

LA BIBBIA DEFINITIVA DI CALCOLO NUMERICO

Prof. Federico Piazzon - Università di Padova, A.A. 2024/25

TUTTI I 45 ARGOMENTI DEL GLOSSARIO COMPLETO

INDICE COMPLETO - 45 ARGOMENTI

PARTE I: RAPPRESENTAZIONE E ARITMETICA MACCHINA (1-5)

1. Sistema binario e sistema floating-point IEEE754 dei numeri macchina
2. Errori assoluto e relativo, errore di rappresentazione
3. Operazioni macchina, loro (non) proprietà, precisione macchina
4. Stabilità di un algoritmo, stabilità delle operazioni macchina e cancellazione numerica
5. Problema matematico, buona posizione

PARTE II: CONDIZIONAMENTO (6-7)

6. Condizionamento numerico assoluto e relativo di un problema ben posto
7. Numero di condizionamento di una matrice e stima dell'errore relativo della soluzione di un sistema lineare con termine noto affetto da errore (con dimostrazione). Cosa succede se perturbiamo anche la matrice (no dimostrazione)

PARTE III: ALGEBRA LINEARE - METODI DIRETTI (8-10)

8. Algoritmi di sostituzione avanti e sostituzione indietro, condizioni per l'applicabilità
9. Fattorizzazione LU senza pivoting: algoritmo, limitazioni alla sua applicabilità e problematiche numeriche
10. Algoritmo LU con pivoting parziale per righe "idea" del pivoting e sua motivazione, output dell'algoritmo (no dettagli dell'algoritmo)

PARTE IV: CONTRAZIONI E METODI ITERATIVI (11-16)

11. Punti fissi, lemma delle contrazioni con dimostrazione
12. Metodi iterativi lineari stazionari per soluzione di $Ax=b$: idea generale, condizioni per la consistenza, possibili vantaggi, risultato di convergenza generale che si deduce dal lemma delle contrazioni
13. Metodo di Richardson: definizione, analisi della convergenza tramite norma e tramite proprietà spettrali (serie di Neumann)
14. Precondizionamento del metodo di Richardson e metodi di Jacobi e di Gauss Seidel
15. Lemma dei cerchi di Gerschgorin e matrici strettamente diagonalmente dominanti, applicazione alla convergenza di Jacobi

16. Criteri di arresto per metodi iterativi per la soluzione di sistemi lineari: residuo relativo e step

PARTE V: SISTEMI SOVRADETERMINATI E MINIMI QUADRATI (17-19)

17. Sistemi sovradeterminati esistenza e unicità della soluzione, minimi quadrati come generalizzazione del concetto di soluzione
18. Teorema delle proiezioni ortogonali ed equazioni normali con dimostrazione
19. Metodo QR per la soluzione delle eq normali

PARTE VI: EQUAZIONI NON LINEARI (20-25)

20. Radici di un eq non lineare: condizioni per l'unicità, condizioni per l'esistenza e metodo di bisezione (con dimostrazione)
21. Radici semplici e radici di molteplicità maggiore di 1, condizionamento del calcolo di radici
22. Metodo di Newton: euristica e definizione. Teorema di convergenza locale per radici semplici con dimostrazione
23. Test di arresto per il metodo di Newton
24. Newton come metodo iterativo: mappa di iterazione, versione del lemma delle contrazioni con $g: [a,b] \rightarrow \mathbb{R}$, e applicazione alla convergenza di Newton per radici semplici
25. Caso di radici multiple: convergenza lineare locale del metodo di Newton

PARTE VII: INTERPOLAZIONE POLINOMIALE (26-34)

26. Problema generale dell'interpolazione di dati e funzioni su uno spazio lineare di funzioni continue: linearità e sistema di Vandermonde
27. Invertibilità della matrice di Vandermonde (con dimostrazione nel caso algebrico e base monomiale)
28. Base dei polinomi di Lagrange definizione uso derivazione e calcolo
29. Stabilità dell'interpolazione rispetto a perturbazioni in norma uniforme: costante di Lebesgue
30. Convergenza del polinomio interpolante: teoremi di Weierstrass e di Jackson, stima di Lebesgue considerazioni qualitative che ne discendono
31. Nodi di Chebyshev e di Chebyshev Lobatto vs nodi equispaziati
32. Calcolo della base di Lagrange e sua valutazione con algoritmi di algebra lineare
33. Base dei polinomi di Chebyshev, qualche proprietà utile
34. Formula di rappresentazione dell'errore di interpolazione

PARTE VIII: APPROSSIMAZIONE AI MINIMI QUADRATI (35-40)

35. Approssimazione ai minimi quadrati condizioni per poter applicare il teorema delle proiezioni ortogonali
36. Prodotti scalari e matrici simmetriche definite (o semidefinite positive) matrice Gramiana e teorema di Pitagora
37. Calcolo di basi ortogonali e identità di Parseval
38. Versione generale del teorema delle proiezioni ortogonali (no dim perché uguale a quella già fatta)
39. Approssimazione prodotta con il teorema delle proiezioni ortogonali, nucleo di riproduzione

40. Stima alla Lebesgue dell'errore di approssimazione prodotta con il teorema delle proiezioni ortogonali

PARTE IX: QUADRATURA NUMERICA (41-45)

- 41. Quadratura numerica: idea generale, quadratura interpolatoria, grado di esattezza, equazione dei momenti e pol di Lagrange
 - 42. Formule del trapezio e della parabola: calcolo dei pesi
 - 43. Rappresentazione dell'errore per formule del trapezio e della parabola semplici
 - 44. Stabilità della quadratura, importanza dei pesi positivi e conseguente necessità delle formule composte
 - 45. Formule composte dei trapezi e delle parabole (Cavalieri Simpson) definizione e rappresentazione dell'errore
-

PARTE I: RAPPRESENTAZIONE E ARITMETICA MACCHINA

1. SISTEMA BINARIO E SISTEMA FLOATING-POINT IEEE754

Definizione 1.1 (Sistema binario)

Il sistema binario è un sistema di numerazione posizionale in base 2. Un numero reale x può essere rappresentato in base 2 come:

$$x = \text{sign}(x) \cdot \sum_{i=-\infty \text{ to } +\infty} d_i \cdot 2^i$$

dove $d_i \in \{0, 1\}$ sono le cifre binarie e $\text{sign}(x) \in \{-1, +1\}$.

Conversione da decimale a binario

Parte intera: Divisione successiva per 2, annotando i resti **Parte frazionaria:** Moltiplicazione successiva per 2, annotando le parti intere

Esempio: $(0.0625)_{10} = (0.0001)_2$

$$\begin{array}{lcl} 0.0625 \times 2 = 0.125 & \rightarrow & 0 \\ 0.125 \times 2 = 0.250 & \rightarrow & 0 \\ 0.250 \times 2 = 0.500 & \rightarrow & 0 \\ 0.500 \times 2 = 1.000 & \rightarrow & 1 \\ 0.000 \times 2 = 0.000 & \rightarrow & 0 \end{array}$$

Definizione 1.2 (Standard IEEE754)

Lo standard IEEE754 rappresenta i numeri reali in virgola mobile nella forma normalizzata:

$$x = (-1)^s \cdot m \cdot 2^e$$

dove:

- $s \in \{0, 1\}$ è il bit di segno
- $m = 1.f_1f_2\dots f_t$ è la mantissa normalizzata con $1 \leq m < 2$
- e è l'esponente con $e_{\min} \leq e \leq e_{\max}$
- t è il numero di bit per la mantissa frazionaria

Per IEEE754 double precision (64 bit):

- 1 bit per il segno
- 11 bit per l'esponente ($e_{\min} = -1022$, $e_{\max} = 1023$)
- 52 bit per la parte frazionaria della mantissa

Per IEEE754 single precision (32 bit):

- 1 bit per il segno
- 8 bit per l'esponente ($e_{\min} = -126$, $e_{\max} = 127$)
- 23 bit per la parte frazionaria della mantissa

Definizione 1.3 (Precisione macchina)

La precisione macchina è definita come:

$$\varepsilon_m = 2^{-t}$$

Per IEEE754:

- Double precision: $\varepsilon_m = 2^{-53} \approx 2.22 \times 10^{-16}$
- Single precision: $\varepsilon_m = 2^{-24} \approx 5.96 \times 10^{-8}$

Teorema 1.1 (Spaziatura dei numeri macchina)

Sia x un numero floating-point normalizzato con $2^k \leq |x| < 2^{k+1}$. La distanza al successivo numero rappresentabile è:

$$\text{gap}(x) = 2^{k-t+1} = 2^k \cdot 2\varepsilon_m$$

2. ERRORI ASSOLUTO E RELATIVO, ERRORE DI RAPPRESENTAZIONE

Definizione 2.1 (Errore assoluto e relativo)

Dato un valore esatto x e una sua approssimazione \tilde{x} :

- **Errore assoluto:** $e_{\text{ass}} = |x - \tilde{x}|$
- **Errore relativo:** $e_{\text{rel}} = |x - \tilde{x}|/|x|$ per $x \neq 0$

Teorema 2.1 (Errore di rappresentazione)

Sia $x \in \mathbb{R}$ un numero nell'intervallo dei numeri normalizzati e $\text{fl}(x)$ il suo rappresentante floating-point. Allora:

$$|x - \text{fl}(x)| \leq (\varepsilon_m/2) |x|$$

ovvero l'errore relativo di rappresentazione è limitato da $\varepsilon_m/2$.

Arrotondamento vs Troncamento

- **Troncamento:** Errore $\leq \varepsilon_m|x|$
 - **Arrotondamento:** Errore $\leq (\varepsilon_m/2)|x|$
-

3. OPERAZIONI MACCHINA, LORO (NON) PROPRIETÀ, PRECISIONE MACCHINA

Definizione 3.1 (Operazioni macchina)

Le operazioni aritmetiche sui numeri floating-point sono definite come:

$$\begin{aligned}x \oplus y &= \text{fl}(x + y) & x \ominus y &= \text{fl}(x - y) \\x \otimes y &= \text{fl}(x \cdot y) & x \oslash y &= \text{fl}(x / y)\end{aligned}$$

⚠ PROPRIETÀ PERSE NELLE OPERAZIONI MACCHINA:

1. **Non associatività:** $(a \oplus b) \oplus c \neq a \oplus (b \oplus c)$
2. **Non distributività:** $a \otimes (b \oplus c) \neq (a \otimes b) \oplus (a \otimes c)$
3. **Non elemento neutro esatto:** $a \oplus 0 \neq a$ per a molto piccolo
4. **Non invertibilità:** $a \ominus a \neq 0$ in generale

Esempio di non-associatività:

$$\begin{aligned}a &= 1, \quad b = \varepsilon_m, \quad c = \varepsilon_m \\(a \oplus b) \oplus c &= 1 \oplus \varepsilon_m = 1 \\a \oplus (b \oplus c) &= a \oplus \text{fl}(2\varepsilon_m) \text{ può essere diverso}\end{aligned}$$

4. STABILITÀ DI UN ALGORITMO, STABILITÀ DELLE OPERAZIONI MACCHINA E CANCELLAZIONE NUMERICA

Definizione 4.1 (Stabilità di un algoritmo)

Un algoritmo si dice **stabile** se piccole perturbazioni nei dati di input producono piccole perturbazioni nel risultato. Formalmente, se l'errore relativo sul risultato è dello stesso ordine di grandezza dell'errore relativo sui dati moltiplicato per una costante moderata.

