

MANI-LO5

Abbiamo trovato α per Richardson stazionario,
e dinamico

Nel caso dinamico precondizionato $\alpha_u = \frac{[z^{(u)}]^T r^{(u)}}{[z^{(u)}]^T A z^{(u)}}$

non precondizionato $P = I$

Implementazione di Richardson

Ciclo con Richardson

$$x^{(0)} \quad (\text{TOL}, N_{\max}) \quad r^{(0)} = b - Ax^{(0)}$$

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

Per aggiornare i passi per l'algoritmo

il ciclo

$$1. Pz^{(k)} = r^{(k)} \rightarrow \text{Ricavo } z^{(k)} \rightarrow LU \text{ 1 volta!}$$

$$2. \alpha_k = \dots$$

$$3. x^{(k+1)} = x^{(k)} + \alpha_k z^{(k)}$$

$$4. r^{(k+1)} = b - Ax^{(k+1)} = b - A(x^{(k)} + \alpha_k z^{(k)}) = b - Ax^{(k)} - \alpha_k Az^{(k)} =$$

Gradient Precondizionato

G_P Gradient non precondizionato

Se non precondizionato il primo passo è inutile

RS ~ Precondizionato $\rightarrow 3$ passi $\rightarrow 2$ inutile

Non precondizionato $\rightarrow 2$ passi $\rightarrow 2$ e 2 inutile

Metodo del gradiente coniugato \rightarrow casi difficili

\hookrightarrow Metodi di punta \rightarrow è più importante il significato
oltre ricordarselo

Partiamo dalla mortalità non precondizionata

Ricordiamo $A \rightarrow sdp \rightarrow$ ci sono altri algoritmi

$$x^{(k+1)} = x^{(k)} + \tilde{\alpha}_k p^{(k)}$$

Se no
Perché non è lo stesso di prima
 \hookrightarrow D'azione diversa

Algoritmo

Punto di partenza:

$$x^{(0)} \rightarrow r^{(0)}$$

$$p^{(0)} = r^{(0)}$$

$$1. \tilde{\alpha}_k = \frac{[p^{(k)}]^T r^{(k)}}{[p^{(k)}]^T A p^{(k)}}$$

$$2. x^{(k+1)} = x^{(k)} + \tilde{\alpha}_k p^{(k)}$$

$$3. r^{(k+1)} = r^{(k)} - \tilde{\alpha}_k A p^{(k)}$$

$$4. \beta_k = \frac{[A p^{(k)}]^T r^{(k+1)}}{[A p^{(k)}]^T p^{(k)}}$$

$$5. p^{(k+1)} = r^{(k+1)} - \beta_k p^{(k)}$$

\hookrightarrow Dimensioni a coniugate

$p^{(i)}, p^{(j)}$ A ortogonali

Ortogonalità $\rightarrow [p^{(i)}]^T p^{(j)} = 0$
nonale

A-ortogonalità $\rightarrow [A p^{(i)}]^T p^{(j)} = 0$

$[p^{(i)}]^T A p^{(j)} = 0 \forall i \neq j$

Prima $r^{(i)}$ era ortogonale a quelli
prima e quelli dopo, qui ad
ogni altra direzione.

Questo passo rende il paraboloida
a sezione ellittica più circolare,
portandoci più vicino ad

Ci toglierà da problemi
non piccoli,
non ci verrà mai
cliesto

un passo per il calcolo.

Prima erano a due ortogonali,
qui sono tutte ortogonali

Più le sezioni del paraboloidale è
circolare più vicini siamo a
fermici un passo, con le
sezioni ellittiche dobbiamo
zig-zagare.

$$\|e^{(k)}\|_A \leq \left(\frac{k(\rho^{-1}A) - 1}{k(\rho^{-1}A) + 1} \right)^k \|e^{(0)}\|_A$$

se $\rho = 1 \rightarrow O(k(A)) \rightarrow$ per il non coniugato

con il coniugato $\rightarrow O(\sqrt{k(A)}) \rightarrow$ non insignificativo

rende l'algoritmo molto più veloce.

Ci la velocità di convergenza dipende da $\sqrt{k(A)}$ non
 $k(A)$

ordine di k

In casi specifici è possibile convergere in n
finti passi.

