Lezione 05

Abbiamo trovato α_{opt} per Richardson stazionario e dinamico. Nel caso dinamico precondizionato:

$$\alpha_k = \frac{[z^{(k)}]^T r^{(k)}}{[z^{(k)}]^T A z^{(k)}}$$

Ciclo di Implementazione di Richardson

Prendendo come valori iniziali:

- $x^{(0)}$
- Tollerenza
- Numero massimo di iterazioni
- Restante iniziale

per k da 0,1,. . .,N_{max}

I quattro passi dell'algoritmo sono:

1. $P\underline{z}^{(k)} = r^{(k)} \rightarrow$ Ricavo $z^{(k)} \rightarrow$ facciamo la fattorizzazione LU 1 volta
2. α_k = (quello che abbiamo visto prima)
3. $x^{(k+1)} = x^{(k)} + \alpha_k z^{(k)}$
4. $r^{(k+1)} = b - Ax^{(k+1)} = b - A(x^{(k)} - \alpha_k z^{(k)}) = \underbrace{b - Ax^{(k)}}_{r^{(k)}} - \alpha_k A z^{(k)} = r^{(k)} - \alpha_k A z^{(k)}$

   Questi 4 passi sono con per con il metodo del gradiente precondizionato, nel caso del gradiente non precondizionato il primo passo è inutile, quindi sono 3 passi.

Per Richardson stazionario precondizionato servono 3 passi, perché il primo passo è inutile. Invece, nel caso non precondizionato, servono solo 2 passi perché il passo 1 e 2 sono inutili.

Metodo del Gradiente Coniugato

Non ci chiederà mai di fare il metodo del gradiente coniugato, ci chiede solo di capire perché è più veloce il fatto che è più veloce.

Partiamo dalla modalità non precondizionata, richiedendo che A sia sdp, se non lo è ci sono latri algoritmi.

Prendiamo $\{p^{(k)}\}$ come le direzione a coniugate, dove per ogni $p^{(k)}$ troviamo che è A-ortogonale ad ogni altra $p^{(k)}$. La A-ortogonalità è quando:

$$[p^{(i)}]^T p^{(j)} = 0 \rightarrow \text{Ortogonolità normale}$$
$$[Ap^{(i)}]^T p^{(j)} = 0 \rightarrow \text{A-Ortogonalità}$$
$$[p^{(i)}]^T A^T p^{(j)} = 0 \;\; \forall i \neq j$$

Le condizioni iniziali sono:

- $x^{(0)} \rightarrow r^{(0)}$
- $p(0)$ = $r^{(0)}$

Il ciclo dell'algoritmo è:

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. β_k = $\frac{[Ap^{(k)}]^T r^{(k+1)}}{[Ap^{(k)}]^T p^{(k)}}$
5. $p^{(k+1)}$ = $r^{(k+1)} - \beta_k p^{(k)}$

Prima definevamo $r^{(k)}$ come ortogonali solo con quello prima e quello dopo, ora definiamo che ogni $p^{(k)}$ sia A-ortogonale con ogni altro.

Visto che Q parabolico può esser ellittica in sezione, l'aggiunta di A rende la sezione più circolare, riducendo il numero di passi necessari per arrivare al minimo del paraboloide, perché ci servono meno zig-zag.

L'errore di questo metodo sarà:

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

Se $P = I$ allora per il non-coniugato il fattore di convergenza è funzione di $K(A)$ solamente.
Con il coniugato invece sarà funzione di $\sqrt{K(A)}$, questa differenza non è in-significativa, rendendo l'algoritmo molto più veloce.
In casi specifici è possibile che converga in n(dimensione del sistema) passi.

Con il precondizionatore la velocità aumenta ancora di più.

Definizione di K(A)

Per il sistema $Ax = b$ usando il metodo diretto di LU+pivoting, PA-LU = 0.

Sappiamo che LU è accurata, questo implica che x anche lei è accurata?
No, se il problema è malcondizionato allora la matrice x non sarà accurata.

Possiamo usare il numero di condizionamento $K(A)$ per determinare prima che iniziamo a fare il ciclo, per vedere se il problema è malcondizionato o no.

Se K(A) ha valore basso, allora è ben condizionato, invece se è molto grande lo consideriamo come mal condizionato.

