

FORMULARIO CALCOLO NUMERICO ED ELEMENTI DI ANALISI

Concetti principali

Epsilon macchina:	$\epsilon_M = \beta^{1-t}$
Errori numerici	$\frac{ x - fl(x) }{ x } \leq \frac{1}{2} \epsilon_M$
Fattore di abbattimento dell'errore	$\ e^{(k)}\ \leq (\rho(B))^k$ $\lim_{k \rightarrow \infty} e^{(k)} = 0 \Leftrightarrow \rho(B) < 1$
Errore relativo	$e_{rel} = \frac{\ x - \hat{x}\ }{\ x\ } = K(A)r_{rel}$
Residuo relativo	$r_{rel} = \frac{\ b - A\hat{x}\ }{\ b\ }$
Autovalori: indicano di quanto vengono scalati (deformati) i vettori della matrice nella direzione degli autovettori	λ (max), μ (min)
Errore nel calcolo di un autovalore	$ \lambda_1(A) - \lambda_1^{(k)} \leq C \left \frac{\lambda_2(A)}{\lambda_1(A)} \right ^{2k}$
Raggio spettrale: È il massimo valore assoluto tra gli autovalori della matrice. Determina la velocità di convergenza di metodi iterativi per la risoluzione di sistemi lineari. Se il raggio spettrale di una matrice iterativa è minore di 1, il metodo converge	$\rho = \max \lambda_i $
Numero di condizionamento: Misura la sensibilità della soluzione di un problema numerico rispetto a piccole perturbazioni nei dati di input. Se il numero di condizionamento è elevato, anche piccoli errori nei dati di input possono causare grandi errori nei risultati, rendendo il problema mal condizionato	$k_P(A) = \ A\ _P \ A^{-1}\ _P$
Condizionamento spettrale: È il rapporto tra il massimo e il minimo autovalore in modulo della matrice. Determina la stabilità e l'efficacia dei metodi numerici, in particolare per la risoluzione di sistemi lineari. Un alto condizionamento spettrale implica instabilità numerica	$k(A) = \rho(A)\rho(A^{-1})$ $k(A) = \frac{\max \lambda_i(A) }{\min \lambda_i(A) }$
Norma A (o norma energia): Immagina che la norma standard misuri la lunghezza di un vettore in uno spazio "piatto", dove tutte le direzioni sono ugualmente importanti. Quando usiamo la norma A, invece, stiamo misurando la lunghezza in uno spazio in cui le direzioni contano in modo diverso, a seconda di come A "deforma" lo spazio (la norma A misura la distanza in uno spazio dove le unità di misura non sono uniformi, ma dipendono dalla struttura imposta da A)	$\ x\ _A = \sqrt{x^T A x}$
Norma Frobenius: è l'analogo matriciale della norma euclidea di un vettore. È la radice quadrata della somma dei quadrati di tutti gli elementi della matrice. Spesso è utilizzata per misurare l'errore globale tra due matrici.	$\ A\ _F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n a_{ij} ^2}$

Se $p = 1$ $e^{(k+1)} = C e^{(k)}$

Se $p = 2$ $e^{(k+1)} \approx \frac{1}{2} \left| \frac{f''(\alpha)}{f'(\alpha)} \right| (e^{(k)})^2$

Funzioni

$[M, I] = \min(\text{vettore})$ trova il valore minimo del vettore e il suo indice

$R = \text{chol}(A)$ implementa il metodo di Cholesky

Risoluzione di sistemi lineari

Metodi diretti: la soluzione del sistema è ottenuta in un numero finito di passi.

- A è triangolare inferiore (A=L): algoritmo delle **sostituzioni in avanti** (n^2);
- A è triangolare superiore (A=U): algoritmo delle **sostituzioni indietro** (n^2);
- A è fattorizzabile: metodo di **fattorizzazione LU** ($\frac{2}{3}n^3$ per trovare LU più $2n^2$ per risolvere $LUx = b$) più MEG:

$$A = LU \quad Ax = b \quad \begin{cases} Ly = b \\ Ux = y \end{cases}$$

[condizione necessaria e sufficiente per la fattorizzazione LU: data A invertibile, la sua fattorizzazione LU esiste ed è unica se e solo se ogni sua sottomatrice è invertibile;

condizioni sufficienti per la fattorizzazione LU: A è simmetrica e definita positiva;
A è a dominanza diagonale stretta per righe;
A è a dominanza diagonale stretta per colonne.]

