



Riassunto Calcolo Numerico

Calcolo numerico (Politecnico di Torino)



Scan to open on Studocu

ARITMETICA e ERRORI

RAPPRESENTAZIONE FLOATING-POINT

$$[a = pN^q]$$

$p \rightarrow$ mantissa } se normalizzata
 $q \rightarrow$ esponente }

è NORMALIZZATA se $N^{-1} \leq |p| < 1 \Rightarrow$ questa rappresentazione (NORMALIZZATA) è unica

Il calcolatore memorizza:

- segno (\pm)
- mantissa
- caratteristica o esponente

1) $|p|$ può avere al massimo t cifre
 2) $m \leq q \leq M$ con $m < 0$ e $M > 0, \in \mathbb{Z}$

Il calcolatore memorizza $q^* = q - m \geq 0$ (esponente normalizzato)

NUMERI MACCHINA \rightarrow numeri rappresentabili dal calcolatore

- Se $q < m \Rightarrow$ UNDERFLOW (a $\approx 0 \rightarrow$ il processo non si arresta - WARNING)
- Se $q > M \Rightarrow$ OVERFLOW (a non è un numero di macchina, il processo si arresta - ERROR)

OVERFLOW

TECNICA DI TRONCAMENTO *

TECNICA DI ARROTONDAMENTO
 (si aggiunge $\frac{1}{2}N^{-t}$ a p e poi si tronca alla t -esima cifra)

$$[a = pN^q + \frac{1}{2}N^{-t}]$$

**

ERRORI

Assoluto $\rightarrow |a - \bar{a}| = e_a$

Relativo $\rightarrow \frac{|a - \bar{a}|}{|a|} = e_r$

Errore di arrotondamento (approssimazione)

$$* |p - \bar{p}| < N^{-t}$$

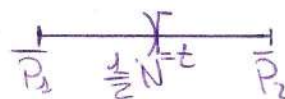
$$* e_a < N^{q-t}$$

$$* e_r < N^{1-t}$$

$$** |p - \bar{p}| \leq \frac{1}{2}N^{-t}$$

$$** e_a \leq \frac{1}{2}N^{q-t}$$

$$** e_r \leq \frac{1}{2}N^{1-t}$$



eps \rightarrow precisione di macchina \rightarrow il più piccolo numero rappresentabile

$\epsilon(\epsilon)$

$$[\epsilon = \frac{\bar{a} - a}{a} \rightarrow \bar{a} = (1 + \epsilon)a]$$

$$* \epsilon = N^{1-t}$$

$$** \epsilon = \frac{1}{2}N^{1-t}$$

più piccolo numero che sommato ad 1 dà contributo.

OPERAZIONI DI MACCHINA

①

↳ operazioni che restituiscano un numero di macchina ottenuto arrotondando l'esatto risultato al numero di macchina corrispondente.

$$\bar{a}_1 \oplus \bar{a}_2 = \overline{\bar{a}_1 + \bar{a}_2} = (a_1 + a_2)(1 + e_{\oplus}) \quad \text{dove}$$

$$\bar{a}_1 \ominus \bar{a}_2 = \overline{\bar{a}_1 - \bar{a}_2} = (a_1 - a_2)(1 + e_{\ominus}) \quad \text{RHS } \varepsilon$$

$$\bar{a}_1 \otimes \bar{a}_2 = \overline{\bar{a}_1 \times \bar{a}_2} = (a_1 \times a_2)(1 + e_{\otimes})$$

$$\bar{a}_1 \oslash \bar{a}_2 = \overline{\bar{a}_1 / \bar{a}_2} = (a_1 / a_2)(1 + e_{\oslash})$$

Proprietà

Valle la proprietà commutativa

Non valgono le proprietà associativa e distributiva

• Due espressioni sono considerate equivalenti nell'aritmetica del calcolatore se

$$\frac{|e_1 - e_2|}{|e_1|} \approx \varepsilon \quad \text{oppure} \quad \frac{|e_1 - e_2|}{|e_2|} \approx \varepsilon$$

CANCELLAZIONE NUMERICA

↳ Perdita di cifre della mantissa che si verifica quando si esegue un'operazione di sottrazione fra due numeri "quasi uguali" e arrotondati ai loro numeri macchina.

↳ l'arrotondamento produce un errore che si propaga.

↳ è possibile evitare questo problema:

1) RAZIONALIZZANDO

2) FORMULE TRIGONOMETRICHE

3) POLINOMI DI TAYLOR NEL PUNTO DEL PROBLEMA.

CONDIZIONAMENTO DI UN PROBLEMA

• PROBLEMA NUMERICO → descrizione non ambigua di una relazione funzionale f tra i dati di input (x) e di output (y)

• Problema BEN CONDIZIONATO se: $x \rightarrow \boxed{f} \rightarrow y$

$$\frac{\|x - \bar{x}\|}{\|x\|} \approx \frac{\|f(x) - f(\bar{x})\|}{\|f(x)\|}$$

Per studiare il condizionamento si utilizza $K(f, x) \rightarrow$ numero di condizionamento

$$\frac{\|f(x) - f(\bar{x})\|}{\|x - \bar{x}\|} \leq \frac{\|x - \bar{x}\|}{\|x\|} \cdot K(f, x) \quad \text{se } K \approx 1 \Rightarrow \text{BEN CONDIZIONATO}$$

STABILITÀ DI UN ALGORITHM

②

• Algoritmo → sequenza finita di operazioni che consente di ottenere y^*

$$\text{Se } \frac{\|f(\bar{x}) - y^*\|}{\|f(\bar{x})\|} \approx \varepsilon \Rightarrow \text{ALGORITHM STABLE}$$

dove y^* è il risultato dell'algoritmo con precisione di macchina finita

$f(\bar{x})$ è il risultato dell'algoritmo, a partire dai numeri macchina \bar{x} , con precisione infinita di calcolo.