Precondizionati, funziona ancora meglio

Definizione di $K(A)$

$$Ax = b$$

Diretti

LU+pivotting

Matrice di permutazione
non precondizionata
 $PA - LU = 0$

LU accurata $\xrightarrow{?} x \text{ è accurata}$
 ↳ No $\not\Rightarrow$

Se il problema è mal condizionato allora no

Potremo usare il numero di condizionamento $K(A)$
per determinare se è malcondizionato o no
 ↳ Se ≈ 1 bene, se $\gg 1$ è mal condizionato

Prendiamo

$$A_n \in \mathbb{R}^{n \times n} \quad A_n x_n = b_n$$

,

n aumenta

$$b_n : x_n = [1, \dots, 1]^T$$

\tilde{x}_n ↳ Soluzione ricavata

Avrà sempre la
stessa struttura

$$\begin{bmatrix} 1 & 1/2 & 1/3 & \dots \\ 1/2 & 1/3 & 1/4 & \dots \\ \vdots & \vdots & \vdots & \ddots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

$$a_{ij} = \frac{1}{i+j-2}$$

Matrice di Hilbert

sop

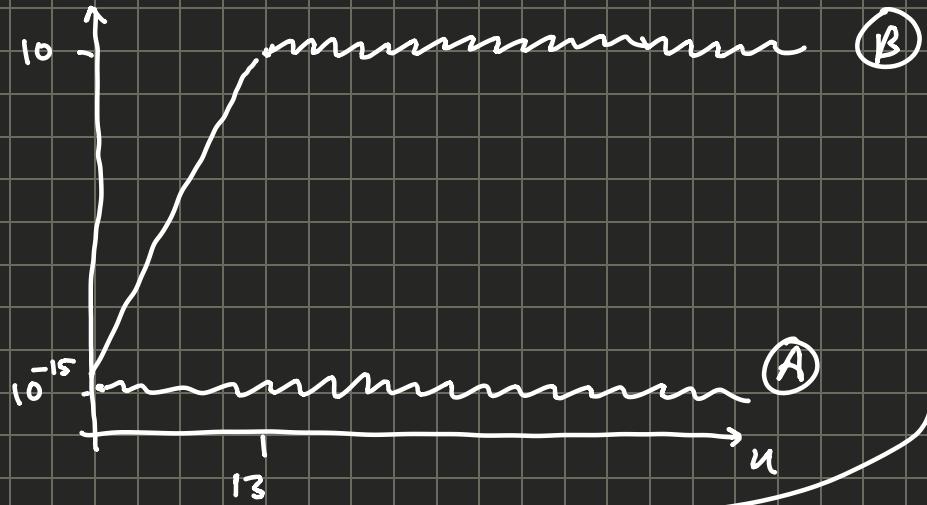
(A)

$$L_n = P_n A_n - L_n U_n \quad \text{faccio il monitoraggio di } \max_{i,j} |r_{ij}|$$

Il peggio dovrebbe
essere lo stesso vicino a 0.

Per monitorare la qualità dell'accuratezza

$$E_n = \frac{\|x_n - \tilde{x}_n\|}{\|x_n\|} \quad \textcircled{1} \rightarrow \text{di solito basta se accurato}$$



→ La matrice di Hilbert ci dà un errore innanzitutto con i metodi diretti.
→ Perché?

Quando si risolve $Ax = b$

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

Quello che stiamo risolvendo veramente

$\delta A \in \mathbb{R}^{n \times n}$

$\delta b \in \mathbb{R}^n$

\rightarrow se δA e δb sono piccole allora anche $\delta x \in \mathbb{R}^n$ sono piccole, con la matrice di Hilbert chi errori sono particolarmente grandi

Perturbazione
per varie ragioni
come floating point

Vogliamo capire la relazione fra le perturbazioni:

$$\frac{\|\delta A\|}{\|A\|}, \frac{\|\delta b\|}{\|b\|} \rightsquigarrow \frac{\|\delta x\|}{\|x\|}$$

Prendendo $\delta A = 0$ c'è la relazione $\frac{\|\delta x\|}{\|x\|} \leq k(A) \frac{\|\delta b\|}{\|b\|}$

→ determinare se il problema va bene o male

$$k(A) \geq 1$$

Se $k(A) = 3$ va bene, se $k(A) = 10^5$ va male

Se $\frac{\|\delta b\|}{\|b\|} = 10^{-10}$ vorrei $\frac{\|\delta x\|}{\|x\|} = 10^{-10}$, con $k(A) = 3$ è

dello stesso ordine e va bene, se $k(A) = 10^5$ va male

$k(A)$ agisce come fattore di controllo quando se basso

$k(A)$ agisce come amplificatore se alto

$k(A_n)$ per Hilbert $\uparrow n \rightarrow \infty$

$k(A)$ piccolo \rightarrow ben condizionato } Determinato a
 $k(A)$ grande \rightarrow male condizionato } buon senso