Definizione 4.2 (Cancellazione numerica)

Si ha cancellazione numerica quando si sottrae due numeri molto vicini:

$$x - y \text{ con } x \approx y$$

Questo causa una perdita **catastrofica** di cifre significative.

Esempio di cancellazione numerica:

Calcolare $(1 + x) - 1$ per $x = 10^{-10}$:

- Teoricamente: risultato = $x = 10^{-10}$
- In macchina: $\text{fl}((1 + x) - 1) \approx 0$ (perdita completa di precisione)

Teorema 4.1 (Stabilità della somma) - CON DIMOSTRAZIONE

Siano x_1, x_2, \dots, x_n numeri floating-point. L'errore relativo nel calcolare la somma $S_n = \sum_{i=1}^n x_i$ è limitato da:

$$|S_n - \tilde{S}_n| / |S_n| \leq n\epsilon_m / (1 - n\epsilon_m) \approx n\epsilon_m \quad \text{per } n\epsilon_m \ll 1$$

Dimostrazione: Procediamo per induzione. Per $n = 2$: $\tilde{S}_2 = x_1 \oplus x_2 = (x_1 + x_2)(1 + \delta)$ dove $|\delta| \leq \epsilon_m$
Quindi: $|\tilde{S}_2 - S_2| = |S_2||\delta| \leq |S_2|\epsilon_m$

Per il passo induttivo, assumiamo che per $k < n$: $\tilde{S}_k = S_k(1 + \theta_k)$ con $|\theta_k| \leq k\epsilon_m / (1 - k\epsilon_m)$

Allora: $\tilde{S}_{k+1} = \tilde{S}_k \oplus x_{k+1} = (\tilde{S}_k + x_{k+1})(1 + \delta_{k+1}) = (S_k(1 + \theta_k) + x_{k+1})(1 + \delta_{k+1}) = S_{k+1} + S_k\theta_k + S_{k+1}\delta_{k+1} + S_k\theta_k\delta_{k+1}$

Trascurando i termini di ordine superiore e iterando si ottiene la tesi. \square

5. PROBLEMA MATEMATICO, BUONA POSIZIONE

Definizione 5.1 (Problema matematico ben posto - Hadamard)

Un problema matematico $P: X \rightarrow Y$ si dice ben posto se:

1. **Esistenza:** $\forall x \in X, \exists y \in Y$ tale che $P(x) = y$
2. **Unicità:** La soluzione è unica
3. **Stabilità:** La soluzione dipende con continuità dai dati

Se anche solo una di queste condizioni non è soddisfatta, il problema è **malposto**.

Esempi:

- **Ben posto:** $Ax = b$ con A invertibile
- **Malposto:** Derivazione numerica (instabile rispetto a perturbazioni)

PARTE II: CONDIZIONAMENTO

6. CONDIZIONAMENTO NUMERICO ASSOLUTO E RELATIVO DI UN PROBLEMA BEN POSTO

Definizione 6.1 (Condizionamento assoluto)

Sia $f: X \rightarrow Y$ un problema ben posto e sia $x_0 \in X$. Il numero di condizionamento assoluto è:

$$\kappa_{\text{ass}}(f, x_0) = \limsup_{\delta \rightarrow 0} \sup_{\|\delta x\| \leq \delta} \|f(x_0 + \delta x) - f(x_0)\| / \|\delta x\|$$

Definizione 6.2 (Condizionamento relativo)

Il numero di condizionamento relativo è:

$$\kappa_{\text{rel}}(f, x_0) = \kappa_{\text{ass}}(f, x_0) \cdot \|x_0\| / \|f(x_0)\|$$

Interpretazione:

- $\kappa_{\text{rel}} \approx 1$: Problema ben condizionato
- $\kappa_{\text{rel}} \gg 1$: Problema malcondizionato

7. NUMERO DI CONDIZIONAMENTO DI UNA MATRICE E STIMA DELL'ERRORE RELATIVO

Definizione 7.1 (Numero di condizionamento di una matrice)

Per una matrice A invertibile, il numero di condizionamento rispetto alla norma $\|\cdot\|$ è:

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

Proprietà:

- $\kappa(A) \geq 1$ sempre
- $\kappa(I) = 1$
- $\kappa(\alpha A) = \kappa(A)$ per $\alpha \neq 0$
- Per matrici simmetriche: $\kappa_2(A) = \lambda_{\max} / \lambda_{\min}$

Teorema 7.1 (Stima dell'errore per $Ax = b$) - CON DIMOSTRAZIONE COMPLETA

Sia $Ax = b$ un sistema lineare con A invertibile. Se $A(x + \delta x) = b + \delta b$ (solo termine noto perturbato), allora:

$$\|\delta x\| / \|x\| \leq \kappa(A) \cdot \|\delta b\| / \|b\|$$

Dimostrazione: Da $Ax = b$ abbiamo $x = A^{-1}b$. Da $A(x + \delta x) = b + \delta b$ abbiamo $x + \delta x = A^{-1}(b + \delta b)$. Quindi: $\delta x = A^{-1}\delta b$.

Prendendo le norme: $\|\delta x\| = \|A^{-1}\delta b\| \leq \|A^{-1}\| \cdot \|\delta b\|$

Da $Ax = b$: $\|b\| = \|Ax\| \leq \|A\| \cdot \|x\|$, quindi $\|x\| \geq \|b\| / \|A\|$

Sostituendo: $\|\delta x\| / \|x\| \leq \|A^{-1}\| \cdot \|\delta b\| / (\|b\| / \|A\|) = \|A^{-1}\| \cdot \|A\| \cdot \|\delta b\| / \|b\| = \kappa(A) \cdot \|\delta b\| / \|b\| \quad \square$

Caso generale con perturbazione di A e b

Se $(A + \delta A)(x + \delta x) = b + \delta b$, allora:

$$\|\delta x\| / \|x\| \leq \kappa(A) / (1 - \kappa(A)\|\delta A\| / \|A\|) \cdot (\|\delta b\| / \|b\| + \|\delta A\| / \|A\|)$$

purché $\kappa(A)\|\delta A\| / \|A\| < 1$.

PARTE III: ALGEBRA LINEARE - METODI DIRETTI

8. ALGORITMI DI SOSTITUZIONE AVANTI E SOSTITUZIONE INDIETRO

Algoritmo 8.1 (Sostituzione indietro)

Per risolvere $Ux = b$ con U triangolare superiore:

```
for i = n down to 1:  
    x[i] = (b[i] -  $\sum_{j=i+1}^n U[i,j] * x[j]$ ) / U[i,i]
```

Condizioni di applicabilità:

- $U[i,i] \neq 0 \forall i = 1, \dots, n$
- **Complessità:** $O(n^2)$ operazioni

Algoritmo 8.2 (Sostituzione avanti)

Per risolvere $Lx = b$ con L triangolare inferiore:

```
for i = 1 to n:  
    x[i] = (b[i] -  $\sum_{j=1}^{i-1} L[i,j] * x[j]$ ) / L[i,i]
```

Condizioni di applicabilità:

- $L[i,i] \neq 0 \forall i = 1, \dots, n$
 - **Complessità:** $O(n^2)$ operazioni
-

9. FATTORIZZAZIONE LU SENZA PIVOTING

Teorema 9.1 (Esistenza e unicità della fattorizzazione LU)

Sia $A \in \mathbb{R}^{n \times n}$. Se tutti i minori principali di ordine k ($k = 1, \dots, n-1$) sono non nulli, allora esiste un'unica fattorizzazione $A = LU$ con:

- L triangolare inferiore con diagonale unitaria ($L_{ii} = 1$)
- U triangolare superiore

Algoritmo LU senza pivoting (eliminazione gaussiana):

```
for k = 1 to n-1:  
    if A[k,k] = 0: FAIL ("pivot nullo")  
    for i = k+1 to n:  
        L[i,k] = A[i,k] / A[k,k]    // calcolo moltiplicatore  
        for j = k+1 to n:  
            A[i,j] = A[i,j] - L[i,k] * A[k,j]    // eliminazione  
U = A    // parte triangolare superiore risultante
```


Limitazioni e problematiche numeriche:

1. **Limitazione algebrica:** Richiede minori principali $\neq 0$
2. **Instabilità numerica:** Se elementi pivot sono piccoli
3. **Crescita degli elementi:** Gli elementi possono crescere esponenzialmente

Esempio di instabilità:

$$A = \begin{bmatrix} \varepsilon & 1 \\ 1 & 1 \end{bmatrix} \quad \text{con } \varepsilon \approx \varepsilon_m$$

$$L = \begin{bmatrix} 1 & 0 \\ 1/\varepsilon & 1 \end{bmatrix} \quad U = \begin{bmatrix} \varepsilon & 1 \\ 0 & 1-1/\varepsilon \end{bmatrix}$$

Gli elementi di L crescono come $1/\varepsilon \rightarrow \infty$

Complessità: $O(n^3/3)$ operazioni

10. ALGORITMO LU CON PIVOTING PARZIALE PER RIGHE

Idea del pivoting parziale

Motivazione: Evitare elementi pivot piccoli che causano instabilità numerica.

Strategia: Ad ogni passo k , scegliere come pivot l'elemento di modulo massimo nella colonna k sotto la diagonale:

$$|A[p, k]| = \max\{|A[i, k]| : i = k, \dots, n\}$$

Quindi scambiare le righe k e p .

Output dell'algoritmo

Il risultato è una fattorizzazione della forma:

$$PA = LU$$

dove:

- P è una matrice di permutazione
- L è triangolare inferiore con diagonale unitaria
- U è triangolare superiore

Vantaggi del pivoting:

1. **Stabilità numerica:** Garantisce $|L_{ij}| \leq 1$
2. **Applicabilità generale:** Funziona per tutte le matrici non singolari
3. **Controllo della crescita:** Limita la crescita degli elementi

Teorema 10.1 (Fattorizzazione con pivoting)

Per ogni matrice $A \in \mathbb{R}^{n \times n}$ non singolare, esiste una matrice di permutazione P tale che PA ammette fattorizzazione LU.

PARTE IV: CONTRAZIONI E METODI ITERATIVI

11. PUNTI FISSI, LEMMA DELLE CONTRAZIONI CON DIMOSTRAZIONE

Definizione 11.1 (Punto fisso)

Sia $g: [a,b] \rightarrow \mathbb{R}$. Un punto $\xi \in [a,b]$ si dice **punto fisso** di g se $g(\xi) = \xi$.

Definizione 11.2 (Contrazione)

Una funzione $g: [a,b] \rightarrow [a,b]$ si dice **contrazione** se:

1. $g \in C[a,b]$
2. $\exists L \in (0,1): |g(x) - g(y)| \leq L|x - y| \quad \forall x,y \in [a,b]$ (condizione di Lipschitz con costante < 1)

Lemma 11.1 (Lemma delle contrazioni - Banach) - CON DIMOSTRAZIONE COMPLETA

Sia $g: [a,b] \rightarrow [a,b]$ una contrazione con costante $L \in (0,1)$. Allora:

1. Esiste **unico** punto fisso $\xi \in [a,b]$
2. La successione $x_{k+1} = g(x_k)$ converge a $\xi \quad \forall x_0 \in [a,b]$
3. **Stima a priori:** $|x_k - \xi| \leq L^k |x_0 - \xi|$
4. **Stima a posteriori:** $|x_k - \xi| \leq L^k / (1-L) |x_1 - x_0|$

Dimostrazione: Unicità: Se ξ_1, ξ_2 sono punti fissi: $|\xi_1 - \xi_2| = |g(\xi_1) - g(\xi_2)| \leq L|\xi_1 - \xi_2|$ Poiché $L < 1$, necessariamente $\xi_1 = \xi_2$.