- A è invertibile: tecnica del **pivoting per righe**:

$$PA = LU \quad PAx = Pb \quad \begin{cases} Ly = Pb \\ Ux = y \end{cases}$$

- A è invertibile: tecnica del **pivoting totale**:

$$PAQ = LU \quad PAQQ^{-1}x = Pb \quad \begin{cases} x^* = Q^{-1}x \\ y^* = Ux^* \end{cases} \quad \begin{cases} Ly^* = Pb \\ Ux^* = y^* \\ x = Qx^* \end{cases}$$

- A è invertibile e tridiagonale: algoritmo di **Thomas** ($8n - 7$);
- A è simmetrica e definita positiva: fattorizzazione di **Cholesky** ($\frac{1}{3}n^3$):

$$A = R^T R \quad Ax = b \quad \begin{cases} R^T y = b \\ Rx = y \end{cases}$$

Su matlab: $R = \text{chol}(A)$

- A non è quadrata ($A \in \mathbb{R}^{m \times n}$, sistema sovradeterminato): **fattorizzazione QR**:

$$A = QR$$

con $Q \in \mathbb{R}^{m \times m}$ sottomatrice quadrata ortogonale

$R \in \mathbb{R}^{m \times n}$ matrice rettangolare con gli elementi sotto la diagonale principale nulli

Metodi iterativi: $x = \lim_{k \rightarrow +\infty} x^{(k)}$ in generale: $x^{(k+1)} = Bx^{(k)} + g$

se $\rho(B) < 1$ il metodo converge

- A è sparsa e diagonale dominante: decomposizione additiva (o **splitting**):

$$B = I - P^{-1}A$$

- A ha elementi diagonali non nulli: **Jacobi**:

$$P = \text{diag}(A)$$

$$P^{-1} = \text{diag}(1./\text{diag}(A))$$

- A ha elementi diagonali non nulli: **Gauss-Seidel**:

$$P = \text{tril}(A)$$

Se A è SDA, Gauss-Seidel e Jacobi convergono ad x per ogni $x^{(0)}$.

Se A è a dominanza diagonale stretta per riga e non-singolare, allora sia Jacobi, che Gauss-Seidel convergono ad x per ogni $x^{(0)}$.

Se A è non-singolare e tridiagonale (con tutti gli elementi diagonali non nulli), allora sia Jacobi, che Gauss-Seidel divergono, oppure convergono ad x (in tal caso G-S converge più velocemente di Jacobi).

- A è ben condizionata e diagonale dominante: **Richardson preconditionato**:

$$B_\alpha = I - \alpha P^{-1}A$$

Dove, se α è fisso: Richardson stazionario; se non è costante durante le iterazioni: Richardson dinamico.

Se A e P sono non-singolari, Richardson converge ad x per ogni $x^{(0)}$ se: $\alpha |\lambda_i(P^{-1}A)^2| < 2 \text{Re}\{\lambda_i(P^{-1}A)\}$

Più il numero di condizionamento è vicino ad 1, più la convergenza del metodo di Richardson stazionario è rapida.

- A è SDA: metodo del **gradiente**: metodo di Richardson dinamico senza preconditionamento con:

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

Uguale al metodo del gradiente preconditionato, ma $P=Id$.

- A è SDA: metodo del **gradiente preconditionato**: metodo di Richardson preconditionato dinamico con:

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

Dove z è il residuo preconditionato di P .

Se A e P sono SDA, il metodo del gradiente preconditionato converge ad x per ogni $x^{(0)}$.

- A è SDA: metodo del **gradiente coniugato**:

Ad ogni iterazione è necessario determinare il valore di α_k e β_k per calcolare la direzione di discesa di $p^{(k)}$.

Se A e P sono SDA, il metodo del gradiente preconditionato converge ad x per ogni $x^{(0)}$.

In aritmetica esatta, il metodo converge in al più in n iterazioni ($A \in \mathbb{R}^{n \times n}$)

Autovalori ed autovettori

Per una matrice $A \in \mathbb{C}^{n \times n}$ vi sono n autovalori e n autovettori corrispondenti.

Se $A = VDV^{-1}$ con D diagonale e V base di autovettori, allora A è diagonalizzabile.

