^{-1}r_k = x_k + P^{-1}(b - Ax_k) = x_k + P^{-1}b + P^{-1}Ax_k = x_k + P^{-1}b - P^{-1}(P - N)x_k \\ &= x_k + P^{-1}b + P^{-1}Nx_k - x_k = P^{-1}b + P^{-1}Nx_k = \Phi(x) \end{aligned}$$

Notiamo che la scrittura con r_k è preferibile in aritmetica con posizione finita perché evita pericolose cancellazione e evita somme di termini con ordini di grandezza diversi mano a mano che x converge.

Quindi l'algoritmo è:

$x_0 \in \mathbb{R}^n$	ha costo un flop
$r_0 = b - Ax_0$	ha costo un flop
Per $k = 0, 1, \dots$	
$x_{k+1} = x_k + P^{-1}r_k$	ha costo n flops più il costo di un sistema lineare $Pw = r_k$
$r_{k+1} = b - Ax_{k+1}$	ha il costo di un Mxv e di n flops
$\frac{\ r_{k+1}\ }{\ r_0\ } \leq tol$	ha costo $2n$ flops

Nota: Mxv è l'abbreviazione di prodotto matrice per vettore

Il costo computazionale è pari a $(n + n + 2n)$ flops + il costo di $Pw + r_k$ + il costo di Mxv

I metodi si differenzieranno per la scelta di P

Prima di andare a vedere nell'effettivo alcuni metodi iterativi, andiamo prima a vedere qualche teorema sui metodi iterativi.

Per questione di comodità, poniamo $B = P^{-1}N$ come una matrice di iterazione.

Da notare che noi questa matrice non la costruiremo mai in matlab, ci serve soltanto per dare qualche risultato.

Osservazione: Se voglio trovare l'errore al passo $k + 1$ allora:

$$e_{k+1} = x_{k+1} - x^*$$

Tuttavia sappiamo che x^* rappresenta la soluzione di un sistema di punto fisso mentre possiamo ricavare x_{k+1} tramite la stessa equazione, quindi possiamo andare a sostituire e ottenere:

$$e_{k+1} = x_{k+1} - x^* = P^{-1}N x_k + P^{-1}b - (P^{-1}N x^* + P^{-1}b) = P^{-1}N x_k - P^{-1}N x^* = P^{-1}N(x_k - x^*) = B(e_k)$$

L'errore al passo $k + 1$ non è altro che il calcolo di una matrice su uno stesso vettore:

$$e_{k+1} = B(e_k) = B^2(e_k - 1) = \dots = B^{k+1}(e_0)$$

Inoltre, se voglio l'errore tenda a zero, ci serve che la matrice B riduca l'errore di continuo

Diamo prima questa definizione.

Definizione di Raggio Spettrale

Sia $A \in \mathbb{C}^{n \times n}$ una matrice. Si definisce allora il raggio spettrale $\rho(A)$ come:

$$\rho(A) = \max_{\lambda \in \text{Spec}(A)} |\lambda|$$

Teorema

La successione degli $(x)_k$ con $k \geq 0$ converge a x^* per ogni x_0 iniziale se e solo se il raggio spettrale di B è minore di 1, ossia $\rho(B) < 1$

Considerazioni del Teorema: Sapendo che l'errore al passo $k+1$ -esimo si ottiene come prodotto della matrice B per l'errore al passo k -esimo, il teorema ci dice che converge se e solo se $\rho(B) < 1$, ciò implica che la norma dell'autovalore λ più grande possibile deve essere minori di 1, ma in particolare deve essere il più piccolo possibile. Inoltre la matrice non deve essere necessariamente simmetrica, così come gli autovalori non devono essere necessariamente reali. Vogliamo che tutti gli autovalori abbiano norma minore di 1, compreso il più grande. Sapendo questa informazione, possiamo stabilire che la funzione Φ che abbiamo trattato fino adesso prende il nome di Contrazione

Dimostrazione:

\Leftarrow) Sia $\rho(B) < 1$ e mostriamo che $\|e_k\| \rightarrow 0$ per $k \rightarrow +\infty$

Limitiamo la dimostrazione al caso in cui B sia una matrice diagonalizzabile, allora possiamo scrivere B come

$$B = X^{-1} \Lambda X \quad \text{con } \Lambda \text{ matrice con gli autovalori di } B$$

Questo implica che:

$$B^k = (X \Lambda X^{-1})^k = (X \Lambda X^{-1}) \cdot (X \Lambda X^{-1}) \cdots (X \Lambda X^{-1}) = X \Lambda^k X^{-1}$$

Da quanto abbiamo detto prima segue che:

$$e_k = B^k e_0 = (X \Lambda X^{-1}) e_0$$

Sapendo poi che Λ è la matrice diagonale con gli autovalori di B , segue che le varie potenze di Λ non sono altro che le potenze dei vari elementi della diagonale.

Avendo però che $\rho(B) < 1$, segue che $\lambda_i < 1$ per ogni autovalore di B , quindi abbiamo che:

$$\forall i \in \{1, \dots, n\} \quad \lambda_i \xrightarrow{k \rightarrow +\infty} 0$$

Da cui otteniamo che:

$$\|\Lambda^k\| \xrightarrow{k \rightarrow +\infty} 0$$

E quindi è verificata la convergenza della successione degli errori a zero al variare di k , quindi è verificata la convergenza dei $(x)_k$ a x^* , al variare di k e per ogni x_0 scelto.

\Rightarrow) Supponiamo che $\|e_k\| \rightarrow 0$ e dimostriamo che $\rho(B) < 1$

Supponiamo quindi per assurdo che esista un'autocoppia (λ, v) tale che $|\lambda| \geq 1$

Scegliamo come x_0 un vettore iniziale tale che $e_0 = v$ (Mi basta per esempio prendere $x_0 = x^* + v$)

Allora otteniamo che:

$$e_k = B^k(e_0) = B^k v = \lambda^k v$$

Ma abbiamo che questo non tende a 0, infatti:

$$|\lambda| = 1 \Rightarrow \|\lambda^k v\| = \|v\| \quad |\lambda| > 1 \Rightarrow \|\lambda^k v\| \rightarrow +\infty$$

E ciò è assurdo in quanto abbiamo trovato un x_0 per cui l'errore non diminuisce, o resta costante o diverge

□

Considerazioni del Teorema: Per poter parlare di convergenza abbiamo che e_k deve tendere a 0 indipendentemente dal valore x_0 scelto, altrimenti non si parlerebbe di convergenza

Corollario

Se $\|B\| < 1$ per qualche norma matriciale indotta, allora $(x)_k$ converge ad un valore maggiore di 0

Dimostrazione:

Segue direttamente dal teorema precedente

□

Osservazione: Più è piccolo $\rho(B)$ meglio è.

Volendo possiamo prendere lo splitting come $A = P - N$ ponendo $A = P$ e $N = 0$ però non porta a niente perché:

$$B = P^{-1}A = A^{-1}(A - A) = 0$$

Se devo risolvere un sistema lineare, questo non mi serve a niente

Ruolo di $\rho(B)$

Vediamo come $\rho(B)$ e $\frac{\|e_k\|}{\|e_{k-1}\|}$ sono collegate, in particolare andiamo a vedere come diminuisce $\frac{\|e_k\|}{\|e_{k-1}\|}$

Per vedere come cambia possiamo vedere come si comporta con la media geometrica per ogni passo k :

$$\left(\frac{\|e_k\|}{\|e_{k-1}\|} \cdot \frac{\|e_{k-1}\|}{\|e_{k-2}\|} \cdots \cdot \frac{\|e_2\|}{\|e_1\|} \cdot \frac{\|e_1\|}{\|e_0\|} \right)^{1/k}$$

Notiamo tuttavia che la maggior parte dei fattori si semplifica, quindi otteniamo che:

$$\left(\frac{\|e_k\|}{\|e_{k-1}\|} \cdot \frac{\|e_{k-1}\|}{\|e_{k-2}\|} \cdots \cdot \frac{\|e_2\|}{\|e_1\|} \cdot \frac{\|e_1\|}{\|e_0\|} \right)^{1/k} = \left(\frac{\|e_k\|}{\|e_0\|} \right)^{1/k} \leq \left(\frac{\|B^k\| \cdot \|e_0\|}{\|e_0\|} \right)^{1/k} = \|B^k\|^{1/k}$$

Questo elemento può essere considerato come fattore medio di convergenza dopo k iterazioni.

Teorema di Gelfand

Sia $B \in \mathbb{R}^{n \times n}$ e sia $\|\cdot\|$ una norma matriciale, allora

$$\lim_{k \rightarrow +\infty} \|B^k\|^{1/k} = \rho(B)$$

Quindi rappresenta un fattore asintotico di convergenza

Dimostrazione:

Sia $\|\cdot\|$ la norma indotta dalla norma euclidea

Notiamo che

$$(\rho(B))^k = \rho(B^k) \leq \|B^k\| \Rightarrow \rho(B) \leq \|B\|^{1/k}$$

Fissiamo arbitrariamente $\varepsilon > 0$ e definiamo \hat{B} come:

$$\hat{B} = \frac{1}{\rho(B) + \varepsilon} B \quad \Rightarrow \quad \rho(\hat{B}) = \left(\frac{1}{\rho(B) + \varepsilon} \right) \rho(B) < 1$$

Da cui segue che:

$$\lim_{k \rightarrow +\infty} \|\hat{B}^k\| = 0$$

Esiste quindi un \bar{k} tale che per $k > \bar{k}$ si ha che:

$$\|\hat{B}^k\| < 1 \quad \Leftrightarrow \quad \left\| \left(\frac{1}{\rho(B) + \varepsilon} B \right)^k \right\| < 1$$

Abbiamo quindi ottenuto una stima dall'alto

Quindi riassumendo abbiamo che:

$$\rho(B) < \|B^k\|^{1/k} < \rho(B) + \varepsilon$$

Inoltre, per l'arbitrarietà di ε abbiamo che per $k \rightarrow +\infty$ otteniamo la tesi del teorema

□

Metodo di Jacobi

Si differenza dagli altri metodi in base alla scelta della matrice P

Per quanto abbiamo visto precedentemente abbiamo che $A = P - N$

Il metodo di Jacobi prevede che

$$A = -E + D - F$$

Con D matrice diagonale, E matrice strettamente triangolare inferiore e F matrice triangolare strettamente superiore

In questo caso poniamo:

$$P = D \quad N = E + F$$

Per simmetria rispetto a prima, visto che P era non singolare, allora anche D deve essere non singolare, in particolare

$$\forall i \in \{1, \dots, n\} \quad A_{i,i} \neq 0$$

Algoritmo di Jacobi:

Scelti $x_0 \in \mathbb{R}^n$, $r_0 = b - Ax_0$ e $\|r_0\| = \rho_0$

Per $k = 0, 1, \dots, maxit, tol$

$$x_{k+1} = x_k + D^{-1}r_k$$

$$r_{k+1} = b - Ax_{k+1}$$

Criterio di Arresto: $\|r_{k+1}\| < tol \cdot \rho_0$

L'unica cosa che cambia nell'effettivo è la risoluzione del sistema lineare $Dw_k = r_k \Rightarrow w_k = r_k / d_k$

Tutto il resto è uguale a prima, quindi ha un costo computazionale pari a $\Theta(n)$

Osservazione: Il calcolo del raggio spettrale è costosissimo, infatti è un problema non lineare.

Visto che vogliamo risolvere un sistema lineare $Ax = b$ non ha senso ottenere utilizzare un'informazione ottenuta attraverso un procedimento non lineare. Aumenterebbe semplicemente il costo, in quanto sono già esistenti metodi che ne fanno a meno

Metodo di Gauss - Seidel

Partendo sempre dalla divisione $A = -E + D - F$ prendiamo

$$P = -E + D \quad N = F$$

Quindi per l'ipotesi di non singolaritàabbiamo che

$$\forall i \in \{1, \dots, n\} \quad A_{i,i} \neq 0$$

Algoritmo di Gauss - Seidel

Scelti $x_0 \in \mathbb{R}^n$, $r_0 = b - Ax_0$ e $\|r_0\| = \rho_0$

Per $k = 0, 1, \dots, maxit, tol$

$$x_{k+1} = x_k + (D - E)^{-1}r_k$$

$$r_{k+1} = b - Ax_{k+1}$$

Criterio di Arresto: $\|r_{k+1}\| < tol \cdot \rho_0$

Quello che cambia dal metodo di Jacobi è che al posto di dover risolvere un sistema lineare con una matrice D diagonale, abbiamo una matrice $(D - E)$ triangolare inferiore, che generalmente ha un costo pari a $\Theta(n^2)$ al posto di un costo lineare $\Theta(n)$, quindi tutto l'algoritmo ha un costo pari a $\Theta(n^2)$

Esempi sul Confronto tra il metodo di Jacobi e il metodo di Guass-Seidel

Analizziamo tre casi in cui è più conveniente o meno usare un metodo rispetto all'altro

1. Prendiamo il sistema lineare $Ax = b$ dato da:

$$\begin{pmatrix} 3 & 0 & 4 \\ 7 & 4 & 2 \\ -1 & -1 & -2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 13 \\ -4 \end{pmatrix}$$

Metodo di Jacobi

Otteniamo che la matrice P è:

$$P = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & -2 \end{pmatrix}$$

Da cui otteniamo che il valore di $\rho(P^{-1}N)$ è all'incirca:

$$\rho(P^{-1}N) \approx 1,33$$

Da ciò segue che non converge

Metodo di Gauss - Seidel

Otteniamo che la matrice P è:

$$P = \begin{pmatrix} 3 & 0 & 0 \\ 7 & 4 & 0 \\ -1 & -1 & -2 \end{pmatrix}$$

Da cui otteniamo che il valore di $\rho(P^{-1}N)$ è all'incirca:

$$\rho(P^{-1}N) \approx 0,25$$

Da ciò segue che converge

2. Prendiamo adesso il sistema lineare:

$$\begin{pmatrix} -3 & -3 & 6 \\ -4 & 7 & -8 \\ 5 & 7 & -9 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ -5 \\ 3 \end{pmatrix}$$

Metodo di Jacobi

Otteniamo che la matrice P è:

$$P = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 7 & 0 \\ 0 & 0 & -9 \end{pmatrix}$$

Da cui otteniamo che il valore di $\rho(P^{-1}N)$ è all'incirca:

$$\rho(P^{-1}N) \approx 0,813$$

Da ciò segue che converge

Metodo di Gauss - Seidel

Otteniamo che la matrice P è:

$$P = \begin{pmatrix} 3 & 0 & 0 \\ -4 & 7 & 0 \\ 5 & 7 & -9 \end{pmatrix}$$

Da cui otteniamo che il valore di $\rho(P^{-1}N)$ è all'incirca:

$$\rho(P^{-1}N) \approx 1,12$$

Da ciò segue che non converge

3. Sia adesso il sistema lineare dato da:

$$\begin{pmatrix} 4 & 1 & 1 \\ 2 & -9 & 0 \\ 0 & -8 & -6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 6 \\ -7 \\ -14 \end{pmatrix}$$

Metodo di Jacobi

Otteniamo che la matrice P è:

$$P = \begin{pmatrix} 4 & 0 & 0 \\ 0 & -9 & 0 \\ 0 & 0 & -6 \end{pmatrix}$$

Da cui otteniamo che il valore di $\rho(P^{-1}N)$ è all'incirca:

$$\rho(P^{-1}N) \approx 0,44$$

Da ciò segue che converge

Metodo di Gauss - Seidel

Otteniamo che la matrice P è:

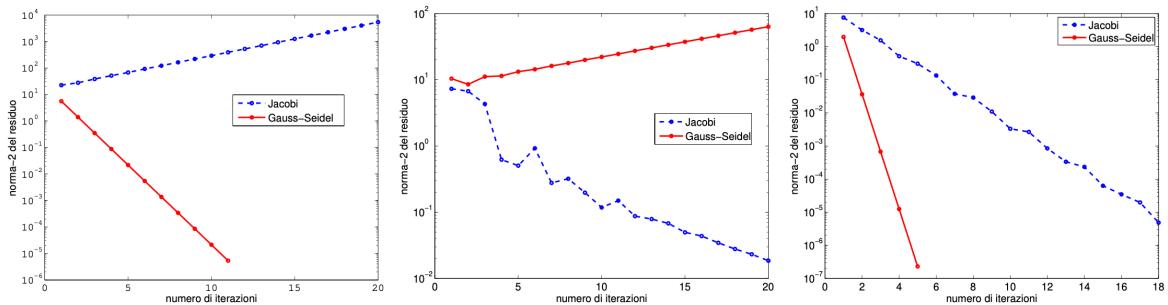
$$P = \begin{pmatrix} 4 & 0 & 0 \\ 2 & -9 & 0 \\ 0 & -8 & -6 \end{pmatrix}$$

Da cui otteniamo che il valore di $\rho(P^{-1}N)$ è all'incirca:

$$\rho(P^{-1}N) \approx 0,018$$

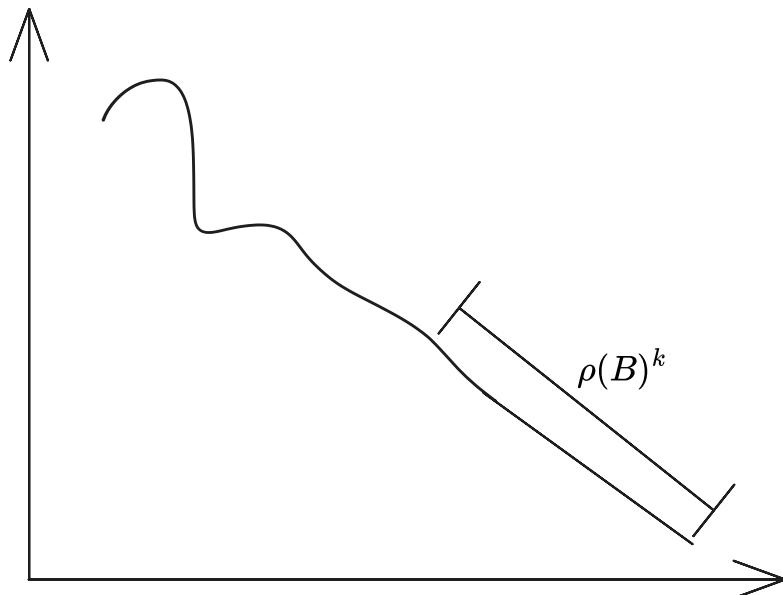
Da ciò segue che converge molto più velocemente

Con i grafici otteniamo che:



In generale per l'andamento bisogna tenere conto anche della sparsità della matrice e dei servizi aperti sulla macchina *che possono influire sul funzionamento della macchina*

Osservazione: La "storia della convergenza" (l'andamento del grafico) per un k grande sarà asintotico a $\rho(B)^k$



Metodo SOR

Questo metodo lo menzioneremo soltanto perché non abbiamo il tempo di farlo nel dettaglio

Questo metodo ha l'obiettivo di migliorare il metodo di Gauss - Seidel

In questo modo la scomposizione di A diventa:

$$A = \underbrace{-E + \frac{1}{\omega}D}_{P} - \underbrace{\left(\left(\frac{1}{\omega} - 1 \right) D + F \right)}_{N} \quad \omega \in]0, 2[$$

Il metodo inoltre è più o meno efficiente al variare di ω

Inoltre SOR sta per *Over Relaxation Method*

Risultati di Convergenza

Prima di vedere i primi risultati importanti di questi metodi diamo la seguente definizione:

Definizione di Matrice a Dominanza Diagonale Stretta

Sia $A \in \mathbb{R}^{n \times n}$ una matrice. Si dice che è a maggioranza diagonale stretta se

$$|A_{i,i}| > \sum_{j=1, j \neq i}^n |A_{i,j}| \quad \forall i \in \{1, \dots, n\}$$

Teorema

Sia $A \in \mathbb{R}^{n \times n}$ non singolare a dominanza diagonale stretta per righe, allora i metodi di Jacobi e di Gauss - Seidel convergono e vale la seguente diseguaglianza:

$$\|B_{GS}\|_\infty \leq \|B_J\|_\infty < 1$$

Dimostrazione:

Mostriamo prima la convergenza dei metodi

Metodo di Jacobi: Per questioni di comodità poniamo $B_J = B$ per questioni di indici.

Definiamo la norma infinito di una matrice come:

$$\|B\|_\infty = \max_i \sum_{j=1}^n |B_{i,j}|$$

Sappiamo inoltre che:

$$B = P^{-1}N = D^{-1}(E + F) = (\setminus) \cdot ((\Delta) + (\nabla))$$

Otteniamo quindi che, per $i \neq j$ abbiamo che:

$$B_{i,j} = \frac{A_{i,j}}{A_{i,i}}$$

Ossia ogni i -esima riga viene divisa dall'elemento $A_{i,i}$

Quindi otteniamo che la matrice B sarà della forma:

$$B = \begin{pmatrix} 0 & \frac{A_{1,2}}{A_{1,1}} & \cdots & \frac{A_{1,n}}{A_{1,1}} \\ \frac{A_{2,1}}{A_{2,2}} & 0 & \cdots & \frac{A_{2,n}}{A_{2,2}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{A_{n,1}}{A_{n,n}} & \frac{A_{n,2}}{A_{n,n}} & \cdots & 0 \end{pmatrix}$$

Abbiamo quindi che:

$$\|B\|_\infty = \max_i \sum_{j=1}^n |B_{i,j}| \xrightarrow{B_{i,i}=0} \max_i \sum_{j=1, j \neq i}^n \frac{|A_{i,j}|}{|A_{i,i}|} = \max_i \frac{1}{|A_{i,i}|} \sum_{j=1, j \neq i}^n |A_{i,j}| < 1$$

In particolare abbiamo che tutto questo è minore di uno in quanto segue che la matrice era dominante diagonale stretta

Metodo di Gauss - Seidel: In questo caso abbiamo che

$$B_{GS} = (-E + D)^{-1}F \quad \text{e} \quad A = -E + D - F$$

Sia (λ, v) un'autocoppia di B_{GS} . Allora abbiamo che $B_{GS}v = \lambda v$ in particolare:

$$(-E + D)^{-1}Fv = \lambda v$$

Riordinando di nuovo i termini otteniamo che:

$$\lambda Dv = Fv + \lambda Ev$$

Prendiamo k come indice tale che

$$|v_k| = \max_j |v_j|$$

Allora per la k -esima riga dell'uguaglianza $(-E + D)^{-1}Fv = \lambda v$ vale:

$$\lambda A_{k,k}v_k = \underbrace{\sum_{j=k+1}^n A_{k,j}v_j}_{k \rightarrow Fv} + \lambda \underbrace{\sum_{j=1}^{k-1} A_{k,j}v_j}_{k \rightarrow Ev}$$

Supponiamo ora per assurdo che esiste un autovalore λ tale che il suo modulo è maggiore o uguale di 1:

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

In particolare, passando per norme e moduli abbiamo che:

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

Dividiamo adesso tutto per $|v_k|$ e otteniamo che:

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

Sapendo inoltre che $\frac{|v_j|}{|v_k|} \leq 1$ allora possiamo maggiorare e otteniamo che:

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

Il che è assurdo in quanto avevamo che A è una matrice a diagonale dominante stretta per righe per Ipotesi, quindi segue che ogni autovalore ha norma minore stretta di 1

□

Vediamo altre considerazioni sul metodo di Gauss - Seidel

Teorema

Sia $A \in \mathbb{R}^{n \times n}$ simmetrica non singolare con diagonale strettamente positiva (cioè la matrice D per come l'abbiamo posta è definita positiva, $D > 0$). Allora il metodo di Gauss - Seidel converge se e solo se A è definita positiva

Prima di dare la dimostrazione, ci servirà questo lemma:

Lemma

La matrice

$$A - B^T AB > 0$$

dove $B \equiv B_{GS}$

Dimostrazione:

Sapendo che

$$A = P - N \Rightarrow N = P - A \Rightarrow P^{-1}N = P^{-1}(P - A) = I - P^{-1}A$$

Ma abbiamo che A è simmetrica, quindi:

$$A = \underbrace{-E + D}_{P} - E^T \Rightarrow P^{-1}N = I - P^{-1}A = I - \underbrace{(-E + D)^{-1}A}_{H} = I - H$$

Abbiamo che H è non singolare in quanto è prodotto di matrici non singolari

Quindi abbiamo che:

$$A - B^T AB = A - (I - H)^T A(I - H) = A - A + AH + H^T A - H^T AH = H^T((H^T)^{-1}A + AH^{-1} - A)H$$

Tuttavia sapendo che:

$$AH^{-1}AA^{-1}(-E + D) = -E + D \quad (H^T)^{-1}A = (AH^{-1})^T = (-E + D)^T$$

Otteniamo allora che:

$$H^T(-E^T + D - E + D + E - D + E^T)H = H^T DH > 0$$

Quindi D ha la stessa segnatura di A e poiché D è definita positiva, anche A è definita positiva

□

Dimostrazione del Teorema:

⇐) Supponiamo A definita, mostriamo che il metodo converge

Sia (λ, v) un'autocoppia di B .

Notiamo che $v \in \mathbb{C}^n, \lambda \in \mathbb{C}$ in quanto difficilmente una matrice qualsiasi B sia simmetrica

Allora abbiamo che:

$$0 < v^*(A - B^T AB)v = v^*Av - \underbrace{v^*B^T A}_{\lambda v^*} \underbrace{Bv}_{\lambda v} = v^*Av - |\lambda|^2 v^*Av = (1 - |\lambda|^2)v^*Av$$

Ma se $v^*Av > 0$ per Ipotesi allora abbiamo che:

$$1 - |\lambda|^2 > 0 \Rightarrow |\lambda|^2 < 1 \Rightarrow |\lambda| < 1$$

⇒) Sappiamo che il metodo converge, mostriamo che A è definita positiva

Ricordiamo che $e_{k+1} = Be_k$ allora segue che:

$$0 < e_k^T(A - B^T AB)e_k = e_k^T Ae_k - \underbrace{e_k^T B^T A}_{e_{k+1}} \underbrace{Be_k}_{e_{k+1}} = e_k^T Ae_k - e_{k+1}^T Ae_{k+1}$$

Otteniamo quindi:

$$e_{k+1}^T Ae_{k+1} < e_k^T Ae_k$$

Cioè la successione $(e_k^T Ae_k)_k$ è monotona decrescente

Supponiamo A indefinita (in particolare non definita positiva) *quindi il raggio spettrale contiene il valore 0*

Scegliamo $x_0 \in \mathbb{R}^n$ tale che $e_0^T Ae_0 < 0$, in particolare abbiamo che:

$$\dots < e_2^T Ae_2 < e_1^T Ae_1 < e_0^T Ae_0 < 0$$

Cioè abbiamo che la successione $e_k^T Ae_k$ non tende a zero per $k \rightarrow +\infty$

Non c'è convergenza, assurdo in quanto avevamo supposto che il metodo convergesse.