Esistenza: La successione $\{x_k\}$ è di Cauchy: $|x_{k+1} - x_k| = |g(x_k) - g(x_{k-1})| \leq L|x_k - x_{k-1}| \leq \dots \leq L^k |x_1 - x_0|$

Per $m > n$: $|x_m - x_n| \leq |x_m - x_{m-1}| + \dots + |x_{n+1} - x_n| \leq (L^{m-1} + \dots + L^n) |x_1 - x_0| \leq L^n / (1-L) |x_1 - x_0| \rightarrow 0$ per $n \rightarrow \infty$

Quindi $\{x_k\}$ converge a qualche ξ . Per continuità di g : $\xi = \lim x_{k+1} = \lim g(x_k) = g(\lim x_k) = g(\xi)$

Convergenza: $|x_{k+1} - \xi| = |g(x_k) - g(\xi)| \leq L|x_k - \xi| \rightarrow L^k |x_0 - \xi| \rightarrow 0$ □

12. METODI ITERATIVI LINEARI STAZIONARI PER SOLUZIONE DI $AX=B$

Definizione 12.1 (Schema iterativo lineare stazionario)

Per risolvere $Ax = b$:

$$x^{(k+1)} = Bx^{(k)} + c$$

dove:

- $B \in \mathbb{R}^{n \times n}$ è la **matrice di iterazione**
- $c \in \mathbb{R}^n$ è il **vettore di iterazione**

Condizioni per la consistenza

Lo schema è **consistente** se il punto fisso x^* soddisfa $Ax^* = b$:

$$x^* = Bx^* + c \Rightarrow x^* = (I - B)^{-1}c$$

E deve valere $Ax^* = b$, quindi: $A(I - B)^{-1}c = b$, ovvero $c = (I - B)A^{-1}b$.

Forma standard: $M^{-1}(Mx - Ax) = M^{-1}(b - Ax)$, quindi:

- $B = I - M^{-1}A$
- $c = M^{-1}b$

dove M è detta **matrice di preconditionamento**.

Possibili vantaggi dei metodi iterativi:

1. **Matrici sparse:** Non alterano la sparsità
2. **Grandi dimensioni:** Memoria $O(n)$ vs $O(n^2)$ per metodi diretti
3. **Precisione controllabile:** Si può fermare quando si raggiunge la precisione desiderata
4. **Parallelizzazione:** Alcuni metodi si parallelizzano facilmente

Teorema 12.1 (Risultato generale di convergenza)

Il metodo iterativo $x^{(k+1)} = Bx^{(k)} + c$ converge per ogni $x^{(0)}$ se e solo se:

$$\rho(B) < 1$$

dove $\rho(B)$ è il raggio spettrale di B .

Dimostrazione: Segue dal lemma delle contrazioni applicato in norma matriciale compatibile. \square

13. METODO DI RICHARDSON: DEFINIZIONE, ANALISI DELLA CONVERGENZA

Definizione 13.1 (Metodo di Richardson)

$$x^{(k+1)} = x^{(k)} + \alpha(b - Ax^{(k)}) = (I - \alpha A)x^{(k)} + \alpha b$$

dove $\alpha > 0$ è il **parametro di rilassamento**.

La matrice di iterazione è $B = I - \alpha A$.

Analisi della convergenza tramite norma

Condizione necessaria e sufficiente: $\|I - \alpha A\| < 1$ per qualche norma matriciale.

Per la norma spettrale (A simmetrica e definita positiva):

$$\rho(I - \alpha A) < 1 \Leftrightarrow 0 < \alpha < 2/\rho(A) = 2/\lambda_{\max}$$

Scelta ottimale del parametro

Se A è simmetrica e definita positiva con autovalori $0 < \lambda_{\min} \leq \dots \leq \lambda_{\max}$:

$$\alpha_{\text{opt}} = 2/(\lambda_{\min} + \lambda_{\max})$$

Con questa scelta:

$$\rho(I - \alpha_{\text{opt}} A) = (\lambda_{\max} - \lambda_{\min})/(\lambda_{\max} + \lambda_{\min}) = (\kappa(A) - 1)/(\kappa(A) + 1)$$

Serie di Neumann

Definizione 13.2 (Serie di Neumann): Per una matrice A con $\rho(I - A) < 1$:

$$\sum_{k=0}^{\infty} (I - A)^k = A^{-1}$$

Applicazione al metodo di Richardson: Se $\rho(I - \alpha A) < 1$:

$$x^* = \sum_{k=0}^{\infty} (I - \alpha A)^k \cdot \alpha b = (\alpha A)^{-1} \alpha b = A^{-1} b$$

14. PRECONDIZIONAMENTO DEL METODO DI RICHARDSON E METODI DI JACOBI E GAUSS-SEIDEL

Definizione 14.1 (Precondizionamento)

Invece di risolvere $Ax = b$, si risolve il sistema equivalente:

$$M^{-1}Ax = M^{-1}b$$

dove M è il **precondizionatore**, scelto in modo che:

- $M^{-1}A$ abbia numero di condizione migliore di A
- M^{-1} sia facile da calcolare

Metodo di Richardson preconditionato:

$$x^{(k+1)} = x^{(k)} + \alpha M^{-1}(b - Ax^{(k)})$$

Splitting classico

Si scrive $A = L + D + U$ dove:

- L = parte triangolare strettamente inferiore
- D = parte diagonale
- U = parte triangolare strettamente superiore

Metodo di Jacobi

Precondizionatore: $M = D$

$$x^{(k+1)} = (I - D^{-1}A)x^{(k)} + D^{-1}b$$

Forma componentwise:

$$x_i^{(k+1)} = (b_i - \sum_{j \neq i} a_{ij} x_j^{(k)}) / a_{ii}$$

Metodo di Gauss-Seidel

Precondizionatore: $M = D + L$

$$x^{(k+1)} = -(D + L)^{-1} U x^{(k)} + (D + L)^{-1} b$$

Forma componentwise:

$$x_i^{(k+1)} = (b_i - \sum_{j < i} a_{ij} x_j^{(k+1)} - \sum_{j > i} a_{ij} x_j^{(k)}) / a_{ii}$$

Implementazione: Non si inverte mai $(D + L)$, si risolve:

$$(D + L)x^{(k+1)} = -Ux^{(k)} + b$$

15. LEMMA DEI CERCHI DI GERSCHGORIN E MATRICI STRETTAMENTE DIAGONALMENTE DOMINANTI

Lemma 15.1 (Cerchi di Gerschgorin)

Gli autovalori della matrice $A = (a_{ij})$ sono contenuti nell'unione dei cerchi:

$$G = \cup_{i=1 \text{ to } n} \{z \in \mathbb{C} : |z - a_{ii}| \leq \sum_{j \neq i} |a_{ij}|\}$$

Definizione 15.1 (Matrice strettamente diagonalmente dominante)

A è strettamente diagonalmente dominante per righe se:

$$|a_{ii}| > \sum_{j \neq i} |a_{ij}| \quad \forall i = 1, \dots, n$$

Teorema 15.1 (Convergenza di Jacobi)

Se A è strettamente diagonalmente dominante, allora il metodo di Jacobi converge.

Dimostrazione: La matrice di iterazione è $B = -D^{-1}(L + U)$. Per Gerschgorin, gli autovalori di B sono in cerchi di raggio:

$$\sum_{j \neq i} |b_{ij}| = \sum_{j \neq i} |a_{ij}| / |a_{ii}| < 1$$

per la dominanza diagonale, quindi $\rho(B) < 1$. \square

16. CRITERI DI ARRESTO PER METODI ITERATIVI

Criterio del residuo relativo

$$\|b - Ax^{(k)}\|/\|b\| < \text{tol}_1$$

Vantaggi: Indipendente dalla scala del problema **Svantaggi:** Non garantisce precisione sulla soluzione se A è malcondizionata

Criterio dello scarto (step)

$$\|x^{(k+1)} - x^{(k)}\|/\|x^{(k)}\| < \text{tol}_2$$

Relazione con l'errore: Se B = matrice di iterazione, allora:

$$\|x^{(k)} - x^*\| \leq \|B\|/(1 - \|B\|) \cdot \|x^{(k+1)} - x^{(k)}\|$$

Criterio combinato (raccomandato)

Usare entrambi i criteri:

$$\|b - Ax^{(k)}\|/\|b\| < \text{tol}_1 \quad \text{AND} \quad \|x^{(k+1)} - x^{(k)}\|/\|x^{(k)}\| < \text{tol}_2$$

PARTE V: SISTEMI SOVRADETERMINATI E MINIMI QUADRATI

17. SISTEMI SOVRADETERMINATI: ESISTENZA E UNICITÀ DELLA SOLUZIONE

Definizione 17.1 (Sistema sovradeterminato)

Sistema $Ax = b$ con $A \in \mathbb{R}^{m \times n}$, $m > n$. Generalmente **non ha soluzione esatta**.

Definizione 17.2 (Soluzione ai minimi quadrati)

x^* è soluzione ai minimi quadrati se:

$$\|Ax^* - b\|_2 = \min\{\|Ay - b\|_2 : y \in \mathbb{R}^n\}$$

Teorema 17.1 (Esistenza e unicità)

- **Esistenza:** La soluzione ai minimi quadrati esiste sempre
- **Unicità:** Se A ha rango pieno ($\text{rank}(A) = n$), allora la soluzione è unica

Minimi quadrati come generalizzazione

Il concetto di "soluzione" viene esteso:

- **Caso classico:** $Ax = b$ ha soluzione esatta
 - **Caso sovradeterminato:** La "soluzione" è quella che minimizza $\|Ax - b\|_2$
-

18. TEOREMA DELLE PROIEZIONI ORTOGONALI ED EQUAZIONI NORMALI CON DIMOSTRAZIONE

Teorema 18.1 (Proiezioni ortogonali) - CON DIMOSTRAZIONE COMPLETA

Sia V sottospazio di dimensione finita di uno spazio con prodotto scalare. Allora per ogni f esiste unico $p^* \in V$ tale che:

$$\|f - p^*\| = \min\{\|f - p\| : p \in V\}$$

e p^* è caratterizzato dalla condizione di ortogonalità:

$$\langle f - p^*, v \rangle = 0 \quad \forall v \in V$$

Dimostrazione: Esistenza: Sia $\{v_1, \dots, v_n\}$ base di V . Cerchiamo $p^* = \sum_i c_i v_i$ che minimizza:

$$\|f - p^*\|^2 = \langle f - \sum_i c_i v_i, f - \sum_j c_j v_j \rangle$$

Derivando rispetto a c_k e uguagliando a zero:

$$\partial/\partial c_k \|f - p^*\|^2 = -2\langle f - \sum_i c_i v_i, v_k \rangle = 0$$

Questo dà: $\langle f - p^*, v_k \rangle = 0$ per $k = 1, \dots, n$, ovvero la condizione di ortogonalità.

Unicità: Se p_1^*, p_2^* sono entrambe soluzioni, allora: $\langle f - p_1^*, v \rangle = \langle f - p_2^*, v \rangle = 0 \quad \forall v \in V$ Quindi $\langle p_1^* - p_2^*, v \rangle = 0 \quad \forall v \in V$ In particolare, scegliendo $v = p_1^* - p_2^*$: $\|p_1^* - p_2^*\|^2 = 0$, quindi $p_1^* = p_2^*$.
□

Corollario 18.1 (Equazioni normali)

Nel caso $Ax = b$ con $A \in \mathbb{R}^{m \times n}$ ($m > n$), la soluzione ai minimi quadrati soddisfa:

$$A^T A x^* = A^T b$$

queste sono le **equazioni normali**.