Ogni matrice $A \in \mathbb{C}^{n \times n}$ diagonalizzabile ammette n autovettori linearmente indipendenti ed n autovalori, che si presentano come coppie di complessi coniugati.

Se A è simmetria è garantito che gli autovalori siano reali.

Se A è SDA è garantito che tutti i suoi autovalori siano positivi.

Se A è triangolare, gli autovalori sono gli elementi sulla diagonale.

Data $A \in \mathbb{C}^{n \times n}$ il problema agli autovalori di A è:

$$Ax_i = \lambda_i x_i$$

Il quoziente di Rayleigh riferito ad un vettore x_i di A è un'approssimazione di un autovalore λ_i di A , specialmente se x_i è vicino ad un autovettore:

$$R = \frac{x_i^H A x_i}{x_i^H x_i}$$

Se x_i è esattamente un autovettore di A , R è esattamente un autovalore di A .

- $A \in \mathbb{C}^{n \times n}$ e i due autovalori di modulo maggiore sono distinti: $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n|$

Metodo delle potenze: trova λ

Il metodo converge se $|\lambda_1| \gg |\lambda_2|$.

- $A \in \mathbb{C}^{n \times n}$ e i due autovalori di modulo minore sono distinti: $|\lambda_1| \geq |\lambda_2| \geq |\lambda_3| \geq \dots > |\lambda_n|$

Metodo delle potenze inverse: trova μ

Il metodo converge se $|\lambda_{n-1}| \gg |\lambda_n|$.

- **Metodo delle potenze inverse con shift:** trova l'autovalore più vicino ad un valore dato

$$A_s = A - sI$$

s è un numero complesso, diverso dagli autovalori di A , che fornisce una stima dell'autovalore da calcolare:

$$\lambda_i(A_s) = \lambda_i(A) - s$$

- $A \in \mathbb{R}^{n \times n}$ e i suoi autovalori sono reali e distinti in modulo: $|\lambda_1| > |\lambda_2| > |\lambda_3| > \dots > |\lambda_n|$

Metodo delle iterazioni QR: trova tutti gli autovalori di A (no autovettori):

$$A_k = Q_k R_k \quad A_{k+1} = R_k Q_k$$

Il metodo converge velocemente se gli autovalori sono lontani in modulo tra loro.

- **Criteri di Greshgorin:** individua l'area dove certamente si trovano gli autovalori (utile per la scelta dello shift);
- **Decomposizione ai valori singolari** di una matrice: Tecnica di diagonalizzazione di una matrice generica. Utile per problemi di regressione, riduzione dimensionale e compressione.

Equazioni non lineari

L'obiettivo è quello di approssimare lo zero $\alpha \in \mathbb{R}$ di una funzione $f(x)$ nell'intervallo $I = (a, b) \subseteq \mathbb{R}$.

La molteplicità m di uno zero α coincide con il grado della prima derivata non nulla di $f(x)$.

- Se $f(x)$ è continua in (a, b) e $f(a)f(b) < 0$: **Metodo di bisezione**

$$x^{(k)} = \frac{a^{(k)} + b^{(k)}}{2}$$

Il metodo è sempre convergente (ma lento) e garantisce la riduzione dell'intervallo a ogni iterazione.

- Se $f(x)$ è differenziabile in (a, b) : **Metodo di Newton**:

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})}$$

Se lo zero α è multiplo, il metodo converge linearmente.

Se lo zero α è semplice, $x^{(0)}$ è sufficientemente vicino (per assicurarmene posso fare qualche iterata con il metodo di bisezione, per trovare un $x^{(0)}$ adatto) e $f \in C^2(I_\alpha)$ il metodo converge quadraticamente.

- Se il metodo di Newton converge troppo lentamente o è instabile: **Metodo di Newton modificato**:

$$x^{(k+1)} = x^{(k)} - m \frac{f(x^{(k)})}{f'(x^{(k)})}$$
$$m^{(k)} = \frac{x^{(k-1)} - x^{(k-2)}}{2x^{(k-1)} - x^{(k)} - x^{(k-2)}} \text{ è la molteplicità di } \alpha$$

Se $x^{(0)}$ è sufficientemente vicino e $f \in C^2(I_\alpha) \cap C^m(I_\alpha)$ il metodo converge quadraticamente.

