Indice

0.1 Premessa

La continua richiesta da parte degli studenti dei corsi di Calcolo Numerico da me tenuti mi ha spinto, dopo non pochi ripensamenti, alla stesura di queste brevi note. Sebbene sia sempre convinto che esistono sulla piazza molti ottimi libri di Calcolo Numerico, mi sono altresì convinto che per gli studenti può essere di grande aiuto, per la preparazione dell' esame, un *testo guida* che contenga gli argomenti svolti nel corso dell' anno. Per questa seconda ragione i presenti appunti sono stati stesi, giorno per giorno, durante lo svolgimento del corso di Calcolo Numerico 1/2 annualita' del secondo semestre 1997. Nonostante l'aspetto tipografico (grazie al "favoloso" TeX) restano dei semplici appunti per cui si invitano fin d'ora gli studenti ad epurare il testo dei vari errori di stampa e dalle eventuali imprecisioni.

Questo sforzo è stato fatto principalmente per gli studenti che, a causa delle sovapposizioni d'orario, non riescono a seguire le lezioni. Per quelli che seguono un invito a segnalarmi le discrepanze fra appunti e lezioni. Le prime bozze di questi appunti saranno disponibili, per gli studenti del corso, sulla mia pagina Web (http://marfro.mate.polimi.it) sotto forma di file post-script o DVI (sperimentale utilizzo didattico del potente mezzo informatico: Internet).

0.2 Introduzione

Gli argomenti del corso, divisi in 6 capitoli, sono i seguenti:

1. Teoria dell' errore
2. Algebra lineare numerica

   Sistemi lineari

   Autovalori
3. Zeri di equazioni e sistemi non lineari
4. Teoria dell' approssimazione

   Interpolazione

   Minimi quadrati

5. Formule di quadratura

6. Equazioni differenziali ordinarie

Nel capitolo 1, dopo aver introdotto i concetti di *errore relativo,* di *numero macchina* e di *epsilon macchina,* viene studiata la propagazione dell' errore relativo nelle operazioni elementari di macchina. Vengono presentati anche i concetti di *stabilità* algoritmica e di *condizionamento* di un problema numerico. Con semplici esempi si introducono anche il concetto di *errore di troncamento,* di *convergenza* ed *accuratezza* (o *stima dell' errore*) per un processo numerico iterativo.

Nel capitolo 2, vengono presentati i principali metodi, sia di tipo *diretto* che di tipo *iterativo,* per la risoluzione di sistemi lineari. Per i metodi diretti si porrà l' accento sulla loro *complessità computazionale*, mentre per gli iterativi si analizzeranno in dettaglio le condizioni di *convergenza* e *maggiorazione dell' errore*. Il problema del *condizionamento* di matrici e la sua influenza sulla "buona" risolubilità di un sistema lineare verrà trattato in dettaglio. Viene fatto un cenno sulla risoluzione di sistemi lineari *nel senso dei minimi quadrati.*

Per il calcolo degli autovalori e autovettori di matrici si presenta in dettaglio il *metodo delle potenze* ed alcune sue varianti, facendo solo un cenno ai metodi globali (*algoritmo QR*).

Nel capitolo 3, vengono presentati alcuni metodi iterativi generali per il calcolo degli zeri di funzioni e sistemi non lineari. Dopo aver introdotto il classico metodo di *bisezione*, viene presentata la famiglia dei metodi di *punto fisso* (metodi di *ordine* k), con le relative condizioni di *convergenza* ed *accuratezza*. Il metodo di *Newton* e le sue varianti vengono studiati in dettaglio anche per sistemi. Per le equazioni algebriche vengono presentati metodi basati sull' uso della matrice di *Frobenious* o *companion matrix*. Viene inoltre discusso il problema della *sensitività* delle radici di un polinomio alle variazioni dei suoi coefficienti.

Nel capitolo 4, si fornisce una breve introduzione alle problematiche dell' approssimazione. Dopo aver richiamato i concetti base di *miglior approssimazione* si dà spazio all' interpolazione *polinomiale* con qualche cenno all' interpolazione mediante funzioni *spline*. Per quanto riguarda l' approssimazione nel senso dei minimi quadrati verrà considerato il caso polinomiale e di Fourier.

Nel capitolo 5, vengono presentate le formule di tipo interpolatorio di *Newton-Cotes* semplici e *composte*, definendone l' accuratezza e provando-

ne la convergenza (composte). Si farà inoltre un breve cenno alle formule Gaussiane e alla loro proprietà di ottimalità.

Nel capitolo 6, si forniscono gli strumenti essenziali per affrontare in modo critico la risoluzione dei problemi differenziali che tanto interessano i moderni ingegneri. Vengono presentate in dettaglio le formule ad un passo (quali i metodi *Runge-Kutta*) dandone una semplice deduzione e soffermandosi sui concetti di *consistenza*, *stabilità* e *convergenza*. Per le formule a più passi verrà fatta una breve introduzione al fine di evidenziarne gli aspetti computazionali.

0.2.1 Un semplice esempio

Prima di addentrarci nel vivo dello studio dei *metodi numerici* mi sembra opportuno dare una pur semplice giustificazione dell' importanza di tali metodi. A tal fine può essere utile considerare il seguente problema modello (modello di Malthus):

$$\begin{cases} y' &= \lambda y \\ y(0) &= y_0, \quad y_0 > 0 \end{cases} \tag{1}$$

la cui soluzione è $y(t) = y_0 e^{\lambda t}$. Se il problema (1) simula la crescita di una colonia di batteri, può essere interessante individuare il *tasso di crescita* λ. A tal fine è sufficiente conoscere il valore $y(t^*) = y^*$, per risolvere:

$$y_0 e^{\lambda t^*} = y(t^*) \tag{2}$$

mediante la formula:

$$\lambda = \frac{1}{t^*} \ln \left(\frac{y(t^*)}{y_0} \right) \tag{3}$$

(è necessaria una calcolatrice tascabile con la funzione logaritmo).

Se si complica un poco il modello includendo il fenomeno di immigrazione ed emigrazione l' equazione differenziale del problema di Cauchy (1) si complica di poco divenendo

$$y' = \lambda y + d \tag{4}$$

dove la costante d tiene conto del nuovo fenomeno. La soluzione di (4) tenuto conto delle condizioni iniziali date in (1) è quindi:

$$y(t) = e^{\lambda t} \left\{ y_0 + \frac{d}{\lambda} \left(1 - e^{-\lambda t} \right) \right\} \tag{5}$$

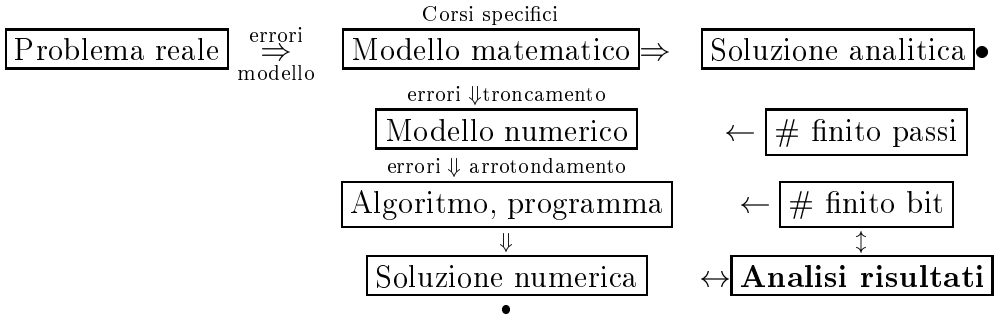
dalla (5) si ha:

$$e^{-\lambda t^*} y(t^*) = y_0 + \frac{d}{\lambda}\left(1 - e^{-\lambda t^*}\right)$$

equazione NON lineare in λ per la cui risoluzione sono necessari metodi numerici.

Il semplice esempio riportato suggerisce uno schema generale di lavoro quando si debba risolvere un *problema reale:*

Schema risolutivo di un problema



Capitolo 1

Teoria dell' errore (M.Frontini)

1.1 Definizioni (M.F.)

Introduciamo alcune definizioni che ci serviranno in seguito. Dati due numeri reali x e $\overline{x}$ dove x è il valore "vero" e $\overline{x}$ è il valore approssimato (di x), definiamo:

Definizione 1.1.1 *Errore assoluto la quantità:*

$$|x - \overline{x}| = e_x$$

Definizione 1.1.2 *Errore relativo la quantità:*

$$\frac{|x - \overline{x}|}{|x|} = \frac{e_x}{|x|} = \varepsilon_x$$

Definizione 1.1.3 *Numero troncato a t cifre (chopping): valore che si ottiene ignorando le cifre dopo la t-esima. (Es. 3,271 è il troncato alla quarta cifra di 3,2716432..).*

Definizione 1.1.4 *Numero arrotondato a t cifre (rounding): valore che si ottiene ignorando le cifre dopo la t-esima se la $(t+1)$-esima è (in base 10) ≤ 4 ed aumentando la t-esima di una unità se la $(t+1)$-esima è (in base 10) ≥ 5. (Es. 3,272 è l' arrotondato alla quarta cifra di 3,2716432..).*

Definizione 1.1.5 *Intervallo d'indeterminazione: intervallo in cui è compreso il valore esatto*

$$\overline{x} - e_x \leq x \leq \overline{x} + e_x$$

Dalle definizioni date per l' errore assoluto e relativo si vede subito che è l' errore relativo quello che più interessa nelle applicazioni (numero di cifre esatte), si può osservare altresì che se x è "piccolo" in modulo (prossimo a zero) l' errore assoluto è sicuramente più significativo. Per questo motivo si preferisce usare nella pratica la seguente definizione di errore "relativo a buon senso":

$$\frac{|x - \overline{x}|}{1 + |x|} = \frac{e_x}{1 + |x|} = \varepsilon_{rx}$$

dove il pedice r sta per "ragionevole". È facile osservare che per x "grandi" $\varepsilon_{rx} \simeq \varepsilon_x$ mentre per x "piccoli" si ha $\varepsilon_{rx} \simeq e_x$. In questo modo un $\varepsilon_{rx} \simeq 10^{-6}$ è indice di una buona stima di x sia che esso sia "grande" o "piccolo".

L' importanza dell' uso dell' errore relativo viene rafforzata dal fatto che i numeri che vengono gestiti da un elaboratore elettronico non sono i classici numeri reali dell' analisi ma sono un sottoinsieme finito di questi.

1.2 Numeri macchina (floating point) (M.F.)

È nota a tutti la *rappresentazione posizionale* dei numeri per cui:

$$745.23 = 7 * 10^2 + 4 * 10^1 + 5 * 10^0 + 2 * 10^{-1} + 3 * 10^{-2}$$

tale rappresentazione non è utile nel *calcolo scientifico* in quanto si deve presupporre a priori quante cifre utilizzare per la *parte intera* e per la *parte decimale*. Più conveniente è la *rappresentazione esponenziale* per cui:

$$745.23 = 0.74523 * 10^3 = 0.074523 * 10^4 = 7.4523 * 10^2 = ...$$

se si usa la rappresentazione esponenziale non si ha unicità a meno di considerarla *normalizzata* ovvero la mantissa deve essere minore di 1 e la cifra più significativa deve essere diversa da zero (la prima delle 3 rappresentazioni esponenziali date).

Quanto detto per la base 10 si estende in modo naturale alla base 2 (la base naturale dei calcolatori). Può essere il caso di far osservare che la rappresentazione di un numero razionale decimale finito può, in base 2, generare un numero razionale binario infinito (periodico), come illustrato dal seguente esempio:

$$0.2_{10} = \frac{1_{10}}{5_{10}} = \frac{1_2}{101_2} = 0.\overline{0011}_2$$

come è facile verificare eseguendo la divisione fra 1_2 e 101_2.

Dato un numero reale x indicheremo d'ora in poi con $\overline{x}$ il suo rappresentante nell' insieme dei numeri macchina, dove

Definizione 1.2.1 *Numero macchina è una stringa di bit della forma:*

$$\overline{x} = \pm \left(\frac{d_1}{2} + \frac{d_2}{2^2} + \frac{d_3}{2^3} + ... + \frac{d_t}{2^t} \right) * 2^e$$

dove:
$0 \leq d_i \leq 1$, *i=1,2,...,t* ($d_1 \neq 0$, *normalizzato)*
$L \leq e \leq U$, *L ed U sono i valori min e max che può assumere* $e \in Z$.
t è il numero di cifre utilizzate per la mantissa.□

Da quanto sopra si evincono le seguenti proprietà dell' insieme $\mathcal{F}$ dei numeri floating point:

1. $\mathcal{F} \subset R$
2. $\mathcal{F}$ è limitato tra un minimo ed un massimo ($\overline{x}_{\min}, \overline{x}_{\max} \in \mathcal{F}$)
3. $\mathcal{F}$ è formato da un numero finito di elementi pari a:

$$Card(\mathcal{F}) = 2^t * (U - L + 1) + 1$$

4. Gli elementi dell' insieme $\mathcal{F}$ non sono equidistanziati sulla retta reale.

Queste proprietà di $\mathcal{F}$ generano le seguenti conseguenze

1. Due numeri reali distinti x ed y possono avere lo stesso rappresentante in $\mathcal{F}$.
2. Esistono numeri reali per cui non esiste il rappresentate in $\mathcal{F}$ (*Overflow*).
3. $\forall x < \epsilon$ ($\epsilon > 0$ è lo *zero macchina*), $x \in R$ è rappresentato in $\mathcal{F}$ da $\overline{x} = 0$ (*Underflow*).
4. L' errore relativo di rappresentazione di un numero reale mediante il suo rappresentante in $\mathcal{F}$ è sempre lo stesso.

5. Dati due numeri in $\mathcal{F}$ la loro somma può non appartenere ad $\mathcal{F}$.

L' aritmetica di un elaboratore digitale può quindi essere caratterizzata dalla terna (t, L, U). Di seguito rappresentiamo l' insieme $\mathcal{F}$ che si ottiene considerando un elaboratore che ha le seguenti caratteristiche:

$$(t, L, U) = (3, -2, 1)$$

il numero degli elementi di $\mathcal{F}$ è quindi 33, le quattro differenti rappresentazioni della mantissa sono:

$$\{.100 = 4/8; .101 = 5/8; .110 = 6/8; .111 = 7/8\}$$

ed i quattro esponenti:

$$\left\{\frac{1}{4}; \frac{1}{2}; 1; 2\right\}$$

per cui i 16 numeri positivi rappresentabili sono:

$$\{4/32; 5/32; 6/32; 7/32; 4/16; 5/16; 6/16; 7/16; 4/8; 5/8; 6/8; 7/8; 4/4; 5/4; 6/4; 7/4\}$$

a cui vanno aggiunti i 16 opposti e lo zero.

Dalla definizione di insieme $\mathcal{F}$ è facile dedurre l' errore che si commette quando si inserisce un numero reale in un elaboratore, precisamente proviamo che:

Teorema 1.2.1 *L'errore dovuto alla memorizzazione all'interno di un elaboratore binario è:*

$$\frac{|x - \overline{x}|}{|x|} \leq \begin{cases} 2^{1-t} & chopping \\ 2^{-t} & rounding \end{cases}$$

dove t è il numero di cifre utilizzato per la mantissa.

Dimostrazione 1.2.1

$$x = \pm .d_1 d_2 ... d_t ... * 2^e \quad mentre \quad \overline{x} = \pm .d_1 d_2 ... d_t * 2^e$$

con $d_1 \neq 0$ (floating point normalizzato), per cui

$$|x - \overline{x}| \leq \begin{cases} 2^{-t+e} & chopping \\ 2^{-t+e-1} & rounding \end{cases}$$

ed essendo

$$|x| \geq 2^{e-1}$$

segue la tesi. ■

Essendo l' errore relativo legato al numero di cifre utilizzate per la mantissa è chiaro perchè l' uso della *doppia precisione* possa risultare utile nella computazione.

La *precisione di macchina* può essere caratterizzata dalla seguente definizione

Definizione 1.2.2 *Si chiama epsilon macchina (o precisione di macchina) il più piccolo numero positivo di $\mathcal{F}$ che sommato ad 1 fornisce un risultato maggiore di 1.*

$$eps := \{\min \overline{x} \in \mathcal{F} \mid 1 + \overline{x} > 1\} .\square$$

Un semplice programma (MATLAB like) che fornisce *eps* è il seguente

```
eps=1.
while (eps+1 > 1)
    eps=0.5*eps
end
eps=2.*eps
```

È importante non confondere il concetto di epsilon macchina con quello di *zero macchina* definito da:

Definizione 1.2.3 *Lo zero macchina è il più piccolo numero macchina (in valore assoluto)*

$$zero\_macchina := \{\min |\overline{x}| , \quad x \in \mathcal{F}\}$$

e dipende dal minimo valore attribuibile all' esponente.$\square$

1.3 Propagazione degli errori (M.F.)

Ci proponiamo ora di analizzare come l' errore relativo (errore di rappresentazione) si propaga nelle operazioni elementari. Siano dati due numeri reali $x, y \in R$ (non nulli) ed i loro "rappresentanti in $\mathcal{F}$": $\overline{x}, \overline{y} \in \mathcal{F}$, si ha:

$$\begin{aligned} \overline{x} &= x(1+\epsilon_x); \quad |\epsilon_x| < eps \\ \overline{y} &= y(1+\epsilon_y); \quad |\epsilon_y| < eps \end{aligned}$$

dove *eps* è la precisione di macchina. Ricordando lo sviluppo in serie di Taylor di una funzione di due variabili arrestato al primo ordine si ha

$$f(\overline{x},\overline{y}) = f(x(1+\epsilon_x), y(1+\epsilon_y)) \simeq f(x,y) + f_x(x,y)x\epsilon_x + f_y(x,y)y\epsilon_y$$

per cui

$$\epsilon_f = \frac{f(\overline{x},\overline{y}) - f(x,y)}{f(x,y)} \simeq \frac{xf_x\epsilon_x + yf_y\epsilon_y}{f(x,y)} \tag{1.1}$$

Mediante la (1.1) è facile dedurre le seguenti valutazioni per le operazioni elementari

1.3.1 Somma algebrica

Essendo $f(x,y) = x \pm y$, supposto $x \pm y \neq 0$, si ha:

$$\frac{(x \pm y) - (\overline{x} \pm \overline{y})}{x \pm y} = \epsilon_{x\pm y} = \frac{x\epsilon_x \pm y\epsilon_y}{x \pm y} = \frac{x}{x \pm y}\epsilon_x \pm \frac{y}{x \pm y}\epsilon_y \tag{1.2}$$

dalla (1.2) è facile osservare che l' errore nella somma algebrica viene amplificato (se $x \pm y \simeq 0$) dai fattori $\frac{x}{x\pm y}$ e $\frac{y}{x\pm y}$, che possono essere "grandi" (*cancellazione numerica*).

1.3.2 Prodotto

Essendo $f(x,y) = x * y$, si ha:

$$\frac{(x * y) - (\overline{x} * \overline{y})}{x * y} = \epsilon_{x*y} \simeq \frac{xy\epsilon_x + xy\epsilon_y}{x * y} = \epsilon_x + \epsilon_y \tag{1.3}$$

dalla (1.3) è facile osservare che l' errore relativo nel prodotto non viene amplificato.

1.3.3 Divisione

Essendo $f(x,y) = x/y$, supposto $y \neq 0$, si ha:

$$\frac{(x/y) - (\overline{x}/\overline{y})}{x/y} = \epsilon_{x/y} \simeq \frac{\frac{1}{y}x\epsilon_x - \frac{x}{y^2}y\epsilon_y}{x/y} = \epsilon_x - \epsilon_y \tag{1.4}$$

dalla (1.4) è facile osservare che l' errore relativo nella divisione non viene amplificato.

1.3.4 Radice quadrata

Sia $f(x) = \sqrt{x}$, supposto $x \neq 0$, si ha:

$$\frac{\sqrt{x} - \sqrt{\overline{x}}}{\sqrt{x}} = \epsilon_{\sqrt{x}} \simeq \frac{\frac{1}{2\sqrt{x}} x \epsilon_x}{\sqrt{x}} = \frac{1}{2}\epsilon_x \tag{1.5}$$

e la (1.5) ci mostra come l' errore relativo venga dimezzato, $\forall x \neq 0$, quando si estrae la radice quadrata.

L' analisi dei risultati precedenti ci fa capire che non tutte le operazioni di macchina si comportano allo stesso modo per quanto riguarda la propagazione degli errori, ma esistono operazioni più, "delicate" (somma algebrica) ed altre più "robuste" (prodotto, estrazione di radice).

È doveroso osservare che le stime fornite sono "deterministiche". Nella pratica, quando si eseguono milioni di operazioni macchina, gli arrotondamenti nelle operazioni migliorano le cose (non abbiamo qui il tempo di considerare la propagazione degli errori da un punto di vista statistico) per cui, a parte casi patologici, le computazioni forniscono "mediamente" risultati ragionevoli.

1.4 Alcuni esempi (M.F.)

Diamo ora alcuni esempi di come le *operazioni di macchina* e gli *algoritmi di calcolo* possano influenzare la bontà di un risultato.

1.4.1 Cancellazione numerica

Prende il nome di cancellazione numerica quel fenomeno che si manifesta quando si sommano due numeri macchina quasi uguali in modulo ma di segno opposto. In questo caso (cfr. (1.2)) la somma di macchina è "instabile" e non gode della proprieta associativa. Consideriamo il seguente esempio: siano $a = 0.23371258e-4$, $b = 0.33678429e+2$, $c = -0.33677811e+2$ tre numeri dati. Supponiamo di operare con una calcolatrice che lavori in base 10 con 8 cifre significative (cifre della mantissa). Verifichiamo che

$$a + (b + c) \neq (a + b) + c$$

infatti si ha:

$a + (b + c) = 0.23371258e-4 + 0.61800000e-3 = 0.64137126e-3$
$(a + b) + c = 0.33678452e+2 + (-0.33677811e+2) = 0.64100000e-3$

1.4.2 Instabilità algoritmica

È quel fenomeno per cui un algoritmo di calcolo propaga (amplifica) gli errori sui dati. Supponiamo di dover fornire una tabella contenente i primi 20 valori dei seguenti integrali

$$E_n = \int_0^1 x^n e^{x-1} dx, \qquad n = 1, 2, ... \tag{1.6}$$

integrando per parti si ottiene la relazione ricorrente

$$E_n = 1 - nE_{n-1}, \qquad n = 2, 3, ... \tag{1.7}$$

sapendo che $E_1 = \frac{1}{e}$, è facile ricavare i valori cercati dalla (1.7). Sfortunatamente, pur dovendo essere $E_n > 0$, $\forall n$, la (1.7) fornisce, utilizzando un comune personal computer, valori negativi già per $n \geq 15$.

Se si inverte la ricorrenza (1.7) nella forma

$$E_{n-1} = \frac{1 - E_n}{n}, \qquad n = ..., 3, 2 \tag{1.8}$$

ponendo $E_{30} = 0$, si possono ottenere i primi 20 valori di E_n alla precisione di macchina. Una più attenta analisi delle (1.7) e (1.8) mostra come nel primo caso l' errore su E_1 venga amplificato, ad ogni passo di un fattore che, dopo n passi, è pari ad $n!$, mentre nel secondo l' errore su $E_{30} > 0$, viene ridotto ad ogni passo di un fattore che, dopo n passi, è pari ad $\frac{1}{n!}$.

1.4.3 Sensitività del problema

Esistono problemi che sono, per loro natura, *sensibili* alle variazioni sui dati. Un semplice esempio è fornito dalla ricerca delle radici del seguente polinomio di secondo grado

$$(x - 2)^2 = 10^{-6} \tag{1.9}$$

che puo', ovviamente, essere scritto nella forma

$$x^2 - 4x + (4 - 10^{-6}) = 0 \tag{1.10}$$

le radici di (1.10) sono:

$$x = 2 \pm 10^{-3}$$

se nella (1.9) sostituiamo 10^{-6} con $4 * 10^{-6}$ introduciamo un errore di $3 * 10^{-6}$ sul termine noto della (1.10) le cui radici divengono

$$x = 2 \pm 2 * 10^{-3}$$

per cui si induce un errore dell' ordine di 10^{-3} sulle radici. Tutto questo operando "con carta e matita" (non ci sono errori dovuti alla rappresentazione dei numeri macchina).