SISTEMI LINEARI

Si considera il sistema lineare:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1m}x_m = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2m}x_m = b_2 \\ \dots \dots \dots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mm}x_m = b_m \end{cases}$$

cioè:

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mm} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix} \Leftrightarrow Ax = b$$

Dove A è non singolare
 \Rightarrow sistema che ammette una e una sola soluzione...

Dati:

$$\bullet Ax = b \text{ (in aritmetica esatta)} \quad \bullet \bar{A}\bar{x} = \bar{b}$$

$$\text{dove: } \bar{A} = A + \delta A, \bar{b} = b + \delta b \text{ e } \bar{x} = x + \delta x$$

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

dove $K(A) = \|A\| \|A^{-1}\|$
numero di condizionamento

$$\text{da cui (supponendo che } \|\delta A\| < \frac{1}{2\|A^{-1}\|})$$

$$\frac{\|\delta x\|}{\|x\|} \leq 2K(A) \left(\frac{\|\delta A\|}{\|A\|} + \frac{\|\delta b\|}{\|b\|} \right)$$

• se $K(A) \approx 1$ il sistema è ben condizionato...

• se $K(A) \gg 1$ il sistema è mal condizionato...

• MATRICE DI HILBERT

$$H_m = \begin{pmatrix} 1 & \frac{1}{2} & \dots & \frac{1}{m} \\ \frac{1}{2} & \frac{1}{3} & \dots & \frac{1}{m+1} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{m} & \frac{1}{m+1} & \dots & \frac{1}{2m-1} \end{pmatrix}$$

$$K_2(H_m) \approx 10^{m+1}$$

COMANDO:
hieb(m)
↳ genera matrice
di ordine m

Matrici mal condizionate

• MATRICE DI VANDERMONDE

$$V_m = \begin{pmatrix} x_1^3 & \dots & x_1 & 1 \\ x_2^3 & \dots & x_2 & 1 \\ \vdots & \ddots & \vdots & \vdots \\ x_m^3 & \dots & x_m & 1 \end{pmatrix}$$

dove $x_i \neq x_j$ per $i \neq j$

vander(x)

↳ genera la matrice
associata al vettore x

~~* METODI DIRETTI~~

- x viene determinata mediante un numero finito di passi (m-1)
- x in matematica con precisione infinita di calcolo viene determinata in modo esatto
- modifica la matrice A
- efficienti per matrici dense e di piccole/medie dimensioni.

• Tecnica di sostituzione all'indietro

↳ risolve sistemi triangolari superiori.

• costo computazionale: $\frac{m^2}{2}$

$$\begin{cases} x_m = \frac{b_m}{a_{mm}} \\ x_i = \frac{b_i - \sum_{d=i+1}^m a_{id} x_d}{a_{ii}} \end{cases}$$

COMANDI MATLAB:

A = [...];

b = [...];

m = length(b);

x = zeros(m,1);

x(m) = b(m)/A(m,m);

for i = m:-1:1

s = A(i,i+1:m)*x(i+1:m);

x(i) = (b(i) - s)/A(i,i);

end

• Tecnica di sostituzione in avanti

↳ risolve sistemi triangolari inferiori

• costo computazionale: $\frac{m^2}{2}$

$$\begin{cases} x_1 = \frac{b_1}{a_{11}} \\ x_i = \frac{b_i - \sum_{d=1}^{i-1} a_{id} x_d}{a_{ii}} \end{cases}$$

COMANDI MATLAB:

A = [...];

b = [...];

m = length(b);

x = zeros(m,1);

x(1) = b(1)/A(1,1);

for i = 2:m

s = A(i,1:i-1)*x(1:i-1);

x(i) = (b(i) - s)/A(i,i);

end

• Metodo delle eliminazioni di Gauss

(3)

↳ Dato un sistema $Ax=b$, questo metodo, trasforma in $m-1$ passi questo sistema in un sistema equivalente $Ux=b$. Dove U è una matrice triangolare superiore.

↳ Il sistema rimane invariato se si sostituisce un'equazione con una combinazione lineare con la stessa ed un'altra o se si scambiano tra loro.

$$\begin{cases} m_{ik} = -\frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \\ a_{ij}^{(k+1)} = a_{ij}^{(k)} + m_{ik} a_{kj}^{(k)} \\ b_i^{(k+1)} = b_i^{(k)} + m_{ik} b_k^{(k)} \end{cases} \quad \text{con } i = k+1, \dots, m \rightsquigarrow \begin{cases} a_{ik} = -\frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \\ a_{ij}^{(k+1)} = a_{ij}^{(k)} + a_{ik} a_{kj}^{(k)} \\ b_i^{(k+1)} = b_i^{(k)} + a_{ik} b_k^{(k)} \end{cases}$$

con $i = k+1, m$ e $k = 1: m-1 \rightarrow$ contatore dei passi

Costo computazionale Dopo la risoluzione (che avviene con sostituzioni):

$$\frac{m^3}{3}$$

CONVANDI MATLAB

$A[-i; \dots; i; \dots]; b[-i; \dots; i; \dots]; m = \text{length}(b);$

for $k = 1: m-1$

$$A(k+1:m, k) = -(A(k+1:m, k) / A(k, k));$$

$$A(k+1:m, k+1:m) = A(k+1:m, k+1:m) + A(k+1:m, k) * A(k, k+1:m);$$

$$b(k+1:m) = b(k+1:m) + A(k+1:m, k) * b(k);$$

end

NOTA:
Sostituisco ad i e j
 $\rightsquigarrow k+1:m$

• Pivoting Parziale

↳ Consiste nello scegliere a_{kk} in modo che sia il più grande elemento (uno scambio di equazioni garantisce una migliore stabilità di Gauss ed è superfluo se:

- A è a diagonale dominante per righe
- " " " " " " " " colonne
- A è simmetrica e definita positiva.

• Fattorizzazione di Gauss

↳ Effettua scambi con matrici P (matrici diagonali con colonne scambiate) e elimina gli elementi sotto l'elemento diagonale con matrici M (matrice triangolare inferiore con diagonale unitaria dove sotto la diagonale è tutto zero tranne la k colonna) per $m-1$ volte

$$M_{m-1} P_{m-1} \dots M_2 P_2 M_1 P_1 A x = M_{m-1} P_{m-1} \dots M_2 P_2 M_1 P_1 b$$

$$G A x = G b$$

$$\Updownarrow \\ U x = b$$

$$G A = U \rightarrow \text{FATTORIZ. DI GAUSS}$$

• Tattorizzazione PAZU

$$\overline{M}_{m-1} \overline{M}_{m-2} \dots \overline{M}_2 \overline{M}_1 P_{m-1} \dots P_2 P_1 A = U$$

$$\underline{G} \overline{G} = \overline{M} P$$

Dove le matrici \overline{M}_k sono dello stesso tipo di M_k

MATRICE DI PERMUTAZIONE: $P = P_{m-1} \dots P_1$ e definendo $L = \overline{M}^{-1}$:

$$\overline{M} P A = U \Rightarrow P A = L U$$

P = matrice di permutazione
(vengono memorizzati gli scambi)

A = matrice di partenza
 L = matrice diagonale inferiore
con diagonale unitaria (moltiplicatori)
 U = matrice diagonale superiore utile a
risolvere il sistema mediante riduzione

COMANDI MATLAB

$x = A \backslash b$ → calcola la soluzione di $Ax = b$ con il metodo opportuno

$[L, U, P] = \text{lu}(A)$ → calcola i fattori L, U, P di $PA = LU$

Utilizzo:

• Se ho $Ax = b \rightarrow \underbrace{PA}_{PA=LU} x = Pb \rightarrow \underbrace{LU}_{L} x = Pb \rightarrow \begin{cases} Lx = Pb \\ Ux = y \end{cases}$ IN COMANDI
 $y = L \backslash P * b$
 $x = L \backslash U$

• $\det(A) = (-1)^s \prod_{i=1}^m U_{ii}$

• $\text{inv}(A)$ con costo quadratico ($O(m^3)$)
 $PA = LU \rightarrow (PA)^{-1} = (LU)^{-1} \rightarrow A^{-1} P^{-1} = U^{-1} L^{-1} \rightarrow$
 $\rightarrow A^{-1} P = U^{-1} L^{-1} P$

• Tattorizzazione di Cholesky

Nel caso in cui gli scambi siano superflui

$$A = LU \rightarrow A = LL^T$$

$$\rightarrow A = R^T R$$

COMANDO

$$R = \text{chol}(A)$$

* METODI ITERATIVI

- x viene determinato come limite di una successione di vettori convergenti
- x viene determinato in precisione finita di calcolo
- Non modificano A
- efficienti per matrici sparse e di grandi dimensioni.

↳ A partire da un vettore iniziale arbitrario $x^{(0)}$ si determina una sequenza di vettori che converge (sotto opportune condizioni) alla soluzione esatta del sistema.

$$x^{(k)} \xrightarrow{k \rightarrow \infty} x$$

si ricava l' i -esima variabile e si procede partendo da un vettore arbitrario

• METODO DI JACOBI *

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

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

METODI DI GAUSS-SEIDEL **

4

$$x_i^{(k+1)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^m a_{ij} x_j^{(k)}}{a_{ii}} \quad k=0,1,\dots$$

Caratteristiche:

* e ** fanno parte di una classe ampia di metodi dove:

$$Ax=b \rightarrow (D+C)x=b \rightarrow Dx=b-Cx \rightarrow D x^{(k+1)}=b-C x^{(k)}$$

D deve essere:

- non singolare
- di forma semplice
- tale da garantire la convergenza

* $D = \begin{pmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_{mm} \end{pmatrix}$ COMANDO
 $D = \text{diag}(\text{diag}(A))$

** $D = \begin{pmatrix} a_{11} & 0 & \dots & 0 \\ a_{21} & a_{22} & \dots & 0 \\ a_{31} & a_{32} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mm} \end{pmatrix}$ COMANDO
 $D = \text{tril}(A)$

* comandi

```
A = [...; ...; ...];
b = [...; ...; ...];
m = length(b);
max = ...; tol = ...; // tolleranza
x0 = zeros(m,1);
D = diag(diag(A));
C = (A-D);
for k=1:mmax
    x1 = D \ (b - C*x0);
    if norm(x1-x0) < tol
        break
    end
    x0 = x1;
end
```

