CONSORZIO NETTUNO

POLITECNICO DI TORINO

DIPLOMI UNIVERSITARI TELEDIDATTICI

Eugenio BRUSA – Cristiana DELPRETE – Paolo GAY

Tutorato di CALCOLO NUMERICO

Settembre 2001

Questa raccolta di esercizi e note, prodotta ad uso interno, viene utilizzata per i tutorati dell'insegnamento di *Calcolo Numerico* dei Corsi di Diploma Universitario Teledidattico in Ingegneria Automatica e Informatica, Ingegneria Elettrica, Ingegneria Elettronica e Ingegneria Meccanica.
Nel corso della stesura si è fatto riferimento al testo G. MONEGATO, *100 Pagine di ... Elementi di Calcolo Numerico*, Levrotto & Bella, Torino, 1996.
Il materiale proposto non vuole in alcun modo sostituire il libro di testo, a cui si fa riferimento per quanto riguarda le basi teoriche e l'impostazione formale dell'insegnamento, ma viene fornito esclusivamente a titolo di supporto didattico volto a facilitare l'approccio alla materia.

Eugenio Brusa
Cristiana Delprete
Paolo Gay

Torino, settembre 2001

TUTORATO DI CALCOLO NUMERICO

Contenuti

- *Fondamenti del calcolo scientifico*. Sistema di numerazione e sistemi aritmetici. Errore assoluto ed errore relativo. Cifre decimali corrette e cifre significative corrette. Errori di arrotondamento. Troncamento semplice e arrotondamento simmetrico. Cancellazione numerica. Esercizi svolti. Esercizi proposti.
- *Richiami sul calcolo matriciale*. Matrici e operazioni tra matrici. Matrice trasposta. Matrice simmetrica. Determinante. Matrice simmetrica definita o semidefinita positiva. Matrice non singolare. Rango. Matrice inversa. Norma. Esercizi svolti. Esercizi proposti.
- *Sistemi di equazioni lineari*. Generalità sui metodi di soluzione: diretti e iterativi. Soluzione di sistemi triangolare superiori e inferiori. Algoritmo Backsost. Metodo di Gauss: procedura, decomposizione GA=U, fattorizzazione PA=LU, algoritmo Factor e algoritmo Solve. Metodo di Choleski: procedura e algoritmo. Esercizi svolti. Esercizi proposti. Cenni sul condizionamento dei sistemi lineari. Metodi iterativi: procedura e cenni sulla convergenza. Metodo di Jacobi. Metodo di Gauss-Seidel e algoritmo Gseidel. Cenni sul metodo SOR.
- *Autovalori e autovettori*. Autovalori di matrici. Metodo delle potenze: autovalore di modulo massimo e minimo. Algoritmo Pow. Metodo delle potenze inverse: autovalore di nota approssimazione. Algoritmo Invpow. Cenni sulle trasformazioni di similitudine. Cenni sul metodo QR per il calcolo di tutti gli autovalori.
- *Approssimazione di funzioni e dati sperimentali*. Classi delle funzioni interpolanti. Criteri di scelta della funzione interpolante. Interpolazione polinomiale: metodo di Lagrange e metodo di Newton. Algoritmo Difdiv e algoritmo Interp. Interpolazione polinomiale a tratti. Spline e spline cubica. Algoritmo Spline e algoritmo Valspl. Approssimazione di dati. Minimi quadrati. Regressione lineare. Esercizi svolti. Esercizi proposti.
- *Equazioni non lineari*. Metodo di bisezione e algoritmo Bisez. Regula falsi. Metodo di Newton o delle tangenti. Metodo delle secanti e algoritmo Secant. Confronto tra i diversi metodi. Criteri di arresto. Esercizi svolti. Esercizi proposti.
- *Quadratura*. Generalità. Formule di quadratura di base (Newton-Cotes): rettangolo, punto medio, trapezio, di Simpson. Formule di quadratura Gaussiane. Formule di quadratura composte. Tecniche di quadratura automatica. Esercizi svolti. Esercizi proposti.
- *Equazioni differenziali ordinarie*. Generalità. Condizioni di Lipschitz. Metodi espliciti a passo singolo e a passo multiplo. Metodo di Eulero. Metodi di Runge-Kutta.
- *Temi d'esame*. È fornito il testo di tre prove di esonero.

Testo di riferimento

- G. MONEGATO, *100 Pagine di … Elementi di Calcolo Numerico*, Levrotto & Bella, Torino, 1996.

SOMMARIO

SISTEMI NUMERICI

Riferimento al testo: Cap. I

SISTEMA DI NUMERAZIONE POSIZIONALE

Qualunque numero intero $N > 1$ può essere scelto come base del sistema di numerazione e, in sistema posizionale con base *N*, qualsiasi reale si può scrivere come

$$(a)_N = \pm\left(a_m N^m + a_{m-1}N^{m-1} + \cdots + a_1 N + a_0 + .\, a_{-1}N^{-1} + \cdots\right) = \pm\, a_m a_{m-1} \cdots a_1 a_0\, .a_{-1} \cdots$$

dove le cifre a_i sono numeri interi appartenenti all'intervallo $[0, N-1]$:

$$0 \le a_i \le N-1$$

Esempi

1) $(245.27)_{10} = 2\cdot 10^2 + 4\cdot 10^1 + 5 + 2\cdot 10^{-1} + 7\cdot 10^{-2}$

2) $(10.111011)_2 = 1\cdot 2^1 + 0 + 1\cdot 2^{-1} + 1\cdot 2^{-2} + 1\cdot 2^{-3} + 0\cdot 2^{-4} + 1\cdot 2^{-5} + 1\cdot 2^{-6} = (2.921875)_{10}$

I sistemi aritmetici di un calcolatore sono a **precisione finita** (spazio di memoria finito) ed effettuano operazioni con precisione finita (**operazioni di macchina**: $\oplus, \otimes \ldots$); le **basi** utilizzate sono $N = 2$ o $N = 16$.

I **numeri di macchina** sono numeri rappresentabili esattamente nello spazio di memoria disponibile.

SISTEMA ARITMETICO INTERO

È un sottoinsieme finito dell'insieme infinito degli interi *Z*: $I \subset Z$

$$I(N,l) = \{-i_{\max} \ldots -1\ 0\ 1 \ldots i_{\max}\} \quad \text{con } i_{\max} = N^l - 1\,.$$

Nello spazio di memoria si usano 1 bit per il segno e 7 bit per il numero (l=7).

Si ha Overflow o Underflow sugli interi quando $a \in Z$, ma $\notin I$, cioè $a > i_{\max}$ o $a < -i_{\max}$.

Esempio

3) N=10 , l=2 quindi $i_{\max}=10^2-1=99$

$60 \oplus (50 \oplus (-40)) = 60 \oplus 10 = 70$

$(60 \oplus 50) \oplus (-40) = \text{Overflow} \Rightarrow$ proprietà associativa = NO

SISTEMA ARITMETICO REALE

È un sottoinsieme dell'insieme infinito dei reali *R*: $F \subset R$

$$F(N,t,q) = \left\{ a = p \cdot N^q \ , N^{-t} \le |p| < 1 \ , q_{\min} \le q \le q_{\max} \right\}$$

dove p è la **mantissa** (numero reale $p=.p_1p_2p_3...$) e q è la **caratteristica** o esponente (numero intero).

- La rappresentazione $a = p \cdot N^q$ è **floating point normalizzata** (fpn) se la prima cifra decimale della mantissa è diversa da zero ($p=.p_1p_2p_3...$ con $p_1 \neq 0$).
- La condizione di normalizzazione è: $N^{-1} \le |p| < 1$.

Nello spazio di memoria si usano 1 bit per il segno, 8 bit per la caratteristica (r=8) e 23 bit per la mantissa (t=23) nel caso di singola precisione (SP=32 bit) oppure 11 bit per la caratteristica (r=11) e 52 bit per la mantissa (t=52) nel caso di doppia precisione (DP=64 bit).

Si ha Overflow o Underflow sui reali quando $a \in R$, ma $\notin F$.

Esempi

4) $a=(12.235)_{10}=.12235\cdot10^2=.12235e2$ rappresentazione fpn
5) $a=(.00784)_{10}=.784\cdot10^{-2}=.784e{-}2$ rappresentazione fpn
6) $a=(30000000000)_{10}=.3\cdot10^{11}=.3e11$ rappresentazione fpn

ERRORE ASSOLUTO ED ERRORE RELATIVO

Dato il valore esatto x e la sua approssimazione $\bar{x} = x(1+\varepsilon)$, per $x \neq 0$ si definiscono gli errori assoluto e relativo rispettivamente come:

$$EA \doteq |x - \bar{x}|$$
$$ER \doteq \frac{|x - \bar{x}|}{|x|} = \frac{EA}{|x|}$$

cioè ER è EA scalato con x.

EA ed ER **quantificano** la **bontà** dell'**approssimazione** $\bar{x}$ e forniscono rispettivamente il numero di decimali corretti e il numero di cifre significative corrette.

CIFRE DECIMALI CORRETTE

P1) Se $\bar{x}$ è un'approssimazione di x con p decimali corretti allora $EA = |x - \bar{x}| < 10^{-p}$

Esempi

7) x=8467915.78755.. , $\bar{x}$ =8467915.78654..
   $\bar{x}$ corretta a 2 decimali, EA=1e–3=$1\cdot10^{-3}<10^{-2}$
8) x=741.329642.. , $\bar{x}$ =741.320631..
   $\bar{x}$ corretta a 2 decimali, EA=9e–3=$9\cdot10^{-3}<10^{-2}$
9) x=.00741329642.. , $\bar{x}$ =.00741320631..
   $\bar{x}$ corretta a 7 decimali, EA=9e–8=$9\cdot10^{-8}<10^{-7}$

P2) Se x e $\bar{x}$ sono numeri interi con le ultime p cifre diverse, allora $EA = |x-\bar{x}| < 10^{p}$

Esempio
10) x=1245976 , $\bar{x}$ =1245679 si ha EA=2.9e2=$2.9\cdot10^{2}<10^{3}$

CIFRE SIGNIFICATIVE CORRETTE

P1) Se $\bar{x}$ è approssimazione di x con p cifre significative, allora $ER=\dfrac{|x-\bar{x}|}{|x|}<10^{-p+1}$

Esempi
11) 3000000000 e 0.00000100 1 cifra significativa; 0.004500300 5 cifre significative
12) x=8467915.787.. , $\bar{x}$ =8467915.786..
$\bar{x}$ corretta a 9 cifre significative, ER=1.1e−10=$1.1\cdot10^{-10}$
13) x=741.329642 .. , $\bar{x}$ =741.320631..
$\bar{x}$ corretta a 5 cifre significative, ER=1.2e−5=$1.2\cdot10^{-5}$
14) x=.00741329642.. , $\bar{x}$ =.00741320631..
$\bar{x}$ corretta a 5 cifre significative, ER=1.2e−5=$1.2\cdot10^{-5}$

ERRORI DI ROUND-OFF

Per memorizzare un numero reale, non di macchina, è necessario limitare la sua mantissa alle t cifre disponibili. Le tecniche di round−off utilizzate sono due.

TRONCAMENTO SEMPLICE

Le mantisse p comprese tra due mantisse di macchina consecutive p_1 e p_2, distanti tra loro N^{-t}, vengono troncate alla mantissa di macchina inferiore: $\bar{p}=p_1$.

Dati $x=pN^{q}$ e $\bar{x}=fl_T(x)=\bar{p}N^{q}$ si ha

$$EA \text{ nella mantissa}: \quad |p-\bar{p}|<N^{-t}$$
$$EA \text{ di round - off}: \quad |x-\bar{x}|=|p-\bar{p}|N^{q}<N^{q-t}$$
$$ER \text{ di round - off}: \quad \frac{|x-\bar{x}|}{|x|}=\frac{|p-\bar{p}|N^{q}}{|p|N^{q}}<N^{1-t}\rightarrow eps=N^{1-t}$$

Esempi
15) t=4, N=10, p=.4783 | 952 $\Rightarrow$ $\bar{p}$=.4783
16) t=6, N=2, p=.100110 | 111 $\Rightarrow$ $\bar{p}$=.100110

ARROTONDAMENTO SIMMETRICO

Le mantisse p comprese tra due mantisse di macchina consecutive p_1 e p_2, distanti tra loro N^{-t}, vengono così limitate:
- se la mantissa p cade prima del punto medio tra le due mantisse di macchina p_1 e p_2, si arrotonda a quella inferiore: $\bar{p}=p_1$;

– se la mantissa p cade dopo il punto medio tra le due mantisse di macchina p_1 e p_2, si arrotonda a quella superiore: $\overline{p} = p_2$.

Dati $x = pN^q$ e $\overline{x} = fl_A(x) = \overline{p}N^q$ si ha

$$EA \text{ nella mantissa} : \quad |p - \overline{p}| \le \frac{N^{-t}}{2}$$

$$EA \text{ di round - off} : \quad |x - \overline{x}| = |p - \overline{p}|N^q \le \frac{N^{q-t}}{2}$$

$$ER \text{ di round - off} : \quad \frac{|x - \overline{x}|}{|x|} = \frac{|p - \overline{p}|N^q}{|p|N^q} \le \frac{N^{1-t}}{2} \rightarrow eps = \frac{N^{1-t}}{2}$$

Esempi

17) t=4, N=10, p=.4783 | 952 $\Rightarrow$ $\overline{p}$ =.4784
t=4, N=10, p=.4783 | 252 $\Rightarrow$ $\overline{p}$ =.4783

18) t=6, N=2, p=.100110 | 111 $\Rightarrow$ $\overline{p}$ =.100111
t=6, N=2, p=.100110 | 011 $\Rightarrow$ $\overline{p}$ =.100110

La tecnica di arrotondamento simmetrico è più precisa della tecnica di troncamento semplice in quanto gli errori di round–off risultano dimezzati.

PRECISIONE DI MACCHINA

La precisione di macchina è una costante caratteristica dell'aritmetica floating–point e della tecnica di round–off utilizzate.
È la massima precisione di calcolo raggiungibile.
È il più piccolo numero macchina (potenza di 2: $eps = 2^{-t-1}$) tale che $(1+eps) \ge 1$ (vale il segno = nel caso di troncamento semplice (fl_T), il segno > nel caso di arrotondamento simmetrico (fl_A)).

$$eps \equiv \min\{\varepsilon \in F \,;\, fl(1+\varepsilon) \ge 1\}$$

CANCELLAZIONE NUMERICA

È una conseguenza della rappresentazione con precisione finita dei numeri reali e consiste nella perdita di cifre significative nel risultato della differenza tra due numeri quasi uguali.

Siano dati due numeri floating point $a = p_1N^q$ e $b = p_2N^q$; le mantisse p_1 e p_2 abbiano più di t cifre, ma siano rappresentabili soltanto con t cifre.
Se le mantisse p_1 e p_2 hanno le prime s cifre coincidenti, la mantissa $\overline{p}$ dell'operazione $\overline{a} - \overline{b}$, sottrazione tra i due numeri arrotondati a t cifre, ha soltanto le prime $t-s$ cifre significative, in quanto provengono da p_1 e p_2, mentre le restanti s cifre sono prive di significato.

Esempio

19) t=6 , N=10 , arrotondamento simmetrico
p_1=.147554 | 32·10^3 $\Rightarrow$ $\overline{p}_1$=.147554·10^3 e p_2=.147251 | 74·10^3 $\Rightarrow$ $\overline{p}_2$=.147252·10^3 quindi

$$p=(p_1-p_2)=.302584 \Rightarrow \bar{p}=(\bar{p}_1-\bar{p}_2)=.302\underline{000}$$

i tre zeri finali non sono significativi, infatti s=3 e quindi $t-s$=6−3=3 cifre significative in $\bar{p}$.

La cancellazione numerica è un problema insito negli operandi, mentre l'operazione di sottrazione amplifica soltanto gli errori di round−off degli operandi stessi.

Per evitare la cancellazione numerica a volte si può riformulare il problema in modo da non effettuare sottrazioni vere e proprie.

Esempio

20) L'equazione di secondo grado $x^2-2ax+b=0$ ha soluzioni $x_1=a+\sqrt{a^2-b}$ e $x_2=a-\sqrt{a^2-b}$;

se $|b|<<|a|$, cioè $b<<a^2$, la soluzione x_2 subisce cancellazione numerica in quanto $\sqrt{a^2-b}\approx a$.

La cancellazione numerica può essere evitata riformulando le due soluzioni nel modo seguente:

$$x_1=a+\text{sgn}(a)\sqrt{a^2-b}$$
$$x_2=b/x_1$$

CONSIDERAZIONI FINALI

Un generico problema numerico può essere scritto in forma esplicita come $y=f(x)$.

1. Nei dati è presente un errore di round−off $x-\bar{x}$ a cui corrisponde un errore finale $e_1=f(x)-f(\bar{x})$ dovuto al **condizionamento del problema**;
2. L'algoritmo è solitamente un'approssimazione f_1 più semplice del problema numerico e quindi comporta un errore $e_2=f(\bar{x})-f_1(\bar{x})$ dovuto alla **discretizzazione dell'algoritmo**;
3. Le operazioni di macchina hanno precisione finita, quindi anziché $f_1(\bar{x})$ si valuta $f_2(\bar{x})$ con un errore $e_3=f_1(\bar{x})-f_2(\bar{x})$ dovuto alla **stabilità dell'algoritmo**.

L'**errore complessivo** risulta

$$e=f(x)-f_2(\bar{x})=e_1+e_2+e_3$$

Obiettivo del Calcolo Numerico è lo sviluppo, con errori di discretizzazione e_2 nulli o arbitrariamente piccoli, di algoritmi stabili (cioè con e_3 piccoli).

ESERCIZI SVOLTI

1. I numeri reali a=123.54337624 e b=123.11111111 vengono introdotti in un calcolatore dove sono rappresentati in aritmetica floating point normalizzata con base N=10, t=7 cifre riservate alla mantissa e tecnica di troncamento semplice (fl_T).

Determinare il risultato $\bar{c}=\bar{a}-\bar{b}=\text{fpn}(\text{fpn}(a)-\text{fpn}(b))$ e confrontare $\bar{c}$ con il risultato esatto $c=a-b$, indicando il numero di cifre significative presenti di $\bar{c}$.

N=10, t=7, troncamento semplice $\rightarrow eps = N^{1-t} = 10^{-6}$

$a = .12354337624 \cdot 10^3$ $\qquad \bar{a} = .\underline{123}5433 \cdot 10^3$

$b = .12311111111 \cdot 10^3$ $\qquad \bar{b} = .\underline{123}1111 \cdot 10^3$

Le mantisse hanno s=3 cifre uguali e quindi il risultato avrà $t-s$=4 cifre significative:

$c = a - b = .43226513$

i 3 zeri finali non hanno significato.

$\bar{c} = \bar{a} - \bar{b} = .\underline{4322}000$

Poiché i due operandi sono 'quasi uguali' e 'contengono errori' (perché sono stati troncati a t cifre di mantissa), la sottrazione di macchina amplifica gli errori; il risultato presenta una perdita di cifre significative dovuta a cancellazione numerica.

---

2. Si consideri un elaboratore con aritmetica floating-point normalizzata con base N=10, t=8 cifre riservate alla mantissa e tecnica di arrotondamento per troncamento semplice (fl_T). Dati i numeri di macchina a=.23371258·10^{-4}, b=.33678429·10^{2} e c= −.33677811·10^{2}, calcolare le somme $x = a \oplus (b \oplus c)$ e $y = (a \oplus b) \oplus c$. Confrontare i risultati ottenuti con il risultato esatto s=a+b+c=.64137258 10^{-3} e spiegare il diverso comportamento delle due somme di macchina.

N=10, t=8, troncamento semplice $\rightarrow eps = N^{1-t} = 10^{-7}$

Prima somma: $x = a \oplus (b \oplus c)$

$$b \oplus c = d = .61800000 \cdot 10^{-3}$$

$$a \oplus d = .64137125|8 \cdot 10^{-3}$$

$$x = .64137125 \cdot 10^{-3}$$

Gli ultimi 5 zeri di d hanno significato perché gli operandi b e c sono privi di errore (sono numeri macchina) e quindi nel calcolo di d non si verifica cancellazione numerica.
Il risultato finale x è quello esatto, troncato a t=8 cifre.

Seconda somma: $y = (a \oplus b) \oplus c$

$$a \oplus b = d = .33678452|37 \cdot 10^{2}$$

$$\bar{d} = .\underline{3367}8452 \cdot 10^{2}$$

$$d \oplus c = .6410|0000 \cdot 10^{-3}$$

$$y = .6410|0000 \cdot 10^{-3}$$

L'operando $\bar{d}$ è un'approssimazione di d (mantissa troncata a t=8 cifre) e quindi contiene errori, inoltre $\bar{d}$ ha le prime 4 cifre della mantissa coincidenti con quelle di c (.3367). Nel corso dell'operazione

successiva, quindi, si verifica cancellazione numerica: soltanto le prime $t-s=8-4=4$ cifre sono significative (gli ultimi 4 zeri di y non hanno significato).

---

3. Supponendo di lavorare in aritmetica floating-point con $t=4$ cifre riservate alla mantissa e tecnica di arrotondamento simmetrico (fl_A), sommare i seguenti numeri prima in ordine ascendente (dal più piccolo al più grande) e poi in ordine discendente.
0.1580, 0.2653, $0.2581 \cdot 10^1$, $0.4298 \cdot 10^1$, $0.6266 \cdot 10^2$, $0.7555 \cdot 10^2$, $0.7889 \cdot 10^3$, $0.7767 \cdot 10^3$, $0.8999 \cdot 10^4$. Confrontare i risultati ottenuti con il risultato esatto $0.107101023 \cdot 10^5$.

$\sum$ ordine ascendente	$\sum$ ordine discendente
$.1580 \quad +$	$.8999 \cdot 10^4 \; +$
$.2653$	$.7767 \cdot 10^3$
$.4233 \quad +$	$.9776 \cdot 10^4 \; +$
$.2581 \cdot 10^1$	$.7889 \cdot 10^3$
$.3004 \cdot 10^1 +$	$.1056 \cdot 10^5 \; +$
$.4298 \cdot 10^1$	$.7555 \cdot 10^2$
$.7302 \cdot 10^1 +$	$.1064 \cdot 10^5 \; +$
$.6266 \cdot 10^2$	$.6266 \cdot 10^2$ non conta
$.6996 \cdot 10^2 +$	$.1070 \cdot 10^5 \; +$
$.7555 \cdot 10^2$	$.4298 \cdot 10^1$ non conta
$.1455 \cdot 10^3 \; +$	$.1070 \cdot 10^5 \; +$
$.7889 \cdot 10^3$	$.2581 \cdot 10^1$ non conta
$.9344 \cdot 10^3 \; +$	$.1070 \cdot 10^5 \; +$
$.7767 \cdot 10^3$	$.2653$ non conta
$.1711 \cdot 10^4 \; +$	$.1070 \cdot 10^5 \; +$
$.8999 \cdot 10^4$	$.1580$ non conta
$.1071 \cdot 10^5$	$.1070$
$x = .1071 \cdot 10^5$	$y = .1070 \cdot 10^5$

$s = .107101023 \cdot 10^5$

E' più corretto effettuare la somma in ordine ascendente (risultato x); infatti nel caso di somma in ordine discendente (risultato y) i valori più piccoli 'non contano' e vengono persi.

---

4. Riformulare il problema $y = \sqrt{x+\delta} - \sqrt{x}$, in modo da evitare la cancellazione numerica.

Si razionalizza in modo da eliminare la sottrazione effettiva:

$$y = \left(\sqrt{x+\delta} - \sqrt{x}\right)\frac{\sqrt{x+\delta}+\sqrt{x}}{\sqrt{x+\delta}+\sqrt{x}} = \frac{\delta}{\sqrt{x+\delta}+\sqrt{x}}$$

ESERCIZI PROPOSTI

1. Scrivere le rappresentazioni floating-point normalizzate dei seguenti numeri: 125.375; .0075343; 1.47512 e2.

[.125375 e3; .75343 e–2; .147512 e3]

2. Calcolare la soluzione dei seguenti gruppi di sistemi e confrontare i risultati (Si noti l'effetto dal condizionamento del problema).

a) $\begin{cases} 3\cdot x + 6\cdot y = 8 \\ 2\cdot x + 4.001\cdot y = 8 \end{cases}$ $\begin{cases} 3\cdot x + 6\cdot y = 8 \\ 2\cdot x + 4.002\cdot y = 8 \end{cases}$ $\begin{cases} 3\cdot x + 6\cdot y = 8 \\ 2\cdot x + 4.003\cdot y = 8 \end{cases}$

b) $\begin{cases} 2\cdot x + 6\cdot y = 8 \\ 2\cdot x + 6.001\cdot y = 8 \end{cases}$ $\begin{cases} 2\cdot x + 6\cdot y = 8 \\ 2\cdot x + 6.001\cdot y = 8.001 \end{cases}$ $\begin{cases} 2\cdot x + 6\cdot y = 8 \\ 2\cdot x + 6.002\cdot y = 8.001 \end{cases}$

[a) x=–5331 y=2667; x= –2664 y=1333; x= –1775 y=889]
[b) x=4 y=0; x=1 y=1; x=2.5 y=0.5]

OPERAZIONI TRA MATRICI

Riferimento al testo: Cap. II

Una matrice **A** è una tabella ordinata di numeri, in generale con *m* righe e *n* colonne.

Il generico elemento a_{ij} è posizionato nella casella intersezione tra la riga *i*-esima (1° indice) e la colonna *j*-esima (2° indice)

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & \cdot & \cdot & a_{1n} \\ a_{21} & a_{22} & \cdot & \cdot & \cdot & \cdot & a_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{m1} & a_{m2} & \cdot & \cdot & \cdot & \cdot & a_{mn} \end{bmatrix} \in R^{m,n}$$

A seconda dei valori assunti dall'indice di riga *m* e di colonna *n* si distingue tra:
- vettore riga $(a_{11},...,a_{1n})$ se $m=1$;
- vettore colonna $(a_{11},...,a_{1n})^T$ se $n=1$;
- matrice quadrata se $m=n$.

Nel seguito si farà riferimento a matrici $\mathbf{A} \in R^{n,n}$, cioè reali e quadrate, che potranno essere:
- *piene*: elementi a_{ij} quasi tutti non nulli;
- *sparse*: gran parte degli elementi a_{ij} nulli;
- *strutturate*: elementi disposti in modo particolare;
- *diagonali*: $a_{ij}=0$ per $i \neq j$;
- *tridiagonali*: $a_{ij}=0$ per $|i-j|>1$;
- *identità*: $a_{ij}=0$ per $i \neq j$ e $a_{ij}=1$ per $i=j$ ($a_{ij}=d_{ij}$ delta di Kronecker);
- *zero*: $a_{ij}=0$ per $\forall\, i, j$;
- *a banda* (ampiezza $2k+1$): $a_{ij}=0$ per $|i-j|>k$ (ampiezza = max n. elementi $\neq 0$ su riga/colonna);
- *triangolari superiori*: $a_{ij}=0$ per $i>j$;
- *triangolari inferiori*: $a_{ij}=0$ per $i<j$.

La *somma* (e la *differenza*) di matrici sono definite soltanto tra matrici dello stesso ordine (stesso numero di righe e stesso numero di colonne). Il risultato è una matrice i cui elementi si ottengono sommando o sottraendo gli elementi corrispondenti

$$\mathbf{A}, \mathbf{B} \in R^{m,n} \quad \rightarrow \quad \mathbf{C} = \mathbf{A} \pm \mathbf{B} \quad ; \quad c_{ij} = a_{ij} \pm b_{ij}$$

Esempio

$$1) \quad \mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 3 & 2 & 1 \\ 2 & 3 & 4 \end{bmatrix} \quad \rightarrow \quad \mathbf{C} = \mathbf{A} + \mathbf{B} = \begin{bmatrix} 1+3 & 2+2 & 3+1 \\ 4+2 & 5+3 & 6+4 \end{bmatrix} = \begin{bmatrix} 4 & 4 & 4 \\ 6 & 8 & 10 \end{bmatrix}$$

- Vale la proprietà commutativa: $\mathbf{A}+\mathbf{B}=\mathbf{B}+\mathbf{A}$;
- vale la proprietà associativa $(\mathbf{A}+\mathbf{B})+\mathbf{C}=\mathbf{A}+(\mathbf{B}+\mathbf{C})$;

- la proprietà distributiva vale mantenendo l'ordine: **A**(**B**+**C**)=**AB**+**AC**, (**A**+**B**)**C** = **AC**+**BC**.

Il *prodotto* di una *matrice* per uno *scalare* è una matrice ottenuta moltiplicando ciascun elemento per lo scalare

$$\mathbf{A} \in R^{m,n} \qquad k \in R \rightarrow \mathbf{B} = k\mathbf{A}\ ;\ b_{ij} = k a_{i,j}$$

Esempio

2) $\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \quad k = 5 \quad \rightarrow \quad \mathbf{B} = \begin{bmatrix} 5 & 10 & 15 \\ 20 & 25 & 30 \end{bmatrix}$

Il *prodotto* tra due *matrici* è definito se e solo se il numero di colonne della 'prima' matrice (quella di sinistra) è uguale al numero di righe della 'seconda' (quella di destra). Il risultato è una matrice i cui elementi si ottengono effettuando il prodotto righe×colonne

$$\mathbf{A} \in R^{m,p}\ , \mathbf{B} \in R^{p,n} \rightarrow \mathbf{C} = \mathbf{A}\,\mathbf{B} \in R^{m,n}\ ;\ c_{ij} = \sum_{k=1}^{p} a_{ik} \cdot b_{kj}$$

Esempio

3) $\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 5 \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \rightarrow \mathbf{C} = \mathbf{A}\,\mathbf{B} = \begin{bmatrix} 1\cdot 1 + 2\cdot 1 & 1\cdot(-1) + 2\cdot 1 \\ 3\cdot 1 + 5\cdot 1 & 3\cdot(-1) + 5\cdot 1 \end{bmatrix} = \begin{bmatrix} 3 & 1 \\ 8 & 2 \end{bmatrix}$

- La proprietà commutativa del prodotto non vale: **AB**≠**BA**;
- la proprietà distributiva vale mantenendo l'ordine: (**AB**)**C** = **A**(**BC**).