1.5 Riepilogo (M.F.)

Dalle precedenti note emerge che:

1. gli errori sono sempre presenti nella computazione (finitezza della parola di un elaboratore);
2. gli elaboratori che arrotondano sono meglio di quelli che troncano;
3. bisogna utilizzare algoritmi stabili per non propagare gli errori;
4. esistono problemi sensibili alle variazione sui dati (mal condizionati) che vanno trattati con molta attenzione;
5. l' uso della doppia (multipla) precisione è utile come verifica dei risultati più che come strumento naturale di lavoro (aumento del tempo di calcolo, inefficacia sui problemi mal condizionati);

1.5.1 Esercizi

Esercizio 1.5.1 *Qual è il risultato della seguente sequenza di istruzioni MATLAB ?*

```
x=0
while x ~ =1
x=x+0.1;
x,sqrt(x)
end
```

Esercizio 1.5.2 *Qual è il risultato della seguente sequenza di istruzioni MATLAB ?*

```
x=0
while x ~ =1
x=x+0.5;
x,sqrt(x)
end
```

Esercizio 1.5.3 *Dire come si opererebbe per calcolare la seguente espressione*

$$f(x) = x(\sqrt{x+1} - \sqrt{x})$$

per valori di $x = 10^n$, $n = 1:5$.

Esercizio 1.5.4 *Dire come si opererebbe per calcolare la seguente espressione*

$$f(x) = \frac{1 - \cos(x)}{x^2}$$

per valori di $x = 10^{-n}$, $n = 4:9$.

Esercizio 1.5.5 *Commentare il grafico che si ottiene eseguendo il plot in MATLAB della funzione*

$$f(x) = x^6 - 6x^5 + 15x^4 - 20x^3 + 15x^2 - 6x + 1$$

per $0.998 \leq x \leq 1.002$.

Esercizio 1.5.6 *Dovendo calcolare* $z = \sqrt{x^2 + y^2}$ *si suggeriscono le seguenti formule*

$$|x|\sqrt{1 + (y/x)^2}; \qquad 0 \leq |y| \leq |x|$$
$$|y|\sqrt{1 + (x/y)^2}; \qquad 0 \leq |x| \leq |y|$$

giustificare la bontà delle formule proposte.

Capitolo 2

Algebra lineare numerica (M.Frontini)

2.1 Sistemi lineari (M.F.)

2.1.1 Richiami di algebra lineare

Per sistema lineare si intende un insieme di m equazioni in n incognite della forma

$$\begin{cases} a_{11}x_1+ & a_{12}x_2+ & ...+ & a_{1n}x_n = & b_1 \\ a_{21}x_1+ & a_{22}x_2+ & ...+ & a_{2n}x_n = & b_2 \\ ... & ... & ... & ... & ... \\ a_{m1}x_1+ & a_{m2}x_2+ & ...+ & a_{mn}x_n = & b_m \end{cases} \tag{2.1}$$

dove $a_{ij} \in R$ (eventualmente a C), $b_i \in R$ (eventualmente a C). In forma matriciale (2.1) si può scrivere

$$A\underline{x} = \underline{b}$$

dove A è una matrice in $R^{m,n}$ di elementi a_{ij}, $\underline{b}$ è un vettore in R^m ed $\underline{x}$ è in R^n.

Richiamiamo le seguenti definizioni, perchè utili in seguito:

Definizione 2.1.1 *Vettore è una n-pla ordinata di numeri (in colonna).*

Definizione 2.1.2 *Matrice è una tabella rettangolare o quadrata di numeri (insieme di vettori colonna).*

Definizione 2.1.3 *Versore i-esimo* $\underline{e}_i$ *è un vettore colonna formato da tutti 0 tranne l'elemento i-esimo che vale 1.*

$$\underline{e}_i = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leftarrow i$$

Definizione 2.1.4 *Matrice identità I è la matrice formata dagli n versori* $\underline{e}_i$*, i=1,2,...,n.*

Definizione 2.1.5 *Matrice diagonale è una matrice con gli elementi* $a_{ii} \neq 0$*, i=1,2,...,n e gli altri nulli.*

Definizione 2.1.6 *Matrice tridiagonale è una matrice per cui solo* $a_{ii} \neq 0$*,* $a_{i+1i} \neq 0$*,* $a_{ii+1} \neq 0$*.*

$$T = \begin{bmatrix} a_{11} & a_{12} & 0 & \cdots & 0 \\ a_{21} & a_{22} & a_{23} & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & a_{n-1n-2} & a_{n-1n-1} & a_{n-1n} \\ 0 & \cdots & 0 & a_{nn-1} & a_{nn} \end{bmatrix}$$

Definizione 2.1.7 *Matrice bidiagonale inferiore (superiore) è una matrice per cui solo* $a_{ii} \neq 0$*,* $a_{i+1i} \neq 0$*, (* $a_{ii} \neq 0$*,* $a_{ii+1} \neq 0$*).*

$$\begin{bmatrix} a_{11} & 0 & \cdots & \cdots & 0 \\ a_{21} & a_{22} & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & a_{n-1n-2} & a_{n-1n-1} & 0 \\ 0 & \cdots & 0 & a_{nn-1} & a_{nn} \end{bmatrix}; \quad \begin{bmatrix} a_{11} & a_{12} & 0 & \cdots & 0 \\ 0 & a_{22} & a_{23} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & a_{n-1n-1} & a_{n-1n} \\ 0 & \cdots & \cdots & 0 & a_{nn} \end{bmatrix}$$

Definizione 2.1.8 *Matrice triangolare inferiore (superiore) è una matrice per cui $a_{ij} = 0$, $j > i$ ($a_{ij} = 0$, $j < i$).*

$$\begin{bmatrix} a_{11} & 0 & \cdots & \cdots & 0 \\ a_{21} & a_{22} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ a_{n-11} & \ddots & a_{n-1n-2} & a_{n-1n-1} & 0 \\ a_{n1} & a_{n2} & \cdots & a_{nn-1} & a_{nn} \end{bmatrix}; \quad \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n-1} & a_{1n} \\ 0 & a_{22} & a_{23} & \ddots & a_{2n} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{n-1n-1} & a_{n-1n} \\ 0 & \cdots & \cdots & 0 & a_{nn} \end{bmatrix}$$

Definizione 2.1.9 *Matrice in forma di Hessemberg inferiore (superiore) è una matrice per cui $a_{ij} = 0$, $j > i + 1$ ($a_{ij} = 0$, $j < i - 1$).*

$$\begin{bmatrix} a_{11} & a_{12} & 0 & \cdots & 0 \\ a_{21} & a_{22} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ a_{n-11} & \ddots & a_{n-1n-2} & a_{n-1n-1} & a_{n-1n} \\ a_{n1} & a_{n2} & \cdots & a_{nn-1} & a_{nn} \end{bmatrix}; \quad \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n-1} & a_{1n} \\ a_{21} & a_{22} & a_{23} & \ddots & a_{2n} \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & a_{n-1n-1} & a_{n-1n} \\ 0 & \cdots & 0 & a_{nn-1} & a_{nn} \end{bmatrix}$$

Definizione 2.1.10 *Matrice trasposta è una matrice che si ottiene scambiando le righe con le colonne (si indica con A^T).*

Definizione 2.1.11 *Matrice simmetrica è una matrice ad elementi reali, per cui $A = A^T$.*

Definizione 2.1.12 *Matrice definita positiva è una matrice ad elementi reali, per cui $\underline{x}^T A \underline{x} > 0, \quad \forall \underline{x} \neq \underline{0}$.*

Definizione 2.1.13 *Matrice inversa di una matrice quadrata A è una matrice per cui $A^{-1}A = AA^{-1} = I$.*

Definizione 2.1.14 *Matrice ortogonale è una matrice ad elementi reali, per cui $A^T A = AA^T = I$ ($A^T = A^{-1}$).*

Definizione 2.1.15 *Matrice a diagonale dominante è una matrice per cui*

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

2.1.2 Operazioni sulle matrici

Richiamiamo brevemente la definizione delle operazioni definite fra vettori e matrici.

Definizione 2.1.16 *Somma, differenza di due vettori* $\underline{x}, \underline{y} \in R^n$ *e' un vettore* $\underline{z} \in R^n$ *i cui elementi sono dati da* $z_i = x_i \pm y_i$

Definizione 2.1.17 *Somma, differenza di due matrici* $A, B \in R^{m,n}$ *e' una matrice* $C \in R^{m,n}$ *i cui elementi sono dati da* $c_{ij} = a_{ij} \pm b_{ij}$.

Definizione 2.1.18 *Prodotto scalare fra due vettori* $\underline{x}, \underline{y} \in R^n$ *è lo scalare* $s = \underline{y}^T \underline{x} = \sum_{i=1}^{n} x_i y_i$. *Il costo computazionale per ottenere* s *è di* n *flops (per flops si intende un prodotto + una somma).*

Definizione 2.1.19 *Prodotto fra una matrice* $A \in R^{m,n}$ *ed un vettore* $\underline{x} \in R^n$ *è un vettore* $\underline{y} \in R^m$ *i cui elementi sono dati da* $y_i = \sum_{k=1}^{n} a_{ik} * x_k$. *Il costo computazionale per ottenere* $\underline{y}$ *è di* mn *flops.*

Definizione 2.1.20 *Prodotto fra due matrici* $A \in R^{m,n}$ *e* $B \in R^{n,p}$ *è una matrice* $C \in R^{m,p}$ *i cui elementi sono dati da* $c_{ij} = \sum_{k=1}^{n} a_{ik} * b_{kj}$. *Il costo computazionale per ottenere* C *è di* mnp *flops.*

Il prodotto fra matrici quadrate non è, in generale, commutativo $AB \neq BA$.

2.2 Metodi diretti (M.F.)

Data una matrice quadrata A, dovendo risolvere il sistema lineare

$$A\underline{x} = \underline{b} \tag{2.2}$$

è noto, dai corsi di Analisi e Geometria, che la condizione $\det(A) \neq 0$ è equivalente all'esistenza della matrice inversa A^{-1} di A per cui

$$\underline{x} = A^{-1}\underline{b}. \tag{2.3}$$

La determinazioine di $\underline{x}$, soluzione di (2.2), nota la matrice inversa ha un costo computazionale di n^2 flops.

Prima di addentrarci nel calcolo della soluzione di (2.2), quando l'inversa non è nota, è istruttivo dare una chiara rappresentazione geometrica di cosa vuol dire risolvere un sistema lineare. E' chiaro che una matrice A applicata ad un vettore $\underline{x}$ lo trasforma in un vettore $\underline{y}$, è naturale chiedersi: tutti i vettori $\underline{y}$ sono immagine di qualche vettore $\underline{x}$ mediante la A? La risposta è sì, se e solo se $\det(A) \neq 0$ (o equivalentemente se $\exists A^{-1}$).

La ricerca della soluzione di un sistema lineare può quindi essere vista come la ricerca di una n-pla di numeri (le componenti del vettore $\underline{x}$) che entrano nella combinazine lineare delle colonne della matrice A ($\underline{A}_i$) per rappresentare il termine noto (vettore $\underline{b}$) , ovvero

$$x_1\underline{A}_1 + x_2\underline{A}_2 + x_3\underline{A}_3 + ... + x_n\underline{A}_n = \underline{b}.$$

L'esistenza di una soluzione di un sistema lineare è quindi legata al fatto che il vettore $\underline{b}$ appartenga allo spazio generato dalle colonne della matrice A:

Definizione 2.2.1 *Spazio delle colonne (range di A)*

$$R(A) := \left\{\underline{y} \in R^n | \underline{y} = A\underline{x}, \quad \forall \underline{x} \in R^n\right\}.\square$$

Per l'unicità della soluzione è importante la definizione di spazio nullo (nucleo) di una matrice:

Definizione 2.2.2 *Spazio nullo (null di A)*

$$N(A) := \{\underline{x} \in R^n | A\underline{x} = \underline{0}\}.\square$$

E' facile far vedere che se $\widetilde{\underline{x}}$ è soluzione di un sistema lineare e $\widehat{\underline{x}} \neq \underline{0}$ appartiene allo spazio nullo allora anche il vettore ($\widetilde{\underline{x}}+ \widehat{\underline{x}}$) è soluzione dello stesso sistema lineare ($A\,\widetilde{\underline{x}} = \underline{b},\ A\widehat{\underline{x}} = \underline{0}, \longrightarrow A(\widetilde{\underline{x}}+ \widehat{\underline{x}}) = \underline{b} + \underline{0} = \underline{b}$).

Esempio 2.2.1 *Come esempio si considerino la matrice A ed il vettore* $\underline{b}$

$$A = \begin{bmatrix} 1 & 1 \\ 2 & 2 \end{bmatrix}; \quad \underline{b} = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

la matrice A è singolare ed il suo spazio delle colonne coincide con i punti della retta $y_2 = 2y_1$*, per cui il sistema* $A\underline{x} = \underline{b}$ *non ammette soluzione essendo* $\underline{b} \notin R(A)$*. Lo spazio nullo di A è formato dai punti della retta* $x_2 = -x_1$*, infatti* $A \begin{bmatrix} x_1 \\ -x_1 \end{bmatrix} = \underline{0}$*, ovvio.*$\square$

Se la matrice A è diagonale con elementi diagonali tutti diversi da zero è immediato risolvere il corrispondente sistema lineare (disaccoppiato) con n divisioni.

Se la matrice A è triangonale superiore (inferiore) con elementi diagonali tutti diversi da zero è possibile risolvere il sistema lineare associato con la *sostituzione all'indietro* (*in avanti*) (back/forward-substitution) con un costo dell'ordine di $\frac{n^2}{2}$ flops.

Osservazione 2.2.1 *Per flops si intende una moltiplicazione (divisione) più una somma (sottrazione) in virgola mobile (floating-point). In algebra lineare è naturale questa unità di misura in quanto il prodotto scalare, che è alla base del calcolo matriciale, costa proprio n prodotti ed $n-1$ addizioni (n flops).*

Si fa osservare che in entrambi i casi il sistema è non singolare essendo il determinante diverso da zero (prodotto degli elementi diagonali).

Il problema è quindi passare da un sistema con matrice piena ad un sistema equivalente con matrice diagonale o triangolare. Gauss fornì per primo un tale algoritmo combinando in modo opportuno le righe della matrice e il termine noto.

2.2.1 Il metodo di eliminazione di Gauss

L'idea di Gauss si basa sulla semplice osservazione che il vettore soluzione verifica tutte le equazioni del sistema lineare e quindi verifica anche un' equazione ottenuta combinando linearmente fra loro due di tali equazioni. E' quindi possibile passare da un sistema "pieno" ad uno equivalente triangolare in $n-1$ passi eliminando al j-esimo passo l'incognita x_j da tutte le equazioni dalla (j+1)-esima fino alla n-esima. Schematicamente scriviamo la (2.2) nella forma:

$$\begin{cases} a_{11}x_1+ & a_{12}x_2+ & ...+ & a_{1n}x_n = & b_1 \\ a_{21}x_1+ & a_{22}x_2+ & ...+ & a_{2n}x_n = & b_2 \\ ... & ... & ... & ... & ... \\ a_{n1}x_1+ & a_{n2}x_2+ & ...+ & a_{nn}x_n = & b_n \end{cases}.$$

1° passso Si moltiplica la prima riga per $\left(-\frac{a_{21}}{a_{11}}\right)$ e si sostituisce la seconda riga con la somma delle prime due, si moltiplica la prima riga per $\left(-\frac{a_{31}}{a_{11}}\right)$ e si sostituisce la terza riga con la somma fra la prima e la terza, ... si

moltiplica la prima riga per $\left(-\frac{a_{n1}}{a_{11}}\right)$ e si sostituisce la n-esima riga con la somma fra la prima e la n-esima, giungendo a

$$\begin{cases} a_{11}x_1+ & a_{12}x_2+ & ...+ & a_{1n}x_n = & b_1 \\ & a^1_{22}x_2+ & ...+ & a^1_{2n}x_n = & b^1_2 \\ & ... & ... & ... & ... \\ & a^1_{n2}x_2+ & ...+ & a^1_{nn}x_n = & b^1_n \end{cases}$$

2° passo Si opera analogamente sul sottosistema

$$\begin{cases} a^1_{22}x_2+ & ...+ & a^1_{2n}x_n = & b^1_2 \\ ... & ... & ... & ... \\ a^1_{n2}x_2+ & ...+ & a^1_{nn}x_n = & b^1_n \end{cases}$$

usando come moltiplicatori $\left(-\frac{a^1_{32}}{a^1_{22}}\right)$ ... $\left(-\frac{a^1_{n2}}{a^1_{22}}\right)$ si otterrà

$$\begin{cases} a^1_{22}x_2+ & a^1_{23}x_3+ & ...+ & a^1_{2n}x_n = & b^1_2 \\ & a^2_{33}x_3+ & ...+ & a^2_{3n}x_n = & b^2_3 \\ & ... & ... & ... & ... \\ & a^2_{n3}x_3+ & ...+ & a^2_{nn}x_n = & b^2_n \end{cases}$$

j° passo Si opera analogamente sul sottosistema

$$\begin{cases} a^{j-1}_{jj}x_j+ & ...+ & a^{j-1}_{jn}x_n = & b^{j-1}_j \\ ... & ... & ... & ... \\ a^{j-1}_{nj}x_j+ & ...+ & a^{j-1}_{nn}x_n = & b^{j-1}_n \end{cases}$$

usando come moltiplicatori $\left(-\frac{a^{j-1}_{j+1j}}{a^{j-1}_{jj}}\right)$ ... $\left(-\frac{a^{j-1}_{nj}}{a^{j-1}_{jj}}\right)$.

In n-1 passi si arriva (se tutti gli $a^{j-1}_{jj} \neq 0$) ad un sistema equivalente triangolare superiore. Il costo computazionale dell'algoritmo di Gauss è $\frac{n^3}{3}$ flops; infatti sono necessarie $(n-1)(n+2)$ flops al primo passo, $(n-2)(n+1)$ al secondo, $(n-3)n$ al terzo, fino a $1*4$ al passo (n-1)-esimo, per cui in totale

$$\begin{aligned} flops &= \sum_{i=1}^{n-1} i(i+3) = \sum_{i=1}^{n-1} i^2 + 3\sum_{i=1}^{n-1} i = \\ &= \frac{n(n-1)(2n-1)}{6} + 3\frac{n(n-1)}{2} = \frac{n^3}{3} + O(n^2). \end{aligned}$$

Vale la pena di ricordare che la risoluzione di un sistema lineare mediante la regola di Cramer (calcolo di $n+1$ determinanti) utilizzando la definizione nel calcolo dei determinanti ($\det(A) = \sum(-1)^s a_{1s_1}a_{2s_2}...a_{ns_n}$, $(s_1, s_2, ..., s_n)$ permutazione degli indici $(1, 2, ..., n)$, s numero di scambi fra $(s_1, s_2, ..., s_n)$ e $(1, 2, ..., n)$) ha una complessità dell'ordine di $(n+1)(n-1)n!$ flops.

Su un elaboratore che esegue 100Megaflops al secondo la risoluzione di un sistema lineare di 20 equazioni in 20 incognite richiede circa

$$\begin{array}{ll} 2.7 * 10^{-5} \text{ sec} & \text{con l'algoritmo di Gauss} \\ 9.2 * 10^{12} \text{ sec} & \text{con la regola di Cramer} \end{array}$$

sfortunatamente 10^{12} sec sono circa 31709 anni !!

Nell'algoritmo di Gauss abbiamo supposto che ad ogni passo fosse ben definito il fattore moltiplicativo $-\frac{a_{rj}^{j-1}}{a_{jj}^{j-1}}$, $(r = j+1, ..., n)$ il che equivale a supporre $a_{jj}^{j-1} \neq 0$, $\forall j$. Se $a_{jj}^{j-1} = 0$, per qualche j, l'algoritmo di Gauss *naturale* si blocca. E' possibile modificare l'algoritmo eseguendo la ricerca dell'elemento di massimo modulo ($\neq 0$) fra i restanti elementi della colonna (*pivoting parziale*) e, se esiste, scambiare fra loro le righe j ed s (se s è la riga contenente il massimo). Questa variante non richiede ulteriori flops per cui non aumenta la complessità computazionale. Va osservato che se tutti gli elementi della j-esima colonna sono nulli (partendo dalla j-esima riga) allora il sistema non è determinato (il determinante di A è uguale a zero). La tecnica del pivoting parziale è stabile computazionalmente per cui viene sempre effettuata la ricerca del massimo anche se $a_{jj}^{j-1} \neq 0$.

2.2.2 Decomposizione LU

Il metodo di Gauss permette, se tutti i minori principali di testa della matrice A sono diversi da zero (il che è equivalente a dire $a_{jj}^{j-1} \neq 0$, $\forall j$, provarlo per esercizio), di decomporre la matrice A nel prodotto di due matrici, L triangolare inferiore (con 1 sulla diagonale principale) ed U triangolare superiore

$$A = LU. \tag{2.4}$$

La matrice U coincide con la vecchia matrice triangolare superiore del sistema equivalente, mentre

$$L^{-1} = L_{n-1}L_{n-2}L_{n-3}...L_2L_1$$

è ottenuta moltiplicando le matrici L_j del j-esimo passo del processo di Gauss

$$L_j = \begin{bmatrix} 1 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & \ddots & 0 & & & \vdots \\ 0 & \ddots & 1 & \ddots & & \vdots \\ 0 & & -\frac{a_{j+1j}^{j-1}}{a_{jj}^{j-1}} & 1 & 0 & \vdots \\ 0 & & & \ddots & \ddots & 0 \\ 0 & \cdots & -\frac{a_{nj}^{j-1}}{a_{jj}^{j-1}} & \cdots & 0 & 1 \end{bmatrix}.$$

Con il metodo di Gauss si ha quindi

$$L_{n-1}L_{n-2}L_{n-3}...L_2L_1A = L^{-1}A = U \tag{2.5}$$

dalla (2.5) segue immediatamente la (2.4). Come si calcola la matrice L? Essendo, verificarlo per esercizio, $L_j^{-1} = 2I - L_j$, si ha che la matrice $L = L_1^{-1}L_2^{-1}L_3^{-1}...L_{n-1}^{-1}$ è, senza eseguire alcuna operazione,

$$L = \begin{bmatrix} 1 & 0 & \cdots & \cdots & \cdots & 0 \\ +\frac{a_{21}}{a_{11}} & \ddots & 0 & & & \vdots \\ +\frac{a_{31}}{a_{11}} & \ddots & 1 & \ddots & & \vdots \\ \vdots & +\frac{a_{jj-1}^{j-2}}{a_{j-1j-1}^{j-2}} & +\frac{a_{j+1j}^{j-1}}{a_{jj}^{j-1}} & 1 & 0 & \vdots \\ +\frac{a_{n-11}}{a_{11}} & \vdots & & \ddots & \ddots & 0 \\ +\frac{a_{n1}}{a_{11}} & \cdots & +\frac{a_{nj}^{j-1}}{a_{jj}^{j-1}} & \cdots & +\frac{a_{nn-1}^{n-1}}{a_{nn}^{n-1}} & 1 \end{bmatrix} \tag{2.6}$$

(la (2.6) si ottiene semplicemente tenendo conto della particolare struttura delle matrici L_j^{-1}).

La seguente matrice

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 2 \end{bmatrix}$$

non ammette decomposizione LU ($a_{11} = 0$, primo minore principale di testa nullo), è facile osservare che, se si scambiano la prima e la seconda riga, la matrice A diviene la matrice $U = \begin{bmatrix} 1 & 2 \\ 0 & 1 \end{bmatrix}$, per cui la decomposizione è implicita ($L = I$). Possiamo quindi formulare il seguente

Teorema 2.2.1 *Data una matrice A, se* $\det(A) \neq 0$, *esiste una matrice P di permutazione per cui*

$$PA = LU.$$

Dimostrazione 2.2.1 *La dimostrazione del 2.2.1 è costruttiva (coincide con il metodo di Gauss con pivoting parziale).*■

L'importanza della decomposizione $PA = LU$ nasce dal fatto che non è necessario conoscere il vettore dei termini noti quando si esegue la fase di decomposizione. La risoluzione di un sistema lineare può quindi essere schematizzata nel modo seguente

1. No pivoting

$$\left.\begin{array}{l} A\underline{x} = \underline{b} \\ A = LU \end{array}\right\} \longrightarrow LU\underline{x} = \underline{b} \longrightarrow \left\{\begin{array}{l} L\underline{y} = \underline{b} \\ U\underline{x} = \underline{y} \end{array}\right. ,$$

2. pivoting

$$\left.\begin{array}{l} A\underline{x} = \underline{b} \\ PA = LU \end{array}\right\} \longrightarrow LU\underline{x} = P\underline{b} \longrightarrow \left\{\begin{array}{l} L\underline{y} = P\underline{b} \\ U\underline{x} = \underline{y} \end{array}\right. .$$

Si fa osservare che la parte computazionalmente onerosa è la decomposizione LU (n^3 flops), mentre la risoluzione dei 2 sistemi triangolari, parte destra dello schema, costa solo n^2 flops. La fase di determinazione del vettore $P\underline{b}$, nel secondo schema, costa zero flops (sono solo scambi di righe!).