Dimostrazione: Il sottospazio è $V = \text{Range}(A) = \{Ax : x \in \mathbb{R}^n\}$. La condizione di ortogonalità diventa: $b - Ax^* \perp V$ $A^T(b - Ax^*) = 0$ $A^T A x^* = A^T b$ □

19. METODO QR PER LA SOLUZIONE DELLE EQUAZIONI NORMALI

Problema delle equazioni normali

Le equazioni normali possono essere malcondizionate:

$$\kappa(A^T A) = \kappa(A)^2$$

Esempio: Se $\kappa(A) = 10^6$, allora $\kappa(A^T A) = 10^{12}$, perdendo 6 cifre decimali!

Fattorizzazione QR

Teorema 19.1: Ogni matrice $A \in \mathbb{R}^{m \times n}$ ($m \geq n$) con rango pieno può essere fattorizzata come:

$$A = QR$$

dove:

- $Q \in \mathbb{R}^{m \times n}$ è ortogonale ($Q^T Q = I$)
- $R \in \mathbb{R}^{m \times n}$ ha la forma $R = [R_0; 0]$ con $R_0 \in \mathbb{R}^{n \times n}$ triangolare superiore invertibile

Algoritmo QR per minimi quadrati

Input: $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $m \geq n$, $\text{rank}(A) = n$

1. Calcola $A = QR$ (fattorizzazione QR)

2. $\hat{b} = Q^T b$

3. Risolvi $R_0 x = \hat{b}_{1:n}$ (sostituzione all'indietro)

Output: x soluzione del problema ai minimi quadrati

Vantaggi del metodo QR:

1. **Migliore condizionamento:** $\kappa_2(R_0) = \kappa_2(A)$ invece di $\kappa_2(A)^2$
2. **Maggiore stabilità numerica**
3. **Evita il calcolo di $A^T A$**

Complessità: $O(mn^2 - n^3/3)$ vs $O(mn^2 + n^3/6)$ per equazioni normali

Fattorizzazione QR tramite Gram-Schmidt

Algoritmo Gram-Schmidt modificato:

```
for j = 1 to n:
    v_j = a_j
    for i = 1 to j-1:
        r_{ij} = <a_j, q_i>
        v_j = v_j - r_{ij} q_i
    r_{jj} = ||v_j||
    q_j = v_j / r_{jj}
```

PARTE VI: EQUAZIONI NON LINEARI

20. RADICI DI UN'EQUAZIONE NON LINEARE: CONDIZIONI PER L'UNICITÀ, ESISTENZA E METODO DI BISEZIONE

Problema: Trovare ξ tale che $f(\xi) = 0$

Teorema 20.1 (Esistenza delle radici) - CON DIMOSTRAZIONE

Se $f \in C[a,b]$ e $f(a)f(b) < 0$, allora esiste almeno una radice $\xi \in (a,b)$.

Dimostrazione: Teorema dei valori intermedi. \square

Condizioni per l'unicità

Se $f \in C^1[a,b]$, $f(a)f(b) < 0$ e $f'(x) \neq 0 \forall x \in [a,b]$, allora la radice è unica.

Metodo di bisezione - CON DIMOSTRAZIONE DI CONVERGENZA

Algoritmo:

```
Input:  $f \in C[a,b]$ ,  $f(a)f(b) < 0$ ,  $\text{tol} > 0$ 
while  $(b - a)/2 > \text{tol}$ :
     $c = (a + b)/2$ 
    if  $f(c) = 0$ : return  $c$ 
    if  $f(a)f(c) < 0$ :
         $b = c$ 
    else:
         $a = c$ 
return  $(a + b)/2$ 
```

Teorema 20.2 (Convergenza del metodo di bisezione)

Dopo n iterazioni, l'errore è limitato da:

$$|x_n - \xi| \leq (b_0 - a_0)/2^{(n+1)}$$

Dimostrazione: Ad ogni iterazione l'intervallo si dimezza: $[a_1, b_1] = [a_0, (a_0+b_0)/2]$ o $[(a_0+b_0)/2, b_0]$

Quindi: $b_1 - a_1 = (b_0 - a_0)/2$ Per induzione: $b_n - a_n = (b_0 - a_0)/2^n$ Poiché $\xi \in [a_n, b_n]$: $|x_n - \xi| \leq (b_n - a_n)/2 = (b_0 - a_0)/2^{(n+1)} \square$

Convergenza: Lineare con fattore $1/2$.

Vantaggi e svantaggi

 **Vantaggi:** Sempre converge, robusto, semplice  **Svantaggi:** Lento, richiede cambio di segno

21. RADICI SEMPLICI E RADICI DI MOLTEPLICITÀ MAGGIORE DI 1, CONDIZIONAMENTO

Definizione 21.1 (Molteplicità di una radice)

ξ è radice di molteplicità m di f se:

$$f(\xi) = f'(\xi) = \dots = f^{(m-1)}(\xi) = 0, \quad f^{(m)}(\xi) \neq 0$$

- Se $m = 1$: radice **semplice**
- Se $m > 1$: radice **multipla**

Condizionamento del calcolo di radici

Per radici semplici:

Il numero di condizionamento è:

$$\kappa = |\xi f''(\xi)| / |f'(\xi)|$$

Se $f'(\xi) \neq 0$, il problema è ben condizionato.

Per radici multiple:

Sono sempre **malcondizionate**! Se ξ è radice di molteplicità m :

$$\kappa \approx m / |f^{(m)}(\xi)|$$

Piccole perturbazioni in f possono causare grandi variazioni nella posizione della radice.

Esempio

$$f(x) = (x - 1)^{10}$$

La radice $\xi = 1$ ha molteplicità 10. Una piccola perturbazione $\tilde{f}(x) = (x - 1)^{10} + \varepsilon$ può spostare le radici di $O(\varepsilon^{1/10})$.

22. METODO DI NEWTON: EURISTICA E DEFINIZIONE, TEOREMA DI CONVERGENZA LOCALE

Euristica del metodo di Newton

Idea: Approssimare $f(x)$ con la retta tangente in x_k :

$$f(x) \approx f(x_k) + f'(x_k)(x - x_k)$$

La radice dell'approssimazione lineare è:

$$x_{k+1} = x_k - f(x_k) / f'(x_k)$$

Definizione 22.1 (Metodo di Newton)

$$x_{k+1} = x_k - f(x_k) / f'(x_k)$$

purché $f'(x_k) \neq 0$.

Teorema 22.1 (Convergenza locale per radici semplici) - CON DIMOSTRAZIONE COMPLETA

Sia $f \in C^2[a,b]$, $\xi \in (a,b)$ radice semplice ($f(\xi) = 0$, $f'(\xi) \neq 0$). Allora $\exists \delta > 0$ tale che se $|x_0 - \xi| < \delta$, la successione di Newton converge **quadraticamente**:

$$|x_{k+1} - \xi| \leq M |x_k - \xi|^2$$

dove $M = |f''(\xi)|/(2|f'(\xi)|)$.

Dimostrazione: Sviluppo di Taylor di f intorno a ξ :

$$f(x_k) = f(\xi) + f'(\xi)(x_k - \xi) + \frac{f''(\theta_k)}{2}(x_k - \xi)^2$$

dove $\theta_k \in (\xi, x_k)$.

Poiché $f(\xi) = 0$:

$$f(x_k) = f'(\xi)(x_k - \xi) + \frac{f''(\theta_k)}{2}(x_k - \xi)^2$$

Il passo di Newton:

$$\begin{aligned} x_{k+1} - \xi &= x_k - \xi - f(x_k)/f'(x_k) \\ &= (x_k - \xi) - [f'(\xi)(x_k - \xi) + \frac{f''(\theta_k)}{2}(x_k - \xi)^2]/f'(x_k) \\ &= (x_k - \xi)[1 - f'(\xi)/f'(x_k)] - \frac{f''(\theta_k)}{2f'(x_k)}(x_k - \xi)^2 \end{aligned}$$

Per continuità di f' , se $x_k \rightarrow \xi$: $f'(x_k) \rightarrow f'(\xi) \neq 0$, quindi il primo termine $\rightarrow 0$. Il termine dominante è:

$$|x_{k+1} - \xi| \approx |f''(\theta_k)|/(2|f'(x_k)|) |x_k - \xi|^2$$

Per δ sufficientemente piccolo, $M = \max\{|f''(x)|/(2|f'(x)|) : x \in [\xi-\delta, \xi+\delta]\}$ è finito. \square

Interpretazione geometrica

Il metodo segue le tangenti alla curva $y = f(x)$, intersecando l'asse x per trovare la prossima approssimazione.

23. TEST DI ARRESTO PER IL METODO DI NEWTON

Criterio sul residuo

$$|f(x_k)| < \text{tol}_1$$

Interpretazione: Quanto è vicino x_k ad essere una radice esatta.

Criterio sullo step

$$|x_{k+1} - x_k| < \text{tol}_2$$

Interpretazione: Variazione tra iterate successive.

Criterio combinato (raccomandato)

$$|f(x_k)| < \text{tol}_1 \quad \text{AND} \quad |x_{k+1} - x_k| < \text{tol}_2$$

Criterio relativo sullo step

$$|x_{k+1} - x_k| / |x_k| < \text{tol}_3$$

Vantaggi: Indipendente dalla scala del problema.

Considerazioni pratiche

1. **Massimo numero di iterazioni:** Evitare loop infiniti
 2. **Controllo $f'(x_k)$:** Se $|f'(x_k)| < \text{tol}_4$, fermarsi (derivata troppo piccola)
 3. **Monitoraggio:** Verificare che $\|f(x_k)\|$ stia effettivamente diminuendo
-

24. NEWTON COME METODO ITERATIVO: MAPPA DI ITERAZIONE, LEMMA DELLE CONTRAZIONI

Mappa di iterazione

Il metodo di Newton si può scrivere come:

$$x_{k+1} = g(x_k)$$

dove $g(x) = x - f(x)/f'(x)$ è la **mappa di iterazione di Newton**.

Versione del lemma delle contrazioni per $g: [a,b] \rightarrow \mathbb{R}$

Teorema 24.1: Se $g: [a,b] \rightarrow [a,b]$ e $\exists L \in (0,1): |g'(x)| \leq L \quad \forall x \in [a,b]$, allora g ha un unico punto fisso $\xi \in [a,b]$ e $x_{k+1} = g(x_k)$ converge a ξ per ogni $x_0 \in [a,b]$.

Applicazione alla convergenza di Newton

Per la mappa di Newton $g(x) = x - f(x)/f'(x)$:

$$g'(x) = 1 - [f'(x)]^2 - f(x)f''(x)/[f'(x)]^2 = f(x)f''(x)/[f'(x)]^2$$

In prossimità della radice semplice ξ : $g'(\xi) = f(\xi)f''(\xi)/[f'(\xi)]^2 = 0$

Quindi $|g'(x)| < 1$ in un intorno di ξ , garantendo convergenza locale.

Raggio di convergenza

Il **bacino di attrazione** è l'insieme degli x_0 per cui Newton converge a ξ .