MATRICE TRASPOSTA

Data una matrice **A**, la sua matrice trasposta $\mathbf{A}^T$ si ottiene scrivendo le righe di **A**, nell'ordine, come colonne (cioè 'scambiando' le righe con le colonne): $a_{ij}{}^T = a_{ji}$.

Valgono le seguenti proprietà:
- $(\mathbf{A}^T)^T = \mathbf{A}$, $(k_1\mathbf{A})^T = k_1\mathbf{A}^T$;
- $(\mathbf{A}+\mathbf{B})^T = \mathbf{A}^T + \mathbf{B}^T$;
- $(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T$ (si inverte l'ordine!).

Esempio

4) $\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \rightarrow \mathbf{A}^T = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$

MATRICE SIMMETRICA

Una matrice **A** è simmetrica se $\mathbf{A}^T = \mathbf{A}$ (la 'riflessione' della matrice rispetto alla diagonale principale la lascia inalterata).
La matrice **A** è necessariamente quadrata, con elementi $a_{ij} = a_{ji}$.

Esempio

5) $\mathbf{A} = \begin{bmatrix} 2 & -3 & 0 \\ -3 & 1 & 4 \\ 0 & 4 & -7 \end{bmatrix}$ è simmetrica.

DETERMINANTE DI MATRICE

Il determinante det(**A**) o |**A**| è uno scalare associato a ogni matrice quadrata $\mathbf{A} \in R^{n,n}$.

Nel caso di matrice 2×2, il determinante si calcola come:

$$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{21}a_{12}$$

Esempio

6) $\begin{vmatrix} 1 & 2 \\ 3 & 4 \end{vmatrix} = 4 - 6 = -2$

Nel caso di matrice 3×3, il determinante si calcola come:

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11}\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12}\begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13}\begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}$$

Ogni elemento della prima riga risulta moltiplicato per il determinante della sottomatrice ottenuta cancellando la prima riga e la colonna contenente l'elemento stesso; tra i diversi termini si deve rispettare la seguente successione dei segni:

$$\begin{pmatrix} + & - & + \\ - & + & - \\ + & - & + \end{pmatrix}$$

Facendo riferimento a una riga qualsiasi (riga 2 o riga 3) si ottiene ovviamente lo stesso risultato

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = -a_{21}\begin{vmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{vmatrix} + a_{22}\begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} - a_{23}\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix}$$

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{31}\begin{vmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \end{vmatrix} - a_{32}\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} + a_{33}\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$$

È quindi conveniente fare riferimento alla riga che contiene più elementi nulli (fare attenzione alla sequenza dei segni da applicare!)

Esempio

7) $$\begin{vmatrix} 1 & -2 & -1 \\ 1 & -1 & -3 \\ 2 & -1 & 9 \end{vmatrix} = 1\cdot\begin{vmatrix} -1 & -3 \\ -1 & 9 \end{vmatrix} - (-2)\cdot\begin{vmatrix} 1 & -3 \\ 2 & 9 \end{vmatrix} + (-1)\cdot\begin{vmatrix} 1 & -1 \\ 2 & -1 \end{vmatrix} = 1\cdot(-12) - (-2)\cdot(15) + (-1)\cdot(1) = 17$$

Nel caso di matrice $n\times n$, l'estensione è ovvia; si ricorda di mantenere la sequenza dei segni.

Valgono le seguenti proprietà:

- $\det(\mathbf{A})=\det(\mathbf{A}^T)$;
- se $\mathbf{A}$ ha una riga (colonna) nulla $\rightarrow \det(\mathbf{A})=0$;
- se $\mathbf{A}$ ha due righe (colonne) uguali $\rightarrow \det(\mathbf{A})=0$;
- se $\mathbf{A}$ è triangolare (sup. o inf.) $\rightarrow \det(\mathbf{A})=\prod_{i=1}^{n} a_{ii}$.

MATRICE SIMMETRICA DEFINITA (O SEMIDEFINITA) POSITIVA

Una matrice quadrata e simmetrica $\mathbf{A}\in R^{n,n}$ è definita positiva (semidefinita positiva) se $\mathbf{x}^T\cdot\mathbf{A}\cdot\mathbf{x} > 0$ ($\mathbf{x}^T\cdot\mathbf{A}\cdot\mathbf{x} \geq 0$) per ogni vettore non nullo $\mathbf{x}\in R^n$.

Poichè lo scalare $\mathbf{x}^T\cdot\mathbf{A}\cdot\mathbf{x}$ ha il significato fisico di un'energia, è ovvio che sia sempre positivo.

CRITERIO DI SYLVESTER

È un criterio per stabilire se una matrice simmetrica $\mathbf{A}$ è definita positiva o meno.

La matrice simmetrica $\mathbf{A}\in R^{n,n}$ è definita positiva se e solo se $\det(\mathbf{A}_k)>0$ per $k=1, ..., n$ dove gli $\mathbf{A}_k$ sono i minori principali della matrice che si ottengono dall'intersezione delle prime k righe e colonne della matrice stessa.

Due conseguenze di tale criterio affermano che:

- se $\mathbf{A}$ è definita positiva allora gli elementi sulla diagonale principale sono tutti positivi ($a_{ii}>0$ per $i=1, ..., n$);
- se $\mathbf{A}$ è definita positiva allora l'elemento di modulo massimo si trova sulla diagonale principale e inoltre $|a_{ij}|^2<a_{ii}a_{jj}$ per $i\neq j$.

Esempio

8) $$\mathbf{A} = \begin{bmatrix} 4 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 8 \end{bmatrix} \rightarrow \det(\mathbf{A}_1) = 4 \;\; \det(\mathbf{A}_2) = 3 \;\; \det(\mathbf{A}_3) = 24 \rightarrow \mathbf{A} \text{ è simmetrica definita positiva.}$$

MATRICE A DIAGONALE DOMINANTE

Una matrice $\mathbf{A}\in R^{n,n}$ è a diagonale dominante per righe se e solo se $|a_{ii}|>\sum_{\substack{j=1\\ j\neq i}}^{n}\left|a_{ij}\right|$ per i=1, ..., n.

- Se **A** è simmetrica a diagonale dominante e con elementi diagonale tutti positivi allora la matrice **A** è necessariamente definita positiva.

VETTORI LINEARMENTE INDIPENDENTI

k vettori, riga o colonna, sono linearmente indipendenti se l'unica combinazione lineare nulla è quella che si ottiene utilizzando coefficienti tutti nulli.

- Se $a_1\cdot\mathbf{a_1}+a_2\cdot\mathbf{a_2}+....+a_k\cdot\mathbf{a_k}=0$ con $a_1=a_2=...=a_k=0$ allora i vettori $\mathbf{a_i}$ (i=1, ..., k) sono LI fra loro.

MATRICE NON SINGOLARE

Una matrice $\mathbf{A}\in R^{n,n}$ è non singolare se e solo se le sue righe (colonne) sono vettori linearmente indipendenti, cioè se e solo se det(**A**)≠0.
Viceversa se det(**A**)=0 allora la matrice **A** è singolare.

RANGO DI MATRICE

Il rango di una matrice $\mathbf{A}\in R^{n,n}$ corrisponde al massimo numero di vettori riga (colonna) linearmente indipendenti.

- Se det(**A**)≠0, cioè se **A** è non singolare, allora rango(**A**)=n;
- se det(**A**)=0, cioè se **A** è singolare allora rango(**A**)<n e bisogna considerarne i minori finchè se ne trova uno, di ordine r, non singolare (rango(**A**)=r).

Esempi

9) $\mathbf{A}=\begin{bmatrix}1 & 2\\ 3 & 4\end{bmatrix}$ $\rightarrow$ $\det(\mathbf{A})=4-6\neq 0$ $\quad\text{rango}(\mathbf{A})=2$, i vettori riga sono tutti LI

10) $\mathbf{A}=\begin{bmatrix}1 & 2\\ 2 & 4\end{bmatrix}$ $\rightarrow$ $\det(\mathbf{A})=4-4=0$ $\quad\text{rango}(\mathbf{A})<2=1$, infatti $\mathbf{r_2}=2\cdot\mathbf{r_1}$ e quindi i due vettori riga $\mathbf{r_2}$ e $\mathbf{r_1}$ sono LD tra loro.

MATRICE INVERSA

Se $\mathbf{A}\in R^{n,n}$ è non singolare, cioè se det(**A**)≠0, allora esiste una e una sola matrice inversa $\mathbf{A}^{-1}\in R^{n,n}$ non singolare tale che $\mathbf{A}\cdot\mathbf{A}^{-1}=\mathbf{A}^{-1}\cdot\mathbf{A}=\mathbf{I}$ (matrice identità).

Valgono le seguenti proprietà:
- $(\mathbf{AB})^{-1}=\mathbf{B}^{-1}\mathbf{A}^{-1}$;
- $(\mathbf{A}^{T})^{-1}=(\mathbf{A}^{-1})^{T}$;

- se **A** è a diagonale dominante allora **A** è non singolare (det(**A**)≠0) e quindi esiste $\mathbf{A}^{-1}$;
- se **A** è simmetrica definita positiva allora **A** è non singolare e quindi esiste $\mathbf{A}^{-1}$ simmetrica definita positiva.

Le seguenti affermazioni sono tra loro equivalenti:
- **A** è invertibile;
- rango(**A**)=*n*;
- le righe (colonne) di **A** sono vettori tutti LI (linearmente indipendenti).

NORMA DI VETTORI E MATRICI

La norma è una funzione che a ogni vettore o matrice associa un numero reale e positivo, che si indica con $\| \ \|$, e che misura una distanza.
Si può quindi utilizzare per valutare la convergenza, in termini di distanza, del vettore soluzione dell'algoritmo al vettore soluzione esatta.

Dati due vettori $\mathbf{u},\mathbf{v}\in R^n$, la lunghezza o norma-2 del vettore **u**, indicata $\|\mathbf{u}\|_2$, è definita come radice quadrata del prodotto scalare:

$$\|\mathbf{u}\|_2 = \sqrt{\mathbf{u}^{\mathrm{T}} \cdot \mathbf{u}} = \sqrt{u_1^2 + \cdots + u_n^2} \, .$$

La distanza tra i vettori **u** e **v** si calcola quindi come

$$\| \mathbf{u} - \mathbf{v} \|_2 = \sqrt{(\mathbf{u}-\mathbf{v})^{\mathrm{T}} \cdot (\mathbf{u}-\mathbf{v})} = \sqrt{(u_1 - v_1)^2 + \cdots + (u_n - v_n)^2}$$

Esempio

11) $\mathbf{u}=(1, 7)$, $\mathbf{v}=(6, -5) \rightarrow \|\mathbf{u}-\mathbf{v}\| = \sqrt{(1-6)^2 + (7+5)^2} = \sqrt{25+144} = \sqrt{169} = 13$

NORMA DI VETTORE

La norma di un vettore **x**, o $\|\mathbf{x}\|$, è una funzione $R^n \rightarrow R$ (cioè da R^n a R) con le seguenti proprietà:
- $\|\mathbf{x}\| > 0 \ \forall \ x \neq 0$ e $\|\mathbf{x}\| = 0$ se e solo se $x = 0$ (non esistono distanze negative);
- $\| k_1 \mathbf{x} \| = | k_1 | \cdot \| \mathbf{x} \| \quad \forall \, k_1 \in R$;
- $\| \mathbf{x} + \mathbf{y} \| \leq \| \mathbf{x} \| + \| \mathbf{y} \|$ (disuguaglianza triangolare);
- $\| \mathbf{x} - \mathbf{y} \| \geq \| \mathbf{x} \| - \| \mathbf{y} \|$ (deducibile dalle precedenti).

Le norme di vettore più usate sono la norma-∞, la norma-1 e la norma-2 (o norma euclidea):

$$\| \mathbf{x} \|_\infty = \max_{1 \leq i \leq n} |x_i|$$

$$\| \mathbf{x} \|_1 = \sum_{i=1}^{n} |x_i|$$

$$\| \mathbf{x} \|_2 = \sqrt{\sum_{i=1}^{n} x_i^2} = \sqrt{\mathbf{x}^{\mathrm{T}} \cdot \mathbf{x}}$$

La norma-2 gode della proprietà pitagorica $\|\mathbf{x}+\mathbf{y}\|_2^2=\|\mathbf{x}\|_2^2+\|\mathbf{y}\|_2^2$ e della disuguaglianza di Cauchy-Schwarz $\|\mathbf{x}\|_2\cdot\|\mathbf{y}\|_2\geq\left|\mathbf{x}^T\mathbf{y}\right|$.

Esempio

12) $\mathbf{u}=(1, 7) \rightarrow \|\mathbf{u}\|_\infty=\max|1,7|=7 \qquad \|\mathbf{u}\|_1=1+7=8 \qquad \|\mathbf{u}\|_2=\sqrt{1^2+7^2}=\sqrt{50}$

NORMA DI MATRICE

La norma di una matrice **A**, o $\|\mathbf{A}\|$, è una funzione $R^{n,n}\rightarrow R$ con le seguenti proprietà:

- $\|\mathbf{A}\|>0\ \forall\ \mathbf{A}\neq 0$ e $\|\mathbf{A}\|=0$ se e solo se **A**=0 (non esistono distanze negative);
- $\|k_1\mathbf{A}\|=|k_1|\cdot\|\mathbf{A}\| \quad \forall k_1\in R$;
- $\|\mathbf{A}+\mathbf{B}\|\leq\|\mathbf{A}\|+\|\mathbf{B}\|$ (disuguaglianza triangolare);
- $\|\mathbf{A}\cdot\mathbf{B}\|\leq\|\mathbf{A}\|\cdot\|\mathbf{B}\|$.

Le norme di matrice più usate sono la norma-∞, la norma-1 e la norma-2 (o norma spettrale):

$$\|\mathbf{A}\|_\infty=\max_{1\leq i\leq n}\sum_{j=1}^{n}|a_{ij}|$$

$$\|\mathbf{A}\|_1=\max_{1\leq j\leq n}\sum_{i=1}^{n}|a_{ij}|$$

$$\|\mathbf{A}\|_2=\sqrt{\rho\left(\mathbf{A}^T\cdot\mathbf{A}\right)}=\sqrt{\left|\lambda_{\max}\,\mathrm{di}\left(\mathbf{A}^T\cdot\mathbf{A}\right)\right|}$$

dove λ_{max} è l'autovalore di modulo massimo della matrice **A**.

- La norma-∞ è il valore massimo calcolato facendo la somma degli elementi sulle righe;
- la norma-1 è il valore massimo calcolato facendo la somma degli elementi sulle colonne.

Esempio

13) Data la matrice $\mathbf{A}=\begin{bmatrix}1&2&1\\2&1&2\\1&0&1\end{bmatrix}$ si ha

$$\|\mathbf{A}\|_\infty=\max_{1\leq i\leq n}\sum_{j=1}^{n}|a_{ij}|=\max\{(1+2+1)\ (2+1+2)\ (1+1)\}=\max\{4\ 5\ 2\}=5$$

$$\|\mathbf{A}\|_1=\max_{1\leq j\leq n}\sum_{i=1}^{n}|a_{ij}|=\max\{(1+2+1)\ (2+1)\ (1+2+1)\}=\max\{4\ 3\ 4\}=4$$

COMPATIBILITÀ DI NORMA

La condizione di compatibilità tra norma di matrice e norma di vettore è che $\|\mathbf{Ax}\|\leq\|\mathbf{A}\|\|\mathbf{x}\|$.

Le norme utilizzate (∞, 1 e 2) sono compatibili e la condizione di compatibilità viene solitamente usata nelle maggiorazioni.

ESERCIZI SVOLTI

1. Calcolare il prodotto $(\mathbf{x}\ \mathbf{y})\cdot\begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{C} & \mathbf{D}\end{bmatrix}\cdot(\mathbf{x}\ \mathbf{y})^{\mathrm{T}}$ sapendo che $\mathbf{x}, \mathbf{y} \in R^n$ sono vettori riga e $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D} \in R^{n,n}$ sono matrici quadrate.

$$(\mathbf{x}\ \mathbf{y})\cdot\begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{C} & \mathbf{D}\end{bmatrix}\cdot(\mathbf{x}\ \mathbf{y})^{\mathrm{T}}=(\mathbf{x}\ \mathbf{y})\cdot\begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{C} & \mathbf{D}\end{bmatrix}\cdot\begin{pmatrix}\mathbf{x}^{\mathrm{T}}\\ \mathbf{y}^{\mathrm{T}}\end{pmatrix}=(\mathbf{x}\ \mathbf{y})\cdot\begin{pmatrix}\mathbf{A}x^{\mathrm{T}}+\mathbf{B}y^{\mathrm{T}}\\ \mathbf{C}x^{\mathrm{T}}+\mathbf{D}y^{\mathrm{T}}\end{pmatrix}=\mathbf{x}\mathbf{A}\mathbf{x}^{\mathrm{T}}+\mathbf{x}\mathbf{B}\mathbf{y}^{\mathrm{T}}+\mathbf{y}\mathbf{C}\mathbf{x}^{\mathrm{T}}+\mathbf{y}\mathbf{D}\mathbf{y}^{\mathrm{T}}\in R$$

---

2. Verificare che il prodotto di 2 matrici triangolari superiori è una matrice triangolare superiore.

$$\begin{bmatrix}a & b\\ 0 & c\end{bmatrix}\cdot\begin{bmatrix}d & e\\ 0 & f\end{bmatrix}=\begin{bmatrix}ad & ae+bf\\ 0 & cf\end{bmatrix}$$ verificato.

---

3. Sviluppare l'algoritmo per calcolare il prodotto scalare tra vettori $\mathbf{x}^{\mathrm{T}}\cdot\mathbf{y}=z$ dove $\mathbf{x}, \mathbf{y} \in R^n$ mentre $z \in R$.

Algoritmo. Vetvet(n,**x**,**y**,z)

Commento. L'algoritmo calcola il prodotto scalare vettore×vettore $\mathbf{x}^{\mathrm{T}}\cdot\mathbf{y}=\sum_{i=1}^{n} x_i y_j = z$

Parametri. Input: n, **x**, **y**
Output: z

1. $z=0$
2. Ciclo 1: $i=1, ..., n$
3. $z=z+x_i\cdot y_i$
4. Fine Ciclo 1

---

4. Sviluppare l'algoritmo per calcolare il prodotto matrice×vettore $\mathbf{A}\chi\mathbf{x}=\mathbf{y}$ dove $\mathbf{A} \in R^{n,n}$ mentre $\mathbf{x}, \mathbf{y} \in R^n$.

Algoritmo. Matvet(n,**A**,**x**,**y**)

Commento. L'algoritmo calcola il prodotto matrice×vettore $\mathbf{A}\chi\mathbf{x}=\mathbf{y}$ con $y_i=\sum_{j=1}^{n} a_{ij}x_j$ dove $i=1, n$

Parametri. Input: n, **A**, **x**
Output: **y**

1. Ciclo 1: $i=1, ..., n$

2. $y_i=0$
3. Ciclo 2: $j=1, ..., n$
4. $y_i=y_i+a_{ij}\cdot x_j$
5. Fine Ciclo 2
6. Fine Ciclo 1

---

5. Data una matrice **A** dimostrare che la matrice $\mathbf{B}=\mathbf{A}^T\cdot\mathbf{A}$ è semidefinita positiva.

Bisogna dimostrare che $\mathbf{x}^T\cdot\mathbf{B}\cdot\mathbf{x}\geq 0$.

$\mathbf{x}^T\cdot\mathbf{B}\cdot\mathbf{x}=\mathbf{x}^T\cdot\mathbf{A}^T\cdot\mathbf{A}\cdot\mathbf{x}=(\mathbf{A}\cdot\mathbf{x})^T\cdot(\mathbf{A}\cdot\mathbf{x})=\mathbf{y}^T\cdot\mathbf{y}=\sum_{i=1}^{n} y_i\cdot y_i=\sum_{i=1}^{n} y_i^2\geq 0$ sempre. (cvd)

---

6. Sia data la matrice $\begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{B}^T & \mathbf{C}\end{bmatrix}$ definita positiva. Dimostrare che anche le matrici **A** e **C** lo sono.

Analogamente all'esercizio **1.** si ha:

$$\begin{pmatrix}\mathbf{x}^T & \mathbf{y}^T\end{pmatrix}\cdot\begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{B}^T & \mathbf{C}\end{bmatrix}\cdot\begin{pmatrix}\mathbf{x}\\ \mathbf{y}\end{pmatrix}>0\rightarrow\begin{pmatrix}\mathbf{x}^T & \mathbf{y}^T\end{pmatrix}\cdot\begin{pmatrix}\mathbf{Ax}+\mathbf{By}\\ \mathbf{B}^T\mathbf{x}+\mathbf{Cy}\end{pmatrix}=\mathbf{x}^T\mathbf{Ax}+\mathbf{x}^T\mathbf{By}+\mathbf{y}^T\mathbf{B}^T\mathbf{x}+\mathbf{y}^T\mathbf{Cy}>0$$

Poiché i vettori **x** e **y** possono essere scelti qualsiasi:
- se $\mathbf{x}=0$ allora $\mathbf{y}^T\mathbf{Cy}>0$ e quindi **C** è definita positiva;
- se $\mathbf{y}=0$ allora $\mathbf{x}^T\mathbf{Ax}>0$ e quindi **A** è definita positiva. (cvd)

ESERCIZI PROPOSTI

1. Verificare che il prodotto di 2 matrici triangolari inferiori è una matrice triangolare inferiore.

2. Sviluppare l'algoritmo per calcolare il prodotto matrice tridiagonale×vettore $\mathbf{Ax}=\mathbf{y}$ con $\mathbf{A}\in R^{n,n}$ tridiagonale e $\mathbf{x},\mathbf{y}\in R^n$. Utilizzare esclusivamente vettori.

[Tridvet]

3. Sviluppare l'algoritmo per calcolare il prodotto matrice×matrice $\mathbf{AB}=\mathbf{C}$ con $\mathbf{A},\mathbf{B},\mathbf{C}\in R^{n,n}$. Ottimizzare quindi l'algoritmo nel caso di matrici triangolari superiori.

[Matmat]

4. Calcolare i risultati delle seguenti operazioni tra matrici.

$$\mathbf{A}=\begin{bmatrix}-1 & 3 & 2\\ 4 & 0 & 1\\ -2 & 1 & 5\end{bmatrix}\quad \mathbf{B}=\begin{bmatrix}4 & 1 & 1\\ 3 & 2 & -2\\ 1 & 2 & -3\end{bmatrix}\rightarrow \mathbf{C}=\mathbf{A}+\mathbf{B} \qquad \left[\mathbf{C}=\begin{bmatrix}3 & 4 & 3\\ 7 & 2 & -1\\ -1 & 3 & 2\end{bmatrix}\right]$$

$$\mathbf{A}=\begin{bmatrix}3 & -2 & 1\\ 2 & 1 & -4\end{bmatrix}\quad k=\frac{1}{2}\rightarrow \mathbf{B}=k\mathbf{A} \qquad \left[\mathbf{B}=\begin{bmatrix}\frac{3}{2} & -1 & \frac{1}{2}\\ 1 & \frac{1}{2} & -2\end{bmatrix}\right]$$

$$\mathbf{A}=\begin{bmatrix}1 & 2 & 3\\ 2 & 3 & 4\\ 4 & 5 & 6\end{bmatrix}\quad \mathbf{B}=\begin{bmatrix}1 & -1\\ 3 & -2\\ -1 & 1\end{bmatrix}\rightarrow \mathbf{C}=\mathbf{AB} \qquad \left[\mathbf{C}=\begin{bmatrix}4 & -2\\ 7 & -4\\ 13 & -8\end{bmatrix}\right]$$

5. Calcolare il determinante delle seguenti matrici.

$$A=\begin{bmatrix}0 & 0\\ 2 & 5\end{bmatrix}\quad \mathbf{B}=\begin{bmatrix}1 & 2 & 3\\ 0 & 1 & 2\\ 1 & 0 & 3\end{bmatrix}\quad \mathbf{C}=\begin{bmatrix}1 & 2 & 3\\ 0 & 1 & 2\\ 0 & 0 & 3\end{bmatrix} \qquad [\det(\mathbf{A})=0\ ,\ \det(\mathbf{B})=4\ ,\ \det(\mathbf{C})=3]$$

6. Calcolare il rango delle seguenti matrici.

$$A=\begin{bmatrix}1 & 2\\ -2 & 1\end{bmatrix}\quad \mathbf{B}=\begin{bmatrix}1 & 1 & 1\\ 1 & 2 & 3\\ 0 & 1 & 1\end{bmatrix}\quad \mathbf{C}=\begin{bmatrix}1 & 2 & -1\\ 2 & 4 & -2\\ -1 & -2 & 1\end{bmatrix} \qquad [\text{rango}(\mathbf{A})=2\ ,\ \text{rango}(\mathbf{B})=3\ ,\ \text{rango}(\mathbf{C})=1]$$

7. Data la matrice diagonale $A=\begin{bmatrix}a & 0\\ 0 & b\end{bmatrix}$ calcolare la matrice inversa $\mathbf{A}^{-1}$. $\qquad \left[A^{-1}=\begin{bmatrix}1/a & 0\\ 0 & 1/b\end{bmatrix}\right]$

8. Calcolare la distanza tra i vettori **u**=(3, −5, 4) e **v**=(6, 2, −1) e tra i vettori **x**=(5, 3, −2, −4, −1) e **y**=(2, −1, 0, −7, 2).

$$[\|\mathbf{u}-\mathbf{v}\|_2=\sqrt{83}\ ,\ \|\mathbf{x}-\mathbf{y}\|_2=\sqrt{47}\ ]$$

9. Calcolare le norme ∞, 1 e 2 del vettore **v**=(6, −5).

$$[\|\mathbf{v}\|_\infty=6\ ,\ \|\mathbf{v}\|_1=11\ ,\ \|\mathbf{v}\|_2=\sqrt{61}\ ]$$

10. Calcolare le norme ∞ e 1 della matrice $\mathbf{A}=\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\end{bmatrix}$

$$[\|\mathbf{A}\|_\infty=24\ ,\ \|\mathbf{A}\|_1=18\ ]$$

SISTEMI DI EQUAZIONI LINEARI

(Riferimento al testo: Cap. III)

INTRODUZIONE

Un sistema di n equazioni lineari in n incognite x_i ($i = 1, n$) può essere scritto in forma matriciale come:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots\ldots + a_{1n}x_n = b_1 \\ \ldots\ldots\ldots\ldots\ldots \\ a_{i1}x_1 + a_{i2}x_2 + a_{i3}x_3 + \ldots\ldots + a_{in}x_n = b_i \\ \ldots\ldots\ldots\ldots\ldots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \ldots\ldots + a_{nn}x_n = b_n \end{cases} \quad \Rightarrow \quad \mathbf{Ax} = \mathbf{b}$$

dove $\mathbf{A} \in R^{n,n}$ è la matrice dei coefficienti (dimensione (n, n)), $\mathbf{x} \in R^n$ il vettore delle incognite (dimensione (n, 1)), $b \in R^n$ il vettore dei termini noti (dimensione (n, 1)).

La soluzione del sistema di equazioni lineari esiste ed è unica se e solo se la matrice **A** è non singolare, cioè le sue colonne (righe) sono vettori linearmente indipendenti. In questo caso det(**A**)$\neq 0$ (ovvero rango(**A**) = n) e quindi esiste la matrice inversa $\mathbf{A}^{-1}$ tale che $\mathbf{x}=\mathbf{A}^{-1}\mathbf{b}$.

I metodi di soluzione si dividono in:

METODI DIRETTI Gauss (decomposizione GA=U e fattorizzazione LU)
Choleski

- si applicano a matrici **A** dense e di ordine non elevato (10^2, 10^3)
- effettuano un numero finito di operazioni su **A** e su **b** che trasformano il sistema iniziale in un sistema equivalente più semplice con matrice dei coefficienti triangolare
- occorre memorizzare tutti gli elementi della matrice **A**
- sono affetti soltanto da errori di round-off

METODI ITERATIVI Jacobi
Gauss-Seidel
SOR

- si applicano a matrici **A** sparse e di ordine elevato (10^4, 10^6)
- operano esclusivamente con la matrice **A** iniziale e generano una successione (infinita) di vettori convergente alla soluzione **x**
- è sufficiente memorizzare soltanto gli elementi non nulli della matrice **A**
- sono affetti sia da errori di round-off sia da errori di troncamento analitico (discretizzazione)

Nel seguito si affronteranno per primi i metodi di soluzione diretti e successivamente quelli iterativi.

SISTEMI TRIANGOLARI

Se il sistema di equazioni lineari ha *matrice dei coefficienti triangolare superiore*, a partire dall'ultima equazione si risolve il sistema a ritroso con tecnica di *back-sostitution*: dall'ultima equazione si ricava il valore dell'incognita x_n e, per sostituzione progressiva nelle equazioni via via precedenti, si ricavano tutte le altre x_i. Le equazioni quindi si disaccoppiano via via e le soluzioni x_i si ricavano mediante la formula ricorsiva riportata nel seguito.