Anche la costruzione della matrice P è immediata e può essere fatta utilizzando solo un vettore di n interi. Precisamente, definito

$$sp = [1, 2, 3, ..., n-1, n] \tag{2.7}$$

il vettore dei primi n interi ($sp(k) = k$), per costruire la matrice P basta spostare, al j-esimo passo dell'eliminazione di Gauss, il contenuto della cella j con quello della cella r del vettore sp (dove r è l'indice di riga del pivot al passo j). Il vettore definito in (2.7) rappresenta, in modo compatto, la matrice identità (1 in posizione $(k, sp(k)), \quad k = 1, 2, ...n$, 0 altrove), analogamente la matrice P avrà 0 ovunque e 1 solo nelle posizioni $(k, sp(k)), \quad k = 1, 2, ...n$.

2.2.3 Decomposizione di Cholesky

Se la matrice A è simmetrica e definita positiva è possibile dimostrare l'esistenza della seguente decomposizione (decomposizione di *Cholesky*)

$$A = V^T V \tag{2.8}$$

dove V è una matrice triangolare superiore con elementi diagonali positivi.

La dimostrazione dell'esistenza di tale decomposizione è costruttiva, e viene qui riportata perchè evidenzia come sia proprio la decomposizione di Cholesky la via per determinare se una matrice data A è definita positiva. Essendo

$$V^T = \begin{bmatrix} v_{11} & & & & \\ v_{21} & v_{22} & & & \\ \vdots & \vdots & \ddots & & \\ v_{n-11} & v_{n-12} & \cdots & v_{n-1n-1} & \\ v_{n1} & v_{n2} & \cdots & v_{nn-1} & v_{nn} \end{bmatrix};$$

$$V = \begin{bmatrix} v_{11} & v_{21} & \cdots & v_{n-11} & v_{n1} \\ & v_{22} & \cdots & v_{n-12} & v_{n2} \\ & & \ddots & \vdots & \vdots \\ & & & v_{n-1n-1} & v_{nn-1} \\ & & & & v_{nn} \end{bmatrix},$$

dalla (2.8) si ha, eseguendo il prodotto righe per colonne, partendo dalla prima riga

$$v_{11}v_{11} = v_{11}^2 = a_{11} \rightarrow v_{11} = \sqrt{a_{11}}$$

$$v_{11}v_{j1} = a_{1j} = a_{j1} \rightarrow v_{j1} = \frac{a_{j1}}{v_{11}},\ j = 2, 3, ..., n$$

per la seconda riga, v_{j1} sono ora noti $\forall j \geq 1$, si ha

$$v_{21}^2 + v_{22}^2 = a_{22} \rightarrow v_{22} = \sqrt{a_{22} - v_{21}^2}$$

$$v_{21}v_{j1} + v_{22}v_{j2} = a_{2j} = a_{j2} \rightarrow v_{j2} = \frac{a_{j2} - v_{21}v_{j1}}{v_{22}},\ j = 3, 4, ..., n$$

in generale al passo k-esimo si ottiene

$$v_{kk} = \sqrt{a_{kk} - \sum_{m=1}^{k-1} v_{mk}^2}, \quad k = 1, ..., n \tag{2.9}$$

$$v_{ki} = \frac{a_{ki} - \sum_{m=1}^{k-1} v_{km} v_{mi}}{v_{kk}}, \quad k = 1, ..., n-1; \; i = k+1, ..., n \tag{2.10}$$

nelle (2.9,2.10) la sommatoria va ignorata se il valore d'arrivo è minore del valore di partenza.

Il costo computazionale della decomposizioni di Cholesky è $\frac{n^3}{6}$ flops (oltre a n estrazioni di radice quadrata). Proprio la necessità di dover estrarre la radice quadrata permette di verificare se la matrice di partenza era definita positiva, infatti se un radicando risulta negativo l'algoritmo si interrompe (non esiste la decomposizione per cui la matrice di partenza non era definita positiva). E' facile dimostrare che, ad ogni passo k, il radicando è dato dal quoziente fra il minore principale di testa, della matrice di partenza, di ordine k e quello di ordine $k-1$.

Si può osservare che nelle (2.9,2.10) i prodotti scalari indicati possono essere eseguiti in doppia precisione, pur lasciando le variabili in semplice, aumentando così l'accuratezza del risultato (algoritmo stabile oltre che efficiente).

E' evidente che una volta nota la decomposizione di Cholesky il sistema lineare

$$A\underline{x} = \underline{b}$$

può essere risolto mediante il seguente schema

$$\left.\begin{array}{l} A\underline{x} = \underline{b} \\ A = V^T V \end{array}\right\} \longrightarrow V^T V \underline{x} = \underline{b} \longrightarrow \left\{\begin{array}{l} V^T y = \underline{b} \\ V\underline{x} = \underline{y} \end{array}\right. .$$

2.3 Analisi dell'errore (M.F.)

La soluzione del sistema lineare $A\underline{x} = \underline{b}$ calcolata con uno dei metodi proposti (chiamiamola $\underline{x}_c$) differirà, per l'inevitabile presenza di errori (cfr. capitolo 1), dalla soluzione vera (chiamiamola $\underline{x}_v$), quanto grande sarà questa differenza? Dipenderà solo dal metodo usato? Dipenderà dalla matrice del sistema lineare?

Prima di rispondere a queste domande è necessario definire una misura di distanza fra vettori e matrici.

2.3.1 Richiami sulle norme di vettore

Definizione 2.3.1 *Dato un vettore $\underline{x}$ si definisce norma del vettore $\underline{x}$ ($\|\underline{x}\|$) una funzione a valori non negativi che gode delle seguenti proprietà*

1. $\|\underline{x}\| \geq 0 \ \forall \underline{x} \in R^n, \ \|\underline{x}\| = 0 \Leftrightarrow \underline{x} = \underline{0}$
2. $\|\alpha \underline{x}\| = |\alpha| \, \|\underline{x}\| \ \forall \underline{x} \in R^n \ e \ \forall \alpha \in R$
3. $\|\underline{x} + \underline{y}\| \leq \|\underline{x}\| + \|\underline{y}\| \quad \forall \underline{x}, \underline{y} \in R^n$

Tre classici esempi di norma di vettore sono

1. $\|\underline{x}\|_1 = |x_1| + |x_2| + ... + |x_n|$
2. $\|\underline{x}\|_2 = \sqrt{x_1^2 + x_2^2 + ... + x_n^2}$
3. $\|\underline{x}\|_\infty = \max_i \{|x_i|\}$

Analogamente per le matrici

Definizione 2.3.2 *Data una matrice A si definisce norma della matrice A ($\|A\|$) una funzione a valori non negativi che gode delle seguenti proprietà*

1. $\|A\| \geq 0 \ \forall A \in R^{n,n}, \ \|A\| = 0 \Leftrightarrow A = 0$
2. $\|\alpha A\| = |\alpha| \, \|A\| \ \forall A \in R^{n,n} \ e \ \forall \alpha \in R$
3. $\|A + B\| \leq \|A\| + \|B\| \quad \forall A, B \in R^{n,n}$
4. $\|AB\| \leq \|A\|\|B\| \quad \forall A, B \in R^{n,n}$

Norme di matrice possono essere ricavate dalle norme di vettore (*norma indotta o naturale*) ponendo

$$\|A\| = \max_{\underline{x}} \frac{\|A\underline{x}\|}{\|\underline{x}\|} \quad \forall A \in R^{n,n}, \underline{x} \in R^n,$$

norme di vettore e norme di matrice si dicono *compatibili* se

$$\|A\underline{x}\| \leq \|A\|\|\underline{x}\|.$$

Esempi di norme di matrice sono

1. $\|A\|_1 = \max_j \left\{ \sum_{i=1}^{n} |a_{ij}| \right\}$
2. $\|A\|_2 = \sqrt{\rho(A^T A)}$ (dove $\rho(A)$ è il raggio spettrale di A).
3. $\|A\|_\infty = \max_i \left\{ \sum_{j=1}^{n} |a_{ij}| \right\}$
4. $\|A\|_E = \sqrt{\sum_{j=1}^{n} \sum_{i=1}^{n} a_{ij}^2}$ (questa norma è compatibile con la norma 2 di vettore ma non è indotta).

2.3.2 Stima dell'errore

Dato il sistema lineare $A\underline{x} = \underline{b}$, detta $\underline{x}_c$ la soluzione calcolata si può determinare il residuo

$$\underline{r} = A\underline{x}_c - \underline{b},$$

essendo $\underline{x}_v = A^{-1}\underline{b}$, posto $\underline{e} = \underline{x}_c - \underline{x}_v$, si ha

$$\underline{r} = A\underline{x}_c - \underline{b} = A\underline{x}_c - A\underline{x}_v = A(\underline{x}_c - \underline{x}_v) = A\underline{e}$$

per cui

$$\underline{e} = A^{-1}\underline{r}$$

da cui segue

$$\|\underline{e}\| = \|\underline{x}_c - \underline{x}_v\| = \|A^{-1}\underline{r}\| \leq \|A^{-1}\|\|\underline{r}\| \tag{2.11}$$

la (2.11) non è molto significativa in quanto anche se $\|\underline{r}\|$ è piccola l'errore può essere grande se $\|A^{-1}\|$ è grande. Quello che si può dire è che se $\|\underline{r}\|$ è grande allora la soluzione calcolata non è "buona".

Possiamo dimostrare il seguente

Teorema 2.3.1 *Dato il sistema $A\underline{x} = \underline{b}$, con A matrice non singolare, ed il sistema $A(\underline{x} + \delta\underline{x}) = \underline{b} + \delta\underline{b}$, si ha*

$$\frac{1}{K(A)} \frac{\|\delta\underline{b}\|}{\|\underline{b}\|} \leq \frac{\|\delta\underline{x}\|}{\|\underline{x}\|} \leq K(A) \frac{\|\delta\underline{b}\|}{\|\underline{b}\|} \tag{2.12}$$

dove $K(A) = \|A\|\|A^{-1}\|$ è il numero di condizionamento della matrice A.

Dimostrazione 2.3.1 *Valgono le seguenti 4 uguaglianze*

$$\begin{array}{lll} (1)\ A\underline{x} = \underline{b} & \longrightarrow & (2)\ \underline{x} = A^{-1}\underline{b} \\ (3)\ A\delta\underline{x} = \delta\underline{b} & \longrightarrow & (4)\ \delta\underline{x} = A^{-1}\delta\underline{b} \end{array}$$

per cui, usando la 2 e la 3, si ha

$$\left.\begin{array}{rcl} \|\underline{x}\| = \|A^{-1}\underline{b}\| & \leq & \|A^{-1}\|\|\underline{b}\| \\ \|\delta\underline{b}\| = \|A\delta\underline{x}\| & \leq & \|A\|\|\delta\underline{x}\| \end{array}\right\} \longrightarrow \|\underline{x}\|\|\delta\underline{b}\| \leq K(A)\|\underline{b}\|\|\delta\underline{x}\|$$

da cui

$$\frac{1}{K(A)}\frac{\|\delta\underline{b}\|}{\|\underline{b}\|} \leq \frac{\|\delta\underline{x}\|}{\|\underline{x}\|}.$$

Analogamente, usando la 1 e la 4, si ha

$$\left.\begin{array}{rcl} \|\underline{b}\| = \|A\underline{x}\| & \leq & \|A\|\|\underline{x}\| \\ \|\delta\underline{x}\| = \|A^{-1}\delta\underline{b}\| & \leq & \|A^{-1}\|\|\delta\underline{b}\| \end{array}\right\} \longrightarrow \|\delta\underline{x}\|\|\underline{b}\| \leq K(A)\|\delta\underline{b}\|\|\underline{x}\|$$

da cui

$$\frac{\|\delta\underline{x}\|}{\|\underline{x}\|} \leq K(A)\frac{\|\delta\underline{b}\|}{\|\underline{b}\|}$$

che prova la (2.12). ■

Si può dimostrare un risultato più generale se si considera una variazione anche sugli elementi della matrice.

Osservazione 2.3.1 *Valgono le seguenti relazioni*

1. $K(A) = K(A^{-1})$;
2. $K(A)$ dipende dalla norma scelta;
3. $K(A) \geq 1$, infatti $1 = \|I\| = \|AA^{-1}\| \leq \|A\|\|A^{-1}\| = K(A)$.■
4. Le matrici ortogonali ($Q^TQ = QQ^T = I$) hanno $K_2(A) = 1$. Infatti essendo $\|A\|_2 = \sqrt{\rho(A^TA)}$ si ha

$$\|Q\|_2 = \sqrt{\rho(Q^TQ)} = \sqrt{\rho(I)} = 1.$$■

5. Se A è simmetrica ($A = A^T$) allora $K_2(A) = \left|\frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}\right|$. Infatti si può facilmente dimostrare che $\|A\|_2 = |\lambda_{\max}(A)|$

$$\|A\|_2 = \sqrt{\rho(A^T A)} = \sqrt{\rho(A^2)} = \sqrt{\lambda_{\max}^2} = |\lambda_{\max}(A)|$$

per cui, essendo $\lambda(A^{-1}) = \frac{1}{\lambda(A)}$,

$$\|A^{-1}\|_2 = \frac{1}{|\lambda_{\min}(A)|}.\blacksquare$$

6. Un classico esempio di matrice mal condizionata è la matrice di Hilbert di ordine n

$$H_n = \begin{bmatrix} 1 & \frac{1}{2} & \cdots & \frac{1}{n-1} & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n} & \frac{1}{n+1} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \frac{1}{n-1} & \frac{1}{n} & \cdots & \frac{1}{2n-3} & \frac{1}{2n-2} \\ \frac{1}{n} & \frac{1}{n+1} & \cdots & \frac{1}{2n-2} & \frac{1}{2n-1} \end{bmatrix}.$$

2.4 Metodi iterativi (M.F.)

Quando la matrice A del sistema lineare (2.2) risulta *sparsa* e *non strutturata* (quindi con un numero di elementi diversi da zero dell'ordine di n e non disposti a banda o con altra particolare struttura) può essere conveniente l'uso di *metodi iterativi*. Il vantaggio nasce dal fatto che i metodi iterativi, sfruttando la sparsità della matrice, eseguono il prodotto matrice vettore in meno di n^2 flops, per cui possono essere assai efficienti. Per contro i metodi diretti, applicati a matrici sparse non strutturate, generano il *fill-in* (riempimento) della matrice durante la fase di eliminazione, per cui hanno una complessità computazionale che è sempre di n^3 flops. Poichè la soluzione del sistema di partenza è ottenuta come limite di una successione

$$\underline{x}_{k+1} = B\underline{x}_k + \underline{g} \tag{2.13}$$

dove la matrice d'iterazione B è scelta in modo che il sistema

$$\underline{x} = B\underline{x} + \underline{g} \tag{2.14}$$

sia equivalente al sistema (2.2), bisognerà garantire non solo la convergenza della (2.13) alla soluzione $\underline{x}$, ma anche essere in grado di stimare l'errore che si commette arrestandosi ad una data iterazione.

2.4.1 Raffinamento iterativo di Wilkinson

Un primo metodo, proposto da Wilkinson, viene utilizzato anche quando la matrice A non è sparsa per ottenere una miglior stima della soluzione del sistema (2.2) calcolata con un metodo diretto.

Sia $\underline{x}_c$ la soluzione del sistema (2.2) calcolata, per esempio, mediante l'algoritmo di Gauss, calcoliamo il residuo

$$\underline{r} = A\underline{x}_c - \underline{b} \tag{2.15}$$

in doppia precisione (l'uso di una precisione superiore nel calcolo del residuo è fondamentale per non rischiare di ottenere un residuo identicamente nullo), sia $\underline{e}_c$ la soluzione del sistema

$$A\underline{e} = \underline{r}$$

calcolata utilizzando la stessa decomposizione LU della matrice del sistema. Si può osservare che, se i calcoli sono eseguiti "senza errori" allora

$$\underline{x}^v = \underline{x}_c - \underline{e}_c = \underline{x}_c - A^{-1}\underline{r} = \underline{x}_c - A^{-1}(A\underline{x}_c - \underline{b}) = A^{-1}\underline{b}$$

di fatto, a causa degli inevitabili errori dovuti alla finitezza della macchina, si ha

$$\|\underline{x}_{vera} - \underline{x}_c\| > \|\underline{x}_{vera} - \underline{x}^v\|$$

ed, in generale, sono sufficienti 2 o 3 iterazioni per raggiungere la soluzione nella precisione di macchina.

Se $\|\underline{e}_c\|$ non decresce vuol dire che la matrice del sistema è mal condizionata. Se è nota la decomposizione LU di A il metodo non è molto costoso (bastano solo n^2 flops per iterazione).

2.4.2 Metodo di Jacoby

Si trasforma il sistema (2.2) nella forma equivalente (2.14) utilizzando il seguente *splitting* (partizionatura) della matrice $A = L + D + U$, dove L è triangolare inferiore con zeri sulla diagonale principale, D è diagonale e U è triangolare superiore con zeri sulla diagonale principale. La matrice B ed il vettore $\underline{g}$ sono quindi dati da

$$B = -D^{-1}(L + U); \quad \underline{g} = D^{-1}\underline{b}.$$

Il metodo diventa quindi

$$\underline{x}_{k+1} = -D^{-1}(L+U)\underline{x}_k + D^{-1}\underline{b}. \tag{2.16}$$

Per l'equivalenza con il sistema (2.2), detta $\underline{x}_{vera}$ la soluzione del sistema, si avrà, dalla (2.16)

$$\underline{x}_{vera} = -D^{-1}(L+U)\underline{x}_{vera} + D^{-1}\underline{b}$$

o, equivalentemente,

$$\underline{x}_{vera} = (I-B)^{-1}\underline{g}.$$

La convergenza del metodo di Jacoby, ed in generale di tutti i metodi della forma (2.13), dipenderà quindi dalle proprietà di B, più precisamente vale il seguente

Teorema 2.4.1 *Condizione necessaria e sufficiente per la convergenza del metodo (2.13) è che* $\rho(B) < 1$.

Dimostrazione 2.4.1 *Preso un vettore iniziale* $\underline{x}_0$ *arbitrario la (2.13) ci fornisce*

$$\begin{aligned}
\underline{x}_1 &= B\underline{x}_0 + \underline{g} \\
\underline{x}_2 &= B\underline{x}_1 + \underline{g} = B^2\underline{x}_0 + (B+I)\underline{g} \\
\underline{x}_3 &= B\underline{x}_2 + \underline{g} = B^3\underline{x}_0 + (B^2+B+I)\underline{g} \\
&\cdots\cdots \\
\underline{x}_k &= B\underline{x}_{k-1} + \underline{g} = B^k\underline{x}_0 + (B^{k-1}+\cdots+B+I)\underline{g}
\end{aligned} \tag{2.17}$$

vale il seguente lemma (la cui dimostrazione è lasciate come esercizio).

Lemma 2.4.2 *Se e solo se* $\rho(B) < 1$ *allora*

$$\begin{aligned}
\lim_{k\to\infty} B^k &= 0 \\
\lim_{k\to\infty} \left(\sum_{i=0}^{k} B^i\right) &= I + B + \cdots = (I-B)^{-1}.\blacksquare
\end{aligned}$$

per cui, dalle (2.17), utilizzando il lemma (2.4.2) si ha

$$k \to \infty, \quad \underline{x}_{vera} = (I-B)^{-1}\underline{g}.\blacksquare$$

Osservazione 2.4.1 *Ricordando che, data una matrice A, per ogni definizione di norma si ha*

$$|\lambda(A)| \leq \|A\|$$

vale il seguente

Teorema 2.4.3 *Condizione sufficiente per la convergenza della sucessione (2.13) è che esista una norma di B per cui* $\|B\| < 1$.■

Oltre a garantire la convergenza del metodo è importante poter stimare l'errore che si commette fermandosi alla k-esima delle iterazioni (2.13) (errore di troncamento). Vale il seguente risultato

Teorema 2.4.4 *Se* $\|B\| < 1$ *allora per l'errore alla k-esima iterazione si ha*

$$\begin{aligned} \|\underline{x}_{vera} - \underline{x}_k\| &\leq \|B\|^k \|\underline{x}_{vera} - \underline{x}_0\| \\ \|\underline{x}_{vera} - \underline{x}_k\| &\leq \frac{\|B\|^k}{1 - \|B\|} \|\underline{x}_1 - \underline{x}_0\| \end{aligned}$$

Dimostrazione 2.4.2 *Per la prima disuguaglianza si ha*

$$\begin{aligned} \underline{x}_k &= B\underline{x}_{k-1} + \underline{g} \\ \underline{x}_{vera} &= B\underline{x}_{vera} + \underline{g} \end{aligned}$$

sottraendo

$$\underline{x}_{vera} - \underline{x}_k = B(\underline{x}_{vera} - \underline{x}_{k-1})$$

e quindi

$$\underline{x}_{vera} - \underline{x}_k = B^k(\underline{x}_{vera} - \underline{x}_0)$$

passando alle norme

$$\|\underline{x}_{vera} - \underline{x}_k\| = \left\|B^k(\underline{x}_{vera} - \underline{x}_0)\right\| \leq \left\|B^k\right\| \|\underline{x}_{vera} - \underline{x}_0\| . \tag{2.18}$$

Per la seconda disuguaglianza, osserviamo che

$$\begin{aligned} \|\underline{x}_{vera} - \underline{x}_0\| &= \|\underline{x}_{vera} - \underline{x}_1 + \underline{x}_1 - \underline{x}_0\| \leq \|\underline{x}_{vera} - \underline{x}_1\| + \|\underline{x}_1 - \underline{x}_0\| \\ &\leq \|B\| \|\underline{x}_{vera} - \underline{x}_0\| + \|\underline{x}_1 - \underline{x}_0\| \end{aligned}$$

essendo $\|B\| < 1$, *si ha*

$$\begin{aligned} (1 - \|B\|) \|\underline{x}_{vera} - \underline{x}_0\| &\leq \|\underline{x}_1 - \underline{x}_0\| \\ \|\underline{x}_{vera} - \underline{x}_0\| &\leq \frac{1}{1 - \|B\|} \|\underline{x}_1 - \underline{x}_0\| \end{aligned} \tag{2.19}$$

dalle (2.18,2.19) si ha

$$\|\underline{x}_{vera} - \underline{x}_k\| \leq \frac{\|B\|^k}{1 - \|B\|} \|\underline{x}_1 - \underline{x}_0\| . \blacksquare \tag{2.20}$$

Osservazione 2.4.2 *Nella (2.20) la quantità $\|\underline{x}_1 - \underline{x}_0\|$ è nota dopo il primo passo per cui si può calcolare quanti passi saranno necessari per ottenere una prefissata accuratezza essendo $\|B\| < 1$.*

2.4.3 Metodo di Gauss-Seidel

Sia dato il sistema

$$A\underline{x} = \underline{b}$$

e la matrice A sia partizionata come

$$A = L + D + U$$

la j-esima componente del vettore $\underline{x}_k$ ($\underline{x}_k^j$) fornita dal metodo di Jacoby è data da

$$D_{jj}\underline{x}_k^j = -L_j\underline{x}_{k-1} - U_j\underline{x}_{k-1} + \underline{b}^j, \quad k = 1, 2, \cdots \tag{2.21}$$

dove L_j e U_j sono le righe j-esime delle matrici L e U. Essendo le componenti del vettore L_j

$$L_{ji} = 0, \quad i \geq j$$

nella (2.21) al generico passo k-esimo, il primo vettore $\underline{x}_{k-1}$ può essere sostituito dal vettore $\underline{x}_k$ in quanto sono già state calcolate le prime (j-1) componenti del vettore $\underline{x}_k$ stesso. In questo modo si tiene subito conto delle variazioni apportate nell'iterazione. Si osservi che le restanti $n - j + 1$ componenti, non ancora note, vengono moltiplicate per zero. Si ottiene quindi il metodo (di Gauss-Seidel)

$$D_{jj}\underline{x}_k^j = -L_j\underline{x}_k - U_j\underline{x}_{k-1} + \underline{b}^j, \quad k = 1, 2, \cdots, n$$

che, con ovvio significato dei simboli, può essere scritto in forma matriciale

$$D\underline{x}_k = -L\underline{x}_k - U\underline{x}_{k-1} + \underline{b}.$$

La matrice d'iterazione del metodo ed il vettore $\underline{g}$ sono quindi

$$B = -(D + L)^{-1}U; \quad \underline{g} = (D + L)^{-1}\underline{b}$$

per cui sono applicabili, per la convergenza del metodo, i teoremi (2.4.1,2.4.3). Si può osservare, analizzando la struttura dei due metodi, che se entrambi i metodi convergono, in generale, il metodo di Gauss-Seidel convergerà più rapidamente del metodo di Jacoby.