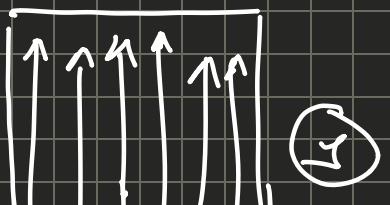
per perturbazioni di b piccole, causano errori grandi
 in x

Definizione di $k(A)$

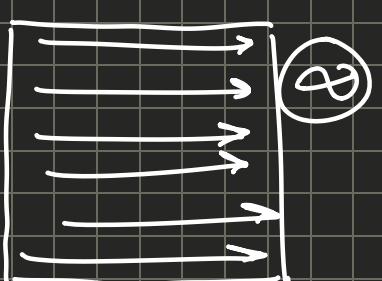
$$k(A) = \|A\| \|A^{-1}\|$$

Definizione di norme di matrice

Massimo delle somme per colonne $\rightarrow \|A\|_1 = \max_j \left(\sum_{i=1}^n |a_{ij}| \right)$



Massimo somma per righe $\rightarrow \|A\|_\infty = \max_i \left(\sum_{j=1}^n |a_{ij}| \right)$



Possiamo definire un $\kappa(A)$ per ogni norma

$$\kappa_1(A) = \|A\|_1 \|A^{-1}\|_1$$

↑ Norma 1

se è mal condizionato in uno è well condizionato in altro.

$$\kappa_2(A) = \|A\|_2 \|A^{-1}\|_2$$

$$\kappa_\infty(A) = \|A\|_\infty \|A^{-1}\|_\infty$$

Matlab : `cond(A)` \rightarrow Default : $\| \cdot \|_2$

\hookrightarrow `condest(A)` \rightarrow Default : $\| \cdot \|_1$

\hookrightarrow Condizionamento per matrici sparse

se A è sdp $\|A\|_2 < \lambda_{\max}(A)$

$$\|A^{-1}\| = \frac{1}{\lambda_{\min}(A)} \Rightarrow \kappa_2(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}$$

Se A non è singolare $K_2(A) = \sqrt{\frac{\lambda_{\max}(A^T A)}{\lambda_{\min}(A^T A)}}$

Nel caso $SA=0$

$$\frac{\|\delta x\|}{\|x\|} \leq K(A) \frac{\|\delta b\|}{\|b\|}$$

Nel caso generale, $SA \neq 0$

Se $\|SA\| \|A^{-1}\| < 1$

Tale che il denominatore è strettamente positivo

da rilevare sono:

\leftarrow se spengo $\|SA\|$ è uguale a prima.

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

$$\|e^{(k)}\|_A = \left[\frac{K(P^{-1}A) - 1}{K(P^{-1}A) + 1} \right]^k \|e^{(0)}\|_A$$

Requisiti per finire:

$P \rightarrow$ invertibile e "facile"

se $K(P^{-1}A) = 10$

se $K(P^{-1}A) = 10^{10}$ \leftarrow convergenza c'è
la soluzione ma è lenta

Nuovo requisito

$$K(P^{-1}A) \ll K(A)$$

Più dura precondizionatrice

perché agisce sul condizionamento

Sappiamo che $K(A) \geq 1$

Idealmente $\rho = A$

$\rightarrow \rho$ deve essere il più possibile vicino all'inverso di A .

Tornando al problema di Hilbert

Condizione Caratteristica

Nel
diretto

n	$K(A_n)$	\	G P Iter		GCP Iter		Scegliamo:
			Precondizionato	Precondizionato	GCP Iter	GCP Iter	
4	$O(10^4)$	$O(10^{-13})$	$O(10^{-3})$	995	$O(10^{-2})$	3	$\rho = \begin{bmatrix} a_{11} & & \\ & \ddots & \\ & & a_{nn} \end{bmatrix}$
6	$O(10^6)$	$O(10^{-10})$	$O(10^{-1})$	1813	$O(10^{-3})$	4	
8	:	:	:	:	:	:	
10	:	:	:	:	:	:	
12	:	:	:	:	:	:	
14	$O(10^{17})$	$O(10)$	$O(10^{-3})$	1579	$O(10^{-3})$	5	

(In realtà)

in realtà sono
molto più grandi

Molto più
piccolo

Molto, molto
più efficiente

Quando attivo uno schema iterativo

$$\|e^{(k)}\| \leq c \left[S \leq \frac{TOL}{10^{-10}} \right]$$

\rightarrow Nel laboratorio potremmo usare 2 schermi