Quindi A è definita positiva

□

Teorema

Sia $A \in \mathbb{R}^{n \times n}$ è tridiagonale, non necessariamente simmetrica, e non singolare. Allora si ha che

$$\rho(B_{GS}) = (\rho(B_J))^2$$

Dimostrazione:

Non è da fare ma volendo c'è sulle dispense

□

Considerazioni del Teorema: Se $\rho(B) > 1$ divergono entrambi. Se invece $\rho(B) < 1$ invece convergono entrambe e il metodo di Gauss - Seidel converge più velocemente di Jacobi. È anche più veloce a livello computazionale. Ma in generale è un po' una mazzata usare un metodo iterativo, in quanto abbiamo già l'algoritmo di Thomas.

Metodi Non Stazionari: Metodo dei Gradienti Coniugati

Questo tipo di metodi è utilizzato per i problemi di grandi dimensioni. In particolare possono essere utili per le equazioni del moto, di cui infatti non conosciamo le soluzioni esatte

Le uniche soluzioni che possiamo avere sono delle approssimazioni, per esempio passando da un dominio concavo ad uno convesso. Ma anche in tal caso non riusciremmo ad ottenere una soluzione accurata in forma chiusa, il meglio che possiamo ottenere sono delle serie, che da qualche parte dovremmo troncare.

Il metodo che utilizzeremo è quello dei gradienti coniugati. (seppur troppo specifico)

Dato un sistema lineare $A\underline{x} = \underline{b}$ con $A \in \mathbb{R}^{n \times n}$ simmetrica e definita positiva $A > 0$

Vogliamo determinare una successione $(x)_k$ con $x_0 \in \mathbb{R}^n$ fissato che sia del tipo

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

dove p_k è un vettore direzione.

Diversamente dagli altri metodi, in cui andavamo a modificare la matrice, qui siamo in uno spazio vettoriale e possiamo muoverci lungo appunto il vettore direzionale.

Nel fare ciò, consideriamo la seguente funzione:

$$\phi : \mathbb{R}^n \rightarrow \mathbb{R} \quad \phi(x) = \frac{1}{2} x^T A x - b^T x$$

Il problema sarà allora trovare il punto minimo della funzione, in particolare trovare:

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

In particolare, se x^* è la soluzione esatta, ossia $Ax^* = b$ allora x^* è minimo dei ϕ e quindi:

$$\nabla \phi(x)|_{x=x^*} = Ax - b|_{x=x^*} = 0$$

Andiamo a vedere in particolare il caso di $n = 2$:

$$\begin{aligned} \phi(x_1, x_2) &= \frac{1}{2} (x_1 \ x_2) \begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - (b_1 \ b_2) \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \frac{1}{2} (x_1 \ x_2) \begin{pmatrix} a_{1,1}x_1 + a_{1,2}x_2 \\ a_{2,1}x_1 + a_{2,2}x_2 \end{pmatrix} - (b_1x_1 + b_2x_2) \\ &= \frac{1}{2} (a_{1,1}x_1^2 + a_{1,2}x_1x_2 + a_{2,1}x_1x_2 + a_{2,2}x_2^2) - (b_1x_1 + b_2x_2) \end{aligned}$$

Da tutto ciò otteniamo che:

$$\left. \begin{aligned} \frac{\partial \phi}{\partial x_1} &= \frac{1}{2} (2a_{1,1}x_1 + a_{1,2}x_2 + a_{2,1}x_2) - b_1 = \begin{pmatrix} a_{1,1} & a_{1,2} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = b_1 \\ \frac{\partial \phi}{\partial x_2} &= \frac{1}{2} (2a_{2,2}x_1 + a_{1,2}x_1 + a_{2,1}x_1) - b_2 = \begin{pmatrix} 0 & 0 \\ a_{2,1} & a_{2,2} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = b_2 \end{aligned} \right\} A\underline{x} = \underline{b}$$

Abbiamo sommato $a_{1,2}x_i$ e

Se sostituiamo x^* otteniamo esattamente 0

Verifichiamo che è un punto di minimo

Prendiamo un $y \in \mathbb{R}^n$, $y \neq 0$ e vediamo che $\phi(x^* + y) > \phi(x^*)$

$$\phi(x^* + y) = \frac{1}{2} (x^* + y)^T A (x^* + y) - b^T (x^* + y) = \frac{1}{2} ((x^*)^T A x^* + (x^*)^T A y + y^T A x^* + y^T A y) - b^T x^* - b^T y$$

Facciamo un paio di osservazioni che ci semplificheranno un po' i conti:

- $\frac{1}{2} (x^*)^T A x^* - b^T x^* = \phi(x^*)$ dalla definizione della funzione

- Visto che A è simmetrica abbiamo che $(x^*)^T A y = y^T A x^*$

Andando avanti otteniamo che:

$$\phi(x + y) = \phi(x) + (x^*)^T A y - b^T y + \frac{1}{2} y^T A y = \phi(x) + ((x^*)^T A - b^T)y + \frac{1}{2} y^T A y = \phi(x) + (\underbrace{A x^* - b}_0)^T y + \frac{1}{2} y^T A y = \phi(x) + \frac{1}{2} y^T A y$$

E questo è senz'altro maggiore di $\phi(x)$ in quanto y è non nullo e A è definita positiva

Andiamo adesso a cercare un'iterazione che sia della forma:

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

con l'unico vincolo che

$$p_k^T \nabla \phi(x_k) < 0$$

Questo perché normalmente $\nabla \phi(x)$ ci dice dove la funzione cresce, però per sapere dove sta il minimo ci serve sapere dove decresce e per questo ci serve che p_k deve puntare nella direzione di decrescita.

Andrà bene qualsiasi vettore p_k purché valga la condizione $p_k^T \nabla \phi(x_k) < 0$

Quindi possiamo già determinare una ricorrenza per il residuo r_{k+1} , ossia:

$$r_{k+1} = b - Ax_{k+1}$$

Infatti abbiamo che, andando a sostituire x_k con la definizione della successione abbiamo che:

$$r_{k+1} = b - Ax_{k+1} = b - A(x_k + \alpha_k p_k) = \underbrace{b - Ax_k}_{r_k} - \alpha_k Ap_k = r_k - \alpha_k Ap_k$$

Ora ci rimane da scegliere α_k e p_k

Per la scelta di α_k possiamo restringere la funzione alla retta lungo la direzione p_k e trovare il minimo di tale funzione, in base a quel minimo possiamo scegliere α_k , ossia dobbiamo andare a scegliere:

$$\min_{\alpha \in \mathbb{R}} \phi(x_k + \alpha p_k)$$

In particolare, enunciamo la seguente proposizione:

Proposizione

Per x_k e $p_k \in \mathbb{R}^n$ fissati, la scelta di

$$\alpha = \frac{p_k^T r_k}{p_k^T A p_k} \quad \text{dove } r_k = b - Ax_k$$

soddisfa:

$$\phi(x_{k+1}) = \min_{\alpha} \phi(x_k + \alpha p_k) \quad \text{dove } x_{k+1} = x_k + \alpha p_k$$

Dimostrazione:

Facendo i calcoli abbiamo che:

$$\phi(x_k + \alpha p_k) = \frac{1}{2} (x_k + \alpha p_k)^T A (x_k + \alpha p_k) - b^T (x_k + \alpha p_k)$$

Per cui, facendo la derivata per α abbiamo che:

$$\frac{d\phi}{d\alpha} = \frac{1}{2} p_k^T A x_k - \frac{1}{2} 2\alpha p_k^T A p_k + \frac{1}{2} x_k^T A p_k - b^T p_k$$

Sapendo che la matrice è simmetrica abbiamo che $p_k^T A x_k = x_k^T A p_k$ e da ciò abbiamo che:

$$\frac{d\phi}{d\alpha} = x_k^T A p_k - b^T p_k + \alpha p_k^T A p_k = (x_k^T A - b^T) p_k = -r_k^T p_k + \alpha p_k^T A p_k$$

Ma queste deve essere nullo in quanto abbiamo posto che era il minimo

Allora abbiamo che

$$\frac{d\phi}{d\alpha} = 0 \quad \Leftrightarrow \quad \alpha = \frac{r_k^T p_k}{p_k^T A p_k}$$

che è esattamente quanto detto nell'enunciato

□

Osservazione: La derivata seconda di ϕ è

$$\frac{d^2\phi}{d\alpha^2} = p_k^T A p_k > 0 \quad \text{in quanto } A > 0$$

Abbiamo inoltre che α è ben posta in quanto $p_k^T A p_k > 0$ sempre perché A è definita positiva
Abbiamo anche che, per come abbiamo posto ϕ , segue che:

$$\nabla\phi(x_k) = b - Ax_k = -r_k$$

Da cui otteniamo che:

$$r_k^T p_k = -\nabla\phi(x_k)^T p_k > 0 \quad \Rightarrow \quad \alpha > 0$$

Quindi abbiamo che il vettore punta nella direzione giusta.

Andiamo a studiare $p_k^T r_{k+1}$:

$$p_k^T r_{k+1} \xrightarrow{r_{k+1}=r_k-\alpha A p_k} p_k^T r_k - \alpha p_k^T A p_k = p_k^T r_k - \frac{r_k^T p_k}{p_k^T A p_k} (p_k^T A p_k) = 0$$

da cui segue che

$$r_{k+1} \perp p_k$$

Ora abbiamo un modo per calcolare α , dobbiamo però trovare un modo per calcolare p_k

Ne esistono di vari modi, ma a seconda della scelta esistono modi diversi:

- *Metodo di Discesa Ripida:* Ci basta porre

$$p_k = r_k = -\nabla\phi(x_k)$$

In questo caso troviamo delle direzioni ortogonali tra loro, dato in base alla scelta di α

È buono utilizzare questo metodo se abbiamo una circonferenza e non un'ellisse attorno al punto di minimo
Comunque sia, nonostante il nome, non è il metodo più veloce

- *Metodo del gradiente coniugato:* Ci basta prendere i vettori direzione p_k come A -ortogonali o A -coniugati

Prendiamo inizialmente

$$p_0 = r_0 \quad \Rightarrow \quad p_{k+1} = r_{k+1} + \beta_K r_k$$

Prendendo β_k in modo da avere:

$$p_{k+1}^T A p_k = 0$$

Cioè, andando nel dettaglio:

$$(r_{k+1} + \beta_k p_k)^T A p_k = 0 \Rightarrow r_{k+1}^T A p_k + \beta_k p_k^T A p_k = 0 \Rightarrow \beta_k = \frac{-r_{k+1}^T A p_k}{p_k^T A p_k}$$

Il valore di β è ben posto in quanto abbiamo che $A > 0$, ossia è definita positiva

Da quanto avevamo dai punti precedenti abbiamo che:

$$\alpha_k = \frac{p_k^T r_k}{p_k^T A p_k} \quad r_{k+1} = r_k - \alpha_k A p_k \quad x_{k+1} = x_k + \alpha_k p_k \quad p_{k+1} = r_{k+1} + \beta_k r_k \quad \beta_k = \frac{-r_{k+1}^T A p_k}{p_k^T A p_k}$$

Viene però soddisfatta la condizione $p_{k+1}^T \nabla\phi(x_{k+1}) < 0$?

Osservazione: p_{k+1} effettivamente precedente, infatti:

$$p_{k+1}^T \nabla\phi(x_{k+1}) = -p_{k+1}^T r_{k+1} = -(r_{k+1} + \beta_k p_k)^T r_{k+1} = -r_{k+1}^T r_{k+1} - \underbrace{\beta_k p_k^T r_{k+1}}_0 = -\|r_{k+1}\|^2 < 0$$

$\beta_k p_k^T r_{k+1} = 0$ perché per le osservazioni precedenti avevamo che erano ortogonali

Finché ci avviciniamo alla soluzione va bene, non ci sono vincoli sulla scelta di una direzione rispetto ad un'altra.

Tutto questo mi permette di scrivere α_k in un'altra forma:

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

Il poter scrivere α_k come norma di un vettore su una costante è molto meglio di scrivere come prodotto scalare su una costante, non ci sono problemi di arrotondamento e soprattutto evitiamo somme di valori positivi con valori negativi, che sono spesso motivo di origine di arrotondamenti o altro ancora

Osservazione: Notiamo che i residui sono ortogonali tra loro $r_k^T r_{k-1} = 0$. Infatti sapendo che

$$p_{k+1} = r_{k+1} + \beta_k p_k \quad \Leftrightarrow \quad p_{k-1} = r_{k-1} + \beta_{k-2} p_{k-2}$$

Otteniamo che:

$$r_k^T r_{k+1} = r^T (p_{k-1} - \beta_{k-2} p_{k-2}) = \underbrace{r_k^T p_{k-1}}_0 - \beta_{k-2} r_k^T p_{k-2}$$

Questo è nullo sempre per sopra

Da ciò otteniamo che:

$$-\beta_{k-2} r_k^T p_{k-2} = -\beta_{k-2} (r_{k-1} - \alpha_{k-1} A p_{k-1})^T p_{k-2} = -\beta_{k-2} (\underbrace{r_{k-1}^T p_{k-2}}_0 - \underbrace{\alpha_{k-1} p_{k-1}^T A p_{k-2}}_0) = 0$$

Il primo è nullo sempre per il fatto che il residuo è ortogonale con il vettore posizione all'iterazione precedente, mentre il secondo è nullo perché sono vettori A -coniugati, per la condizione che avevamo posto e per la scelta di β

Proprio da questa proprietà segue il metodo che prende il nome di metodo dei gradienti coniugati:

$$r_k^T r_{k-1} = \nabla \phi(x_k)^T \nabla \phi(x_{k-1})$$

Osservazione: Da qui si ottiene che

$$\beta_k = \frac{r_{k+1}^T A p_k}{p_k^T A p_k} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k} = \frac{\|r_{k+1}\|^2}{\|r_k\|^2}$$

Algoritmo dei Gradienti Coniugati:

Fissiamo $x_0 \in \mathbb{R}^n$, $r_0 = b - Ax_0$, $p_0 = r_0$, $\rho_0 = \|r_0\|$, $maxit$ e tol

Per $k = 0, 1, \dots, maxit$

$w = Ap_k$	<i>che ha un costo di Mxv</i>
$\alpha_k = \frac{r_k^T r_k}{p_k^T w}$	<i>che ha un costo di $2(n-1)$</i>
$x_{k+1} = x_k + \alpha_k p_k$	<i>che ha un costo di $2n$</i>
$r_{k+1} = r_k + \alpha_k w$	<i>che ha un costo di $2n + o(1)$</i>
$\rho_{k+1} = \ r_{k+1}\ $	<i>che ha un costo di $2n + o(1)$</i>
Criterio d'Arresto: $\ r_{k+1}\ < tol \cdot \rho_0$	
$\beta_k = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$	
$p_{k+1} = r_{k+1} + \beta_k p_k$	<i>che ha un costo di $2n$</i>

Osservazione:

1. Avere α_k e β_k in questo modo non è solo positivo per quanto riguarda il round-off, ma anche il costo computazionale è molto migliore
2. Possiamo salvare $Ap_k = w$ in modo da non dover rifare le stesse operazioni ogni volta, serve per minimizzare i conti il più possibile
3. Il costo totale è quello di $10n$ flop + la risoluzione di un sistema lineare Mxv che dipende dalla scelta della matrice da un valore trascurabile

Il metodo dei gradienti coniugati è già implementato in Matlab e basta scrivere il comando `cg(A, b)`

Osservazione: Se $A > 0$ ma λ autovalore è molto molto piccolo, allora si potrebbero avere comunque dei problemi

Proposizione

Con la notazione precedente, supponiamo che $\|r_k\| \neq 0$ per $k < \bar{k}$, allora abbiamo:

1. $r_i^T r_j = 0, \forall i, h < \bar{k}, i \neq j$
2. $p_i^T A p_j = 0, \forall i \neq h, i, j < \bar{k}$

Osservazione: Se ho i vettori coniugati, allora posso prendere la matrice:

$$(p_0 \quad p_1 \quad \cdots \quad p_{n-1})$$

Questa è una matrice con vettori colonna linearmente indipendenti, quindi rappresentano una base di uno spazio vettoriale, in particolare, se chiamiamo

$$K_k = \text{Span}\{p_0, p_1, \dots, p_{n-1}\} \quad \Rightarrow \quad \dim(K_k) = k \quad (k < \bar{k})$$

Analogamente possiamo prendere la matrice:

$$(r_0 \quad r_1 \quad \cdots \quad r_n)$$

Anche questa matrice ha i vettori colonna che sono linearmente indipendenti e, se consideriamo lo Span di questi, otteniamo lo stesso spazio vettoriale K_k

Ma è anche vero che K_k è generato dalla base: $\{r_0, Ar_0, A^2r_0, \dots, A^{n-1}r_0\}$

Tutte queste sono basi diverse dello stesso spazio vettoriale, i vettori sono dei polinomi in A che godono delle stesse proprietà dei polinomi, infatti:

$$v \in K_k \text{ si può scrivere come } \sum_{j=1}^n \eta_j x^j$$

Prima di dimostrare le effettive uguaglianze, diamo un nome a questo spazio vettoriale che abbiamo appena ottenuto:

Definizione di Spazio Vettoriale di Krylov

Si definisce Spazio Vettoriale di Krylov e si indica con K_k oppure $K_k(A, r_0)$ lo spazio vettoriale:

$$K_k = \text{Span}\{r_0, Ar_0, A^2r_0, \dots, A^{n-1}r_0\}$$

Mostriamo la prima uguaglianza:

Per come abbiamo definito il metodo abbiamo che: $r_0 = p_0$

L'elemento successivo è $p_1 = r_1 + \beta_0 p_0 = r_1 + \beta_0 r_0 \in \text{Span}\{r_0, r_1\}$

Quello dopo ancora sarà: $p_2 = r_2 + \beta_1 p_1 = r_2 + \beta_1(r_1 + \beta_0 r_0) = r_2 + \beta_1 + \beta_0 \beta_1 r_0 \in \text{Span}\{r_0, r_1, r_2\}$

E così via anche per gli altri

Mostriamo la seconda uguaglianza: banalmente $r_0 \in \text{Span}\{r_0\}$

Per come abbiamo definito la successione dei residui abbiamo che: $r_1 = r_0 - \alpha_0 A p_0 = r_0 - \alpha_0 A r_0 \in \text{Span}\{r_0, Ar_0\}$

E poi:

$$r_2 = r_1 - \alpha_1 A p_1 = r_1 - \alpha_1 A(r_1 + \beta_0 r_0) = r_0 - \alpha_0 A r_0 - \alpha_1(Ar_0 - \alpha_0 A^2 r_0) - \alpha_1 \beta_0 r_0 \in \text{Span}\{r_0, Ar_0, A^2 r_0\}$$

Osservazione: $x_k \in K_k$

Teorema

Con la notazione precedente e l'ipotesi che A è simmetrica e definita positiva, valgono:

$$1 : \|x_k - x^*\|_A = \min_{x \in K_k} \|x - x^*\| \quad 2 : \|x_k - x^*\| \leq \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^k \|x_0 - x^*\|_A$$

Dimostrazione:

Dimostriamo giusto il primo punto

$$\|x^* - x_k\|_A = (x^* - x_k)^T A (x^* - x_k) = (x^*)^T A x^* - (x^*)^T A x_k - x_k^T A x^* + x_k^T A x_k$$

Sapendo che A è simmetrica si ottiene che:

$$\begin{aligned} \|x^* - x_k\|_A &= (x^*)^T A x^* - (x^*)^T A x_k - x_k^T A x^* + x_k^T A x_k = (x^*)^T A x^* - \underbrace{2(x^*)^T A x_k}_{b^T} + x_k^T A x_k = \\ &= (x^*)^T A x^* + x_k^T A x_k - 2b^T x_k = (x^*)^T A x^* + 2\underbrace{\left(\frac{1}{2} x_k^T A x_k - b^T x_k\right)}_{\phi(x_k)} = (x^*)^T A x^* + 2\phi(x_k) \end{aligned}$$

Quindi abbiamo che la norma dell'errore dipende esclusivamente da $\phi(x_k)$ e quindi per avere il minimo errore bisogna minimizzare $\phi(x_k)$ per $x \in \mathbb{K}_k$

□

Considerazioni del Teorema: Il teorema di dice che:

$$\|x_k - x^*\| \leq \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^k \|x_0 - x^*\|_A$$

Abbiamo che questa è una stima dall'alto ed è piccola se e solo se è piccolo il coefficiente davanti all'errore al passo 0 in norma energia di A . Possiamo notare come valga lo stesso ragionamento del raggio spettrale, ossia dipende esclusivamente da $\kappa(A)$, ossia da quanto ben posizionata è A effettivamente, in particolare:

$$\left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right) \approx 1 \Rightarrow \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^k \text{decrece rapidamente} \quad \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right) \ll 1 \Rightarrow \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^k \text{decrece rapidamente}$$

Il fatto che sia un rapporto implica che il condizionamento della matrice non deve stare in una posizione particolare, ma basta che l'intervallo tra $\sqrt{\kappa(A)} - 1$ e $\sqrt{\kappa(A)} + 1$ non sia troppo distante

Osservazione: Il metodo di discesa ripida, cioè ponendo $p_k = r_k$ per arrivare al punto velocemente ha una soglia superiore pari a:

$$\|x_k - x^*\| \leq \left(\frac{\kappa(A) - 1}{\kappa(A) + 1} \right)^k \|x_0 - x^*\|_A$$

Possiamo vedere appunto come potrebbe essere più lenta

Osservazione: Queste stime sono di tipo "asintotico"

Fattorizzazione QR di una Matrice

Il nome della fattorizzazione deriva proprio dai nomi delle matrici con cui andiamo a fattorizzare una matrice. Infatti, data una matrice $A \in \mathbb{R}^{n \times m}$, la fattorizzazione QR determina una matrice $Q \in \mathbb{R}^{n \times n}$ ortogonale e $R \in \mathbb{R}^{n \times m}$ triangolare superiore tali da:

$$A = QR$$

Scrivendolo con i simboli abbiamo che:

$$\text{se } n \geq m \quad (\mathbb{I}) = (\square) \cdot \begin{pmatrix} \nabla \\ \mathbf{0} \end{pmatrix}$$

Dove (∇) è una matrice triangolare superiore $m \times m$ e $\mathbf{0}$ è una matrice nulla $(n - m) \times m$

In particolare, se suddividiamo la matrice Q come:

$$Q = \left(\underbrace{Q_1}_{m \times n} : \underbrace{Q_2}_{(m-n) \times n} \right) \Rightarrow A = (Q_1 \ Q_2) \begin{pmatrix} R_1 \\ 0 \end{pmatrix} = Q_1 R_1$$

La fattorizzazione del tipo $Q_1 R_1$ come sopra prende il nome di fattorizzazione ridotta, o economy size, o **Skinny**

In Matlab la Fattorizzazione QR è già implementata, basta scrivere `help qr` per avere tutto il necessario

Osservazione: Se R_1 non è singolare, allora l'immagine di $\text{Range}(A) = \text{Range}(Q_1)$, infatti:

$$(a_1 \ a_2 \ a_3) = (q_1 \ q_2 \ q_3) \begin{pmatrix} r_{1,1} & r_{1,2} & r_{1,3} \\ & r_{2,2} & r_{2,3} \\ & & r_{3,3} \end{pmatrix} \Rightarrow a_j = \sum_{k=1}^3 q_j r_{k,j}$$

Se invece fosse singolare, possiamo supporre la stessa situazione di prima con $r_{3,3} = 0$ e in tal caso avremo che:

$$a_3 = q_1 r_{1,3} + q_2 r_{2,3} \Rightarrow a_3 \in \text{Span}\{q_1, q_2\} = \text{Span}\{a_1, a_2\}$$

Quindi a_3 è linearmente dipendente dagli altri, quindi A non ha Range (Immagine) massima

Quindi R_1 contiene informazioni sull'Immagine di A

In aritmetica esatta finché non ci sono 0 va tutto bene, numericamente parlando, basta un numero dell'ordine di grandezza di 10^{-14} , per rendere la matrice singolare, quindi non di Rango massimo

Prima di poter dire ancora altro, diamo le seguenti definizioni che non sono altro che ripasso di [Geometria 1A](#)

Definizione di Rango

Si definisce il Rango di A come il numero massimo di colonne/righe linearmente indipendenti

Definizione di Range

Si definisce Range di una matrice e si indica con $\text{Im}(A)$ lo spazio vettoriale:

$$\text{Im}(A) = \text{Span}\{y : Ax : x \in \mathbb{R}^m\} = \text{Span}\{a_1, a_2, \dots, a_m\} \quad a_i \text{ colonne di } A =$$

Riprendendo i discorsi precedenti si ha che:

$$A = Q_1 R_1 \Rightarrow \text{Range}(A) = \text{Range}(Q_1)$$

Ossia le colonne di Q_1 rappresentano una base ortonormale per A

Vedremo poi che, con Householder, possiamo trovare anche Q_2 sapendo che:

$$A = (Q_1 \ Q_2) \begin{pmatrix} R_1 \\ 0 \end{pmatrix} = Q_1 R_1$$

Osservazione: Visto che Q è ortogonale, possiamo trovare facilmente R come

$$A = QR \Rightarrow Q^T A = R \Rightarrow \begin{pmatrix} Q_1^T \\ Q_2^T \end{pmatrix} A = \begin{pmatrix} R_1 \\ 0 \end{pmatrix} \Rightarrow \begin{cases} Q_1^T A = R_1 \\ Q_2^T A = 0 \end{cases}$$

In particolare

$$Q_2^T A = 0 \Rightarrow A^T Q_2 = 0$$

Quindi le colonne di Q_2 generano $\text{Ker}(A^T) = N(A) = \text{Null}(A) = \{x \in \mathbb{R}^n : A^T x = 0\}$

Quindi Q_2 non è fondamentalmente inutile, genera $\text{Ker}(A^T)$

Tutto questo risulta estremamente comodo, anzi danno degli strumenti pratici per calcolare una base ortonormale di uno spazio vettoriale