Dato il seguente sistema lineare con matrice dei coefficienti triangolare superiore:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots\ldots + a_{1n}x_n = b_1 \\ a_{22}x_2 + a_{23}x_3 + \ldots\ldots + a_{2n}x_n = b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots = \ldots. \\ a_{nn}x_n = b_n \end{cases} \quad \text{con } a_{ii} \neq 0, \text{ cioè } \mathbf{A} \text{ non singolare}$$

la sua soluzione si ricava con la formula ricorsiva di back-sostitution (crf. Algoritmo: Backsost):

$$\begin{cases} x_n = \dfrac{b_n}{a_{nn}} \\ x_k = \dfrac{(b_k - \sum_{j=k+1}^{n} a_{kj}x_j)}{a_{kk}} \end{cases} \quad \text{con} \quad k = n-1,\ n-2,\ \ldots,1$$

Esempi

1) $$\begin{cases} 2x_1 - 1x_2 + 1x_3 - 2x_4 = 0 \\ 2x_2 + 0x_3 - 1x_4 = 1 \\ -\frac{5}{2}x_3 + \frac{9}{4}x_4 = -\frac{1}{4} \\ \frac{29}{10}x_4 = \frac{29}{10} \end{cases}$$ da cui si ricavano:

r4: $\frac{29}{10}x_4 = \frac{29}{10} \rightarrow x_4 = 1$

r3: $-\frac{5}{2}x_3 + \frac{9}{4} = -\frac{1}{4} \rightarrow x_3 = 1$

r2: $2x_2 + 0 - 1 = 1 \rightarrow x_2 = 1$

r1: $2x_1 - 1 + 1 - 2 = 0 \rightarrow x_1 = 1$

In modo analogo se il sistema di equazioni lineari ha *matrice dei coefficienti triangolare inferiore*, a partire dalla prima equazione si risolve il sistema con tecnica di *forward-sostitution*: dalla prima equazione si ricava il valore dell'incognita x_1 e, per sostituzione progressiva nelle equazioni via via successive, si ricavano tutte le altre x_i.

Il costo per risolvere un sistema triangolare (inferiore o superiore) è pari a circa $n^2/2$ operazioni.

METODO DI GAUSS

L'idea di base consiste nel trasformare, con un numero finito di combinazioni lineari ed eventuali permutazioni di equazioni (cioè scambi di righe), il sistema iniziale denso

$$\mathbf{Ax} = \mathbf{b} \quad : \quad \begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots\ldots + a_{1n}x_n = b_1 \\ \ldots\ldots\ldots\ldots\ldots \\ a_{i1}x_1 + a_{i2}x_2 + a_{i3}x_3 + \ldots\ldots + a_{in}x_n = b_i \\ \ldots\ldots\ldots\ldots\ldots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \ldots\ldots + a_{nn}x_n = b_n \end{cases}$$

in un sistema equivalente $\mathbf{Ux} = \tilde{\mathbf{b}}$ con matrice dei coefficienti **U** triangolare superiore, che si risolve mediante tecnica di back-sostitution.

Procedura

1) Si effettua l'eliminazione delle variabili in ($n-1$) passi.
   Al generico passo k-esimo ($k = 1, \ldots, n-1$):
   a) si individua la riga pivot (cioè la riga il cui elemento sulla diagonale principale di **A** è $a_{ii} \neq 0$) e la colonna da annullare (operazione di *pivoting*)
   b) si calcola il moltiplicatore m_{ik} come $m_{ik} = -a_{ik}^{(k)}/a_{kk}^{(k)} \quad i=k+1, \ldots, n$, da applicare agli elementi della colonna da annullare che seguono quello in esame
   c) si trasformano gli elementi $a_{ij}^{(k)}$ e $b_j^{(k)}$ della matrice dei coefficienti e del vettore dei termini noti (con i e $j > k$, cioè appartenenti alle righe sottostanti quella in esame) come

$$\begin{cases} a_{ij}^{(k+1)} = a_{ij}^{(k)} + m_{ik} \cdot a_{kj}^{(k)} & i=k+1, \ldots, n \\ b_i^{(k+1)} = b_i^{(k)} + m_{ik} \cdot b_k^{(k)} & j=k+1, \ldots, n \end{cases}$$

   L'elemento $a_{kk}^{(k)}$ è il pivot.

   Il costo dell'eliminazione è pari a $n^3/3$ operazioni.

2) Si risolve il sistema triangolare superiore equivalente mediante back-sostitution.
   Il costo di soluzione ($n^2/2$ operazioni circa) è molto inferiore al costo dell'eliminazione e quindi il costo complessivo del metodo di Gauss è circa pari a $n^3/3$ operazioni (conta soltanto il processo di eliminazione delle variabili).

OSSERVAZIONI SUL PIVOTING

1) La scelta dell'elemento pivot viene solitamente effettuata con tecnica di *pivoting parziale*.
   Si sceglie come pivot l'elemento di modulo massimo della colonna da annullare in modo da avere tutti i moltiplicatori minori di 1 in modulo.
   Si sceglie cioè r pari al più piccolo intero maggiore o uguale di k e tale che $\left|a_{rk}^{(k)}\right| = \max_{k \le i \le n} \left|a_{ik}^{(k)}\right|$ e si scambia la riga k-esima con la riga r-esima.

2) Se il pivot $a_{kk}^{(k)}$ è nullo, il metodo di Gauss si blocca perché è impossibile calcolare i moltiplicatori m_{ik} (divisione per 0). Si deve quindi scegliere l'elemento pivot non nullo scambiando di posto due equazioni, p.e. la k-esima con una delle successive. Questo è possibile perché il sistema è non singolare e quindi se $a_{kk}^{(k)}=0$ necessariamente qualche altro elemento della colonna k-esima è $\neq 0$.
3) Si migliora la stabilità del metodo permutando le righe anche se l'elemento pivot è piccolo in modulo rispetto agli altri elementi (un pivot piccolo potrebbe derivare dalla differenza di due numeri quasi uguali e quindi essere affetto da cancellazione numerica).
4) L'eliminazione delle variabili non necessita di permutazioni di equazioni nel caso di matrici a diagonale dominante o simmetriche definite positive, purché gli elementi pivot $a_{kk}^{(k)}$ siano tutti non nulli.

DECOMPOSIZIONE GA=U

Il metodo di Gauss può essere interpretato come una successione finita di trasformazioni di **A** e **b**, cioè come moltiplicazione di **A** e **b** per un numero finito di matrici opportune.

Questo tipo di interpretazione consente di riformulare il metodo in due parti distinte:
- la prima (crf. Algoritmo: Factor) determina una matrice non singolare **G** tale che **GA**=**U** è una matrice triangolare superiore,
- la seconda (crf. Algoritmo: Solve) utilizza la matrice **G** e risolve il sistema **Ax**=**b**.

Il sistema lineare iniziale **Ax**=**b** può essere riscritto come:

$$\mathbf{GAx}=\mathbf{Gb} \quad \text{Þ} \quad \mathbf{Ux}=\tilde{\mathbf{b}}$$

con:

$$\mathbf{G}=\mathbf{M_{n-1}}\cdot\mathbf{P_{n-1}}\cdot\mathbf{.....}\cdot\mathbf{M_2}\cdot\mathbf{P_2}\cdot\mathbf{M_1}\cdot\mathbf{P_1}$$

Scambi di righe - Lo scambio di due equazioni del sistema **Ax**=**b**, p.e. la riga i-esima con la riga j-esima, equivale a moltiplicare (da sinistra) entrambi i membri del sistema per la matrice **P** che è una matrice identità con le righe i e j scambiate.

Combinazioni lineari - La sostituzione nel sistema della riga i-esima con la medesima più la riga j-esima moltiplicata per m_{ij}, equivale a moltiplicare (da sinistra) il sistema per la matrice **M** che ha diagonale principale unitaria e in posizione (i, j) il moltiplicatore m_{ij}.

FATTORIZZAZIONE PA=LU

La decomposizione **GA**=**U** può essere riformulata ipotizzando di effettuare prima tutti gli scambi di righe e successivamente tutte le combinazioni lineari.
Le matrici che contengono i moltiplicatori m_{ij} saranno ovviamente ordinate in modo diverso perchè gli scambi di righe avranno agito anche su di esse; quindi invece di avere:

$$\mathbf{GA}=\left(\mathbf{M_{n-1}}\cdot\mathbf{P_{n-1}}\cdot\mathbf{.....}\cdot\mathbf{M_2}\cdot\mathbf{P_2}\cdot\mathbf{M_1}\cdot\mathbf{P_1}\right)\mathbf{A}=\mathbf{U}$$

ora si ha:

$$\mathbf{GA} = \left(\mathbf{M}^{\circ}_{\mathbf{n-1}} \cdot \ldots\ldots \cdot \mathbf{M}^{\circ}_{\mathbf{2}} \cdot \mathbf{M}^{\circ}_{\mathbf{1}}\right)\left(\mathbf{P_{n-1}} \cdot \ldots\ldots \cdot \mathbf{P_2} \cdot \mathbf{P_1}\right)\mathbf{A} = \mathbf{U} \quad \text{Þ} \quad \mathbf{GA} = \mathbf{M}^{\circ}\mathbf{PA} = \mathbf{U}$$

dove $\mathbf{M}^{\circ}$ è la matrice triangolare inferiore, a diagonale unitaria, dei moltiplicatori con le righe riordinate dal vettore **pivot**.

In particolare si può scrivere:

$$\mathbf{PA} = \mathbf{M}^{\circ -1}\mathbf{U} \quad \text{cioè} \quad \mathbf{PA}=\mathbf{LU}$$

dove $\mathbf{L} = \mathbf{M}^{\circ -1}$ è una matrice triangolare inferiore con diagonale unitaria ed elementi fuori diagonale pari ai moltiplicatori m_{ij} cambiati di segno e riordinati come detto dal vettore **pivot**.

L'algoritmo Factor fornisce direttamente le matrici **L** e **U** se si effettuano le seguenti modifiche ai passi:

6. $a_{kj} \leftrightarrow a_{ij}$ $j=1, ..., n$ (scambio di righe anche nella parte inferiore di **A** che via via contiene i moltiplicatori)
9. $a_{ik} = -m_{ik} = \dfrac{a_{ik}}{a_{kk}}$ (si memorizza il moltiplicatore cambiato di segno)
10. $a_{ij} = a_{ij} - a_{ik} \times a_{kj}$, $j=k+1, ..., n$ (cambia il segno della combinazione lineare)

Nota la fattorizzazione **PA**=**LU**, per ricavare la soluzione del sistema iniziale **Ax**=**b** è sufficiente risolvere in cascata i sistemi triangolari:

$$\begin{cases} \mathbf{Ux} = \mathbf{y} \\ \mathbf{Ly} = \mathbf{Pb} = \mathbf{b}^{\circ} \end{cases}$$

Infatti da **Ax**=**b** si ricava **PAx**=**Pb** (moltiplicando ambo i membri per **P**), cioè **LUx**=**Pb** (perchè **PA**=**LU**) da cui $\mathbf{Ux} = \mathbf{y}$ e $\mathbf{Ly} = \mathbf{b}^{\circ}$.

UTILITÀ DELLA FATTORIZZAZIONE

La fattorizzazione **PA**=**LU**, come anche la decomposizione **GA=U**, possono essere utilizzate per:

1. Calcolare la matrice inversa $\mathbf{A}^{-1}$

$$\mathbf{PA}=\mathbf{LU} \quad \text{Þ} \quad (\mathbf{PA})^{-1} = (\mathbf{LU})^{-1} = \mathbf{U}^{-1}\mathbf{L}^{-1} \quad \text{Þ} \quad \mathbf{A}^{-1}\mathbf{P}^{-1} = \mathbf{U}^{-1}\mathbf{L}^{-1}$$

Dato che $\mathbf{P}^{-1} = \mathbf{P}$ (ci sono soltanto 0 e 1) si ha

$$\mathbf{A}^{-1} = (\mathbf{U}^{-1}\mathbf{L}^{-1})\mathbf{P} \quad \text{Þ} \quad \mathbf{A}^{-1} = \mathbf{BP}$$

Fare il prodotto **PA** significa eseguire su **A** gli scambi di righe riportati nel vettore **pivot**; analogamente il prodotto **BP** corrisponde a eseguire gli stessi scambi sulle colonne.

2. Risolvere *p* sistemi lineari multipli $\begin{cases} \mathbf{A}\mathbf{x}_1 = \mathbf{b}_1 \\ \mathbf{A}\mathbf{x}_2 = \mathbf{b}_2 \\ \ldots\ldots = \ldots \\ \mathbf{A}\mathbf{x}_p = \mathbf{b}_p \end{cases}$

la cui soluzione mediante decomposizione **GA=U** e soluzione dei sistemi triangolari costerebbe $p\left(n^3/3 + n^2/2\right)$ operazioni.

La fattorizzazione **PA**=**LU** (costo $n^3/3$) viene calcolata una sola volta perché le matrici **L**, **U** e **P** sono le stesse per tutti i *p* sistemi lineari.
Si ottengono quindi i seguenti sistemi triangolari:

$$\begin{cases} \begin{cases} \mathbf{L}\mathbf{y}_1 = \mathbf{b}_1^* & \text{costo } n^2/2 \\ \mathbf{U}\mathbf{x}_1 = \mathbf{y}_1 & \text{costo } n^2/2 \end{cases} \\ \ldots\ldots\ldots \\ \begin{cases} \mathbf{L}\mathbf{y}_p = \mathbf{b}_p^* \\ \mathbf{U}\mathbf{x}_p = \mathbf{y}_p \end{cases} \end{cases} \qquad \text{il cui costo di soluzione è } p\left(n^2/2 + n^2/2\right) = pn^2$$

Il costo complessivo (fattorizzazione+soluzione dei sistemi triangolari) è quindi pari a $n^3/3 + pn^2$ ed è decisamente inferiore al costo da sostenere con la decomposizione **GA=U**.

METODO DI CHOLESKI

Se la matrice **A** è a diagonale dominante o simmetrica definita positiva, il metodo di Gauss procede senza necessità di effettuare scambi di righe. La conseguente fattorizzazione **A**=**LU** può essere interpretata come prodotto di particolari matrici triangolari, una inferiore e l'altra superiore, $\mathbf{A}=\mathbf{L}_1\mathbf{L}_1^T$.

La fattorizzazione **A**=**LU** può infatti essere riscritta come:

$$\mathbf{A}=\mathbf{L}\mathbf{D}\mathbf{U}_1$$

dove **D**=diag(**U**), **L** e $\mathbf{U}_1$ sono triangolari (superiore e inferiore) con diagonale unitaria.

Poichè la matrice **A** è simmetrica allora $\mathbf{U}_1=\mathbf{L}^T$ e quindi $\mathbf{A}=\mathbf{L}\mathbf{D}\mathbf{L}^T$.
La matrice **A** è anche definita positiva e quindi gli elementi diagonale $(\mathbf{D})_{ii}$ sono tutti positivi; si può introdurre $\sqrt{\mathbf{D}}$ scrivendo $\mathbf{A}=\mathbf{L}\mathbf{D}^{1/2}\,\mathbf{D}^{1/2}\mathbf{L}^T$
Poiché la matrice **D** è diagonale allora $\mathbf{D}^T=\mathbf{D}$ e quindi $\mathbf{A}=(\mathbf{L}\mathbf{D}^{1/2})\,((\mathbf{D}^{1/2})^T\mathbf{L}^T)=\mathbf{L}_1\mathbf{L}_1^T$ con $\mathbf{L}_1=\mathbf{L}\mathbf{D}^{1/2}$.

$$\mathbf{L_1} = \begin{bmatrix} l_{11} & 0 & 0 & \cdot & \cdot & \cdot & 0 \\ l_{21} & l_{22} & 0 & \cdot & \cdot & \cdot & 0 \\ l_{31} & l_{32} & l_{33} & 0 & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ l_{j1} & l_{j2} & \cdot & \cdot & l_{ii} & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ l_{n1} & l_{n2} & l_{n3} & \cdot & \cdot & \cdot & l_{nn} \end{bmatrix}$$

Gli elementi l_{ij} si calcolano con la seguente formula ricorsiva:

$$l_{11} = \sqrt{a_{11}}$$

$$\begin{cases} l_{ij} = \dfrac{a_{ij} - \displaystyle\sum_{k=1}^{j-1} l_{ik} l_{jk}}{l_{jj}} \\ l_{ii} = \left(a_{ii} - \displaystyle\sum_{k=1}^{i-1} l_{ik}^2 \right)^{1/2} \end{cases} \qquad i = 2,\ldots,n \quad j = 1,\ldots,i-1$$

Il costo è pari a $n^3/6$, cioè la metà della fattorizzazione **LU** vera e propria.

CONDIZIONAMENTO DEI SISTEMI LINEARI

La sensitività della soluzione di un sistema lineare alle variazioni dei coefficienti della matrice **A** e dei termini noti del vettore **b** viene esaminata attraverso lo studio del condizionamento.

Sia nel caso di perturbazione del solo vettore dei termini noti **b** sia nel caso di perturbazione del vettore dei termini noti **b** e della matrice dei coefficienti **A**, si definisce *indice di condizionamento* del problema il numero:

$$K(\mathbf{A}) \equiv \|\mathbf{A}\| \cdot \|\mathbf{A}^{-1}\|$$

che rappresenta il fattore di amplificazione delle perturbazioni relative (cioè degli errori relativi) introdotte nel vettore **b** e nella matrice **A**.

- Se $K(\mathbf{A}) >> 1$ il sistema lineare $\mathbf{Ax}=\mathbf{b}$ è mal condizionato;
- $K(\mathbf{A})$ è tanto maggiore quanto più la matrice dei coefficienti **A** tende a essere singolare.

Tipici esempi di mal condizionamento sono la matrice di Hilbert $\mathbf{H_n}$ e la matrice di Vandermonde $\mathbf{V_n}$:

$$\mathbf{H_n}=\begin{bmatrix} 1 & 1/2 & \cdot & \cdot & 1/n \\ 1/2 & 1/3 & \cdot & \cdot & 1/(n+1) \\ 1/3 & 1/4 & \cdot & \cdot & 1/(n+2) \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 1/n & 1/(n+1) & \cdot & \cdot & 1/(2n-1) \end{bmatrix} \qquad \mathbf{V_n}=\begin{bmatrix} 1 & 1 & \cdot & \cdot & 1 \\ x_1 & x_2 & \cdot & \cdot & x_n \\ x_1^2 & x_2^2 & \cdot & \cdot & x_n^2 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ x_1^{n-1} & x_2^{n-1} & \cdot & \cdot & x_n^{n-1} \end{bmatrix}$$

METODI DI SOLUZIONE ITERATIVI

Questi metodi si applicano a matrici dei coefficienti **A** sparse e di ordine elevato (10^4, 10^6).
Operano esclusivamente con la matrice iniziale **A** e generano una successione infinita di vettori che converge alla soluzione (vettore delle incognite **x**).

Procedura

La matrice dei coefficienti **A** può essere pensata come somma di due matrici reali, quadrate di ordine *n*:

$$\mathbf{A}=\mathbf{C}+\mathbf{D}$$

In questo modo si ha:

$$\mathbf{Ax}=\mathbf{b} \quad \text{Þ} \quad (\mathbf{C}+\mathbf{D})\mathbf{x}=\mathbf{b} \quad \text{Þ} \quad \mathbf{Cx}+\mathbf{Dx}=\mathbf{b} \quad \text{Þ} \quad \mathbf{Dx}=\mathbf{b}-\mathbf{Cx} \quad \text{Þ} \quad \mathbf{Dx}=\mathbf{d}$$

con $\mathbf{b}, \mathbf{x} \in R^n$ e **d** funzione di **x**.

Il procedimento iterativo è costituito dai seguenti punti:

1. si assume quale vettore soluzione iniziale un vettore qualsiasi $\mathbf{x}^{(0)}$ (p.e. $\mathbf{x}^{(0)}=\mathbf{0}$ vettore nullo)
2. si costruisce il vettore $\mathbf{d}^{(0)}=\mathbf{b}-\mathbf{C}\mathbf{x}^{(0)}$
3. si risolve il sistema lineare $\mathbf{D}\mathbf{x}^{(1)}=\mathbf{d}^{(0)}$ ricavando il vettore $\mathbf{x}^{(1)}$
4. si costruisce il vettore $\mathbf{d}^{(1)}=\mathbf{b}-\mathbf{C}\mathbf{x}^{(1)}$
5. si risolve il sistema lineare $\mathbf{D}\mathbf{x}^{(2)}=\mathbf{d}^{(1)}$ ricavando il vettore $\mathbf{x}^{(2)}$
6. …. e così via
7. si considera raggiunta la convergenza quando la differenza tra due vettori soluzione successivi è minore di un prefissato valore di soglia

Assunto il vettore iniziale $\mathbf{x}^{(0)}$, il procedimento iterativo viene quindi espresso dalle formule:

$$\begin{cases} \mathbf{d}^{(k)} = \mathbf{b}-\mathbf{C}\mathbf{x}^{(k)} & \text{con } k = 0, 1, 2, \ldots \text{, numero di iterazioni} \\ \mathbf{D}\,\mathbf{x}^{(k+1)} = \mathbf{d}^{(k)} \end{cases}$$

Per calcolare il vettore soluzione $\mathbf{x}^{(k+1)}$ si deve quindi risolvere il sistema lineare $\mathbf{D}\,\mathbf{x}^{(k+1)} = \mathbf{d}^{(k)}$.
La matrice dei coefficienti **D** deve perciò essere non singolare ed è conveniente se ha struttura idonea da consentire calcoli rapidi e agevoli.
E' inoltre necessario che la successione dei vettori soluzione converga alla soluzione **x**: la matrice **D** deve essere scelte in modo da garantire tale convergenza (la matrice **C**=**A**–**D** viene di conseguenza).

CONVERGENZA

Un metodo iterativo è convergente se l'errore assoluto $\mathbf{e}^{(k)} = \mathbf{x} - \mathbf{x}^{(k)}$ tende a zero all'aumentare del numero k di iterazioni.
La scelta della matrice **D** (e quindi della matrice **C**) che garantisce la convergenza viene fatta proprio ragionando sull'errore assoluto.

Dalla scrittura $\mathbf{D}\,\mathbf{x}^{(k+1)} = \mathbf{d}^{(k)} = \mathbf{b} - \mathbf{C}\,\mathbf{x}^{(k)}$ si ricava $\mathbf{x}^{(k+1)} = \mathbf{d}^{(k)} = \mathbf{D}^{-1}\left(\mathbf{b} - \mathbf{C}\,\mathbf{x}^{(k)}\right)$ e quindi:

$$\mathbf{e}^{(k+1)} = \mathbf{x} - \mathbf{x}^{(k+1)} = \mathbf{D}^{-1}\left(\mathbf{b} - \mathbf{C}\,\mathbf{x}\right) - \mathbf{D}^{-1}\left(\mathbf{b} - \mathbf{C}\,\mathbf{x}^{(k)}\right) = -\mathbf{D}^{-1}\mathbf{C}\left(\mathbf{x} - \mathbf{x}^{(k)}\right) = -\mathbf{D}^{-1}\mathbf{C}\mathbf{e}^{(k)} = \mathbf{B}\mathbf{e}^{(k)} =$$
$$= \mathbf{B}\mathbf{B}\mathbf{e}^{(k-1)} = \ldots = \mathbf{B}\mathbf{B}^{k}\mathbf{e}^{(0)} = \mathbf{B}^{k+1}\mathbf{e}^{(0)}$$

dove $\mathbf{e}^{(0)}$ è l'errore assoluto iniziale e $\mathbf{B}=-\mathbf{D}^{-1}\mathbf{C}$ è la matrice di iterazione che, a ogni iterazione, moltiplica l'errore.

La matrice di iterazione $\mathbf{B}=-\mathbf{D}^{-1}\mathbf{C}$ rimane costante nel corso di tutta la soluzione perché dipende dalla matrice dei coefficienti iniziale **A** che a sua volta non viene modificata dai metodi iterativi.

Il processo iterativo di soluzione è certamente convergente se:

$$\|\mathbf{I}-\mathbf{D}^{-1}\mathbf{A}\|<1$$

La matrice **D** scelta deve rispondere ai seguenti requisiti:

1. **D** non singolare, cioè $\det(\mathbf{D})\neq 0$
2. l'insieme delle matrici **A** per cui $\|\mathbf{I}-\mathbf{D}^{-1}\mathbf{A}\|<1$ non deve essere vuoto (norma naturale, 1, ∞)
3. **D** diagonale o triangolare (calcoli rapidi)

A seconda della scelta della matrice **D** Ii metodi iterativi di soluzione dei sistemi di equazioni lineari si particolarizzano in quelli illustrati nel seguito.

METODO DI JACOBI

Scelto un vettore iniziale $\mathbf{x}^{(0)}$ e ordinata la matrice dei coefficienti **A** in modo che tutti gli elementi a_{ii} siano non nulli (con relativo ordinamento del vettore noto **b**), si costruisce la successione dei vettori approssimazione $\mathbf{x}^{(k+1)}$, di componenti $\left(x_1^{(k+1)}, x_2^{(k+1)}, \ldots, x_n^{(k+1)}\right)$, mediante la seguente formula iterativa:

$$x_i^{(k+1)} = \frac{b_i - \sum_{j=1,\, j\neq i}^{n} a_{ij} x_j^{(k)}}{a_{ii}} \qquad i = 1, \ldots, n$$

dove **tutte** le componenti del nuovo vettore approssimazione $\mathbf{x}^{(k+1)}$ dipendono dall'approssimazione precedente $\mathbf{x}^{(k)}$.

Il metodo di Jacobi corrisponde a scegliere una matrice **D** diagonale e pari alla diagonale principale della matrice dei coefficienti **A** (**D**=diag(**A**)) con $a_{ii} \neq 0$, eventualmente avendo effettuato un riordinamento delle righe.

METODO DI GAUSS-SEIDEL

A differenza del metodo precedente di Jacobi, le componenti del nuovo vettore approssimazione $\mathbf{x}^{(k+1)}$ man mano calcolate, sono subito utilizzate per determinare le componenti successive; la formula iterativa risulta quindi:

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

Il metodo di Gauss-Seidel corrisponde a scegliere una matrice **D** triangolare inferiore e pari alla parte triangolare inferiore della matrice dei coefficienti **A** (**D**=triang inf.(**A**)) con $a_{ii} \neq 0$, eventualmente avendo effettuato un riordinamento delle righe.

La convergenza del metodo di Gauss-Seidel non implica necessariamente la convergenza del metodo di Jacobi e viceversa.

Quando entrambi i metodi convergono, il metodo di Gauss-Seidel è più veloce di Jacobi.

METODO SOR

Il metodo SOR (di "sovra" o "sotto-rilassamento") è un metodo di Gauss-Seidel "accelerato" mediante un parametro di accelerazione ω che si introduce nello sdoppiamento della matrice **A**=**C**+**D** in modo che la matrice di iterazione $\mathbf{B}=-\mathbf{D}^{-1}\mathbf{C}=\mathbf{I}-\mathbf{D}^{-1}\mathbf{A}$ venga a dipendere da ω (se ω=1 allora SOR coincide con Gauss-Seidel).
Al parametro ω si assegna valore tale da massimizzare la velocità di convergenza del metodo.

Dalla formula iterativa di Gauss-Seidel, sommando e sottraendo $x_i^{(k)}$ a secondo membro, si ricava:

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

cioè:

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

Pensando il vettore approssimazione $\mathbf{x}^{(k+1)}$ dato dalla somma dell'approssimazione precedente $\mathbf{x}^{(k)}$ e di un vettore correzione $\mathbf{r}^{(k)}$:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{r}^{(k)}$$

e pesando il vettore correzione $\mathbf{r}^{(k)}$ con il parametro ω si ricava la formula iterativa del metodo SOR:

$$x_i^{(k+1)} = x_i^{(k)} + \omega \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i}^{n} a_{ij} x_j^{(k)}}{a_{ii}} = x_i^{(k)} + \omega \cdot r_i^{(k)} \qquad i = 1, \dots, n$$

In termini di matrici si può riscrivere in forma più compatta pensando la matrice dei coefficienti **A** come somma di tre matrici (**A**=**D**+**L**+**U**) con **D**=diag(**A**), **L**=triang inf.(**A**) priva della diagonale principale e **U**= triang sup.(**A**) anch'essa priva della diagonale principale. La matrice di iterazione del metodo SOR è funzione del parametro ω:

Si dimostra che se $\omega \leq 0$ oppure $\omega \geq 2$ il metodo SOR non converge.
Se la matrice dei coeffienti **A** è simmetrica definita positiva il metodo SOR converge per qualsiasi ω compreso tra 0 e 2; se ω è compreso tra 0 e 1 il metodo è detto sotto-rilassato, se ω è compreso tra 1 e 2 il metodo è detto sovra-rilassato.

ALGORITMI

Algoritmo: Backsost (*n*, **U, b, x**)
Commento. Risolve il sistema triangolare superiore **Ux**=**b** mediante tecnica di back-sostitution.
Parametri. Input: *n*, **A**, **b**
Output: **x**

1. $x_n = b_n / a_{nn}$
2. Ciclo 1: $i = n-1, \dots, 1$ (step −1)
3. $x_i = b_i$
4. Ciclo 2: $j = i+1, \dots, n$
5. $x_i = x_i - u_{ij} x_j$
6. Fine Ciclo 2
7. $x_i = x_i / u_{ii}$
8. Fine Ciclo 1
9. Exit

End

Algoritmo: Factor (*n*, **A**, **pivot**, det, ier)
Commento. L'algoritmo per determina la decomposizione **GA**=**U** di una matrice **A** di ordine *n*; viene utilizzato il pivoting parziale. La matrice triangolare superiore **U** viene memorizzata nella parte superiore di **A** mentre i moltiplicatori m_{ij} ($i>j$) sono memorizzati nelle corrispondenti posizioni di **A**.
Il vettore **pivot**, di dimensioni $n-1$, contiene tutti gli scambi di riga effettuati durante il processo di Gauss. Se la riga *k*-esima non viene rimossa pivot(*k*)=*k*; se invece allo stadio *k*-esimo la riga *k*-esima viene scambiata con la riga i-esima, pivot(*k*)=*i*.
La variabile det contiene il valore det(**A**).