** COMANDI:

```
A = [...; ...; ...];
b = [...; ...; ...];
m = length(b);
mmax = ...; tol = ...; // tolleranza
D = tril(A);
C = A-D;
x0 = zeros(m,1);
for k=0:mmax
    x1 = D \ (b - C*x0);
    if norm(x1-x0) < tol
        break
    end
    x0 = x1;
end
```

Convergenza:

Se la norma della matrice di iterazione $B = I - D^{-1}A$ è minore di 1 allora il metodo converge (non vale l'implicazione inversa)

$$\|B\| < 1 \Rightarrow \text{converge}$$

Il metodo converge \Leftrightarrow

spettro è minore di 1

$$\rho(B) < 1$$

Se A è

- DIAGONALE DOMINANTE PER RIGHE OPP. COLONNE OPP.
 - SIMMETRICA DEFINITA POSITIVA
- \Rightarrow GAUSS SEIDEL CONVERGE
- \Rightarrow JACOBI CONVERGE.

APPROSSIMAZIONE DI FUNZIONI

Approssimare con una funzione \tilde{f} che sia ~~più~~ "vicina" ad f con una forma più semplice

Per approssimare occorre:

- individuare una classe di funzioni idonea
- adottare un criterio per la scelta di \tilde{f}
- valutare la bontà di \tilde{f} , ovvero la "distanza" di \tilde{f} da $f \rightarrow \|\tilde{f} - f\|_\infty$

INTERPOLAZIONE POLINOMIALE

Scegliere una funzione (possibilmente unica) $\tilde{f} = f_m$ soddisfacente la seguente condizione

$$f_m(x_i) = y_i$$

(condizione di interpolazione)

con $i = 0, 1, \dots, m+1$

Con $m+1$ punti consideriamo il polinomio:

$$\tilde{f}(x) = P_m(x) = c_1 x^m + c_2 x^{m-1} + \dots + c_m x + c_{m+1} \quad (Vx = x_i)$$

Questo polinomio è definito dalla matrice di Vandermonde e si dimostra che è unico per ogni funzione

Il polinomio è chiamato polinomio di interpolazione e i punti x_i sono chiamati NODI DI INTERPOLAZIONE.

• NOTA:

Per disegnare un grafico:

$f = \text{imfime}$ ('espressione funzionale');

$x = \text{linspace}(a, b, m)$; // vettore di m punti

$y = f(x)$;

$\text{plot}(x, y)$;

QUANDO PER INTERPOLAZIONE

$c = \text{polyfit}(x, y, m)$

Calcola e memorizza in c , i coefficienti del polinomio interpolante i punti x_i e y_i definiti in x e y .

$p = \text{polyval}(c, z)$

Calcola e memorizza in p , i valori che il polinomio assume nei punti z_i contenuti in z .

ESAMI DI PROGRAMMI MATLAB:

Polinomio interpolante dei punti:

$$x = [0, 1, 2, 1/2]; \quad \left\{ \begin{array}{l} \text{punti } (x_i, y_i) \\ y = [-1, -2, 1, 2]; \end{array} \right.$$

$$z = \text{linspace}(\min(x), \max(x));$$

$$c = \text{polyfit}(x, y, \text{length}(x)-1);$$

$$p = \text{polyval}(c, z);$$

$$\text{plot}(x, y, 'b', z, p, 'r')$$

Polinomio interpolante $f = \frac{1}{1+x^2}$

con 13 punti in $[0, 1]$:

$$f = \text{inline}('1./(1+x.^2)');$$

$$x = \text{linspace}(0, 1, 13)$$

$$y = f(x);$$

$$z = \text{linspace}(0, 1);$$

$$c = \text{polyfit}(x, y, 14);$$

$$p = \text{polyval}(c, z);$$

$$\text{plot}(x, y, 'b', z, p, 'r')$$

Rappresentazione di Lagrange

$L_i(x)$ polinomio interpolante i punti (x_i, y_i) (con i e j da 1 a $m+1$)

$$L_i(x) = y_i = \begin{cases} 1 & \text{se } i=j \\ 0 & \text{se } i \neq j \end{cases} \quad (x_1, 0), \dots, (x_{i-1}, 0), (x_i, 1), (x_{i+1}, 0), \dots, (x_{m+1}, 0)$$

$$L_i(x) = \frac{(x-x_1)(x-x_2)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_{m+1})}{(x_i-x_1)(x_i-x_2)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_{m+1})} \quad \left\{ \begin{array}{l} \text{polinomi fondamentali} \\ \text{di Lagrange} \end{array} \right.$$

$$P_m(x_i) = \sum_{j=1}^{m+1} L_j(x_i) = y_i$$

CAVITÀ: Il polinomio dipende dall'insieme completo di dati se vogliamo aggiungere un punto si deve rifare i calcoli.