- Per radici semplici: tipicamente grande
 - Per radici multiple: può essere molto piccolo
 - Dipende dal comportamento globale di f
-

25. CASO DI RADICI MULTIPLE: CONVERGENZA LINEARE LOCALE DEL METODO DI NEWTON

Teorema 25.1 (Newton per radici multiple)

Se ξ è radice di molteplicità $m > 1$, allora Newton converge **linearmente** con fattore $(m-1)/m$:

$$\lim_{k \rightarrow \infty} \frac{|x_{k+1} - \xi|}{|x_k - \xi|} = (m-1)/m$$

Dimostrazione (sketch)

Per radice di molteplicità m : $f(x) = (x - \xi)^m h(x)$ con $h(\xi) \neq 0$.

$$f'(x) = m(x - \xi)^{m-1} h(x) + (x - \xi)^m h'(x)$$

La mappa di Newton:

$$g(x) = x - f(x)/f'(x) = x - (x - \xi)h(x)/[mh(x) + (x - \xi)h'(x)]$$

Calcolando $g'(\xi)$:

$$g'(\xi) = (m-1)/m$$

Metodo di Newton modificato per radici multiple

Se la molteplicità m è nota:

$$x_{k+1} = x_k - m \cdot f(x_k)/f'(x_k)$$

Questo ripristina la convergenza **quadratica**.

Metodo alternativo (molteplicità incognita)

Definire $u(x) = f(x)/f'(x)$, quindi:

$$x_{k+1} = x_k - u(x_k)/u'(x_k)$$

Se ξ è radice di f con molteplicità m , è radice semplice di u .

PARTE VII: INTERPOLAZIONE POLINOMIALE

26. PROBLEMA GENERALE DELL'INTERPOLAZIONE: LINEARITÀ E SISTEMA DI VANDERMONDE

Problema dell'interpolazione

Dati $n+1$ punti distinti $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$, trovare $p \in \pi_n$ (spazio dei polinomi di grado $\leq n$) tale che:

$$p(x_i) = y_i, \quad i = 0, 1, \dots, n$$

Linearità del problema

L'operatore di interpolazione è **lineare**:

$$L_n[\alpha f + \beta g] = \alpha L_n[f] + \beta L_n[g]$$

dove $L_n[f]$ è il polinomio che interpola f nei nodi x_0, \dots, x_n .

Sistema di Vandermonde

Scrivendo $p(x) = a_0 + a_1x + \dots + a_nx^n$, le condizioni di interpolazione danno:

$$\begin{bmatrix} 1 & x_0 & x_0^2 & \dots & x_0^n \\ 1 & x_1 & x_1^2 & \dots & x_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^n \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{bmatrix}$$

La matrice $V \in \mathbb{R}^{(n+1) \times (n+1)}$ è la **matrice di Vandermonde**.

27. INVERTIBILITÀ DELLA MATRICE DI VANDERMONDE (CON DIMOSTRAZIONE)

Teorema 27.1 (Invertibilità di Vandermonde) - CON DIMOSTRAZIONE

La matrice di Vandermonde V è invertibile se e solo se i nodi x_0, x_1, \dots, x_n sono distinti.

Dimostrazione: Il determinante di Vandermonde è:

$$\det(V) = \prod_{\{0 \leq i < j \leq n\}} (x_j - x_i)$$

Dimostriamo per induzione su n . **Base** ($n = 1$):

$$V = \begin{bmatrix} 1 & x_0 \\ 1 & x_1 \end{bmatrix} \Rightarrow \det(V) = x_1 - x_0$$

Passo induttivo: Supponiamo vero per $n-1$. Per la matrice $(n+1) \times (n+1)$:

Sviluppiamo lungo la prima riga. L'elemento $(1,1)$ ha cofattore che è una Vandermonde $(n \times n)$ con nodi x_1, \dots, x_n :

$$\text{Cofattore}_{11} = \prod_{\{1 \leq i < j \leq n\}} (x_j - x_i)$$

Dopo calcoli (somma telescopica), si ottiene:

$$\det(V) = \prod_{\{1 \leq i < j \leq n\}} (x_j - x_i) \cdot \prod_{i=1}^n (x_i - x_0) = \prod_{\{0 \leq i < j \leq n\}} (x_j - x_i)$$

Conclusione: $\det(V) \neq 0 \Leftrightarrow x_i \neq x_j$ per $i \neq j$. \square

Corollario 27.1 (Esistenza e unicità dell'interpolazione)

Se i nodi sono distinti, esiste un unico polinomio $p \in \pi_n$ che interpola i dati.

28. BASE DEI POLINOMI DI LAGRANGE: DEFINIZIONE, USO, DERIVAZIONE E CALCOLO

Definizione 28.1 (Polinomi di Lagrange)

$$L_i(x) = \prod_{j=0, j \neq i}^n (x - x_j) / (x_i - x_j), \quad i = 0, 1, \dots, n$$

Proprietà fondamentali

1. $L_i \in \pi_n$ (grado n)
2. **Proprietà di interpolazione:** $L_i(x_j) = \delta_{ij}$ (delta di Kronecker)
3. **Partizione dell'unità:** $\sum_{i=0}^n L_i(x) = 1$

Uso: Formula di interpolazione di Lagrange

$$p(x) = \sum_{i=0}^n y_i L_i(x)$$

Derivazione

I polinomi di Lagrange sono costruiti per essere una **base duale** ai funzionali di valutazione:

- $L_i(x_i) = 1$: il polinomio vale 1 nel nodo x_i
- $L_i(x_j) = 0$ per $j \neq i$: si annulla negli altri nodi

Calcolo efficiente

Algoritmo diretto: $O(n^2)$ per valutare tutti gli L_i in un punto.

Forma baricitrica (più efficiente):

$$p(x) = \sum_{i=0}^n y_i \cdot w_i / (x - x_i) / \sum_{j=0}^n w_j / (x - x_j)$$

dove $w_i = 1 / \prod_{j \neq i} (x_i - x_j)$ sono i **pesi baricentrici**.

29. STABILITÀ DELL'INTERPOLAZIONE: COSTANTE DI LEBESGUE

Problema della stabilità

Se i dati sono perturbati: $y_i \rightarrow \tilde{y}_i = y_i + \varepsilon_i$, come cambia il polinomio interpolante?

Definizione 29.1 (Costante di Lebesgue)

$$\Lambda_n = \max_{x \in [a, b]} \sum_{i=0}^n |L_i(x)|$$

Teorema 29.1 (Stabilità dell'interpolazione)

Se $\max |\varepsilon_i| \leq \varepsilon$, allora:

$$\max_{x \in [a, b]} |\tilde{p}(x) - p(x)| \leq \Lambda_n \cdot \varepsilon$$

Dimostrazione:

$$|\tilde{p}(x) - p(x)| = |\sum_i \varepsilon_i L_i(x)| \leq \varepsilon \sum_i |L_i(x)| \leq \varepsilon \Lambda_n$$

Importanza della costante di Lebesgue

- Λ_n **piccolo**: Interpolazione stabile
- Λ_n **grande**: Piccoli errori nei dati causano grandi errori nell'interpolazione

Confronto per diversi tipi di nodi

- **Nodi equispaziati**: $\Lambda_n \sim O(2^n)$ (crescita esponenziale!)
- **Nodi di Chebyshev**: $\Lambda_n \sim O(\log n)$ (crescita logaritmica)

30. CONVERGENZA DEL POLINOMIO INTERPOLANTE: TEOREMI DI WEIERSTRASS E JACKSON

Domanda fondamentale: L'interpolazione polinomiale converge sempre all'aumentare del grado?

Teorema 30.1 (Weierstrass)

Ogni funzione continua su $[a,b]$ può essere approssimata uniformemente da polinomi:

$$\forall f \in C[a,b], \forall \varepsilon > 0, \exists p \text{ polinomio: } \|f - p\|_{\infty} < \varepsilon$$

Teorema 30.2 (Jackson)

Se $f \in C^k[a,b]$, allora esiste una sequenza di polinomi p_n di grado n tali che:

$$\|f - p_n\|_{\infty} \leq M_k \cdot \omega_k(f, (b-a)/n) / n^k$$

dove ω_k è il modulo di continuità di ordine k .

Stima di Lebesgue per l'interpolazione

L'errore di interpolazione p_n soddisfa:

$$\|f - p_n\|_{\infty} \leq (1 + \Lambda_n) \cdot \min\{\|f - p\|_{\infty} : p \in \pi_n\}$$

Considerazioni qualitative

1. **Convergenza teorica**: Possibile per ogni $f \in C[a,b]$ con scelta opportuna dei nodi
 2. **Convergenza pratica**: Dipende da Λ_n
 3. **Nodi equispaziati**: Possono dare **divergenza** (controesempio di Runge)
 4. **Nodi di Chebyshev**: Garantiscono convergenza per ogni $f \in C[a,b]$
-

31. NODI DI CHEBYSHEV E DI CHEBYSHEV-LOBATTO VS NODI EQUISPAZIATI

Fenomeno di Runge

Esempio: Funzione di Runge $f(x) = 1/(1 + 25x^2)$ su $[-1, 1]$ Con nodi equispaziati e n grande, si osservano **forti oscillazioni** vicino ai bordi.

Nodi di Chebyshev (prima specie)

$$x_j = \cos((2j + 1)\pi / (2(n + 1))), \quad j = 0, 1, \dots, n$$

Proprietà ottimali:

1. **Minimizzano asintoticamente la costante di Lebesgue**
2. Sono i **punti di massimo** dei polinomi di Chebyshev
3. **Addensamento verso i bordi:** Più punti dove servono di più






Nodi di Chebyshev-Lobatto (seconda specie)

$$x_j = \cos(j\pi/n), \quad j = 0, 1, \dots, n$$






Differenza: Includono gli estremi dell'intervallo $[-1, 1]$.

Confronto delle performance

Nodi equispaziati:

-  Facili da generare
-  Intuitivi
-  $\Lambda_n \sim O(2^n/n)$ (crescita esponenziale)
-  Fenomeno di Runge
-  Adatti solo per n piccolo

Nodi di Chebyshev:

-  $\Lambda_n \sim (2/\pi)\log(n+1)$ (crescita logaritmica)
-  Convergenza garantita per $f \in C[a,b]$
-  Evitano il fenomeno di Runge
-  Ottimali per interpolazione di grado elevato
-  Meno intuitivi da calcolare

Regola pratica: Usare sempre nodi di Chebyshev per $n > 10$.

32. CALCOLO DELLA BASE DI LAGRANGE E SUA VALUTAZIONE CON ALGORITMI DI ALGEBRA LINEARE

Calcolo diretto dei polinomi di Lagrange

Complessità: $O(n^2)$ operazioni per valutare tutti gli $n+1$ polinomi in un punto.

```
for i = 0 to n:
    L_i(x) = 1
    for j = 0 to n, j ≠ i:
        L_i(x) *= (x - x_j) / (x_i - x_j)
```

Riformulazione con algebra lineare

Idea: Calcolare i coefficienti dei polinomi di Lagrange nella base monomiale.

Se $L_j(x) = \sum_{k=0}^n a_{jk}x^k$, il problema si riduce a risolvere $n+1$ sistemi lineari:

$$V a_j = e_j, \quad j = 0, 1, \dots, n$$

dove V è la matrice di Vandermonde e e_j è il j -esimo vettore della base canonica.

Vantaggi dell'approccio algebrico:

1. **Riuso:** I coefficienti si calcolano una volta sola
2. **Stabilità:** Metodi numericamente stabili per sistemi di Vandermonde
3. **Parallelizzazione:** I sistemi si possono risolvere indipendentemente

Forma baricitrica (algoritmo più efficiente)

Algoritmo baricitrico:

1. Precalcolo: $w_i = 1/\prod_{j \neq i}(x_i - x_j)$ [$O(n^2)$ una volta]
2. Valutazione: $p(x) = [\sum y_i * w_i/(x - x_i)] / [\sum w_j/(x - x_j)]$ [$O(n)$ per punto]

Vantaggi: $O(n)$ per valutazione dopo precalcolo $O(n^2)$.