La variabile ier è un indicatore di errore. Se ier=0 il processo di Gauss è stato portato a termine e in **A** si trovano le matrici **G** e **U**; se ier=1 la matrice **A** è singolare.

Parametri. Input: *n*, **A**
Output: **A**, **pivot**, det, ier

1. det = 1
2. Ciclo 1: k=1, ..., n–1
3. $a_{max}=\max_{k\le i\le n}|a_{ik}|$; sia i_0 il più piccolo indice $i \ge k$ tale che $|a_{i0\,k}|=amax;$ poni pivot(k)=i_0
4. se a_{max}=0 poni det=0, ier=1; Exit
5. se $i_0=k$ vai al punto 8
6. $a_{kj} \leftrightarrow a_{i0\,j}$ $j=k, ..., n$ (scambio di righe nella parte superiore di **A**: pedici dei termini scambiati)
7. det= –det
8. Ciclo 2: $i=k+1, ..., n$
9. $a_{ik}=m_{ik}=-\dfrac{a_{ik}}{a_{kk}}$ (si memorizza il moltiplicatore)
10. $a_{ij}=a_{ij}+a_{ik}\cdot a_{kj}$, $j=k+1, ..., n$ (combinazione lineare con eliminazione delle variabili)
11. Fine Ciclo 2
12. det=det$\cdot a_{kk}$
13. Fine Ciclo 1
14. se a_{nn}=0 poni det=0, ier=1; Exit
15. det=det$\cdot a_{nn}$
16. ier=0
17. Exit

End

Algoritmo: Solve (*n*, **A**, **pivot**, **b**)

Commento. L'algoritmo risolve il sistema non singolare, di ordine *n*, $\mathbf{Ux} = \tilde{\mathbf{b}}$ **;** in particolare si ha $\tilde{\mathbf{b}} = \mathbf{M}_{n-1}\mathbf{P}_{n-1}...\mathbf{M}_2\mathbf{P}_2\mathbf{M}_1\mathbf{P}_1\mathbf{b}$, $(\mathbf{U})_{ij}=(\mathbf{A})_{ij}$, $i\le j$, e $m_{ij}=(\mathbf{A})_{ij}$, $i>j$. La matrice input **A** è stata ottenuta dall'algoritmo Factor. Il vettore **pivot** contiene gli scambi di riga effettuati da Factor.
Al termine il vettore **b** contiene la soluzione **x**.

Parametri. Input: *n*, A, **pivot**, **b**
Output: **b**

1. Ciclo 1: k=1, ..., n–1
2. j=pivot(k)
3. se $j\neq k$, $b_j \leftrightarrow b_k$
4. Ciclo 2: $i=k+1, ..., n$
5. $b_i=b_i+a_{ik}\,b_k$
6. Fine Ciclo 2
7. Fine Ciclo 1
8. $b_n=b_n / a_{nn}$
9. Ciclo 3: $i=n-1, ..., 1$

10. $b_i = \left(b_i - \sum_{l=i+1}^{n} a_{il} b_l \right) / a_{ii}$
11. Fine Ciclo 3
12. Exit
End

Algoritmo: Gseidel (*n*, **A**, **b**, toll, *k*max, **x**, ier)

Commento. L'algoritmo, utilizzando il processo iterativo di Gauss-Seidel, migliora l'approssimazione iniziale $\mathbf{x}=\mathbf{x}^{(0)}$ della soluzione del sistema non singolare di ordine *n* **Ax=b**. Le successive approssimazioni vengono memorizzate nel vettore **x**. Se la precisione richiesta toll è raggiunta con un numero di iterazioni ≤ *k*max, si pone ier=0, altrimenti ier=1.

Parametri. Input: A, , **b**, toll, *k*max, **x**
Output: **x**, ier

1. Ciclo 1: $k=1, \ldots, k\max$
2. $y=x_1$
3. $x_1 = \left(b_1 - \sum_{j=2}^{n} a_{1j} x_j \right) \Big/ a_{11}$
4. ermax = $|y - x_1|$
5. Ciclo 2: $i=2, \ldots, n$
6. $y=x_i$
7. $x_i = \left(b_i - \sum_{j=1}^{i-1} a_{1j} x_j - \sum_{j=i+1}^{n} a_{ij} x_j \right) \Big/ a_{ii}$
8. er=$|y - x_i|$
9. se ermax < er , ermax=er
10. Fine Ciclo 2
11. se ermax < toll·$\|x\|_\infty$, ier=0; Exit
12. Fine Ciclo 1
13. ier=1
14. Exit
End

ESERCIZI SVOLTI

1. Risolvere il sistema lineare proposto mediante la decomposizione **GA**=**U** e con tecnica di pivoting parziale.

$$\begin{bmatrix} 2 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & -2 & 1 & 1 \\ 2 & 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \\ 0 \\ 4 \end{bmatrix}$$

Indicare inoltre le matrici **U** e **M** (matrice triangolare e matrice dei moltiplicatori) e i vettori $\tilde{\mathbf{b}}$ (termine noto trasformato) e *pivot* (contenente gli scambi di righe) e calcolare det(**A**).

passo 1:

$a_{11} \neq 0$ e di più grande degli altri elementi della prima colonna quindi la 1^ riga non si scambia (rimane al suo posto) e pivot(1)=1

$$\left[\begin{array}{cccc|c} 2 & 0 & 1 & 0 & 3 \\ 1 & 1 & 0 & 1 & 3 \\ 0 & -2 & 1 & 1 & 0 \\ 2 & 1 & 0 & 1 & 4 \end{array}\right]$$

si azzera la colonna1: $\begin{cases} 2^{\wedge} \leftarrow 2^{\wedge} - \frac{1}{2} 1^{\wedge} & m_{21} = -\frac{1}{2} \\ 4^{\wedge} \leftarrow 4^{\wedge} - 1 \cdot 1^{\wedge} & m_{41} = -1 \end{cases}$ ottenendo $\left[\begin{array}{cccc|c} 2 & 0 & 1 & 0 & 3 \\ 0 & 1 & -1/2 & 1 & 3/2 \\ 0 & -2 & 1 & 1 & 0 \\ 0 & 1 & -1 & 1 & 1 \end{array}\right]$

passo 2:

$a_{22} \neq 0$ ma è più piccolo (in modulo) di a_{32}: si effettua lo scambio 2^ riga $\leftarrow$ 3^ riga e quindi pivot(2)=3

$$\left[\begin{array}{cccc|c} 2 & 0 & 1 & 0 & 3 \\ 0 & -2 & 1 & 1 & 0 \\ 0 & 1 & -1/2 & 1 & 3/2 \\ 0 & 1 & -1 & 1 & 1 \end{array}\right]$$

si azzera la colonna2: $\begin{cases} 3^{\wedge} \leftarrow 3^{\wedge} + \frac{1}{2} \cdot 2^{\wedge} & m_{32} = \frac{1}{2} \\ 4^{\wedge} \leftarrow 4^{\wedge} + \frac{1}{2} \cdot 2^{\wedge} & m_{42} = \frac{1}{2} \end{cases}$ ottenendo $\left[\begin{array}{cccc|c} 2 & 0 & 1 & 0 & 3 \\ 0 & -2 & 1 & 1 & 0 \\ 0 & 0 & 0 & 3/2 & 3/2 \\ 0 & 0 & -1/2 & 3/2 & 1 \end{array}\right]$

passo 3:

$a_{33} = 0$: si effettua lo scambio 3^ riga $\leftarrow$ 4^ riga e quindi pivot(3)=4

$$\left[\begin{array}{cccc|c} 2 & 0 & 1 & 0 & 3 \\ 0 & -2 & 1 & 1 & 0 \\ 0 & 0 & -1/2 & 3/2 & 1 \\ 0 & 0 & 0 & 3/2 & 3/2 \end{array}\right]$$

la colonna3 è già azzerata: $m_{43} = 0$

Alla fine del processo di eliminazione delle variabili si è giunti a:

$$\mathbf{U}\,\mathbf{x} = \tilde{\mathbf{b}} \;\rightarrow\; \begin{bmatrix} 2 & 0 & 1 & 0 \\ 0 & -2 & 1 & 1 \\ 0 & 0 & -1/2 & 3/2 \\ 0 & 0 & 0 & 3/2 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \\ 1 \\ 3/2 \end{bmatrix} \qquad \textbf{pivot}=(1,3,4)^{\mathrm{T}}$$

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1/2 & 1 & 0 & 0 \\ 0 & 1/2 & 1 & 0 \\ -1 & 1/2 & 0 & 1 \end{bmatrix}$$

$$\det(\mathbf{A}) = (-1)^2 \cdot 2 \cdot (-2) \cdot \left(-\frac{1}{2}\right) \cdot \left(\frac{3}{2}\right) = 3$$

Risolvendo $\mathbf{Ux} = \tilde{\mathbf{b}}$ con back-sostitution:

$$\begin{cases} x_n = b_n / u_{nn} & n=4 \\ x_k = \dfrac{b_k - \sum_{j=k+1}^{n} u_{kj} x_j}{u_{kk}} & k = n-1,\ldots,1 = 3,2,1 \end{cases}$$

$$\begin{cases} 2x_1 + 0x_2 + 1x_3 + 0x_4 = 3 \\ -2x_2 + 1x_3 + 1x_4 = 0 \\ -\frac{1}{2}x_3 + \frac{3}{2}x_4 = 1 \\ \frac{3}{2}x_4 = \frac{3}{2} \end{cases} \quad \rightarrow \quad x_4=1,\ x_3=1,\ x_2=1,\ x_1=1 \quad \rightarrow \quad \mathbf{x}=(1,\ 1,\ 1,\ 1)^T$$

---

2. Ricavare la fattorizzazione **LU** della matrice **A** proposta nell'esercizio, noto il vettore **Pivot**.

$$\mathbf{A} = \begin{bmatrix} 2 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & -2 & 1 & 1 \\ 2 & 1 & 0 & 1 \end{bmatrix} \qquad \mathbf{Pivot} = \begin{pmatrix} 1 \\ 3 \\ 4 \end{pmatrix}$$

r_1 non scambia (passo $k = 1$)
$r_2 \leftarrow r_3$ (passo $k = 2$)
$r_3 \leftarrow r_4$ (passo $k = 3$)

Rispetto alla decomposizione **GA**=**U** è sufficiente applicare le seguenti modifiche:
- cambiare di segno i moltiplicatori m_{ij};
- effettuare gli scambi memorizzati nel pivot anche sulla parte inferiore di **A**.

passo 1:
Pivot(1)=1 quindi la riga r_1 non si scambia.
Si azzera la colonna 1 e si memorizza $-m_{21}=1/2$ e $-m_{41}=1$ nelle rispettive posizioni di **A**

$$\begin{bmatrix} 2 & 0 & 1 & 0 \\ 1/2 & 1 & -1/2 & 1 \\ 0 & -2 & 1 & 1 \\ 1 & 1 & -1 & 1 \end{bmatrix}$$

passo 2:
Pivot(2)=3 quindi si scambia 2^ riga ← 3^ riga

$$\begin{bmatrix} 2 & 0 & 1 & 0 \\ 0 & -2 & 1 & 1 \\ 1/2 & 1 & -1/2 & 1 \\ 1 & 1 & -1 & 1 \end{bmatrix}$$

Si azzera la colonna 2 e si memorizza $-m_{32}=-1/2$ e $-m_{42}=-1/2$ nelle rispettive posizioni di **A**

$$\begin{bmatrix} 2 & 0 & 1 & 0 \\ 0 & -2 & 1 & 1 \\ 1/2 & -1/2 & 0 & 3/2 \\ 1 & -1/2 & -1/2 & 3/2 \end{bmatrix}$$

passo 3:

Pivot(3)=4 quindi si scambia 3^ riga ← 4^ riga

$$\begin{bmatrix} 2 & 0 & 1 & 0 \\ 0 & -2 & 1 & 1 \\ 1 & -1/2 & -1/2 & 3/2 \\ 1/2 & -1/2 & 0 & 3/2 \end{bmatrix}$$

La colonna 3 è già azzerata; si memorizza $-m_{43}=0$ nella rispettiva posizione di **A**

Alla fine del processo di eliminazione delle variabili si è giunti a:

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & -1/2 & 1 & 0 \\ 1/2 & -1/2 & 0 & 1 \end{bmatrix} \qquad \mathbf{U} = \begin{bmatrix} 2 & 0 & 1 & 0 \\ 0 & -2 & 1 & 1 \\ 0 & 0 & -1/2 & 3/2 \\ 0 & 0 & 0 & 3/2 \end{bmatrix}$$

Si può verificare che **PA=LU**.

---

3. Sviluppare l'algoritmo di soluzione del sistema lineare **Ax**=**b**, con **A** tridiagonale (si può eliminare il pivoting) di ordine *n*, utilizzando esclusivamente vettori.

Algoritmo: Tridigsolu (*n*, **c, d, f, b, l, u, x**)

Commento. Calcola la soluzione di **Ax**=**b** con **A** tridiagonale. Utilizza soltanto i vettori **d**=diag., **c**=codiag. inf. e **f**=codiag. sup. di **A**. Ricava la fattorizzazione **LU** in termini di vettori **l**=codiag. inf. di **L**, **u**=diag. di **U** e **f**=codiag. sup. di **U** (coincide con la codiag. sup. di **A**). La soluzione è effettuata come cascata di sistemi triangolari.

Parametri. Input: *n*, **c, d, f, b**
Output: **x**

1. $u_1=d_1$
2. Ciclo 1: $i = 2,..., n$ (fattorizzazione **LU** di **A** tridiagonale)
3. $l_i= c_i/u_{i-1}$
4. $u_i= d_i-l_i f_{i-1}$

5. Fine Ciclo 1
6. $y_1=b_1$
7. Ciclo 2: $i = 2,..., n$ (soluzione sistema triangolare $\mathbf{L}y= b$)
8. $y_i=b_i-l_i y_{i-1}$
9. Fine Ciclo 2
10. $x_n=y_n/u_n$
11. Ciclo 3: $i = n-1,..., 1$ (soluzione sistema triangolare $\mathbf{U}x = y$)
12. $x_i=y_i-f_i x_{i+1}/u_i$
13. Fine Ciclo 3
14. Exit

End

ESERCIZI PROPOSTI

1. Sviluppare l'algoritmo per la soluzione di un sistema diagonale **Dx**=**b** di ordine n.

[Diagsolu]

2. Sviluppare l'algoritmo per la soluzione di un sistema triangolare inferiore **Lx**=**b** di ordine n.

[Forwsost]

3. Risolvere i seguenti sistemi lineari triangolari superiori con tecnica di Forward-sostitution.

$$\begin{bmatrix} 2 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 2 & 2 & 1 & 0 \\ 2 & 1 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \\ 0 \\ 4 \end{bmatrix}$$

[$\mathbf{x}=(1.5, 1.5, -6, -0.5)^T$]

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 3 & 2 & 0 & 0 \\ 4 & 5 & 3 & 0 \\ 2 & 1 & 2 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \\ 6 \\ 6 \end{bmatrix}$$

[$\mathbf{x}=(3, -3, 3, -3)^T$]

4. Ricavare la fattorizzazione **LU** della seguente matrice **A**, noto il relativo vettore **Pivot**.

$$\mathbf{A} = \begin{bmatrix} 4 & 7 & 9 & 2 \\ 5 & 4 & 6 & 1 \\ 3 & 3 & 4 & 1 \\ 5 & 7 & 2 & 8 \end{bmatrix} \quad \mathbf{Pivot} = \begin{pmatrix} 2 \\ 2 \\ 4 \end{pmatrix}$$

$$\left[L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 4/5 & 1 & 0 & 0 \\ 1 & 15/19 & 1 & 0 \\ 3/5 & 3/19 & 5/139 & 1 \end{bmatrix} \qquad U = \begin{bmatrix} 5 & 4 & 6 & 1 \\ 0 & 19/5 & 21/5 & 6/5 \\ 0 & 0 & -139/19 & 115/19 \\ 0 & 0 & 0 & -1/139 \end{bmatrix} \right]$$

5. Ricavare la fattorizzazione **LU** della seguente matrice **A**, noto il relativo vettore **Pivot**.

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 8 & 2 \\ 2 & 4 & 3 & 2 \\ 5 & 1 & 2 & 1 \\ 5 & 3 & 4 & 2 \end{bmatrix} \quad \mathbf{Pivot} = \begin{pmatrix} 3 \\ 2 \\ 3 \end{pmatrix}$$

$$\left[L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2/5 & 1 & 0 & 0 \\ 1/5 & 1/2 & 1 & 0 \\ 1 & 5/9 & 14/117 & 1 \end{bmatrix} \qquad U = \begin{bmatrix} 5 & 1 & 2 & 1 \\ 0 & 18/5 & 11/5 & 8/5 \\ 0 & 0 & 13/2 & 1 \\ 0 & 0 & 0 & -1/117 \end{bmatrix} \right]$$

6. Ricavare la fattorizzazione **LU** della seguente matrice **A**, noto il relativo vettore **Pivot**.

$$\mathbf{A} = \begin{bmatrix} 6 & 5 & 3 & 8 \\ 2 & 2 & 10 & 3 \\ 7 & 2 & 8 & 2 \\ 1 & 4 & 9 & 7 \end{bmatrix} \quad \mathbf{Pivot} = \begin{pmatrix} 3 \\ 4 \\ 3 \end{pmatrix}$$

$$\left[L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1/7 & 1 & 0 & 0 \\ 6/7 & 23/26 & 1 & 0 \\ 2/7 & 5/13 & -122/281 & 1 \end{bmatrix} \qquad U = \begin{bmatrix} 7 & 2 & 8 & 2 \\ 0 & 26/7 & 55/7 & 47/7 \\ 0 & 0 & -281/26 & 9/26 \\ 0 & 0 & 0 & -1/281 \end{bmatrix} \right]$$

AUTOVALORI DI MATRICI

Riferimento al testo: Cap. IV

INTRODUZIONE

Molti problemi dell'ingegneria sono descritti da sistemi di equazioni lineari con funzioni che dipendono, oltre che dalle incognite, anche da un parametro λ.
Il sistema ammette soluzione diversa dalla soluzione nulla per particolari valori λ_i detti autovalori e le corrispondenti soluzioni $\mathbf{x_i}$ sono dette autovettori

$$\begin{cases} f_1(x_1, \ldots, x_n;\ l) = 0 \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ f_n(x_1, \ldots, x_n;\ l) = 0 \end{cases}$$ in forma matriciale si ha $\mathbf{A}\times\mathbf{x}=\lambda\times\mathbf{x}$ cioè $(\mathbf{A}-\lambda\mathbf{I})\times\mathbf{x}=\mathbf{0}$

$\lambda \in R$ è autovalore della matrice $\mathbf{A} \in R^{n,n}$ se e solo se la matrice $[\mathbf{A}-\lambda\mathbf{I}]$ è singolare, cioè se e solo se $\det(\mathbf{A}-\lambda\mathbf{I})=|\ \mathbf{A}-\lambda\mathbf{I}\ |= 0$ (equazione caratteristica).

Se $\mathbf{A} \in R^{n,n}$ è simmetrica allora $\lambda \in R$ e i corrispondenti $\mathbf{x} \in R^n$ sono un sistema di vettori ortogonali.
Se $\mathbf{A}$ è simmetrica definita positiva allora i suoi $\lambda \in R$ e sono tutti positivi.
Se λ è autovalore di $\mathbf{A}$ allora $1/\lambda$ è autovalore di $\mathbf{A}^{-1}$.
$\mathbf{A}$ e $\mathbf{A}^T$ hanno gli stessi autovalori.

L'equazione caratteristica $\det(\mathbf{A}- l\,\mathbf{I})=0$ è un polinomio di grado n

$$\det(\mathbf{A}- l\,\mathbf{I})=0 \quad \text{cioè} \quad (-1)^n \lambda^n + \alpha_1 \lambda^{n-1} + \ldots.. + \alpha_{n-1}\lambda + \alpha_n = 0$$

La soluzione diretta dell'equazione caratteristica non viene effettuata in quanto si tratta spesso di problema mal condizionato.
Gli autovalori λ_i si calcolano utilizzando diversi metodi numerici e successivamente si risolve ciascun sistema lineare omogeneo associato $(\mathbf{A}-\lambda_i\mathbf{I})\cdot\mathbf{x_i}=\mathbf{0}$ per calcolare i corrispondenti autovettori $\mathbf{x_i}$.

METODO DELLE POTENZE

Serve per calcolare l'autovalore di modulo massimo o quello di modulo minimo della matrice$\mathbf{A} \in R^{n,n}$.

Autovalore di modulo massimo

Data una matrice $\mathbf{A}$ si vuole calcolare l'autovalore λ di modulo massimo (e il corrispondente autovettore $\mathbf{x}$). Vengono fatte due ipotesi.

Hp1: esiste un solo $\lambda_1 \in R^n$ di modulo massimo $\left(|l_1| > |l_2| \geq |l_3| \geq \ldots \geq |l_n|\right)$, cioè $|\lambda_i/\lambda_1| < 1$.

Hp2: $\mathbf{A}$ è diagonalizzabile (cioè $\mathbf{X}^{-1}\cdot\mathbf{A}\times\mathbf{X}=\mathrm{L}$ con $\mathrm{L}=\mathrm{diag}(\lambda_1, \ldots, \lambda_n)$) e quindi tutti gli autovettori $\mathbf{x_i}$ sono linearmente indipendenti tra loro.

Grazie all'*Hp2*, il generico vettore $\mathbf{v_0}$ può essere scritto come combinazione lineare degli autovettori.

$$\mathbf{v_0} = a_1\mathbf{x_1} + a_2\mathbf{x_2} + \ldots + a_n\mathbf{x_n} \quad \mathbf{v}, \mathbf{x_i} \in R^n$$

Nel metodo delle potenze si sceglie un vettore iniziale $\mathbf{v_0}$ qualsiasi (p.e. $\mathbf{v_0}=(1, 1, 1, \ldots, 1)^T$) e quindi si genera la seguente successione di vettori:

$$\mathbf{v_1} = \mathbf{A}\cdot\mathbf{v_0} = a_1\mathbf{A}\mathbf{x_1} + \ldots + a_n\mathbf{A}\mathbf{x_n} = a_1 l_1\mathbf{x_1} + \ldots + a_n l_n\mathbf{x_n} = l_1\left(a_1\mathbf{x_1} + \left(\frac{l_2}{l_1}\right)a_2\mathbf{x_2} + \ldots + \left(\frac{l_n}{l_1}\right)a_n\mathbf{x_n}\right)$$

$$\mathbf{v_2} = \mathbf{A}\cdot\mathbf{v_1} = \mathbf{A}\cdot\mathbf{A}\mathbf{v_0} = a_1 l_1^{\,2}\mathbf{x_1} + \ldots + a_n l_n^{\,2}\mathbf{x_n} = l_1^2\left(a_1\mathbf{x_1} + \left(\frac{l_2}{l_1}\right)^2 a_2\mathbf{x_2} + \ldots + \left(\frac{l_n}{l_1}\right)^2 a_n\mathbf{x_n}\right)$$

……

$$\mathbf{v_m} = \mathbf{A}\cdot\mathbf{v_{m-1}} = \mathbf{A}\cdot\ldots\cdot\mathbf{A}\mathbf{v_0} = a_1 l_1^{\,m}\mathbf{x_1} + \ldots + a_n l_n^{\,m}\mathbf{x_n} = l_1^m\left(a_1\mathbf{x_1} + \left(\frac{l_2}{l_1}\right)^m a_2\mathbf{x_2} + \ldots + \left(\frac{l_n}{l_1}\right)^m a_n\mathbf{x_n}\right)$$

Al passo *m* il vettore $\mathbf{v_m}$, per effetto della moltiplicazione progressiva per la matrice $\mathbf{A}$, tende a disporsi parallelamente all'autovettore $\mathbf{x_1}$. Inoltre, grazie all' *Hp1* ($|\lambda_i/\lambda_1| < 1$) si ha:

$$\lim_{m\to\infty}\left(\frac{l_i}{l_1}\right)^m = 0 \quad \text{quindi} \quad \lim_{m\to\infty}\frac{1}{l_1^m}\cdot\mathbf{v_m} = a_1\mathbf{x_1}.$$

L'autovalore di modulo massimo cercato si ottiene come rapporto tra le generiche componenti $k0$ dei vettori $\mathbf{v}_m$ e $\mathbf{v}_{m+1}$:

$$\left(\frac{(\mathbf{v_{m+1}})_{k0}}{(\mathbf{v_m})_{k0}}\right) = l_1\frac{a_1(\mathbf{x_1})_{k0} + \text{termini} \to 0}{a_1(\mathbf{x_1})_{k0} + \text{termini} \to 0} = l_1$$

Algoritmo. Pow(*n*, **A**, toll, *m*max, λ, **y**)

Commento. Determina l'autovalore di modulo massimo della matrice **A** e l'autovettore corrispondente. Se la precisione relativa richiesta toll viene raggiunta con un numero di iterazioni ≤ *m*max la variabile ier assume il valore 0, altrimenti ier=1.

Parametri. Input: *n*, **A**, toll, *m*max
Output: λ, **y**

1. $\mathbf{y_0}=(1, 1, \ldots, 1)^T$
2. Ciclo 1: $m=0, \ldots, m\text{max}$
3. $\mathbf{w_{m+1}}=\mathbf{A}\,\mathbf{y_m}$
4. $l_1^{(m+1)} = \dfrac{(\mathbf{w_{m+1}})_{k_0}}{(\mathbf{y_m})_{k_0}}$ (approssimazione dell'autovalore di modulo massimo cercato)
5. $\mathbf{y_{m+1}} = \dfrac{\mathbf{w_{m+1}}}{\|\mathbf{w_{m+1}}\|_\infty}$ (normalizzazione dell'autovettore per evitare overflow o underflow.)
6. $\text{er}=\left|l_1^{(m)} - l_1^{(m+1)}\right|$
7. se $\text{er}\le\text{toll}\cdot\left|l_1^{(m+1)}\right|$ oppure er≤toll allora poni ier=0 e vai al punto 10.
8. Fine Ciclo 1

9. ier=1
10. $\mathbf{y}=\mathbf{y}_{m+1}$
11. $\lambda = l_1^{(m+1)}$
12. Exit
    End

Osservazioni

– Se $\alpha_1=0$ e $|\lambda_2|>|\lambda_3|$, in teoria il metodo delle potenze dovrebbe convergere sull'autovalore λ_2. Però, causa degli inevitabili errori di round-off , dopo pochi passi si ha $\alpha_1\neq 0$ e il metodo ricade su λ_1.

– Se $|l_1| \cong |l_2|$ la convergenza può essere eccessivamente lenta allora si utilizza il metodo delle potenze per avere una stima iniziale p di λ_1 e poi si raffina con il metodo delle potenze inverse.

Autovalore di modulo minimo

Data una matrice **A** si vuole calcolare l'autovalore λ di modulo minimo (e il corrispondente autovettore **x**). Vengono nuovamente fatte 2 ipotesi.

Hp1: esiste un solo $\lambda_1 \in R^n$ di modulo minimo $\left(|\lambda_1| < |\lambda_2| \leq |\lambda_3| \leq \ldots \leq |\lambda_n|\right)$.

Hp2: Tutti gli autovettori $\mathbf{x_i}$ sono linearmente indipendenti tra loro.

Poiché se λ è autovalore di **A**, l'autovalore di $\mathbf{A}^{-1}$ è pari a $1/\lambda$ (crf. Pag. 7-1), per calcolare l'autovalore di modulo minimo della matrice **A** è sufficiente calcolare l'autovalore di modulo massimo della matrice inversa $\mathbf{A}^{-1}$ e quindi calcolarne il reciproco.

Lo schema di calcolo è il seguente

$$\mathbf{A}^{-1}\mathbf{x} = \frac{1}{\lambda}\,\mathbf{x} \quad \rightarrow \quad \mathbf{B}\mathbf{x} = \mu\mathbf{x}$$

$\downarrow$ metodo delle potenze per calcolare

μ_1 autovalore di modulo max di $\mathbf{B} = \mathbf{A}^{-1}$

$\downarrow$ quindi si calcola

$$\lambda_{\min} = \frac{1}{\mu_1}$$

Ovviamente anziché calcolare esplicitamente la matrice inversa $\mathbf{A}^{-1}$ (costo n^3) conviene effettuare la fattorizzazione LU (costo $n^3/3$) e risolvere i due sistemi triangolari risultanti (costo $n^2/2$ ciascuno).

Il metodo delle potenze viene portato avanti finché $\left|m_1^{(m+1)} - m_1^{(m)}\right| \leq e\left|m_1^{(m+1)}\right|$; a questo punto si assume $m_1^{(m+1)}$ quale migliore approssimazione cercata e se ne calcola il reciproco.