Rappresentazione di Newton

$$P_m(x) = f(x_1) + f[x_1, x_2](x-x_1) + f[x_1, x_2, x_3](x-x_1)(x-x_2) + \dots + f[x_1, x_2, \dots, x_{m+1}](x-x_1)(x-x_2)\dots(x-x_m)$$

dove $f[x_1, x_2]$ è la differenza divisa di ordine 2

$$f[x_1, x_2] = \frac{f(x_2) - f(x_1)}{x_2 - x_1}$$

$$f[x_1, x_2, x_3] = \frac{f[x_2, x_3] - f[x_1, x_2]}{x_3 - x_1}$$

$$f[x_1, x_2, \dots, x_{m+1}] = \frac{f[x_2, x_3, \dots, x_{m+1}] - f[x_1, x_2, \dots, x_m]}{x_{m+1} - x_1}$$

Dati $(x_1, f(x_1)), (x_2, f(x_2)), (x_3, f(x_3)), (x_4, f(x_4))$

$$\begin{array}{lcl} x_1 & f(x_1) & \rightarrow \\ x_2 & f(x_2) & \rightarrow \frac{f(x_2) - f(x_1)}{x_2 - x_1} \rightarrow f[x_1, x_2] \\ x_3 & f(x_3) & \rightarrow \frac{f(x_3) - f(x_2)}{x_3 - x_2} \rightarrow f[x_2, x_3] \\ & & \rightarrow \frac{f[x_2, x_3] - f[x_1, x_2]}{x_3 - x_1} \rightarrow f[x_1, x_2, x_3] \\ x_4 & f(x_4) & \rightarrow \frac{f(x_4) - f(x_3)}{x_4 - x_3} \rightarrow f[x_3, x_4] \\ & & \rightarrow \frac{f[x_3, x_4] - f[x_2, x_3]}{x_4 - x_2} \rightarrow f[x_2, x_3, x_4] \\ & & \rightarrow \frac{f[x_2, x_3, x_4] - f[x_1, x_2, x_3]}{x_4 - x_1} \rightarrow f[x_1, x_2, x_3, x_4] \end{array}$$

ERRORE DI INTERPOLAZIONE

$$\hookrightarrow E_m(x) = f(x) - P_m(x)$$

Anche se, $\forall x = x_i, E_m(x_i) = 0$ non è detto che per $m \rightarrow \infty$ $E_m(x) \equiv 0$ (FUNZIONE DI RUNGE con nodi x_i equispaziati) per essere così la funzione deve essere analitica.

La convergenza uniforme è garantita con i nodi di Chebyshev

$$z_i = -\cos\left(\frac{2i-1}{2m}\pi\right), \text{ dove } z_i \in (-1, 1)$$

Se l'intervallo non è $(-1, 1)$ si deve traslare l'intervallo:

$$x_i = \frac{b-a}{2} z_i + \frac{b+a}{2} \quad \text{per un intervallo } [a, b] \text{ generico}$$

SPLINE (funzioni polinomiali a tratti)

↳ funzione rappresentata in una partizione di $[a, b]$ da un'unione di tratti contigui di polinomi di grado "d" diversi.

↳ generalmente essi sono derivabili nei punti di raccordo ma le funzioni spline godono di una certa regolarità nei punti di raccordo.

↳ vengono utilizzate quando il polinomio è eccessivamente oscillatorio

spline cubica

Si definisce spline cubica una funzione $S_3(x)$ che soddisfa:

* POLINOMIO DI GRADO 3 IN TUTTI I TRATTI:

$$S_3(x) = a_i + b_i x + c_i x^2 + d_i x^3, \quad x \in [x_i, x_{i+1}], \quad i = 1:m$$

** Regolare nei punti di raccordo ($\in C^2([x_1, x_m])$)

$$S_3^{(k)}(x_i^-) = S_3^{(k)}(x_i^+), \quad k = 0, 1, 2$$

essa è interpolante se soddisfa la condizione di interpolazione

$$*** S_3(x_i) = y_i, \quad i = 1, \dots, m+1$$

S_3 dipende da $4m$ parametri e le 3 condizioni danno un sistema di $4m-2$ equazioni quindi si devono aggiungere altre 2 condizioni a scelta

tra:

$$1) S_3^{(2)}(x_1) = 0, \quad S_3^{(2)}(x_{m+1}) = 0 \rightarrow \text{spline cubiche naturali}$$

$$2) S_3^{(3)}(x_2) = S_3^{(3)}(x_2^+) \text{ e } S_3^{(3)}(x_{m-1}) = S_3^{(3)}(x_{m-1}^+) \rightarrow \text{condizioni NOT-A-KNOT}$$

$$3) S_3^{(1)}(x_1) = f'(x_1), \quad S_3^{(1)}(x_{m+1}) = f'(x_{m+1})$$

avere 4m equazioni \Rightarrow si definisce univocamente una spline cubica

$O(h^4)$ è il massimo ordine di convergenza delle spline cubiche

Ma caso di convergenza la convergenza dei polinomi di interpolazione è più rapida

COMANDI MATLAB

$s = \text{spline}(x, y, z) \rightarrow$ calcola e memorizza in s i valori ^{assunti in z_i} della spline interpolante, i dati (x_i, y_i) definiti in x e y (condiz. z)

$s = \text{spline}(x, [y_1 \dots y_m], \dots) \rightarrow$ calcola e memorizza in s i valori della spline cubica interpolante assunte in z e soddisfacente la condizione z

CRITERIO DEI MINIMI QUADRATI

è utile per estrapolare informazioni dai dati.

genera valutazioni in punti che stanno al di fuori dell'intervallo di interpolazione

si usa quando si vuole approssimare un insieme di dati di m elementi con una funzione che dipende da un numero molto minore di m elementi

$$\hat{y}(x) = c_1 \psi_1(x) + c_2 \psi_2(x) + \dots + c_m \psi_m(x)$$

si impone che il residuo $E^2(c_1, \dots, c_m)$ risulti minimo

$$\begin{aligned} E^2(c_1, \dots, c_m) &= \sum_{i=1}^m [y_i - \hat{y}(x_i)]^2 \\ &= \sum_{i=1}^m \left[y_i - \sum_{j=1}^m c_j \psi_j(x_i) \right]^2 \end{aligned}$$