Diamo ora due teoremi di convergenza

Teorema 2.4.5 *Se la matrice A del sistema lineare (2.2) è fortemente diagonalizzata allora i metodi di Jacoby e Gauss-Seidel convergono per ogni scelta del vettore iniziale.*

Dimostrazione 2.4.3 *La dimostrazione, per il metodo di Jacoby, è immediata ricordando che, essendo A fortemente diagonalizzata, allora*

$$|a_{ii}| > \sum_{\substack{j=1 \\ j\neq i}}^{n} |a_{ij}|\,, \quad \forall i = 1, 2, \cdots, n$$

per cui $\|B\|_\infty < 1$.■

Teorema 2.4.6 *Se la matrice A del sistema lineare (2.2) è simmetrica e definita positiva allora il metodo di Gauss-Seidel converge per ogni scelta del vettore iniziale.*■

2.4.4 Metodi di rilassamento SOR

Si basano sulle seguenti due osservazioni:

1. la velocità di convergenza di un metodo iterativo della forma (2.13) dipende dal raggio spettrale della matrice d'iterazione B, più precisamente più $\rho(B)$ è prossimo a zero tanto più velocemente converge il metodo (2.13) (se $\rho(B) = 0$ il metodo converge al più in n passi, diventa un metodo diretto);

2. se la matrice B dipende da un parametro ω ($B = B(\omega)$) si può pensare di trovare un valore ottimale di ω (ω_{opt}) per cui il raggio spettrale risulti minimo ($\rho(B(\omega_{opt})) < \rho(B(\omega)) < 1$).

Partendo dal metodo di Gauss-Seidel, scritto nella forma

$$\underline{x}_{k+1} = D^{-1}\left[-L\underline{x}_{k+1} - U\underline{x}_k + \underline{b}\right]$$

sottraendo ad entrambi i membri $\underline{x}_k$ si ha

$$\underline{x}_{k+1} - \underline{x}_k = D^{-1}\left[-L\underline{x}_{k+1} - U\underline{x}_k + \underline{b}\right] - \underline{x}_k \tag{2.22}$$

definito il vettore $\underline{v}_k$

$$\underline{v}_k = D^{-1}\left[-L\underline{x}_{k+1} - U\underline{x}_k + \underline{b}\right] - \underline{x}_k$$

la (2.22) si può scrivere, introducendo il parametro $\omega > 0$, nella forma

$$\underline{x}_{k+1} = \underline{x}_k + \omega\underline{v}_k. \tag{2.23}$$

Nella (2.23) il vettore variato è ottenuto dal vettore precedente muovendosi nella direzione di $\underline{v}_k$ con un passo pari a $\omega \|\underline{v}_k\|$; l'importante sarà trovare il passo ω in modo che $\underline{x}_{k+1}$ sia più vicino a $\underline{x}_{vera}$ di quanto non lo sia $\underline{x}_k$. In forma matriciale la (2.23) è equivalente a

$$\underline{x}_{k+1} - \underline{x}_k = \omega D^{-1}\left[-L\underline{x}_{k+1} - (D+U)\underline{x}_k + \underline{b}\right]$$

ovvero

$$(D + \omega L)\underline{x}_{k+1} = \left[(1-\omega)D - \omega U\right]\underline{x}_k + \omega\underline{b}$$

e in definitiva

$$\underline{x}_{k+1} = (D + \omega L)^{-1}\left[(1-\omega)D - \omega U\right]\underline{x}_k + \omega\underline{b}. \tag{2.24}$$

Nella (2.24) la matrice d'iterazione è quindi

$$B(\omega) = (D + \omega L)^{-1}\left[(1-\omega)D - \omega U\right]$$

che si riduce, per $\omega = 1$, alla matrice del metodo di Gauss-Seidel, mentre per la convergenza del metodo è necessario (non sufficiente) che $0 < \omega < 2$. Se ω è costante ad ogni passo si ha un *metodo stazionario*, se viene variato ad ogni iterazione si ha un metodo *non stazionario* (si cambia il passo). La ricerca del valore ottimale di ω dipende da problema a problema e non può essere qui trattata.

2.4.5 Metodi non stazionari

L'idea alla base dei metodi non stazionari è quella di cercare ad ogni iterazione di muoversi lungo un'*opportuna direzione* con un *opportuno passo* in modo

che l'iterata (k+1)-esima sia "più vicina" alla soluzione dell'iterata k-esima. Dedurremo i metodi nel caso di matrici simmetriche e definite positive.

Definiamo l'errore alla generica iterazione k

$$\underline{e}_k = \underline{x}_{vera} - \underline{x}_k$$

ed associamogli la seguente forma quadratica definita positiva

$$\begin{aligned} \phi(\underline{e}_k) &= (A\underline{e}_k, \underline{e}_k) = \underline{e}_k^T A \underline{e}_k = (\underline{x}_{vera} - \underline{x}_k)^T A (\underline{x}_{vera} - \underline{x}_k) \\ &= \underline{x}_{vera}^T A \underline{x}_{vera} - 2\underline{x}_{vera}^T A \underline{x}_k + \underline{x}_k^T A \underline{x}_k \end{aligned} \tag{2.25}$$

in modo analogo si definisce $\phi(\underline{e}_{k+1})$. Cerchiamo, analogamente a quanto fatto in (2.23),

$$\underline{x}_{k+1} = \underline{x}_k + \alpha \underline{v}_k. \tag{2.26}$$

Utilizzando le (2.25,2.26) si può cercare α e $\underline{v}_k$ in modo che

$$\phi(\underline{e}_{k+1}) < \phi(\underline{e}_k),$$

e la succesione (2.26) converga, per $k \longrightarrow \infty$, a $\underline{x}_{vera}$.

Per la determinazione di α vale il seguente

Teorema 2.4.7 *Se A è simmetrica e definita positiva*

$$\phi(\underline{e}_{k+1}) < \phi(\underline{e}_k),$$

per ogni α dato da

$$\alpha = \theta \frac{\underline{r}_k^T \underline{v}_k}{\underline{v}_k^T A \underline{v}_k}; \quad 0 < \theta < 2. \tag{2.27}$$

dove $\underline{r}_k^T$ è il residuo (2.15).

Dimostrazione 2.4.4 *Dalle (2.25,2.26) si ha*

$$\begin{aligned} \phi(\underline{e}_{k+1}) &= (\underline{x}_{vera} - \underline{x}_{k+1})^T A (\underline{x}_{vera} - \underline{x}_{k+1}) \\ &= \underline{x}_{vera}^T A \underline{x}_{vera} - 2\underline{x}_{vera}^T A \underline{x}_{k+1} + \underline{x}_{k+1}^T A \underline{x}_{k+1} \\ &= \underline{x}_{vera}^T A \underline{x}_{vera} - 2\underline{x}_{vera}^T A (\underline{x}_k + \alpha \underline{v}_k) + (\underline{x}_k + \alpha \underline{v}_k)^T A (\underline{x}_k + \alpha \underline{v}_k) \\ &= \phi(\underline{e}_k) - 2\alpha \underline{x}_{vera}^T A \underline{v}_k + 2\alpha \underline{x}_k^T A \underline{v}_k + \alpha^2 \underline{v}_k^T A \underline{v}_k \\ &= \phi(\underline{e}_k) - 2\alpha \underline{e}_k^T A \underline{v}_k + \alpha^2 \underline{v}_k^T A \underline{v}_k. \end{aligned}$$

Perchè sia

$$\phi(\underline{e}_{k+1}) < \phi(\underline{e}_k),$$

deve essere

$$\alpha^2 \underline{v}_k^T A \underline{v}_k - 2\alpha \underline{e}_k^T A \underline{v}_k < 0 \tag{2.28}$$

ovvero, essendo $A\underline{e}_k = \underline{r}_k$,

$$\alpha = \theta \frac{\underline{e}_k^T A \underline{v}_k}{\underline{v}_k^T A \underline{v}_k} = \theta \frac{\underline{r}_k^T \underline{v}_k}{\underline{v}_k^T A \underline{v}_k}. \blacksquare$$

Osservazione 2.4.3 *La condizione su* θ *è la stessa data su* ω *nei metodi SOR. E' necessaria per la convergenza, ma non sufficiente (bisogna ancora scegliere la direzione lungo cui muoversi). Il valore di* $\theta = 1$ *è quello che offre, in generale, il miglior guadagno essendo corrispondente al valore minimo della (2.28).*

Metodo delle coordinate

Il vettore $\underline{v}_k$ che compare nella (2.26) viene scelto (*metodo delle coordinate ciclico*) ciclicamente uguale al versore $\underline{e}_j$

$$\underline{v}_k = \underline{e}_j, \quad k = j \bmod n$$

mentre α, dato dalla (2.27), risulta uguale a

$$\alpha = \theta \frac{\underline{r}_k^j}{a_{jj}} \tag{2.29}$$

dove $\underline{r}_k^j$ è la componente j-esima di $\underline{r}_k$. Con queste scelte è garantita la convergenza del metodo.

Se il vettore $\underline{v}_k$ che compare nella (2.26) viene scelto uguale al versore $\underline{e}_j$, dove j è l'indice della massima componente in modulo di $\underline{r}_k$, ed α è dato dalla (2.29), questo è equivalente ad annullare, al passo k-esimo, la massima delle componenti del residuo, si ha il *metodo delle coordinate NON ciclico*. La convergenza NON è più garantita (potrebbero esserci componenti di $\underline{x}_k$ che non vengono mai modificate) per ripristinare la convergenza bisogna garantire che dopo m passi (ovviamente $m > n$) tutte le componenti di $\underline{x}_k$ siano state modificate. Se ciò non avviene si devono modificare ciclicamente le componenti non variate.

Metodo del gradiente

Si sceglie come direzione $\underline{v}_k$ che compare nella (2.26) il vettore residuo $\underline{r}_k$ che risulta essere il gradiente della forma quadratica (2.25) (da qui il nome del metodo). Lo scalare α, dato dalla (2.27), risulterà quindi uguale a

$$\alpha = \theta \frac{\underline{r}_k^T \underline{r}_k}{\underline{r}_k^T A \underline{r}_k}.$$

Vale la pena di osservare che la velocità di convergenza del metodo del gradiente dipende dal valore

$$\frac{K(A) - 1}{K(A) + 1}$$

per cui la convergenza può essere molto lenta se A è mal condizionata.

2.5 Sistemi sovradeterminati (M.F.)

In svariati problemi si ha a che fare con modelli lineari in cui è importante individuare dei parametri (vettore delle incognite $\underline{x}$), potendo disporre di un numero di osservazioni (vettore dei termini noti $\underline{b}$) superiore al numero dei parametri, per cui la matrice del sistema lineare (A) risulta rettangolare (sistemi sovradeterminati). Le osservazioni sono sovente affette da errori (errori di misura) per cui non esiste una soluzione, in senso classico, del problema (il vettore termini noti non appartiene allo spazio delle colonne di A).

Dato il sistema lineare

$$A\underline{x} = \underline{b}$$

con $A \in R^{mn}$, $\underline{x} \in R^{n1}$ e $\underline{b} \in R^{m1}$, $m > n$, si cerca il vettore $\underline{x}$ che verifica

$$\min_{\underline{x}} \|A\underline{x} - \underline{b}\|_2^2 \tag{2.30}$$

($\underline{x}$ è soluzione nel senso dei *minimi quadrati*).

Vale il seguente

Teorema 2.5.1 *Se* $\exists \underline{y}^*$, *con* $\underline{y}^* = A\ \underline{x}^*$, *tale per cui*

$$(\underline{b} - \underline{y}^*)^T \underline{y} = 0, \quad \forall \underline{y} \in R(A),$$

(il vettore $(\underline{b} - \underline{y}^*)$ *risulta ortogonale a tutti i vettori di* $R(A)$*) allora* $\underline{x}^*$ *è una soluzione di (2.30).*

Dimostrazione 2.5.1 *Utilizzeremo la tecnica del completamento del quadrato*

$$0 \leq \left\|\underline{b}-\underline{y}\right\|_2^2 = (\underline{b}-\underline{y})^T(\underline{b}-\underline{y}) = \underline{b}^T\underline{b} - 2\underline{b}^T\underline{y} + \underline{y}^T\underline{y}$$

aggiungendo $2(\underline{b}-\underline{y}^*)^T\underline{y} = 0$*, per ipotesi, si ha*

$$\begin{aligned} 0 &\leq \underline{b}^T\underline{b} - 2\underline{b}^T\underline{y} + \underline{y}^T\underline{y} + 2(\underline{b}-\underline{y}^*)^T\underline{y} \\ &= \underline{b}^T\underline{b} + 2\underline{b}^T\underline{y} - 2\underline{y}^{*T}\underline{y} - 2\underline{b}^T\underline{y} + \underline{y}^T\underline{y} \\ &= \underline{b}^T\underline{b} - 2\underline{y}^{*T}\underline{y} + \underline{y}^T\underline{y} \end{aligned}$$

aggiungendo e togliendo $\underline{y}^{*T}\underline{y}^*$

$$\begin{aligned} 0 &\leq \underline{b}^T\underline{b} - \underline{y}^{*T}\underline{y}^* + (\underline{y}^{*T}\underline{y}^* - 2\underline{y}^{*T}\underline{y} + \underline{y}^T\underline{y}) \\ &= \|\underline{b}\|_2^2 - \left\|\underline{y}^*\right\|_2^2 + (\underline{y}^*-\underline{y})^T(\underline{y}^*-\underline{y}) \end{aligned}$$

in conclusione

$$0 \leq \left\|\underline{b}-\underline{y}\right\|_2^2 = \|\underline{b}\|_2^2 - \left\|\underline{y}^*\right\|_2^2 + (\underline{y}^*-\underline{y})^T(\underline{y}^*-\underline{y})$$

che risulta minima quando $\underline{y} = \underline{y}^*$*.*■

Osservazione 2.5.1 *Risolvere un sistema lineare nel senso dei minimi quadrati è quindi equivalente a trovare un vettore* $\underline{x}$ *la cui immagine, mediante* A*, è il vettore* $\underline{y}$ *proiezione (ortogonale) del vettore termine noto* $\underline{b}$ *sullo spazio delle colonne di* A*. Se* A *ha rango massimo tale soluzione sarà unica.*

Per determinare il vettore $\underline{x}^*$ soluzione, osserviamo che, in virtù del teorema (2.5.1) e della proprietà commutativa del prodotto scalare, si ha

$$(\underline{b}-\underline{y}^*)^T\underline{y} = \underline{y}^T(\underline{b}-\underline{y}^*) = 0, \quad \forall \underline{y} \in R(A),$$

ed, essendo

$$\underline{y}^* = A\underline{x}^*, \qquad \underline{y} = A\underline{x}$$

si ha

$$\begin{aligned} \underline{y}^T(\underline{b}-\underline{y}^*) &= \underline{x}^TA^T(\underline{b}-A\underline{x}^*) = 0, \quad \forall \underline{x} \in R^n \\ 0 &= \underline{x}^T(A^T\underline{b} - A^TA\underline{x}^*) \end{aligned}$$

da cui

$$A^T\underline{b} - A^TA\underline{x}^* = \underline{0}$$

e quindi

$$A^T A\underline{x}^* = A^T\underline{b} \tag{2.31}$$

che prende il nome di *equazione normale*. Se A è a rango massimo allora A^TA è invertibile, per cui

$$A^+ = (A^TA)^{-1}A^T$$

e

$$\underline{x}^* = A^+\underline{b}$$

è l'unica soluzione (la matrice A^+ prende il nome di *pseudo inversa* di Moore-Penrose).

Se A è mal condizionata peggio sarà il condizionamento di A^TA, per cui la risoluzione di (2.31) può risultare poco accurata. Facciamo un breve cenno ad una tecnica risolutiva, utile quando A è a rango massimo ma mal condizionata, che si basa sul seguente teorema

Teorema 2.5.2 *Data una matrice $A \in R^{mn}$, a rango massimo, esistono due matrici, $Q \in R^{mm}$ ortogonale e $R \in R^{mn}$ pseudo-triangolare superiore (il prefisso pseudo per ribadire che è rettangolare), con elementi $\underline{r}_{ii} \neq 0$, tali per cui*

$$A = QR.\blacksquare \tag{2.32}$$

(Forniremo un algoritmo per la determinazione della decomposizione QR nel paragrafo (2.7.3) relativo al calcolo degli autovalori).

Utilizzando il teorema (2.5.2) si ha

$$\begin{aligned}\min_{\underline{x}} \|A\underline{x} - \underline{b}\|_2^2 &= \min_{\underline{x}} \|QR\underline{x} - \underline{b}\|_2^2 = \min_{\underline{x}} \left\|Q(R\underline{x} - Q^T\underline{b})\right\|_2^2 \\ &= \min_{\underline{x}} \left\|R\underline{x} - Q^T\underline{b}\right\|_2^2 = \min_{\underline{x}} \left\|R\underline{x} - \widehat{\underline{b}}\right\|_2^2 .\end{aligned}$$

Scriviamo la matrice R ed il vettore $\widehat{\underline{b}} = Q^T\underline{b}$, nella forma

$$R = \begin{bmatrix} r_{11} & \cdots & r_{1n} \\ 0 & \ddots & \vdots \\ \vdots & \ddots & r_{nn} \\ 0 & \cdots & 0 \\ 0 & \cdots & 0 \end{bmatrix} = \begin{bmatrix} R_{11} \\ 0 \end{bmatrix}; \qquad \widehat{\underline{b}} = \begin{bmatrix} \widehat{b}_1 \\ \vdots \\ \widehat{b}_n \\ \vdots \\ \widehat{b}_m \end{bmatrix} = \begin{bmatrix} \widehat{\underline{b}}_1 \\ \widehat{\underline{b}}_2 \end{bmatrix}$$

dove $R_{11} \in R^{nn}$, $\underline{\widehat{b}}_1 \in R^n$ e $\underline{\widehat{b}}_2 \in R^{m-n}$; è sufficiente quindi risolvere il sistema (quadrato non singolare)

$$R_{11}\underline{x} = \underline{\widehat{b}}_1$$

mantenendo il condizionamento in norma 2 uguale ($K_2(A) = K_2(R) = K_2(R_{11})$).

Se A non è a rango massimo allora A^TA è singolare, si può ancora parlare di soluzione nel senso dei minimi quadrati e di pseudo-inversa, ma l'argomento va oltre i confini di questo corso.

2.6 Riepilogo (M.F.)

Sintetizziamo quanto esposto relativamente alla risoluzione dei sistemi lineari abbiamo

1. Matrici piene (elementi diversi da zero dell'ordine di n^2)
   (a) NON simmetriche: metodo di Gauss e sue varianti (A=LU, PA=LU) con costo di $\frac{n^3}{3}$ flops;
   (b) matrici sparse e strutturate: Gauss, A=LU, PA=LU con costo dipendente dalla ampiezza della banda;
   (c) simmetriche definite positive: decomposizione di Choleshy con costo di $\frac{n^3}{6}$ flops;
2. Matrisci sparse (numero di elementi diversi da zero $<< n^2$)
   (a) NON definite positive: Jacoby, Gauss-Seidel e SOR. Convergenza e velocità di convergenza dipendono da $\rho(B)$.
   (b) simmetriche e definite positive: Coordinate e Gradiente. Velocità di convergenza dipende da $K(A)$.
3. Accuratezza della soluzione dipende dal numero di condizionamento, $K(A) = \|A\| \|A^{-1}\|$.
4. Raffinamento iterativo di Wilkinson per migliorare la soluzione, quando $K(A)$ non è troppo grande.
5. Soluzione nel senso dei minimi quadrati per sistemi sovradeterminati. Decomposizione QR.

2.6.1 Esercizi

Esercizio 2.6.1 *Calcolare, se esiste, la decomposizione LU della seguente matrice*

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 6 & 8 \\ 4 & 3 & 2 & 1 \\ 8 & 6 & 4 & 2 \end{bmatrix}.$$

Giustificare la risposta.

Esercizio 2.6.2 *Risolvere, mediante il metodo di Jacoby, il seguente sistema lineare* $A\underline{x} = \underline{b}$*, dove*

$$A = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ & 2 & 3 & 4 & 5 \\ & & 3 & 4 & 5 \\ & & & 4 & 5 \\ & & & & 5 \end{bmatrix}; \quad \underline{b} = \begin{bmatrix} 3 \\ 2 \\ 4 \\ 1 \\ 5 \end{bmatrix}$$

usando diversi valori per la tolleranza richiesta nel test di arresto e partendo sempre da $\underline{x}_0 = \underline{b}$*. Giustificare i risultati ottenuti.*

Esercizio 2.6.3 *Se una matrice è triangolare allora è ben condizionata ? Giustificare la risposta proponendo anche un esempio.*

Esercizio 2.6.4 *Sia A una matrice simmetrica e fortemente diagonalizzata dimostrare che può essere decomposta nel prodotto*

$$A = LDL^T$$

dove L è triangolare inferiore con 1 sulla diagonale principale e D è diagonale.

Esercizio 2.6.5 *Dati i punti del piano xy, le cui coordinate sono date nella seguente tabella*

x_i	1.1	1.5	1.7	1.8	2
y_i	1.32	2.47	3	3.41	3.99

calcolare la retta di regressione.

2.7 Autovalori di matrici (M.F.)

2.7.1 Richiami e definizioni

Data una matrice quadrata A si definiscono autovalori e autovettori rispettivamente quei valori reali o complessi e quei vettori non nulli che verificano l'equazione

$$A\underline{u} = \lambda\underline{u}. \tag{2.33}$$

Il sistema (2.33) è equivalente al sistema omogeneo

$$(A - \lambda I)\underline{u} = \underline{0}$$

che ammette soluzioni non banali ($\underline{u} \neq \underline{0}$) se e solo se λ è tale per cui

$$\det(A - \lambda I) = p_n(\lambda) = 0$$

dove $p_n(\lambda)$ è un polinomio di grado n.

Teorema 2.7.1 *Se la matrice A è diagonalizzabile allora ammette n autovettori linearmente indipendenti.*

Dimostrazione 2.7.1 *Se A è diagonalizzabile allora*

$$A\underline{u} = C^{-1}DC\underline{u} = \lambda\underline{u}$$

$$DC\underline{u} = \lambda C\underline{u}$$

posto

$$C\underline{u} = \underline{w}$$

segue

$$D\underline{w} = \lambda\underline{w}$$

ed essendo D diagonale $\lambda_i = D_{ii}$ sono autovalori e $\underline{w} = \underline{e}_i$ sono i corrispondenti autovettori, per cui, essendo $\underline{u} = C^{-1}\underline{w}$ anche gli $\underline{u}$ risulteranno linearmente indipendenti.■

Teorema 2.7.2 *Se la matrice B è trasformata per similitudine della matrice A ($B = C^{-1}AC$), allora $\lambda(A) = \lambda(B)$ e $\underline{u}(A) = C\underline{u}(B)$.*

Dimostrazione 2.7.2 *Essendo* $B = C^{-1}AC$, *da* $B\underline{u} = \lambda\underline{u}$ *segue*

$$\begin{aligned} C^{-1}AC\underline{u} &= \lambda\underline{u} \\ AC\underline{u} &= \lambda C\underline{u} \\ A\underline{w} &= \lambda\underline{w}. \blacksquare \end{aligned}$$

Teorema 2.7.3 *Se la matrice* A *ha autovalori* $\lambda(A) \neq 0$, *allora la matrice* A^{-1} *ha autovalori* $\lambda(A^{-1}) = \frac{1}{\lambda(A)}$ *e gli autovettori sono uguali* $\underline{u}(A) = \underline{u}(A^{-1})$.$\blacksquare$

Teorema 2.7.4 *Se la matrice* A *ha autovalori* $\lambda(A)$, *allora la matrice* $(A - \alpha I)$ *ha autovalori* $\lambda(A - \alpha I) = \lambda(A) - \alpha$ *e gli autovettori sono uguali* $\underline{u}(A) = \underline{u}(A - \alpha I)$.$\blacksquare$

Si lascia al lettore, come esercizio, la dimostrazione di questi due ultimi teoremi.