1. $\mathbf{y_0}=(1, 1, ..., 1)^T$
2. Fattorizzazione **PA**=**LU** (algoritmo Factor)
3. Ciclo 1: m=0, ..., $m$max
4. $\mathbf{LUw}_{m+1}=\mathbf{Py}_m \rightarrow \begin{cases} \mathbf{Lz}_{m+1} = \mathbf{Py}_m \\ \mathbf{Uw}_{m+1} = \mathbf{z}_{m+1} \end{cases}$

5. $\mu_1^{(m+1)} = (w_{m+1})_{k_0} / (y_m)_{k_0}$ (approssimazione di μ_1 autovalore di modulo massimo di $\mathbf{A}^{-1}$)
6. $y_{m+1} = w_{m+1} / \|w_{m+1}\|_\infty$ (normalizzazione autovettore)
7. Fine Ciclo 1
8. $l_n = \dfrac{1}{m_1}$ (autovalore di modulo minimo di **A** cercato)

METODO DELLE POTENZE INVERSE

E' una generalizzazione del metodo delle potenze e serve per calcolare un particolare autovalore l di cui si conosca una stima p.

Utilizzando la stima p, il sistema $\mathbf{Ax} = l\,\mathbf{x}$ può essere riscritto come

$$(\mathbf{A}-p\mathbf{I})\mathbf{x} = \mathbf{Ax} - p\mathbf{x} = l\,\mathbf{x} - p\mathbf{x} = (l-p)\mathbf{x}$$

cioè $(l-p)$ è autovalore della matrice $(\mathbf{A}-p\mathbf{I})$ e l'autovettore corrispondente è sempre **x**.

Se $(l-p)$ è autovalore della $(\mathbf{A}-p\mathbf{I})$ allora $1/(l-p)$ è autovalore della matrice inversa $(\mathbf{A}-p\mathbf{I})^{-1}$; inoltre se p è una buona stima di l (p è "vicino" a l) allora $1/(l-p)$ è l'autovalore di modulo massimo della matrice $(\mathbf{A}-p\mathbf{I})^{-1}$ che può essere calcolato con il metodo delle potenze.

Lo schema di calcolo è il seguente

$$(\mathbf{A} - p\mathbf{I})^{-1}\mathbf{x} = \frac{1}{l-p}\,\mathbf{x} \quad \rightarrow \quad \mathbf{Bx} = m\mathbf{x} \ \text{ con } m = \frac{1}{l-p}$$

$\downarrow$ metodo delle potenze per calcolare

m_1 autovalore di modulo max di $\mathbf{B} = (\mathbf{A} - p\mathbf{I})^{-1}$

$\downarrow$ quindi si calcola

$$l = \frac{1}{m_1} + p$$

Algoritmo. Invpow (*n*, **A**, toll, *m*max, *p*, **x**, ier)

Commento. Utilizza il metodo delle potenze inverse per determinare l'autovalore della matrice **A,** di ordine *n*, più vicino al numero *p* stimato e il corrispondente autovettore **x**. Se la precisione relativa richiesta toll viene raggiunta con un numero di iterazioni $\leq$ *m*max si pone ier=0, altrimenti ier=2. Se la matrice (**A**–*p***I**) è singolare, allora ier=1.

Parametri. Input: *n*, **A**, toll, *m*max, *p*
Output: *p*, **x**, ier

1. $(\mathbf{A})_{ii} = (\mathbf{A})_{ii} - p \ , \quad i = 1, \ldots, n$
2. richiama l'algoritmo Factor (*n*, **A**, **pivot**, det, ier)
3. se ier=1 Exit
4. $y_\circ = (1,1,\ldots 1)^T$
5. $\lambda_p^{(0)}=p$
6. Ciclo 1: $m=0, \ldots, m\max$
7. $\mathbf{U}^{\chi}\mathbf{w_{m+1}}=\mathbf{G}^{\chi}\mathbf{y_m} \Rightarrow \mathbf{w_{m+1}}$
8. $a=\left\|\mathbf{w}_{\mathbf{m+1}}\right\|_\infty$; sia k_0 la posizione della prima componente di modulo massimo di $\mathbf{w_{m+1}}$
9. $\mathbf{y_{m+1}} = \dfrac{\mathbf{w_{m+1}}}{a}$ (normalizzazione del vettore $\mathbf{w_{m+1}}$)
10. $\lambda_p^{(m+1)}=p+\dfrac{(\mathbf{y_m})_{k_\circ}}{(\mathbf{w_{m+1}})_{k_\circ}}$
11. $\mathrm{er}=\left| l_p^{(m)} - l_p^{(m+1)} \right|$
12. se er$\leq$toll$\cdot\left| l^{(m+1)}{}_p \right|$ oppure er$\leq$toll poni ier=0 e vai al punto 15
13. Fine Ciclo 1
14. ier=2
15. $\mathbf{x}=\mathbf{y_{m+1}}$
16. $p=\lambda_p^{(m+1)}$
17. Exit

End

TRASFORMAZIONI DI SIMILITUDINE

Sono necessarie alcune definizioni preliminari.

1. Due **vettori** sono **ortogonali** se e solo se il loro prodotto scalare è nullo ($\mathbf{x}^T\mathbf{y}=\sum_{i=1}^{n} x_i y_i =0$).
2. Un **sistema di vettori** è **ortonormale** se i vettori $\mathbf{a_1}, \ldots, \mathbf{a_n}$ sono ortogonali due a due.
3. Una **matrice** è **ortogonale** se e solo se le sue righe (colonne) formano un sistema di vettori ortonormale.
4. Se **A** è ortogonale si ha $\mathbf{A}^T\mathbf{A}=\mathbf{A}\mathbf{A}^T=\mathbf{I}$, $\mathbf{A}^{-1}=\mathbf{A}^T$; se **A** è ortogonale e simmetrica allora $\mathbf{A}^{-1}=\mathbf{A}^T=\mathbf{A}$.

Un **riflettore elementare U** è una matrice di ordine *n* non singolare e ortogonale ($\mathbf{U}^{-1}=\mathbf{U}^T$). Una **trasformazione di similitudine** è una trasformazione che associa a una matrice **A** la matrice simile $\mathbf{U}^{-1}\mathbf{A}\mathbf{U}=\mathbf{U}^T\mathbf{A}\mathbf{U}$. Essere *simili* significa che la matrice iniziale e quella trasformata hanno gli stessi autovalori λ e autovettori semplicemente trasformati.

METODO QR

Serve per calcolare tutti gli autovalori λ di una matrice **A**.
Poiché il calcolo di tutti gli autovalori risulta estremamente costoso se applicato a matrici dense, il metodo trasforma la matrice iniziale **A** in una matrice simile, di forma più semplice (tridiagonale se la **A** è simmetrica), di cui calcola autovalori e autovettori.
Data la matrice **A** si costruisce una matrice **R** triangolare superiore come prodotto di $(n-1)$ riflettori elementari $\mathbf{U}_i$

$$\mathbf{U}_{n-1}\mathbf{U}_{n-2}\ldots\mathbf{U}_1\mathbf{A}=\mathbf{R} \text{ che si scrive anche come } \mathbf{Q}^T\mathbf{A}=\mathbf{R}$$

dove $\mathbf{Q}^T=\mathbf{Q}^{-1}$, ortogonale in quanto prodotto di matrici ortogonali.

Quindi si può scrivere **A**=**QR** dove la matrice **R**, simile alla matrice iniziale **A**, è triangolare superiore e ha gli autovalori posizionati sulla sua diagonale principale.

Gli autovalori della matrice iniziale **A** sono gli elementi sulla diagonale principale della matrice simile $\mathbf{A}_\infty$ ottenuta dalla seguente successione di trasformazioni

1. $\mathbf{A}_1=\mathbf{A}$
2. Ciclo 1: $i=1, \ldots, i\max$
3. $\mathbf{A}_i=\mathbf{Q}_i\mathbf{R}_i$ (fattorizzazione QR)
4. $\mathbf{A}_{i+1}=\mathbf{R}_i\mathbf{Q}_i$
5. Fine Ciclo 1
6. $\lambda=\mathrm{diag}(\mathbf{A}_{i\max})$

Il processo iterativo viene fermato quando gli elementi fuori diagonale della $\mathbf{A}_i$, che dovrebbero convergere a zero dopo infiniti passi, sono sufficientemente piccoli. Per accelerare la convergenza si utilizzano parametri di accelerazione t_i opportunamente scelti e modificare il metodo nei passi

3. $(\mathbf{A}_i-t_i\cdot\mathbf{I})=\mathbf{Q}_i\mathbf{R}_i$ (fattorizzazione QR)
4. $\mathbf{A}_{i+1}=\mathbf{R}_i\mathbf{Q}_i+t_i\cdot\mathbf{I}$

Il costo della fattorizzazione QR è doppio rispetto alla fattorizzazione LU ($2n^3/3$ contro $n^3/3$ operazioni), però è applicabile anche se la matrice è singolare (al contrario della fattorizzazione LU) o rettangolare (si utilizza nel metodo dei minimi quadrati).

APPROSSIMAZIONE DI DATI E DI FUNZIONI

Riferimento al testo: Cap. V

INTRODUZIONE

- **Approssimazione di funzioni**: la $f(x)$ è nota analiticamente, ma risulta difficile o impossibile eseguire operazioni (p.e. integrazione) con strumenti dell'analisi matematica; si approssima la $f(x)$ con una $f_n(x)$ più semplice.

- **Approssimazione di dati**: la $f(x)$ non è nota analiticamente, ma si ha a disposizione una raccolta di dati di cui si conoscono i valori $\{y_i\}$ (ordinate) corrispondenti ai nodi $\{x_i\}$ (ascisse).
  Si costruisce un modello matematico $f_n(x)$ (cioè una funzione) che approssima in modo attendibile il valore della y in punti diversi dai nodi.

Per effettuare l'approssimazione di dati e funzioni è necessario:

1. individuare la classe delle funzioni approssimanti $F_n=\{f_n(x)\}$ in base alle caratteristiche del fenomeno
2. scegliere il particolare elemento $f_n(x)$ applicando un criterio di scelta.

CLASSI F_N DI FUNZIONI APPROSSIMANTI

- **Polinomi algebrici** di grado n

$P_n=\left\{ f_n(x) = a_0 + a_1 x + a_2 x^2 + ... + a_n x^n \right\}$. Si utilizzano per funzioni continue su intervalli chiusi e limitati. Richiedono la determinazione di $(n+1)$ parametri $a_0, ..., a_n$.

- **Polinomi trigonometrici** di grado n e pulsazione ω

  $T_n=\left\{ f_n(x) = a_0 + \sum_{k=1}^{n} (a_k \cos(k\omega x) + b_k \sin(k\omega x) \right\}$ Si utilizzano per funzioni periodiche.

- **Funzioni razionali** $R_{n,d}=\frac{P_n}{P_d}=\left\{ f_n(x) = \frac{a_0 + a_1 x + a_2 x^2 + ... + a_n x^n}{b_0 + b_1 x + b_2 x^2 + ... + x^d} \right\}$. Si utilizzano per funzioni aperiodiche su intervalli illimitati. Richiedono la determinazione di $(n+d+1)$ parametri (il denominatore viene normalizzato in modo che il coefficiente del termine x^d sia unitario).

- **Polinomiali a tratti** $F_{n,d}$ associate a sotto intervalli dell'intervallo di interesse. Ogni tratto è un polinomio di ordine basso (primo o secondo).

- **Spline** $S_{n,d}$ di grado d. E' un particolare sottoinsieme delle funzioni polinomiali a tratti che garantisce la continuità della funzione e di tutte le sue derivate di ordine $\leq (d-1)$ in tutto l'intervallo d'interesse.

- **Somme esponenziali** di ordine n $E_n=\left\{ f_n(x) = \sum_{k=1}^{n} a_k e^{-b_k x} \right\}$. Richiedono la determinazione di $2n$ parametri a_k, b_k.

CRITERI DI SCELTA DELLA FUNZIONE $f_N(x) \in F_N$

Il criterio di scelta garantisce l'individuazione della funzione approssimante $f_n(x)$ in modo univoco.
I criteri più comunemente utilizzati per l'individuazione della funzione $f_n(x)$ sono:

1. **Interpolazione di dati** - si applica nel caso di dati (punti) privi di errore e serve per selezionare la **funzione** che passa **per i punti** in esame
2. **Approssimazione (o Smoothing) di dati** - si applica nel caso di dati (punti) affetti da errore o dispersi e serve per selezionare la **funzione** che passa (al meglio possibile) **tra i punti** in esame.

INTERPOLAZIONE DI DATI

Riferimento al testo: Cap. V

CRITERIO DI INTERPOLAZIONE POLINOMIALE DI DATI

- Dati gli (n+1) punti noti $\{x_i, y_i\}$, con i=0, ..., n, si determina il polinomio interpolatore $P_n(x)$ di grado n

$$f_n(x)= P_n(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + ... + a_n \cdot x^n$$

imponendone il passaggio per i punti, cioè imponendo le seguenti (n+1) condizioni sulle ordinate

$$f_n(x_i)=y_i \text{ per } i=0, ..., n \qquad \text{con } f_n(x) \in P_n(x).$$

- Esiste un teorema (Weierstrass) che garantisce l'esistenza e unicità del polinomio interpolatore $\mathbb{P}_n(x)$ tale che

$$\|f(x)-P_n(x)\|<\varepsilon \text{ (con tolleranza } \varepsilon>0) \text{ per tutti gli } x \in [a,b] \text{ intervallo di interesse.}$$

- Se si impongono le condizioni indicate (passaggio della funzione per i punti) e si ricerca il polinomio come risoluzione del sistema di equazioni lineare in (n+1) incognite

$$\begin{cases} a_0 + a_1 x_0 + \ldots + a_n x_0^n = y_0 \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ a_0 + a_1 x_n + \ldots + a_n x_n^n = y_n \end{cases} \qquad \begin{pmatrix} 1 & x_0 & \ldots. & x_0^n \\ 1 & x_1 & \ldots. & x_1^n \\ \ldots. & \ldots. & \ldots. & \ldots. \\ 1 & x_n & \ldots. & x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \ldots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \ldots \\ y_n \end{pmatrix} \qquad \mathbf{V_n}^T \cdot \mathbf{a} = \mathbf{y}$$

si ricava una matrice dei coefficienti che é la $\mathbf{V_n}$ (matrice di Vandermonde).

- Se gli (n+1) nodi x_i sono distinti, cioè $x_i \neq x_j$, $\Rightarrow \det(\mathbf{V_n}) \neq 0 \Rightarrow$ la $\mathbf{V_n}$ è non singolare e quindi esiste un **unico polinomio interpolante** $P_n(x)$. Il sistema è però mal condizionato e quindi non è possibile risolverlo direttamente per determinare i coefficienti del polinomio.

- La determinazione dell'unico polinomio di interpolazione tra i punti dati si effettua mediante due tecniche numeriche: il metodo di Lagrange e il metodo di Newton.

METODO DI LAGRANGE: POLINOMI FONDAMENTALI DI LAGRANGE

- Il polinomio di interpolazione $P_n(x)$ si ricava dalla combinazione lineare di polinomi fondamentali

$$P_n(x)= \sum_{j=0}^{n} y_j \cdot l_j(x)$$

- Gli $l_j(x)$ sono detti polinomi fondamentali di Lagrange associati ai nodi, costituiscono una base dello spazio dei polinomi e si rappresentano come

$$l_j(x)=\frac{\prod_{i=0,\, i\neq j}^{n}(x-x_i)}{\prod_{i=0,\, i\neq j}^{n}(x_j-x_i)} \qquad \text{tali che} \qquad l_j(x_i)=\delta_{ij}=\begin{cases}1 & \text{se } i=j\\ 0 & \text{se } i\neq j\end{cases}$$

cioè i polinomi fondamentali di Lagrange sono nulli in tutti i nodi tranne che nel nodo j-esimo.

Esempio

1) Costruire il polinomio fondamentale di Lagrange $l_0(x)$.

$l_0(x)$ é nullo in tutti i nodi tranne che nel nodo 0, cioè $l_0(x_0)=1$ e $l_0(x_j)=0$ per $j=1, ...,n$.
$l_0(x)$ deve annullarsi in $x_1, x_2, ..., x_n$, quindi é necessaria una forma del tipo

$$(x-x_1)\cdot(x-x_2)\cdot\ldots\cdot(x-x_n)$$

che deve contemporaneamente essere unitaria in x_0, cioè

$$l_0(x)=\frac{(x-x_1)\cdot(x-x_2)\cdot\ldots\cdot(x-x_n)}{(x_0-x_1)\cdot(x_0-x_2)\cdot\ldots\cdot(x_0-x_n)}=\frac{\prod_{i=1}^{n}(x-x_i)}{\prod_{i=1}^{n}(x_0-x_i)}$$

Si spiega quindi la forma generale dei polinomi fondamentali di Lagrange $l_j(x)$.

- Il calcolo di ciascuno degli $n+1$ polinomi fondamentali costa $2n-2$ operazioni, quindi il costo totale è pari a $(n+1)(2n-2)=2n^2-2$ che risulta molto meno oneroso della soluzione del sistema di equazioni lineari (costo $n^3/3$). Inoltre il metodo di Lagrange è preferibile visto il mal condizionamento del sistema di equazioni per la presenza della matrice di Vandermonde.

Esempio

2) Dati i seguenti $n+1=3$ punti (0, 0) , (1, 1) , (2, 0) scrivere il polinomio di interpolazione $P_2(x)$ utilizzando il metodo di Lagrange.

Il polinomio si presenta nella forma

$$P_2(x)= y_o\frac{\prod_{i=1}^{2}(x-x_i)}{\prod_{i=1}^{2}(x_0-x_i)}+y_1\frac{\prod_{i=0,i\neq 1}^{2}(x-x_i)}{\prod_{i=0,i\neq 1}^{2}(x_1-x_i)}+y_2\frac{\prod_{i=0,i\neq 2}^{2}(x-x_i)}{\prod_{i=0,i\neq 2}^{2}(x_2-x_i)}=$$

$$=0+1\frac{(x-0)(x-2)}{(1-0)(1-2)}+0=1\frac{x(x-2)}{(-1)}=-x(x-2)=-x^2+2x$$

Poiché una sola ordinata è diversa da zero, l'applicazione del metodo è particolarmente rapida.

- Nel caso generale di ordinate "tutte" non nulle é sufficiente effettuare la combinazione lineare dei polinomi fondamentali $l_j(x)$ associati ai j nodi in esame per j=0, ..., n.

ESERCIZI SVOLTI

1. Calcolare il polinomio interpolatore per i 3 punti (0, 0) , (1, 1) , (2, 3) utilizzando il metodo di Lagrange.

Sono dati n+1=3 punti $\Rightarrow$ il polinomio cercato è di grado n=2.

$$P_2(x) = y_o \frac{\prod_{i=0,i\neq 0}^{2}(x-x_i)}{\prod_{i=0,i\neq 0}^{2}(x_0-x_i)} + y_1 \frac{\prod_{i=0,i\neq 1}^{2}(x-x_i)}{\prod_{i=0,i\neq 1}^{2}(x_1-x_i)} + y_2 \frac{\prod_{i=0,i\neq 2}^{2}(x-x_i)}{\prod_{i=0,i\neq 2}^{2}(x_2-x_i)} =$$

$$= 0 + 1\frac{\prod_{i=0,i\neq 1}^{2}(x-x_i)}{\prod_{i=0,i\neq 1}^{2}(1-x_i)} + 3\frac{\prod_{i=0,i\neq 2}^{2}(x-x_i)}{\prod_{i=0,i\neq 2}^{2}(2-x_i)} =$$

$$= 1\frac{(x-x_0)(x-x_2)}{(1-x_0)(1-x_2)} + 3\frac{(x-x_0)(x-x_1)}{(2-x_0)(2-x_1)} = 1\frac{(x-0)(x-2)}{(1-0)(1-2)} + 3\frac{(x-0)(x-1)}{(2-0)(2-1)} =$$

$$= 1\frac{x(x-2)}{(-1)} + 3\frac{x(x-1)}{2} = -x(x-2) + \frac{3}{2}x(x-1) = -x^2 + 2x + \frac{3}{2}x^2 - \frac{3}{2}x =$$

$$= \frac{1}{2}x^2 + \frac{1}{2}x$$

METODO DI NEWTON: DIFFERENZE DIVISE

- Il polinomio di interpolazione $P_n(x)$ viene costruito esplicitamente nella forma

$$P_n(x) = a_0 + a_1 \cdot (x-x_0) + a_2 \cdot (x-x_0)\cdot(x-x_1) + \ldots + a_n \cdot (x-x_0)\cdot(x-x_1)\ldots(x-x_{n-1})$$

dove i coefficienti a_i sono dati dalle cosiddette **differenze divise**, rapporti incrementali (numeri) che si definiscono sugli n+1 nodi $(x_0, x_1, x_2, \ldots, x_n)$

$$a_i = f[x_0, \ldots, x_i] = \begin{cases} f(x_0) & i = 0 \\ \dfrac{f[x_1, \ldots, x_i] - f[x_0, \ldots, x_{i-1}]}{x_i - x_0} & i > 0 \end{cases}.$$

- Nel caso di 2 nodi (x_0, x_1), si definisce la differenza divisa di ordine 1

$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{(x_1 - x_0)} = f[x_1, x_0]$$

- Nel caso di 3 nodi (x_0, x_1, x_2), si definisce la differenza divisa di ordine 2

$$f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{(x_2 - x_0)} = f[x_1, x_0, x_2] = f[x_2, x_1, x_0]$$

- Nel caso generale di n+1 nodi $(x_0, x_1, x_2, ..., x_n)$, si definisce la differenza divisa di ordine n

$$f[x_0, x_1, x_2, ..., x_n] = \frac{f[x_1, ..., x_n] - f[x_0, ..., x_{n-1}]}{(x_n - x_0)} = ... = f[x_1, x_2, x_n, ..., x_0]$$

La differenza divisa è un numero invariante alle permutazioni, cioè dipende soltanto dai nodi e non dall'ordine in cui si trovano.

- Le differenze divise associate agli n+1 nodi x_i (con i=0, ..., n) si costruiscono mediante una tabella a partire dai punti noti (x_i, y_i=f(x_i))

	ordine 0	ordine 1	ordine 2	ordine 3	ordine 4	ordine n
x_i	$y_i=f(x_i)$					
x_0	$f(x_0)$					
x_1	$f(x_1)$	$f[x_0, x_1]$				
x_2	$f(x_2)$	$f[x_1, x_2]$	$f[x_0, x_1, x_2]$			
x_3	$f(x_3)$	$f[x_2, x_3]$	$f[x_1, x_2, x_3]$	$f[x_0, x_1, x_2, x_3]$		
x_4	$f(x_4)$	$f[x_3, x_4]$	$f[x_2, x_3, x_4]$	$f[x_1, x_2, x_3, x_4]$	$f[x_0, x_1, x_2, x_3, x_4]$	
.	.	.	.	.	.	
.	.	.	.	.	.	
.	.	.	.	.	.	
.	.	.	.	.	.	
x_n	$f(x_n)$	$f[x_{n-1}, x_n]$				
	$f[x_{n-2}, x_{n-1}, x_n]$	.....			$f[x_0, x_1, ..., x_n]$	

- I coefficienti del polinomio interpolatore coincidono con gli elementi sulla diagonale principale della tabella così costruita

$$P_n(x) = a_0 + a_1 \cdot (x - x_0) + a_2 \cdot (x - x_0) \cdot (x - x_1) + ... + a_n \cdot (x - x_0) \cdot (x - x_1) ... (x - x_{n-1})$$

con

$$a_0 = f(x_0)$$
$$a_1 = f[x_0, x_1]$$
$$a_2 = f[x_0, x_1, x_2]$$
$$a_3 = f[x_0, x_1, x_2, x_3]$$
$$a_4 = f[x_0, x_1, x_2, x_3, x_4]$$
$$a_n = f[x_0, x_1, \ldots\ldots\ldots\ldots, x_n]$$

- Il metodo di interpolazione di Newton è preferibile a quello di Lagrange perché:
  1. richiede meno operazioni ($O(n)$ a fronte di $O(n^2)$) - minore complessità
  2. ha una maggiore stabilità numerica (conseguenza di 1.)
  3. bisogna memorizzare meno dati per l'interpolazione (gli elementi diagonale della tabella)
  4. esistenza di un algoritmo semplice per il calcolo dei coefficienti a_i del polinomio (Difdiv)
  5. per migliorare il polinomio e costruire il $P_{n+1}(x)$ non bisogna ripartire dall'inizio, ma è sufficiente aggiungere un punto noto, calcolare una nuova riga della tabella delle differenze divise, calcolare il nuovo elemento sulla diagonale e ... usarlo.

Esempio

1) Dati i 3 punti (0, 0) , (1, 1) , (2, 3) costruire la tabella delle differenze divise e quindi il polinomio interpolatore utilizzando il metodo di Newton.

Sono dati n+1=3 punti $\Rightarrow$ il polinomio interpolante è di grado n=2 e si scrive nella forma:

$$P_2(x) = a_0 + a_1 \cdot (x - x_0) + a_2 \cdot (x - x_0) \cdot (x - x_1)$$

dove i coefficienti a_i sono le differenze divise che si calcolano sulla diagonale della tabella:

ord.0		ord.1	ord.2
x_i	$y_i=f(x_i)$		
0	**0**		
1	1	(1−0)/(1−0)= **1**	
2	3	(3−1)/(2−1)= 2	(2−1)/(2−0)= **1/2**

Il polinomio cercato è:

$$P_2(x) = \underline{0} + \underline{1} \cdot (x - 0) + \underline{\frac{1}{2}} (x - 0)(x - 1) =$$
$$= x + \frac{x}{2}(x - 1) =$$
$$= \frac{x^2}{2} + \frac{x}{2}$$

ESERCIZI SVOLTI

1. Calcolare il polinomio interpolatore per i 4 punti (0, 0) , (1, 2) , (2, -1) , (3, 0) utilizzando il metodo di Newton il metodo di Lagrange.

Sono dati n+1=4 punti $\Rightarrow$ il polinomio cercato è di grado n=3.

Ovviamente il polinomio sarà lo stesso (unicità del polinomio interpolatore).

Metodo di Newton

Il polinomio si presenta nella forma:

$$P_3(x) = f(x_0) + (x - x_0) \cdot \left(f[x_0, x_1] + (x - x_1) \cdot \left(f[x_0, x_1, x_2] + (x - x_2) \cdot \left(f[x_0, x_1, x_2, x_3] \right) \right) \right)$$

Si costruisce la tabella delle differenze divise a partire dai punti noti:

	ord.0	ord.1	ord.2	ord.3
x_i	$y_i=f(x_i)$			
0	**0**			
1	2	(2−0)/(1−0)= **2**		
2	−1	(−1−2)/(2−1)= −3	(−3−2)/(2−0)= **- 5/2**	
3	0	(0−(−1))(3−2)= 1	(1−(−3))/(3−1)= 2	(2−(−2.5))/(3−0)= **3/2**

Si costruisce il polinomio utilizzando i coefficienti che compaiono sulla diagonale della tabella.

$$\begin{aligned} P_3(x) &= 0 + (x-0) \cdot \left(2 + (x-1) \cdot \left(-\frac{5}{2} + (x-2) \cdot \frac{3}{2} \right) \right) = \\ &= x\left(2 + \frac{3}{2}x^2 - \frac{11}{2}x - \frac{3}{2}x + \frac{11}{2} \right) = \\ &= \frac{3}{2}x^3 - 7x^2 + \frac{15}{2}x \end{aligned}$$

Metodo di Lagrange

Il polinomio si presenta nella forma:

$$P_3(x) = \sum_{j=0}^{3} y_j \cdot l_j(x) = \sum_{j=0}^{3} y_j \frac{\prod_{i=0, i\neq j}^{3} (x - x_i)}{\prod_{i=0, i\neq j}^{3} (x_j - x_i)}$$

Sostituendo i punti noti si ottiene:

$$P_3(x) = y_0 \frac{\prod_{i=0,i\neq 0}^{3}(x-x_i)}{\prod_{i=0,i\neq 0}^{3}(x_0-x_i)} + y_1 \frac{\prod_{i=0,i\neq 1}^{3}(x-x_i)}{\prod_{i=0,i\neq 1}^{3}(x_1-x_i)} + y_2 \frac{\prod_{i=0,i\neq 2}^{3}(x-x_i)}{\prod_{i=0,i\neq 2}^{3}(x_2-x_i)} + y_3 \frac{\prod_{i=0,i\neq 3}^{3}(x-x_i)}{\prod_{i=0,i\neq 3}^{3}(x_3-x_i)} =$$

$$= 0 \cdot \frac{(x-1)(x-2)(x-3)}{(0-1)(0-2)(0-3)} + 2 \cdot \frac{(x-0)(x-2)(x-3)}{(1-0)(1-2)(1-3)} + (-1) \cdot \frac{(x-0)(x-1)(x-3)}{(2-0)(2-1)(2-3)} + 0 \cdot \frac{(x-0)(x-1)(x-2)}{(3-0)(3-1)(3-2)} =$$

$$= 0 + 2 \cdot \frac{(x)(x-2)(x-3)}{(1)(-1)(-2)} + (-1) \cdot \frac{(x)(x-1)(x-3)}{(2)(1)(-1)} + 0 =$$

$$= 2 \cdot \frac{(x)(x-2)(x-3)}{2} + \frac{(x)(x-1)(x-3)}{2} =$$

$$= \frac{3}{2}x^3 - 7x^2 + \frac{15}{2}x$$

ALGORITMI

Algoritmo. Difdiv (*n*, **x**, **f**)

Commento. I vettori **x** e **f** (dimensione *n*+1), inizialmente contengono i dati x_i e $y_i=f(x_i)$ (cioé gli *n*+1 punti noti da interpolare) con *i*=0, ..., *n*. Le colonne della tabella alle differenze divise vengono successivamente determinate e memorizzate nel vettore **f**. Alla fine il vettore **f** contiene gli elementi diagonale della tabella.