Andiamo adesso a vedere il caso in cui A sia larga, ossia $A \in \mathbb{R}^{n \times m}$ con $n < m$. Allora possiamo scomporre:

$$A = QR \quad \text{con } Q \in \mathbb{R}^{n \times n} \text{ ortogonale e } R \in \mathbb{R}^{m \times n} \text{ triangolare superiore con } R = (\nabla \ *)$$

Per le prime colonne di A possiamo trovare una matrice triangolare superiore (∇), per le altre invece troviamo un complemento in quanto le altre colonne di A sono linearmente dipendenti dalle altre

In questo caso possiamo dire che il rango massimo di A è $Rg(A) = n$

Un altro tipo di fattorizzazione è del tipo:

$$A = (* \ *) \Rightarrow A^T = \begin{pmatrix} * \\ * \end{pmatrix} = QR \Rightarrow A = R^T Q^T = (\Delta \ 0)(Q)$$

Osservazione: Se ho un sistema lineare $Ax = b$ con $A \in \mathbb{R}^{n \times n}$, allora possiamo sfruttare la fattorizzazione QR per risolverlo:

$$Ax = B \Rightarrow Q \underbrace{Rx}_y = b \Rightarrow y = Q^T b \Rightarrow Rx = y$$

Questo può essere effettivamente più comodo se le colonne di A non sono delle migliori

Metodo di Gram - Schmidt

Sia la matrice A come:

$$A = (a_1 \ a_2 \ \cdots \ a_m) \in \mathbb{R}^{n \times m} \quad \text{con } a_i \text{ colonne di } A \quad \text{con } n \geq m$$

Vogliamo determinare una matrice

$$Q = (q_1 \ q_2 \ \cdots \ q_m) \in \mathbb{R}^{m \times m} \quad \text{tale che } Q^T Q = I$$

Supponiamo che i vettori delle colonne di A siano linearmente indipendenti, allora con il metodo di Gram - Schmidt:

Per il primo vettore abbiamo che:

$$q_1 = \frac{a_1}{\|a_1\|}$$

Una volta scelto il primo vettore, dobbiamo scegliere un secondo vettore che non abbia componenti in q_1 , quindi:

$$\hat{q}_2 = a_2 - q_1(q_1^T a_2)$$

Dobbiamo poi normalizzare il vettore:

$$q_2 = \frac{\hat{q}_2}{\|\hat{q}_2\|}$$

E poi andando avanti si ottiene che:

$$\hat{q}_m = a_m - \sum_{j=1}^{i-1} q_j q_j^T a_m \Rightarrow q_m = \frac{\hat{q}_m}{\|\hat{q}_m\|}$$

Possiamo far vedere che otteniamo vettori linearmente indipendenti perché:

$$q_1^T \hat{q}_2 = q_1^T a_2 - q_1^T q_1 q_1^T a_2 = q_1^T a_2 - \|q_1\| q_1^T a_2 = q_1^T a_2 - q_1^T a_2 = 0$$

Osservazione: Notiamo che

$$\hat{q}_2 = a_2 - q_1 q_1^T a_2 = (I - q_1 q_1^T) a_2$$

Matrici della forma $q_1 q_1^T$ e $I - q_1 q_1^T$ sono dette matrici di proiezioni ortogonali

Definizione di Proiezione Ortogonale

Una matrice P prende il nome di matrice di proiezione ortogonale se P è simmetrica e $P = P^2 \Leftrightarrow P(I - P) = 0$

Entrambe le matrici soddisfano questa proprietà. Infatti, se prendiamo P come l'avevamo posta precedentemente, abbiamo che

$$Px = q_1 q_1^T x$$

Questa non è altro che la naturale rappresentazione della proiezione di x lungo la direzione definita da q_1

Se invece prendo $P = I - q_1 q_1^T$, proietto un vettore nella direzione ortogonale al vettore q_1

Questo processo lo posso fare anche con più vettori, in tal caso avremmo che:

$$P = (q_1 \quad q_2) \begin{pmatrix} q_1^T \\ q_2^T \end{pmatrix} \in \mathbb{R}^{n \times n} \quad \Rightarrow \quad Px = q_1 q_1^T x + q_2 q_2^T x \in \text{Span}\{q_1, q_2\}$$

Se invece prendessi $I - P$, allora proietterei il vettore x nella direzione $(\text{Span}\{q_1, q_2\})^\perp$

Questa è la stessa identica cosa che faccio quando vado a calcolare \hat{q}_3

Ora sorge spontanea la domanda, come possiamo costruire R_1 ?

Riprendiamo i passaggi fatti in precedenza: per il primo vettore abbiamo che:

$$\hat{q}_1 = a_1 \quad \Rightarrow \quad q_1 = \frac{\hat{q}_1}{\|\hat{q}_1\|} \quad \Rightarrow \quad a_1 = q_1 \|\hat{q}_1\|$$

Per il secondo vettore abbiamo che:

$$\hat{q}_2 = a_2 - q_1 q_1^T a_2 \quad \Rightarrow \quad q_2 = \frac{\hat{q}_2}{\|\hat{q}_2\|} \quad \Rightarrow \quad a_2 = q_1 q_1^T a_2 + q_2 \|\hat{q}_2\| = (q_1 \quad q_2) \begin{pmatrix} q_1^T a_2 \\ \|\hat{q}_2\| \end{pmatrix}$$

Per il terzo vettore avremmo che:

$$\begin{aligned} \hat{q}_3 &= a_3 - q_1 q_1^T a_3 - q_2 q_2^T a_3 \\ q_3 &= \frac{\hat{q}_3}{\|\hat{q}_3\|} \end{aligned} \quad \Rightarrow \quad a_3 = q_1 q_1^T a_3 + q_2 q_2^T a_3 + q_3 \|\hat{q}_3\| = (q_1 \quad q_2 \quad q_3) \begin{pmatrix} q_1^T a_3 \\ q_2^T a_3 \\ \|\hat{q}_3\| \end{pmatrix}$$

Se la matrice fosse 3×3 otterremmo che:

$$\underbrace{(a_1 \quad a_2 \quad a_3)}_A = \underbrace{(q_1 \quad q_2 \quad q_3)}_Q \underbrace{\begin{pmatrix} \|\hat{q}_1\| & q_1^T a_2 & q_1^T a_3 \\ 0 & \|\hat{q}_2\| & q_2^T a_3 \\ 0 & 0 & \|\hat{q}_3\| \end{pmatrix}}_R$$

Abbiamo quindi che R_1 contiene i coefficienti dell'ortogonalizzazione di R_1 ed in particolare, sulla sua diagonale ci sono le norme dei vettori ottenuti dopo il processo di ortogonalizzazione

Se un elemento fosse nullo sulla diagonale di R , allora avremmo che alcune delle colonne di A sono linearmente dipendenti. Andiamo ad analizzare cosa succederebbe in questo caso.

Supponiamo che un elemento sulla diagonale sia nullo, per semplicità supponiamo che

$$\|\hat{q}_3\| = 0 \quad \Rightarrow \quad \hat{q}_3 = 0 \quad \Rightarrow \quad a_3 \text{ è combinazione lineare dei vettori precedenti}$$

Ossia:

$$a_3 = \alpha_1 a_1 + \alpha_2 a_2 \quad \text{con } \alpha_1, \alpha_2 \in \mathbb{R}$$

Quindi R ci dà informazioni sul rango di A

Nota: Se $\|\hat{q}_3\|$ è piccola (per esempio all'ordine di 10^{-8}) allora avremmo che il vettore a_3 è molto "vicino" dall'essere linearmente dipendente dai vettori precedenti

In aritmetica esatta questo non ha una grande valenza, in quanto a_3 avrebbe comunque una componente perpendicolare alle direzioni q_1 e q_2 , in aritmetica finita questo invece ha una grande valenza, per quanto riguarda soprattutto il round-off

Osservazione: Che si fa quando una di queste norme è nulla?

Penso comunque andare avanti con l'algoritmo, l'unica differenza è che non mi creerebbe una base, cioè:

$$\text{Se } \exists j : \hat{q}_j = 0 \Rightarrow \{q_1, q_2, \dots, q_{j-1}, *, q_{j+1}, \dots, q_m\}$$

Sono vettori linearmente dipendenti che mi generano uno spazio vettoriale di dimensione $m - 1$ (cioè ci sono $m - 1$ vettori linearmente indipendenti), quindi A non ha rango massimo

Osservazione Importante: L'algoritmo classico di Gram-Schmidt **non** fornisce una base ortonormale in aritmetica e precisione finita:

$$\hat{q}_j = a_j - q_1 q_1^T a_j - q_2 q_2^T a_j - \cdots - q_{j-1} q_{j-1}^T a_j \quad \text{Crea tantissimi problemi di round-off}$$

Metodo di Riflessione di Householder

Definizione di Matrice di Riflessione di Householder

Dato un vettore $v \in \mathbb{R}^n$ non nullo, che verrà chiamato vettore di Householder, la matrice di riflessione di Householder è definita come:

$$P = I_n - \beta v^T v \quad \text{con } \beta = \frac{2}{v^T v} = \frac{2}{\|v\|^2} \quad P \in \mathbb{R}^{n \times n}$$

Nota: Se β fosse stato la metà di come è definito, cioè se:

$$\beta = \frac{1}{\|v\|^2} \Rightarrow P = I - \frac{vv^T}{v^T v} = I - \frac{v}{\|v\|} \frac{v^T}{\|v\|}$$

Sarebbe stata una matrice di proiezione e avrebbe proiettato un vettore nell'ortogonale di v

Nonostante siano così vicini, hanno delle proprietà completamente diverse (basta pensare al fatto che una è singolare, l'altra è non singolare ortogonale)

Proprietà di P

1. P è simmetrica
2. P è ortogonale:

$$PP^T = P^2 = (I - \beta vv^T)(I - \beta vv^T) = I - 2\beta vv^T + \beta^2 vv^T vv^T = I - \frac{4vv^T}{v^T v} + \frac{4}{(v^T v)^2} v(v^T v)v^T = I$$

3. P è una riflessione rispetto a $(Span\{v\})^\perp$:

$$x \in \mathbb{R}^n : x = \gamma v + x_2 \Rightarrow Px = (I - \beta vv^T)(\gamma v + x_2) = \gamma v + x_2 - \gamma \beta v \underbrace{v^T v}_{\beta/2} - \gamma \beta v v^T x_2 = x_2 - \gamma v$$

4. Per una scelta opportuna di v vettore, dato un vettore x (con v scelto di conseguenza) si ha che:

$$Px = \beta vv^T x = \begin{pmatrix} * \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Inoltre questa matrice mantiene la norma, infatti se per una opportuna scelta di v abbiamo che $Px = \alpha e_1$, allora α è proprio la norma di x

Andiamo ora a capire che è v :

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

Tutto questo serve per aggiustare il tiro, in modo che la somma nella definizione di v sia con numeri dello stesso segno, sennò ci sono problemi di arrotondamento. *Secondo questa logica, potremmo anche impostare che x e $-\alpha$ hanno lo stesso segno*

Quindi v deve soddisfare la seguente relazione:

$$Px = \left(I - \frac{2}{v^T v} vv^T \right) x = x - \frac{2}{v^T v} v^T xv$$

Andiamo a fare un paio di sostituzioni:

$$\begin{aligned} v^T x &= (x - \alpha e_1)^T x = \|x\|^2 - \alpha x_1 \\ v^T v &= (x - \alpha e_1)^T (x - \alpha e_1) = \|x\|^2 - 2\alpha x_1 + \alpha^2 \xrightarrow{\|x\|^2 = \alpha^2} 2\|x\|^2 - 2\alpha x_1 = 2(\|x\|^2 - \alpha x_1) \end{aligned}$$

Da questo otteniamo che:

$$Px = x - \frac{2}{v^T v} v^T xv = x - \frac{2(\|x\|^2 - \alpha x_1)}{2(\|x\|^2 - \alpha x_1)} v = x - v = x - x + \alpha e_1 = \alpha e_1$$

La matrice di Householder ci permette di fare la fattorizzazione QR molto più velocemente

Applicazione delle matrici di Householder per QR

Come riferimento ci saranno delle dimensioni piccole, ma la notazione è per il caso generale

Supponiamo di avere una matrice $A \in \mathbb{R}^{n \times m}$ con $n \geq m$, cioè supponiamo di avere una matrice alta. Con il metodo di Householder vogliamo azzerare tutti gli elementi sotto alla diagonale principale, cioè:

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ \color{blue}{a_{2,1}} & a_{2,2} & a_{2,3} \\ \color{blue}{a_{3,1}} & \color{blue}{a_{3,2}} & a_{3,3} \\ \color{blue}{a_{4,1}} & \color{blue}{a_{4,2}} & a_{4,3} \end{pmatrix} \quad \text{Gli elementi blu sono quelli che vogliamo azzerare}$$

Lavoriamo colonna per colonna, quindi vogliamo azzerare gli elementi $a_{2,1}, a_{3,1}, \dots, a_{n,1}$

Andiamo a definire la matrice $P_1 \in \mathbb{R}^{n \times n}$ come:

$$P_1 = I_n - \beta_1 v_1 v_1^T \quad \text{con } v_1 = a_1 - \alpha e_1 \quad \text{con } a_1 \text{ la prima colonna di A}$$

Andando a fare i conti abbiamo che:

$$P_1 A = \begin{pmatrix} \alpha_1 & \times & \times \\ 0 & \color{green}{\times} & \times \\ 0 & \color{green}{\times} & \times \\ 0 & \color{green}{\times} & \times \end{pmatrix}$$

Per poter ancora applicare le matrici di Householder, andiamo a restringerci al blocco $\hat{A}^{(2)} \in \mathbb{R}^{(n-1) \times (m-1)}$ che otteniamo togliendo la prima riga e la prima colonna di A . Andiamo a definire come \hat{a}_2 la prima colonna di questo blocco (*in verde in A*) e andiamo ad applicare Householder.

Andiamo quindi a costruire la matrice:

$$\hat{P}_2 = I_{n-1} - \beta_2 v_2 v_2^T \quad \text{con } v_2 = \hat{a}_2 - \alpha_2 \hat{e}_1 \quad \hat{e}_1 \in \mathbb{R}^{n-1}$$

Visto che, dimensionalmente parlando, i conti non tornano, andiamo a definire $P_2 \in \mathbb{R}^n$ come:

$$P_2 = \begin{pmatrix} I_2 \\ \hat{P}_2 \end{pmatrix}$$

Abbiamo quindi che:

$$P_2 P_1 A = \begin{pmatrix} \alpha_1 & \times & \times \\ 0 & \alpha_2 & \times \\ 0 & 0 & \times \\ 0 & 0 & \times \end{pmatrix}$$

In maniera del tutto analoga, restringiamoci a $\hat{A}_3 \in \mathbb{R}^{(n-2) \times (m-2)}$ ottenuta togliendo le prime due righe e le prime due colonne.

Quindi possiamo costruire la matrice $\hat{P}_3 \in \mathbb{R}^{(n-2) \times (n-2)}$ come:

$$\hat{P}_3 = I_{n-2} - \beta_3 v_3 v_3^T \quad \text{con } v_3 = \hat{a}_3 - \alpha_3 \hat{e}_1 \quad \hat{e}_1 \in \mathbb{R}^{n-2}$$

Sempre per questioni di dimensioni adiamo creare $P_3 \in \mathbb{R}^n$ come:

$$P = \begin{pmatrix} I_2 & \\ & \hat{P}_3 \end{pmatrix}$$

Da cui otteniamo che

$$P_3 P_2 P_1 A = \begin{pmatrix} \alpha_1 & \times & \times \\ 0 & \alpha_2 & \times \\ 0 & 0 & \alpha_3 \\ 0 & 0 & 0 \end{pmatrix}$$

Notiamo che per ogni i , Q_i è una matrice ortogonale, quindi anche

$$P_m P_{m-1} \cdots P_2 P_1 \text{ è una matrice ortogonale}$$

Inoltre abbiamo che facendo il prodotto tra questa matrice e A restituisce una matrice triangolare superiore. Quindi abbiamo effettivamente trovato una fattorizzazione QR , in quanto:

$$\underbrace{P_m P_{m-1} \cdots P_2 P_1}_{Q^T} A = \begin{pmatrix} R_1 \\ 0 \end{pmatrix} \Rightarrow A = QR$$

Vediamo ora il costo computazionale:

Per ogni j colonna, dobbiamo creare v_j , creare β_j e poi applicare \hat{P}_h a tutta la matrice. Ad essere sinceri, non è esattamente ad ogni colonna, ma alle $m-j$ colonne rimanenti (*è questa la parte più costosa*)

Vediamo i costi uno per volta:

- v_j : Sappiamo che per la creazione di v_j dobbiamo fare: $\hat{a}_j - \alpha_j \hat{e}_1$, quindi il costo principale è dato da $\|a_j\|$ che ha un costo di $2(n-j)$ flops
- β_j : Per il calcolo di β_j dobbiamo fare: $\frac{2}{\|v_j\|^2}$ che ha un costo di $2(n-j)$
- \hat{P}_j : Per il costo di \hat{P}_j dobbiamo effettivamente costruire una matrice:

$$\hat{P}_j = I_{n-j} - \beta_j v_j v_j^T \Rightarrow \hat{P}_j x = x - \beta_j v_j v_j^T$$

La creazione di questa matrice ha un costo di $(4-j+1) \approx 4(n-j)$

Se andiamo a calcolare per le $m-j$ colonne rimanenti abbiamo che il costo è $4(n-j+1)(m-j)$

Facendo i calcoli in totale abbiamo che il costo totale è di:

$$\sum_{j=1}^m (n-j+1)(m-j) + 4(n-j) \approx 2nm^2 - \frac{2}{3}m^3 + \Theta(nm^2, n^3)$$

Quindi possiamo dire che, se $n \gg m$ allora il costo si approssima ancora di più a $2nm^2$

Osservazione: In matlab, per la fattorizzazione QR , possiamo scrivere `[Q,R] = qr(A)` per quella standard, altrimenti, per quella ridotta, `[Q,R] = qr(A,0)`, dove $Q = Q_1$ e $R = R_1$

Metodo per le matrici di Hessenberg

Esiste un altro tipo di fattorizzazione che è quella di Hessenberg che, data una matrice $A \in \mathbb{R}^n$ e sfruttando la riflessione di Householder:

$$A = QHQ^T \quad \text{con } Q \text{ ortogonale e } H \text{ Hessenberg superiore}$$

Definizione di Matrice Hessenberg Superiore

Una matrice quadrata si dice che è Hessenberg superiore se è una matrice triangolare superiore con la prima diagonale sotto quella principale non nulla

Andiamo a vedere come funziona. Negli esempi useremo una matrice 4×4 , ma le notazioni saranno per quelle generiche

Sia $A \in \mathbb{R}^{n \times n}$ una matrice quadrata, vogliamo applicare il metodo di Hessenberg.

Come prima cosa vogliamo azzerare gli elementi che stanno due posizioni sotto la diagonale principale della prima riga, ossia:

$$A = \begin{pmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \textcolor{blue}{\times} & \times & \times & \times \\ \textcolor{blue}{\times} & \times & \times & \times \end{pmatrix} \quad \text{Vogliamo che } \textcolor{blue}{\times} = 0$$

Quello che vogliamo fare è trovare una matrice P_1 tale che:

$$P_1 = \begin{pmatrix} 1 & \\ & \hat{P}_1 \end{pmatrix} \quad \Rightarrow \quad P_1 A = \begin{pmatrix} \times & \times & \times & \times \\ * & * & * & * \\ 0 & * & * & * \\ 0 & * & * & * \end{pmatrix}$$

Una volta trovata la applichiamo anche a destra, ottenendo:

$$P_1 A P_1 = \begin{pmatrix} \times & \square & \square & \square \\ * & \square & \square & \square \\ 0 & \square & \square & \square \\ 0 & \textcolor{blue}{\square} & \square & \square \end{pmatrix}$$

Adesso come prima vogliamo annullare l'elemento blu, possiamo fare come prima, quindi possiamo creare una matrice P_2 :

$$P_2 P_1 A P_1 P_2 = \begin{pmatrix} I_2 & \\ & \hat{P}_2 \end{pmatrix} \begin{pmatrix} \times & \square & \square & \square \\ * & \square & \square & \square \\ 0 & \square & \square & \square \\ 0 & \square & \square & \square \end{pmatrix} \begin{pmatrix} I_2 & \\ & \hat{P}_2 \end{pmatrix} = \begin{pmatrix} \times & \square & \circ & \circ \\ * & \square & \circ & \circ \\ 0 & \circ & \circ & \circ \\ 0 & 0 & \circ & \circ \end{pmatrix}$$

La matrice che abbiamo ottenuto è esattamente Hessemberg Superiore.

Chiamiamo H la matrice Hessenberg Superiore che abbiamo appena ottenuto, allora abbiamo che:

$$P_n P_{n-1} \cdots P_2 P_1 A P_1 P_2 \cdots P_{n-1} P_n$$

Sapendo che le matrici di Householder sono simmetricheabbiamo che:

$$\begin{aligned} H &= P_n P_{n-1} \cdots P_2 P_1 A P_1^T P_2^T \cdots P_{n-1}^T P_n \Rightarrow A = P_1^T P_2^T \cdots P_{n-1}^T P_n^T H P_n P_{n-1} \\ &= \underbrace{(P_n P_{n-1} \cdots P_2 P_1)}_Q^T \underbrace{H P_n P_{n-1} \cdots P_2 P_1}_{Q^T} = Q H Q^T \end{aligned}$$

A cosa ci serve questa fattorizzazione se abbiamo già quella di Householder?

Risulta estremamente comoda in quanto abbiamo che vengono preservati gli autovalori della matrice A , cioè

$$\text{Spec}(A) = \text{Spec}(H)$$

Rotazione di Givens

In questo caso siamo in \mathbb{R}^2

Dobbiamo trovare una rotazione che porta il vettore:

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \mapsto \begin{pmatrix} \|x\| \\ 0 \end{pmatrix}$$

Possiamo utilizzare la matrice di Rotazione definita in questo modo:

Definizione di Matrice di Givens / Matrice di Rotazione

In $\mathbb{R}^{2 \times 2}$, si definisce matrice di rotazione la matrice:

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

La matrice G di Given è una matrice ortogonale, infatti:

$$Gx = (\|x\| \ 0) \quad \Rightarrow \quad \|Gx\| = \left\| \begin{pmatrix} \|x\| \\ 0 \end{pmatrix} \right\| = \|x\|$$

Per la determinazione di θ possiamo porre:

$$\begin{cases} \cos \theta = \frac{x_1}{\|x\|} \\ \sin \theta = \frac{x_2}{\|x\|} \end{cases} \Rightarrow \begin{pmatrix} \frac{x_1}{\|x\|} & \frac{x_2}{\|x\|} \\ -\frac{x_2}{\|x\|} & \frac{x_1}{\|x\|} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \frac{x_1^2 + x_2^2}{\|x\|} \\ \frac{-x_1 x_2 + x_1 x_2}{\|x\|} \end{pmatrix} = \begin{pmatrix} \|x\| \\ 0 \end{pmatrix}$$

Andiamo a vedere nel dettaglio come funziona la fattorizzazione QR sfruttando le rotazioni di Givens,
Come nei casi precedenti utilizziamo una matrice piccola, ma useremo le notazioni per i casi generali
Sia $A \in \mathbb{R}^{n \times m}$ con $n \geq m$, cioè:

$$A = \begin{pmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \textcolor{blue}{\times} & \times & \times \end{pmatrix}$$

Andiamo ad azzerare l'elemento blu. Utilizzando le rotazioni di Givens, otteniamo che:

$$\begin{pmatrix} I_{n-2} & \\ & G_1 \end{pmatrix} A = \begin{pmatrix} \times & \times & \times \\ \times & \times & \times \\ * & * & * \\ 0 & * & * \end{pmatrix}$$

Quest'operazione ha un costo di 6 flops.