Per il calcolo degli autovalori ed autovettori considereremo due famiglie di metodi: i *metodi locali*, che permettono il calcolo di un particolare autovalore e del corrispondente autovettore, ed i *metodi globali* che, basandosi su trasformazioni per similitudine, permettono di stimare simultaneamente tutti gli autovalori ed autovettori.

Vale la pena di notare che, se si conosce un autovettore della matrice A, è immediato il calcolo del corrispondente autovalore, vale infatti (*quoziente di Rayleigh*)

$$\begin{aligned} A\underline{x} &= \lambda\underline{x} \\ \underline{x}^T A\underline{x} &= \lambda\underline{x}^T\underline{x} \\ \lambda &= \frac{\underline{x}^T A\underline{x}}{\underline{x}^T\underline{x}}. \blacksquare \end{aligned}$$

al contrario, noto λ il calcolo di $\underline{x}$ mediante la definizione (2.33) risulta difficoltoso, se non impossibile, in quanto anche un piccolo errore su λ rende il sistema (2.33) non più omogeneo, per cui solo $\underline{x} = \underline{0}$ verifica la (2.33).

2.7.2 Metodi locali

Fra i metodi locali presenteremo il *metodo delle potenze*, che permette il calcolo dell'autovalore di modulo massimo, ed il *metodo delle potenze inverse*.

Metodo delle potenze

Per semplificare la dimostrazione della convergenza del metodo delle potenze faremo l'ipotesi che

1. la matrice A abbia n autovettori linearmente indipendenti,
2. che esista un solo autovalore di modulo massimo λ_1,
3. il vettore iniziale $\underline{x}_0$, della successione $\underline{x}_{k+1} = A\underline{x}_k$, dipenda da $\underline{u}_1$ autovettore associato a λ_1.

Vale il seguente

Teorema 2.7.5 *Data una matrice A con un solo autovalore di modulo massimo*

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n|$$

e n autovettori $\underline{u}_i$ linearmente indipendenti, posto

$$\underline{x}_0 = \alpha_1\underline{u}_1 + \alpha_2\underline{u}_2 + \cdots + \alpha_n\underline{u}_n = \sum_{i=1}^{n} \alpha_i\underline{u}_i$$

con $\alpha_1 \neq 0$, la successione

$$\underline{x}_{k+1} = A\underline{x}_k \tag{2.34}$$

converge ad $\underline{u}_1$ mentre

$$\beta_k^j = \frac{\underline{x}_{k+1}^j}{\underline{x}_k^j}, \quad \underline{x}_k^j \neq 0$$

converge a λ_1.

Dimostrazione 2.7.3 *Dalla (2.34), per la definizione di autovalore e autovettore, si ha*

$$\begin{aligned}
\underline{x}_1 &= A\underline{x}_0 = \sum_{i=1}^{n} \alpha_i A\underline{u}_i = \sum_{i=1}^{n} \alpha_i\lambda_i\underline{u}_i \\
\underline{x}_2 &= \sum_{i=1}^{n} \alpha_i\lambda_i A\underline{u}_i = \sum_{i=1}^{n} \alpha_i\lambda_i^2\underline{u}_i \\
&\cdots \\
\underline{x}_{k+1} &= \sum_{i=1}^{n} \alpha_i\lambda_i^k A\underline{u}_i = \sum_{i=1}^{n} \alpha_i\lambda_i^{k+1}\underline{u}_i
\end{aligned}$$

ovvero, raccogliendo λ_1^{k+1}

$$\underline{x}_{k+1} = \lambda_1^{k+1}\left\{\alpha_1\underline{u}_1 + \left(\frac{\lambda_2}{\lambda_1}\right)^{k+1}\alpha_2\underline{u}_2 + \cdots + \left(\frac{\lambda_n}{\lambda_1}\right)^{k+1}\alpha_n\underline{u}_n\right\} \qquad (2.35)$$

dalla (2.35) sia ha

$$\lim_{k\to\infty}\underline{x}_{k+1} = \underline{u}_1$$

(si ricordi che gli autovettori sono definiti a meno di una costante).

La (2.35), scritta per k*, diviene*

$$\underline{x}_k = \lambda_1^k\left\{\alpha_1\underline{u}_1 + \left(\frac{\lambda_2}{\lambda_1}\right)^{k}\alpha_2\underline{u}_2 + \cdots + \left(\frac{\lambda_n}{\lambda_1}\right)^{k}\alpha_n\underline{u}_n\right\}$$

facendo il quoziente fra le j*-esime componenti non nulle,*

$$\beta_k^j = \frac{\underline{x}_{k+1}^j}{\underline{x}_k^j} = \lambda_1\frac{\alpha_1\underline{u}_1^j + \left(\frac{\lambda_2}{\lambda_1}\right)^{k+1}\alpha_2\underline{u}_2^j + \cdots}{\alpha_1\underline{u}_1^j + \left(\frac{\lambda_2}{\lambda_1}\right)^{k}\alpha_2\underline{u}_2^j + \cdots}$$

per cui

$$\lim_{k\to\infty}\beta_k^j = \lambda_1.\blacksquare$$

Dalla (2.35) si vede che la velocità di convergenza a $\underline{u}_1$ e λ_1 dipende dalla velocità con cui il quoziente $\frac{\lambda_2}{\lambda_1}$ va a zero (convergenza lineare).

Il fattore β_k può essere calcolato in generale nella forma

$$\beta_k = \frac{\underline{v}_k^T\underline{x}_{k+1}}{\underline{v}_k^T\underline{x}_k}$$

dove $\underline{v}_k$ è un generico vettore (la scelta precedente equivale a $\underline{v}_k = \underline{e}_j$). Se si pone $\underline{v}_k = \underline{x}_k$, si ottiene il quoziente di Rayleigh

$$\beta_k = \frac{\underline{x}_k^T A\underline{x}_k}{\underline{x}_k^T\underline{x}_k} = \frac{\underline{x}_k^T\underline{x}_{k+1}}{\underline{x}_k^T\underline{x}_k}$$

che garantisce una convergenza quadratica, dipendente da $\left(\frac{\lambda_2}{\lambda_1}\right)^2$, se la matrice A è simmetrica.

Vediamo ora se si possono rilassare, o modificare, le ipotesi fatte nel precedente teorema. La lineare indipendenza degli autovettori può essere sostituita con la diagonalizzabilità di A o dall'ipotesi che gli autovalori siano distinti (perchè ?). Il vincolo $\alpha_1 \neq 0$, se rimosso fa si che inizialmente si abbia convergenza su λ_2 (ovviamente se $\alpha_2 \neq 0$), ma con il crescere delle iterazioni, a causa degli inevitabili errori di arrotondamento, si converge ancora a λ_1. Se esistono due autovalori di modulo massimo reali il metodo delle potenze funziona ancora pur di applicarlo alla matrice A^2 (basta ricordare che $\lambda(A^2) = \lambda(A)^2$).

Operativamente, nella pratica, converrà normalizzare i vettori $\underline{x}_k$ al fine di evitare l'overflow.

Un semplice programma in MATLAB per il metodo delle potenze è il seguente

```
x=rand(max(size(A)),1);
x=x/norm(x,2);
betaold=0;
beta=1;
for i=1:100;
  if abs((beta-betaold)/beta) < eps*1e+4
     break
  end
  betaold=beta;
  y=A*x;
  beta=y'*x;
  x=y/norm(y,2);
end
```

Metodo delle potenze inverse

E' una variante del metodo delle potenze che si basa sull'osservazione che, se una matrice è invertibile, allora

$$\lambda(A^{-1}) = \frac{1}{\lambda(A)}$$

ed i corrispondenti autovettori coincidono.

Si può quindi calcolare l'autovalore di modulo minimo di A, ed il corrispondente autovettore, calcolando l'autovalore di modulo massimo di A^{-1}.

Per evitare il calcolo della matrice inversa presente in

$$\underline{x}_{k+1} = A^{-1}\underline{x}_k$$

basta risolvere (per esempio mediante la decomposizione LU) il sistema equivalente

$$A\underline{x}_{k+1} = \underline{x}_k.$$

Il valore

$$\beta_k = \frac{\underline{x}_k^T \underline{x}_{k+1}}{\underline{x}_k^T \underline{x}_k}$$

convergerà al valore

$$\beta = \lim_{k\to\infty} \beta_k = \lambda_{\max}(A^{-1}) = \frac{1}{\lambda_{\min}(A)}$$

mentre $\underline{x}_{k+1}$ convergerà al corrispondente autovettore.

Ricordando inoltre che

$$\lambda(A - \alpha I) = \lambda(A) - \alpha$$

mentre i corrispondenti autovettori coincidono, il metodo delle potenze inverse può essere utilizzato per stimare l'autovalore più prossimo ad un valore prefissato α ed il suo corrispondente autovettore. Posto

$$B = (A - \alpha I)^{-1}$$

il metodo delle potenze applicato alla matrice B è equivalente a

$$(A - \alpha I)\underline{x}_{k+1} = \underline{x}_k$$

per cui il valore

$$\beta_k = \frac{\underline{x}_k^T \underline{x}_{k+1}}{\underline{x}_k^T \underline{x}_k}$$

adesso convergerà al valore

$$\beta = \lim_{k\to\infty} \beta_k = \lambda_{\max}(A - \alpha I)^{-1} = \frac{1}{\lambda_\alpha(A) - \alpha}$$

dove con $\lambda_\alpha(A)$ si è indicato l'autovalore di A più prossimo ad α, ed allora

$$\lambda_\alpha(A) = \frac{1}{\beta} + \alpha.$$

Per poter applicare il metodo delle potenze inverse con shift è necessario avere una idea di dove sono dislocati, nel piano complesso, gli autovalori di A. A tale fine richiamiamo, senza dimostrazione, alcuni classici risultati.

Teorema 2.7.6 *Se una matrice A è simmetrica allora i suoi autovalori sono reali.*

Teorema 2.7.7 *Se una matrice A è simmetrica e definita positiva allora i suoi autovalori sono reali e positivi.*

Teorema 2.7.8 *Per ogni autovalore di A vale la limitazione*

$$|\lambda(A)| \leq \|A\|$$

per ogni norma di A.

Teorema 2.7.9 *(di Gerschgorin) Data la matrice A di elementi a_{ij}, definiamo*

$$r_k = \sum_{\substack{j=1 \\ j\neq k}}^{n} |a_{kj}|\,; \qquad c_k = \sum_{\substack{j=1 \\ j\neq k}}^{n} |a_{jk}|$$

$$\begin{aligned} R_k &= \{z|\,|z - a_{kk}| \leq r_k\} \\ C_k &= \{z|\,|z - a_{kk}| \leq c_k\} \end{aligned}$$

gli autovalori di A appartengono ai cerchi R_k e C_k. Più precisamente appartengono alla regione del piano complesso che è data dall'intersezione delle unioni dei due insiemi di cerchi.■

Sebbene grossolane, le limitazioni fornite dai precedenti teoremi, in alcuni casi risultano sufficienti per stimare alcuni autovalori ed autovettori della matrice A.

2.7.3 Metodi globali

Sono quei metodi che permettono di calcolare tutti gli autovalori della matrice A trasformando, mediante trasformazioni per similitudine, la matrice in una matrice in forma diagonale o triangolare (forme per le quali è immediato il calcolo degli autovalori). E' importante osservare che tali trasformazioni porteranno, in generale per $n \geq 5$, ad una matrice triangolare o diagonale solo dopo un "numero infinito" di passi. (E' una naturale conseguenza del teorema di Abel sul calcolo delle radici di equazioni algebriche).

Le matrici di trasformazione, che stanno alla base di questi metodi, sono le stesse che permettono la decomposizione QR di una matrice o la sua trasformazione in forma di Hessemberg.

Fra le diverse tecniche di trasformazione presenteremo in dettaglio quella basata sulle *matrici di riflessione* di Householder che, a differenza delle matrici di rotazione, riflettono i vettori a cui sono applicate rispetto degli iper-piani.

Matrici di Householder

Definizione 2.7.1 *Dato un vettore $\underline{u}$ definiamo matrice di Householder la matrice*

$$Q = I - \frac{2\underline{u}\underline{u}^T}{\|\underline{u}\|_2^2}.$$

Teorema 2.7.10 *La matrice Q è simmetrica, ortogonale ed effettua la riflessione di un vettore rispetto all'iper-piano ortogonale al vettore $\underline{u}$.*

Dimostrazione 2.7.4 *La simmetria segue dalla definizione di Q. Per l'ortogonalità*

$$\begin{aligned} Q^TQ &= \left(I - \frac{2\underline{u}\underline{u}^T}{\|\underline{u}\|_2^2}\right)^T \left(I - \frac{2\underline{u}\underline{u}^T}{\|\underline{u}\|_2^2}\right) = \left(I - \frac{2\underline{u}\underline{u}^T}{\|\underline{u}\|_2^2}\right)^2 = \\ &= I - \frac{4\underline{u}\underline{u}^T}{\|\underline{u}\|_2^2} + \frac{4\underline{u}\left(\underline{u}^T\underline{u}\right)\underline{u}^T}{\|\underline{u}\|_2^4} = I, \end{aligned}$$

essendo $\left(\underline{u}^T\underline{u}\right) = \|\underline{u}\|_2^2$.

Sia $\underline{z}$ un vettore dell'iper-piano ortogonale ad $\underline{u}$, ovvero $\underline{z}^T\underline{u} = 0$, allora

$$Q\underline{z} = \left(I - \frac{2\underline{u}\underline{u}^T}{\|\underline{u}\|_2^2}\right)\underline{z} = \underline{z}$$

mentre

$$Q\underline{u} = \left(I - \frac{2\underline{u}\underline{u}^T}{\|\underline{u}\|_2^2}\right)\underline{u} = -\underline{u}$$

essendo $\underline{x} \equiv \alpha\underline{u} + \beta\underline{z}$, $\forall\underline{x} \in R^n$, (il vettore $\underline{x}$ è decomposto nelle sue due componenti rispetto a $\underline{u}$ e $\underline{z}$), allora

$$Q\underline{x} = Q(\alpha\underline{u} + \beta\underline{z}) = -\alpha\underline{u} + \beta\underline{z}$$

risulta il riflesso di $\underline{x}$ rispetto all'iper-piano ortogonale ad $\underline{u}$.■

Nasce spontanea la domanda: è possibile costruire delle matrici Q che applicate a sinistra ad una matrice A (eventualmente anche rettangolare) generino una matrice triangolare superiore ? Questa operazione è simile alla decomposizione LU ma adesso la matrice sinistra è ortogonale. La risposta è affermativa basta osservare che è possibile (mediante una opportuna Q) rendere nulle tutte le componenti di un dato vettore da un certo indice in poi, mantenendone inalterata la norma 2. In definitiva vale il seguente

Teorema 2.7.11 *Dato un vettore $\underline{v}$ arbitrario, per ottenere il vettore $\underline{w} = Q\underline{v}$ della forma*

$$\underline{w} = \begin{cases} w_j = v_{j;} & j = 1, \cdots, k-1 \\ w_k | \, \|\underline{w}\| = \|\underline{v}\| \, ; & \\ w_j = 0; & j = k+1, \cdots, n \end{cases} \tag{2.36}$$

basta prendere $\underline{u}$ della forma

$$\underline{u} = \begin{cases} u_j = 0 & j = 1, \cdots, k-1 \\ u_k = v_k \pm \sqrt{\sum_{j=k}^{n} v_j^2}; & \\ u_j = v_j; & j = k+1, \cdots, n \end{cases} . \tag{2.37}$$

Dimostrazione 2.7.5 *Si ha*

$$\begin{aligned} \|\underline{u}\|_2^2 &= \underline{u}^T \underline{u} = v_k^2 + \sum_{j=k}^{n} v_j^2 \pm 2 v_k \sqrt{\sum_{j=k}^{n} v_j^2} + \sum_{j=k+1}^{n} v_j^2 = \\ &= 2 \sum_{j=k}^{n} v_j^2 \pm 2 v_k \sqrt{\sum_{j=k}^{n} v_j^2}; \end{aligned}$$

e

$$\begin{aligned} \underline{u}^T \underline{v} &= v_k^2 \pm v_k \sqrt{\sum_{j=k}^{n} v_j^2} + \sum_{j=k+1}^{n} v_j^2 = \\ &= \sum_{j=k}^{n} v_j^2 \pm v_k \sqrt{\sum_{j=k}^{n} v_j^2} = \frac{\|\underline{u}\|_2^2}{2}; \end{aligned}$$

essendo $\underline{w} = Q\underline{v}$*, si ha*

$$\underline{w} = Q\underline{v} = \left(I - \frac{2\underline{u}\underline{u}^T}{\|\underline{u}\|_2^2}\right)\underline{v} = \underline{v} - \frac{2\underline{u}\left(\underline{u}^T\underline{v}\right)}{\|\underline{u}\|_2^2} = \underline{v} - \underline{u}.$$

che è della forma (2.36).■

La fattorizzare QR di una matrice A si può quindi ottenere in $n-1$ passi costruendo le opportune matrci Q_j di Householder che, al j-esimo passo, annullano gli elementi della j-esima colonna della matrice A^j sotto la diagonale principale, dove con $A^j = Q_{j-1}Q_{j-2}\cdots Q_1 A$, abbiamo indicato la matrice al j-esimo passo e $A^1 = A$,

$$Q_{n-1}Q_{n-2}\cdots Q_2 Q_1 A = R.$$

Osservazione 2.7.1 *Il prodotto* $Q\underline{y}$ *può essere eseguito con n flops al posto di* n^2 *osservando che*

$$Q\underline{y} = \left(I - \frac{2\underline{u}\underline{u}^T}{\|\underline{u}\|_2^2}\right)\underline{y} = \underline{y} - \frac{2\underline{u}^T\underline{y}}{\|\underline{u}\|_2^2}\underline{u} = \underline{y} - \gamma\underline{u}$$

essendo

$$\gamma = \frac{2\underline{u}^T\underline{y}}{\|\underline{u}\|_2^2}.$$

Osservazione 2.7.2 *Nella (2.37) il segno* $\pm$ *viene scelto in modo da ridurre la cancellazione numerica (+ se* $v_k > 0$*, - se* $v_k < 0$*).*

Algoritmo QR

Data la fattorizzazione

$$A = A_0 = Q_0 R_0 \tag{2.38}$$

si consideri la matrice

$$A_1 = R_0 Q_0 \tag{2.39}$$

è immediato dimostrare che la matrice A_1 è simile alla A, infatti

$$A_1 = R_0 Q_0 = \left(Q_0^T Q_0\right) R_0 Q_0 = Q_0^T \left(Q_0 R_0\right) Q_0 = Q_0^T A_0 Q_0.$$■

Iterando la decomposizione (2.38) ed il prodotto (2.39) si genera una succesione di matrici A_k per la quali vale il seguente

Teorema 2.7.12 *Se la matrice A ha autovalori reali distinti, $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_n|$, allora*

$$\lim_{k\to\infty} A_k = R$$

dove R è triangolare superiore, se A è simmetrica $R = D$ diagonale.■

Poichè l'algoritmo QR salva la forma di Hessemberg di una matrice, è conveniente trasformare preventivamente la matrice A in forma di Hessemberg (tridiagonale se A è simmetrica). Tale trasformazione può essere effettuata, mantenendo la similitudine, ancora con matrici di Householder in $n-2$ passi. Questa volta si dovrà moltiplicare, sia a sinistra che a destra, per le matrici Q_j costruite in modo di azzerare gli elementi della colonna j-esima della matrice A^j sotto la sotto-diagonale principale. Senza entrare nei dettagli (che vengono lasciati al lettore come esercizio di verifica)

$$H = Q_{n-2}Q_{n-3}\cdots Q_2Q_1AQ_1Q_2\cdots Q_{n-3}Q_{n-2}. \tag{2.40}$$

Osservazione 2.7.3 *La (2.40) è ovviamente una trasformazione per similitudine essendo le Q_j ortogonali e simmetriche.*

La velocità di convergenza dell'algoritmo QR dipende dallo spettro della matrice A per cui, nella pratica, si preferisce utilizzare al posto delle (2.38,2.39) le

$$A - \alpha I = A_0 - \alpha I = Q_0R_0$$

e

$$A_1 = R_0Q_0 + \alpha I$$

dove α è un opportuno valore di shift, scelto per accelerare la velocità di convergenza del metodo.

2.7.4 Analisi dell'errore

Analogamente a quanto fatto per la risoluzione di un sistema lineare è naturale chiedersi di quanto può variare la stima di un autovalore a fronte di una variazione degli elementi della matrice. Vale il seguente

Teorema 2.7.13 *(di Bauer-Fike) Se μ è un autovalore della matrice $(A + E)$ e*

$$X^{-1}AX = D = diag(\lambda_1, \lambda_2, \cdots, \lambda_n)$$

allora

$$\min |\lambda - \mu| \leq K_p(X) \|E\|_p$$

dove $\|\bullet\|_p$ *denota una norma p di matrice.*■

E' interessante osservare che è il condizionamento della matrice X che diagonalizza la A che influenza il calcolo degli autovalori e non il condizionamento di A.

Se A è simmetrica, essendo diagonalizzabile mediante una matrice ortogonale, non crea problemi per il calcolo degli autovalori (si pensi alla matrice di Hilbert, così perfida per la risoluzione dei sistemi lineari, ma di cui si riescono a calcolare bene gli autovalori).

Essendo le colonne della matrice X gli autovettori della matrice A, il teorema di Bauer-Fike evidenzia come siano gli autovettori ad influire sul calcolo degli autovalori (più gli angoli formati da questi sono prossimi a $\frac{\pi}{2}$ e più le cose vanno bene).

2.8 Riepilogo (M.F.)

Sintetizzando quanto esposto relativamente al calcolo di autovalori ed autovettori abbiamo

1. Metodi locali: permettono il calcolo di un autovalore. Metodo delle potenze e sue varianti
   (a) autovalori distinti non creano problemi (autovettori linearmente indipendenti);
   (b) velocità di convergenza dipende da $\frac{\lambda_2}{\lambda_1}$, matrici simmetriche $\left(\frac{\lambda_2}{\lambda_1}\right)^2$;
   (c) shift per accellerare convergenza e calcolare autovettore associato ad un dato autovalore.
2. Metodi globali: tutti gli autovalori. Trasformazioni per similitudine. Algoritmo QR.
   (a) velocità di convergenza legato allo spettro della matrice;
   (b) shift per accellerare convergenza.
3. Condizionamento: dipende dagli autovettori; non ci sono problemi per le matrici simmetriche essendo gli autovettori ortogonali.

2.8.1 Esercizi

Esercizio 2.8.1 *Dato il vettore* $\underline{x} = [1, 2, 3, 4, 5, 6]$ *costruire la matrice* Q *di Householder in modo che* $\underline{y} = Q\underline{x} = [1, 2, y_3, 0, 0, 0]$.

Esercizio 2.8.2 *Utilizzare il metodo delle potenze per calcolare gli autovalori di modulo massimo e minimo della matrice di Hilbert di ordine 6, e quindi il suo condizionamento in norma 2.*

Esercizio 2.8.3 *Sapendo che* $\underline{e}_3 = [0, 0, 1, 0, 0]^T$ *è autovettore della matrice dell'esercizio 2.6.2 calcolare il corrispondente autovalore utilizzando, nel quoziente di Rayleigh,* $\underline{e}_3$ *ed* $\underline{e}_3 + \delta\underline{e}_3$*, dove* $\delta\underline{e}_3$ *è una perturbazione di* $\underline{e}_3$*. Provare per varie perturbazioni* $\delta\underline{e}_3$ *e commentare i risultati ottenuti.*

Capitolo 3

Zeri di funzioni (M.Frontini)

La risoluzione di molti problemi conduce al calcolo degli zeri di una funzione, si veda per esempio il problema della determinazione del tasso di crescita nel modello di Maltius presentato nell'introduzione, oppure il problema della ricerca del minimo (massimo) di una funzione e tutti quei problemi legati alla massimizzazione del profitto o minimizzazione della spesa.

Considereremo in dettaglio il caso monodimensionale estendendo, ove possibile, alcuni risultati al caso di sistemi non lineari. Partiremo quindi dal problema: data $f(x) \in C^0[a,b]$ con $f(a)f(b) < 0$, trovare $\xi \in [a,b]$ tale che $f(\xi) = 0$.

Un primo algoritmo per la determinazione di ξ ci viene fornito dal teorema degli zeri visto nel corso di Analisi.

Teorema 3.0.1 *(degli zeri) Sia $f(x) \in C^0[a,b]$ con $f(a)f(b) < 0$, allora esiste (almeno) uno $\xi \in (a,b)$ tale che $f(\xi) = 0$.*■

Se alle ipotesi del teorema degli zeri si aggiunge la monotonia della $f(x)$ si ha anche l'unicità del punto ξ.