I coefficienti c sono la soluzione del seguente sistema lineare

$$\begin{pmatrix} \sum_{i=1}^m \psi_1(x_i) \psi_1(x_i) & \sum_{i=1}^m \psi_1(x_i) \psi_2(x_i) & \dots \\ \sum_{i=1}^m \psi_2(x_i) \psi_1(x_i) & \sum_{i=1}^m \psi_2(x_i) \psi_2(x_i) & \dots \\ \vdots & \vdots & \ddots \\ \sum_{i=1}^m \psi_m(x_i) \psi_1(x_i) & \sum_{i=1}^m \psi_m(x_i) \psi_2(x_i) & \dots \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^m y_i \psi_1(x_i) \\ \sum_{i=1}^m y_i \psi_2(x_i) \\ \vdots \\ \sum_{i=1}^m y_i \psi_m(x_i) \end{pmatrix}$$

B (simmetrica e def. positiva)

$* C = d$

EQUAZIONI NON LINEARI

↳ Ricerca di soluzioni $f(x)=0$ dove $f(x)$ è una funzione non risolvibile analiticamente

- 1) espressione di f complicata
 - 2) risoluzione di $f(x)=0$ complicata
 - 3) $f(x)$ è nota solo a punti
- ⇒ si usa il metodo numerico
↳ i metodi numerici sono di tipo iterativo...

A partire da un valore x_0 si determina una successione di valori $\{x_m\}$ che, sotto opportune ipotesi, converge ad una radice α .

La convergenza relativa dipende dal valore x_0 :
"se x_0 è sufficientemente "vicino" ad α ed f è "sufficientemente regolare" ⇒ $\lim_{m \rightarrow \infty} x_m = \alpha \leadsto$ si può procedere graficamente

CONVERGENZA

ordine di convergenza di una successione $\{x_m\}$ convergente ad α è il numero reale p tale che

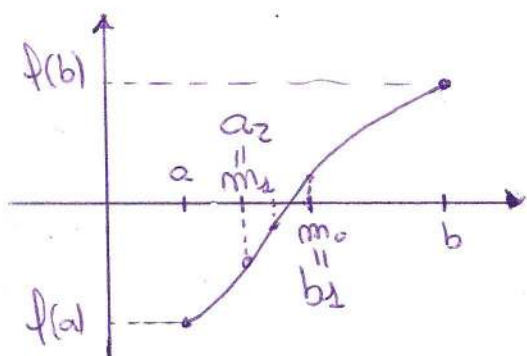
$$\lim_{m \rightarrow \infty} \frac{|\alpha - x_{m+1}|}{|\alpha - x_m|^p} = C \neq 0, +\infty$$

- se $p=1 \Rightarrow C < 1$, convergenza lineare
- se $1 < p < 2 \Rightarrow$ convergenza superlineare
- se $p=2 \Rightarrow$ convergenza quadratica
- se $p=3 \Rightarrow$ convergenza cubica

Da cui:

$$|\alpha - x_{m+1}| \approx C |\alpha - x_m|^p \leadsto p = \frac{\log |x_{m+2} - x_{m+1}|}{\log |x_{m+1} - x_m|}$$

metodo della bisezione



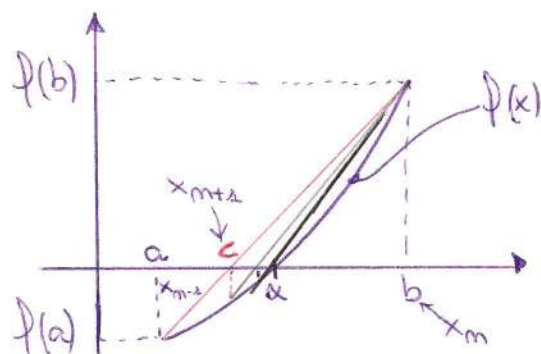
se $f(a)f(b) < 0$ e $f \in C([a, b])$
⇒ $\exists f(x)=0$

- 1) prendo il punto medio $m_0 = \frac{a+b}{2}$
- 2) se $f(m_0)$ è positivo la radice cade nell'intervallo di sinistra
se $f(m_0)$ è negativo la radice cade nell'intervallo di destra
- 3) continuo a dimezzare i sottointervalli e man mano il punto medio ~~si~~ tende ad α

$$\begin{cases} a_0 = a \\ b_0 = b \\ m_0 = \frac{b_0 - a_0}{2} \end{cases} \rightsquigarrow \begin{cases} a_1 = a_0 \\ b_1 = m_0 \\ m_1 = \frac{b_1 - a_1}{2} \end{cases} \rightsquigarrow \begin{cases} a_2 = m_1 \\ b_2 = b_1 \\ m_2 = \frac{b_2 - a_2}{2} \end{cases} \rightsquigarrow m_m = \alpha \quad (7)$$

esso viene utilizzato per prendere un sufficientemente "vicino" e non per determinare α poiché converge lentamente

METODO DELLE SECANTI



considero, una volta individuato l'intervallo in cui ricade la radice, la retta secante i punti $f(a)$ e $f(b)$, questa si interseca con l'asse x nel punto c , se $f(c) < 0$, considero la retta secante i punti $f(c)$ e $f(b)$ esso si interseca con l'asse delle x ... e così via... Altrimenti se $f(c) > 0$ considero la

retta secante i punti $f(a)$ e $f(c)$ ecc...