Parametri. Input : *n*, **x**, **f**
Output: **f**

1. $f_i=y_i$ per *i*=0, ..., *n* (inizializzazione f_i)
2. Ciclo 1: *i*=1, ..., *n*
3. Ciclo 2: *j*=*n*, ..., *i*
4. $$f_j = \frac{(f_j - f_{j-i})}{(x_j - x_{j-i})}$$
5. Fine Ciclo 2
6. Fine Ciclo 1
7. Exit

End

Algoritmo. Interp (*n*, **x**, **f**, *t*, *p*)

Commento. I vettori **x** e **f** (dimensione *n*+1), inizialmente contengono i dati $\{x_i\}$ e gli elementi diagonale della tabella delle differenze divise (Output dall'algoritmo Difdiv). Assegnato un valore *t*, l'algoritmo valuta il valore che il polinomio interpolatore di Newton assume in *t* e lo memorizza in *p*.

Parametri. Input : *n*, **x**, **f**, *t*
Output: *p*

1. $p=f_n$
2. Ciclo 1: *i*=*n*–1, ..., 0
3. $$p = f_i + (t - x_i)\,p$$

4. Fine Ciclo 1
5. Exit
   End

ESERCIZI PROPOSTI

1. Calcolare il polinomio interpolatore per i 3 punti (-2, 0) , (0, 0) , (2, 1), utilizzando i metodi di Lagrange e di Newton.

$$\left[\frac{1}{8}x^2+\frac{1}{4}x\right]$$

2. Calcolare il polinomio interpolatore per i 3 punti (-2, 0) , (0, 6) , (2, 1), utilizzando i metodi di Lagrange e di Newton.

$$\left[-\frac{11}{8}x^2+\frac{1}{4}x+6\right]$$

3. Calcolare il polinomio interpolatore per i 5 punti $\left(k, \text{sen}\left(\frac{k\pi}{4}\right)\right)$ con k=0, ..., 4, utilizzando i metodi di Lagrange e di Newton.

$$\left[0.0143x^4-0.1144x^3+0.0360x^2+0.7712x\right]$$

4. Calcolare il polinomio interpolatore per i 5 punti $\left(k, \exp\left(\frac{k}{2}\right)\right)$ con k=0, ..., 4, utilizzando i metodi di Lagrange e di Newton.

$$\left[0.007379x^4-0.0012248x^3+0.15509x^2+0.4850x+1\right]$$

INTERPOLAZIONE POLINOMIALE A TRATTI

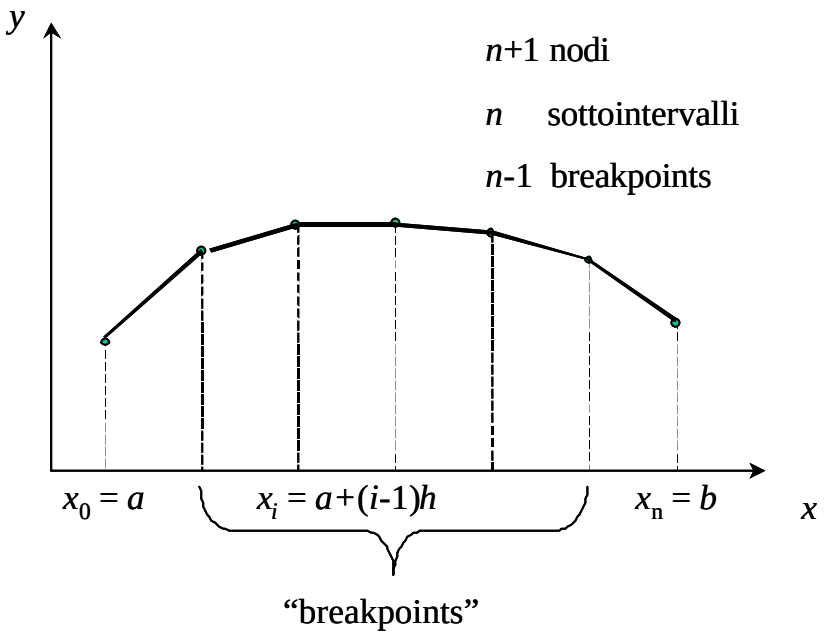
In presenza di un numero elevato di punti da interpolare possono manifestarsi problemi con il metodo della interpolazione ordinaria, che utilizza un solo polinomio interpolatore su tutto l'intervallo.

Il polinomio interpolante è di grado progressivamente sempre più elevato all'aumentare del numero di punti e tende ad avere un profilo molto irregolare (a "carattere oscillatorio"), in molti casi indesiderato.

Tale problema può essere eliminato adottando una tecnica di interpolazione polinomiale a tratti, ovvero suddividendo l'intervallo in sottointervalli all'interno dei quali è effettuata una interpolazione con polinomi di grado inferiore.

INTERPOLAZIONE LINEARE A TRATTI

Nel caso più semplice i polinomi utilizzati nei sottointervalli sono di primo grado. Tale interpolazione lineare a tratti corrisponde a unire in successione con tratti rettilinei i punti da interpolare.



Noto un insieme di $(n+1)$ punti $\{(x_i, y_i),\ i = 0, \ldots, n\}$ da interpolare, con $x_0 = a$, e $x_n = b$, si definiscono n sottointervalli dell'intervallo $[a,b]$:

$[x_{i-1}, x_i]$, $\quad i = 1,\ldots, n$ $\quad$ è il generico sottointervallo

- le $(n+1)$ ascisse x_i per $i = 0,\ldots, n$ sono, come sempre, dette nodi;
- le $(n-1)$ ascisse x_i per $i = 1, \ldots, n-1$, di "frontiera" tra i sottointervalli, sono dette breakpoints.

INTERPOLAZIONE CON FUNZIONI SPLINES

Questo tipo di interpolazione riproduce in forma matematica quello che nella pratica è il disegno effettuato con il tracciacurve (flessibile di materiale plastico e modellabile a piacere).

DEFINIZIONE DI FUNZIONE SPLINE

La funzione $S(x)$ è la *spline* di grado m associata agli $(n+1)$ nodi x_i in $[a,b]$ con $i = 0,\dots, n$

SE si verificano le seguenti CONDIZIONI:

1. $S(x)$ è un polinomio di grado m in ogni sottointervallo $[x_{i-1}, x_i]$ per $i = 1,\dots, n$:

$$S_i(x) = a_i + b_i x + c_i x^2 + d_i x^3 + \dots + coeff(m+1)_i x^m$$

2. ogni polinomio $S_i(x)$ e le sue prime $m-1$ derivate sono continue in tutto l'intervallo $[a,b]$ e in particolare negli $(n-1)$ breakpoints x_i per $i = 1,\dots, n-1$:

$$S_i^{(k)}(x_i) = S_{i+1}^{(k)}(x_i) \text{ per } i = 1, \dots, n-1 \text{ per } k = 0, \dots, m-1$$

3. la $S(x)$ interpola gli $(n+1)$ punti:

$$S(x_i) = y_i \text{ per } i = 0, \dots, n$$

Si noti che la funzione $S(x)$ è definita a tratti:

$$S(x) = S_i(x) \quad \text{in ogni sottointervallo} \quad [x_{i-1}, x_i] \text{ per } i = 1,\dots, n$$

NUMERO DI INCOGNITE PER DEFINIRE LA SPLINE

Dato che ciascuna delle n splines è (condizione 1.) un polinomio di grado m e quindi ha $m+1$ coefficienti incogniti, la funzione interpolante sull'intero intervallo $[a,b]$ è univocamente definita se si determinano gli ***n*(*m*+1) coefficienti incogniti** totali.

NUMERO DI EQUAZIONI DISPONIBILI

Le equazioni utili alla determinazione dei coefficienti incogniti si ricavano dalle condizioni imposte alla funzione $S(x)$ per essere una spline:

2) $m(n-1)$ equazioni di continuità della funzione e delle sue derivate nei breakpoints:

$$S_i^{(k)}(x_i) = S_{i+1}^{(k)}(x_i) \text{ per } i = 1, \dots, n-1 \text{ per } k = 0, \dots, m-1$$

3) $(n+1)$ equazioni di interpolazione: $S(x_i) = y_i$

- QUINDI SI HANNO: ***n*(*m*+1) - (*m* - 1) equazioni lineari** con ***n*(*m*+1) coefficienti incogniti**.

EQUAZIONI SUPPLEMENTARI

- Condizione necessaria affinchè la *spline* sia univocamente determinata è che il numero di equazioni sia pari al numero delle incognite, ovvero i coefficienti dei polinomi $S_i(x)$;
- occorre pertanto aggiungere alle precedenti equazioni le mancanti $(m-1)$ supplementari, in corrispondenza degli estremi a e b dell'intervallo di definizione della *spline;*

- queste nuove equazioni sono le cosiddette **(*m* - 1) condizioni al contorno**, utilizzando la terminologia generalmente applicata alle equazioni differenziali.

LA SPLINE CUBICA NATURALE

- È la funzione spline più utilizzata;
- è detta spline cubica perchè si basa su polinomi di grado $m=3$ e richiede $m-1=2$ condizioni al contorno;
- se le condizioni al contorno implicano l'annullamento della derivata seconda agli estremi dell'intervallo $[a,b]$ si parla convenzionalmente di **spline cubica naturale**.

CONDIZIONI che si devono verificare:

1. $S_i(x)$ è un polinomio di grado 3 in ogni sottointervallo $[x_{i-1}, x_i]$ per $i = 1,\dots, n$:

$$S_i(x) = a_i + b_i x + c_i x^2 + d_i x^3$$

2. ogni polinomio $S_i(x)$, la sua derivata prima e la sua derivata seconda sono continue in tutto l'intervallo $[a,b]$ e in particolare negli $(n-1)$ breakpoints x_i per $i = 1,\dots, n-1$:

$$S_i^{(k)}(x_i) = S_{i+1}^{(k)}(x_i) \text{ per } i = 1, \dots, n-1 \text{ per } k = 0, 1, 2$$

3. la spline $S(x)$ interpola gli $(n+1)$ punti:

$$S(x_i)=y_i \text{ per } i = 0, \dots, n$$

- La **spline cubica** ha **4*n* coefficienti incogniti** totali (4 in ogni sottointervallo, *n* sottointervalli)
- essendo spline cubica **naturale**, le 2 equazioni supplementari necessarie implicano l'annullamento della derivata seconda negli estremi *a* e *b* dell'intervallo:

$$S_1^{(2)}(a) = 0 \quad S_n^{(2)}(b) = 0$$

- Il problema si riconduce alla soluzione di un sistema di equazioni lineari di ordine $4n$.

DIVERSA IMPOSTAZIONE

È possibile utilizzare un'impostazione differente che porta a un sistema di equazioni lineari di ordine decisamente più basso, pari a $(n+1)$, e di forma più semplice: sistema simmetrico tridiagonale a diagonale dominante.

- Come **incognite**, anziché i $4n$ coefficienti delle spline cubica (4 in ogni sottointervallo, *n* sottointervalli), si scelgono gli $(n+1)$ **valori** M_i che le **derivate seconde** della spline cubica assumono **nei nodi**:

$$S_i^{(2)} = M_i \text{ per } i = 0, \dots, n$$

- Dato che la spline in esame è cubica (polinomio di grado 3) la sua generica **derivata seconda è** un polinomio di grado 1, cioè **una retta**.
-
- **In ogni sottointervallo** si scrive l'equazione della **retta** che passa **per** i **due punti** (inizio e fine) del sottointervallo $[x_{i-1}\ ,\ x_i]$ stesso:

$$\frac{x-x_{i-1}}{x_i-x_{i-1}}=\frac{S_i^{(2)}-M_{i-1}}{M_i-M_{i-1}} \text{ cioè } \frac{x-x_{i-1}}{h_i}=\frac{S_i^{(2)}-M_{i-1}}{M_i-M_{i-1}}$$

  con h_i ampiezza del sottointervallo.

- L'equazione della derivata seconda incognita risulta quindi:

$$S_i^{(2)}(x)=\frac{(x_i-x)M_{i-1}+(x-x_{i-1})M_i}{h_i} \qquad \text{derivata seconda}$$

- **Integrando** l'equazione della derivata seconda si ricavano le espressioni della derivata prima e della funzione:

$$S_i^{(1)}(x)=\frac{-(x_i-x)^2M_{i-1}+(x-x_{i-1})^2M_i}{2h_i}+C_i \qquad \text{derivata prima}$$

$$S_i(x)=\frac{(x_i-x)^3M_{i-1}+(x-x_{i-1})^3M_i}{6h_i}+C_i(x-x_{i-1})+D_i \qquad \text{funzione}$$

- Le **costanti d'integrazione** C_i e D_i si ricavano imponendo il rispetto della condizione d'interpolazione 3.:

  $$S(x_{i-1})=y_{i-1} \text{ e } S(x_i)=y_i$$

  da cui:

$$C_i=\frac{y_i-y_{i-1}}{h_i}-\frac{h_i(M_i-M_{i-1})}{6}$$

$$D_i=y_{i-1}-\frac{h_i^2}{6}M_{i-1}$$

- Perché la funzione $S_i(x)$ sia una spline bisogna ancora imporre il rispetto della condizione di continuità 1. sulla derivata prima nei breakpoints:

$$S_i{}^{(1)}(x_i)=S_{i+1}{}^{(1)}(x_i) \text{ per } i=1\ ,\ \dots\ ,\ n-1$$

- In questo modo si ricava il sistema di equazioni lineari di $(n-1)$ equazioni in $(n+1)$ incognite (le M_i per $i=0\ ,\ \dots\ ,\ n$):

$$h_i M_{i-1} + 2(h_i + h_{i+1})M_i + h_{i+1}M_{i+1} = 6\frac{y_{i+1} - y_i}{h_{i+1}} - 6\frac{y_i - y_{i-1}}{h_i} \qquad i = 1, n-1$$

(*)

a cui vanno aggiunte le 2 equazioni al contorno dovute alla 'naturalità' della spline:

$$S_1^{(2)}(x_0) = M_0 = 0$$
$$S_1^{(2)}(x_n) = M_n = 0$$

- per giungere al sistema di equazioni lineari di ordine (n+1) finale:

$$\begin{cases} M_0 = 0 \\ h_i M_{i-1} + 2(h_i + h_{i+1})M_i + h_{i+1}M_{i+1} = 6\dfrac{y_{i+1} - y_i}{h_{i+1}} - 6\dfrac{y_i - y_{i-1}}{h_i} \qquad i = 1, n-1 \\ M_n = 0 \end{cases}$$

Si nota che il sistema di equazioni (*), di ordine (n−1), è tridiagonale a diagonale dominante e simmetrico con matrice dei coefficienti pari a:

$$\mathbf{A} = \begin{bmatrix} 2(h_1 + h_2) & h_2 & \dots & 0 \\ h_2 & 2(h_2 + h_3) & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 2(h_{n-1} + h_n) \end{bmatrix}$$

e pertanto risolvibile con il metodo di Gauss senza pivoting.

ESERCIZI SVOLTI

1. Calcolare la spline cubica naturale interpolante a tratti i tre punti: (0, 0) , (1, 1) , (2, 0).

Sono dati n+1=3 punti $\Rightarrow$ i sottointervalli sono quindi 2 e in ciascun sottointervallo la spline cubica si scrive come:

$$S_i(x) = \frac{(x_i - x)^3 M_{i-1} + (x - x_{i-1})^3 M_i}{6h_i} + C_i(x - x_{i-1}) + D_i$$

con:

$$C_i = \frac{y_i - y_{i-1}}{h_i} - \frac{h_i(M_i - M_{i-1})}{6}$$

$$D_i = y_{i-1} - \frac{h_i^2}{6}M_{i-1}$$

$$x \in [x_{i-1}, x_i] \quad i = 1,2 \quad h_i = x_i - x_{i-1}$$

Trattandosi di spline naturale le condizioni al contorno sono:

$$M_0 = M_2 = 0$$

Il sistema per determinare i coefficienti della spline cubica naturale si riduce all'equazione:

$$2(h_1 + h_2)M_1 = 6\left(\frac{y_2 - y_1}{h_2} - \frac{y_1 - y_0}{h_1}\right)$$

$$h_1 = 1 - 0 = 1;$$

$$h_2 = 2 - 1 = 1;$$

$$\text{da cui } M_1 = -3$$

Esaminando ciascun sottointervallo:

nel primo sottointervallo [(0, 0) , (1, 1)]

$$C_1 = \frac{y_1 - y_0}{h_1} - \frac{h_1(M_1 - M_0)}{6} = \frac{3}{2};$$

$$D_1 = y_0 - \frac{h^2{}_1}{6} M_0 = 0;$$

$$S_3(x) = \frac{(x_1 - x)^3 M_0 + (x - x_0)^3 M_1}{6h_1} + C_1(x - x_0) + D_1 =$$

$$= \frac{(1-x)^3 \cdot 0 + (x-0)^3(-3)}{6 \cdot 1} + \frac{3}{2}(x-0) + 0 = -\frac{1}{2}x^3 + \frac{3}{2}x \quad x \in [0,1]$$

nel secondo sottointervallo [(1, 1) , (2, 0)]

$$C_2 = \frac{y_2 - y_1}{h_2} - \frac{h_2(M_2 - M_1)}{6} = -\frac{3}{2};$$

$$D_2 = y_1 - \frac{h^2{}_2}{6} M_1 = \frac{3}{2};$$

$$S_3(x) = \frac{(x_2 - x)^3 M_1 + (x - x_1)^3 M_2}{6h_2} + C_2(x - x_1) + D_2 =$$

$$= \frac{(2-x)^3 \cdot -3 + 0}{6 \cdot 1} - \frac{3}{2}(x-1) + \frac{3}{2} = -\frac{1}{2}(2-x)^3 - \frac{3}{2}x + 3 \quad x \in [1,2]$$

ALGORITMI

Algoritmo : Spline (*n, x, y, z)*

Commento. L'algoritmo determina i coefficienti $M_1, M_2,...,M_{n-1}$ necessari per rappresentare la spline cubica naturale passante per i punti (x_i, y_i), $i = 0,1,...n$. Tali numeri vengono memorizzati nel vettore **z**. I vettori **d** e **c** vengono introdotti per memorizzare rispettivamente la diagonale e la codiagonale del sistema tridiagonale simmetrico la cui sola soluzione fornisce i valori $\{M_i\}$. Il termine noto del sistema viene memorizzato nel vettore **b**.

Parametri. Input: *n, x, y*

Output : *z*

Ciclo 1 : i= 1,..., *n*-2

$d_i = 2(x_{i+1}-x_{i-1})$

$c_i = x_{i+1} - x_i$

$$b_i = 6\left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i} - \frac{y_i - y_{i-1}}{x_i - x_{i-1}}\right)$$

Fine ciclo 1

$d_{n-1} = 2(x_n-x_{n-2})$

$$b_{n-1} = 6\left(\frac{y_n - y_{n-1}}{x_n - x_{n-1}} - \frac{y_{n-1} - y_{n-2}}{x_{n-1} - x_{n-2}}\right)$$

** Processo di eliminazione di Gauss per il sistema tridiagonale*

Ciclo 2 : i = 2,..., *n*-1

$d_i = d_i - c^2_{i-1} / d_{i-1}$

$b_i = b_i - (c_{i-1} / d_{i-1}) b_{i-1}$

Fine ciclo 2

** Soluzione sistema bidiagonale*

$z_{n-1} = b_{n-1} / d_{n-1}$

Ciclo 3 : i=2,..., n-1

$z_{n-i} = (b_{n-1} - c_{n-i} z_{n+1-i}) / d_{n-i}$

Fine ciclo 3

Exit

End

Algoritmo : Valspl (*n, x, y, z, t, s,* ier*)*

Commento . Noti i coefficienti $M_1, M_2, ...M_{n-1}$, contenuti nel vettore **z** ,della spline cubica naturale passante per i punti (x_i, y_i), $i = 0, 1,..., n$, Valpl valuta il valore *s* che tale spline assume nel punto t , $x_0 \le t \le x_n$. Se $t \in [x_0, x_n]$ la variabile ier = 0, altrimenti ier=1.

Parametri. Input :*n, x, y, z, t*

Output :*s,* ier

Ciclo 1 : i=1,.., *n*

se $x_{i-1} \le t \le x_i$ poni ier = 0; vai al punto 6.

Fine ciclo 1

ier = 1

Exit

$h = x_i - x_{i-1}$

$$s = \frac{(x_i - t)^3 z_{i-1} + (t - x_{i-1})^3 z_i}{6h} + \left[\frac{y_i - y_{i-1}}{h} - \frac{h}{6}(z_i - z_{i-1})\right](t - x_{i-1}) + y_{i-1} - \frac{h^2}{6} z_{i-1}$$

Exit

End

ESERCIZI PROPOSTI

1. Calcolare la spline cubica naturale per l'insieme di dati {(0,0) (1,2), (2,0)}

[$3x-x^3$, $-2+9x-6x^2+x^3$]

2. Calcolare la spline cubica naturale per l'insieme di dati {(0,0), (1,2), (2,-2), (3,0)}

[$4x-2x^3$, $-6+22x-18x^2+4x^3$, $42-50x+18x^2-2x^3$]

3. Determinare la *spline* interpolante per l'insieme di dati $\left(k, sin\left(\frac{kp}{2}\right)\right)_{k=0}^{4}$

$$\left[\frac{3}{2}x-\frac{1}{2}x^3,-1+\frac{9}{2}x-3x^2+\frac{1}{2}x^3,-1+\frac{9}{2}x-3x^2+\frac{1}{2}x^3,26-\frac{45}{2}x+6x^2-\frac{1}{2}x^3\right]$$

APPROSSIMAZIONE DI DATI

Riferimento al testo: Cap. V

CRITERIO DI APPROSSIMAZIONE DI DATI

Il problema dell'approssimazione consiste nel, assegnati $m+1$ dati (x_0,y_0), ... , (x_m,y_m), individuare una funzione $f(x)$, appartenente ad un certo insieme o classe, che meglio li approssimi, secondo un qualche criterio che andremo ad indicare.

Ciò che distingue l'approssimazione dei dati dall'interpolazione è che la funzione approssimante non ha il vincolo di passare per i punti assegnati (dati), ovvero non è tenuta ad assumere in corrispondenza delle ascisse dei dati $\{x_i\}$ il valore delle corrispondenti ordinate $\{y_i\}$.

Si preferisce all'interpolazione quando i dati sono disponibili in numero elevato, eventualmente affetti da errore (*rumore*). In questo caso, il procedimento di approssimazione mira a ridurre in parte l'effetto degli errori.

SOLUZIONE DEL PROBLEMA

Lo scostamento tra dati e funzione approssimante è in genere definito dalla norma del vettore degli errori

$$e_i = f(x_i) - y_i \quad , \quad i = 0, ..., m.$$

La scelta del tipo di norma conduce a problemi di approssimazione diversi: nel caso della norma l_2 abbiamo il problema dei minimi quadrati mentre nel caso di norma l_∞ si ha approssimazione minimax.

- **Criterio dei Minimi Quadrati** (caso discreto)

  Noti $m+1$ punti $\{x_i, y_i\}$ con $i=0, \ldots, m$, la $f(x)$ viene scelta in modo da minimizzare la norma due del vettore degli scarti. Questo equivale a minimizzare la somma dei quadrati degli scarti tra la funzione approssimante valutata nei nodi $\{x_i\}$ ed i corrispondenti dati $\{y_i\}$

$$f_n(x) = \arg\min_f \sum_{i=0}^{m} (f(x_i) - y_i)^2$$

  Se esistono dati a cui si attribuisce maggiore importanza (es. derivanti da misurazioni più precise) allora è utile introdurre dei **pesi** w_i

$$f_n(x) = \arg\min_f \sum_{i=0}^{m} w_i (f(x_i) - y_i)^2$$

- **Criterio Minimax** (caso discreto)

  La $f_n(x)$ viene scelta in modo da minimizzare la norma infinito del vettore degli scarti. Questo corrisponde a minimizzare lo scarto massimo

$$f_n = \min_f \max_{i=0\ldots m} |f(x_i) - y_i|$$

REGRESSIONE LINEARE

Abbiamo un problema di regressione lineare ogniqualvolta la funzione approssimante possa essere espressa e ricercata come combinazione lineare di un insieme di funzioni di base.
Più in dettaglio, assegnate n+1 funzioni elementari $j_k(x)$, $k = 0, \ldots, n$, la funzione approssimante può essere espressa come combinazione lineare queste

$$f(x) = \sum_{k=0}^{n} c_k j_k(x)$$

MINIMI QUADRATI: CALCOLO DELL'APPROSSIMAZIONE

Il metodo si applica a problemi di regressione lineare.
L'errore quadratico tra i dati e la funzione approssimante è in questo caso esprimibile come:

$$R(c) = e^2 = \sum_{i=0}^{m} \left[y_i - \sum_{k=0}^{n} c_k j_k(x_i) \right]^2$$

L'obiettivo del criterio è quello di determinare i coefficienti incogniti c_k della funzione approssimante che minimizzino l'errore ε^2. La ricerca di tale minimo può essere condotta annullando le derivate parziali della funzione $R(c)$ rispetto ai coefficienti c_k.

Derivando $R(c)$ rispetto ai coefficienti c_k si ottengono le n+1 equazioni:

$$\sum_{i=0}^{m} \left[y_i - \sum_{j=0}^{n} c_j j_j(x_i) \right] j_k(x_i) = 0 \qquad k = 0, n$$

necessarie per il calcolo dei coefficienti che definiscono la funzione $f(x)$.
Tali equazioni possono essere trascritte sotto forma di sistema di equazioni lineari

$$\begin{bmatrix} \sum_i j_0(x_i)^2 & \sum_i j_0(x_i) j_1(x_i) & \cdots & \sum_i j_0(x_i) j_n(x_i) \\ \sum_i j_1(x_i) j_0(x_i) & \sum_i j_1(x_i)^2 & \cdots & \sum_i j_1(x_i) j_n(x_i) \\ \vdots & \vdots & \vdots & \vdots \\ \sum_i j_n(x_i) j_0(x_i) & \sum_i j_n(x_i) j_1(x_i) & \cdots & \sum_i j_n(x_i)^2 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} \sum_i y_i j_0(x_i) \\ \sum_i y_i j_1(x_i) \\ \vdots \\ \sum_i y_i j_n(x_i) \end{bmatrix}$$

che, raccolto in forma vettoriale, assume la forma

$$Bc = d \, .$$

Le equazioni costituenti tale sistema vengono definite come *equazioni normali*.
La determinazione dei coefficienti c_k , ossia l'individuazione della funzione $f(x)$, è ricondotta alla risoluzione del precedente sistema lineare.

ELABORAZIONE DEL RISULTATO

Il sistema delle equazioni normali può essere rielaborato nella forma per agevolare l'applicazione delle tecniche di risoluzione già note dalle precedenti esercitazioni.

In particolare il sistema $\mathbf{Bc} = \mathbf{d}$ può essere scritto come $\mathbf{A}^T\mathbf{A}\,\mathbf{c} = \mathbf{A}^T\,\mathbf{y}$ dove $\mathbf{c}$ è il vettore colonna dei coefficienti incogniti, $\mathbf{y}$ è il vettore colonna delle ordinate y_i e $\mathbf{A}$ è la seguente matrice di dimensioni $(m+1)\times(n+1)$:

$$\mathbf{A} = \begin{bmatrix} \varphi_0(x_0) & \varphi_1(x_0) & \dots & \varphi_n(x_0) \\ \varphi_0(x_1) & \varphi_1(x_1) & \dots & \varphi_n(x_1) \\ . & . & . & . \\ . & . & . & . \\ \varphi_0(x_m) & \varphi_1(x_m) & \dots & \varphi_n(x_m) \end{bmatrix}$$

Il vettore **c** può quindi essere espresso come

$$\mathbf{c} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{y}$$

dove la matrice $(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$ prende il nome di *pseudo-inversa* di **A.**
Questa coincide ovviamente con l'inversa di **A**, ogniqualvolta **A** risulti essere quadrata (si ricorda a tale riguardo che in ogni caso non è possibile definire la matrice inversa di una matrice rettangolare).
Si osservi che $\mathbf{B} = \mathbf{A}^T \mathbf{A}$ è una matrice simmetrica definita positiva per cui la soluzione del sistema di equazioni lineari può fare ricorso alla fattorizzazione di Cholesky, che è più efficiente della fattorizzazione **LU** mediante algoritmo di Gauss.

INTERPRETAZIONE GEOMETRICA DEL RISULTATO

E' interessante rilevare che poiché $\mathbf{A}^T (\mathbf{y} - \mathbf{A}\mathbf{c}) = 0$ (dalla $\mathbf{A}^T\mathbf{A}\mathbf{c} = \mathbf{A}^T\mathbf{y}$), il vettore $\mathbf{y} - \mathbf{A}\mathbf{c}$ risulta essere ortogonale allo spazio generato delle colonne di **A** (composte dai valori assunti da ciascuna $j_k(x)$ in corrispondenza delle ascisse dei dati). L'approssimazione secondo il metodo dei minimi quadrati equivale ad una proiezione geometrica del vettore delle ordinate *y* sullo spazio generato dalle colonne di **A** : la misura della distanza euclidea di **y** da questo spazio è l'errore e^2.

DEGENERAZIONE DELL'APPROSSIMAZIONE IN INTERPOLAZIONE

Qualora si verifichi che il numero delle funzioni elementari sia uguale al numero di dati da approssimare, l'approssimazione degenera in interpolazione, **A** è in questo caso una matrice quadrata e l'errore quadratico si annulla (perché l'interpolazione garantisce che il valore assunto dalla funzione in corrispondenza dell'ascissa di un dato sia l'ordinata di quel dato).