3.1 Metodo di bisezione (M.F.)

Sotto le ipotesi precedenti definiamo la seguente sequenza di operazioni:

$$x_0 = a, \qquad x_1 = b, \qquad i = 1$$

$$\begin{aligned} x_{i+1} &= \frac{x_i + x_{i-1}}{2}, \qquad valuta \quad f(x_{i+1}) \\ f(x_{i+1}) &= 0 \rightarrow \xi = x_{i+1} \qquad FINE. \\ altrimenti \quad f(x_i)f(x_{i+1}) &< 0 \rightarrow x_{i-1} = x_{i+1}; \quad i = i+1 \\ altrimenti \quad f(x_{i-1})f(x_{i+1}) &< 0 \rightarrow x_i = x_{i+1}; \quad i = i+1 \\ & \quad continua. \end{aligned}$$

Nella pratica non si arriverà mai ad avere $f(x_{i+1}) = 0$, per cui nella sequenza precedente bisogna introdurre un opportuno test di arresto in modo d'ottenere un algoritmo utilizzabile. Un buon test è il seguente

$$|x_i - x_{i-1}| < toll \tag{3.1}$$

dove *toll* è la precisione richiesta sul risultato. Essendo $\xi \in (x_i, x_{i-1})$ la (3.1) fornisce l'intervallo di indeterminazione di ξ.

Osservazione 3.1.1 *Nel calcolo del punto medio è più stabile l'uso della seguente formula*

$$x_{i+1} = x_i + \frac{x_{i-1} - x_i}{2},$$

che permette di evitare l'overflow ed assicura, come deve essere, che $x_{i-1} \leq x_{i+1} \leq x_i$ *(*$x_i \leq x_{i+1} \leq x_{i-1}$*).*

E' possibile sapere a priori il numero di iterazioni necessarie per raggiungere la tolleranza richiesta essendo

$$x_i - x_{i-1} = \frac{b-a}{2^{i-1}}$$

per cui, dalla (3.1),

$$i \geq 1 + \frac{\log(\frac{b-a}{toll})}{\log 2}.$$

Il costo computazionale del metodo è dato dal calcolo, ad ogni passo, della $f(x_{i+1})$. La convergenza è lenta, si dimezza l'intervallo ad ogni passo, ma si ha il vantaggio d'avere sempre una stima per difetto e una per eccesso della radice. Il metodo non può essere usato per problemi in dimensione maggiore di 1.

3.1.1 Falsa posizione

E' una variante del metodo di bisezione, l'unica differenza è che il punto x_{i+1} viene calcolato come intersezione della retta congiungente i punti $(x_i, f(x_i))$ e $(x_{i-1}, f(x_{i-1}))$ ovvero

$$x_{i+1} = \frac{x_i f(x_{i-1}) - x_{i-1} f(x_i)}{f(x_{i-1}) - f(x_i)}$$

e, per il resto, si procede analogamente. Un inconveniente di questa variante è che, in generale, non fornisce più l'intervallo di indeterminazione della ξ e, se la $f(x)$ è monotona, uno dei due estremi (a o b) non viene mai modificato (confronta figura 3.1).

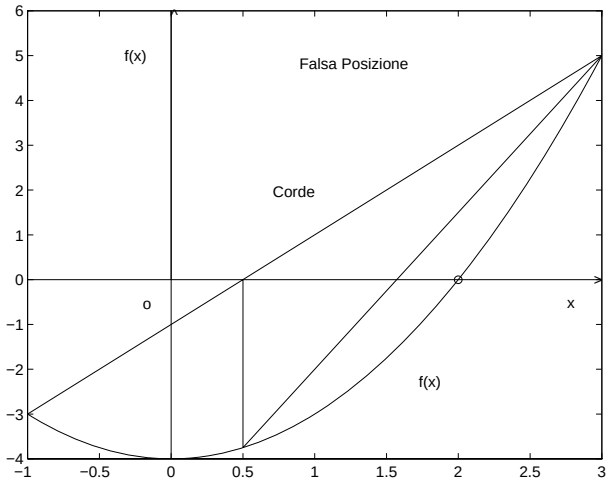


Figura 3.1:

Anche il test di convergenza deve essere modificato per contemplare sia il caso in cui la derivata prima di $f(x)$ sia "piccola" (funzioni piatte vicono a ξ) che quando è "grande" (funzioni ripide vicono a ξ). Un buon test di arresto è dato quindi dalla contemporanea verifica delle seguenti disuguaglianze

$$\begin{aligned} |f(x_{i+1})| &< zero \\ |x_i - x_{i-1}| &< toll \end{aligned}$$

dove *zero* e *toll* sono in generale diversi (confronta figure 3.2 e 3.3).

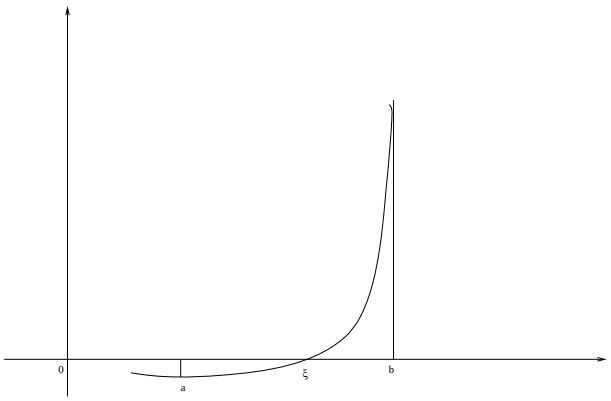


Figura 3.2:

3.2 Metodi di punto fisso (M.F.)

Sono i metodi che si ottengono trasformando il problema del calcolo dello zero di $f(x)$ nel problema equivalente della costruzione di una successione (convergente)

$$x_{k+1} = g(x_k) \tag{3.2}$$

dove $g(x)$ è scelta in modo che

$$\xi = g(\xi) \tag{3.3}$$

se

$$f(\xi) = 0.$$

Osservazione 3.2.1 *Si invita lo studente a soffermarsi sulle analogie con i metodi iterativi stazionari, presentati per la risoluzione dei sistemi lineari.*

Vediamo un semplice esempio

Esempio 3.2.1

$$\begin{aligned} f(x) &= x^2 - 2x + 1 \\ g_1(x) &= \frac{x^2+1}{2} \\ g_2(x) &= \sqrt{2x-1}; \quad x \geq \frac{1}{2} \\ g_3(x) &= x^2 - x + 1 \end{aligned}$$

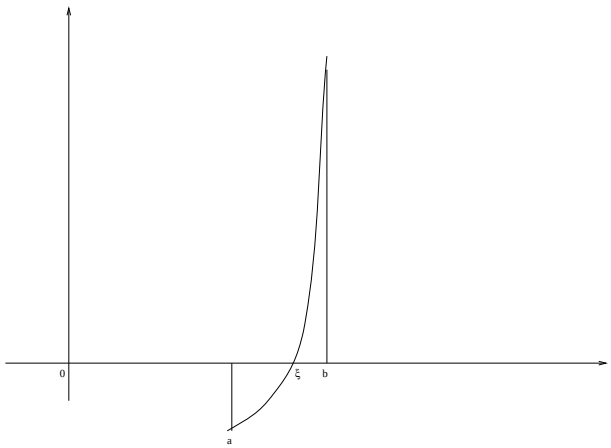


Figura 3.3:

Le tre $g_i(x)$ ammettono tutte $\xi = 1$ come "punto unito", ma non tutte e tre vanno bene per generare una successione convergente (partendo, per esempio, da $x_0 = \frac{1}{2}$).

La convergenza della (3.2) a ξ dipenderà quindi dalla $g(x)$.Valgono i seguenti teoremi:

Teorema 3.2.1 *Se la funzione $g(x) \in C^0[a,b]$ e se $g[a,b] \subseteq [a,b]$, allora esiste (almeno) un punto unito $\xi \in [a,b]$ per la trasformazione $g(x)$.*

Dimostrazione 3.2.1 *Essendo*

$$g[a,b] \subseteq [a,b]$$

si ha

$$\begin{aligned} a &\leq g(a) \leq b \\ a &\leq g(b) \leq b. \end{aligned}$$

Se $g(a) = a$ oppure $g(b) = b$ si ha la tesi con $\xi = a$ oppure $\xi = b$; altrimenti si ha

$$\begin{aligned} g(a) - a &> 0 \\ g(b) - b &< 0. \end{aligned}$$

Posto

$$F(x) := g(x) - x$$

allora $F(x) \in C^0[a,b]$ *ed inoltre* $F(a)F(b) < 0$, *per cui, applicando il teorema degli zeri* $F(\xi) = 0 = g(\xi) - \xi \rightarrow \xi = g(\xi)$.■

Teorema 3.2.2 *Se la funzione* $g(x) \in C^1[a,b]$ *e se* $g[a,b] \subseteq [a,b]$, *ed inoltre* $|g'(x)| \leq L < 1$ *allora esiste un solo punto unito* $\xi \in [a,b]$ *per la trasformazione* $g(x)$.

Dimostrazione 3.2.2 *(per assurdo) Supponiamo che esistano due distinti punti uniti* ξ_1 *e* ξ_2

$$\xi_1 = g(\xi_1); \quad \xi_2 = g(\xi_2); \quad \xi_1 \neq \xi_2$$

per il teorema della media $\exists \xi \in [\xi_1, \xi_2]$ *per cui*

$$\begin{aligned} |\xi_2 - \xi_1| &= |g(\xi_2) - g(\xi_1)| = |g(\xi)(\xi_2 - \xi_1| \\ &\leq L\,|\xi_2 - \xi_1| < |\xi_2 - \xi_1| \end{aligned}$$

che è assurdo.■

Teorema 3.2.3 *Se la funzione* $g(x) \in C^1[a,b]$ *e se* $g[a,b] \subseteq [a,b]$, *con* $|g'(x)| \leq L < 1$ *allora la sucessione* $x_{k+1} = g(x_k)$ *converge a* ξ *ed inoltre, definito* $e_n = x_n - \xi$, *si ha*

$$|e_n| \leq \frac{L^n}{1-L} |x_1 - x_0| .■$$

Osservazione 3.2.2 *Sebbene le condizioni indicate nel teorema (3.2.3) possano essere difficili da verificare a priori, il teorema fornisce una chiara risposta riguardo alle condizioni di convergenza e, se è nota* L, *il numero di iterazioni necessarie per ottenere* ξ *con una prefissata tolleranza.*

Osservazione 3.2.3 *Vale la pena di osservare che l'equazione (3.3) è l'equazione risolvente il sistema*

$$\begin{cases} y = x \\ y = g(x) \end{cases}$$

per cui il metodo (3.2) può essere interpretato geometricamente come indicato in figura 3.4.

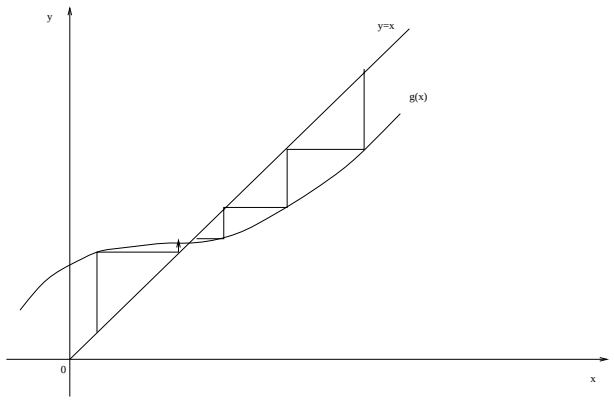


Figura 3.4:

Per quanto riguarda la *velocità di convergenza* e l'*ordine* di un metodo del tipo (3.2) diamo la seguente definizione

Definizione 3.2.1 *Un metodo del tipo (3.2) si dice di ordine k se*

$$\lim_{n\to\infty} \frac{e_{n+1}}{e_n^k} = c \text{ (costante).}$$

dove $e_n = x_n - \xi$, $g^{(i)}(\xi) = 0$ *per* $i < k$ *e* $g^{(k)}(\xi) \neq 0$.

I metodi per cui $g'(\xi) \neq 0$, vengono detti del *primo ordine* (a convergenza *lineare*), infatti

$$\begin{aligned} e_{n+1} &= x_{n+1} - \xi = g(x_n) - g(\xi) = \cdots Taylor \ \ldots \\ &= g'(\xi)e_n + \frac{g^{(2)}(\xi)}{2!}e_n^2 + \cdots + \frac{g^{(k)}(\alpha_n)}{k!}e_n^k \end{aligned} \tag{3.4}$$

dove $\alpha_n \in [x_n, \xi]$. Per $k = 1$ si ha

$$e_{n+1} = g'(\alpha_n)e_n$$

e per il teorema (3.2.3) essendo

$$\lim_{n\to\infty} x_n = \xi$$

sarà anche

$$\lim_{n\to\infty} \alpha_n = \xi$$

per cui

$$\lim_{n\to\infty} \frac{e_{n+1}}{e_n} = g'(\xi) \leq L < 1.\blacksquare$$

In generale se $g'(\xi) = g^{(2)}(\xi) = \cdots = g^{(k-1)}(\xi) = 0$ dalla (3.4) segue

$$\lim_{n\to\infty} \frac{e_{n+1}}{e_n^k} = \frac{g^{(k)}(\xi)}{k!} = c.$$

Osservazione 3.2.4 *Mentre per i metodi del primo ordine il guadagno ad ogni passo è costante (tanto più si guadagna quanto più L è minore di 1), nei metodi di ordine superiore più si è vicini alla soluzione maggiore è il guadagno al passo sucessivo (poco importa il valore di c, ciò che conta è che $k > 1$).*

3.3 Metodo di Newton (M.F.)

Si può ottenere un metodo del secondo ordine dalla seguente osservazione, sia

$$g(x) = x + h(x)f(x)$$

dove la nuova funzione $h(x)$ verrà scelta in modo che $g'(\xi) = 0$. E' immediato osservare che, se $f(\xi) = 0$ allora $\xi = g(\xi)$, costruiamoci la $h(x)$, derivando si ha

$$g'(x) = 1 + h'(x)f(x) + h(x)f'(x)$$

per $x = \xi$, essendo $f(\xi) = 0$, si ha

$$g'(\xi) = 1 + h'(\xi) \underbrace{f(\xi)}_{=0} + h(\xi)f'(\xi) = 1 + h(\xi)f'(\xi)$$

da cui

$$h(\xi) = -\frac{1}{f'(\xi)}.$$

Otteniamo quindi il metodo, almeno del secondo ordine (*metodo di Newton*), nella forma

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

Vale il seguente

Teorema 3.3.1 *Se la $f^{(2)}(x)$ è continua e la $f'(x) \neq 0$ in un aperto contenente ξ, allora esiste un $\epsilon > 0$ per cui $\forall\, |x_0 - \xi| < \epsilon$ il metodo di Newton converge quadraticamente a ξ.*■

Osservazione 3.3.1 *Il precedente teorema afferma che, pur di partire "abbastanza vicino" alla soluzione, la convergenza è quadratica. Nulla dice se si parte "lontano".*

Per garantire la convergenza "comunque si parta" bisogna richiedere che la $f(x)$ non cambi concavità in $[a, b]$, ovvero

Teorema 3.3.2 *Se $f(x) \in C^2[a, b]$, $f(a)f(b) < 0$, $f^{(2)}(x)$ è di segno costante in $[a, b]$ e $f'(x) \neq 0$, allora il metodo di Newton converge quadraticamente a ξ se si parte da un $x_0 \in [a, b]$ tale per cui $f(x_0)f^{(2)}(x_0) > 0$.*■

Una chiara interpretazione del precedente teorema è data nella figura 3.5.

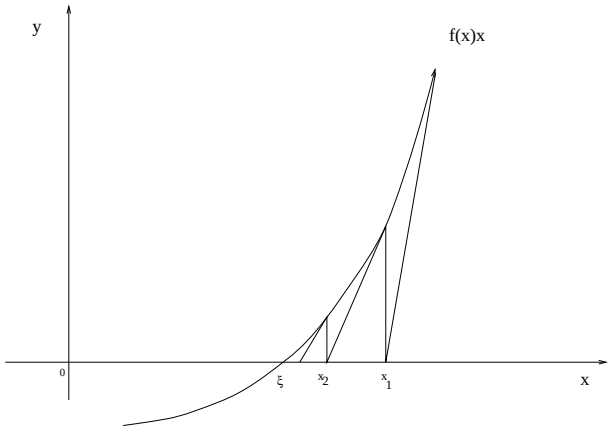


Figura 3.5:

Il metodo di Newton è, in generale, un metodo del secondo ordine ma ad ogni iterazione richiede la valutazione di due funzioni la $f(x)$ e la $f'(x)$. Se la $f'(x)$ non è nota, o se è molto oneroso il suo computo, il metodo non può essere usato o può risultare lento. Vedremo più avanti come si può ovviare a questi inconvenienti.

Se la radice ξ è multipla (per esempio doppia) il metodo non è più del secondo ordine ma decade al primo (con costante $L = \frac{1}{2}$), infatti

$$g'(x) = 1 - \frac{f'(x)^2 - f(x)f^{(2)}(x)}{f'(x)^2} = \frac{f(x)f^{(2)}(x)}{f'(x)^2}$$

per cui, utilizzando il teorema di de L'Hopital,

$$\begin{aligned}\lim_{x\to\xi} g'(x) &= \lim_{x\to\xi} \frac{f'(x)f^{(2)}(x) + f(x)f^{(3)}(x)}{2f'(x)f^{(2)}(x)} = \\ &= \lim_{x\to\xi} \frac{f^{(2)}(x)^2 + 2f'(x)f^{(3)}(x) + f(x)f^{(4)}(x)}{2(f^{(2)}(x)^2 + f'(x)f^{(3)}(x))} = \frac{1}{2}.\blacksquare\end{aligned}$$

E' facile dimostrare che se p è la molteplicita' della radice ξ, allora

$$x_{n+1} = x_n - p\frac{f(x_n)}{f'(x_n)}$$

è ancora del secondo ordine, mentre il metodo classico di Newton sarebbe del primo con costante $L = 1 - \frac{1}{p}$.

Presentiamo due varianti al metodo di Newton che permettono d'evitare il calcolo della $f'(x)$.

3.3.1 Metodo delle secanti

Si approssima la derivata prima con il rapporto incrementale per cui

$$f'(x_n) \simeq \frac{f(x_n) - f(x_{n-1})}{x_n - x_{n-1}}$$

ed il metodo diviene

$$x_{n+1} = x_n - \frac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})}.$$

Il metodo delle secanti richiede ad ogni passo il calcolo della funzione in un solo punto. La convergenza è iperlineare, ovvero

$$\lim_{n\to\infty} \frac{e_{n+1}}{e_n^p} = c; \qquad p = \frac{\sqrt{5}+1}{2} \simeq 1.61.$$

La successione generata dal metodo delle *secanti* NON coincide con quella generata dal metodo della *falsa posizione* (cfr. figura 3.6).

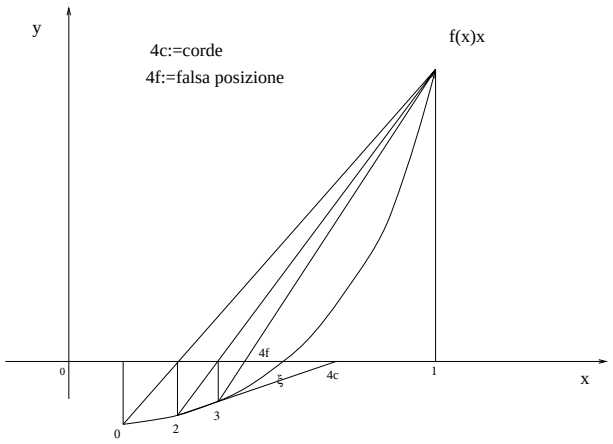


Figura 3.6:

3.3.2 Metodo di Steffensen

Si approssima la $f'(x)$ con

$$f'(x_n) \simeq \frac{f(x_n + f(x_n)) - f(x_n)}{f(x_n)}$$

per cui il metodo diviene

$$x_{n+1} = x_n - \frac{f(x_n)^2}{f(x_n + f(x_n)) - f(x_n)}.$$

Il metodo di Steffensen richiede il calcolo della funzione in due punti ad ogni passo (come Newton) ed è un metodo del secondo ordine.

3.4 Sistemi non lineari (M.F.)

3.4.1 Metodi di punto fisso

Dato il sistema non lineare

$$\underline{f}(\underline{x}) = \underline{0} \tag{3.5}$$

lo si trasforma nel problema equivalente

$$\underline{x}_{k+1} = \underline{\phi}(\underline{x}_k) \tag{3.6}$$

per cui se, $\underline{\xi}|\underline{f}(\underline{\xi}) = \underline{0}$ allora $\underline{\xi} = \underline{\phi}(\underline{\xi})$.

Indicata con

$$J(\underline{\phi}(\underline{x}_k)) = \left[\begin{array}{ccccc} \frac{\partial\phi_1}{\partial x_1} & \frac{\partial\phi_1}{\partial x_2} & \cdots & \frac{\partial\phi_1}{\partial x_{n-1}} & \frac{\partial\phi_1}{\partial x_n} \\ \frac{\partial\phi_2}{\partial x_1} & \frac{\partial\phi_2}{\partial x_2} & \cdots & \frac{\partial\phi_2}{\partial x_{n-1}} & \frac{\partial\phi_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\partial\phi_{n-1}}{\partial x_1} & \frac{\partial\phi_{n-1}}{\partial x_2} & \cdots & \frac{\partial\phi_{n-1}}{\partial x_{n-1}} & \frac{\partial\phi_{n-1}}{\partial x_n} \\ \frac{\partial\phi_n}{\partial x_1} & \frac{\partial\phi_n}{\partial x_2} & \cdots & \frac{\partial\phi_n}{\partial x_{n-1}} & \frac{\partial\phi_n}{\partial x_n} \end{array}\right]_{\underline{x}\equiv\underline{x}_k} \tag{3.7}$$

la matrice jacobiana di $\underline{\phi}(\underline{x})$ valutata nel punto $\underline{x}_k$, vale il seguente

Teorema 3.4.1 *Condizione sufficiente per la convergenza del metodo (3.6) è che*

$$\left\| J(\underline{\phi}(\underline{x}_k)) \right\| \leq L < 1, \quad \forall k. \blacksquare$$

Osservazione 3.4.1 *Si confronti il teorema (3.4.1) con il teorema (2.4.3).*

3.4.2 Metodo di Newton per sistemi

E' la generalizzazione ai sistemi non lineari del metodo di Newton visto al paragrafo (3.3). Per calcolare la soluzione del sistema non lineare (3.5) si costruisce la successione

$$\underline{x}_{k+1} = \underline{x}_k + \underline{h}_k$$

dove il vettore $\underline{h}_k$ è ottenuto come soluzione del sistema lineare

$$J(\underline{f}(\underline{x}_k))\underline{h}_k = -\underline{f}(\underline{x}_k) \tag{3.8}$$

dove con $J(\underline{f}(\underline{x}_k))$ si è indicata (cfr. (3.7)) la matrice jacobiana di $\underline{f}(\underline{x})$ valutata nel punto $\underline{x}_k$. Il metodo può essere dedotto facilmente sviluppando la $\underline{f}(\underline{x}_k)$ in un intorno di $\underline{x}_k$. Dopo aver posto $\underline{\xi} = \underline{x}_k + \underline{h}_k$, si ha

$$\underline{f}(\underline{\xi}) = \underline{0} = \underline{f}(\underline{x}_k + \underline{h}_k) = ...sviluppando...$$

$$\left\{\begin{array}{lcl} f_1(\underline{x}_k) + h_{k1}\left(\frac{\partial f_1}{\partial x_1}\right)_{\underline{x}\equiv\underline{x}_k} + \cdots + h_{kn}\left(\frac{\partial f_1}{\partial x_n}\right)_{\underline{x}\equiv\underline{x}_k} + \cdots & = & 0 \\ \cdots & = & \cdots \\ f_n(\underline{x}_k) + h_{k1}\left(\frac{\partial f_n}{\partial x_1}\right)_{\underline{x}\equiv\underline{x}_k} + \cdots + h_{kn}\left(\frac{\partial f_n}{\partial x_n}\right)_{\underline{x}\equiv\underline{x}_k} + \cdots & = & 0 \end{array}\right. \tag{3.9}$$

dove con h_{ki} si è indicata la i-esima componente di $\underline{h}_k$. La (3.9), trascurando i termini di ordine superiore al primo, può essere scritta nella forma matriciale (3.8).