Matematicamente:

$$\begin{cases} y = f(x_m) + \frac{f(x_m) - f(x_{m-1})}{x_m - x_{m-1}} (x - x_m) \\ y = 0 \end{cases} \rightsquigarrow \text{SOLUZIONE} \quad x_{m+1} \equiv x = x_m - f(x_m) \frac{x_m - x_{m-1}}{f(x_m) - f(x_{m-1})}$$

ORDINE $p = \frac{1+\sqrt{5}}{2} \approx 1,618 \rightarrow$ superlineare

COMANDI MATLAB:

$f = \text{impime}('...');$

$x_0 = \dots; x_1 = \dots;$

$m_{\max} = \dots; \text{toll} = \dots;$

for $n = 0:m_{\max}$

$x_2 = x_1 - f(x_1) * (x_1 - x_0) / (f(x_1) - f(x_0));$

if $\text{abs}(x_2 - x_1) < \text{toll};$

break

end

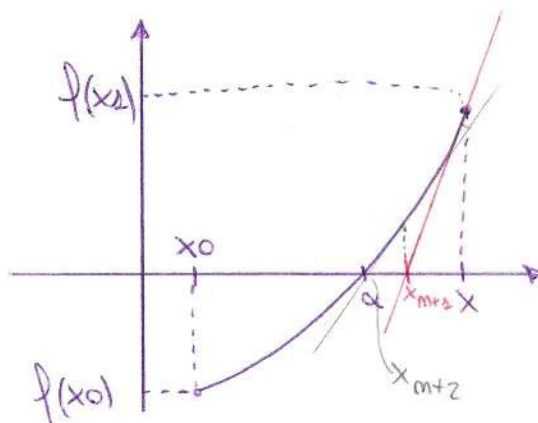
$x_0 = x_1;$

$x_1 = x_2;$

end

x_2

METODO DELLE TANGENTI



Individuato un intervallo contenente la radice si calcola la retta tangente al grafico nel punto x .
 La retta si interseca in un punto x_{m+1} e calcola la retta tangente al grafico in quel punto, ecc...
 fino a che la successione $x_m \xrightarrow{m \rightarrow \infty} \alpha$

$$\begin{cases} y = f(x_m) \cdot f'(x_m)(x - x_m) \\ y = 0 \end{cases} \xrightarrow{\text{SOLUZIONE}} x_{m+1} \equiv x = x_m - \frac{f(x_m)}{f'(x_m)}$$

ORDINE di convergenza:

- se α è una radice semplice ($f(\alpha) = 0$ e $f'(\alpha) \neq 0$) $\Rightarrow p = 2$
- se α non è semplice ($f(\alpha) = 0$, $f'(\alpha) = 0$) $\Rightarrow p = 1$

PRO:

converge più rapidamente
 si può ripartire
 $p = 2$ nel caso
 conosciamo la molteplicità
 (μ) della radice

$$x_{m+2} = x_m - \mu \frac{f(x_m)}{f'(x_m)}$$

CONTRO

più costoso perché richiede il calcolo della derivata prima e la valutazione di $f(x_m)$ e $f'(x_m)$

COMANDI MATLAB:

$f = \text{inline}('...');$ $fd = \text{inline}('...');$

$mmax = \dots;$ $toll = \dots;$

$x0 = \dots;$

for $m = 1:mmax$

$$x1 = x0 - \frac{f(x0)}{fd(x0)}$$

if $\text{abs}(x1 - x0) < \text{toll};$

break

end

$x0 = x1$

end

$x1$

METODO DEL PUNTO FISSO

Il problema del tipo:

$$\alpha: f(\alpha) = 0 \xrightarrow{\text{DIVENTA}} \alpha: \alpha - g(\alpha) = 0$$

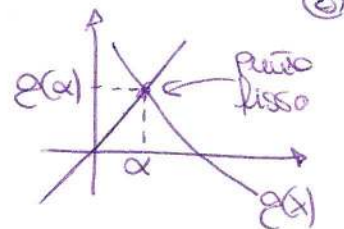
$g(\alpha)$ si definisce punto fisso della funzione $g(x)$

$g(x)$ si definisce per esempio

$$\bullet f(\alpha) = 0 \longrightarrow \alpha + f(\alpha) = \alpha \longrightarrow g(x) = x + f(x)$$

$$\bullet f(\alpha) = 0 \longrightarrow \alpha + \frac{f(\alpha)}{k} = \alpha \longrightarrow g(x) = x + \frac{f(x)}{k}$$

per $k = -f'(x)$: metodo delle tangenti



limite della successione $x_{m+1} = g(x_m)$

TEOREMA 1:

H_p: $\alpha \rightarrow$ punto fisso di g

$g \in C^1$ in un intorno di α

$$|g'(\alpha)| < 1$$

$|\alpha - x_0|$ è sufficientemente piccolo

\Rightarrow la successione $x_{m+1} = g(x_m)$ è convergente ad α

$$\lim_{n \rightarrow \infty} x_n = \alpha$$

$$|g'(\alpha)| > 1 \Rightarrow \text{non converge ad } \alpha$$

TEOREMA 2

Sia I un intervallo dell'asse reale

$$* g(x) \in I \quad \forall x \in I \quad (g(I) \subseteq I)$$

$$* g \in C^1(I)$$

$$* \exists m < 1: |g'(x)| \leq m \quad \forall x \in I$$

\Rightarrow esiste uno e un solo punto fisso α di g in I

$$p=1 \text{ se } g(\alpha) = \alpha, g'(\alpha) \neq 0$$

$$p=2 \text{ se } g(\alpha) = \alpha, g'(\alpha) = 0, g''(\alpha) \neq 0$$

$$p=k \text{ se } g(\alpha) = \alpha, g'(\alpha) = 0, \dots, g^{(k)}(\alpha) \neq 0$$

COMANDI MATLAB

`g = inline('...');`

`x0 = ...;`

`mmax = ...; toll = ...;`

`for m=1:mmax`

`x1 = g(x0);`

`if abs(x1-x0) < toll;`

`break`

`end`

`x0 = x1`

`end`

This document is available free of charge on

studocu

Downloaded by Gianluca Catastrofe (giancatastrofe@gmail.com)

CALCOLO DI INTEGRALI

↳ I metodi iterativi sono utili per risolvere integrali impossibili da integrare analiticamente o quando è complicato calcolarlo o quando l'età solo a punti...

FORMULA DI QUADRATURA

$$\int_a^b f(x) dx \approx \sum_{j=1}^m w_j f(x_j)$$

$x_j \rightarrow$ nodi di quadratura
 $w_j \rightarrow$ coefficienti o pesi

FORMULE DI QUADRATURA INTERPOLATORIE

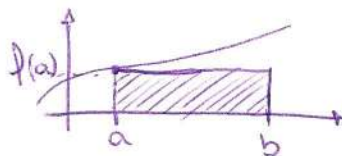
↳ formule di quadratura che si ottengono approssimando la funzione integranda con un polinomio che interpola la funzione

$$\int_a^b \sum_{j=1}^m l_j(x) f(x_j) dx = \sum_{j=1}^m \underbrace{\left(\int_a^b l_j(x) dx \right)}_{w_j} f(x_j)$$

FORMULE DI NEWTON COTES

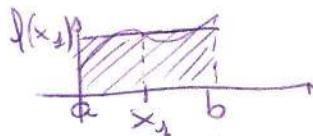
$|m=1|$ e $|x_1=a| \rightarrow$ formula del rettangolo

$$\int_a^b f(x) dx \approx f(a)(b-a)$$



$|m=1|$ e $|x_1 = \frac{a+b}{2}| \rightarrow$ formula del punto medio

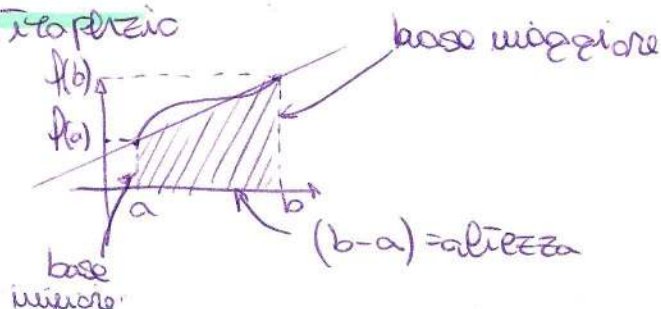
$$\int_a^b f(x) dx \approx f\left(\frac{b+a}{2}\right)(b-a)$$



$|m=2|$ e $|x_1=a \text{ e } x_2=b| \rightarrow$ formula del trapezio

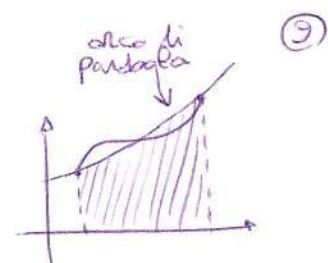
$$\int_a^b f(x) dx \approx \underbrace{\frac{b-a}{2}}_h [f(a) + f(b)]$$

base maggiore + base minore



$[m=3] \mid x_1=a \quad x_2=\frac{b+a}{2} \quad x_3=b \mid \rightarrow$ formula di Simpson

$$\int_a^b f(x) dx \approx \frac{b-a}{6} \left[f(a) + 4f\left(\frac{b+a}{2}\right) + f(b) \right]$$



Se f non è abbastanza regolare ma comunque integrabile si può fattorizzare

$$f(x) = \frac{x e^{-x}}{\sqrt{x-a}} \quad \begin{cases} w(x) = \frac{1}{\sqrt{x-a}} \\ \varphi(x) = x e^{-x} \end{cases}$$

$\hookrightarrow f(x) = w(x) \varphi(x)$

$w(x) \rightarrow$ funzione contenente le singolarità
 $\varphi(x) \rightarrow$ parte regolare

si approssima solo la parte regolare

$$\int_a^b w(x) \varphi(x) dx \approx \int_a^b w(x) p_{m-1}(\varphi; x) dx = \int_a^b w(x) \sum_{j=1}^m \ell_j(x) \varphi(x_j) dx =$$

$$= \sum_{j=1}^m \left(\int_a^b w(x) \ell_j(x) dx \right) \varphi(x_j) \rightarrow \text{FORMULA INTERPOLATORIA PESATA}$$

$\overline{w}_j \rightarrow$ deve essere tale da garantire l'esistenza dell'integrale
 e deve essere integrabile analiticamente