ERRORE QUADRATICO

Trascrivendo il problema dell'approssimazione ai minimi quadrati utilizzando la matrice A, anche l'errore quadratico può essere riscritto in forma matriciale:

$$\varepsilon^2 = \mathbf{y}^T \left[\mathbf{I} - \mathbf{A}\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T \right] \mathbf{y} \qquad \text{(A)}$$

Essendo la definizione di errore quadratico medio:

$$R(c) = e^2 = \sum_{i=0}^{m} \left[y_i - \sum_{k=0}^{n} c_k j_k(x_i) \right]^2 \qquad \text{(B)}$$

l'obiettivo è quello di dimostrare la corrispondenza tra le due formulazioni (A) e (B).

1) Dapprima si sviluppi il prodotto in (A):

$$e^2 = \mathbf{y}^T\left[\ \mathbf{I} - \mathbf{A}\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\right]\mathbf{y} = \mathbf{y}^T\mathbf{y} - \mathbf{y}^T\mathbf{A}\left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{y}$$

2) si consideri che : $\mathbf{c} = (\mathbf{A}^T \times \mathbf{A})^{-1} \times \mathbf{A}^T \times \mathbf{y}$;

inoltre che $\mathbf{B} \times \mathbf{c} = \mathbf{d} \Rightarrow \mathbf{A}^T\mathbf{A} \times \mathbf{c} = \mathbf{A}^T \times \mathbf{y}$ da cui $\mathbf{c}^T \times \mathbf{A}^T = \mathbf{y}^T$

3) si riscriva l'errore quadratico medio nella forma:

$$e^2 = \mathbf{y}^T\mathbf{y} - \mathbf{y}^T\mathbf{A}\mathbf{c} = \mathbf{y}^T\mathbf{y} - \mathbf{c}^T\mathbf{A}^T\mathbf{A}\mathbf{c}$$

Quest'ultima trascrizione corrisponde, in forma matriciale, alla definizione (B) espressa in forma di serie.

ESEMPIO: TROVARE UNA RETTA APPROSSIMANTE

Un caso specifico di applicazione del criterio dei minimi quadrati nel campo dell'ingegneria e dell'economia consiste nel ricercare la funzione retta che meglio approssimi un certo insieme dato di punti (x_i , y_i).
Questo equivale a porre la funzione $f(x) = a+bx$, con $n+1 = 2$ funzioni di base, rispettivamente $\varphi_0(x) = 1$, $\varphi_1(x) = x$. Assegnati $m+1$ dati (x_i, y_i), le equazioni normali divengono:

$$\begin{bmatrix} m+1 & \sum_i x_i \\ \sum_i x_i & \sum_i x_i^2 \end{bmatrix}\begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} \sum_i y_i \\ \sum_i x_i y_i \end{bmatrix}$$

$$\begin{bmatrix} m+1 & \sum_i x_i \\ \sum_i x_i & \sum_i x_i^2 \end{bmatrix}\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum_i y_i \\ \sum_i x_i y_i \end{bmatrix}$$

Le equazioni normali soprascritte seguono dall'applicazione delle definizioni e del metodo espresse nel caso generale per descrivere la matrice **B** e i vettori **c** e **d.**

Esempio
1) Determinare i coefficienti della retta di regressione lineare e l'errore quadratico e^2 per i seguenti dati: (0, 1), (2, 3), (3, 4), (4, 6).

Le equazioni normali per la regressione lineare sono:

$$\begin{bmatrix} m+1 & \sum_i x_i \\ \sum_i x_i & \sum_i x_i^2 \end{bmatrix}\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum_i y_i \\ \sum_i x_i y_i \end{bmatrix}$$

dove
$(m+1)=4$ $\varphi_0(x) = 1$ $\varphi_1(x) = x$
$\sum x_i = 0 + 2 + 3 + 4 = 9;$
$\sum x^2{}_i = 0 + 4 + 9 + 16 = 29;$
$\sum y_i = 1 + 3 + 4 + 6 = 14;$

$\sum x_i y_i = 0 \cdot 1 + 2 \cdot 3 + 3 \cdot 4 + 4 \cdot 6 = 42$;

$$\begin{bmatrix} 4 & 9 \\ 9 & 29 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} 14 \\ 42 \end{bmatrix}$$

$a = 0.8 \quad b = 1.2 \quad f(x) = 0.8 + 1.2\,x$

L'errore quadratico medio è dato da:

$$e^2 = \sum_{i=0}^{m} \left[y_i - (a + bx) \right]^2$$

$$\left[1 - 0.8 \cdot 1 - 1.2 \cdot 0\right]^2 + \left[3 - 0.8 \cdot 1 - 1.2 \cdot 2\right]^2 + \left[4 - 0.8 \cdot 1 - 1.2 \cdot 3\right]^2 + + \left[6 - 0.8 \cdot 1 - 1.2 \cdot 4\right]^2 = 0.4$$

ESERCIZI PROPOSTI

1. Determinare i coefficienti della retta di regressione lineare e l'errore quadratico e^2 per i seguenti dati: (0, 5), (1, 3), (2, 0), (3, -1).

[$f(x) = 4.9 - 2.1x$, $e^2 = 0.7$]

2. Determinare l'approssimazione ai minimi quadrati e l'errore e^2 dei dati: (0, 2), (1, 0), (2, 2), (3, 6) con $j_0(x) = 1$, $j_1(x) = x$, $j_2(x) = x^2$.

[$f(x) = 1.9 - 3.1x + 1.5x^2$, $e^2 = 0.2$]

3. Determinare l'approssimazione ai minimi quadrati e l'errore e^2 dei dati: (0, 2), (1, 0), (2, 2), (3, 6), $j_0(x) = 1$, $j_1(x) = x$, $j_2(x) = x^3$.

[$f(x) = (120 - 97x + 22x^3) /69$, $e^2 = 18 / 23$]

4. Determinare l'approssimazione ai minimi quadrati e l'errore e^2 dei dati : (0, 2), (1, 0), (2, 2), (3, 6) con $j_0(x) = 1$, $j_1(x) = \sin(\pi x / 2)$, $j_2(x) = \cos(\pi x / 2)$.

[$f(x) = 2.5 - 3 \sin(\pi x / 2)$, $e^2 = 1$]

EQUAZIONI NON LINEARI

Riferimento al testo: Cap. VI

INTRODUZIONE

- Data l'equazione $f(x)=0$ con $f(x)$ funzione non lineare nel suo argomento x, se ne vogliono calcolare le soluzioni o radici (valori della x che verificano l'uguaglianza a 0).

- Le radici dell'equazione non lineare $f(x)=0$ possono non essere, in generale, esprimibili in forma chiusa e quindi vengono calcolate per via numerica mediante metodi iterativi che, a partire da una o più approssimazioni iniziali, producono una successione $x_0, x_1, x_2, ..., x_n, ...$ convergente (sotto certe ipotesi) alla radice cercata.

- Verranno presentati i seguenti metodi iterativi per il calcolo delle radici reali di equazioni non lineari: metodo di bisezione, "regula falsi", metodo delle tangenti o di Newton-Raphson e metodo delle secanti.

- Si dice che una successione di approssimazioni $\{x_k\}$ converge al valore cercato con ordine di convergenza p se esiste un numero $p \geq 1$ tale che

$$\lim_{k\to\infty} \frac{|x_{k+1} - x_\infty|}{|x_k - x_\infty|^p} = \lim_{k\to\infty} \frac{|e_{k+1}|}{|e_k|^p} = c \neq 0 \text{ , costante positiva.}$$

  L'ordine di convergenza rappresenta indicativamente il fattore di cui aumentano i decimali corretti delle approssimazioni x_k a partire da un certo punto in avanti.

- I metodi che verranno proposti presenteranno diverso ordine di convergenza: i metodi di bisezione e "regula falsi" convergono con ordine 1 (convergenza lineare); il metodo delle tangenti o di Newton-Raphson con ordine 2 (convergenza quadratica); il metodo delle secanti con ordine circa 1.6.

METODO DI BISEZIONE

Si applica a **funzioni continue** di cui si conosce un **intervallo iniziale** $[a, b]$ che ne contiene una radice. Se $f(a)\cdot f(b)<0$ significa che la funzione cambia segno in $[a, b]$; per continuità esistono quindi un numero dispari di radici nell'intervallo $[a, b]$.

Si costruisce una **successione di intervalli incapsulati** che contengono tutti la radice cercata.
Vediamo in dettaglio:

- Inizio: $k = 1 \Rightarrow$ si calcola il punto medio dell'intervallo iniziale $m_1=(a+b)/2$
- si valuta il segno di $f(m_1) f(b)$:
  - se $f(m_1) f(b) > 0 \Rightarrow$ nuovo intervallo d'interesse il semi-intervallo di sinistra $[a, m_1]$
  - se $f(m_1) f(b) < 0 \Rightarrow$ nuovo intervallo d'interesse il semi-intervallo di destra $[m_1, b]$
  - se $f(m_1) f(b) = 0 \Rightarrow m_1$ è la radice cercata
- se $f(m_1)\neq 0$ si prende il nuovo sotto-intervallo e si ricomincia il procedimento ($k = k +1$); a ogni iterazione l'ampiezza dell'intervallo che contiene la radice si dimezza.
- L'ampiezza dell'intervallo tende a 0 come $1/2^k$.

- Al passo iterativo k-esimo il punto medio $m_k=(a_{k-1}+b_{k-1})/2$ è la stima della radice cercata.

Algoritmo. Bisez (a_0, b_0, f, toll, x, ier)
Commento. Determina, con tolleranza relativa toll, una radice x dell'equazione non lineare $f(x)=0$, nell'intervallo $[a_0, b_0]$. Se $f(a_0)\ f(b_0)<0$, ier assume valore 0 e il calcolo di x viene effettuato; altrimenti l'algoritmo si arresta e segnala l'inconveniente ponendo ier=1.
Parametri. Input: a_0, b_0, f, toll
Output: x, ier

1. se $f(a_0)\ f(b_0)>0$ poni ier=1; Exit
2. ier = 0, k = 0
3. $k = k+1$
4. $m_k = \frac{1}{2}\ (a_{k-1} + b_{k-1})$
5. se $|\ b_{k-1} - a_{k-1}\ | < 2$ toll $|\ m_k|$, oppure $|\ b_{k-1} - a_{k-1}\ | < 2$ toll allora poni $x = m_k$; Exit
6. se $f(b_0)\ f(m_k) < 0$ allora poni $a_k = m_k$ e $b_k = b_{k-1}$ e vai al punto 3
7. $a_k = a_{k-1}$, $b_k = m_k$
8. vai al punto 3
9. Exit

End

La convergenza è sempre assicurata, ma è relativamente lenta (**convergenza lineare**).

Questo metodo è solitamente usato per calcolare una stima di prima approssimazione della radice (con 1 o 2 decimali corretti) che verrà successivamente raffinata con un metodo più veloce (ad esempio Newton-Raphson e secanti, con ordine di convergenza >1).

Il metodo di bisezione non può essere esteso al caso di sistemi di equazioni non lineari.

GENERALITÀ SU "REGULA FALSI", METODI DELLE TANGENTI E DELLE SECANTI

- Vengono tutti costruiti a partire da **una o più approssimazioni iniziali** della radice cercata.

- Ad ogni passo del processo iterativo **si approssima localmente con una retta il problema iniziale** $f(x)=0$ e come nuova approssimazione della radice cercata si considera l'intersezione della retta approssimante con l'asse x delle ascisse. Questo corrisponde a considerare la soluzione dell'equazione (lineare)

$$y_n + k_n(x - x_n) = 0 \quad \text{ossia} \quad f(x_n) + k_n \cdot (x - x_n) = 0 \qquad n=0, 1, \ldots$$

  La nuova approssimazione risulta quindi pari a

$$x_{n+1} = x_n - \frac{y_n}{k_n} \ \text{cioè} \ x_{n+1} = x_n - \frac{f(x_n)}{k_n} \qquad n=0, 1, \ldots$$

- A seconda di come viene scelto il coefficiente angolare k_n della retta approssimante si individuano i diversi metodi.

"REGULA FALSI"

A partire da **una approssimazione iniziale** x_0 della radice cercata, il coefficiente angolare della retta approssimante si sceglie pari a

$$k_n = \frac{y_n - f(x_0)}{x_n - x_0} \text{ cioè } k_n = \frac{f(x_n) - f(x_0)}{x_n - x_0} \qquad n=0, 1, ...$$

Si applica a **funzioni continue** di cui si conosce un **intervallo iniziale** [a, b] che ne contiene una radice ($f(a)\, f(b) < 0$). A partire dall'approssimazione iniziale x_0 si calcola la **successione**

$$x_{n+1} = x_n - f(x_n) \cdot \frac{(x_n - x_0)}{f(x_n) - f(x_0)} \qquad n=0, 1, ... \text{ fino a } |f(x_{n+1})| \text{ sufficientemente piccolo.}$$

E' un raffinamento del metodo di bisezione perché sfrutta le proprietà analitiche della funzione; ha **convergenza lineare**.

Può anche essere formalizzato come un metodo di bisezione, con una particolare regola per la scelta del punto m_k.

- è noto l'intervallo iniziale [a, b], si pone $k = 0$ e si calcola il punto $x_0 = \frac{a \cdot f(b) - b \cdot f(a)}{f(b) - f(a)}$, intersezione tra la retta per i punti (a, $f(a)$) , (b, $f(b)$) e l'asse x delle ascisse
- si procede come nel metodo di bisezione
  - si valuta il segno della $f(x)\, f(b)$ in x_0:
    - se $f(x_0)\, f(b) > 0 \Rightarrow$ nuovo intervallo d'interesse il semi-intervallo di sinistra [a, x_0]
    - se $f(x_0)\, f(b) < 0 \Rightarrow$ nuovo intervallo d'interesse il semi-intervallo di destra [x_0, b]
    - se $f(x_0)\, f(b) = 0 \Rightarrow x_0$ è la radice cercata
  - se $f(x_0)\, f(b) \neq 0$ si prende il nuovo sotto-intervallo e si ricomincia il procedimento ($k = k + 1$).
- Al passo iterativo n- il punto x_n (viene scelto quello più vicino all'estremo in cui la funzione $f(x)$ assume valore minore in modulo) è la stima della radice cercata.

METODO DELLE TANGENTI (NEWTON-RAPHSON)

Nota una approssimazione iniziale x_0 della radice cercata, il coefficiente angolare della retta approssimante si sceglie pari a

$$k_n = f'(x_n) \qquad \text{per ogni } n$$

Si applica a **funzioni derivabili**; dall'approssimazione iniziale x_0 si calcola la **successione**

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \qquad n=0, 1, ... \qquad \text{fino a } |f(x_{n+1})| \text{ sufficientemente piccolo.}$$

E' **indispensabile** che la funzione sia derivabile.

Ha **convergenza quadratica** (ordine 2), ma se la $f'(x_n)$ è piccola il **metodo può divergere**.

METODO DELLE SECANTI

Note due approssimazioni iniziali x_0 e x_1 della radice cercata, il coefficiente angolare della retta approssimante si sceglie pari a

$$k_n = \frac{y_n - y_{n-1}}{x_n - x_{n-1}} \text{ cioè } k_n = \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}} \qquad n = 0, 1, \ldots$$

Si applica a **funzioni continue**; dalle approssimazioni iniziali x_0 e x_1 si calcola la **successione**

$$x_{n+1} = x_n - f(x_n) \cdot \frac{(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})} \qquad n = 0, 1, \ldots \text{ fino a } |f(x_{n+1})| \text{ sufficientemente piccolo.}$$

E' una modifica del metodo delle tangenti: la derivata prima $f'(x_n)$ richiesta nel metodo delle tangenti viene sostituita con il **rapporto incrementale** $\frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$

Ha ordine di **convergenza 1.618**. E' applicabile anche se la funzione non è derivabile.

Algoritmo. Secant (x_0, x_1, f, x_{toll}, f_{toll}, $n_{\max}$, x, ier)

Commento. Dato l'intervallo $[x_0, x_1]$ contenente una radice semplice dell'equazione $f(x)=0$, con $f(x_0)\ f(x_1) < 0$, si calcola una successione di approssimazioni della radice. Quando due approssimazioni successive x_n e x_{n+1} sono tali che $|x_n - x_{n-1}| \le x_{\text{toll}} |x_{n+1}|$ e $|f(x_{n+1})| \le f_{\text{toll}}$, l'algoritmo si arresta, pone x_{n+1} in x e definisce ier=0. Se la precisione richiesta non viene raggiunta con la $n$max iterazioni prefissate, l'algoritmo pone in x l'ultima approssimazione trovata e definisce ier=1 oppure ier=2 a seconda che il test non soddisfatto sia quello sulla x_{n+1} oppure quello sulla $f(x_{n+1})$.
Alla fine la variabile $n$max contiene il numero di iterazioni eseguite.
Se $f(x_0)\ f(x_1) > 0$ l'algoritmo non procede oltre e pone ier= −1.

Parametri. Input: x_0, x_1, f, x_{toll}, f_{toll}, $n_{\max}$
Output: $n_{\max}$, x, ier

1. se $f(x_0)\ f(x_1) > 0$, ier=−1; Exit
2. Ciclo 1: n=1, ..., $n_{\max}$
3. $x_{n+1} = x_n - f(x_n) \cdot \frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}$
4. se $|x_n - x_{n+1}| > x_{\text{toll}} \cdot |x_{n+1}|$ oppure $|x_n - x_{n+1}| > x_{\text{toll}}$ poni ier=1 e vai al punto 10
5. se $|f(x_{n+1})| > f_{\text{toll}}$ poni ier=2 e vai al punto 10
6. ier=0
7. $x = x_{n+1}$
8. $n_{\max} = n$
9. Exit
10. Fine Ciclo 1
11. $x = x_{n+1}$
12. Exit

End

CONFRONTO TRA I DIVERSI METODI

Bisezione e "Regula Falsi" sono **metodi globali**; sono metodi **chiusi**, cioè è sufficiente localizzare la radice o meglio un intervallo iniziale che la contiene per garantire la convergenza.

Tangenti e Secanti sono **metodi locali**; sono metodi **aperti** in quanto non esiste un intervallo prefissato di partenza e richiedono la conoscenza di una o due approssimazioni iniziali nel dominio d'attrazione della radice. Il metodo delle tangenti richiede anche la conoscenza della funzione derivata $f'(x)$.

La **scelta operativa** che viene solitamente effettuata consiste nell'utilizzo di un metodo globale per avvicinarsi alla radice cercata e quindi l'utilizzo di un metodo locale per il successivo raffinamento.

	APPLICABILITA'	CONVERGENZA
BISEZIONE	$f(x)$ continua [a, b] noto $f(a)\,f(b) < 0$	$p = 1$ sempre
"REGULA FALSI"	$f(x)$ continua [a, b] noto $f(a)\,f(b) < 0$	$p = 1$ sempre
TANGENTI	$f(x)$ continua e derivabile 1 approx. iniziale x_0 nota	$p = 2$ soltanto se x_0 vicina alla radice
SECANTI	$f(x)$ continua 2 approx. iniziali x_0 e x_1 note	$p = 1.618$ soltanto se x_0 e x_1 vicine alla radice

CRITERI DI ARRESTO

A seconda del valore della derivata di $f(x)$ nella radice x_∞, si adotta un diverso criterio di arresto della successione di approssimazioni alla radice cercata.

- Se $|f'(x_\infty)| >> 1$ conviene verificare la convergenza di $|f(x_n)|$. Il criterio di arresto è del tipo

$$|f(x_n)| < f_{\text{toll}}$$

  Ci possono essere problemi nel caso di funzioni particolarmente "piatte" o "ripide".

- Se $|f'(x_\infty)| << 1$, conviene verificare la convergenza di x_n. Il criterio di arresto è del tipo

$$|x_n - x_{n-1}| < x_{\text{toll}} \text{ oppure } |x_n - x_{n-1}| < x_{\text{toll}} \cdot |x_n|$$

  Ci possono essere problemi nel caso di convergenza soltanto lineare.

Se non si conosce l'ordine di grandezza di $|f'(x_\infty)|$ conviene utilizzare entrambi i criteri.

ESERCIZI SVOLTI

Risolvere l'equazione non lineare $x^2=2$ con un errore assoluto sulla soluzione inferiore a $5\cdot10^{-2}$ utilizzando i metodi di bisezione, "regula falsi", delle tangenti e delle secanti.
Adottare come intervallo di partenza [1, 2] per i primi due metodi.
Adottare un'approssimazione iniziale pari a 1 per il metodo di Newton.
Adottare due approssimazioni iniziali pari rispettivamente a 1 e 1.5 per il metodo delle secanti.
Confrontare il numero di iterazioni richieste da ciascun metodo.

Metodo di Bisezione:

$$f(x) = x^2-2 = 0 \qquad \text{con } [a, b]=[1, 2]$$

$$m_1 = \frac{a+b}{2} = \frac{1+2}{2} = 1.5$$

$$\operatorname{sgn}\left(f\left(m_1\right)\right) = \operatorname{sgn}\left(1.5^2-2\right) = 2.5\cdot10^{-1} > 0 \Rightarrow a=1 \text{ e } b=1.5$$

$$m_2 = \frac{a+b}{2} = \frac{1+1.5}{2} = 1.25$$

$$\operatorname{sgn}\left(f\left(m_2\right)\right) = \operatorname{sgn}\left(1.25^2-2\right) = -4.37\ldots\cdot10^{-1} < 0 \Rightarrow a=1.25 \text{ e } b= 1.5$$

$$m_3 = \frac{a+b}{2} = \frac{1.25+1.5}{2} = 1.375$$

$$\operatorname{sgn}\left(f\left(m_3\right)\right) = \operatorname{sgn}\left(1.375^2-2\right) = -1.09\ldots\cdot10^{-1} < 0 \Rightarrow a=1.375 \text{ e } b= 1.5$$

$$m_4 = \frac{a+b}{2} = \frac{1.375+1.5}{2} = 1.4375$$

$$\operatorname{sgn}\left(f\left(m_4\right)\right) = \operatorname{sgn}\left(1.4375^2-2\right) = 6.64\ldots\cdot10^{-2} > 0 \Rightarrow a=1.375 \text{ e } b= 1.4375$$

$$m_5 = \frac{a+b}{2} = \frac{1.375+1.4375}{2} = 1.40625$$

$\Rightarrow$ EA$\approx 3.1\cdot10^{-2} < 5\cdot10^{-2}$, radice $\cong m_5 = 1.40625$

Regula Falsi:

$$f(x)=x^2-2=0 \qquad \text{con } [a, b] = [1, 2]$$

$$x_0 = \frac{a\cdot f(b)-b\cdot f(a)}{f(b)-f(a)} = \frac{1\cdot 2-2\cdot(-1)}{2-(-1)} = 1.\overline{3}$$

$$\operatorname{sgn}\left(f\left(x_0\right)\right) = \operatorname{sgn}\left(1.\overline{3}^2-2\right) = -2.2\overline{2}\cdot10^{-1} < 0 \Rightarrow a=1.\overline{3} \text{ e } b=2$$

$$x_1 = \frac{a\cdot f(b)-b\cdot f(a)}{f(b)-f(a)} = \frac{1.\overline{3}\cdot 2-2\cdot\left(-0.2\overline{2}\right)}{2-\left(-2\overline{2}\right)} = 1.40\ldots$$

$$\operatorname{sgn}\left(f\left(x_1\right)\right) = \operatorname{sgn}\left(1.40\ldots^2-2\right) = -3.99\ldots\cdot10^{-2} < 0 \Rightarrow a=1.40\ldots \text{ e } b=2$$

$$x_2 = \frac{a\cdot f(b)-b\cdot f(a)}{f(b)-f(a)} = \frac{1.40\ldots\cdot 2-2\cdot(-0.039\ldots)}{2-(-0.039)} = 1.4117\ldots$$

$\Rightarrow$ EA$\approx 1.1\cdot10^{-2} < 5\cdot10^{-2}$, radice $\cong x_2 = 1.4117\ldots$

Metodo delle Tangenti (Newton-Raphson):

$$f(x) = x^2 - 2 = 0 \qquad \text{con } x_0 = 1$$

$f'(x) = 2x$

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 1 - \frac{-1}{2} = 1.5 \Rightarrow f(x_1) = 2.5 \cdot 10^{-1}$$

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} = 1.5 - \frac{0.25}{3} = 1.4166\ldots \Rightarrow f(x_2) = 6.9\ldots \cdot 10^{-3}$$

$$x_3 = x_2 - \frac{f(x_2)}{f'(x_2)} = 1.4166\ldots - \frac{6.94445 \cdot 10^{-3}}{2.8334} = 1.4142\ldots$$

$\Rightarrow$ EA$\approx 2.5 \cdot 10^{-2} < 5 \cdot 10^{-2}$, radice $\cong x_3 = 1.4142\ldots$

Metodo delle Secanti:

$$f(x) = x^2 - 2 = 0 \qquad \text{con } x_0 = 1 \text{ e } x_1 = 1.5$$

$$x_2 = x_1 - f(x_1) \cdot \frac{(x_1 - x_0)}{f(x_1) - f(x_0)} = 1.5 - 0.25\frac{1.5 - 1}{0.25 + 1} = 1.4$$

$$x_3 = x_2 - f(x_2) \cdot \frac{(x_2 - x_1)}{f(x_2) - f(x_1)} = 1.4 + 0.04\frac{1.4 - 1.5}{-0.04 - 0.25} = 1.4137\ldots$$

$\Rightarrow$ EA$\approx 1.4 \cdot 10^{-2} < 5 \cdot 10^{-2} \Rightarrow$ radice $\cong x_2 = 1.4137\ldots$

ESERCIZI PROPOSTI

Risolvere l'equazione non lineare $x^3 - 3x - 1 = 0$ (intervallo [1, 2], approssimazioni 1, 1 e 2).

[1.8794]

Risolvere l'equazione non lineare $\cos(x) - x = 0$ (intervallo [0, 1], approssimazioni 0, 0 e 1).

[0.7391]

Risolvere l'equazione non lineare $e^{-x^2} - x = 0$ (intervallo [0, 1], approssimazioni 0, 0 e 1).

[0.6529]

Risolvere l'equazione non lineare $\cosh(x) = 2$ (intervallo [1, 2], approssimazioni 1, 1 e 2).

[1.3170]

Risolvere l'equazione non lineare $\sin h(x) + x = 3$ (intervallo [1, 2], approssimazioni 1, 1 e 2).

[1.3005]

Risolvere l'equazione non lineare $\sin(x) = 1 - x$ (intervallo [0, 1], approssimazioni 0, 0 e 1).

[0.5110]

Risolvere l'equazione non lineare $x\tan(x) - 1 = 0$ (intervallo [0, 1], approssimazioni 0, 0 e 1).

[0.8603]

CALCOLO DI INTEGRALI MEDIANTE FORMULE DI QUADRATURA

Riferimento al testo: Cap. VII

INTRODUZIONE

L'integrale di una funzione $f(x)$, nota in n punti (nodi) distinti x_i con $i=1,\ldots, n$ appartenenti all'intervallo [a,b] in cui essa è definita, può essere calcolato in maniera approssimata per mezzo delle cosiddette formule di quadratura.

La tecnica consiste nell'interpretare l'integrale come l'area sottesa dalla funzione nell'intervallo [a, b] e nel calcolarne il valore avendo preventivamente interpolato gli n punti noti con un polinomio di grado (n-1). L'integrale viene quindi approssimato con una sommatoria:

$$\int_a^b f(x)dx \cong \sum_{i=1}^{n} w_i f(x_i)$$

dove i numeri reali x_i e w_i sono, rispettivamente, i **nodi** e i **pesi** della formula di quadratura.

Il significato dei nodi e dei pesi discende dalla procedura di interpolazione degli n punti in cui è nota la funzione $f(x)$ che viene approssimata con la funzione interpolante

$$f(x) \cong L_{n-1}(x) = \sum_{i=1}^{n} f(x_i) l_i(x)$$

in cui sono riconoscibili i polinomi fondamentali di Lagrange $l_i(x)$:

$$l_i(x) = \frac{\prod_{j=0,\, j\neq i}^{n} (x - x_j)}{\prod_{j=0,\, j\neq i}^{n} (x_i - x_j)}$$

L'integrale viene calcolato dunque come

$$\int_a^b f(x)dx = \int_a^b [L_{n-1}(x) + E_n(x)]dx = \int_a^b L_{n-1}(x)dx + \int_a^b E_n(x)dx = \sum_{i=1}^{n} \left(\int_a^b l_i(x)dx \right) \cdot f(x_i) + R_n(f)$$

In primo termine è l'approssimazione dell'integrale ottenuta con la formula di quadratura, il secondo è l'errore commesso, o resto. Nel primo termine i pesi sono gli integrali delle funzioni di Lagrange:

$$w_i = \int_a^b l_i(x)\,dx$$

Le formule di quadratura vengono generalmente costruite facendo riferimento a un intervallo normalizzato [-1, 1] e poi vengono estese al caso di un generico intervallo [a, b] utilizzando il semplice cambiamento di variabili

$$\int_a^b f(x)dx = \int_{-1}^{1} g(t)dt \cong \sum_{i=1}^{n} w_i \cdot g(t_i)$$

$$t = a \cdot x + b \; t \in [-1, 1] \Rightarrow \qquad \begin{cases} x = a & t = -1 \\ x = b & t = 1 \end{cases}$$

da cui

$$a = \frac{2}{b-a} \quad b = -\frac{a+b}{b-a} \quad t = \frac{2}{b-a}x - \frac{a+b}{b-a}; \quad x = \frac{b-a}{2}t + \frac{a+b}{2}$$

e quindi

$$\int_a^b f(x)dx = \frac{b-a}{2}\int_{-1}^{1} f\left(\frac{b-a}{2}t + \frac{b+a}{2}\right)dt \cong \frac{b-a}{2}\sum_{i=1}^{n} w_i \cdot f\left(\frac{b-a}{2}t_i + \frac{b+a}{2}\right)$$

Se la funzione integranda, o una delle sue prime derivate, presenta irregolarità o singolarità, per effettuarne l'integrazione può essere conveniente applicare una **fattorizzazione** che evidenzi due contributi, moltiplicati tra loro, di cui uno regolare $g(x)$ e uno irregolare, ma possibilmente di forma semplice, $p(x)$.
La funzione irregolare $p(x)$ entra nei pesi della formula di quadratura, purché sia garantita l'esistenza degli integrali che definiscono tali pesi

$$\int_a^b f(x)dx = \int_a^b p(x)\sum_{i=1}^{n} l_i(x)\, g(x_i)dx = \sum_{i=1}^{n} w_i g(x_i)$$

In generale, si faccia riferimento a formule interpolatorie pesate del tipo

$$\int_a^b w(x) \cdot f(x)dx = \sum_{i=1}^{n} w_i \cdot f(x_i) + R_n(f)$$

se la funzione integranda $f(x)$ è un polinomio di grado $\leq (n\text{-}1)$, per gli n nodi distinti passa uno e un solo polinomio di interpolazione e il resto $R_n(f)$ è nullo, cioè la formula di quadratura costruita ha grado di precisione almeno (n-1).