Poi possiamo andare avanti con:

$$\begin{pmatrix} I_{n-3} & & \\ & G_1 & \\ & & 1 \end{pmatrix} \begin{pmatrix} I_{n-2} & \\ & G_1 \end{pmatrix} A = \begin{pmatrix} \times & \times & \times \\ * & * & * \\ 0 & * & * \\ 0 & * & * \end{pmatrix}$$

Per evitare di avere danni con gli altri elementi, ci conviene fare prima tutta la prima colonna, e poi passare alle altre

Osservazione: Utilizzare le rotazioni di Givens è estremamente costoso, però è perfetta per quando si hanno pochi elementi non nulli

Esempi delle Rotazioni di Givens

Azzera gli elementi sotto la diagonale principale delle seguenti matrici:

- Sia $A \in \mathbb{R}^{5 \times 4}$:

$$A = \begin{pmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \\ 0 & 0 & 0 & 0 \end{pmatrix} \Rightarrow \begin{pmatrix} G_1 & \\ & I_3 \end{pmatrix} \begin{pmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \\ 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

- Sia $B \in \mathbb{R}^{5 \times 4}$:

$$\begin{pmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ \times & 0 & \times & \times \\ 0 & 0 & 0 & \times \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

In questo caso abbiamo due sistemi per risolvere quest'esercizio, il primo è quello di fare la moltiplicazione:

$$\begin{pmatrix} 1 & & \\ & G_1 & \\ & & I_2 \end{pmatrix} \begin{pmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ \times & 0 & \times & \times \\ 0 & 0 & 0 & \times \\ 0 & 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \\ 0 & 0 & 0 & 0 \end{pmatrix} \text{ e fare come prima}$$

Oppure possiamo fare direttamente il prodotto:

$$\begin{pmatrix} \cos \theta & 0 & \sin \theta & \\ 0 & 1 & 0 & \\ -\sin \theta & 0 & \cos \theta & \\ & & & I_2 \end{pmatrix} \begin{pmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ \times & 0 & \times & \times \\ 0 & 0 & 0 & \times \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

In questo caso bisogna però fare attenzione all'elemento in posizione 3,2, che dopo il prodotto, difficilmente sarà nullo

Il costo computazionale delle rotazioni di Givens è molto simile al metodo di Householder

Problema dei Minimi Quadrati

Data una matrice $A \in \mathbb{R}^{n \times m}$, con $n >> m$ (*data A una matrice alta*) e dato $b \in \mathbb{R}^n$, vogliamo risolvere il sistema lineare $Ax = b$:

Differentemente da prima, abbiamo che la matrice è alta, quindi il vettore x è di dimensione minore rispetto al vettore b , quindi abbiamo un problema sovradeterminato (cioè la x deve risolvere tante uguaglianze)

Differentemente da quanta facevamo in precedenza, non è ovvio che ci sia una soluzione

Proprio per questi motivi, in questi casi ci possiamo accontentare di una approssimazione della soluzione, cioè vogliamo trovare la x che più si avvicini alla soluzione finale:

$$\min_{x \in \mathbb{R}^m} \|b - Ax\|_2$$

Cioè vogliamo che questa distanza dall'origine sia minima

Osservazione: Iniziamo a gestire il problema da un'*ottica diversa*: al posto di guardare il problema come $Ax = b$, in cui vogliamo trovare una x che risolva il problema, poniamo il problema come $b = Ax$, cioè vogliamo che la b sia uguale (o molto vicina) ad Ax , con la scelta da parte nostra di x . Cioè vogliamo trovare una x tale che b sia una combinazione lineare delle colonne di A :

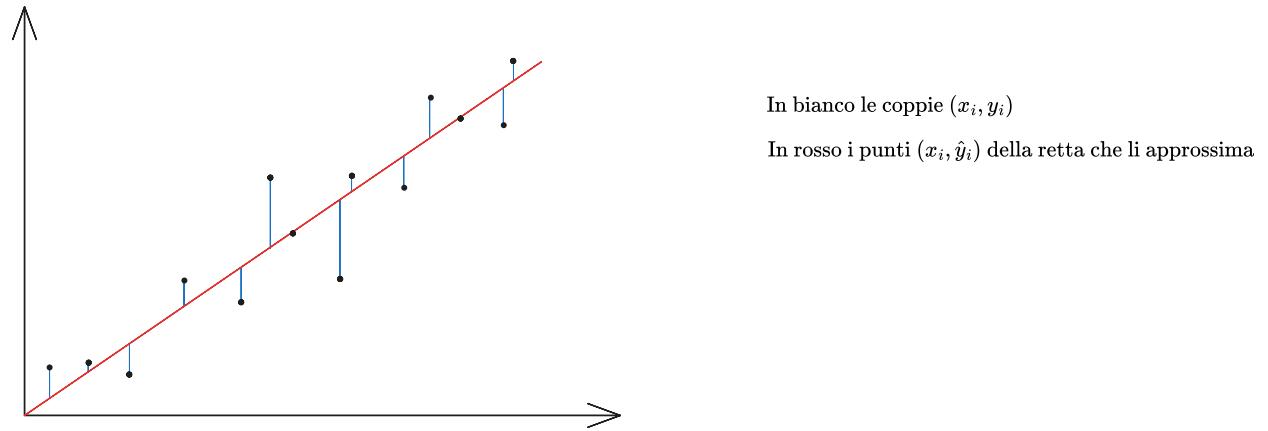
$$b = (a_1 \quad a_2 \quad \cdots \quad a_m) \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

Se minimizziamo questo residuo, otteniamo una combinazione lineare che meglio approssima b

Esempi: Sfruttano questo principio l'Image Recognition, il Text mining e il riconoscimento dell'IA per i numeri scritti a mano

Esempio Concreto del Problema dei Minimi Quadrati

Un esempio cardine è quello della Regressione Lineare:



Siano quindi (x_i, y_i) dei punti nel piano, con $i \in \{1, \dots, n\}$ e vogliamo sapere se c'è una dipendenza lineare tra le x e le y

Vogliamo sapere quindi se c'è una retta che meglio approssima i punti del piano. Non ci serve sapere che sia perfetta, basta che si avvicini il più possibile

Quindi vogliamo trovare una retta $y = mx + q$ che meglio approssimi questi punti

Nel fare ciò dobbiamo porre che la distanza di questi punti dalla retta sia la più piccola possibile, cioè:

$$\forall i \in \{1, \dots, n\} \quad |y_i - \hat{y}_i| \text{ sia la più piccola possibile}$$

Possiamo ragionare anche in termini di vettori, quindi:

$$\min_{m,q \in \mathbb{R}} \|y - \hat{y}\|_2^2 \quad \text{con } y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \text{ e } \hat{y} = \begin{pmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_n \end{pmatrix}$$

Vogliamo minimizzare quella distanza (sono somme di quadrati, per questo il nome)
Andiamo a sviluppare quello che vogliamo trovare:

$$\|y - \hat{y}\|_2^2 = \|y - (mx + q\mathbf{1})\|_2^2 = \left\| \underbrace{y}_b - (\underbrace{\mathbf{1}}_A \underbrace{x}_x) \begin{pmatrix} q \\ m \end{pmatrix} \right\|_2^2 = \|b - Ax\|_2^2$$

Per chiarezza, y, \hat{y}, x sono vettori, $\mathbf{1}$ è un vettore di tutti 1, q, m sono degli scalari

Se avessimo avuto più variabili, cioè ci fossimo trovati in $\mathbb{R}^n, n > 2$, allora la matrice A avrebbe avuto tante più colonne

Esistenza e Unicità della Soluzione

Lemma

Sia $A \in \mathbb{R}^{n \times m}$ con $n \geq m$, allora ha rango massimo (m) se e solo se $A^T A$ non è singolare

Dimostrazione:

Consideriamo la fattorizzazione ridotta QR di $A = Q_1 R_1$

Allora A ha rango massimo se e solo se A ha m colonne linearmente indipendenti, che è vero se e solo se R_1 è non singolare.
In particolare, sapendo che Q_1 è ortogonale:

$$A^T A = R_1^T Q_1^T Q_1 R_1 = R_1^T R_1$$

Ma poiché R_1 è non singolare, abbiamo che $R_1^T R_1 \geq 0$, da cui segue che $A^T A \geq 0$

Infatti abbiamo che, per ogni vettore x non nullo:

$$x^T A^T A x = \|Ax\|^2 \geq 0 \Leftrightarrow x^T R_1^T R_1 x = \|R_1 x\|^2 \geq 0$$

Abbiamo che tale norma è sempre nulla se R_1 è singolare, quindi è sempre strettamente positiva se R_1 è non singolare, ma se

$$R_1^T R_1 > 0 \Rightarrow A^T A > 0$$

□

Teorema

Sia X l'insieme dei vettori $x \in \mathbb{R}^m$ soluzioni del problema

$$\min_{x \in \mathbb{R}^m} \|b - Ax\|_2^2$$

Allora valgono:

1. $x \in X$ se e solo se x è soluzione di $A^T Ax = A^T b$ (definita come equazione normale)
2. X ha un solo elemento (cioè la soluzione è unica) se e solo se A ha rango massimo

Dimostrazione:

1) Definiamo R come:

$$R = Range(A) = Im(A) = \{y \in \mathbb{R}^n : \exists x \in \mathbb{R}^m : y = Ax\}$$

Definiamo poi R^\perp come:

$$R^\perp = \{z \in \mathbb{R}^n : z \perp y, y \in R\} = \{z \in \mathbb{R}^n : z^T y = 0\}$$

Allora possiamo riscrivere $b = b_1 + b_2$ in modo che $b_1 \in R$ e $b_2 \in R^\perp$. Otteniamo quindi che:

$$b - Ax = b_1 + b_2 - Ax = \underbrace{b_2}_{\in R^\perp} + \underbrace{(b_1 - Ax)}_{\in R} \Rightarrow b_2^T (b_1 - Ax) = 0$$

Con le norme otteniamo che:

$$\begin{aligned}\|b - Ax\|^2 &= (b - Ax)^T(b - Ax) = (b_2 + b_1 - Ax)^T(b_2 + b_1 - Ax) = \\ &= b_2^T b_2 + \underbrace{b_2^T(b_1 - Ax)}_0 + \underbrace{(b_1 - Ax)^T b_2}_0 + (b_1 - Ax)^T(b_1 - Ax) = \|b_2\|^2 + \|b_1 - Ax\|^2\end{aligned}$$

Va sottolineato che se non mettiamo i quadrati quest'uguaglianza non è vera

Torniamo adesso al problema del minimo quadrato e otteniamo che:

$$\min_{x \in \mathbb{R}^n} \|b - Ax\|^2 = \min_{x \in \mathbb{R}^n} (\|b_2\|^2 + \|b_1 - Ax\|^2) = \|b_2\|^2 + \min_{x \in \mathbb{R}^n} \|b_1 - Ax\|^2$$

Notiamo che la parte dentro il minimo è contenuto è contenuta in R quindi possiamo scegliere x tale che:

$$Ax = b_1 \quad \Rightarrow \quad \min_{x \in \mathbb{R}^m} \|b_1 - Ax\|^2 = 0$$

Quindi otteniamo che:

$$\min_{x \in \mathbb{R}^m} \|b - Ax\|^2 = \|b_2\|^2$$

Quindi otteniamo che:

$$r = b - Ax = b_1 + b_2 - Ax = b_2 \in R^\perp$$

Quindi il residuo r è ortogonale a tutti i vettori v in R , tra cui le stesse colonne di A :

$$A = \begin{pmatrix} \underline{a}_1 & \underline{a}_2 & \cdots & \underline{a}_m \end{pmatrix} \quad \Rightarrow \quad r^T \underline{a}_i = 0 \quad \text{con } i \in \{1, \dots, m\} \quad \Rightarrow \quad A^T r = 0$$

Da cui otteniamo che

$$A^T(b - Ax) = 0 \quad \Leftrightarrow \quad A^A x = A^T b$$

Quindi l'equazione normale è verificata.

2) La soluzione è unica se e solo se esiste un'unica soluzione alla soluzione normale, cioè

$$\exists! x \in \mathbb{R}^n : A^T A x = A^T b$$

E questo è vero se e solo se $A^T A$ è non singolare, e per il lemma sopra sappiamo che A rango massimo

□

Da ora in poi supponiamo sempre che A abbia rango massimo.

In caso non avesse rango massimo, allora potremmo accontentarci di avere la norma minima e poter fare altro ancora, ma la cosa non interessa questo corso, quindi ci limitiamo al caso di A con rango massimo

Osservazione: Come si risolve un'equazione normale? $A^T A x = A^T b$

Con Cholesky possiamo trasformare $A^T A = \hat{L} \hat{L}^T$

Pensiamo al momento al costo computazionale, in generale in entrambi i casi è molto elevato.

Nel caso di $A^T A$, il prodotto è pari a $2n$, ma questo va fatto per tutte le colonne di A , quindi m volte, e poi va rifatto ancora per tutte le righe, quindi ha un costo di $(2n \text{ flops}) \cdot n \cdot m = \Theta(nm^2)$

Se invece facciamo con Cholesky abbiamo che il costo è pari a $\Theta(m^3)$

Poi bisogna anche considerare la determinazione del termine noto che è pari a $2mn$ flops in entrambi i casi.

La sola determinazione del problema ha un costo elevato, ma poi saltano fuori anche problemi con la stabilità

Esempio di Instabilità dell'Equazione Normale

Siano $A \in \mathbb{R}^{3 \times 2}$ e $b \in \mathbb{R}^n$:

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

Supponiamo di avere $\alpha \in \mathbb{R}$ tale che:

$$\varepsilon < \alpha < \sqrt{\varepsilon} \quad \text{con } \varepsilon \text{ il valore } \texttt{eps} \text{ della macchina}$$

In aritmetica esatta avremmo che:

$$A^T A = \begin{pmatrix} 3 & 4 & 0 \\ 3 & 4 & \alpha \\ 0 & \alpha & 1 \end{pmatrix} \begin{pmatrix} 3 & 3 \\ 4 & 4 \\ 0 & \alpha \end{pmatrix} = \begin{pmatrix} 25 & 25 \\ 25 & 25 + \alpha^2 \end{pmatrix}$$

Ossia le colonne di $A^T A$ sono linearmente indipendenti.

Tuttavia, se dovessimo fare $\text{fl}(A^T A)$ otterremo che:

$$\text{fl}(A^T A) = \begin{pmatrix} 25 & 25 \\ 25 & 25 \end{pmatrix} \quad \Rightarrow \quad \text{Range}(A^T A) = 1$$

Questo è un problema perché la matrice aveva le colonne linearmente indipendenti, mentre in macchina diventano linearmente dipendenti per la scelta di α , in quanto $\alpha^2 < \varepsilon$

La soluzione di questo sistema comunque è

$$x = \begin{pmatrix} \frac{7}{25} - \frac{1}{\alpha} \\ \frac{1}{\alpha} \end{pmatrix}$$

Vediamo come poter usare la fattorizzazione QR :

$$A = QR = (Q_1 \quad Q_2)(R_1 0) = Q_1 R_1$$

Con le norme al quadrato otteniamo che:

$$\begin{aligned} \|b - Ax\|^2 &= \|b - QRx\|^2 = \|QQ^T b - QRx\|^2 = \|Q(Q^T b - Rx)\|^2 \xrightarrow{\|Q\|=1} \|Q^T b - Rx\|^2 = \\ &= \left\| \begin{pmatrix} Q_1^T \\ Q_2^T \end{pmatrix} b - \begin{pmatrix} R_1 \\ 0 \end{pmatrix} x \right\|^2 = \left\| \begin{pmatrix} Q_1^T b - R_1 x \\ Q_2^T b \end{pmatrix} \right\|^2 \end{aligned}$$

Sapendo che, dati v_1, v_2 vettori:

$$\left\| \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \right\|^2 = \|v_1\|^2 + \|v_2\|^2$$

Otteniamo che:

$$\|b - Ax\|^2 = \left\| \begin{pmatrix} Q_1^T b - R_1 x \\ Q_2^T b \end{pmatrix} \right\|^2 = \|Q_1 b - R_1 x\|^2 + \|Q_2^T b\|^2$$

Adesso andiamo a calcolare il minimo:

$$\min_{x \in \mathbb{R}^m} \|b - Ax\|^2 = \min_{x \in \mathbb{R}^m} (\|Q_1^T b - R_1 x\|^2 + \|Q_2^T b\|^2)$$

Otteniamo il minimo se e solo se:

$$Q_1^T b - R_1 x = 0 \quad \Leftrightarrow \quad R_1 x = Q_1^T b \quad \Leftrightarrow \quad x = R_1^{-1} Q_1^T b$$

Possiamo nell'effettivo calcolare l'inversa di R_1 in quanto avevamo supposto, A rango massimo, quindi $A^T A$ non singolare, quindi R_1 non singolare

Osservazione: $Q^T b$ può essere calcolato durante la generazione della fattorizzazione QR , infatti con Householder:

$$P_m P_{m-1} \cdots P_2 P_1 (A - b) = Q^T (A - b) = (R_1 \quad Q^T b) = \begin{pmatrix} R_1 & Q_1^T b \\ 0 & Q_2^T b \end{pmatrix}$$

In maniera gratuita otteniamo quindi quello che ci serve per risolvere il problema dei minimi quadrati

Esempio di Instabilità dell'Equazione Normalae parte 2

Riprendiamo l'esempio precedente

Con la fattorizzazione QR otteniamo che:

$$A = \begin{pmatrix} 3 & 3 \\ 4 & 4 \\ 0 & \alpha \end{pmatrix} = \begin{pmatrix} \frac{3}{5} & 0 \\ \frac{4}{5} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 5 & 5 \\ 0 & \alpha \end{pmatrix} \quad \Rightarrow \quad \kappa_F(R_1) = \|R_1\|_F \cdot \|R_1^{-1}\|_F = \Theta\left(\frac{1}{\alpha}\right)$$

Osservazione: Notiamo che la fattorizzazione QR va a toccare manualmente quanto visto nel teorema precedente:

$$b = \underbrace{(I - Q_1 Q_1^T)b}_{\in R^\perp} + \underbrace{Q_1 Q_1^T b}_{\in R}$$

Cioè stiamo scomponendo b in $b_1 \in R$ e $b_2 \in R^\perp$

Osserviamo che $I - Q_1 Q_1^T = Q_2 Q_2^T$, infatti:

$$I = QQ^T = (Q_1 \quad Q_2) \begin{pmatrix} Q_1^T \\ Q_2^T \end{pmatrix} = Q_1 Q_1^T + Q_2 Q_2^T \Rightarrow Q_2 Q_2^T = I - Q_1 Q_1^T$$

In questo modo otteniamo che:

$$b = \underbrace{Q_2 Q_2^T b}_{\in R^\perp} + \underbrace{Q_1 Q_1^T b}_{\in R} = b_2 + b_1$$

In questo modo otteniamo che, con le norme:

$$\|b_2\|^2 = \|Q_2 Q_2^T b\|^2 \stackrel{*}{=} \|Q_2^T b\|^2 = \|r\|^2$$

Dove c'è \star utilizziamo il fatto che:

$$\|Q_2 v\|^2 = v^T \underbrace{Q_2^T Q_2}_I v = v^T v = \|v\|^2$$

È esattamente quanto fatto nel teorema senza tuttavia toccare l'equazione normale

Osservazione: Per fare la fattorizzazione QR , possiamo moltiplicare anche il vettore b :

$$P_m \cdots P_2 P_1(A, b) = (R, \hat{b})$$

Cioè applico la trasformazione di A a matrice triangolare superiore anche a b

Questa è una variante di Householder, in cui aggiungiamo anche b , infatti:

$$\|b - Ax\| = \|b - QRx\| = \|Q^T b - Rx\|$$

In questo modo posso:

1. trovare la nuova soluzione \hat{b} senza dover costruire nuovamente la fattorizzazione QR
2. risolvere l'equazione senza dover memorizzare Q

Infatti in questo modo aggiorno A e b senza dover salvare Q . L'unico posto, infatti, in cui mi potrebbe poi tornare comoda è quando devo realizzare \hat{b}

Inoltre, siccome ho che:

$$R = \begin{pmatrix} R_1 \\ 0 \end{pmatrix} \quad \text{Posso trovare } x \text{ come } R_1 x = \hat{b}_{(1:m)}$$

È importante sottolineare le entrate che vogliamo, in quanto abbiamo che $\hat{b} \in \mathbb{R}^n$ ma abbiamo che $n \geq m$ e ce ne servono solamente m . Idealmente quello che facciamo è:

$$\hat{b} = Q^T b = Q_1^T b + Q_2^T b$$

e prendere solamente $Q_1^T b$ solo che noi non abbiamo bisogno di costruire Q

Sfruttando il fatto che R_1 è una matrice triangolare superiore, possiamo implementare l'algoritmo.

Il residuo sarà quindi:

$$\|r\| = \|\hat{b}_{(m+1:n)}\| = \|Q_2^T b\| = \|b_2\|$$

Solo idealmente, in quanto non vediamo Q_2 , visto che non abbiamo motivo e bisogno di costruire Q

Teorema

Il vettore $\hat{x} \in \mathbb{R}^m$ soluzione del problema di minimi quadrati ottenuto mediante fattorizzazione QR con matrici di Householder è tale che:

$$\hat{x} = \arg \left(\min_{x \in \mathbb{R}^m} \|(A + \delta A)x - (b + \delta b)\|_2 \right)$$

Dove si ha che:

$$\|\delta A\|_F \leq c(m, n)u \cdot \|A\|_F + \Theta(u^2) \quad \text{e} \quad \|\delta b\|_2 \leq c(m, n)u \cdot \|b\|_2 + \Theta(u^2)$$

Con $c(m, n) = (6n - 3m + 40)m = \Theta(nm, m^2)$

Considerazioni del Teorema: *Questo significa che la soluzione è un problema dei minimi quadrati vicino ad un problema di partenza. Quindi la soluzione con QR con Householder è molto buona perché è la soluzione esatta di un problema molto vicino, più vicine più $\|\delta A\|_F$ e $\|\delta b\|_2$ sono piccole*

Problema degli Autovalori

Problema: Data $A \in \mathbb{C}^{n \times n}$ determinare (λ, x) detta autocoppia di A , con $\lambda \in \mathbb{C}$ e $x \in \mathbb{C}^n$ non nullo tali che:

$$Ax = \lambda x$$

Anche se avessimo avuto A reale non sarebbe cambiato nulla, avremmo avuto λ e x complessi

Diciamo subito che il problema è molto sensibile alle perturbazioni.

Esempio di Perturbazione nel calcolo degli Autovalori

Sia J_0 il blocco di Jordan definito come:

$$\begin{pmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & 0 \end{pmatrix}$$

Sia poi J_ε la matrice definita nel seguente modo:

$$\begin{pmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ \varepsilon & & & 0 \end{pmatrix}$$

Andiamo a calcolare gli autovalori in entrambi i casi:

- Per J_0 abbiamo che:

$$\det(J_0 - \lambda I_n) = (-1)^n \lambda$$

Per cui l'unico autovalore è 0 molteplicità algebrica $m_a(\lambda) = n$

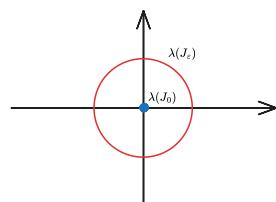
- Per J_ε abbiamo che:

$$\begin{aligned} \det(J_\varepsilon - \lambda I_n) &= \det \begin{pmatrix} -\lambda & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ \varepsilon & & & -\lambda \end{pmatrix} = (-\lambda) \det \begin{pmatrix} -\lambda & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & -\lambda \end{pmatrix} + (-1)^{n+1} \varepsilon \det \begin{pmatrix} 1 & & & \\ -\lambda & \ddots & & \\ & \ddots & \ddots & \\ & & -\lambda & 1 \end{pmatrix} \\ &= (-\lambda)^n + (-1)^{n+1} \varepsilon \cdot 1 = (-\lambda)^n + (-1)^{n+1} \varepsilon \end{aligned}$$

Ponendolo uguale a 0 e guardando i moduli abbiamo che:

$$|(-\lambda)^n| = |\varepsilon| \Rightarrow |\lambda| = \sqrt[n]{|\varepsilon|}$$

Hanno tutti modulo uguale, quindi gli autovalori di J_ε stanno tutti in una circonferenza di raggio $\sqrt[n]{\varepsilon}$



Per calcolare gli autovalori di una matrice si può utilizzare `eig(A)`

In questo capitolo vedremo due classi di metodi, ognuno con un intento ben preciso:

- Approssimazione di un'autocoppia

- Approssimazione di tutti gli autovalori

Prima di andare avanti facciamo un ripasso delle varie matrici in complesse:

- **Caso $A \in \mathbb{C}^{n \times n}$ Hermitiana:** In questo caso abbiamo che

$$A = A^H = \bar{A}^T \Rightarrow A = X\Lambda X^H$$

Nel caso di $A \in \mathbb{R}^{n \times n}$ avremmo che A è ortogonale e che $A = X\Lambda X^T$

In questo caso abbiamo che X è la matrice degli autovettori di A , quindi abbiamo che:

$$\forall i \in \{1, \dots, n\} \quad Ax_i = x_i \lambda \Leftrightarrow AX = X\Lambda \Leftrightarrow A = X\Lambda X^H$$

Nel caso in cui scriviamo $A = X^H \Lambda X$ intendiamo dire che gli autovettori sono le righe di X , non le colonne di X e in generale, quando parliamo di autovettori, parliamo di autovettori le colonne di X

- **Caso A Non Hermitiana ma Diagonalizzabile:** Con il caso reale A non simmetrica ma diagonalizzabile

In questo caso abbiamo una base \mathcal{B} di autovettori in \mathbb{C}^n :

$$A = X\Lambda X^{-1} \quad \text{Sempre per seguire la logica del } AX = X\Lambda$$

In particolare, se poniamo X come:

$$X = (\underline{x}_1, \dots, \underline{x}_n) \text{ con } \underline{x}_i \text{ autovettori} \Rightarrow X \text{ è invertibile ma non ortogonale}$$

Inoltre abbiamo che $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n) \in \mathbb{C}^{n \times n}$ con gli autovalori complessi. La stessa cosa vale anche per $A \in \mathbb{R}^{n \times n}$ non simmetrica, allora molto probabilmente gli autovalori saranno complessi-

Inoltre, sempre con matrici reali, se λ_i è autovalore, allora anche $\bar{\lambda}_i$ è autovalore

- **Caso A né Hermitiana né Diagonalizzabile:** In questo caso possiamo limitarci alle forme di Jordan:

$$A = XJX^{-1} \quad \text{con } J \text{ metrice diagonale a blocchi con blocchi di Jordan}$$

A livello di macchina è fondamentalmente inutile, perché sulla macchina non è possibile determinarlo facilmente

- **Stesso Caso di Prima:** Oltre a quanto fatto nel punto precedente, possiamo fare la fattorizzazione di Schur:

$$A = QRQ^H$$

Con $Q \in \mathbb{C}^{n \times n}$ matrice unitaria ($QQ^H = Q^HQ = I$) e $R \in \mathbb{C}^{n \times n}$ triangolare superiore con λ_i autovalori sulla diagonale principale

Questa fattorizzazione è sempre possibile e stabile in quanto usiamo principalmente trasposizioni e questo è il metodo che c'è dietro il comando `eig`

Ovviamente tutto questo sarà approssimato perché non troveremo mai il risultato perfetto

Osservazione: Se H è Hermitiana, allora

$$A = QRQ^H = X\Lambda X^H \Rightarrow R \text{ è diagonale e } Q = X$$

In generale Q non ha autovettori sulle colonne

Osservazione: Sia $A \in \mathbb{C}^{n \times n}$ e sia y autovettore sinistro, cioè:

$$y^H A = \lambda y^H$$

In questo modo otteniamo una tripletta (λ, x, y) dove x è autovettore destro e y è autovettore sinistro.

Supponiamo di avere A matrice diagonalizzabile, allora anche A^H è diagonalizzabile, allora esistono n autovettore sinistri di A :

$$y_i^H A = \lambda y_i^H \quad i \in \{1, \dots, n\}$$

In particolare, se andiamo a definire $Y = (\underline{y}_1, \dots, \underline{y}_n)$, otteniamo che è equivalente a:

$$Y^H A = \Lambda Y^H$$

In particolare, se A è diagonalizzabile, allora le righe di Y^H formano una base, quindi Y^H è non singolare, da cui:

$$A(Y^H)^{-1} = (Y^H)^{-1}\Lambda$$

Da tutto ciò otteniamo che $(Y^H)^{-1} = X$, quindi è la matrice degli autovettori sinistri, in particolare:

$$X = (Y^H)^{-1} \Leftrightarrow X^{-1} = Y^H$$

Questo è vero se abbiamo normalizzato le righe di Y^H in modo opportuno, cioè tali che $Y^H X = I$, cioè che per ogni i , $y_i^H x_i = 1$

Quindi abbiamo che:

$$A = X\Lambda X^{-1} = X\Lambda Y^H$$

Questa è la vera decomposizione spettrale.