33. BASE DEI POLINOMI DI CHEBYSHEV, QUALCHE PROPRIETÀ UTILE

Definizione 33.1 (Polinomi di Chebyshev di prima specie)

$$T_n(\cos \theta) = \cos(n\theta)$$

Relazione di ricorrenza

$$T_0(x) = 1$$

$$T_1(x) = x$$

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \quad n \geq 1$$

Proprietà fondamentali

1. Ortogonalità

Su $[-1, 1]$ con peso $w(x) = 1/\sqrt{1-x^2}$:

$$\int_{-1}^1 T_m(x) T_n(x) / \sqrt{1-x^2} dx = \begin{cases} 0 & \text{se } m \neq n \\ \pi & \text{se } m = n = 0 \\ \pi/2 & \text{se } m = n > 0 \end{cases}$$

2. Proprietà di estremi

$$|T_n(x)| \leq 1 \quad \text{per } x \in [-1, 1]$$

I massimi sono raggiunti in $x_k = \cos(k\pi/n)$, $k = 0, \dots, n$.

3. Zeri

$$x_k = \cos((2k-1)\pi/(2n)), \quad k = 1, \dots, n$$

4. Proprietà di minimizzazione

Tra tutti i polinomi monici di grado n , $T_n(x)/2^{n-1}$ ha norma uniforme minima su $[-1, 1]$.

Applicazioni

1. Nodi di interpolazione ottimali
2. Approssimazione di funzioni
3. Integrazione numerica (quadratura di Gauss-Chebyshev)
4. Accelerazione della convergenza

Trasformazione di intervalli

Da $[-1, 1]$ a $[a, b]$:

$$x = ((b-a)t + (b+a))/2, \quad t \in [-1, 1]$$

34. FORMULA DI RAPPRESENTAZIONE DELL'ERRORE DI INTERPOLAZIONE

Teorema 34.1 (Errore di interpolazione) - FORMULA ESPLICITA

Sia $f \in C^{n+1}[a,b]$ e siano $x_0, \dots, x_n \in [a,b]$ nodi di interpolazione distinti. Se $p \in \pi_n$ è il polinomio interpolante, allora $\forall x \in [a,b]$:

$$f(x) - p(x) = f^{(n+1)}(\xi(x)) / ((n+1)!) \cdot \prod_{i=0}^n (x - x_i)$$

dove $\xi(x) \in (\min\{x, x_0, \dots, x_n\}, \max\{x, x_0, \dots, x_n\})$.

Dimostrazione (sketch)

Per $x \neq x_i$, consideriamo la funzione ausiliaria:

$$G(t) = f(t) - p(t) - K \prod_{i=0}^n (t - x_i)$$

dove K è scelto in modo che $G(x) = 0$.

$G(t)$ si annulla in x, x_0, \dots, x_n ($n+2$ zeri). Applicando ripetutamente il teorema di Rolle:

- $G'(t)$ ha almeno $n+1$ zeri
- $G''(t)$ ha almeno n zeri
- ...
- $G^{(n+1)}(t)$ ha almeno 1 zero ξ

Poiché $G^{(n+1)}(\xi) = f^{(n+1)}(\xi) - K(n+1)! = 0$, si ottiene la formula. \square

Conseguenze pratiche

1. Stima dell'errore

$$|f(x) - p(x)| \leq (\|f^{(n+1)}\|_{\infty}) / ((n+1)!) \cdot \max_x \left| \prod_{i=0}^n (x - x_i) \right|$$

2. Ottimizzazione dei nodi

Per minimizzare l'errore, bisogna minimizzare $\max_x \left| \prod_{i=0}^n (x - x_i) \right|$. **Soluzione:** Nodi di Chebyshev!

3. Convergenza

Se $f \in C^{\infty}[a,b]$ e $|f^{(n+1)}(x)| \leq M \cdot R^n$ per qualche M, R , allora:

$$\|f - p_n\|_{\infty} \leq (M \cdot R^n) / ((n+1)!) \cdot \max \left| \prod_{i=0}^n (x - x_i) \right| \rightarrow 0$$

PARTE VIII: APPROSSIMAZIONE AI MINIMI QUADRATI

35. APPROSSIMAZIONE AI MINIMI QUADRATI: CONDIZIONI PER APPLICARE IL TEOREMA DELLE PROIEZIONI ORTOGONALI

Setup generale

Spazio: V spazio vettoriale con prodotto scalare $\langle \cdot, \cdot \rangle$ **Sottospazio:** $W \subset V$ di dimensione finita

Obiettivo: Trovare $w^* \in W$ che minimizza $\|f - w\|$

Condizioni per l'applicabilità del teorema delle proiezioni ortogonali

Condizione 1: Spazio con prodotto scalare

V deve essere dotato di un prodotto scalare che soddisfa:

1. **Bilinearità:** $\langle \alpha f + \beta g, h \rangle = \alpha \langle f, h \rangle + \beta \langle g, h \rangle$
2. **Simmetria:** $\langle f, g \rangle = \langle g, f \rangle$
3. **Definitezza positiva:** $\langle f, f \rangle \geq 0, \langle f, f \rangle = 0 \Leftrightarrow f = 0$

Condizione 2: Sottospazio di dimensione finita

W deve avere dimensione finita $n < \infty$.

Condizione 3: Base del sottospazio

Deve esistere una base $\{\varphi_1, \dots, \varphi_n\}$ di W linearmente indipendente.

Esempi di prodotti scalari

Caso continuo

$$\langle f, g \rangle = \int_a^b f(x)g(x)w(x)dx$$

dove $w(x) > 0$ è una funzione peso.

Caso discreto

$$\langle f, g \rangle = \sum_{i=1}^m w_i f(x_i)g(x_i)$$

dove $w_i > 0$ sono pesi e x_i punti di campionamento.

Verifiche necessarie

1. **Indipendenza lineare:** La matrice Gramiana $G = [\langle \varphi_i, \varphi_j \rangle]$ deve essere definita positiva
 2. **Esistenza degli integrali/somme:** I prodotti scalari devono essere finiti
 3. **Regolarità:** Le funzioni base devono essere sufficientemente regolari
-

36. PRODOTTI SCALARI E MATRICI SIMMETRICHE DEFINITE, MATRICE GRAMIANA E TEOREMA DI PITAGORA

Definizione 36.1 (Matrice Gramiana)

Data una matrice $A \in \mathbb{R}^{m \times n}$, la matrice Gramiana è:

$$G = A^T A \in \mathbb{R}^{n \times n}$$

con elementi $G_{ij} = \langle a_i, a_j \rangle$ dove a_i sono le colonne di A.

Proprietà della matrice Gramiana

1. **Simmetria:** $G = G^T$
2. **Semidefinita positiva:** $x^T G x = \|Ax\|^2 \geq 0$
3. **Definita positiva** \Leftrightarrow A ha rango pieno
4. **Elementi diagonali:** $G_{ii} = \|a_i\|^2$

Teorema 36.1 (Teorema di Pitagora)

Se x e y sono vettori **ortogonali** rispetto a un prodotto scalare ($\langle x, y \rangle = 0$), allora:

$$\|x + y\|^2 = \|x\|^2 + \|y\|^2$$

Dimostrazione:

$$\|x + y\|^2 = \langle x + y, x + y \rangle = \langle x, x \rangle + 2\langle x, y \rangle + \langle y, y \rangle = \|x\|^2 + \|y\|^2$$

Teorema di Pitagora generalizzato

Se W è un sottospazio e $P_W(x)$ è la proiezione ortogonale di x su W , allora:

$$\|x\|^2 = \|P_W(x)\|^2 + \|x - P_W(x)\|^2$$

Interpretazione: La norma si decompone nella parte "spiegata" dalla proiezione e nella parte "residua".

Applicazione all'approssimazione

Per il problema ai minimi quadrati $\|f - \sum c_i \varphi_i\|^2$, la decomposizione di Pitagora dà:

$$\|f\|^2 = \|p^*\|^2 + \|f - p^*\|^2$$

dove p^* è l'approssimazione ottimale.

37. CALCOLO DI BASI ORTOGONALI E IDENTITÀ DI PARSEVAL

Basi ortogonali e ortonormali

- **Base ortogonale:** $\langle \varphi_i, \varphi_j \rangle = 0$ per $i \neq j$
- **Base ortonormale:** $\langle \varphi_i, \varphi_j \rangle = \delta_{ij}$

Processo di ortogonalizzazione di Gram-Schmidt**Algoritmo Gram-Schmidt classico:**

```

u1 = v1
for k = 2 to n:
    uk = vk - ∑(j=1 to k-1) proj_{uj}(vk)
    dove proj_{uj}(vk) = ⟨vk, ujj, ujj

```

Algoritmo Gram-Schmidt modificato (più stabile):

```

for k = 1 to n:
    uk = vk
    for j = 1 to k-1:
        uk = uk - ⟨uk, qjj    // qj = uj / ||uj|| normalizzati
    qk = uk / ||uk||

```

Vantaggi delle basi ortogonali

1. **Coefficienti espliciti:** $c_i = \langle f, \varphi_i \rangle / \langle \varphi_i, \varphi_i \rangle$
2. **Matrice Gramiana diagonale:** $G_{ij} = \delta_{ij} \|\varphi_i\|^2$
3. **Stabilità numerica:** Non richiede risoluzione di sistemi

Identità di Parseval

Per una base ortonormale $\{\varphi_i\}$ e $f = \sum c_i \varphi_i$:

$$\|f\|^2 = \sum_i |c_i|^2$$

Forma generale (non necessariamente base completa):

$$\|f\|^2 = \|p^*\|^2 + \|f - p^*\|^2 = \sum_i |\langle f, \varphi_i \rangle|^2 / \|\varphi_i\|^2 + \|f - p^*\|^2$$

dove $p^* = \sum_i \langle f, \varphi_i \rangle / \|\varphi_i\|^2 \varphi_i$.

Applicazione: L'identità di Parseval fornisce una decomposizione dell'energia del segnale.

38. VERSIONE GENERALE DEL TEOREMA DELLE PROIEZIONI ORTOGONALI

Teorema 38.1 (Proiezioni ortogonali - Versione generale)

Sia V uno spazio con prodotto scalare e $W \subset V$ sottospazio di dimensione finita. Allora $\forall f \in V$ esiste unico $p^* \in W$ tale che:

$$\|f - p^*\| = \min\{\|f - w\| : w \in W\}$$

p^* è caratterizzato dalla **condizione di ortogonalità**:

$$\langle f - p^*, \varphi \rangle = 0 \quad \forall \varphi \in W$$

Equivalentemente (con base $\{\varphi_1, \dots, \varphi_n\}$ di W):

$$\langle f - p^*, \varphi_i \rangle = 0, \quad i = 1, \dots, n$$

Sistema normale

Scrivendo $p^* = \sum c_i \varphi_i$, la condizione di ortogonalità dà:

$$\sum_j c_j \langle \varphi_j, \varphi_i \rangle = \langle f, \varphi_i \rangle, \quad i = 1, \dots, n$$

Ovvero: $Gc = b$ dove:

- $G = [\langle \varphi_i, \varphi_j \rangle]$ (matrice Gramiana)
- $c = [c_1, \dots, c_n]^T$ (coefficienti)
- $b = [\langle f, \varphi_1 \rangle, \dots, \langle f, \varphi_n \rangle]^T$

Proprietà della proiezione

1. **Linearità:** $P_W(\alpha f + \beta g) = \alpha P_W(f) + \beta P_W(g)$
 2. **Idempotenza:** $P_W(P_W(f)) = P_W(f)$
 3. **Conservazione:** $P_W(w) = w \quad \forall w \in W$
-

39. APPROSSIMAZIONE CON IL TEOREMA DELLE PROIEZIONI ORTOGONALI, NUCLEO DI RIPRODUZIONE

Approssimazione prodotta dal teorema

Data f e sottospazio W con base $\{\varphi_1, \dots, \varphi_n\}$, l'approssimazione ottimale è:

$$p^*(x) = \sum_{i=1}^n c_i \varphi_i(x)$$

dove i coefficienti c_i sono soluzione del sistema $Gc = b$.