Vale la pena d'osservare che nella determinazione del vettore $\underline{x}_{k+1}$ la parte più onerosa computazionalmente è la risoluzione del sistema lineare (3.8), per cui può essere utile tenere fissa la matrice $J(\underline{f}(\underline{x}_k))$ per un numero fissato (diciamo r) di iterazioni, in modo da poter utilizzare la decomposizione LU già effettuata al passo k, e riaggiornare la matrice solo al passo $(k+r)$.

Osservazione 3.4.2 *Anche nel caso monodimensionale esiste una variante analoga che consiste nel non calcolare ad ogni passo la $f'(x_k)$ mandando, ad ogni passo k, la retta parallela alla precedente (non più la tangente) passante per il punto $(x_k, f(x_k))$ (cfr. figura 3.7).*

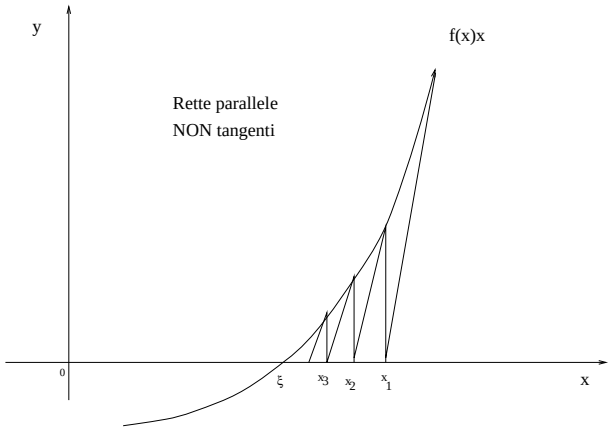


Figura 3.7:

Molto costoso è anche il preventivo calcolo simbolico della matrice jacobiana (n^2 derivate parziali da calcolare !) per cui, quando queste derivate sono molto complicate o non sono computabili (non si hanno le $\underline{f}(\underline{x})$ in forma analitica), si può operare con una variante, simile al metodo delle corde. Precisamente si approssimano gli elementi della matrice jacobiana con dei rapporti incrementali nella forma:

$$\left(\frac{\partial f_j}{\partial x_i}\right)_{\underline{x}=\underline{x}_k} \simeq B_{ji} = \frac{f_j(\underline{x}_k + \underline{e}_k h_{ki}) - f_j(\underline{x}_k)}{h_{ki}}; \quad \underline{e}_{kj} = \delta_{kj}$$

dove, per esempio, $h_{ki} = \underline{x}_{ki} - \underline{x}_{k-1i}$.

3.5 Zeri di polinomi (M.F.)

Una trattazione a parte può essere fatta per le equazioni algebriche in quanto, oltre ai metodi visti in precedenza, si possono considerare altri metodi numerici sfruttando il legame che esiste fra un polinomio e la sua *matrice di Frobenious* (o *Companion matrix*), vale infatti il seguente

Teorema 3.5.1 *Dato il polinomio monico*

$$p_n(x) = x^n + a_1 x^{n-1} + \cdots + a_{n-1}x + a_n \tag{3.10}$$

questi risulta essere il polinomio caratteristico della matrice

$$C_n = \begin{bmatrix} 0 & 0 & \cdots & 0 & -a_n \\ 1 & \ddots & \ddots & \vdots & -a_{n-1} \\ 0 & 1 & \ddots & 0 & \vdots \\ \vdots & \ddots & \ddots & 0 & -a_2 \\ 0 & \cdots & 0 & 1 & -a_1 \end{bmatrix} .\blacksquare \tag{3.11}$$

In virtù del risultato precedente è quindi possibile calcolare gli zeri di (3.10) mediante il calcolo degli autovalori di (3.11) (cfr. capitolo sugli autovalori).

3.5.1 Schema di Horner

Il polinomio (3.10) può essere scritto nella forma (dovuta ad *Horner*)

$$p_n(x) = a_n + x(a_{n-1} + x(a_{n-2} + x(\ldots + x(a_1 + x)\ldots))) \tag{3.12}$$

(si lascia al lettore la facile verifica).

La forma (3.12) permette di valutare il polinomio $p_n(x)$ nel punto α mediante il seguente schema

$$\begin{cases} \gamma_0 = 1 \\ \gamma_k = \gamma_{k-1}\alpha + a_k \end{cases} ; \qquad k = 1, 2, \ldots, n \tag{3.13}$$

per cui

$$p_n(\alpha) = \gamma_n.$$

E' facile verificare che i valori γ_k $(k = 1, \ldots, n-1)$ sono i coefficienti del polinomio quoziente $p_{n-1}(x)$ della divisione per $(x - \alpha)$ di $p_n(x)$, mentre γ_n è il resto r_0. Lo schema (3.13) rappresenta quindi un modo compatto per effettuare la *divisione sintetica*

$$p_n(x) = (x - \alpha)p_{n-1}(x) + r_0. \tag{3.14}$$

Dimostrazione 3.5.1 *Scriviamo il polinomio*

$$p_{n-1}(x) = x^{n-1} + \gamma_1 x^{n-2} + \cdots + \gamma_{n-2}x + \gamma_{n-1}$$

dalla (3.14), eguagliando i coefficienti dei termini di ugual grado, tenendo conto della (3.10) si ha

1=1	*1=1*
$\gamma_1 - \alpha = a_1$	$\gamma_1 = \alpha + a_1$
$\gamma_2 - \alpha\gamma_1 = a_2$	$\gamma_2 = \alpha\gamma_1 + a_2$
$\cdots$	$\cdots$
$\gamma_{n-1} - \alpha\gamma_{n-2} = a_{n-1}$	$\gamma_{n-1} = \alpha\gamma_{n-2} + a_{n-1}$
$r_0 - \alpha\gamma_{n-1} = a_n$	$r_0 = \alpha\gamma_{n-1} + a_n = \gamma_n$

.■

Analogamente si può calcolare la derivata prima di $p_n(x)$ in $x = \alpha$ operando la stessa divisione con $p_{n-1}(x)$, infatti, posto

$$p_{n-1}(x) = (x - \alpha)p_{n-2}(x) + r_1$$

si ha

$$p_n(x) = (x - \alpha)^2 p_{n-2}(x) + (x - \alpha)r_1 + r_0$$

e derivando

$$p_n'(x) = 2(x - \alpha)p_{n-2}(x) + (x - \alpha)^2 p_{n-2}'(x) + r_1$$

da cui,

$$p_n'(\alpha) = r_1.\blacksquare$$

Si lascia al lettore, per esercizio, la verifica che

$$p_n^{(m)}(\alpha) = m!r_m, \qquad m = 0, 1, \ldots, n.$$

Lo schema di Horner, o divisione sintetica, fornisce un algoritmo efficiente per valutare il polinomio e la sua derivata prima in un punto permettendo, con un costo di $2n$ flops, di eseguire un passo del metodo di Newton

$$x_{n+1} = x_n - \frac{p_n(x_n)}{p_n'(x_n)}.$$

3.5.2 Radici multiple

Abbiamo già visto che, nel caso di radici multiple, il metodo di Newton non converge più quadraticamente. Per quanto riguarda i polinomi è sempre possibile operare con polinomi che hanno radici semplici, infatti il polinomio

$$p_{n-r}(x) = \frac{p_n(x)}{q_r(x)}$$

ha tutte e sole le radici di $p_n(x)$ ma semplici, essendo

$$q_r(x) = MCD\left\{p_n(x), p_n'(x)\right\}$$

dove con $MCD\{\}$ si è indicato il massimo comun divisore fra i polinomi in parentesi $\{\}$.

Ovviamente se le radici di $p_n(x)$ sono semplici allora $q_r(x) = q_0(x) = k$, costante. Ricordiamo che il MCD può essere calcolato agevolmente con il noto algoritmo euclideo.

3.6 Riepilogo (M.F.)

Possiamo sintetizzare i risultati presentati nel modo seguente

1. $f(x) \in C^0(a,b)$ (funzione poco regolare)
   (a) metodo di *bisezione,*
   (b) metodo della *falsa posizione.*
2. $f(x) \in C^k(a,b)$ ($k > 0$, funzione regolare)
   (a) metodi di *ordine* k (punto fisso).
3. Complessità computazionale
   (a) 1 valutazione per passo: bisezione, falsa posizione e corde,
   (b) 2 valutazioni per passo: Newton e Steffensen.
4. Velocità di convergenza

(a) metodi del primo ordine (*lineare*): dipende da L ($|g'(x)| \leq L < 1$),

(b) 1.61 corde,

(c) 2 Newton e Steffensen.

5. Equazioni algebriche:

   (a) tutti i metodi precedenti,

   (b) metodi dell'algebra lineare (calcolo degli autovalori della companion matrix).

I metodi di Newton, corde ed i metodi di punto fisso possono essere estesi al caso di sistemi non lineari.

3.6.1 Esercizi

Esercizio 3.6.1 *Quanti passi sono necessari per stimare, con errore minore di* 10^{-6}, *la radice di*

$$e^x - 2 = 0$$

utilizzando il metodo di bisezione partendo dall'intervallo $[0, 1]$.

Esercizio 3.6.2 *Utilizzare, per calcolare la radice dell'equazione*

$$3e^{2x} - 12e^x + 4 = 0$$

1. *il metodo di Newton*

2. *il metodo*

$$x_{n+1} = \frac{x_n e^{x_n} - e^{x_n} + 2}{e^{x_n}},$$

partendo sempre da $x_0 = 2$. *Quale metodo converge prima? Perchè?*

Esercizio 3.6.3 *Utilizzare, se possibile, i seguenti metodi*

1. $x_{n+1} = -\frac{\ln x}{2}$
2. $x_{n+1} = e^{-2x}$
3. $x_{n+1} = \frac{x_n + e^{-2x_n}}{2}$

per calcolare la radice di

$$2x + \ln x = 0.$$

Quale metodo converge più velocemente? Perchè? Si fornisca un metodo migliore di quelli proposti.

Esercizio 3.6.4 *Dato il sistema non lineare*

$$\begin{cases} x + \frac{y^2}{4} &= \frac{5}{4} \\ y + \frac{x^2}{4} &= \frac{5}{4} \end{cases}$$

studiare la convergenza del seguente metodo iterativo

$$\begin{cases} x_{n+1} &= -\frac{y_n^2}{4} + \frac{5}{4} \\ y_{n+1} &= -\frac{x_n^2}{4} + \frac{5}{4} \end{cases} .$$

Esercizio 3.6.5 *Calcolare le radici del seguente polinomio*

$$x^3 - 4x^2 + x - 18 = 0$$

1. *utilizzando il metodo di Newton,*
2. *il calcolo degli autovalori della companion matrix associata.*

Capitolo 4

Teoria dell'approssimazione (M.Frontini)

Il problema che vogliamo affrontare ora è quello della ricostruzione di una funzione, o semplicemente la determinazione del suo valore in un punto, noti i valori che la funzione stessa assume in un numero finito di punti assegnati. E' evidente che così formulato il problema *non è ben posto*, nel senso che può avere un numero infinito di soluzioni o nessuna soluzione. Si pensi al caso in cui la funzione deve essere continua e si hanno due punti con uguale ascissa e diversa ordinata. Per ripristinare l'esistenza ed unicità bisognerà aggiungere delle condizioni

1. sulla regolarità della funzione
2. sul tipo di approssimazione da usare
3. sulle funzioni approssimanti.

In generale, data una funzione $f(x)$ nota in n punti $y_i = f(x_i)$ $(i = 1, ..., n)$, si cerca un'approssimante della forma

$$g(x) = \sum_{i=1}^{n} \alpha_i \phi_i(x)$$

dove i pesi α_i sono scelti secondo un dato *criterio di merito,* mentre le $\phi_i(x)$ sono funzioni opportune che verificano date condizioni di regolarità e sono facilmente manipolabili (ovvero facilmente computabili, derivabili ed integrabili).

Come esempi di funzioni $\phi_i(x)$ si possono considerare

1. i polinomi di grado n-1;
2. i polinomi a tratti di grado m;
3. le funzioni spline di grado m;
4. i polinomi trigonometrici di Fourier;
5. le funzioni razionali.

Come criteri di merito per determinare gli α_i si possono considerare

1. l'interpolazione $(g(x_i) = f(x_i) = y_i,\ i = 1, ..., n)$;
2. i minimi quadrati (norma L^2) $(\min_{a_i} \left\{\sum_{i=1}^{n} \left(g(x_i) - y_i\right)^2\right\})$;
3. minimo massimo errore (minimax, norma L^∞) $(\min_{a_i} \left\{\max_i |g(x_i) - y_i|\right\})$;
4. minima somma dei moduli (norma L^1) $(\min_{a_i} \{\sum_{i=1}^{n} |g(x_i) - y_i|\})$,

che danno una misura della distanza fra la funzione incognita $f(x)$ e l'approssimante $g(x)$. A seconda dei campi d'applicazione è meglio utilizzare un criterio piuttosto che un altro

L'*interpolazione polinomiale* è utile quando si hanno pochi dati (n piccolo) per cui si ottiene un polinomio di grado basso. La regolarità del polinomio interpolante è utile poi nella deduzione di formule approssimate per il *calcolo integrale* e per l'intergrazione di *equazione differenziali* (cfr. capitoli 5 e 6).

L'interpolazione mediante *spline* è utile nelle applicazioni di grafica (CAD) (non a caso il termine "spline" indica in inglese il "curvilineo mobile" del disegno tecnico).

I *minimi quadrati* sono utilizzati nei problemi di *identificazione* di parametri e modelli, quando si hanno a disposizione più dati ma affetti da errori (di misura casuali ecc.).

Il criterio del *minimo massimo errore* è essenziale quando si deve garantire una certa accuratezza (routine di calcolo delle funzioni elementari).

Contrariamente a quanto avviene in R^n, nel caso dell'approssimazione di funzioni le norme non sono più equivalenti come mostrato dal seguente esempio

Esempio 4.0.1 *Data la funzione $f(x)$ identicamente nulla in $[0,3]$ consideriamo la successione di funzioni $g_k(x)$ (cfr. figura 4.1) così definite*

$$g_k(x) = \begin{cases} k(k^2x-1) & \frac{1}{k^2} \le x \le \frac{2}{k^2} \\ -k(k^2x-3) & \frac{2}{k^2} \le x \le \frac{3}{k^2} \\ 0 & altrove \end{cases} .$$

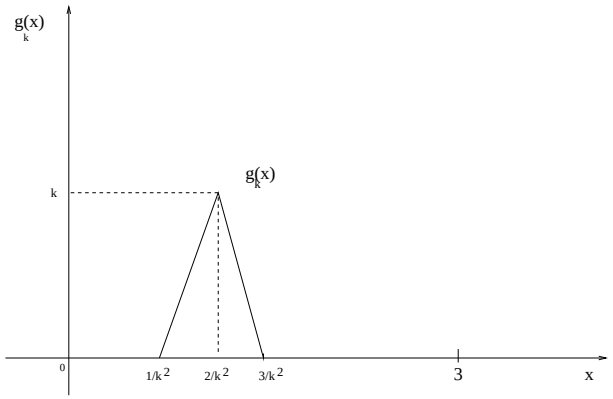


Figura 4.1:

Considerando le tre classiche definizioni di norma di funzioni, si ha:

$$\|f - g_k\|_1 = \tfrac{1}{k}$$
$$\|f - g_k\|_2 = \sqrt{\tfrac{2}{3}}$$
$$\|f - g_k\|_\infty = k$$

per cui la succesione $g_k(x)$ converge a $f(x)$ in norma L^1 ma non converge in norma L^2 e L^∞.□

Valgono i seguenti teoremi

Teorema 4.0.1 *La convergenza in norma L^∞ garantisce la convergenza delle altre due norme.*

Dimostrazione 4.0.1 *Dalla definizione di norma e dal teorema della media si ha*

$$\|f - g\|_1 = \int_a^b |f(x) - g(x)|\, dx = ...teorema\ media...$$

$$\begin{aligned} &= |f(\xi) - g(\xi)| \int_a^b dx \leq \\ &\leq \max_x |f(x) - g(x)| \cdot (b-a) \leq \\ &\leq C \left\| f(x) - g(x) \right\|_\infty , \end{aligned}$$

essendo $\left\| f(x) - g(x) \right\|_\infty = \max_x |f(x) - g(x)|$ *e* $C = b - a$. *Analogamente per la norma* L^2.■

Definito con Π_N lo spazio dei polinomi di grado N valgono i seguenti teoremi

Teorema 4.0.2 *(di Weierstrass) Data una funzione* $f(x) \in C^0[a,b]$ *allora,* $\forall \varepsilon > 0$ *esiste un* N_ε *tale per cui è possibile trovare un polinomio* $p(x) \in \Pi_{N_\varepsilon}$ *per cui*

$$\left\| f - p \right\|_\infty < \varepsilon .\blacksquare$$

Teorema 4.0.3 *(di miglior approssimazione) Data una funzione* $f(x) \in C^0[a,b]$ *e un numero positivo* N, *esiste un unico polinomio* $p^*(x) \in \Pi_N$ *per cui* $\left\| f - p^* \right\| \leq \left\| f - p \right\| , \quad \forall p \in \Pi_N .\blacksquare$

Osservazione 4.0.1 *Questo teorema vale per ogni norma, anche se* $p^*(x)$ *varia al variare della norma.*

Osservazione 4.0.2 *Il teorema di Weierstrass afferma l'esistenza del polinomio* $p(x)$ *di grado* N_ε *ma non dice come il grado* N_ε *dipenda da* ε.

4.1 Interpolazione (M.F.)

Il problema che vogliamo affrontare ora è la costruzione di una funzione $g(x)$ interpolante la $f(x)$ in $n+1$ nodi x_i. La $g(x)$ sarà quindi caratterizzata dal fatto che

$$g(x_i) = f(x_i) \equiv y_i, \quad i = 0, 1, 2, ..., n.$$

Fra tutte le possibili $g(x)$ interpolanti considereremo solo il caso dei polinomi, delle funzioni spline e dei polinomi trigonometrici.

4.1.1 Polinomi di Lagrange

Per costruire $g(x) \in \Pi_n$ definiamo i seguenti $n+1$ *polinomi di Lagrange* (di grado n) $l_j(x)$ definiti sui nodi d'interpolazione e caratterizzati dal nodo x_j

$$l_j(x) = \frac{(x-x_0)(x-x_1)\cdots(x-x_{j-1})(x-x_{j+1})\cdots(x-x_{n-1})(x-x_n)}{(x_j-x_0)(x_j-x_1)\cdots(x_j-x_{j-1})(x_j-x_{j+1})\cdots(x_j-x_{n-1})(x_j-x_n)}.$$

Definito il *polinomio monico* (con coefficiente della potenza di grado massimo uguale ad 1) di grado $n+1$

$$\pi_{n+1}(x) = \prod_{i=0}^{n}(x-x_i)$$

si ha

$$l_j(x) = \frac{\pi_{n+1}(x)}{(x-x_j)\pi'_{n+1}(x_j)}.$$

Osservazione 4.1.1 *Ogni polinomio $l_j(x)$ gode delle seguenti proprietà*

1. *è un polinimio di grado n,*
2. *$l_j(x_i) = \delta_{ij}$ (simbolo di Kroneker) (vale 1 nel nodo $x = x_j$ e vale 0 nei nodi $x \neq x_j$).*

Si ottiene quindi (*esistenza*) il polinomio interpolante

$$g(x) \equiv p_n(x) = \sum_{i=0}^{n} y_i l_i(x)$$

per il quale si verifica banalmente che

$$y_i = p_n(x_i).$$

Se i nodi x_i sono distinti vale il seguente

Teorema 4.1.1 *(unicità) Se i nodi x_i, $i = 0, 1, ..., n$, sono distinti il polinomio $p_n(x) \in \Pi_n$ interpolante è unico.*

Dimostrazione 4.1.1 *(per assurdo) supponiamo che esista* $q_n(x) \in \Pi_n$ *tale che*

$$q_n(x) \neq p_n(x); \quad q_n(x_i) = p_n(x_i) = y_i, \quad i = 0, 1, ..., n$$

definiamo

$$r_n(x) = p_n(x) - q_n(x) \in \Pi_n$$

si avrà

$$r_n(x_i) = p_n(x_i) - q_n(x_i) = y_i - y_i = 0$$

per cui il polinomio $r_n(x)$ *(di grado n) avendo* $n+1$ *zeri è identicamente nullo, da cui*

$$q_n(x) = p_n(x). \blacksquare$$

4.1.2 Sistema di Vandermonde

Il polinomio interpolante può essere scritto nella forma

$$p_n(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1} + a_n x^n$$

gli $n+1$ parametri a_i possono essere determinati mediante la risoluzione del sistema lineare (4.1), che si ottiene imponendo il passaggio del polinomio per gli $n+1$ punti d'interpolazione (x_i, y_i)

$$\begin{cases} a_{0+} a_1 x_0 + a_2 x_0^2 + & \cdots & +a_n x_0^n & = y_0 \\ a_{0+} a_1 x_1 + a_2 x_1^2 + & \cdots & +a_n x_1^n & = y_1 \\ \cdots & \cdots & \cdots & = \cdots \\ a_{0+} a_1 x_n + a_2 x_n^2 + & \cdots & +a_n x_n^n & = y_n \end{cases} \tag{4.1}$$

che in foma matriciale diviene

$$V\underline{a} = \underline{y}$$

dove

$$V = \begin{bmatrix} 1 & x_0 & \cdots & x_0^n \\ 1 & x_1 & \cdots & x_1^n \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & \cdots & x_n^n \end{bmatrix}$$

è la matrice di *Vandermonde* (che è noto essere non singolare quando $x_i \neq x_j$ per $i \neq j$).

Esempio 4.1.1 *Costruiamo i polinomi di Lagrange ed il polinomio interpolante i seguenti punti*

x_i	0	1	2
y_i	-1	2	7

si ha

$$l_1(x) = \frac{(x-1)(x-2)}{(0-1)(0-2)}; \quad l_2(x) = \frac{(x-0)(x-2)}{(1-0)(1-2)}; \quad l_3(x) = \frac{(x-0)(x-1)}{(2-0)(2-1)};$$

e quindi

$$p_2(x) = -1\frac{x^2-3x+2}{2} + 2\frac{x^2-2x}{-1} + 7\frac{x^2-x}{2} = x^2 + 2x - 1.$$

Si lascia al lettore la costruzione dello stesso polinomio interpolante con la tecnica di Vandermonde.□

4.1.3 Stima dell'errore

E' possibile valutare, sotto opportune ipotesi di regolarità della funzione $f(x)$, l'errore che si commette sostituendo per ogni x il polinomio interpolante alla funzione $f(x)$ stessa, vale infatti il seguente

Teorema 4.1.2 *Se la funzione* $f(x) \in C^{n+1}[a,b]$, *definito* $\pi_{n+1}(x) = \prod_{i=0}^{n}(x - x_i)$, *con* $x_i \in [a,b]$, *allora* $\forall x \in [a,b]$

$$e_n(x) = f(x) - p_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!}\pi_{n+1}(x). \tag{4.2}$$

Dimostrazione 4.1.2 *Definiamo la funzione* $u(t) \in C^{n+1}[a,b]$ *come*

$$u(t) = f(t) - p_n(t) - \frac{f(x) - p_n(x)}{\pi_{n+1}(x)}\pi_{n+1}(t)$$

la $u(t)$ *ha* $n+2$ *zeri* ($t = x$, $t = x_i$ $i = 0, 1, ..., n$), *applicando il teorema di Rolle generalizzato alla derivata (n+1)-esima di* $u(t)$ *si ha*

$$u^{(n+1)}(\xi) = f^{(n+1)}(\xi) - \frac{f(x) - p_n(x)}{\pi_{n+1}(x)}(n+1)! = 0 \tag{4.3}$$

($p_n(t)$ *è un polinomio di grado* n *per cui* $p_n^{(n+1)}(t) \equiv 0$, $\pi_{n+1}(t)$ *è un polinomio di grado* $n+1$ *monico per cui* $\pi_{n+1}^{(n+1)}(t) = (n+1)!$) *mentre* $\xi = \xi(x)$ *è un punto dell'intervallo determinato da* x *ed i nodi* x_i, *dalla (4.3) segue la tesi.*■

Osservazione 4.1.2 *La (4.2) è una valutazione "locale" dell'errore (dipende dal punto x). Se per $x = x_i$ si ottiene (come ovviamente deve essere) errore nullo, possono esistere valori di x per cui la (4.2) può fornire valori troppo grandi (si pensi ad x lontani dall'intervallo definito dai nodi x_i). Questo non deve stupire in quanto è logico che lontano dai punti dati (dove si hanno le informazioni su f) sia più difficile ricostruire la funzione.*

Se $\left|f^{(n+1)}(\xi)\right| < K_{n+1}$ costante, $\forall \xi \in [a,b]$ allora dalla (4.2) si ottiene

$$\max_x |e_n(x)| = \max_x |f(x) - p_n(x)| \leq \frac{K_{n+1}}{(n+1)!} \max_x |\pi_{n+1}(x)|,$$

e, se i nodi sono equidistanti in $[a,b]$, $\max_x |\pi_{n+1}(x)|$ è ottenuto per x prossimo al bordo di $[a,b]$ (in generale l'approssimazione è più accurata al centro dell'intervallo che ai bordi, perchè?).