Abbiamo inoltre che, andando a sviluppare quanto scritto sopra:

$$A = X\Lambda Y^H = \sum_{i=1}^n x_i \lambda_i y_i^H$$

Definizione di Matrice Normale

Una matrice $A \in \mathbb{C}^{n \times n}$ si dice normale se soddisfa $AA^H = A^H A$

Osservazione: Se A è Hermitiana, allora è normale, infatti se $A = A^H$, allora $AA^H = A^2 = A^H A$

Se A è Antihermitiana è ancora normale, infatti se $A = -A^H$, allora $AA^H = -A^2 = A^H A$

Se A è Ortonormale unitaria, allora è ancora normale, infatti $A^H A = I = AA^H$

Proposizione

$A \in \mathbb{C}^{n \times n}$ è normale se e solo se $A = Q\Lambda Q^H$ con Q unitaria e $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$ con $\lambda_i \in \mathbb{C}$ autovalori

Considerazioni della Proposizione: Questo ci dice che le matrici normali sono diagonalizzabili tramite trasformazioni unitarie. Inoltre l'unica differenza tra le matrici normali e quelle Hermitiane è il fatto che le matrici Hermitiane hanno autovalori reali, mentre quelle normali hanno autovalori complessi

Osservazione: Sia $U \in \mathbb{R}^{n \times n}$ ortogonale, allora

$$\text{Spec}(U) \subseteq \{z \in \mathbb{C} : |z| = 1\}$$

Infatti, se io ho un'autocoppia (λ, v) , allora:

$$Uv = \lambda v \Rightarrow \|Uv\| = \|\lambda v\| = |\lambda| \|v\| \Rightarrow |\lambda| = 1$$

Localizzazione e Perturbazione di Autovalori

Sappiamo dare informazioni sugli autovalori senza effettivamente calcolarli?

Così su due piedi sappiamo che se A è una matrice reale, allora gli autovalori sono simmetrici rispetto all'asse delle ascisse, mentre se è complessa possono essere ovunque.

Iniziamo da un piccolo risultato, ma comunque notevole:

Teorema di Hirsch

Sia $A \in \mathbb{C}^{n \times n}$ e sia $\|\cdot\|$ una norma matriciale indotta, allora:

$$\text{Spec}(A) \subseteq \{z \in \mathbb{C} : |z| \leq \|A\|\}$$

Dimostrazione:

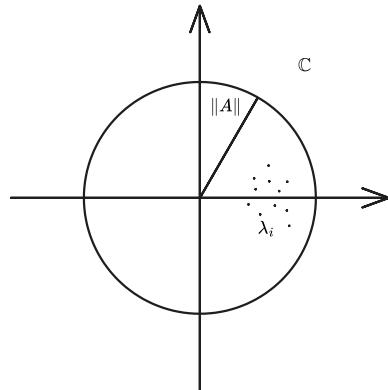
Sia (λ, x) un'autocoppia di A , allora:

$$Ax = \lambda x \Rightarrow \|\lambda x\| = \|Ax\|$$

In particolare abbiamo che:

$$|\lambda| \|x\| = \|\lambda x\| = \|Ax\| \leq \|A\| \cdot \|x\| \Rightarrow |\lambda| \leq \|A\|$$

□



Cerchiamo di affilare per bene quest'informazione:

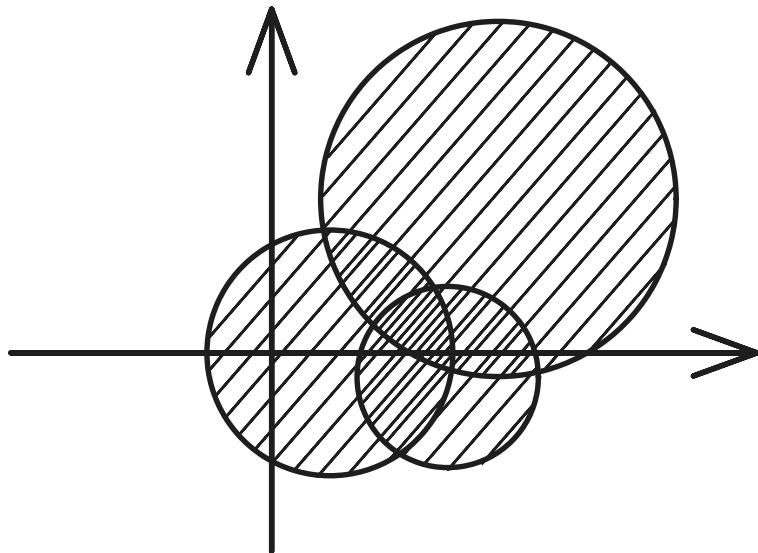
Definizione di Disco di Gerschgorin

Sia $A \in \mathbb{C}^{n \times n}$, si definisce Disco di Gerschgorin per le righe l'insieme:

$$\mathcal{G}_i^{(r)} = \left\{ z \in \mathbb{C} : |z - A_{i,i}| \leq \sum_{j=1, j \neq i}^n |A_{i,j}| \right\} \quad i \in \{1, \dots, n\}$$

In parole povere si ha che l'insieme $\mathcal{G}_i^{(r)}$ è una circonferenza nel piano complesso con centro in $A_{i,i}$ e con raggio somma degli altri elementi della riga escluso quello sulla diagonale principale

Quindi graficamente è una cosa del genere:



Primo Teorema di Gershgorin

Sia $A \in \mathbb{C}^{n \times n}$, allora:

$$\text{Spec}(A) \subseteq \bigcup_{i=1}^n \mathcal{G}_i^{(r)}$$

Dimostrazione:

Sia (γ, x) autocoppia tale che:

$$Ax = \lambda x$$

Consideriamo la i -esima riga:

$$\sum_{j=1}^n A_{i,j}x_j = \lambda x_i$$

Dove con x_i indichiamo la i -esima componente di x .

Questo è uguale a:

$$A_{i,i}x_i + \sum_{j=1, j \neq i}^n A_{i,j}x_j = \lambda x_i \quad \Rightarrow \quad (\lambda - A_{i,i})x_i = \sum_{j=1, j \neq i}^n A_{i,j}x_j$$

Scegliamo adesso \hat{i} in modo

$$\hat{i} = \arg \max_{i \in \{1, \dots, n\}} |x_i|$$

Quindi facendo il modulo in entrambi i lati abbiamo che:

$$|(\lambda - A_{i,\hat{i}})| \cdot |x_{\hat{i}}| \leq \left| \sum_{j=i, j \neq \hat{i}}^n A_{i,j}x_j \right| \leq \sum_{j=1, j \neq \hat{i}}^n |A_{i,j}| \cdot |x_j|$$

Dividiamo da entrambe le parti il fattore comune $|x_{\hat{i}}|$ e otteniamo che:

$$|(\lambda - A_{i,\hat{i}})| \leq \sum_{j=1, j \neq \hat{i}}^n |A_{i,j}| \cdot \underbrace{\frac{|x_j|}{|x_{\hat{i}}|}}_{< 1} \leq \sum_{j=1, j \neq \hat{i}}^n |A_{i,j}|$$

Per cui abbiamo che

$$\lambda \in \mathcal{G}_{\hat{i}}^{(r)}$$

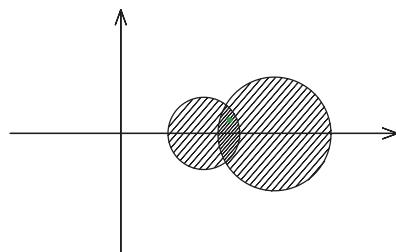
(Ma non sappiamo quale riga, però sappiamo che appartiene all'unione)

Non conoscendo \hat{i} a priori, possiamo solo concludere che:

$$\lambda \in \bigcup_{i=1}^n \mathcal{G}_i^{(r)}$$

□

Osservazione: Appartiene ad almeno un disco. infatti se $A \in \mathbb{R}^{n \times n}$ posso avere il caso in cui:



Osservazione: Il teorema vale anche per A^T , cioè

$$\text{Spec}(A) \subseteq \bigcup_{i=1}^n \mathcal{G}_i^{(r)}(A^T)$$

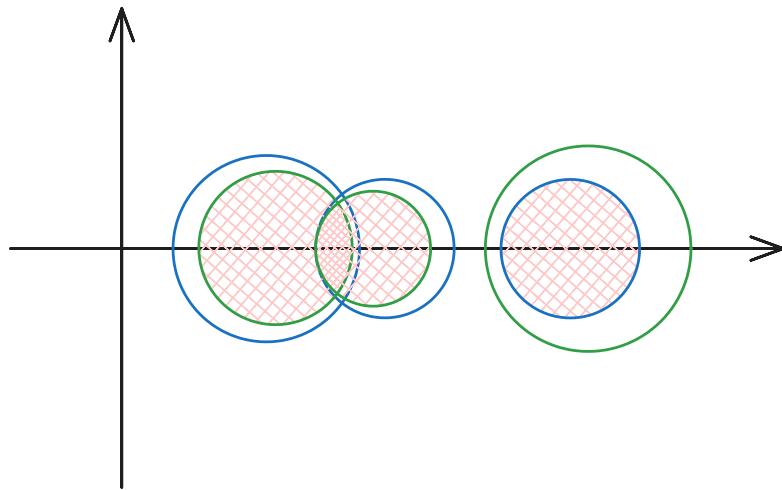
Ma calcolare i dischi di Gershgorin sulle righe di A^T corrisponde con il calcolare i dischi di Gershgorin sulle colonne di A , cioè:

$$\mathcal{G}_i^{(r)}(A^T) = \mathcal{G}_i^{(c)}(A)$$

In particolare abbiamo che:

$$\text{Spec}(A) = \left(\bigcup_{i=1}^n \mathcal{G}_i^{(r)} \right) \cap \left(\bigcup_{i=1}^n \mathcal{G}_i^{(c)} \right)$$

In questo modo prendiamo la regione più piccola possibile



Qui abbiamo che in blu abbiamo le righe, in verde le colonne e il rosso l'intersezione

Osservazione: Se $A \in \mathbb{R}^{n \times n}$ simmetrica, posso usare i dischi di Gershgorin per stimare l'intervallo spettrale

Esempio sulla stima dell'Intervallo Spettrale

Sia

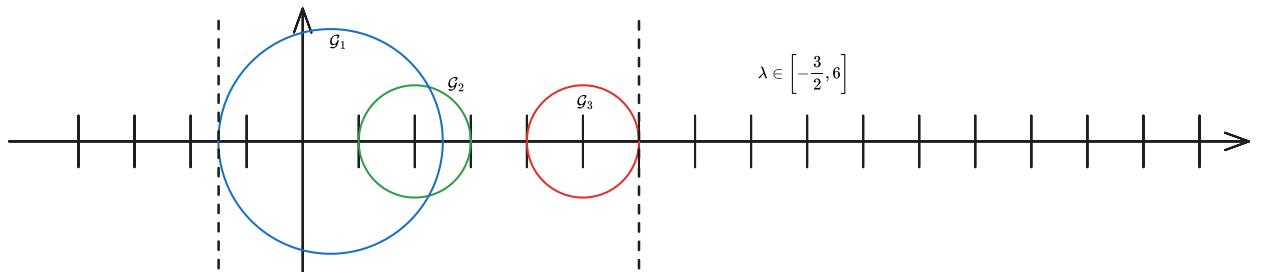
$$A = \begin{pmatrix} 2 & 1 & 0 \\ 1 & \frac{1}{2} & 1 \\ 0 & 1 & 5 \end{pmatrix}$$

Stimare l'intervallo spettrale

Con i dischi di Gershgorin abbiamo che:

$$\mathcal{G}_1^{(r)} = \mathcal{G}_1^{(c)} = \mathcal{G}_1 = \{ |z - 2| \leq 1 \} \quad \mathcal{G}_2 = \left\{ \left| z - \frac{1}{2} \right| \leq 2 \right\} \quad \mathcal{G}_3 = \{ |z - 5| \leq 1 \}$$

Graficamente abbiamo che:



Secondo Teorema di Gershgorin

Sia $\mathcal{J} \subseteq I = \{1, \dots, n\} \subseteq \mathbb{N}$ insieme di indici e $A \in \mathbb{C}^{n \times n}$. Se si ha che l'intersezione:

$$\left(\bigcup_{i \in \mathcal{J}} \mathcal{G}_i^{(r)} \right) \cap \left(\bigcup_{i \in I \setminus \mathcal{J}} \mathcal{G}_i^{(r)} \right) = \emptyset$$

Allora ci sono esattamente $|\mathcal{J}|$ autovalori nel primo insieme e $n - |\mathcal{J}|$ nel secondo insieme

Dimostrazione:

Sulle dispense

□

Considerazioni del Teorema: La stessa cosa vale anche per le colonne, cioè se ci sono due gruppi distinti di dischetti separati, una parte sono in una regione del piano complesso, l'altra dall'altra parte. Se poi c'è un dischetto solo separato dagli altri, allora l'autovalore è necessariamente reale

Esempio sulla stima del Raggio Spettrale

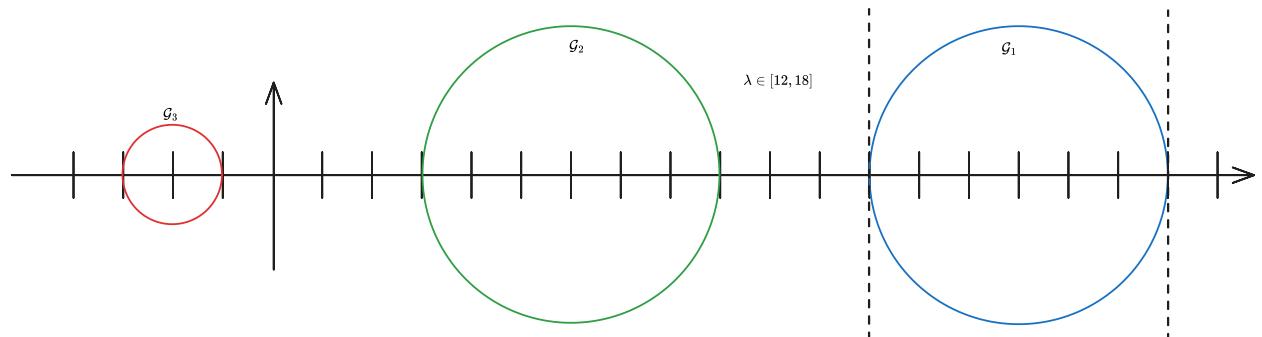
Sia

$$A = \begin{pmatrix} 15 & 1 & 2 \\ -1 & 6 & -2 \\ 1 & 0 & -2 \end{pmatrix}$$

Dobbiamo stimare il raggio spettrale, cioè l'autovalore λ_{min} più grande in modulo

Con i dischi di Gershgorin per le righe otteniamo che:

$$\mathcal{G}_1^{(r)} = \{z - 15| \leq 3\} \quad \mathcal{G}_2^{(r)} = \{|z - 6| \leq 3\} \quad \mathcal{G}_3^{(r)} = \{|z + 2| \leq 1\}$$



Ricordiamo che il raggio spettrale $\rho(A)$ è:

$$\rho(A) = \max_{\lambda \in \text{Spec}(A)} |\lambda|$$

Chiaramente usando i Dischi di Gershgorin anche per le colonne si fa una stima più precisa

Tutto questo risulta comodo per vedere se la matrice è definita positiva, ma torna anche comodo per calcolare $\kappa(A)$
Però non posso concludere niente, sono solo stime

Perturbazione di Autovalori

Teorema (di Bauer - Fike)

Sia $A \in \mathbb{C}^{n \times n}$ diagonalizzabile, cioè:

$$A = X \Lambda X^{-1}$$

Indichiamo con $\lambda(A)$ un autovalore di A .

Sia poi $A + E$ una perturbazione di A , allora per ogni autovalore $\lambda(A + E)$ di $A + E$, esiste un $\lambda(A)$ tale che:

$$|\lambda(A + E) - \lambda(A)| < \kappa(X) \cdot \|E\|$$

Dove $\|\cdot\|$ è una norma matriciale indotta dalla norma euclidea e $\kappa(X) = \|X\| \cdot \|X^{-1}\|$

Considerazioni del Teorema: Se perturbo A con E allora ogni autovalore viene leggermente perturbato e, in particolare, la differenza tra i due è minore della norma di perturbazione $\|E\|$ per un fattore di amplificazione. In questo caso è $\kappa(X)$, cioè più gli autovettori sono vicini all'essere allineati, più è grande la perturbazione è grossa. Se è la matrice è un blocco di Jordan, allora non ci sono abbastanza autovettori e in tal caso abbiamo che $\kappa(X) = +\infty$, quindi la matrice è malcondizionata. Abbiamo quindi che $\kappa(X)$ risente molto da quanto dipende molto da quanto sono buoni gli autovettori. Se A è simmetrica poi, allora esiste una base di autovettori ortonormali, quindi $\kappa(X) = 1$, quindi $\kappa(X) \cdot \|E\| = \|E\|$

Dimostrazione:

Sia (μ, y) un'autocoppia di $A + E$, cioè:

$$(A + E)y = \mu y$$

Riordinando i termini abbiamo che:

$$(A + E)y = \mu y \Rightarrow (\mu I - A)y = Ey$$

In questo caos possiamo distinguere due casi, a seconda della singolarità di $\mu I - A$

Se $\mu I - A$ è singolare, allora abbiamo che $\mu \in Spec(A)$, quindi la diseguaglianza è verificata banalmente-

Se $\mu I - A$ è non singolare, allora posso moltiplicare a sinistra per $(\mu I - A)^{-1}$:

$$(\mu I - A)y = Ey \Rightarrow y = (\mu I - A)^{-1}Ey$$

Con le norme otteniamo che:

$$\|y\| = \|(\mu I - A)^{-1}Ey\| \leq \|(\mu I - A)^{-1}\| \cdot \|E\| \cdot \|y\|$$

Sapendo poi che y è un autovettore, abbiamo che $\|y\| \neq 0$, quindi abbiamo che:

$$1 \leq \|(\mu I - A)^{-1}\| \cdot \|E\|$$

Sfruttiamo adesso la diagonalizzabilità di A :

$$\|(\mu I - A)^{-1}\| = \|(\mu XX^{-1} - X\Lambda X^{-1})^{-1}\| = \|X(\mu I - \Lambda)^{-1}X^{-1}\| \leq \|X\| \cdot \|(\mu I - \Lambda)^{-1}\| \cdot \|X^{-1}\| = \kappa(X) \cdot \|(\mu I - A)^{-1}\|$$

Quindi abbiamo che:

$$1 \leq \|(\mu I - \Lambda)^{-1}\| \leq \kappa(X) \cdot \|(\mu I - A)^{-1}\| \cdot \|E\|$$

Infine abbiamo che:

$$\|(\mu I - \Lambda)^{-1}\| = \left\| \begin{pmatrix} \frac{1}{\mu - \lambda_1} & & \\ & \ddots & \\ & & \frac{1}{\mu - \lambda_n} \end{pmatrix} \right\| \stackrel{*}{=} \max_{i \in \{1, \dots, n\}} \left| \frac{1}{\mu - \lambda_i} \right| = \frac{1}{\min_{i \in \{1, \dots, n\}} |\mu - \lambda_i|}$$

Dove in $*$ abbiamo usato la definizione di norma matriciale indotta:

$$\|D\|^2 = \max_{x \in \mathbb{R}^n} \frac{\|Dx\|^2}{\|x\|^2} = \max_{x \in \mathbb{R}^n} \frac{\left\| \begin{pmatrix} d_1 x_1 \\ \vdots \\ d_n x_n \end{pmatrix} \right\|^2}{\|x\|^2} = \max_{x \in \mathbb{R}^n} \frac{\sum_{i=1}^n d_i^2 x_i^2}{\|x\|^2} \leq \max_{x \in \mathbb{R}^n} \frac{d_{\max}^2 \sum_{i=1}^n x_i^2}{\sum_{i=1}^n x_i^2} = d_{\max}^2$$

E in particolare l'uguaglianza è raggiunta quando:

$$\|D\| = \max_{d_i} |d_i| \Leftrightarrow x = e_i$$

Quindi otteniamo che:

$$1 \leq \kappa(X) \frac{1}{\min |\mu - \lambda_i|} \|E\| \min_{i \in \{1, \dots, n\}} |\mu - \lambda_i| \leq \kappa(X) \cdot \|E\|$$

□

Osservazione: La perturbazione degli autovalori dipende dalla perturbazione di E , ma anche da quanto A è lontana dall'essere normale, infatti se

$$A = X\Lambda X^H \in \mathbb{C}^{n \times n}$$

Cioè A è normale, Λ diagonale e X unitaria, allora si ha che:

$$\|A\| = 1 \Rightarrow \kappa(X) = 1 \Rightarrow |\lambda(A) - \lambda(A + E)| \leq \|E\|$$

Attenzione: Abbiamo $\kappa(X)$ nell'enunciato del teorema, non $\kappa(A)$, in quanto abbiamo un problema che è diverso dalla risoluzione dei sistemi lineari. Dipende esclusivamente da quanto sono ortogonali gli autovettori di A , non dal condizionamento di A . **Sono due cose diverse**

In sintesi, più siamo lontani dall'avere una matrice normale, peggio è

Metodo delle Potenze

Serve per approssimare l'autovalore più grande in modulo

Esempio del Metodo delle Potenze

Sia A la matrice:

$$A = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix} \quad |\lambda_1| > |\lambda_2| > \dots > |\lambda_n| \quad \lambda_i \in \mathbb{R}$$

Facciamo il caso diagonale per questioni di semplicità, non ha senso utilizzare il metodo delle potenze, però serve per rendere le idee.

Iniziamo a moltiplicare. Sia $v \in \mathbb{R}^n$ e facciamo $Av, A^2v, \dots, A^k v, \dots$

Esplcitando le componenti otteniamo che:

$$v = \begin{pmatrix} v_1 \\ \vdots \\ v_n \end{pmatrix} \Rightarrow Av = \begin{pmatrix} \lambda_1 v_1 \\ \vdots \\ \lambda_n v_n \end{pmatrix} \quad A^2 v = \begin{pmatrix} \lambda_1^2 v_1 \\ \vdots \\ \lambda_n^2 v_n \end{pmatrix} \quad \dots \quad A^k v = \begin{pmatrix} \lambda_1^k v_1 \\ \vdots \\ \lambda_n^k v_n \end{pmatrix}$$

Raccogliamo adesso λ_1^k , allora diventa:

$$\begin{pmatrix} \lambda_1^k v_1 \\ \lambda_2^k v_2 \\ \vdots \\ \lambda_n^k v_n \end{pmatrix} = \lambda_1^k \begin{pmatrix} v_1 \\ \frac{\lambda_2^k}{\lambda_1^k} v_2 \\ \vdots \\ \frac{\lambda_n^k}{\lambda_1^k} v_n \end{pmatrix} \xrightarrow{k \rightarrow +\infty} \lambda_1^\infty \begin{pmatrix} v_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad \left(\left| \frac{\lambda_2}{\lambda_1} \right| < 1 \right)$$

Naturalmente mandare k a $+\infty$ è eccessivo, perché alle volte è sufficiente un k più piccolo

In questo modo otteniamo che

$$\begin{pmatrix} v_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \text{ è autovettore per } \lambda_1$$

Quindi in particolare si ottiene che:

$$\frac{A^k v}{\lambda^k} \xrightarrow{k \rightarrow +\infty} e_1 v_1$$

In questo modo faccio quindi saltare fuori l'autovettore legato all'autovalore più grande

Questo metodo è molto costoso, ma si applica principalmente a matrici sparse, in modo da ridurre al minimo i conti
 Ma noi avevamo già qualcosa di simile per i metodi iterativi
 Ma tutto questo è valido anche per matrici non diagonali e la convergenza è all'autovettore dominante
 L'ipotesi che λ_1 è semplice in modulo è fondamentale (infatti se avessimo 2 e -2 come autovalori sarebbe stato un problema, in quanto avrebbero avuto lo stesso modulo/valore assoluto)

Questo tipo di autovalore prende il nome di autovalore semplice:

Definizione di Autovalore Semplice

Un autovalore di $A \in \mathbb{C}^{n \times n}$ si definisce semplice se ha molteplicità 1 e non ci sono autovalori che hanno lo stesso modulo

Per questioni di comodità indichiamo con:

- $x^{(k)}$ la k -esima iterazione
- $(x)_i$ la i -esima componente
- x_j l'autovettore j -esimo

Algoritmo Base del Metodo delle Potenze - Caso $A \in \mathbb{R}^{n \times n}$

Siano $x^{(0)} \in \mathbb{R}^n$ con norma 1 ($\|x^{(0)}\| = 1$)

Per $k = 0, 1, \dots$ fino a convergenza

$$\begin{aligned} y &= Ax^{(k)} \\ x^{(k+1)} &= \frac{y}{\|y\|} \\ \lambda^{(k+1)} &= \frac{(x^{(k+1)})^T A v^{(k+1)}}{(x^{(k+1)})^T x^{(k+1)}} \end{aligned}$$

In particolare, per A simmetrica, quindi per tutti autovalori reali, abbiamo che:

$$\lambda_{min} \leq \frac{x^T A x}{x^T x} \leq \lambda_{max}$$

Abbiamo anche se, se $|\lambda_1| > \dots > |\lambda_n|$, allora:

$$x^{(k+1)} \xrightarrow{k \rightarrow +\infty} x_1$$

Quindi dopo k iterazioni otteniamo che:

$$x^{(k)} = \alpha_k A^k x^{(0)}$$

Lo scalare α_k serve solamente per normalizzare il vettore

Però mancano ancora un paio di cose per risanire l'algoritmo.

Metodo di Arresto

Andiamo a trovare un modo per definire il residuo: data un'autocoppia approssimata $(\lambda^{(k)}, x^{(k)})$ definiamo il residuo come:

$$r^{(k)} = Ax^{(k)} - \lambda^{(k)} x^{(k)}$$

Come per la risoluzione di sistemi lineari serve per vedere quanto siamo vicini alla soluzione esatta

Quindi, nella parte finale dell'algoritmo possiamo aggiungere:

$$\text{Se } \frac{\|r^{(k)}\|}{|\lambda^{(k)}|} < tol \text{ allora stop}$$

In questo caso abbiamo che: $\|r^{(k)}\|$ è l'unica quantità che possiamo monitorare

Attenzione: Bisogna fare attenzione a non confondere questo residuo con $r = b - Ax$, il residuo cambia in base al problema.