Esempio di matrice malcondizionata

Prendiamo n crescente e definiamo il sistema:

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

Dove A_n sarà la matrice:

$$\begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots \\ \vdots & \vdots & \vdots & \end{bmatrix}$$

Cioè ogni elemento è $a_{ij} = \frac{1}{i+j+1}$

b_n sarà il fattore che da soluzione $x_n = [1, \ldots, 1]^T$
$\tilde{x_n}$ sarà la soluzione ricavata.

In un mondo ideale, l'errore con il metodo della fattorizzazione dovrebbe esser 0 per ogni elemento.

Se controlliamo in base all'accuratezza, con l'equazione:

$$E_n = \frac{|x_n - \tilde{x}_n|}{|x_n|}$$

Mappando l'errore massimo per elemento e questo errore di accuratezza in base alla dimensione del sistema troviamo che:

Per n ≥ 13, l'errore di accuratezza sarà ≥ 10, cioè un errore del 1000%.

La matrice di Hilbert (quella che stiamo guardando) da un errore immenso con i metodi diretti. Perche?

Il perché

Quando scriviamo Ax=b
Quello che stiamo scrivendo veramente è:

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

Dove $\delta A \in \mathbb{R}^{n \times n}$ e $\delta b \in \mathbb{R}^n$, sono perturbazioni per varie ragioni come l'errore floating point.

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

Vogliamo capire la relazione tra le perturbazioni:

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

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

$K(A) \geq 1$, sempre.

Se K(A) = 3, va bene, invece se K(A) = 10^5 va male.
Il rapporto tra le perturbazioni rimane simile quindi va bene.

Quando $K(A)$ è basso agisce come fattore di contenimento, invece quando è altro agisce come amplificatore dell'errore.

Nel caso di Hilbert $K(A_n)$ aumenta più n$\rightarrow \infty$

Calcolo di K(A)

Nel modo più semplice:

$$K(A) = |A||A^{-1}|$$

Diverse norme di matrici

La norma tipo 1 di una matrice equazione:

$$|A|_1 = max_j \left(\sum_{i=1}^{n} |a_{ij}| \right)$$

Cioè e il massimo delle somma assolute di ogni colonna.

La norma tipo ∞ di una matrice equazione:

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

Cioè e il massimo delle somma assolute di ogni colonna.

Ogni norma di matrice genera il suo valore $K(A)$, cioè:

$$K_1(A) = |A|_1 |A^{-1}|_1$$
$$K_2(A) = |A|_2 |A^{-1}|_2$$
$$K_\infty(A) = |A|_\infty |A^{-1}|_\infty$$

Tutti questi valori sono connessi tra l'un l'altro, se uno indica che il sistema è mal condizionato anche il resto lo indica.

Se A è sdp: $|A|_2 = \lambda_{max}(A)$ e $|A^{-1}|_2 = \frac{1}{\lambda_{min}(A)}$, questo allora significa che il valore di condizionamento sarà:

$$K_2(A) = \frac{\lambda_{max}(A)}{\lambda_{min}(A)}$$

Invece come abbiamo visto nella prima lezione se A non è sdp, allora:

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

Caso generale, $\delta A \neq 0$

Tale che il determinatore sia strettametne positivo, prendiamo: $|\delta A||A^{-1}| < 1$

La relazione sarà:

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

Ritorno a commentare la stima della convergenza

Ritorniamo a commentare la stima della convergenza che abbiamo detto ha valore:

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

Finora, i requisiti che abbiamo posto su P sono che sia invertibile e "facile". Il nuovo requisito per P è che:

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

Cioè che P agisca prima del condizionatore (precondizionare) per ridurre il numero di condizionamento, riducendo il numero di iterazioni che ci servono.

Sappiamo che $K(A) \geq 1$, idealmente allora $P^{-1} = A$. Per ridurre il numero di iterazioni il più possibile, P deve esser il più vicino possibile all'inversa di A.

Tornando al problema di Hilbert

Se calcoliamo il numero di iterazioni per ogni metodi visto in base alle dimensione del sistema troviamo la tabella:

Stimatori

Quando attivo un schema iterativo una condizione di arresto sarà:

$$|e^{(k)}| \leq S \leq TOL$$

Ci sono due stimatori usati generalmente, il primo è il residuo relativo che ha equazione:

$$\frac{|r^{(k)}|}{|b|} = S_1$$

Se $x^{(0)} = \underline{0}$ allora $r^{(k)} = b - Ax^{(0)} = b$

Il secondo stimatore è l'incremento tra iterazioni:

$$S_2 = |x^{(k+1)} - x^{(k)}|$$

Questo è assoluto, rispetto al primo che era relativo.