- **Metodi di quasi-Newton**:

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{q^{(k)}}$$

Dove la scelta di $q^{(k)}$ determina il metodo:

- **Metodo delle corde**: $q^{(k)} = \frac{f(b) - f(a)}{b - a}$

Se lo zero α è semplice, il metodo converge linearmente.

Se lo zero α è multiplo, il metodo può convergere o non convergere.

- **Metodo delle secanti**: $q^{(k)} = \frac{f(x^{(k)}) - f(x^{(k-1)})}{x^{(k)} - x^{(k-1)}}$

Se lo zero α è semplice, il metodo converge con ordine $\approx 1,6$.

Se lo zero α è multiplo, il metodo converge linearmente.

- **Iterazioni di punto fisso**

Data $\Phi: [a, b] \subseteq \mathbb{R} \rightarrow \mathbb{R}$, α è un punto fisso se e solo se $\Phi(\alpha) = \alpha$.

Per la ricerca degli zeri pongo:

$$\Phi(x) = x + F(f(x)) \quad \text{oppure} \quad \Phi(x) = x - \frac{f(x)}{f'(x)}$$

Se $\Phi \in C^0([a, b])$ esiste almeno un punto fisso $\alpha \in [a, b]$.

Se, inoltre, esiste una costante L tale che $|\Phi(x_1) - \Phi(x_2)| \leq L|x_1 - x_2| \quad \forall x_1, x_2 \in [a, b]$, allora α è unico in $[a, b]$ e l'algoritmo converge per ogni $x^{(0)} \in [a, b]$.

Più derivate di $\Phi(x)$ si annullano, maggiore è l'ordine di convergenza. In particolar modo, l'ordine di convergenza è pari all'ordine della prima derivata non nulla.

SISTEMI DI EQUAZIONI NON LINEARI

Si usa il metodo di newton oppure la funzione bfgs

Se $\det(J(F)) \neq 0$ il metodo converge quadraticamente.

Ottimizzazione numerica

Il problema di minimo non vincolato consiste nel trovare $x \in \mathbb{R}$ tale che $\Phi(x) \leq \Phi(y) \quad \forall y \in \mathbb{R}^n$, ovvero risolvere $\min_{x \in \mathbb{R}^n} \Phi(x)$.

- **Metodi derivative free:**

- **Metodo della sezione aurea:**

- Il metodo è sempre convergente.

- **Metodo di interpolazione quadratica:**

- **Metodi di discesa/line-search:** le iterazioni si muovono nella direzione che riduce maggiormente il valore della funzione; viene quindi scelta una direzione d e una lunghezza del passo α_k :

$$x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}$$

- **Algoritmo di backtracking:** determina un valore adatto di α_k :

$$\alpha_k = \rho \alpha_k \text{ dove } \rho \in \left[\frac{1}{10}, \frac{1}{2} \right]$$

- **Metodo del gradiente:**

$$d^{(k)} = -\nabla \Phi(x^{(k)})$$

- Il metodo converge linearmente.

- **Metodo del gradiente coniugato:**

$$d^{(k+1)} = -\nabla \Phi(x^{(k+1)}) + \beta_k d^{(k)} \quad \text{dove } \beta_k \text{ può assumere diversi valori (vedi pag 109-110)}$$

- Il metodo converge super linearmente ($1 < p < 2$).

- **Metodo di Newton:**

$$d^{(k+1)} = -\left(H_{\Phi}(x^{(k)})\right)^{-1} - \nabla \Phi(x^{(k)})$$

Se Φ è sufficientemente regolare, $x^{(0)}$ è sufficientemente vicino a x e $\det(H_{\Phi}(x^{(k)})) \neq 0$, il metodo converge quadraticamente.

- **Metodi di quasi-Newton: Metodo BFGS:**

$$d^{(k)} = -B^{(k)} \nabla \Phi(x^{(k)})$$

Dove B è un'approssimazione dell'inversa dell'hessiana.

Il metodo converge super linearmente.

Per verificare che A sia SDA:

```
if isequal(A,A')
    disp('A è simmetrica');
    na = size(A,1);

    %calcolo det sottomatrici (criterio di Sylvester)
    for i = 1:na
        if (det(A(1:i,1:i)) > 0)
            if(i == na)
                disp('A è DP')
            end

            else
                error('A non è DP')
            end
        end
    end

else
    disp('A non è simmetrica');
end
```