Se però mettiamo quest'ultima cosa nell'algoritmo abbiamo che dobbiamo fare troppi prodotti di matrici. Cerchiamo quindi di riscrivere l'algoritmo in maniera più economica computazionalmente parlando

Seconda Versione dell'Algoritmo

Siano $x^{(0)} \in \mathbb{C}^n$ che abbia norma uguale a 1 ($\|x\| = 1$) e siano $y = Ax^{(0)}$, $maxit$ e tol

Per $k = 0, 1, \dots, maxit$

$$\begin{aligned}\lambda^{(k)} &= (x^{(k)})^H y && \text{che ha un costo di } 2n - 1 \text{ flops} \\ r^{(k)} &= y - \lambda^{(k)} x^{(k)} && \text{che ha un costo di } 2n \text{ flops} \\ \text{Se } \frac{\|r^{(k)}\|}{|\lambda^{(k)}|} &< tol \text{ allora Stop} && \text{che ha un costo di } 2n \text{ flops} \\ x^{(k+1)} &= \frac{y}{\|y\|} && \text{che ha un costo di } n + 2n \text{ flops} \\ y &= Ax^{(k+1)} && \text{che ha un costo di } n + 2n \text{ flops}\end{aligned}$$

Osservazione Diversamente da quanto avevamo fatto con gli altri algoritmi abbiamo che il vettore di partenza deve essere non nullo, altrimenti non andremmo mai da nessuna parte. Infatti avremmo costantemente:

$$A^k x = A^k \underline{0} = \underline{0}$$

Non andremmo mai da nessuna parte, quindi necessariamente deve essere non nullo.

Rispetto a prima quest'algoritmo è molto più efficiente rispetto a prima e ci siamo solamente limitati a riordinare, infatti ora c'è soltanto un sistema lineare da risolvere.

Facendo i conti otteniamo che il costo computazionale è pari a $9n$ flops più il costo della risoluzione di un sistema lineare.

Variante 1: Metodo delle Potenze Inverse

Di norma abbiamo che con il metodo delle potenze inverse otteniamo l'autovalore più grande in modulo, cioè:

$$A^k x^{(0)} \xrightarrow{k \rightarrow +\infty} \lambda_1$$

Con $|\lambda_1| > |\lambda_2|, \dots, |\lambda_n|$

Come possiamo fare per avere quello più piccolo?

Se A è invertibile, ci basta usare A^{-1} , in modo da trovare il più grande $|\frac{1}{\lambda_i}|$, cioè λ_i più vicino allo zero. Scriviamo una prima bozza dell'algoritmo:

Algoritmo 1

Siano $x^{(0)} \in \mathbb{C}^n$ che abbia norma uguale a 1 ($\|x\| = 1$) e siano $y = A^{-1}x^{(0)}$, $maxit$ e tol

Per $k = 0, 1, \dots, maxit$

$$\begin{aligned}\theta^{(k)} &= (x^{(k)})^H y && \text{che ha un costo di } 2n - 1 \text{ flops} \\ r^{(k)} &= y - \theta^{(k)} x^{(k)} && \text{che ha un costo di } 2n \text{ flops} \\ \text{Se } \frac{\|r^{(k)}\|}{|\theta^{(k)}|} &< tol \text{ allora Stop} && \text{che ha un costo di } 2n \text{ flops} \\ x^{(k+1)} &= \frac{y}{\|y\|} && \text{che ha un costo di } n + 2n \text{ flops} \\ y &= A^{-1}x^{(k+1)} && \text{che ha un costo della risoluzione di un sistema lineare } Ay = x^{(k+1)}\end{aligned}$$

Quest'algoritmo fa esattamente la stessa cosa dell'algoritmo precedente:

$$\begin{aligned}x^{(k)} &\xrightarrow{k \rightarrow +\infty} x_1 \\ \theta^{(k)} &\xrightarrow{k \rightarrow +\infty} \text{Autovalore più grande di } A^{-1} \text{ in modulo} \Leftrightarrow \lambda_{min}\end{aligned}$$

Osservazione: Abbiamo quindi trovato l'autovalore:

$$\lambda^{(k)} = \frac{1}{\theta^{(k)}} \approx \lambda_{min}$$

Cosa possiamo dire però dell'autovettore?

$$Ax = \lambda x \Leftrightarrow \frac{1}{\lambda}x = A^{-1}x$$

L'autovettore è esattamente lo stesso x_1

Osservazione: Ad ogni iterazione devo risolvere un sistema lineare del tipo $Ay = x^{k+1}$

Sapendo che la matrice del sistema lineare è sempre la stessa per ogni iterazione, possiamo prima fattorizzare A utilizzando un metodo, per poi andare a sostituire A dopo con la sua fattorizzazione

Se le dimensioni lo permettono (sono basse), allora posso utilizzare un metodo diretto. Inoltre se le caratteristiche di A permettono un metodo specifico più conveniente allora è meglio sfruttarli.

Riduciamoci al caso di una matrice generica: sfruttiamo la fattorizzazione LU . In questo caso:

1. Prima di cominciare con l'algoritmo calcolare la fattorizzazione LU e memorizzare le due matrici (con un costo di $\Theta(n^3)$)
2. Ad ogni iterazione risolvo il sistema lineare con L e poi con U (con un costo di $\Theta(n^2)$)

Quindi otteniamo che ad ogni iterazione abbiamo abbassato di un grado l'ordine di grandezza

Variante 2: Metodo delle Potenze Inverse Traslate/Shiftate

L'obiettivo di questa variante è quello di approssimare (trovare l'approssimazione) l'autovalore λ ad uno scalare $\sigma \in \mathbb{C}$, *che prende il nome di Shift*. In particolare possiamo distinguere due casi.

Se $\sigma = 0$ allora ci si rifà al Metodo delle Potenze Inverse.

Se $\sigma \neq 0$ allora possiamo portare tale valore a zero e fare come prima, cioè prendendo la matrice $(A - \sigma I)^{-1}$:

$$Ax = \lambda x \Leftrightarrow Ax - \sigma x = \lambda x - \sigma x \Rightarrow (A - \sigma I)x = (\lambda - \sigma)x$$

Supponiamo $A - \sigma I$ non singolare e $\lambda - \sigma \neq 0$ (*altrimenti è banalmente verificato*):

$$Ax - \sigma x = \lambda x - \sigma x \Leftrightarrow \frac{1}{\lambda - \sigma}x = (A - \sigma I)^{-1}x$$

Otteniamo poi che θ è grande se λ è vicino a σ , cioè più λ è vicina a σ , più è grande θ

Possiamo poi approssimare θ come:

$$\theta = \frac{1}{\lambda - \sigma} \Leftrightarrow \lambda - \sigma = \frac{1}{\theta} \Leftrightarrow \lambda = \sigma + \frac{1}{\theta}$$

Però non sappiamo effettivamente quale è l'autovalore, ma sappiamo che è vicino a σ

Algoritmo 2

Siano $x^{(0)} \in \mathbb{C}^n$ che abbia norma uguale a 1 ($\|x\| = 1$) e siano $y = (A - \sigma I)^{-1}x^{(0)}$, $maxit$, tol e $\sigma \in \mathbb{C}$

Per $k = 0, 1, \dots, maxit$

$\theta^{(k)} = (x^{(k)})^H y$	<i>che ha un costo di $2n - 1$ flops</i>
$r^{(k)} = y - \theta^{(k)}x^{(k)}$	<i>che ha un costo di $2n$ flops</i>
Se $\frac{\ r^{(k)}\ }{ \theta^{(k)} } < tol$ allora Stop	<i>che ha un costo di $2n$ flops</i>
$x^{(k+1)} = \frac{y}{\ y\ }$	<i>che ha un costo di $n + 2n$ flops</i>
$y = A^{-1}x^{(k+1)}$	<i>che ha un costo della risoluzione del sistema lineare $(A - \sigma I)y = x^{(k+1)}$</i>

Rispetto a prima abbiamo solo cambiato matrice con cui risolviamo il sistema lineare e tutte le osservazioni fatte prima continuano a valere

Bisogna però fare attenzione perché con quest'algoritmo, otteniamo un autovalore θ che non ha niente a che fare con λ , quindi non è finito in sé per sé, bisogna prima fare una trasformazione

Si perdono tanti punti all'esame per sta stronza

Osservazione: Per $\sigma = 0$ riotteniamo il Metodo delle Potenze Inverse

Algoritmo 2 - Senza Trasformazioni da Fare

Siano $x^{(0)} \in \mathbb{C}^n$ che abbia norma uguale a 1 ($\|x\| = 1$) e siano $maxit$, tol e $\sigma \in \mathbb{C}$

Per $k = 0, 1, \dots, maxit$

$\lambda^{(k)} = (x^{(k)})^H A x^{(k)}$	<i>Questo non è altro che il quoziente di Rayleigh - Per fare i conti possiamo poi fare</i>
$w = A x^{(k)}$	
$r^{(k)} = A x^{(k)} - \lambda^{(k)} x^{(k)}$	$r^{(k)} = w - \lambda^{(k)} x^{(k)}$

Se $\frac{\|r^{(k)}\|}{|\theta^{(k)}|} < tol$ allora Stop

$$x^{(k+1)} = \frac{y}{\|y\|}$$

$$y = A^{-1}x^{(k+1)}$$

Un altro vantaggio che offre questo algoritmo rispetto a quello precedente è che posso utilizzare la tolleranza sul residuo direttamente per $\lambda^{(k)}$, e non per $\theta^{(k)}$, in quanto abbiamo che i residui non sono uguali tra loro, infatti, *indicando con il cappellino sopra quello del primo algoritmo*, avevamo che:

$$\hat{r}^{(k)} = (A - \sigma I)^{-1}x^{(k)} - \theta^{(k)} - x^{(k)} \quad \text{e} \quad \frac{\|\hat{r}^{(k)}\|}{|\theta^{(k)}|} < tol$$

Ma è anche vero che:

$$\|\hat{r}^{(k)}\| \neq \|r^{(k)}\|$$

Se li dovessi prendere come uguali, avrei poi dei problemi di approssimazione. Non posso essere sicuro di ottenere un risultato preciso in quanto gestisco le autocoppie di un'altra matrice

Però per quanto riguarda lo scritto l'uno vale l'altro

Criterio di Arresto

Teorema

Sia $A \in \mathbb{C}^{n \times n}$ normale e sia $(\lambda^{(k)}, x^{(k)})$ coppia scalare-vettore e sia:

$$r^{(k)} = Ax^{(k)} - \lambda^{(k)}x^{(k)}$$

Allora esiste $\lambda \in \text{Spec}(A)$ tale che:

$$|\lambda - \lambda^{(k)}| \leq \|r^{(k)}\|$$

Dimostrazione:

Intanto possiamo scrivere il residuo come:

$$r^{(k)} = Ax^{(k)} - \lambda^{(k)}x^{(k)} = (A - \lambda^{(k)}I)x^{(k)}$$

Se $\lambda^{(k)}$ è autovalore allora è banalmente verificato.

Supponiamo quindi che λ non sia autovalore, allora abbiamo che $(A - \lambda^{(k)}I)$ è non singolare:

$$x^{(k)} = (A - \lambda^{(k)}I)^{-1}r^{(k)}$$

Utilizzando le norme e seguendo la dimostrazione del teorema di Bauer - Fike otteniamo che:

$$\|x^{(k)}\| \leq \|(A - \lambda^{(k)}I)^{-1}\| \cdot \|r^{(k)}\|$$

Sapendo tuttavia che $\|x^{(k)}\| = 1$ e che A è normale abbiamo che $A = X\Lambda X^H$, quindi:

$$1 \leq \|(A - \lambda^{(k)}I)^{-1}\| \cdot \|r^{(k)}\| = \|(A - \lambda^{(k)}I)^{-1}\| \cdot \|r^{(k)}\| = \frac{\|r^{(k)}\|}{\min |\lambda - \lambda^{(k)}|}$$

Da cui si ottiene che:

$$|\lambda - \lambda^{(k)}| \leq \|r^{(k)}\|$$

□

Di solito non calcoliamo questo valore, ma ci sono dei casi particolari in cui conosciamo λ

Osservazione: Il risultato è una conseguenza (o applicazione) del teorema di Bauer - Fike. Infatti:

$$r^{(k)} = Ax^{(k)} - \lambda^{(k)}x^{(k)} \Rightarrow Ax^{(k)} - r^{(k)} = \lambda^{(k)}x^{(k)} \Rightarrow Ax^{(k)} - r^{(k)}(x^{(k)})^H x^{(k)} = \lambda^{(k)}x^{(k)} \Rightarrow (A - \underbrace{x^{(k)}(x^{(k)})^H}_{-E})x^{(k)} = \lambda^{(k)}x^{(k)}$$

Da cui abbiamo che:

$$(A + E)x^{(k)} = \lambda^{(k)}x^{(k)}$$

Quindi possiamo dedurre che $(\lambda^{(k)}, x^{(k)})$ è un'autocoppia esatta di $A + E$

Per dire effettivamente che è una buona soluzione approssimata, dobbiamo mostrare che $\|E\|$ è piccola se $A + E$ è una soluzione ad un problema vicino ad A

Andiamo quindi a calcolare a cosa corrisponde $\|E\|$ (*per comodità di notazioni poniamo $r = r^{(k)}$ e $x = x^{(k)}$*):

$$\|E\| = \|rx^H\| = \max_{\substack{v \neq 0 \\ v \in \mathbb{C}^n}} \frac{\|rx^Hv\|}{\|v\|} = \max_{v \neq 0} \|r\| \frac{|r^Hv|}{\|v\|} \xrightarrow{\|x\|=1} \max_{v \neq 0} \|r\| \frac{|x^Hv|}{\|v\| \cdot \|x\|} = \|r\| \max_{v \neq 0} \frac{|x^Hv|}{\|x\| \cdot \|v\|}$$

Tuttavia abbiamo che quest'uguaglianza è minore uguale di uno per come per ogni v non nullo e in particolare è uguale a 1 se $v = x$.

Otteniamo quindi che: il massimo che raggiungiamo è 1, quindi:

$$\|E\| = \|r\| \max_{v \neq 0} \frac{|x^Hv|}{\|x\| \cdot \|v\|} = \|r\| \quad \Rightarrow \quad \|E\| = \|r^{(k)}\|$$

In questo modo stiamo dicendo che $(\lambda^{(k)}, x^{(k)})$ è la soluzione esatta di $A + E$ ma quanto è vicino questo problema? Più il residuo è piccolo, più è vicino

Inoltre, per il teorema di Bauer - Fike abbiamo che:

$$|\lambda - \lambda^{(k)}| \leq \kappa(X) \cdot \|E\| \stackrel{*}{=} \|E\| = \|r^{(k)}\| \quad * : A \text{ normale} \Rightarrow \kappa(X) = 1$$

Questa non è altro che un'applicazione del teorema di Bauer Fike per matrici normali

Analisi di Convergenza

Avevamo accennato in precedenza che per k che tende a $+\infty$ abbiamo che:

$$\lambda^{(k)} \rightarrow \lambda_1 \quad \text{e} \quad x^{(k)} \rightarrow x_1$$

Supponiamo di avere A diagonalizzabile, cioè $A = X\Lambda X^{-1}$ e supponiamo di avere due successioni $(x^{(k)})_{k>0}$ e $(\lambda^{(k)})_{k>0}$ ottenute dalla successione delle potenze.

Supponiamo anche che gli autovalori siano legati dalla relazione: $|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|$ cioè λ_1 è semplice in modulo

Andiamo a studiare come si comporta $\frac{x^{(k)}}{\lambda_1^k}$ per k che tende a più infinito:

$$\frac{x^{(k)}}{\lambda_1^k} = \frac{A^k x^{(0)}}{\lambda_1^k} = \frac{X\Lambda^k X^{-1} x^{(0)}}{\lambda_1^k}$$

Poniamo poi ξ come:

$$\xi = X^{-1}x^{(0)} \quad \Rightarrow \quad \frac{X\Lambda\xi}{\lambda_1^k}$$

Supponiamo poi che $X = (\underline{x}_1 \quad \underline{x}_2 \quad \dots \quad \underline{x}_b)$, allora abbiamo che:

$$\frac{X\Lambda\xi}{\lambda_1^k} = \frac{\sum_{i=1}^n \underline{x}_i \lambda_i^k (\xi)_i}{\lambda_1^k} = \frac{\underline{x}_1 \lambda_1^k (\xi)_1}{\lambda_1^k} + \frac{\sum_{i=2}^n \underline{x}_i \lambda_i^k (\xi)_i}{\lambda_1^k} = \underline{x}_1 (\xi)_1 + \underbrace{\sum_{i=2}^n \underline{x}_i \left(\frac{\lambda_i}{\lambda_1}\right)^k (\xi)_i}_{<1} \xrightarrow{k \rightarrow +\infty} \underline{x}_1 (\xi)_1$$

In quest'ultimo passaggio ritorna fondamentale il fatto che $|\lambda_1| > |\lambda_i| \forall i$

In matematica esatta è anche importante avere che $(\xi)_1 \neq 0$. Sapendo poi che

$$\xi = X^{-1}x^{(0)} \Leftrightarrow x^{(0)} = X\xi = \underline{x}_1(\xi)_1 + \underline{x}_2(\xi)_2 + \dots + \underline{x}_n(\xi)_n$$

Abbiamo che $(\xi)_1 \neq 0$ implica che $x^{(0)}$ non deve avere componente nulla nel vettore in cui andiamo, altrimenti andremmo ad accrescere gli altri autovalori. Quindi il fatto che il vettore di partenza non abbia componenti nulle nella direzione in cui vogliamo andare è un'ipotesi più che naturale

Però non è una richiesta che possiamo controllare, perché nell'effettivo non sappiamo quale sia la direzione in cui dobbiamo andare. Quindi per ovviare al problema possiamo generare il vettore randomicamente, in questo modo è estremamente improbabile trovare un vettore ortogonale all'autovettore voluto

In matematica finita le cose cambiano completamente, in quanto c'è il round-off, quindi in qualche modo qualcosa viene sempre fuori.

In un certo senso abbiamo quindi dimostrato il seguente risultato:

Teorema

Siano $|\lambda_i|$ autovalori ordinati in modo decrescente con $|\lambda_1| > |\lambda_i|$ per $i \in \{2, \dots, n\}$ e sia x_1 autovettore corrispondente a λ_1 . Sia $x^{(0)} = X\xi$ l'iterato iniziale del metodo delle potenze con $(\xi)_1 \neq 0$, allora esiste una costante $C > 0$ tale che:

$$\|x^{(k)} - x_1\|_2 \leq C \left| \frac{\lambda_2}{\lambda_1} \right|^k$$

Dove $C > 0$ e $x^{(k)}$ è normalizzato in modo opportuno

Considerazioni del Teorema: In questo teorema andiamo a fare una stima per l'errore come vettore (in particolare l'errore dato nella direzione del vettore) come differenza tra il vettore che stiamo iterando e l'autovettore effettivo. In particolare lo stiamo mettendo a confronto con la costante $|\lambda_2/\lambda_1|$ dove $|\lambda_2|$ è il secondo autovalore più grande in modulo. Questo rapporto prende il nome di *Velocità di Convergenza*

In particolare abbiamo che possiamo parlare di *Convergenza Lineare* in quanto l'errore varia secondo il fattore $|\lambda_2/\lambda_1|$, cioè in maniera lineare. Questo tipo di concetto è un concetto che abbiamo già visto quando abbiamo fatto le classi di metodi iterativi.

Abbiamo inoltre che più è piccolo il fattore di convergenza (quindi $|\lambda_2| << |\lambda_1|$), più è veloce il metodo, quindi meno iterazioni servono. Viceversa, più il rapporto tende a 1, più il metodo richiederà iterazioni

Inoltre posso prendere direttamente $|\lambda_2|$ in quanto abbiamo che, essendo il secondo autovalore più grande in modulo, guida la velocità di convergenza

Osservazione: Il fatto che $|\lambda_1|$ sia semplice influenza molto di più se si hanno matrici reali.

Infatti, data $A \in \mathbb{R}^{n \times n}$ non simmetrica, abbiamo che λ_1 potrebbe essere complesso e avere un coniugato. Quindi affinché ci sia la convergenza, si ha che $\lambda_1 \in \mathbb{R}$. In questo caso comunque possiamo anche prendere $x^{(0)} \in \mathbb{R}^n$

Osservazione: Se A è Hermitiana in $\mathbb{C}^{n \times n}$ oppure è simmetrica in $\mathbb{R}^{n \times n}$, allora la convergenza del metodo delle potenze p quadratica per l'autovalore (cioè dipende da $|\lambda_2/\lambda_1|^2$) mentre la convergenza per il vettore resta lineare, infatti:

$$\lambda^{(k)} = \frac{(x^{(k)})^H A x^{(k)}}{(x^{(k)})^H x^{(k)}}$$

Tuttavia, sapendo che: $x^{(k)} = \alpha_k A^k x^{(0)}$ con α scalare per normalizzare il vettore. si ha che:

$$\lambda^{(k)} = \frac{(x^{(k)})^H A x^{(k)}}{(x^{(k)})^H x^{(k)}} \Rightarrow \lambda^{(k)} = \frac{(\alpha_k A^k x^{(0)})^H A (\alpha_k A^k x^{(0)})}{(\alpha_k A^k x^{(0)})^H (\alpha_k A^k x^{(0)})}$$

Sapendo poi che A è Hermitiana, abbiamo che $A = A^H$ e che $A = X\Lambda X^H$

$$\lambda^{(k)} = \frac{(\alpha_k A^k x^{(0)})^H A (\alpha_k A^k x^{(0)})}{(\alpha_k A^k x^{(0)})^H (\alpha_k A^k x^{(0)})} = \frac{(x^{(0)})^H X \Lambda^{2k+1} X^H x^{(0)}}{(x^{(0)})^H X \Lambda^{2k} X^H x^{(0)}}$$

Per questione di comodità poniamo $q = X^H x^{(0)}$, quindi:

$$\lambda^{(k)} = \frac{(x^{(0)})^H X \Lambda^{2k+1} X^H x^{(0)}}{(x^{(0)})^H X \Lambda^{2k} X^H x^{(0)}} = \frac{q^H \Lambda^{2k+1} q}{q^H \Lambda^{2k} q}$$

Sempre per questioni di comodità, sviluppiamo il denominatore separato dalla frazione (per il numeratore sarà la stessa cosa):

$$q^H \Lambda^{2k} q = ((\bar{q})_1 \ \dots \ (\bar{q})_n) \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix} \begin{pmatrix} (q)_1 \\ \vdots \\ (q)_n \end{pmatrix} = ((\bar{q})_1 \ \dots \ (\bar{q})_n) \begin{pmatrix} \lambda_1 (q)_1 \\ \vdots \\ \lambda_n (q)_n \end{pmatrix} = \sum_{i=1}^n (\bar{q})_i \lambda_i^{2k} (q)_i = \sum_{i=1}^n \lambda_i^{2k} |(q)_i|^2$$

Andando a sostituire sopra otteniamo che:

$$\lambda^{(k)} = \frac{q^H \Lambda^{2k+1} q}{q^H \Lambda^{2k} q} = \frac{\sum_{i=1}^n \lambda_i^{2k+1} |(q)_i|^2}{\sum_{i=1}^n \lambda_i^{2k} |(q)_i|^2}$$

Sapendo poi che: $\lambda_1^{2k}|(q)_1|^2$ è minore della somma di tutti i $\lambda_i^{2k}|(q)_i|^2$, si ha che:

$$\lambda^{(k)} = \frac{\sum_{i=1}^n \lambda_i^{2k+1}|(q)_i|^2}{\sum_{i=1}^n \lambda_i^{2k}|(q)_i|^2} \leq \frac{\sum_{i=1}^n \lambda_i^{2k+1}|(q)_i|^2}{\lambda_1^{2k}|(q)_1|^2} = \frac{\lambda_1^{2k+1}|(q)_1|^2}{\lambda_1^{2k}|(q)_1|^2} + \frac{\sum_{i=2}^n \lambda_i^{2k+1}|(q)_i|^2}{\lambda_1^{2k}|(q)_1|^2} = \lambda_1 + \sum_{i=2}^n \left(\frac{\lambda_i}{\lambda_n} \right)^{2k} \lambda_i \frac{|(q)_i|^2}{|(q)_1|^2}$$

Tutto il termine dentro al termine di sommatoria è approssimabile come $\Theta(|\frac{\lambda_i}{\lambda_1}|^2)$

Quindi effettivamente decresce in maniera quadratica.

Va ribadito che questi trucchetti possono essere usati solamente perché A è Hermitiana in \mathbb{C} oppure simmetrica in \mathbb{R}

Osservazione:

Esercizi Svolti

1. Dimostrare direttamente l'unicità della fattorizzazione LU sotto le ipotesi del teorema:

Soluzione dell'Esercizio 1

Supponiamo quindi esistano 2 fattorizzazioni LU , ossia esistano L, L' matrici triangolari inferiori con tutti 1 sulla diagonale principale e due matrici U, U' triangolari superiori tali che:

$$A = LU = L'U'$$

Chiaramente otteniamo che $LU = L'U'$

Poiché L è invertibile otteniamo che

$$LU = L'U' \Rightarrow (L')^{-1}LU = U'$$

Supponiamo che U sia non singolare, allora abbiamo che A è non singolare

Quindi possiamo porre l'uguaglianza

$$(L')^{-1}L = U'U^{-1}$$

Notiamo che il membro a sinistra dell'uguaglianza è una matrice ancora triangolare inferiore, mentre quello a destra è una matrice triangolare superiore, segue quindi che, purché l'uguaglianza sia verificata, devono essere entrambe matrici diagonali.

Sappiamo inoltre che sulla diagonale di L e di L' , e quindi anche su quella di $(L')^{-1}$ ci sono solo 1 sulla diagonale principale, quindi anche sulla diagonale principale di $L(L')^{-1}$ ci sono solo 1.

Ma affinché sia vera l'uguaglianza abbiamo che $U'U^{-1}$ deve essere diagonale e deve avere tutti 1 sulla diagonale principale, ossia deve essere l'identità, quindi $U = U'$

Da qui segue direttamente che $(L')^{-1}L = I \Rightarrow L = L'$

Quindi la fattorizzazione è unica.

Guardiamo ora il caso in cui la matrice U non sia invertibile.

Per questioni di comodità, definiamo

$$\hat{L} = (L')^{-1}L$$

A questo punto otteniamo che:

$$(L')^{-1}LU = U' \Rightarrow \hat{L}U = U'$$

Per le ipotesi del teorema abbiamo che $U_{1,1}, U_{2,2}, U_{n-1,n-1}$ sono elementi non nulli.