Nucleo di riproduzione

Definizione 39.1: Il nucleo di riproduzione $K(x,y)$ associato al sottospazio W e base $\{\varphi_i\}$ è:

$$K(x,y) = \sum_{i=1}^n \varphi_i(x) \varphi_i(y) / \langle \varphi_i, \varphi_i \rangle$$

Proprietà di riproduzione:

$$\langle f, K(\cdot, y) \rangle = f(y) \quad \forall f \in W$$

Rappresentazione dell'approssimazione

L'approssimazione ottimale può essere scritta come:

$$p^*(x) = \int K(x,y) f(y) d\mu(y)$$

dove μ è la misura associata al prodotto scalare.

Esempi di nuclei

1. **Base monomiale su $[0,1]$:** $K(x,y) = \sum_{k=0}^n x^k y^k / (k+1)$
2. **Base trigonometrica:** $K(x,y) = \sum_k \sin(kx) \sin(ky) + \cos(kx) \cos(ky)$

Interpretazione

Il nucleo $K(x,y)$ rappresenta la "correlazione" tra i punti x e y nel sottospazio W .

40. STIMA ALLA LEBESGUE DELL'ERRORE DI APPROSSIMAZIONE

Teorema 40.1 (Stima di Lebesgue per approssimazione ai minimi quadrati)

Sia p^* l'approssimazione ai minimi quadrati di f su W . Allora:

$$\|f - p^*\| \leq \sqrt{(1 + \Lambda_n^2)} \cdot \min\{\|f - w\| : w \in W\}$$

dove Λ_n è analoga alla costante di Lebesgue per l'interpolazione.

Definizione della costante

Nel caso di approssimazione su punti discreti:

$$\Lambda_n = \max_i \sum_{j=1}^n |K_{ij}|$$

dove K è la matrice del nucleo di riproduzione valutata sui punti.

Confronto con l'interpolazione

- **Interpolazione:** $\|f - p_n\| \leq (1 + \Lambda_n)\|f - p^*\|$
- **Minimi quadrati:** $\|f - p_n\| \leq \sqrt{(1 + \Lambda_n^2)}\|f - p^*\|$

L'approssimazione ai minimi quadrati è generalmente **più stabile**.

Casi specifici

1. **Basi ortogonali:** $\Lambda_n = 1$ (stabilità ottimale)
2. **Nodi di Chebyshev:** $\Lambda_n \sim O(\log n)$
3. **Nodi equispaziati:** Λ_n può crescere esponenzialmente

Conseguenze pratiche

1. **Scelta delle basi:** Preferire basi ortogonali o quasi-ortogonali
 2. **Controllo della stabilità:** Monitorare il condizionamento della matrice Gramiana
 3. **Regolarizzazione:** In casi malcondizionati, aggiungere termini di regolarizzazione
-

PARTE IX: QUADRATURA NUMERICA

41. QUADRATURA NUMERICA: IDEA GENERALE, QUADRATURA INTERPOLATORIA, GRADO DI ESATTEZZA

Problema della quadratura numerica

Approssimare l'integrale definito:

$$I[f] = \int_a^b f(x) dx \approx Q_n[f] = \sum_{i=0}^n w_i f(x_i)$$

dove:

- x_i sono i **nodi** di quadratura
- w_i sono i **pesi** di quadratura

Definizione 41.1 (Quadratura interpolatoria)

Una formula di quadratura si dice **interpolatoria** se deriva dall'integrazione del polinomio interpolante:

$$Q_n[f] = \int_a^b p_n(x) dx$$

dove p_n è il polinomio che interpola f nei nodi x_0, \dots, x_n .

Conseguenza: I pesi sono determinati unicamente dai nodi:

$$w_i = \int_a^b L_i(x) dx$$

dove L_i sono i polinomi di Lagrange.

Definizione 41.2 (Grado di esattezza)

Una formula di quadratura ha **grado di esattezza d** se:

1. $Q_n[p] = I[p]$ per tutti i polinomi p di grado $\leq d$
2. \exists polinomio p di grado $d+1$: $Q_n[p] \neq I[p]$

Teorema 41.1

Le formule interpolatorie su $n+1$ nodi hanno grado di esattezza $\geq n$.

Equazione dei momenti

Per determinare i pesi di una formula interpolatoria:

$$\sum_{i=0}^n w_i x_i^k = \int_a^b x^k dx = (b^{k+1} - a^{k+1})/(k+1), \quad k = 0, 1, \dots, n$$

Esempio: Formula del punto medio

- Nodo: $x_0 = (a+b)/2$
 - Peso: $w_0 = \int_a^b 1 dx = b - a$
 - Formula: $\int_a^b f(x) dx \approx (b - a)f((a+b)/2)$
 - Grado di esattezza: 1
-

42. FORMULE DEL TRAPEZIO E DELLA PARABOLA: CALCOLO DEI PESI

Formula del trapezio

Nodi: $x_0 = a, x_1 = b$

Calcolo dei pesi:

$$w_0 = \int_a^b L_0(x) dx = \int_a^b (x - b)/(a - b) dx = \int_a^b (x - b)/(a - b) dx$$

Sostituendo $t = (x - a)/(b - a)$:

$$w_0 = (b - a) \int_0^1 (1 - t) dt = (b - a)[t - t^2/2]_0^1 = (b - a)/2$$

Analogamente: $w_1 = (b - a)/2$

Formula del trapezio:

$$\int_a^b f(x) dx \approx (b - a)/2 [f(a) + f(b)]$$

Grado di esattezza: 1 (esatta per polinomi di grado ≤ 1)

Formula della parabola (Simpson)

Nodi: $x_0 = a$, $x_1 = (a+b)/2$, $x_2 = b$

Calcolo dei pesi:

Sia $h = (b - a)/2$, quindi $x_1 = a + h$.

$$w_0 = \int_a^b L_0(x) dx = \int_a^b (x - x_1)(x - x_2)/((x_0 - x_1)(x_0 - x_2)) dx$$

Dopo i calcoli (sostituzioni e integrazioni):

- $w_0 = h/3 = (b - a)/6$
- $w_1 = 4h/3 = 2(b - a)/3$
- $w_2 = h/3 = (b - a)/6$

Formula di Simpson:

$$\int_a^b f(x) dx \approx (b - a)/6 [f(a) + 4f((a+b)/2) + f(b)]$$

Grado di esattezza: 3 (esatta per polinomi di grado ≤ 3)

Riepilogo pesi:

- **Trapezio:** $(1/2, 1/2) \times (b - a)$
- **Simpson:** $(1/6, 4/6, 1/6) \times (b - a)$

43. RAPPRESENTAZIONE DELL'ERRORE PER FORMULE DEL TRAPEZIO E DELLA PARABOLA SEMPLICI

Errore della formula del trapezio

Teorema 43.1: Se $f \in C^2[a,b]$, allora:

$$\int_a^b f(x) dx - Q_{\text{trap}}[f] = -(b - a)^3/12 \cdot f''(\xi)$$

per qualche $\xi \in (a,b)$.

Dimostrazione (sketch): Usando la formula dell'errore di interpolazione:

$$f(x) - p_1(x) = f''(\eta(x))/2! \cdot (x - a)(x - b)$$

Integrando:

$$\int_a^b [f(x) - p_1(x)] dx = \int_a^b f''(\eta(x))/2 \cdot (x - a)(x - b) dx$$

Per il teorema della media integrale:

$$= f''(\xi)/2 \cdot \int_a^b (x - a)(x - b) dx = f''(\xi)/2 \cdot [-(b - a)^3/6] = -(b - a)^3/12 \cdot f''(\xi)$$

Errore della formula di Simpson

Teorema 43.2: Se $f \in C^4[a,b]$, allora:

$$\int_a^b f(x) dx - Q_{\text{Simpson}}[f] = -(b-a)^5/90 \cdot f^{(4)}(\xi)$$

per qualche $\xi \in (a,b)$.

Osservazione: Il fattore $90 = 2^4 \cdot 3 \cdot 5/4!$ deriva dalla struttura dell'errore di interpolazione.

Considerazioni sull'errore

1. **Trapezio:** Errore $\propto (b-a)^3$, dipende da f''
2. **Simpson:** Errore $\propto (b-a)^5$, dipende da $f^{(4)}$
3. **Per intervalli piccoli:** Simpson converge molto più velocemente
4. **Funzioni lisce:** Entrambi i metodi sono efficaci

Esempio numerico

Per $f(x) = e^x$ su $[0,1]$:

- **Trapezio:** Errore ≈ -0.264
- **Simpson:** Errore ≈ -0.0003

44. STABILITÀ DELLA QUADRATURA, IMPORTANZA DEI PESI POSITIVI E CONSEGUENTE NECESSITÀ DELLE FORMULE COMPOSTE

Definizione 44.1 (Stabilità della quadratura)

Una formula di quadratura è **numericamente stabile** se piccole perturbazioni nei valori della funzione producono piccole variazioni nel risultato:

$$\delta Q_n[f] = \sum_{i=0}^n w_i \delta f(x_i)$$

Teorema 44.1 (Importanza dei pesi positivi)

Se tutti i pesi $w_i > 0$ e $|\delta f(x_i)| \leq \varepsilon$, allora:

$$|\delta Q_n[f]| \leq \varepsilon \sum_{i=0}^n w_i = \varepsilon \int_a^b 1 dx = \varepsilon(b-a)$$

Interpretazione: L'errore è proporzionalmente controllato.

Problema dei pesi negativi

Le formule di Newton-Cotes di grado elevato hanno spesso **pesi negativi**:

Esempio (8 nodi equispaziati): Alcuni pesi sono negativi e $\sum |w_i| \gg \sum w_i$, causando **instabilità numerica**.

Dimostrazione dell'instabilità

Se alcuni $w_i < 0$:

$$|\delta Q_n[f]| \leq \varepsilon \sum_{i=0}^n |w_i|$$

dove $\sum |w_i|$ può essere molto maggiore di $(b - a)$.

Conseguenza: Formule di grado elevato con nodi equispaziati sono inutilizzabili in pratica.