I nodi (distinti) x_i all'interno dell'intervallo [a, b] vengono solitamente scelti in due modi:

- nodi equidistanti $\Rightarrow$ Formule di Newton-Cotes (grado di precisione (n-1) o n a seconda che n sia pari o dispari)
- nodi coincidenti con gli zeri di particolari polinomi (polinomi ortogonali) $\Rightarrow$ Formule Gaussiane (grado di precisione massimo (2n-1))

FORMULE DI QUADRATURA DI BASE (NEWTON-COTES)

Nell'intervallo [*a*, *b*] si prendono n nodi equidistanti $x_i = a + h \cdot (i-1)$. La formula di quadratura è ricavata nella forma

$$\int_a^b f(x)dx = \sum_{i=1}^{n} w_i \cdot f(a + h \cdot (i-1)) + R_n(f)$$

dove l'ampiezza di ciascun sottointervallo è $h = (b-a)/(n-1)$ e i coefficienti dell'interpolazione $c_i = w_i/(b-a)$ sono i numeri di Cotes.

Le formule di quadratura di base più semplici e più utilizzate sono quelle interpolatorie di grado 0, 1 e 2. Ad esse può essere associata un un'interpretazione di tipo geometrico. Nel caso di forme interpolatorie di grado 0, l'integrale di $f(x)$ nell'intervallo [*a*, *b*] è approssimato come l'area del rettangolo di base h_i e di altezza $f(a)$ (o, come variante, come area del rettangolo di base h_i ed altezza $f(m)$, dove m è il punto medio dell'intervallo [*a*,*b*]). Nel caso di forme interpolatorie di grado 1, l'integrale è approssimato con l'area del trapezio ottenibile congiungendo i quattro vertici *a*, *b*, $f(a)$, $f(b)$.

Formula del rettangolo (grado 0):

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

Formula del punto medio (grado 0):

$$\int_a^b f(x)dx \cong (b-a) f(m) \quad , \qquad m = \frac{a+b}{2}$$

Formula dei trapezi (grado 1):

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

Formula di Simpson (grado 2):

$$\int_a^b f(x)dx \cong \frac{b-a}{6}\left[f(a) + 4f(m) + f(b)\right] \qquad , \qquad m = \frac{a+b}{2}$$

I due ultimi casi citati differiscono nel fatto che l'interpolazione dei due nodi consecutivi viene effettuata nella formula dei trapezi con un polinomio di primo grado per i due punti $(a,f(a))$ e $(b,f(b))$, mentre nella formula di Simpson con un polinomio di secondo grado che interpola i punti $(a,f(a))$, $(m,f(m))$ e $(b,f(b))$.

FORMULE DI QUADRATURA COMPOSTE

L'aumento del numero di nodi per il calcolo dell'integrale porta ad un polinomio interpolante di grado sempre più elevato. In precedenza abbiamo già constatato come l'aumento indiscriminato del grado del polinomio interpolante non sia una buona tecnica per migliorare le sue proprietà approssimanti.

Per tale motivo si passa da formule di quadratura su tutto l'intervallo a quelle composte, applicate localmente su sottointervalli di $[a, b]$.

Una volta assegnata una formula di quadratura di base per effettuare l'integrazione della funzione $f(x)$ si opera una suddivisione dell'intervallo $[a, b]$ in N sottointervalli di ampiezza generalmente uguale per ottenere una migliore approssimazione dell'integrale.

A questo fine si pone $x_i = a + i\,h = a + i\,(b-a)/N$, $i = 0, ..., N$ e $m_i = (x_{i-1} + x_i)/2$, $i = 1, ..., N$.

Le precedenti formule di quadratura, una volta espresse sui diversi sottointervalli, assumono la seguente forma di formule composte:

Formula del punto medio (grado 0):

$$M_N = \int_a^b f(x)dx \cong \frac{b-a}{N}\sum_{i=1}^{N} f(m_i)$$

Formula dei trapezi (grado 1):

$$T_N = \int_a^b f(x)dx \cong \frac{b-a}{N}\left[\frac{1}{2}f(a) + \sum_{i=1}^{N-1} f(x_i) + \frac{1}{2}f(b)\right]$$

Formula di Simpson (grado 2):

$$S_N = \int_a^b f(x)dx \cong \frac{b-a}{N}\left[\frac{1}{6}f(a) + \frac{2}{6}\sum_{i=1}^{N-1} f(x_i) + \frac{4}{6}\sum_{i=1}^{N} f(m_i) + \frac{1}{6}f(b)\right]$$

$$S_N = \frac{2}{3}M_N + \frac{1}{3}T_N$$

Si può dimostrare che l'errore delle formule di quadratura composte è $O(N^2)$ per la formula trapezoidale e $O(N^4)$ per la formula di Simpson.

Esempio

Calcolare le approssimazioni M_N , T_N , S_N e valutare l'errore relativo (e_M, e_T, e_S , rispettivamente) di ciascuna rispetto al valore vero del seguente integrale :

$$V = \int_0^1 e^{-x}dx = 1 - \frac{1}{e}\text{ , N = 5}$$

Il valore V dell'integrale è: $V = \int_0^1 e^{-x}\,dx = 1 - \frac{1}{e} = 0.6321205588...$

L'intervallo [0,1] è suddiviso in 5 sottointervalli:

x	e^{-x}
$x_0 = 0$ $m_1 = 0.1$	$f(x_0) = 1$ $f(m_1) = 0.90484$
$x_1 = 0.2$ $m_2 = 0.3$	$f(x_1) = 0.81873$ $f(m_2) = 0.74082$
$x_2 = 0.4$ $m_3 = 0.5$	$f(x_2) = 0.67032$ $f(m_3) = 0.60653$
$x_3 = 0.6$ $m_4 = 0.7$	$f(x_3) = 0.54881$ $f(m_4) = 0.49659$
$x_4 = 0.8$ $m_5 = 0.9$	$f(x_4) = 0.44933$ $f(m_5) = 0.40657$
$x_5 = 1.0$	$f(x_5) = 0.36788$

$$M_5 = \frac{b-a}{N}\sum_{i=1}^{N} f(m_i) = \frac{1}{5}\sum_{1}^{5} f\left(\frac{x_{i-1}+x_i}{2}\right) = 0.2\begin{bmatrix}0.90484 + 0.74082 + 0.60653 + \\ +0.49659 + 0.40657\end{bmatrix} = 0.63107$$

$$e_M = \left(V - M_5\right)/V = 0.0017$$

$$T_5 = \frac{b-a}{N}\left(\frac{f(a)}{2} + \sum_{i=1}^{N-1} f(x_i) + \frac{f(b)}{2}\right) = 0.2[0.5 + 0.81873 + 0.67032 + 0.54881 + 0.44933 + 0.1839] = 0.634226$$

$$e_T = \left(V - T_5\right)/V = -0.0033$$

$$S_5 = \frac{b-a}{N}\left(\frac{f(a)}{6} + \frac{2}{6}\sum_{i=1}^{N-1} f(x_i) + \frac{4}{6}\sum_{i=1}^{N} f(m_i) + \frac{f(b)}{6}\right) = 0.2\begin{bmatrix}\frac{1}{6} + \frac{2}{6}(0.81873 + 0.67032 + 0.54881 + 0.44933) + \\ \frac{4}{6}(0.90484 + 0.74082 + 0.60653 + 0.49659 + 0.40657) + \\ 0.06131\end{bmatrix} =$$

$= 0.632122$

$$e_S = \left(V - S_5\right)/V = 2.3\,10^{-6}$$

POLINOMI ORTOGONALI

Dato l'intervallo [a, b] sia assegnata una funzione peso $w(x)$ non negativa e non identicamente nulla su [a,b] tale che esistano tutti i momenti

$$m_k = \int_a^b w(x)\cdot x^k dx \qquad \text{con } k = 0, 1, 2, ...$$

Un sistema di polinomi $\{P_0(x), P_1(x), ..., P_n(x), ...\}$ con $P_n(x) = k_{n,0}\cdot x^n + k_{n,1}\cdot x^{n-1} + ... + k_{n,n}$ e $k_{n,0} \neq 0$ è detto **ortogonale** in [a, b] rispetto alla funzione peso $w(x)$ se

$$\int_a^b w(x)\cdot P_n(x)\cdot P_m(x)dx = 0 \quad \text{per } n \neq m \quad \text{e} \quad \int_a^b w(x)\cdot P_n(x)\cdot P_m(x)dx \neq 0 \quad \text{per } n = m.$$

I polinomi ortogonali possiedono le seguenti proprietà:

1. per ogni $n \geq 1$ il polinomio $P_n(x)$ ha n zeri reali, distinti e tutti contenuti in [a, b]; tra due zeri consecutivi di $P_n(x)$ esiste un solo zero di $P_{n-1}(x)$.
2. Per ogni polinomio $q(x)$ di grado $\leq (n-1)$ si ha:

$$\int_a^b w(x)\cdot P_n(x)\cdot q(x)dx = 0$$

In particolare la relazione:

$$\int_a^b w(x)\cdot P_n(x)\cdot x^k dx = 0 \qquad \text{con } k = 0, 1, 2, ...$$

definisce univocamente, a meno di una costante moltiplicativa, il polinomio ortogonale $P_n(x)$.

I **polinomi ortogonali classici** hanno le seguenti funzioni peso e intervalli di definizione:

- polinomi di Legendre: $w(x)=1$, [a, b]= [-1, 1]
- polinomi di Jacobi: $w(x) = (1-x)^{\alpha}\cdot(1+x)^{\beta}$ con $\alpha,\beta > -1$, [a, b]= [-1, 1]
  - se $\alpha=\beta \neq -1/2 \Rightarrow$ polinomi ultrasferici
  - se $\alpha=\beta = -1/2 \Rightarrow$ polinomi di Chebyshev di prima specie
  - se $\alpha=\beta = 1/2 \Rightarrow$ polinomi di Chebyshev di seconda specie
- polinomi di Laguerre: $w(x) = e^{-x}$, [a, b]= [0, ∞]
- polinomi di Hermite: $w(x) = e^{-x^2}$, [a, b]= [-∞, ∞]

FORMULE DI QUADRATURA GAUSSIANE

Si supponga che la funzione peso $w(x)$ soddisfi le seguenti ipotesi:

1. $w(x) \neq 0$ e $w(x) \geq 0$ nell'intervallo $[a, b]$
2. esistano tutti i momenti

$$m_k = \int_a^b w(x) \cdot x^k dx \qquad \text{con } k = 0, 1, 2, \ldots$$

Condizione necessaria e sufficiente perché la formula di quadratura

$$\int_a^b w(x) \cdot f(x) dx = \sum_{i=1}^{n} w_i \cdot f(x_i) + R_n(f)$$

sia Gaussiana, e quindi abbia grado di precisione massimo pari a $(2n\text{-}1)$, è che essa sia di tipo interpolatorio e che i nodi $\{x_i\}$ coincidano con gli n zeri del polinomio $P_n(x)$, di grado n, ortogonale nell'intervallo $[a, b]$ rispetto alla funzione peso $w(x)$.

Le formule Gaussiane più utilizzate sono quelle associate ai polinomi ortogonali classici e precisamente: le formule di Gauss-Legendre, di Gauss-Jacobi, di Gauss-Laguerre e di Gauss-Hermite.

FORMULE DI QUADRATURA AUTOMATICA

Le routines per il calcolo degli integrali mediante formule di quadratura seguono due strategie differenti in relazione alla scelta del passo di integrazione (corrispondente all'ampiezza dei sottointervalli):

- non adattativa (sottointervalli di uguale ampiezza): per funzioni molto regolari
- adattativa: per situazioni in cui esistono punti di singolarità nella funzione integranda si addensano i nodi nell'intorno di questa, variando localmente il passo di integrazione.

La variazione del passo di integrazione è regolata sulla base della stima dell'errore. Un algoritmo di integrazione numerica basato sulle formule di quadratura presenta infatti, per ogni ordine N , un errore:

$$E_N = \int_a^b f(x) dx - I_N = \int_a^b f(x) dx - \sum_{i=1}^{N} w_i f(x_i)$$

Se tale errore è $E_N = O(N^{-p})$ (ove p è specifico per il tipo di formula di quadratura che si sta utilizzando) , è possibile stimarlo calcolando la formula di quadratura di ordine $2N$.
Infatti, poichè $I_N + E_N = I_{2N} + E_{2N}$ e, con buona approssimazione quando N è abbastanza grande, $E_{2N} = 2^{-P} E_N$, è immediato ottenere

$$E_{2N} \cong \frac{I_{2N} - I_N}{2^P - 1}$$

Questa stima dell'errore viene utilizzata per definire un criterio di arresto nella generazione in successione delle approssimazioni $I_N, I_{2N}, I_{4N}, \ldots$

Si osservi che, ad ogni raddoppio dell'ordine della quadratura, non è necessario ricalcolare la funzione in tutti i nodi della discretizzazione: si possono utilizzare i valori della funzione integranda calcolati nei passi precedenti mentre è necessario calcolare la $f(x)$ soltanto in corrispondenza dei nuovi nodi introdotti.

Poichè spesso accade che le funzioni integrande abbiano un comportamento non uniforme sull'intervallo di integrazione, spesso si adottano strategie di quadratura di tipo adattativo che aumentano il numero dei nodi utilizzati solo nelle zone di peggiore uniformità della funzione integranda. Questo consente di limitare il numero di calcoli per valutare la $f(x)$ nelle zone che invece presentano una buona regolarità.

ESERIZI PROPOSTI

Calcolare le approssimazioni M_N , T_N , S_N e valutare l'errore relativo (e_M, e_T, e_S , rispettivamente) di ciascuna rispetto al valore vero del seguente integrale :

1. $\int_0^p sin(x)dx = 2$, N = 4

[M_N = 2.05234 , T_N = 1.89612 , S_N = 2.00456 , $e_M = 2.6 \cdot 10^{-2}, e_T = -5.2 \cdot 10^{-2}, e_S = 2.3 \cdot 10^{-3}$]

2. $\int_0^p sin(x)dx = 2$, N = 8

[M_N = 2.00825 , T_N = 1.98352 , S_N = 2.00011 , $e_M = 4.1 \cdot 10^{-3}, e_T = -8.2 \cdot 10^{-3}, e_S = 5.5 \cdot 10^{-5}$]

3. $\int_1^2 \frac{dx}{x} = \ln 2$, N = 8

[M_N = 0.69122 , T_N = 0.69702 , S_N = 0.69325 , $e_M = -2.8 \cdot 10^{-3}, e_T = 5.6 \cdot 10^{-3}, e_S = 1.5 \cdot 10^{-4}$]

4. $\int_1^2 x^2 dx = \frac{7}{3}$, N = 4

[M_N = 2.32813 , T_N = 2.34375 , S_N = 2.33333 , $e_M = -2.2 \cdot 10^{-3}, e_T = 4.5 \cdot 10^{-3}, e_S = 0$]

EQUAZIONI DIFFERENZIALI ORDINARIE (ODE)

Riferimento al testo: Cap. VIII

INTRODUZIONE

La quasi totalità dei fenomeni fisici sono descritti attraverso modelli matematici che conducono alla formulazione di equazioni differenziali. Il caso più semplice è quello dell'equazione differenziale del primo ordine che, associata alla condizione al contorno, viene indicata con il nome di **problema di Cauchy**:

$$y'(x) = f(x, y(x)) \quad , \quad x \in [a, b] \quad y(a) = y_o$$

Questo problema è stato diffusamente studiato in letteratura e se ne conoscono le soluzioni.
L'estensione di questo caso più semplice a quello di più funzioni porta alla formulazione del problema nei seguenti termini:

$$y'_i(x) = f_i(x, y_1(x), ..., y_m(x)) \qquad \text{con } i = 1, \ldots, n \text{ e } x \in [a, b]$$

ove le condizioni al contorno sono date da:

$$y_i(a) = y_{i,0} \qquad i = 1, ..., n$$

che rappresentano ancora un problema differenziale del primo ordine, con più gradi di libertà.

Esistono anche equazioni differenziali di ordine maggiore al primo, che peraltro possono essere ricondotte al primo grado operando un'opportuna sostituzione di variabili. In particolare un'equazione differenziale del secondo ordine del tipo:

$$y'' = f(x, y, y')$$

con condizioni iniziali y_0 , y'_0 può essere scritta come segue:

$$z'_1 = z_2 \quad , \quad z'_2 = f(x, z_1, z_2)$$

avendo operato la sostituzione di variabili:

$$z_1 = y \qquad \text{e} \qquad z_2 = y'$$

In base a tale osservazione nel seguito si prenderanno principalmente in considerazione i problemi con equazioni differenziali del primo ordine, che possono essere scritti in forma vettoriale

$$\mathbf{y}'(x) = \mathbf{f}(x, \mathbf{y}(x)) \quad , \quad x \in [a, b]$$

con la condizione al contorno $\mathbf{y}(a) = \mathbf{y_0}$.

Un'ulteriore generalizzazione del problema consiste nella risoluzione di problemi con equazioni differenziali in più variabili alle derivate parziali, che peraltro non saranno oggetto della presente trattazione e che comportano una maggiore difficoltà sia dal punto di vista analitico che numerico.

SOLUZIONE NUMERICA DI ODE

Per procedere alla soluzione numerica di problemi di equazioni differenziali ordinarie si applica in genere una discretizzazione dell'intervallo $[a\ ,\ b]$, nel quale il problema è definito, in N sottointervalli, descritti dalla successione di ascisse $x_i = a + i \cdot \dfrac{b-a}{N}$ con $i = 0\ ,\ ...,\ N$.

La soluzione del problema consiste nell'approssimazione dei valori $y(x_i)$ assunti dalla funzione $y(x)$, argomento delle equazioni differenziali, nelle ascisse x_i.

E' necessario peraltro un criterio per stabilire a priori se il problema in esame ammette soluzione e se tale soluzione è unica. Tale verifica deve necessariamente precedere il calcolo della soluzione.

CONDIZIONE DI LIPSCHITZ

Esiste un teorema di esistenza e unicità globale delle soluzioni del problema ad equazioni differenziali ordinarie, noto come condizione di Lipschitz, così formulato:

- Sia $f(x)$ definita e continua nella striscia $S = \left\{(x, y) : -\infty < \alpha \le x \le \beta,\ y \in \Re^m \right\}$.
- Sia inoltre $f(x, y)$ lipschitziana nella variabile y, cioè esiste una costante L > 0 tale che

$$\left\| f(x, y_1) - f(x, y_2) \right\| \le L \cdot \left\| y_1 - y_2 \right\| \quad \text{(condizione di Lipschitz)}$$

per ogni $x \in [\alpha, \beta]$ e per ogni coppia $y_1, y_2 \in \Re^m$.

$\Rightarrow$ Allora per ogni $a \in [\alpha, \beta]$ e per ogni $y_0 \in \Re^m$ esiste esattamente una funzione $y(x)$ tale che

1. $y(x) \in C^1(a\ ,\ b\ )$ (la funzione $y(x)$ è continua, derivabile e con derivata prima continua in $[\alpha, \beta]$)
2. $y'(x) = f(x, y(x)) \quad \forall x \in [a\ ,\ b\ ]$
3. $y(a) = y_0$

Nel seguito si assumerà sempre l'ipotesi che $f(x,\ y)$ soddisfi la condizione di Lipschitz sull'intervallo assegnato.

METODI ONE-STEP E MULTISTEP

Esistono due classi di metodi di soluzione numerica di equazioni differenziali: i metodi ad un solo passo (one-step) e quelli a passo multiplo (multi-step).

Dato che si tratta di metodi di risoluzione per convergenza di una successione di approssimazioni verso la soluzione del problema, le due classi si distinguono nel numero di approssimazioni calcolate ai passi precedenti, che vengono utilizzate nel calcolo della approssimazione corrente.

Nel caso di metodi one-step l'unica approssimazione utilizzata è quella calcolata al passo precedente; nel caso dei metodi multi-step sono utilizzate invece numerose approssimazioni, tra quelle precedentemente calcolate.

Assumendo $y_n = y(x_n)$ e $h = \dfrac{b-a}{N}$, si ha infatti:

$$y_{n+1} = \begin{cases} y_n + h\Phi(x_n, y_n; h) & (\text{ one step }) \\ y_n + h\Phi(x_n, y_n, y_{n-1}, \ldots, y_{n-k+1}; h) & (\text{ multistep }) \end{cases}$$

Entrambi i metodi sono espressi in forma esplicita. Qualora compaia la approssimazione corrente della soluzione y_{n+1} tra gli argomenti di $\Phi(\cdots)$, la forma diviene implicita.

METODO DI EULERO

E' il metodo più semplice. Si tratta di un metodo a passo singolo, esplicito, in cui la derivata è approssimata con il rapporto incrementale , ovvero:

$$f(x_n, y_n) = y'_n \cong \frac{y_{n+1} - y_n}{h}$$

da cui:

$$y_{n+1} = y_n + hf(x_n, y_n)$$

METODI RUNGE-KUTTA (RK)

Nel caso del generico metodo RK (esplicito e a passo singolo) ad un numero di stadi pari a r la soluzione viene calcolata nel modo seguente:

$$\begin{cases} y_{n+1} = y_n + h\sum_{i=1}^{r} a_i k_i \\ k_1 = f(x_n, y_n) \\ k_i = f\left(x_n + b_i h, y_n + \sum_{j=1}^{i-1} c_{ij} k_j\right) \end{cases} \qquad i = 2, .., r$$

L'approssimazione corrente è calcolata sulla base di quella ottenuta al passo precedente e dei valori assunti dalla funzione $f(x, y)$ in alcuni punti dell'intervallo, definiti in ragione del passo di integrazione h, secondo una combinazione lineare di coefficienti a_i.

Il metodo di Eulero è in realtà un metodo RK ad uno stadio. Altri metodi utilizzati sono il metodo di Heun e quello di Eulero modificato, entrambi a due stadi.

Heun:
$$\begin{cases} y_{n+1} = y_n + \dfrac{h}{2}(k_1 + k_2) \\ k_1 = f(x_n, y_n) \\ k_2 = f(x_n + h, y_n + hk_1) \end{cases}$$

Eulero modificato:
$$\begin{cases} y_{n+1} = y_n + hk_2 \\ k_1 = f(x_n, y_n) \\ k_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right) \end{cases}$$

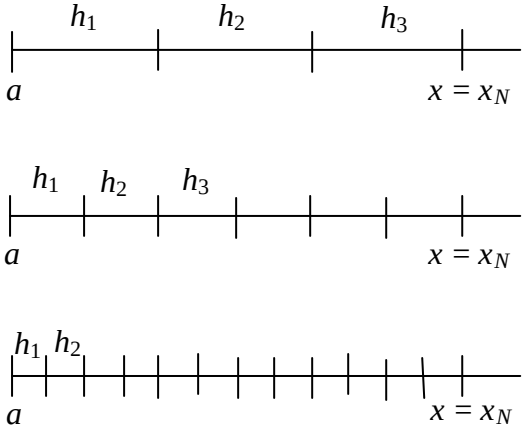
Un RK a 4 stadi molto utilizzato è il seguente

$$\begin{cases} y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ k_1 = f(x_n, y_n) \\ k_2 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1) \\ k_3 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_2) \\ k_4 = f(x_n + h, y_n + hk_3) \end{cases}$$

CONVERGENZA DI UN METODO ONE-STEP E ORDINE DI CONVERGENZA

Un metodo a passo singolo è convergente se, per qualsiasi $x \in [a, b]$ risulta $\lim_{N\to\infty} y_N(x) = y(x)$, ovvero se aumentando la discretizzazione dell'intervallo [*a*, *b*] l'approssimazione converge uniformemente al valore vero. I metodi RK considerati sono convergenti.

Se la discretizzazione dell'intervallo viene gradualmente infittita



nel caso di metodo di tipo adattativo, in cui il passo è variabile nell'intervallo dove è definito il problema, si può definire come passo:

$$h = \max(h_I)$$

e la convergenza è assicurata se:

$$\lim_{h\to 0} y_N = \lim_{N\to\infty} y_N = y(x)$$

Si dice che il metodo ha ordine di convergenza pari a *p* (intero) se:

$$|y_N(x) - y(x)| = O(N^{-p})$$

I metodi RK a r stadi con $r \leq 4$ hanno ordine di convergenza $p = r$.

PROVE di ESONERO
5 maggio 1994

PARTE I

1. Che cosa significa che una matrice $\mathbf{A} \in R^{n,n}$ è:
   - simmetrica definita positiva
   - a diagonale dominante

2. Dimostrare che gli autovalori della matrice $\mathbf{A}^{-1}$ coincidono con i reciproci degli autovalori di $\mathbf{A}$.
   E gli autovettori?

3. Descrivere un algoritmo che effettui il calcolo di $\|x\|_\infty$.

4. Siano dati i seguenti punti (x_i, y_i): (0, 2) , (-1, 1) , (1, 1) , (-2, 0) , (2, 0). Costruire il polinomio di interpolazione.

5. L'equazione $\cos x - \log x = 0$ ha una radice nell'intervallo $(1, \pi/2)$. Calcolare tale radice mediante il metodo delle tangenti (scrivere soltanto la formula risolutiva).

6. Sia data l'equazione differenziale
$$\begin{cases} y' = (x+1) \cdot y \quad , x > 0 \\ y(0) = 1 \end{cases}$$
   Applicare il metodo di Eulero.

PARTE II

7. Dimostrare che la matrice $\mathbf{B}=\mathbf{A}^T\mathbf{A}$ è simmetrica semidefinita positiva.

8. Siano dati $(m+1)$ punti (x_i, y_i), $i=0, \ldots, m$. Determinare con il criterio dei minimi quadrati la retta $y = ax+b$.

PROVA di ESONERO
11 luglio 1995

1. Definire il concetto di condizionamento di un problema numerico
$$\mathbf{y} = f(\mathbf{x}), \qquad \mathbf{x} \in R^n, \mathbf{y} \in R^m.$$

2. Scrivere l'algoritmo che risolve un sistema triangolare inferiore.

3. Illustrare i singoli passi dell'algoritmo di Gauss con pivoting parziale applicando il metodo al seguente sistema:
$$\begin{cases} 2x_2 - x_4 = 1 \\ 2x_1 - x_2 + x_3 - 2x_4 = 0 \\ x_1 - 2x_3 + x_4 = 0 \\ -x_1 + 3x_2 + x_3 + x_4 = 4 \end{cases}$$

4. Scrivere i passi fondamentali, sotto forma di algoritmo, per "ottenere" l'autovalore di modulo minimo di una matrice **A** reale.

5. Dimostrare che per (n+1) punti del piano, con ascisse distinte, passa un unico polinomio di grado n.

6. Costruire il polinomio di interpolazione (nella forma di Newton alle differenze divise) associato ai seguenti dati:
$$\begin{cases} f(0) = 1 \\ f(1) = 2 \\ f(-1) = 0 \\ f(2) = 1 \end{cases}$$

7. Ricavare la formula di Simpson. Successivamente costruire la corrispondente versione composta.

8. Descrivere i metodi numerici per la risoluzione di equazioni differenziali ordinarie (con valori iniziali).

PROVA di ESONERO
14 luglio 1997

1. Scrivere l'algoritmo di Gauss, con pivoting parziale, per risolvere un generico sistema **Ax**=**b**. Applicarlo, descrivendoi singoli passi, sistema

$$\begin{cases} x_1 + x_2 + x_3 = 3 \\ 2x_1 \quad - x_3 = 1 \\ 2x_1 - x_2 - x_3 = 0 \end{cases}$$

2. Scrivere le formule che rappresentano i metodi iterativi di Jacobi e di Gauss-Seidel.
   Quando convergono?

3. Dovendo operare la somma $y=x_1+x_2$ usando un calcolatore che opera con una mantissa di t cifre, in quali casi il risultato y fornito ha una precisione inferiore a quella dei due dati x_1 e x_2?

4. Costruire il polinomio di interpolazione associato ai seguenti punti:

   (0, 1) , (-1, 1) , (-2, 0) , (1, 3)

5. Come procedereste volendo determinare, per esempio con precisione di macchina, il valore dell'integrale $\int_0^{2p} e^{\cos x}\,dx$?

6. Descrivere i metodi per risolvere equazioni differenziali ordinarie con valori iniziali e presentare alcuni esempi.