Notiamo anche ci stroviamo in una situazione del tipo

$$(\Delta) \cdot (\nabla) = (\nabla)$$

Proviamo a moltiplicare l'ultima riga di \hat{L} per la prima colonna di U , in questo modo otteniamo l'elemento $U'_{n,1}$. Tuttavia otteniamo che questo prodotto è nullo in quanto è un elemento di una matrice triangolare superiore situato sotto la diagonale principale. Segue quindi che uno tra $\hat{L}_{n,1}$ e $U_{1,1}$ è nullo, tuttavia sappiamo per l'osservazione appena fatta che $U_{1,1} \neq 0$, quindi $\hat{L}_{n,1} = 0$.

Andiamo avanti con la seconda riga, quindi

$$\hat{L}_{n,:} \cdot U_{:,2} = U'_{n,2} \Rightarrow \hat{L}_{n,2} - U_{2,2} = 0 \Rightarrow \hat{L}_{n,2} = 0$$

In maniera analoga possiamo fare questo procedimento per ogni riga e per ogni colonna. Otterremo quindi che:

$$\hat{L}_{i,j} = 0 \quad \forall i > j$$

Quindi $\hat{L} = I \Rightarrow L = L'$, di conseguenza segue subito che $U = U'$

In questo modo abbiamo dimostrato che la fattorizzazione LU è unica anche per matrici non singolari.

2. Sia $A \in \mathbb{R}^{n \times n}$ una matrice tridiagonale con fattorizzazione LU.

Proprieta un algoritmo che risolvi

$$AX = F \Leftrightarrow X(x_1 \dots x_s) = (f_1 \dots f_s) \quad \text{con } f_i \in \mathbb{R}^n, f_i \neq 0 \quad \forall i \in \{1, \dots, s\}$$

che tenga conto della struttura di A e che minimizzi il costo computazionale.

Soluzione dell'Esercizio 2

Quando abbiamo una matrice tridiagonale, sappiamo che dobbiamo utilizzare l'algoritmo di Thomas

Sapendo che la struttura della matrice A è presentata come nel paragrafo dell'algoritmo di Thomas otteniamo che:

$$\left. \begin{array}{l} \alpha_1 = a_1 \\ \beta_i = b_i/\alpha_{i-1} \\ \alpha_i = a_i - \beta_i c_{i-1} \\ y_1 = f_1 \\ y_i = f_i - \beta_i y_{i-1} \\ x_n = y_n/\alpha_n \\ x_i = \frac{1}{\alpha_i}(y_i - c_i x_{i+1}) \end{array} \right\} \quad \begin{array}{l} i \in \{2, \dots, n\} \\ Ly = f \\ Ux = y \end{array} \quad \text{con } A = \begin{pmatrix} a_1 & c_1 & 0 & \cdots & \cdots & 0 \\ b_2 & a_2 & c_2 & \ddots & & \vdots \\ 0 & b_3 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & c_{n-1} \\ 0 & \cdots & \cdots & 0 & b_n & a_n \end{pmatrix}$$

In questo caso posso fattorizzare una volta soltanto e poi posso risolvere per tutti i termini noti

Dalla fattorizzazione ottengo che costa soltanto $3(n-1)$ flops.

Va sottolineato che non dipende minimamente dalla grandezza di s

Per la risoluzione dei sistemi lineari, posso risolvere uno per volta o tutti insieme, è indifferente a livello computazionale, ma matlab preferisce fare tutto insieme, cioè sfruttando matrici

Nel primo caso otteniamo:

$$Ly_j = f_j \quad Ux_j = y_j \quad \text{per } j \in \{1, \dots, s\}$$

Nel secondo caso ottengo:

$$LY = F \Rightarrow \begin{cases} Y_{1,:} = F_{1,:} \\ Y_{i,:} = F_{i,:} - \beta_i Y_{i-1,:} \end{cases} \quad UX = Y \Rightarrow \begin{cases} X_{n,:} = \frac{1}{\alpha_n} Y_{n,:} \\ X_{i,:} = \frac{1}{\alpha_i} (Y_{i,:} - c_i X_{i+1,:}) \end{cases}$$

Solo per questa parte abbiamo un costo computazionale pari a $\Theta(5n)$ per singolo sistema, quindi in totale $\Theta(5ns)$

Considerando la parte iniziale abbiamo che il costo totale è $5ns + 4n + \Theta(s)$

Osservazione: Questo è un caso tipico in cui può avere senso costruire delle matrici LU esplicitamente per la risoluzione di $Ax = b$ invece di fare l'elemento invece di fare l'eliminazione di Gauss.

3. Dato un sistema della forma

$$\begin{pmatrix} A & b \\ c^T & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}$$

con A tridiagonale, c, b, f vettori in \mathbb{R}^n e $x \in \mathbb{R}^n$ e $y \in \mathbb{R}$

Soluzione dell'Esercizio 3

Facciamo prima i conti e spezziamo i blocchi:

$$\begin{cases} Ax + by = f \\ c^T x = 0 \end{cases}$$

Risolviamolo prima in maniera formale e poi andiamo più nei dettagli

Dalla prima otteniamo che:

$$Ax = f - by \Rightarrow x = A^{-1}(f - by)$$

Sostituendo otteniamo che

$$c^T A^{-1} f - c^T A^{-1} b y = 0 \Rightarrow y = \frac{c^T A^{-1} f}{c^T A^{-1} b} \quad c^T A^{-1} b \neq 0$$

Una volta ottenuta y otteniamo che:

$$x = A^{-1}f - A^{-1}by$$

Andiamo nel dettaglio

Dobbiamo prima risolvere due sistemi in A :

$$Aw_1 = f \quad Aw_2 = b \quad \Rightarrow \quad y = \frac{c^T w_1}{c^T w_2} \quad \Rightarrow \quad x = w_1 - w_2 y$$

Possiamo sfruttare la risoluzione dell'esercizio precedente come

$$AW = (f, b)$$

Chiaramente poi andrà spiegato pienamente nei dettagli

Quindi abbiamo che il costo totale è:

$$\underbrace{4n}_{LU} + \underbrace{2 \cdot 5n}_{\text{Sol. } LU} + \underbrace{2 \cdot 2}_{c^T w_i} + \underbrace{2n}_x = 19n + \Theta(1)$$

4. Sia $A \in \mathbb{R}^{n \times n}$ non singolare con $A_{i,i} \neq 0$ per ogni i , $A_{1,n} = A_{n,1} \neq 0$, tutto il nullo.

1. Mostrare che la fattorizzazione LU senza pivot esiste sempre
2. Descrivi l'algoritmo di Gauss, modificato per la risoluzione di $Ax = b$ col minimo costo

Soluzione dell'Esercizio 4

Abbiamo che la matrice A è della forma:

$$A = \begin{pmatrix} \times & & \times \\ & \ddots & \\ \times & & \times \end{pmatrix}$$

1. Tutti i minori di testa non sono zero, quindi il teorema assicura che tale decomposizione LU esista
2. Se avessimo avuto scelta, avremmo potuto utilizzare la formula di Sherman - Morrison, ma il testo dell'esercizio ce lo impedisce

Con Gauss avevamo che

$$m_{i,1} = \frac{A_{i,1}^{(1)}}{A_{1,1}^{(1)}} \quad \text{per } i \in \{2, \dots, n\}$$

Qui invece, sfruttando la struttura della matrice, ci basta porre $i = n$ e ottenere:

$$m_{n,1} = \frac{A_{n,1}^{(1)}}{A_{1,1}^{(1)}} \quad \Rightarrow \quad A_{n,j}^{(2)} = A_{n,j}^{(1)} - m_{n,1} A_{1,j}^{(1)} \quad \text{con } j = n$$

In questo modo otteniamo che la matrice $A^{(2)}$ è della forma:

$$A^{(2)} = \begin{pmatrix} * & * \\ & \ddots \\ & & * \end{pmatrix} \quad \text{con * elementi non nulli}$$

Quindi ora possiamo procedere alla risoluzione del sistema lineare del tipo:

$$\begin{pmatrix} * & * \\ & \ddots \\ & & * \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1^{(2)} \\ \vdots \\ b_n^{(2)} \end{pmatrix}$$

Da cui possiamo procedere con la usuale risoluzione di un sistema lineare:

$$\begin{cases} x_j = \frac{b_j^{(2)}}{A_{1,1}^{(2)}} & j \in \{2, \dots, n\} \\ A_{1,1}^{(2)}x_1 + A_{1,n}^{(2)}x_n = b_1^{(2)} & j = 1 \end{cases} \Rightarrow \begin{cases} x_j = \frac{b_j^{(2)}}{A_{1,1}^{(2)}} & j \in \{2, \dots, n\} \\ x_1 = \frac{b_1^{(2)} - A_{1,n}^{(2)}x_n}{A_{1,1}^{(2)}} & j = 1 \end{cases}$$

Quindi il costo computazionale totale è pari a $(n-1) + 3 + 3 = n-5$ flops

5. Sia A una matrice simmetrica definita positiva di cui sono noti gli autovalori estremi. Sia $E = -uu^T$ con $u \in \mathbb{R}^n$ non nullo e $\|\cdot\|$ norma matriciale indotta dalla norma euclidea:

1. Mostrare che $\|E\| = \|u\|^2$
2. Mostrare che $\lambda_{\max}(A) - \|u\| \leq \|A + E\| \leq \lambda_{\max}(A) + \|u\|^2$
3. Determinare la norma minima $\|E\|$ tale che la matrice $A + E$ sia singolare.

Soluzione dell'Esercizio 5

Mostriamoli per punti:

1. $\|E\|$ è definita come:

$$\|E\| = \max_{x \neq 0} \frac{\|Ex\|}{\|x\|} = \max_{x \neq 0} \frac{\|uu^Tx\|}{\|x\|} = \max_{x \neq 0} \|u\|^2 \underbrace{\frac{\|ux\|}{\|u\| \cdot \|x\|}}_{\leq 1} \leq \|u\|^2$$

In generale otteniamo che tale valore è raggiunto in quanto abbiamo che

$$\frac{\|Eu\|}{\|u\|} = \frac{\|uu^Tu\|}{\|u\|} = \frac{\|u\| \cdot |u^Tu|}{\|u\|} = |u^Tu| = \|u\|^2$$

Quindi il primo punto è verificato

2. Mostriamo la doppia disegualanza

La prima è banale, in quanto, per le proprietà della norma matriciale si ha che:

$$\|A + E\| \leq \|A\| + \|E\| \leq \lambda_{\max}(A) - \|u\|^2$$

Per l'altra invece possiamo:

$$\|A\| = \|A + E - E\| \leq \|A + E\| + \|E\| \xrightarrow{\text{Togliendo } \|E\|} \|A\| - \|E\| \leq \|A + E\|$$

Da cui poi si ricava la tesi.

3. Dobbiamo determinare la minima norma $\|u\|$ tale che la matrice $A + E$ sia singolare.

$$A + E = A - u^Tu$$

Nel caso di A come matrice simmetrica definita positiva abbiamo che possiamo prendere:

$$E = -\lambda_{\min} u^Tu$$

con u autovettore relativo all'autovalore λ_{\min}

Quindi possiamo prendere un vettore

$$u = \sqrt{\lambda_{\min}} v$$

Con v vettore di norma unitaria.

6. Proponi un algoritmo per risolvere:

$$\min_{X \in \mathbb{R}^{m \times p} \|B - AX\|_F} \quad \text{con } B \in \mathbb{R}^{n \times p} \text{ e } A \in \mathbb{R}^{n \times m}$$

Dove A è una matrice piena alta ($n > m$) e A rango massimo

Soluzione dell'Esercizio 6

Riprendiamo la definizione della norma di Frobenius:

$$\|B - AX\|_F^2 = \sum_{i=1}^p \|\underline{b}_i - Ax_i\|_2^2 \quad \text{dove abbiamo che } B = (\underline{b}_1, \dots, \underline{b}_p) \quad X = (x_1, \dots, x_p)$$

Se volessi minimizzare questa quantità dovremmo:

$$\min_{X \in \mathbb{R}^{m \times p}} \|B - AX\|_F^2 = \min_{\{\underline{x}_1, \dots, \underline{x}_p\} \in \mathbb{R}^p} \sum_{i=1}^p \|\underline{b}_i - A\underline{x}_i\|_2^2 = \sum_{i=1}^p \min_{\underline{x}_i \in \mathbb{R}^b} \|\underline{b}_i - A\underline{x}_i\|_2^2$$

Poi chiaramente devo mettere tutto insieme per avere $X = (\underline{x}_1, \dots, \underline{x}_p)$

Cerchiamo però di ottimizzare il tutto:

Nella scrittura dell'algoritmo, possiamo fare direttamente la stessa fattorizzazione QR :

$$(A, \underline{b}_1, \underline{b}_2, \dots, \underline{b}_p) = (A, B) \Rightarrow P_m \cdots P_2 P_1 (A, B) = (R_1, \hat{B})$$

Dove abbiamo che

$$\hat{B} = (\hat{b}_1, \hat{b}_2, \dots, \hat{b}_p) = (Q^T \underline{b}_1, \dots, Q^T \underline{b}_p)$$

Adesso basta risolvere i vari sistemi lineari:

$$R\underline{x}_i = \hat{b}_{i(1:m)} \quad \text{con } i \in \{1, \dots, p\}$$

Con matlab possiamo direttamente risolvere $RX = \hat{B}_{(1:m)}$ senza dover fare i cicli

Lo scopo di quest'esercizio era non dover fare tutti i sistemi lineari, poi ci sarebbe da calcolare il costo computazionale

7. Data $A \in \mathbb{R}^{n \times m}$ con $n \leq m$, proponi un algoritmo che permetta di determinare il rango numerico di A

Soluzione dell'Esercizio 7

Ci basta determinare la fattorizzazione QR di A e individuare il numero di elementi sulla diagonale di R_1 che sono maggiori di \mathbf{u} (`eps` della macchina) in valore assoluto (quello determina il rango di A), quindi:

$$A = (Q_1 \quad Q_1) \begin{pmatrix} R_1 \\ 0 \end{pmatrix} \Rightarrow |R_{i,i}| > \mathbf{u}$$

In caso andrebbe benissimo modificare il testo dell'esercizio per poterlo semplificare, non ha senso cambiare il testo dell'esercizio se poi si ha che ci allontaniamo dalla soluzione.

Ovviamente quest'esercizio è risolto in maniera sintetica, poi andrebbe spiegato meglio nei minimi dettagli

8. Risolvere i seguenti punti:

1. Descrivi l'algoritmo QR per $A \in \mathbb{R}^{n \times m}$, con $n \geq m$, $rg(A) = m$ e il suo problema di minimo:

$$\min_{x \in \mathbb{R}^n} \|b - Ax\|$$

2. Siano

$$A_1 = \begin{pmatrix} A \\ a^T \end{pmatrix} \in \mathbb{R}^{(n+1) \times m} \quad \text{e} \quad b_1 = \begin{pmatrix} b \\ \beta \end{pmatrix} \in \mathbb{R}^{n+1} \quad \text{con } A \text{ e } b \text{ come sopra}$$

Proponi un algoritmo che risolva:

$$\min_{x \in \mathbb{R}^m} \|b_1 - A_1 x\|$$

che sfrutti quanto fatto nel punto precedente.

Soluzione dell'Esercizio 8

Per il primo punto ok, basta guardare la teoria.

Per il secondo, sfruttiamo la fattorizzazione QR fatta nel punto precedente, quindi supponiamo di avere Q e R :

$$\left\| \begin{pmatrix} b \\ \beta \end{pmatrix} - \begin{pmatrix} A \\ a^T \end{pmatrix} x \right\| = \left\| \begin{pmatrix} b \\ \beta \end{pmatrix} - \begin{pmatrix} QR \\ a^T \end{pmatrix} x \right\| = \left\| \begin{pmatrix} b \\ \beta \end{pmatrix} - \begin{pmatrix} Q & R \\ \beta & 1 \end{pmatrix} \begin{pmatrix} R \\ a^T \end{pmatrix} x \right\|$$

Sapendo che la matrice diagonale con Q in alto a sinistra e 1 in basso a destra è ortogonale, in quanto lo è Q , possiamo fare il giochetto che avevamo fatto precedentemente:

$$\begin{aligned}\left\| \begin{pmatrix} b \\ \beta \end{pmatrix} - \begin{pmatrix} A \\ a^T \end{pmatrix} x \right\| &= \left\| \begin{pmatrix} b \\ \beta \end{pmatrix} - \begin{pmatrix} Q & R \\ 0 & 1 \end{pmatrix} \begin{pmatrix} R \\ a^T \end{pmatrix} x \right\| = \left\| \begin{pmatrix} Q & R \\ 0 & 1 \end{pmatrix} \left[\begin{pmatrix} Q^T & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} b \\ \beta \end{pmatrix} - \begin{pmatrix} R \\ a^T \end{pmatrix} x \right] \right\| \\ &= \left\| \begin{pmatrix} Q^T b \\ \beta \end{pmatrix} - \begin{pmatrix} R \\ a^T \end{pmatrix} x \right\| = \left\| \begin{pmatrix} \hat{b} \\ \beta \end{pmatrix} - \begin{pmatrix} R \\ a^T \end{pmatrix} x \right\|\end{aligned}$$

Graficamente parlando abbiamo che:

$$\begin{pmatrix} b \\ \beta \end{pmatrix} = \begin{pmatrix} | \\ . \end{pmatrix} \quad \begin{pmatrix} R \\ a^T \end{pmatrix} = \begin{pmatrix} R_1 \\ 0 \\ a^T \end{pmatrix} = \begin{pmatrix} \nabla \\ 0 \\ - \end{pmatrix}$$

Noi però vogliamo annullare l'ultima riga della matrice che moltiplica x , cioè vogliamo:

$$\begin{pmatrix} \nabla \\ 0 \\ - \end{pmatrix} \mapsto \begin{pmatrix} \nabla \\ 0 \\ 0 \end{pmatrix}$$

Nel fare ciò abbiamo diverse scelte:

- 1. Givens:** Possiamo fare una matrice di Givens che per la prima riga e per l'ultima riga, quindi possiamo scegliere $\cos \theta$ e $\sin \theta$ per la prima riga e avere:

$$\begin{pmatrix} \cos \theta & & \sin \theta \\ -\sin \theta & I_{n-1} & \cos \theta \end{pmatrix} \begin{pmatrix} R \\ a^T \end{pmatrix} = \begin{pmatrix} \cos \theta & & \sin \theta \\ -\sin \theta & I_{n-1} & \cos \theta \end{pmatrix} \begin{pmatrix} \nabla \\ 0 \\ - \end{pmatrix} = \begin{pmatrix} \times & \times & \cdots & \times \\ 0 & \times & \cdots & \times \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \times \\ \underline{0} & \cdots & \cdots & \underline{0} \\ 0 & \times & \cdots & \times \end{pmatrix}$$

E poi andare avanti con le altre righe:

- Possiamo utilizzare una matrice di permutazione per portare l'ultima riga in cima alla matrice:

$$\Pi \begin{pmatrix} R \\ a^T \end{pmatrix} = \Pi \begin{pmatrix} \nabla \\ 0 \\ - \end{pmatrix} = \begin{pmatrix} - \\ \nabla \\ 0 \end{pmatrix}$$

Per poi utilizzare le matrici di Givens per ottenere nuovamente una matrice triangolare superiore.

Nel fare ciò dobbiamo assicurarci che anche b_1 abbia le stesse permutazioni. Abbiamo poi la sicurezza che il tutto funzioni perché la matrice che otteniamo è una matrice Hessenberg superiore, quindi con Givens tutto funziona

- Siamo nella situazione in cui abbiamo:

$$\left\| \hat{b}_1 - \begin{pmatrix} R \\ a^T \end{pmatrix} x \right\| = \left\| \hat{b}_1 - \begin{pmatrix} \nabla \\ 0 \\ - \end{pmatrix} x \right\| = \left\| \hat{b}_1 - \hat{A}x \right\| \quad \text{con } \hat{A} = \begin{pmatrix} \nabla \\ 0 \\ - \end{pmatrix}$$

Andiamo a vedere cosa otteniamo con l'equazione normale:

$$\hat{A}^T \hat{A}x = \hat{A}^T \hat{b}$$

Iniziamo prima con il fare i prodotti, per la matrice abbiamo che

$$\begin{aligned}\hat{A}^T \hat{A} &= (R_1^T \ 0 \ a) \begin{pmatrix} R_1 \\ 0 \\ a^T \end{pmatrix} = R_1^T R_1 + 0^T 0 + aa^T = R_1^T R_1 + aa^T \\ &= (\Delta \ 0 \ |) \begin{pmatrix} \nabla \\ 0 \\ - \end{pmatrix} = (\Delta)(\nabla) + 0 + (\square)_{rg(\square)=1} = (\Delta)(\nabla) + (\square)_{rg(\square)=1}\end{aligned}$$

Per il termine noto abbiamo che:

$$\hat{A}^T \hat{b} = (R_1^T \quad 0 \quad a) \begin{pmatrix} \hat{b} \\ \beta \end{pmatrix} = R_1^T \hat{b} + a\beta$$

Definiamo quello che otteniamo come $R_1^T \hat{b} + a\beta = d$, allora otteniamo che:

$$(R^T R_1 + aa^T)x = d$$

Poiché abbiamo una matrice con una modifica di rango 1 possiamo usare la formula di Sherman Morrison e otteniamo che:

$$x = (R_1^T R_1)^{-1}d - (R_1^T R_1)^{-1}a(1 + a^T (R_1^T R_1)^{-1}a)^{-1}a^T (R_1^T R_1)^{-1}d$$

A questo punto dobbiamo però poi risolvere dei sistemi lineari:

$$(R_1^T R_1)^{-1}w_1 = d \quad (R_1^T R_1)^{-1}w_2 = a$$

Che sappiamo poi rivelarsi in tutto nella risoluzione di quattro sistemi lineari, due triangolari superiori e due triangolari inferiori. Quindi abbiamo che il costo computazionale di questa seconda parte è pari a $4m^2$ + le somme tra vettori più dei conti minori

9. È dato un sistema lineare $Ax = b$ con $A \in \mathbb{R}^{2n \times 2n}$ non singolare, con questa forma:

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

Proporre una procedura che risolvi il sistema lineare sfruttando tale struttura.

Soluzione dell'Esercizio 9

Notiamo che per la struttura di A possiamo dividere il sistema lineare in due parti. Infatti ponendo:

$$x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \quad b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \quad \Rightarrow \quad \begin{pmatrix} S & uv^T \\ 0 & S^T \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \Leftrightarrow \begin{cases} Sx_1 + uv^T x_2 = b_1 \\ S^T x_2 = b_2 \end{cases}$$

Determiniamo la fattorizzazione LU di S^T . Sia tale fattorizzazione:

$$S^T = \Pi^T LU$$

Allora otteniamo che:

$$S^T x_2 = b_2 \Leftrightarrow \Pi^T L U x_2 = b_2 \Leftrightarrow \underbrace{L U x_2}_y = \underbrace{\Pi b_2}_{b_2} \Leftrightarrow \begin{cases} Ly = \hat{b}_2 \\ Ux_2 = y \end{cases}$$

Chiaramente dovrà poi essere tutto espresso nel dettaglio

In questo modo otteniamo x_2 , sostituendo si ottiene che:

$$Sx_1 = \underbrace{b_1 - uv^T x_2}_{\hat{b}_1}$$

Come possiamo calcolarlo?

Un'idea (sbagliata dal momento che non è efficiente) è di calcolare prima la matrice uv^T e poi andare a risolvere tale sistema, ossia:

$$uv^T x_2 = \begin{pmatrix} u_1 \\ \vdots \\ u_n \end{pmatrix} (v_1 \quad \cdots \quad v_n) \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} u_1 v_1 & \cdots & u_1 v_n \\ \vdots & \ddots & \vdots \\ u_n v_1 & \cdots & u_n v_n \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = Mx_2$$

Ma come detto non è efficiente perché tutto questo è nell'ordine di $\Theta(n^2)$

Se mettessimo delle parentesi in più (ossia sfruttiamo la proprietà associativa del prodotto tra vettori) otteniamo che:

$$uv^T x_2 = u \underbrace{(v^T x_2)}_{\in \mathbb{R}} = \lambda u$$

Tutto questo è molto più efficiente in quanto siamo nell'ordine di $\Theta(3n^2)$

Conoscendo già la fattorizzazione di S^T , possiamo calcolare la fattorizzazione di S . Infatti:

$$S^T = \Pi^T LU \Rightarrow S = U^T L^T \Pi$$

Andando a sostituire otteniamo che:

$$Sx_1 = \hat{b}_1 \Rightarrow U^T \underbrace{L^T \Pi x_1}_{\hat{x}_1} \stackrel{y_1}{\sim} \Rightarrow \begin{cases} U^T y_1 = \hat{b}_1 \\ L^T \hat{x}_1 = y_1 \\ x_1 = \Pi^T \hat{x}_1 \end{cases}$$

10. Sia $A \in \mathbb{R}^{n \times n}$ tridiagonale e non singolare:

1. Descrivi l'algoritmo LU supponendo che esista in modo che abbia costo computazionale pari a $\Theta(n)$
2. Descrivi l'algoritmo per il problema

$$\min_{x \in \mathbb{R}^n} \left\| b - \begin{pmatrix} A \\ \gamma e_n^T \end{pmatrix} x \right\| \quad \text{con } \gamma \neq 0 \text{ e } b \in \mathbb{R}^{n+1}$$

in modo che abbia costo computazionale pari a $\Theta(n)$ e che utilizzi il punto 1

Soluzione dell'Esercizio 10

Per il primo punto basta descrivere l'algoritmo di Thomas, cioè come si ottengono L e U

Per il secondo punto: abbiamo che la matrice è del tipo:

$$\begin{pmatrix} A \\ \gamma e_n^T \end{pmatrix} = \begin{pmatrix} \times & \times & 0 & \cdots & \cdots & 0 \\ \times & \times & \times & \ddots & & \vdots \\ 0 & \times & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \times \\ 0 & \cdots & \cdots & 0 & \times & \times \\ 0 & \cdots & \cdots & \cdots & 0 & \times \end{pmatrix}$$

È importante sapere come è fatta la matrice per capire come lavorarci

Proviamo a sviluppare come facevamo negli esercizi precedenti otteniamo che:

$$\left\| b - \begin{pmatrix} A \\ \gamma e_n^T \end{pmatrix} x \right\| = \left\| b - \begin{pmatrix} LU \\ \gamma e_n^T \end{pmatrix} x \right\|$$

Però una volta arrivati qua ci blocciamo, in quanto non possiamo raccogliere L rispetto a come facevamo precedentemente.