Necessità delle formule composte

Strategia: Invece di usare una formula di grado elevato su $[a, b]$, si usa:

1. **Suddivisione:** $[a, b]$ in m sottointervalli
2. **Formula semplice:** Su ogni sottointervallo (trapezio/Simpson)
3. **Somma:** I contributi di tutti i sottointervalli

Vantaggi delle formule composte:

1. **Pesi positivi:** Stabili numericamente
 2. **Flessibilità:** Controllo locale dell'errore
 3. **Adattabilità:** Facilmente adattabili
 4. **Implementazione semplice:** Algoritmi diretti
-

45. FORMULE COMPOSTE DEI TRAPEZI E DELLE PARABOLE (CAVALIERI-SIMPSON): DEFINIZIONE E RAPPRESENTAZIONE DELL'ERRORE

Formula composta del trapezio

Suddivisione:

Intervallo $[a, b]$ in n sottointervalli di ampiezza $h = (b-a)/n$:

$$x_i = a + ih, \quad i = 0, 1, \dots, n$$

Formula composta:

$$\int_a^b f(x) dx \approx T_n(f) = h/2 [f(a) + 2\sum_{i=1}^{n-1} f(x_i) + f(b)]$$

Algoritmo:

```
T = (f(a) + f(b))/2
for i = 1 to n-1:
    T += f(x_i)
T *= h
```

Errore della formula composta del trapezio

Teorema 45.1: Se $f \in C^2[a, b]$, allora:

$$\int_a^b f(x) dx - T_n(f) = -h^2(b-a)/12 \cdot f''(\xi)$$

per qualche $\xi \in (a,b)$.

Dimostrazione: Su ogni sottointervallo $[x_i, x_{i+1}]$:

$$\int_{x_i}^{x_{i+1}} f(x) dx - h/2[f(x_i) + f(x_{i+1})] = -h^3/12 \cdot f''(\xi_i)$$

Sommando su tutti gli intervalli:

$$\text{Errore totale} = -h^3/12 \sum_{i=0}^{n-1} f''(\xi_i) = -h^3n/12 \cdot f''(\xi) = -h^2(b-a)/12 \cdot f''(\xi)$$

Ordine di convergenza: $O(h^2)$

Formula composta di Simpson (Cavalieri-Simpson)

Suddivisione:

n sottointervalli (n **pari**), $h = (b-a)/n$:

$$x_i = a + ih, \quad i = 0, 1, \dots, n$$

Formula composta:

$$\int_a^b f(x) dx \approx S_n(f) = h/3 [f(a) + 2\sum_{i=1}^{n/2-1} f(x_{2i}) + 4\sum_{i=0}^{n/2-1} f(x_{2i+1}) + f(b)]$$

Schema mnemonico: 1-4-2-4-2-...-4-2-4-1

Algoritmo:

```
S = f(a) + f(b)
for i = 1 to n-1:
    if i is odd: S += 4*f(x_i)
    if i is even: S += 2*f(x_i)
S *= h/3
```

Errore della formula composta di Simpson

Teorema 45.2: Se $f \in C^4[a,b]$, allora:

$$\int_a^b f(x) dx - S_n(f) = -h^4(b-a)/180 \cdot f^{(4)}(\xi)$$

per qualche $\xi \in (a,b)$.

Ordine di convergenza: $O(h^4)$

Confronto delle prestazioni

Per raggiungere errore ϵ :

- **Trapezio composto:** $h = O(\sqrt{\epsilon}) \rightarrow n = O(1/\sqrt{\epsilon})$
- **Simpson composto:** $h = O(\sqrt[4]{\epsilon}) \rightarrow n = O(1/\sqrt[4]{\epsilon})$

Esempio: Per $\epsilon = 10^{-8}$:

- Trapezio: $n \approx 10^4$ valutazioni
- Simpson: $n \approx 10^2$ valutazioni

Complessità computazionale

- **Trapezio:** $O(n)$ valutazioni, $O(h^2)$ errore
 - **Simpson:** $O(n)$ valutazioni, $O(h^4)$ errore
 - **Costo per precisione fissata:** Simpson molto più efficiente
-

APPENDICE: FORMULE E ALGORITMI CHIAVE

CONVERSIONI E RAPPRESENTAZIONE

Conversioni binarie

- **Intera:** Divisioni successive per 2
- **Frazionaria:** Moltiplicazioni successive per 2
- **IEEE754 double:** 1 + 11 + 52 bit, $\epsilon_m = 2^{-53}$

Precisione macchina

$$\epsilon_m = 2^{-t} \quad (t = \text{bit mantissa})$$

$$\text{gap}(x) = 2^k \cdot 2\epsilon_m \text{ per } 2^k \leq |x| < 2^{k+1}$$

CONDIZIONAMENTO E STABILITÀ

Numero di condizionamento

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

$$\kappa_2(A \text{ simmetrica}) = \lambda_{\max} / \lambda_{\min}$$

Errore nei sistemi lineari

$$\|\delta x\| / \|x\| \leq \kappa(A) \cdot \|\delta b\| / \|b\|$$

Stabilità somma

$$\text{Errore relativo} \leq n\epsilon_m / (1 - n\epsilon_m) \approx n\epsilon_m$$

METODI ITERATIVI

Convergenza generale

$$\rho(B) < 1 \Leftrightarrow \text{convergenza}$$

B = matrice di iterazione

Richardson ottimale

$$\alpha_{\text{opt}} = 2 / (\lambda_{\min} + \lambda_{\max})$$

$$\rho(I - \alpha_{\text{opt}} A) = (\kappa(A) - 1) / (\kappa(A) + 1)$$

Jacobi/Gauss-Seidel

- **Jacobi:** $M = D$
- **Gauss-Seidel:** $M = D + L$
- **Convergenza:** Diagonale dominante \rightarrow convergenza garantita

Gerschgorin

Autovalori in $\cup_i \{z : |z - a_{ii}| \leq \sum_{j \neq i} |a_{ij}|\}$



EQUAZIONI NON LINEARI

Bisezione

$$|x_n - \xi| \leq (b_0 - a_0)/2^{n+1}$$

Convergenza lineare con fattore 1/2

Newton

$$x_{k+1} = x_k - f(x_k)/f'(x_k)$$

$$|e_{k+1}| \leq M|e_k|^2 \quad (\text{convergenza quadratica per radici semplici})$$

$$M = |f''(\xi)|/(2|f'(\xi)|)$$

Contrazioni

$$|g'(x)| \leq L < 1 \Rightarrow \text{convergenza}$$

$$|e_k| \leq L^k |e_0|$$



INTERPOLAZIONE

Errore di interpolazione

$$f(x) - p(x) = f^{(n+1)}(\xi)/((n+1)!) \cdot \prod_{i=0}^n (x - x_i)$$

Lagrange

$$L_i(x) = \prod_{j \neq i} (x - x_j)/(x_i - x_j)$$

$$p(x) = \sum_i y_i L_i(x)$$

Costante di Lebesgue

$$\Lambda_n = \max_x \sum_i |L_i(x)|$$

$$|\tilde{p} - p| \leq \Lambda_n \cdot \max |\delta y_i|$$

Nodi ottimali

- **Equispaziati:** $\Lambda_n \sim O(2^n)$ ❌
- **Chebyshev:** $\Lambda_n \sim O(\log n)$ ✅

Chebyshev

$$\text{Prima specie: } x_j = \cos((2j+1)\pi/(2(n+1)))$$

$$\text{Lobatto: } x_j = \cos(j\pi/n)$$

Ricorrenza: $T_0 = 1, T_1 = x, T_{n+1} = 2xT_n - T_{n-1}$

MINIMI QUADRATI

Equazioni normali

$$A^T A x^* = A^T b$$

$$\kappa(A^T A) = \kappa(A)^2 \text{ (malcondizionamento!)}$$

Metodo QR

$$A = QR \Rightarrow R x^* = Q^T b$$

$$\kappa(\text{problema QR}) = \kappa(A)$$

Gram-Schmidt

$$u_k = v_k - \sum_{j < k} \text{proj}_{u_j}(v_k)$$

$$q_k = u_k / \|u_k\|$$

∫ QUADRATURA NUMERICA

Formule semplici

$$\text{Trapezio: } \int f \, dx \approx (b-a)/2 [f(a) + f(b)], \text{ Errore} = O((b-a)^3)$$



$$\text{Simpson: } \int f \, dx \approx (b-a)/6 [f(a) + 4f((a+b)/2) + f(b)], \text{ Errore} = O((b-a)^5)$$

Formule composte

$$\text{Trapezio: } h/2 [f(a) + 2\sum f(x_i) + f(b)], \text{ Errore} = O(h^2)$$

$$\text{Simpson: } h/3 [f(a) + 4\sum f(x_{2i+1}) + 2\sum f(x_{2i}) + f(b)], \text{ Errore} = O(h^4)$$

Stabilità

- **Pesi positivi:** Stabili 
- **Pesi negativi:** Instabili 

STRATEGIE DI STUDIO E MEMORIZZAZIONE

STRUTTURA LOGICA DEL CORSO

Flusso concettuale:

Numeri macchina → Errori → Stabilità → Condizionamento

↓

Sistemi lineari (diretti) → Sistemi lineari (iterativi)

↓

Equazioni non lineari → Punti fissi → Contrazioni

↓



TECNICHE DI MEMORIZZAZIONE

Per le formule:

1. Derivare sempre dal primo principio (Taylor, Lagrange, etc.)
2. Ricordare gli ordini di grandezza ($O(h^2)$, $O(h^4)$, etc.)
3. Collegare teoria-implementazione
4. Studiare i controesempi (Runge, pesi negativi)

Per le dimostrazioni:

1. Schema logico prima dei calcoli
2. Teoremi di analisi come strumenti (Rolle, media integrale)
3. Identificare i passaggi chiave
4. Capire le ipotesi necessarie

Per l'implementazione:

1. Complessità computazionale
2. Controllo degli errori
3. Criteri di arresto
4. Problemi di stabilità

ERRORI COMUNI DA EVITARE

1. Confondere convergenza e stabilità
2. Non verificare le condizioni di applicabilità
3. Trascurare il condizionamento
4. Usare nodi equispaziati per interpolazione di grado alto
5. Non controllare i segni nelle formule
6. Confondere errore assoluto e relativo



DOMANDE CHIAVE PER L'ESAME

Per ogni algoritmo:

- Quando si applica?
- Che convergenza ha?
- È stabile numericamente?
- Qual è la complessità?
- Quali sono le alternative?

Per ogni teorema:








- Ipotesi necessarie?
- Conseguenze pratiche?
- Collegamenti con altri risultati?

- Esempi di applicazione?
-

🏁 CONCLUSIONI

Questa **BIBBIA DI CALCOLO NUMERICO** contiene tutti i 45 argomenti del glossario del prof. Piazzon in forma completa, rigorosa e sistematica.

Ogni sezione include:

-  Definizioni formali e precise
-  Teoremi con dimostrazioni complete (dove richieste)
-  Algoritmi implementabili
-  Analisi di stabilità e convergenza
-  Esempi e controesempi
-  Complessità computazionale
-  Collegamenti logici tra argomenti

Il documento copre:

- **Parte I-II:** Aritmetica macchina e condizionamento (arg. 1-7)
- **Parte III:** Algebra lineare diretta (arg. 8-10)
- **Parte IV:** Metodi iterativi e contrazioni (arg. 11-16)
- **Parte V:** Sistemi sovradeterminati (arg. 17-19)
- **Parte VI:** Equazioni non lineari (arg. 20-25)
- **Parte VII:** Interpolazione polinomiale (arg. 26-34)
- **Parte VIII:** Approssimazione ai minimi quadrati (arg. 35-40)
- **Parte IX:** Quadratura numerica (arg. 41-45)

Usa questo documento come:

1. **Riferimento completo** per tutti gli argomenti
2. **Guida per le dimostrazioni** richieste all'esame
3. **Compendio di formule** e algoritmi
4. **Strumento di ripasso** sistematico

LA BIBBIA completa (per davvero 😊). È tutto qui dentro. In bocca al lupo per l'esame!