Possiamo però utilizzare la formula di Sherman Morrison sull'equazione normale:

$$(A^T, \gamma e_n) \begin{pmatrix} A \\ \gamma e_n^T \end{pmatrix} x = (A^T, \gamma e_n) b \Rightarrow (A^T A + \gamma^2 e_n e_n^T) x = (A^T, \gamma e_n) b$$

Se poi vogliamo sostituire A con LU , otteniamo che:

$$\underbrace{(U^T L^T LU)}_H + \underbrace{\gamma e_n e_n^T}_U x = \underbrace{(A^T, \gamma e_n)}_{\hat{b}} b$$

Ponendo poi $\gamma^2 e_n = u$ e $e_n = v$ otteniamo:

$$x = H^{-1} \hat{b} - H^{-1} u (1 - v^T H^{-1} u)^{-1} v^T H^{-1} \hat{b}$$

Poi andrebbero risolti dei sistemi lineari interni e andare avanti.

Chiaramente poi andrà spiegato ogni passaggio e il costo computazionale

11. Sia $A \in \mathbb{R}^{n \times n}$ non singolare:

1. Descrivi un algoritmo per determinare $A = QH$ con Q ortogonale e H Hessenberg Superiore scrivendone anche il costo computazionale

2. Sfrutta l'algoritmo precedente per il problema:

$$\min_{x \in \mathbb{R}^n} \left\| b - \begin{pmatrix} A \\ \gamma e_n^T \end{pmatrix} x \right\|$$

in modo che sia stabile (questo da già il metodo risolutivo)

3. Sia $A_1 = A + ve_n^T$ con $A = QH$ come in 1. Come si ottiene la fattorizzazione di $A_1 = Q_1 H_1$

Soluzione dell'Esercizio 11

Risolviamo l'esercizio punto per punto.

Per il primo punto ci basta azzerare gli elementi posizionati sotto la prima diagonale sotto quella principale, cioè quelli in blu:

$$\begin{pmatrix} \times & \times & \cdots & \times & \times & \times \\ \times & \times & \cdots & \times & \times & \times \\ \textcolor{blue}{\times} & \times & \cdots & \times & \times & \times \\ \vdots & \ddots & \ddots & \vdots & \vdots & \vdots \\ \textcolor{blue}{\times} & \cdots & \textcolor{blue}{\times} & \times & \times & \times \\ \textcolor{blue}{\times} & \cdots & \cdots & \textcolor{blue}{\times} & \times & \times \end{pmatrix}$$

Possiamo utilizzare Householder a parte dalla seconda:

$$\begin{pmatrix} 1 & \\ & P_1 \end{pmatrix} A = \begin{pmatrix} \times & \times & \cdots & \times \\ \times & \times & \cdots & \times \\ 0 & \times & \cdots & \times \\ \vdots & \vdots & & \vdots \\ 0 & \times & \cdots & \times \end{pmatrix}$$

E poi possiamo proseguire per ottenere la matrice che stiamo cercando:

$$\underbrace{\begin{pmatrix} I_{n-2} & \\ & \hat{P}_{n-2} \end{pmatrix} \begin{pmatrix} I_{n-3} & \\ & \hat{P}_{n-3} \end{pmatrix} \cdots \begin{pmatrix} I_2 & \\ & \hat{P}_2 \end{pmatrix}}_{Q^T} \begin{pmatrix} 1 & \\ & P_1 \end{pmatrix} A = H \Rightarrow A = QH$$

Per il secondo punto, il problema può essere riscritto come:

$$\left\| b - \begin{pmatrix} QH \\ \gamma e_n^T \end{pmatrix} x \right\| = \left\| b - \underbrace{\begin{pmatrix} Q & \\ & 1 \end{pmatrix}}_{\tilde{Q}} \begin{pmatrix} H \\ \gamma e_n^T \end{pmatrix} x \right\| = \left\| \tilde{Q} \left(\tilde{Q}^T b - \begin{pmatrix} H \\ \gamma e_n^T \end{pmatrix} x \right) \right\| = \left\| \tilde{Q}^T b - \begin{pmatrix} H \\ \gamma e_n^T \end{pmatrix} \right\| x$$

Ma H è Hessenberg superiore, quindi dobbiamo eliminare gli elementi sotto la diagonale principale, possiamo usare Givens e abbiamo finito.

Poi ci sarebbero da risolvere un sistema triangolare superiore e da calcolare il costo computazionale

Notiamo che è esattamente quanto fatto in precedenza con la fattorizzazione QR solo che al posto di una triangolare avevamo una Hessenberg superiore

Per il terzo punto abbiamo che:

$$A_1 = A + ve_n^T$$

Dobbiamo trovare una fattorizzazione $Q_1 H_1$. Per quanto fatto nei punti precedenti abbiamo che:

$$A_1 = A + ve_n^T = QH + ve_n^T$$

Notiamo però che ve_n^T è un matrice di rango 1 che va a modificare solo l'ultima colonna, cioè:

$$QH + ve_n^T = (\square) + (0|)$$

Sapendo che Q è ortogonale, possiamo raccoglierla e otteniamo:

$$Q(H + Q^T ve_n^T)$$

Abbiamo sostanzialmente finito perché:

$$Q_1 = Q \quad \text{e} \quad H_1 = H + \underbrace{Q^T v e_n^T}_{v_1} = H + v_1 e_n^T = (\langle \nabla \rangle) + (0|) = (\langle \nabla \rangle)$$

Se avessi avuto un vettore diverso da e_n^T allora questa cosa non avrei potuto farla

12. Date $H \in \mathbb{R}^{(n+1) \times n}$ Hessenberg superiore e con rango massimo e $b \in \mathbb{R}^{n+1}$, è dato il problema:

$$\min_{x \in \mathbb{R}^n} \|b - Hx\|$$

1. Descrivi una procedura per determinare x
2. Descrivi come determinare $\|b - Ax\|$ senza determinare esplicitamente x

Soluzione dell'Esercizio 12

Per il primo punto basta vedere gli esercizi precedenti

Pr il secondo punto vogliamo usare Givens:

$$H = Q \begin{pmatrix} R_1 \\ 0 \end{pmatrix} \Rightarrow \|b - Hx\|^2 = \left\| Q^t b - \begin{pmatrix} R_1 \\ 0 \end{pmatrix} x \right\|^2$$

Tuttavia, sapendo che $Q = (Q_1, Q_2)$:

$$\|b - Hx\|^2 = \left\| \begin{pmatrix} Q_1^T b \\ Q_2^T b \end{pmatrix} - \begin{pmatrix} R_1 \\ 0 \end{pmatrix} x \right\|^2 = \underbrace{\|Q_1^T b - R_1 x\|^2}_0 + \|Q_2^T b\|^2 = \|Q_2^T b\|^2$$

Tutto questo funziona perché siamo in norma Euclidea o Norma 2 e perché tutto è al quadrato

Quindi otteniamo che: $\|b - Hx\| = \|Q_2^T b\|$

13. Sia $X \in \mathbb{R}^{m \times n}$ con $m \geq n$ e sia $\|\cdot\|$ la norma indotta da quella euclidea. Sia quindi $B \in \mathbb{R}^{(n+m) \times (n+m)}$ definita come:

$$B = \begin{pmatrix} I_m & X \\ \underline{0} & Y \end{pmatrix} \quad \text{con } \underline{0} \in \mathbb{R}^{n \times m}, Y \in \mathbb{R}^{n \times n} \text{ non singolare}$$

1. Scrivi esplicitamente l'inversa di B
2. Supponendo $Y = I_n$ metti in relazione $\kappa(B)$ con $\|X\|$
3. Descrivi una procedura per ottenere una base ortogonale per B

14. Sia $\alpha \in \mathbb{R}$ e $A \in \mathbb{R}^{n \times n}$ definita come $A_{j,j} = 3$, $A_{i,j} = (-1)^{i+j}\alpha^2$ per $i = j + 1$ e $A_{i,j}$ per $i = j - 1$

1. Sapendo che gli autovalori sono tutti reali, dai informazioni sufficienti affinché gli autovalori sono tutti reali
2. Scrivere un algoritmo che approssimi l'autovalore più vicino all'origine

Soluzione dell'Esercizio 14

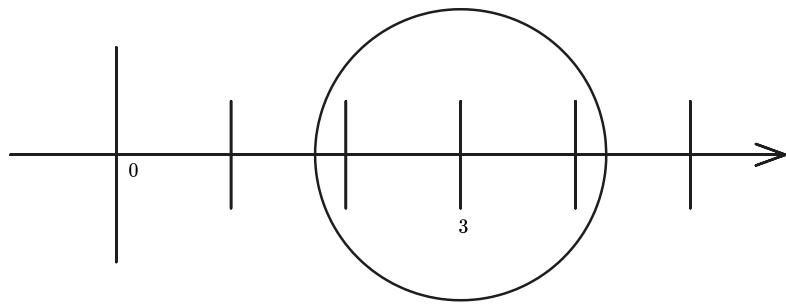
Come prima cosa andiamo a scrivere esplicitamente la matrice A :

$$A = \begin{pmatrix} 3 & (-1)^{i+j}\alpha & & & \\ (-1)^{i+j}\alpha^2 & 3 & (-1)^{i+j}\alpha & & \\ & (-1)^{i+j}\alpha^2 & \ddots & \ddots & \\ & & \ddots & \ddots & (-1)^{i+j}\alpha \\ & & & (-1)^{i+j}\alpha^2 & 3 \end{pmatrix}$$

Per avere trovare facilmente la soluzione possiamo utilizzare i dischi di Gershgorin:

$$\mathcal{G}_1 = \{|z - 3| \leq |\alpha|\} \quad \mathcal{G}_n = \{|z - 3| \leq |\alpha^2|\} = \{|z - 3| \leq \alpha^2\} \quad \mathcal{G}_i = \{|z - 3| \leq \alpha^2 + |\alpha|\} \forall i \in \{2, \dots, n-1\}$$

Siamo quindi nella situazione in cui:



Quindi affinché tutti gli autovalori sono positivi, devono essere verificate queste tre condizioni:

$$\begin{cases} |\alpha| < 3 \\ \alpha^2 < 2 \\ \alpha^2 + |\alpha| < 3 \end{cases} \quad \text{Per comodità possiamo supporre } \alpha > 0 \Rightarrow \begin{cases} \alpha < 3 \\ \alpha < \sqrt{3} \\ \alpha^2 + \alpha < 3 \end{cases}$$

Andiamo a risolvere quest'ultima diseguaglianza:

$$\alpha^2 + \alpha < 3 \Rightarrow \alpha^2 + \alpha - 3 < 0 \Leftrightarrow \alpha = \frac{-1 \pm \sqrt{1+12}}{2} = -\frac{1}{2} \pm \frac{\sqrt{13}}{2} \Rightarrow -\frac{1}{2} - \frac{\sqrt{13}}{2} < \alpha < -\frac{1}{2} + \frac{\sqrt{13}}{2}$$

Ora mettendo tutto insieme abbiamo che:

$$|\alpha| < \frac{\sqrt{13}}{2} - \frac{1}{2}$$

Con questi valori ho tutti autovalori positivi

Per il secondo punto abbiamo due scelte:

- La prima è quella di utilizzare il metodo delle potenze inverse
- La seconda è quella di utilizzare il metodo delle potenze inverse shiftate, prendendo come valore di shift $\sigma = 3 - (\frac{\sqrt{13}}{2} - \frac{1}{2})$
Questa è la via che porta migliori risultati perché sapendo già quale è il valore minimo che può assumere l'autovalore, posso partire direttamente da quello, quindi il metodo convergerà più in fretta.

15. Sia $A \in \mathbb{R}^{n \times n}$ definita come:

$$A = \begin{pmatrix} 4 & 1 & 0 & \frac{1}{2} \\ 1 & 3 & 0 & 1 \\ 0 & 0 & 2 & 1 \\ \frac{1}{2} & 1 & 1 & 3 \end{pmatrix}$$

1. Stima $\text{cond}(A) = \kappa(A)$
2. Approssima l'autovalore più vicino all'origine usando $\sigma = 0$ oppure un altro valore σ scelto opportunamente e confronta la velocità di convergenza

Soluzione dell'Esercizio 15

Sappiamo che A è simmetrica (quindi gli autovalori sono reali) e a diagonale principale
Possiamo quindi dire che

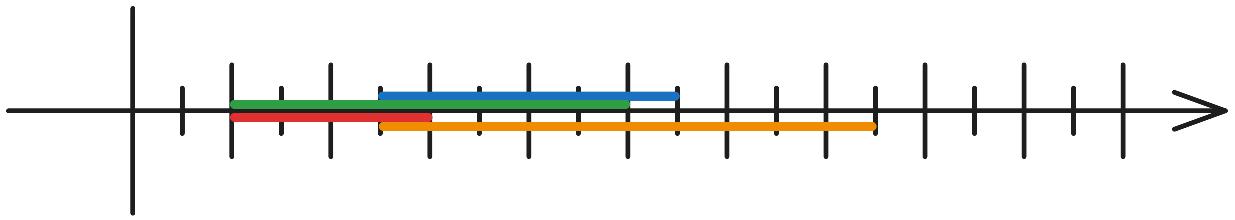
$$\kappa(A) = \frac{|\lambda_{\max}|}{|\lambda_{\min}|}$$

(Se non fosse stata simmetrica non sarebbe stato vero)

Stimiamo quindi l'intervallo spettale, in tal modo possiamo avere informazioni sugli autovalori della matrice
Possiamo usare i dischi di Gershgorin:

$$\mathcal{G}_1 = \left\{ |z - 4| \leq \frac{3}{2} \right\} \quad \mathcal{G}_2 = \{ |z - 3| \leq 2 \} \quad \mathcal{G}_3 = \{ |z - 2| \leq 1 \} \quad \mathcal{G}_4 = \left\{ |z - 5| \leq \frac{5}{2} \right\}$$

Graficamente otteniamo che:



In questo modo otteniamo quindi che:

$$Spec(A) \subseteq \left[1, \frac{15}{2} \right]$$

Dai dischi abbiamo anche che:

$$\lambda_{max} \leq \frac{15}{2} \quad \text{e} \quad \lambda_{min} \geq 1 \quad \kappa(A) \leq \frac{15}{2}$$

Per il secondo punto basta applicare il metodo delle potenze inverse traslate (prima con $\sigma = 0$, poi con $\sigma \neq 0$)

Ovviamente il metodo va descritto nella prova scritta e il costo computazionale va messo solo se viene chiesto

Sappiamo che per il metodo la cosa importante è quella di risolvere il sistema lineare:

$$y = (A - \sigma I)^{-1} x^{(k+1)}$$

Se $\sigma = 0$ allora abbiamo il semplice metodo delle potenze inverse, e la convergenza dipende da:

$$\left| \frac{\lambda_2(A^{-1})}{\lambda_1(A^{-1})} \right|$$

Sapendo poi che gli autovalori dell'inversa di una matrice sono gli inversi degli autovalori della matrice stessa, si ha che:

$$\left| \frac{\lambda_2(A^{-1})}{\lambda_1(A^{-1})} \right| = \left| \frac{\lambda_{n-1}(A)}{\lambda_n(A)} \right|$$

Avendo fatto i dischi di Gershgorin sappiamo che $\lambda_n(A) \geq 1$, quindi possiamo prendere $\sigma = 1$. Più vicini siamo all'intervallo spettrale meglio è e possiamo prendere direttamente $\sigma = 1$ perché la probabilità che $\lambda_n = 1$ è molto remota

Osservazione: Se A non ha un autovalore semplice cioè ha molteplicità maggiore di 1, allora il metodo converge ad un autovalore dell'autospazio.

Ai fini del corso restiamo sul fatto che λ sia semplice in modulo

Quindi $\sigma = 1$ è una buona scelta e quindi la velocità di convergenza è:

$$\left| \frac{\lambda_2((A - \sigma I)^{-1})}{\lambda_1((A - \sigma I)^{-1})} \right|$$

Volendo possiamo togliere il valore assoluto in quanto sappiamo che $(A - I)$ è almeno semidefinita positiva

Se $\sigma = 1 = \lambda_{min}$ allora il metodo è finito, in quanto abbiamo che $A - \sigma I$ è singolare, quindi $\lambda_{min} = 1$ è l'autovalore più vicino all'origine

Se invece $\lambda_{min} \neq \sigma$ allora otteniamo che $A - \sigma I > 0$. Quindi otteniamo che:

$$\frac{\lambda_n(A - \sigma I)}{\lambda_{n-1}(A - \sigma I)} = \frac{\lambda_n - \sigma}{\lambda_{n-1} - \sigma}$$

Questo è vero in quanto abbiamo che:

$$A - \sigma I = X \Lambda X^T - \sigma I = X(\Lambda - \sigma I)X^T$$

quindi tutti gli autovalori sono traslati di un valore σ

Quindi con $\sigma \neq 0$ otteniamo una velocità maggiore di convergenza che è maggiore se:

$$\frac{\lambda_n - \sigma}{\lambda_{n-1} - \sigma} < \frac{\lambda_n}{\lambda_{n-1}}$$

Mostriamolo:

$$\begin{aligned} \frac{\lambda_n - \sigma}{\lambda_{n-1} - \sigma} &< \frac{\lambda_n}{\lambda_{n-1}} \Rightarrow (\lambda_n - \sigma)(\lambda_{n-1}) < (\lambda_{n-1} - \sigma)(\lambda_n) \Rightarrow \\ &\Rightarrow \lambda_n \lambda_{n-1} - \sigma \lambda_{n-1} < \lambda_{n-1} \lambda_n - \sigma \lambda_n \Rightarrow \\ &\Rightarrow -\sigma \lambda_{n-1} < -\sigma \lambda_n \Rightarrow \\ &\Rightarrow \lambda_{n-1} > \lambda_n \end{aligned}$$

Poiché è verificato,abbiamo che torna sempre.

Osservazione: Notiamo che tutto questo funziona in quanto l'intervallo spettrale è contenuto nella semiretta dei numeri reali positivi. Se non avessimo avuto una buona stima avremmo avuto un gran problema.

Quindi in generale quando cerchiamo l'autovalore più vicino all'origine, se non lo prendiamo bene, potremmo trovare un'altro autovalore diverso da quello che volevamo.

È un esempio lampante il caso in cui parte dell'intervallo spettrale è nella semiretta dei numeri negativi e parte nella semiretta dei numeri positivi.

16. Sia A la matrice definita come:

$$A = \begin{pmatrix} I_n & B \\ B^T & O_m \end{pmatrix}$$

Con $B \in \mathbb{R}^{n \times m}$ e $rg(B) = m$ e siano θ_j gli autovalori di $B^T B$

1. Caratterizzare gli autovalori di A in funzione di θ_j
2. Usando 1. proporre un algoritmo per trovare gli autovalori di A , oïù vicini all'origine

Soluzione dell'Esercizio 16

Per il primo punto, il probelma si traduce in un problema del tipo:

$$Ax = \lambda x \Rightarrow \begin{pmatrix} I_n & B \\ B^T & O_m \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \lambda \begin{pmatrix} u \\ v \end{pmatrix} \Rightarrow \begin{pmatrix} I_n & \mathbb{I} \\ \mathbb{0} & O_m \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \lambda \begin{pmatrix} u \\ v \end{pmatrix}$$

Cioè, riscrivendolo in forma di un sistema lineare:

$$\begin{cases} u + Bv = \lambda v \\ B^T y = \lambda v \end{cases} \quad \text{È un problema di autovalori}$$

Dobbiamo quindi ricavare u dalla prima equazione e sostituire (*non dalla seconda, perché non esiste l'inversa di una matrice triangolare*):

$$\begin{cases} Bv = (\lambda - 1)u \\ B^T u = \lambda v \end{cases}$$

Da cui otteniamo due casi:

$$\lambda \neq -1 \Rightarrow u = \frac{1}{\lambda - 1} Bv \Rightarrow B^T \left(\frac{1}{\lambda - 1} Bv \right) = \lambda v \Rightarrow B^T B v = \underbrace{\lambda(\lambda - 1)}_{\theta} v$$

Quindi ho trovato un modo per sfruttare gli autovalori θ_j . In questo modo sappiamo che:

$$\lambda(\lambda - 1) - \theta = 0 \Rightarrow \lambda^2 - \lambda - \theta = 0 \Rightarrow \lambda_{1,2}(\theta) = \frac{1 \pm \sqrt{1 + 4\theta}}{2}$$

Quindi per ogni θ_j autovalore di $B^T B$ abbiamo due autovalori:

$$\lambda_{1,j} = \frac{1 - \sqrt{1 + 4\theta}}{2} \quad \text{e} \quad \lambda_{2,j} = \frac{1 + \sqrt{1 + 4\theta}}{2}$$

Quindi abbiamo trovato una legge che lega gli autovalori θ_j di $B^T B$ agli autovalori $\lambda_{i,j}$ di A

In particolare abbiamo m autovalori del tipo $\lambda_{1,j}$, m autovalori del tipo $\lambda_{2,j}$ e il restante $n - m$ augovalori uguali a 1
Volendo si potrebbe osservare anche che se esistono m autovalori e sapendo che $B^T B > 0$ abbiamo che tutti gli

autovalori θ_j sono tutti positivi, quindi se esistono m autovalori di $B^T B$ allora esistono tutti gli autovalori di A . Se invece $\lambda - 1 = 0$ allora segue che $Bv = 0$, ma per ipotesi abbiamo che B ha rango massimo, quindi necessariamente $v = 0$.

In questo modo otteniamo poi che $u \in \text{Ker}(B^T)$ con $\dim(B^T) = n - m$ che è esattamente quanto volevamo.

Per il secondo punto: dal punto precedente avevamo che:

$$\text{Spec}(A) = \left\{ 1, \frac{1 + \sqrt{1 + 4\theta_j}}{2}, \frac{1 - \sqrt{1 + 4\theta_j}}{2} \right\}$$

L'autovalore più piccolo è quello della forma $\lambda_{1,j}$ dell'esercizio precedente, con θ più piccolo, quindi in tal caso dipende esclusivamente da $\theta_j = \theta_{\min}$.

Il problema diventa quindi trovare θ_{\min} .

Basta utilizzare il metodo delle potenze inverse su $B^T B$ (*qui andrebbe descritto il metodo*), quindi:

$$\lambda^* = \min \left\{ 1, \frac{1 + \sqrt{1 + 4\theta_{\min}}}{2}, \frac{1 - \sqrt{1 + 4\theta_{\min}}}{2} \right\}$$

Osservazione: La soluzione $B^T B y = x^{(k+1)}$ è possibile farla anche con $H = B^T B$ per poi determinare $L = (\text{chol})(H)$ per poi risolvere i due sistemi triangolari.

Un'altra possibilità è quella di fare la fattorizzazione QR di B in modo da ottenere poi:

$$B^T B = R_1^T Q_1^T Q_1 R_1 = R_1^T R_1$$

Che è una fattorizzazione di tipo Cholesky senza effettivamente averla calcolata.

Cosa cambia tra le due fattorizzazioni?

Sulla diagonale di R possono anche esserci termini non strettamente positivi, mentre su quella di L solo termini positivi.

17. Sia dato il sistema $(A + \alpha I)x = b$ con $A_{i,i} = 2$ e $|A_{i,j}| \leq 2$ per $|i - j| = 1$

1. Dai condizioni sufficienti per α affinché Gauss-Seidel converga
2. Descrivi l'algoritmo

Soluzione dell'Esercizio 17

Prima di procedere nell'effettivo con l'esercizio, cerchiamo di visualizzare la matrice:

$$A = \begin{pmatrix} 2 & \beta_1 & & \\ \delta_2 & \ddots & \ddots & \\ & \ddots & \ddots & \beta_{n-1} \\ & & \delta_n & 2 \end{pmatrix} \quad \text{con } |\beta_i|, |\delta_i| \leq 2$$

Per il primo punto ci basta dimostrare che la matrice è dominante diagonale, cioè:

$$\begin{cases} |2 + \alpha| > |\beta_i| + |\delta_i| & i \in \{2, \dots, n-1\} \\ |2 + \alpha| > |\beta_1| & i = 1 \\ |2 + \alpha| > |\delta_n| & i = n \end{cases}$$

Sappiamo però per ipotesi che $|\beta_i| < 2$ e $|\delta_i| < 2 \forall i$. Quindi ci basta porre:

$$\begin{cases} |2 + \alpha| > 2 + 2 & i \in \{2, \dots, n-1\} \\ |2 + \alpha| > 2 & i \in \{1, n\} \end{cases}$$

Se sono verificate queste condizioni, allora la matrice è dominante diagonale, quindi per il teorema il metodo di Gauss-Seidel è convergente. Possiamo limitarci alla prima in quanto è più restrittiva.

$$\begin{aligned} |2 + \alpha| > 4 &\Rightarrow 2 + \alpha > 4 \Rightarrow \alpha > 2 \\ &\Rightarrow -2 - \alpha > 4 \Rightarrow -\alpha > 6 \Rightarrow \alpha < -6 \end{aligned}$$

18. Sia A una matrice simmetrica definita positiva e siano $Ax = b$ e $(A + \alpha I)x = b$ due sistemi lineari.

Determinare \mathcal{I} tale che per $\alpha \in \mathcal{I}$ il metodo dei gradienti coniugati converga più velocemente

Soluzione dell'Esercizio 18

Sfruttiamo l'unico risultato che abbiamo, che è quello della convergenza asintotica.

Sappiamo che la convergenza dei gradienti coniugati (CG) dipende da $\kappa(A)$ e da $\kappa(A + \alpha I)$ e più è piccolo $\kappa(A)$, più è veloce è la convergenza. Vogliamo vedere per quali α si ha

$$\kappa(A + \alpha I) < \kappa(A)$$

Sapendo poi ché $A > 0$ abbiamo che questa condizione è equivalente a chiedere che:

$$\frac{\lambda_{\max}(A + \alpha I)}{\lambda_{\min}(A + \alpha I)} < \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)} \quad \Leftrightarrow \quad \alpha \lambda_{\min}(A) < \alpha \lambda_{\max}(A)$$

Il che è sempre vero per $\alpha > 0$ da cui segue che $\mathcal{I} =]0, +\infty[$