Osservazione 4.1.3 *L'uso dei polinomi di Lagrange ha l'inconveniente che l'aggiunta di un nodo d'interpolazione comporta il ricalcolo di tutti i polinomi $l_i(x)$. Inoltre anche il calcolo della derivata del polinomio interpolante, scritto nella forma*

$$p_n(x) = \sum_{i=0}^{n} y_i l_i(x)$$

risulta poco agevole.

4.1.4 Differenze divise

Un generico polinomio di grado n, che assume prefissati valori y_i in $n+1$ nodi x_i $(i = 0, 1, ..., n)$, può essere scritto nella forma

$$\begin{aligned} p_n(x) &= a_0 + a_1(x - x_0) + a_2(x - x_0)(x - x_1) + \cdots \\ &\quad \cdots + a_n(x - x_0)(x - x_1)\cdots(x - x_{n-1}) \end{aligned} \tag{4.4}$$

dove i coefficienti a_i vengono determinati imponendo i prefissati valori y_i.

E' immediato osservare che

$$p_n(x_0) = a_0; \quad a_0 = y_0,$$

ed inoltre

$$p_n(x_1) = a_0 + a_1(x_1 - x_0); \quad a_1 = \frac{y_1 - y_0}{x_1 - x_0},$$

mentre, dopo alcuni calcoli,

$$p_n(x_2) = a_0 + a_1(x_2 - x_0) + a_2(x_2 - x_0)(x_2 - x_1); \quad a_2 = \frac{\frac{y_2-y_1}{x_2-x_1} - \frac{y_1-y_0}{x_1-x_0}}{x_2 - x_0}.$$

Senza entrare nei dettagli si può dimostrare che

$$a_k = f[x_0, x_1, ..., x_k]$$

dove con $f[x_0, x_1, ..., x_k]$ si è indicata la *differenza divisa k-esima* sui nodi $x_0, x_1, ..., x_k$ definita dalle seguenti relazioni

$$\begin{aligned} f[x_i] &: = f(x_i) \\ f[x_i, x_{i+1}] &: = \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i} \\ &.... \\ f[x_i, x_{i+1}, ..., x_{i+k}] &: = \frac{f[x_{i+1}, ..., x_{i+k}] - f[x_i, ..., x_{i+k-1}]}{x_{i+k} - x_i}. \end{aligned} \tag{4.5}$$

Le (4.5) permettono di costruire, ricorsivamente, tutti i coefficienti a_k. Il vantaggio di scrivere un polinomio nella forma delle (4.4) risiede nel fatto che l'aggiunta di un nuovo nodo x_{n+1} non richiede il ricalcolo dei primi n coefficienti. Si ha infatti

$$p_{n+1}(x) = p_n(x) + a_{n+1}(x - x_0)(x - x_1) \cdots (x - x_{n-1})(x - x_n)$$

e basterà calcolare a_{n+1} mediante le (4.5).

4.1.5 Interpolazione di Hermite

Se oltre ai valori della funzione in dati punti si conoscono anche i valori della/e derivate nei punti si può cercare (se esiste) il polinomio che interpola sia la funzione che le sue derivate. Un esempio in questo senso è dato dal polinomio di Taylor che, come noto, ha gli stessi valori della funzione e delle sue prime n derivate in un dato punto x_0

$$p_n(x) = \sum_{j=0}^{n} \frac{f^{(j)}(x_0)}{j!}(x - x_0)^j$$

che esiste ed è unico essendo ricavabile dalle condizioni

$$p_n^{(j)}(x) = f^{(j)}(x_0), \quad j = 0, 1, ..., n$$

che portano al sistema lineare triangolare superiore, non singolare

$$\begin{cases} a_0 & +a_1x_0 & +a_2x_0^2 & +\cdots+\cdots & +a_nx_0^n & = f(x_0) \\ & +a_1 & +2a_2x_0 & +\cdots+\cdots & +na_nx_0^{n-1} & = f'(x_0) \\ & & \cdots & +\cdots+\cdots & \cdots & = \cdots \\ & & & (n-1)!a_{n-1} & +n!a_nx_0 & = f^{(n-1)}(x_0) \\ & & & & +n!a_n & = f^{(n)}(x_0) \end{cases}.$$

Osservazione 4.1.4 *Val la pena di osservare che nodi, valori della funzione e derivate non possono essere assegnati arbitrariamente, pena la non esistenza (o la non unicità) del polinomio interpolante, come dimostra il seguente*

Esempio 4.1.2 *Costruire (se esiste!?) la parabola passante per i punti (0,0) e (2,2) con derivata -1 nel punto di ascissa 1. Dalle condizioni date si ha*

$$p_2(x) = a_0 + a_1x + a_2x^2$$

e quindi il sistema lineare

$$\begin{cases} a_0 & & = 0 \\ 2a_1 & +4a_2 & = 2 \\ a_1 & +2a_2 & = -1 \end{cases}$$

che è impossibile.□ (Cosa succede se la derivata in 1 vale 1?).

Se conosciamo la funzione (y_i) e la derivata prima (y_i') negli $n+1$ nodi x_i è possibile costruire il polinomio interpolante di Hermite di grado $2n+1$ dato da

$$p_{2n+1}(x) = \sum_{j=0}^{n} \left[y_j A_j(x) + y_j' B_j(x) \right]$$

dove $A_j(x)$ e $B_j(x)$ sono polinomi di grado $2n+1$ definiti da

$$\begin{aligned} A_j(x) &= \left[1 - 2(x-x_j)l_j'(x_j)\right] l_j^2(x) \\ B_j(x) &= (x-x_j)l_j^2(x) \end{aligned}$$

e $l_j(x)$ sono i polinomi di Lagrange sui nodi x_i.

Si può dimostrare (si lasciano i conti al lettore) che

$$\begin{array}{cc} A_j(x_i) = \delta_{ji}; & B_j(x_i) = 0 \\ A_j'(x_i) = 0; & B_j'(x_i) = \delta_{ji} \end{array} \qquad \forall i,j$$

per cui

$$p_{2n+1}(x_i) = y_i; \quad p'_{2n+1}(x_i) = y'_i; \quad i = 0, 1, ..., n.$$

Con tecnica analogo a quella usata per l'interpolazione di Lagrange si può dimostrare che per l'errore si ha

$$e_{2n+1}(x) = f(x) - p_{2n+1}(x) = \frac{f^{(2n+2)}(\xi)}{(2n+2)!}\pi^2_{n+1}(x).$$

4.1.6 Spline

Quando il numero n di nodi in cui è nota la funzione $f(x)$ è elevato (problemi di grafica ecc..) non è conveniente utilizzare il polinomio interpolante in quanto

1. è di grado troppo elevato;
2. può presentare troppe oscillazioni (grafico non "liscio", non smooth);
3. la costruzione diviene instabile (matrice di Vandermonde mal condizionata).

Per ovviare a questi inconvenienti è preferibile l'interpolazione *polinomiale a tratti* o l'uso delle *spline*. L'idea dell'interpolazione a tratti nasce dall'esigenza di mantenere basso il grado del polinomio, per cui gli n nodi di interpolazione vengono divisi in k gruppi di m ($n = k \cdot m$) in modo da costruire k polinomi di grado $m-1$ che si raccordano, a due a due, nel nodo comune (la funzione risultante è quindi a priori solo continua). Per le spline si aggiunge la richiesta di regolarità della funzione (in generale $C^{(m-1)}$) così da ottenere una curva approssimante "più liscia".

Per la costruzione dell'approssimante data dall'interpolazione polinomiale a tratti basta applicare k volte quanto visto nei paragrafi precedenti. Per quanto riguarda le spline soffermeremo la nostra attenzione su quelle lineari e le cubiche.

Definizione 4.1.1 *Dati i nodi $x_0, x_1, ..., x_n \in [a, b]$ si definisce spline interpolante di grado k una funzione $S_k(x)$ che gode delle seguenti proprietà*

1. $S_k(x) \in C^{k-1}[a, b]$; *(regolarità)*
2. $S_k(x_j) = f(x_j)$; $j = 0, 1, 2..., n$; *(interpolazione)*

3. $S_{kj}(x) \in \Pi_k \quad \forall x \in [x_j, x_{j+1}];\ j = 0, 1, 2..., n-1;$ *(forma).*

E' immediato osservare che esistono $k-1$ gradi di libertà per la costruzione delle $S_k(x)$, infatti si devono determinare $n(k+1)$ incognite (i coefficienti degli n polinomi di grado k definiti sugli n intervalli dei nodi), mentre si possono scrivere solo $k(n-1)+(n+1)$ equazioni imponendo la regolarità della funzione e l'interpolazione.

Spline lineari

E' l'equazione della "spezzata" (funzione solo continua) passante per gli $n+1$ punti dati. E' immediata la dimostrazione dell'esistenza ed unicità; la costruzione può essere effettuata banalmente utilizzando la tecnica di Lagrange (retta per due punti!).

Consideriamo

$$S_{1j}(x) = \begin{cases} 0 & x \le x_{j-1} \\ \frac{x-x_{j-1}}{x_j - x_{j-1}}, & x_{j-1} \le x \le x_j \\ \frac{x_{j+1}-x}{x_{j+1}-x_j}, & x_j \le x \le x_{j+1} \\ 0 & x \ge x_{j+1} \end{cases}$$

il cui grafico è dato in figura 4.2

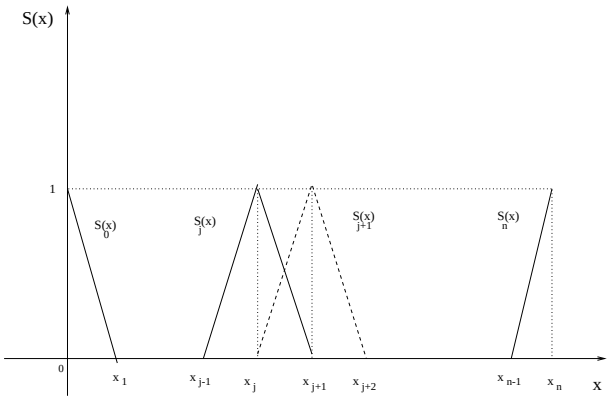


Figura 4.2:

si può allora rappresentare $S_1(x)$ nella forma

$$S_1(x) = \sum_{j=0}^{n} f(x_j) S_{1j}(x).$$

Le $S_{1j}(x)$ godono delle seguenti proprietà

1. sono linearmente indipendenti; (la funzione identicamente nulla in $[a,b]$ si ottiene solo combinando con pesi tutti nulli le S_{1j});
2. $\sum_{j=0}^{n} S_{1j}(x) = 1$; (sono una partizione dell'unità);
3. in ogni intervallo $[x_j, x_{j+1}]$ ci sono solo due $S_{1j}(x)$ diverse da zero;
4. la generica $S_{1j}(x)$ è diversa da zero solo in (x_{j-1}, x_{j+1}) (bi-spline lineari).

Per quanto riguarda l'errore d'interpolazione che si commette sostituendo alla $f(x)$ la spline lineare è possibile dimostrare il seguente risultato

Teorema 4.1.3 *Se $f(x) \in C^2[a,b]$ allora*

$$|f(x) - S_1(x)| \leq \frac{h^2}{8} \left\| f^{(2)}(x) \right\|_\infty , \quad \forall x \in [a,b]. \tag{4.6}$$

Dimostrazione 4.1.3 *Dalla definizione di $S_{1j}(x)$, $\forall x \in [x_j, x_{j+1}]$ si ha, ricordando l'errore nell'interpolazione in Lagrange,*

$$f(x) - S_1(x) = (x - x_j)(x - x_{j+1}) \frac{f^{(2)}(\xi)}{2!}$$

$\xi = \xi(x) \in [x_j, x_{j+1}]$, per cui

$$|f(x) - S_1(x)| \leq \frac{h_j^2}{4} \max_x \frac{\left| f^{(2)}(x) \right|}{2!}$$

e, posto $h = \max_j h_j$, allora

$$|f(x) - S_1(x)| \leq \frac{h^2}{8} \left\| f^{(2)}(x) \right\|_\infty . \blacksquare$$

Dalla (4.6), al crescere del numero di nodi (quindi per $h \to 0$), si ottiene la convergenza uniforme delle spline lineari.

Spline cubiche

Per $k = 3$ si hanno le spline cubiche che possono essere costruite partendo dall'interpolazione d'Hermite nel modo seguente. Dati gli $n+1$ nodi $x_0, x_1, ..., x_n$ (presi equidistanti, per semplificare la notazione), si ha

1. $S_{3i}(x) \in \Pi_3$ (forma)
2. $S_{3i}(x_{i-1}) = y_{i-1}, \qquad i = 1, 2, ..., n$
3. $S_{3i}(x_i) = y_i$; (interpolazione)
4. $S'_{3i}(x_i) = S'_{3i+1}(x_i), \qquad i = 1, 2, ..., n-1$
5. $S''_{3i}(x_i) = S''_{3i+1}(x_i)$; (regolarità)

Posto

$$f'(x_i) = m_i; \qquad h = x_i - x_{i-1}$$

dove i valori m_i NON sono noti, ma verranno determinati per costruire la spline, si ha, per l'interpolazione d'Hermite:

$$\begin{aligned} S_{3i}(x) \;=\; & \left[1 + \frac{2}{h}(x - x_{i-1})\right] \left(\frac{x - x_i}{-h}\right)^2 y_{i-1} + \\ & + \left[1 - \frac{2}{h}(x - x_i)\right] \left(\frac{x - x_{i-1}}{h}\right)^2 y_i + \\ & + (x - x_{i-1}) \left(\frac{x - x_i}{-h}\right)^2 m_{i-1} + (x - x_i) \left(\frac{x - x_{i-1}}{h}\right)^2 m_i \end{aligned} \tag{4.7}$$

dalle (4.7) segue che le $S_{3i}(x)$ verificano le prime 4 condizioni date, per la quinta condizione, essendo

$$\begin{aligned} S''_{3i}(x_{i-1}) = & \;\; \tfrac{2}{h^2}\left[3(y_i - y_{i-1}) - h(m_i + 2m_{i-1})\right] \\ S''_{3i}(x_i) = & \;\; \tfrac{2}{h^2}\left[3(y_{i-1} - y_i) + h(2m_i + m_{i-1})\right] \end{aligned}$$

si ottiene il sistema lineare, di ordine $n+1$:

$$\begin{cases} 2m_0 + m_1 = \frac{3}{h}(y_1 - y_0) \\ m_0 + 4m_1 + m_2 = \frac{3}{h}(y_2 - y_0) \\ m_1 + 4m_2 + m_3 = \frac{3}{h}(y_3 - y_1) \\ \ddots \quad \ddots \quad \ddots \quad \vdots \\ + m_{n-2} + 4m_{n-1} + m_n = \frac{3}{h}(y_n - y_{n-2}) \\ + m_{n-1} + 2m_n = \frac{3}{h}(y_n - y_{n-1}) \end{cases} \tag{4.8}$$

dove la prima e l'ultima equazione sono ottenute (spline naturali) imponendo

$$S_{31}''(x_0) = S_{3n-1}''(x_n) = 0.$$

Il sistema (4.8) è tridiagonale e fortemente diagonalizzato per cui esiste un'unica soluzione calcolabile con un numero di flops dell'ordine di n. Per le spline con data tangente (m_0 e m_n date, *spline vincolate*) il sistema (4.8) si riduce a $n-1$ equazione in $n-1$ incognite "scaricando" m_0 e m_n nel vettore termini noti.

Una giustificazione della liscezza (*smoothness*) delle spline cubiche naturali è data dal seguente

Teorema 4.1.4 *Fra tutte le funzioni $f(x) \in C^2[a,b]$ con $f(x_i) = y_i$ e $x_i \in [a,b]$, la spline cubica naturale $S_3(x)$ interpolante è tale per cui*

$$\int_a^b \left[S_3''(x)\right]^2 dx < \int_a^b \left[f''(x)\right]^2 dx. \blacksquare$$

Il teorema 4.1.4 può essere parafrasato dicendo che la spline cubica naturale passa per tutti i punti fissati "oscillando" il meno possibile.

Per quanto riguarda l'errore è possibile dimostrare i seguenti risultati

Teorema 4.1.5 *Se $f(x) \in C^2[a,b]$ la spline cubica interpolante $S_3(x)$ è tale per cui*

$$\|f(x) - S_3(x)\|_\infty \leq \frac{2}{3} h^2 \left\|f^{(2)}(x)\right\|_\infty . \blacksquare$$

Teorema 4.1.6 *Se $f(x) \in C^4[a,b]$ la spline cubica interpolante $S_3(x)$ è tale per cui, per $r = 0,1,2,3$*

$$\left\|f^{(r)}(x) - S_3^{(r)}(x)\right\|_\infty \leq c_r h^{4-r} \left\|f^{(4)}(x)\right\|_\infty . \blacksquare$$

Gli ultimi due teoremi non solo ci garantiscono la convergenza uniforme della spline $S_3^{(r)}(x)$ alla $f^{(r)}(x)$, ma ci evidenziano come questa sia più rapida al crescere della regolarità di $f(x)$ (più lenta al crescere dell'ordine della derivata!).

4.1.7 Derivazione numerica

Con il termine di derivazione numerica si intende il calcolo della derivata prima (o superiore) di un funzione $f(x)$ nota solo in un numero finito di punti x_i. Avendo calcolato il polinomio o la spline interpolanti è naturale pensare di utilizzare le loro derivate come stime delle derivate della funzione $f(x)$. Mentre per le funzioni spline il teorema 4.1.6 ci garantisce la convergenza uniforme (sulle prime 3 derivate) altrettanto non si può dire per le formule derivanti dall'interpolazione polinomiale. In generale è anzi vero il contrario, al crescere del numero dei nodi (tendere a zero della distranza fra due nodi consecutivi), a parità d'errore sui dati, l'errore totale cresce.

In definitiva derivare numericamente con polinomi di grado elevato può essere assai pericoloso. Per meglio comprendere la precedente affermazione consideriamo la seguente formula di derivazione (*differenza centrale*)

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h} - \frac{h^2}{6} f^{(3)}(\xi)$$

dove $h = x_{i+1} - x_i$ e $\xi \in [x_{i-1}, x_{i+1}]$. L'errore di troncamento che tale formula comporta va a zero come h^2, si può però osservare che se i dati $f(x_i)$ sono noti con un errore prefissato ϵ nel calcolo di

$$\frac{f(x_{i+1}) - f(x_{i-1})}{2h}$$

si commente un errore di arrotondamento dell'ordine di $\frac{\epsilon}{h}$ che è sempre maggiore al diminuire di h. Più precisamente posto $M = \max \left| f^{(3)}(\xi) \right|$ si ha

$$E(h) = \left| f'(x_i) - \frac{f(x_{i+1}) - f(x_{i-1})}{2h} \right| \simeq \frac{\epsilon}{h} + M\frac{h^2}{6} \tag{4.9}$$

che è minimo, come è facile verificare derivando rispetto ad h, quando

$$h = h^* = \sqrt[3]{\frac{3\epsilon}{M}}. \tag{4.10}$$

Il valore h^* prende il nome di *h-ottimo* in quanto minimizza l'errore totale dato dalla (4.9). Dalla (4.10) si vede bene che se ϵ è grande (errori grandi sui dati) anche h deve essere preso grande (punti "distanti") per ridurre l'effetto dell'errore d'arrotondamento (anche se l'errore di troncamento crescerà!).

Lasciamo al lettore la verifica dell'errore di troncamento introdotto dalle seguenti formule di derivazione

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} - \frac{h}{2} f^{(2)}(\xi)$$

$$f'(x_i) = \frac{f(x_i) - f(x_{i-1})}{h} + \frac{h}{2} f^{(2)}(\xi)$$

$$f''(x_i) = \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{h^2} - \frac{h^2}{12} f^{(4)}(\xi). \qquad (4.11)$$

Altre formule di derivazione, anche per derivate di ordine più elevato, possono essere reperite in letteratura.

4.2 Minimi quadrati (M.F.)

Analogamente a quanto fatto nel paragrafo 2.5, relativamente alla risoluzione di sistemi lineari nel senso dei minimi quadrati, ci proponiamo ora di costruire il *polinomio di grado m approssimante*, nel senso dei minimi quadrati, una data funzione. Questo polinomio è più generale del *polinomio interpolante* (i due coincidono quando m coincide con il numero n dei punti d'interpolazione) ed esiste anche se $m < n$.

Siano dati gli n punti (x_i, y_i) $i = 1, 2, ..., n$, dove $y_i = f(x_i)$ è il valore assunto dalla funzione f in x_i, consideriamo il polinomio di grado m

$$p_m(x) = \sum_{j=0}^{m} \alpha_j x^j; \qquad m < n$$

e cerchiamo gli $m + 1$ valori α_j che minimizzano

$$J(\underline{\alpha}) = \sum_{i=1}^{n} [f(x_i) - p_m(x_i)]^2 = \sum_{i=1}^{n} \left[f(x_i) - \sum_{j=0}^{m} \alpha_j x_i^j \right]^2. \qquad (4.12)$$

Derivando la (4.12) rispetto ad α_k $(k = 0, 1, ..., m)$ ed uguagliando a zero le $m + 1$ equazioni che si ottengono si ha

$$\frac{\partial J(\underline{\alpha})}{\partial \alpha_k} = -2 \sum_{i=1}^{n} \left\{ \left[f(x_i) - \sum_{j=0}^{m} \alpha_j x_i^j \right] x_i^k \right\} = 0; \quad k = 0, 1, ..., m;$$

che, scritto in forma matriciale diviene

$$\begin{cases} n\alpha_0 & +\alpha_1 \sum_i x_i & \cdots & +\alpha_m \sum_i x_i^m & = \sum_i y_i \\ \alpha_0 \sum_i x_i & +\alpha_1 \sum_i x_i^2 & \cdots & +\alpha_m \sum_i x_i^{m+1} & = \sum_i y_i x_i^1 \\ \alpha_0 \sum_i x_i^2 & +\alpha_1 \sum_i x_i^3 & \ddots & +\alpha_m \sum_i x_i^{m+2} & = \sum_i y_i x_i^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_0 \sum_i x_i^{m-1} & +\alpha_1 \sum_i x_i^m & \cdots & +\alpha_m \sum_i x_i^{2m-1} & = \sum_i y_i x_i^{m-1} \\ \alpha_0 \sum_i x_i^m & +\alpha_1 \sum_i x_i^{m+1} & \cdots & +\alpha_m \sum_i x_i^{2m} & = \sum_i y_i x_i^m \end{cases} . \quad (4.13)$$

Definiamo la matrice (di Vandermonde rettangolare)

$$V = \begin{bmatrix} 1 & x_1 & x_1^2 & \cdots & x_1^m \\ 1 & x_2 & x_2^2 & \cdots & x_2^m \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n-1} & x_{n-1}^2 & \cdots & x_{n-1}^m \\ 1 & x_n & x_n^2 & \cdots & x_n^m \end{bmatrix} \quad (4.14)$$

ed i vettori

$$\underline{\alpha} = \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_{m-1} \\ \alpha_m \end{bmatrix}; \qquad \underline{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-1} \\ y_n \end{bmatrix}$$

il sistema (4.13) si può scrivere nella forma

$$V^T V \underline{\alpha} = V^T \underline{y};$$

(si confronti il paragrafo 2.5 sui sistemi sovradeterminati).

Osservazione 4.2.1 *La matrice (4.14) è stata ottenuta considerando la base di polinomi x^k ($k = 0, 1, ..., m$), che compaiono nella rappresentazione di $p_m(x) = \sum_{j=0}^{m} \alpha_j x^j$, valutati nei nodi x_i ($i = 1, 2, ..., n$). Se si cambia la base (o se si considerano differenti funzioni approssimanti, per esempio spline) basta cambiare la matrice (4.14), restando tutto il resto inalterato.*

4.2.1 Polinomi trigonometrici

Consideriamo brevemente il caso dei *polinomi trigonometrici di Fourier* il cui utilizzo, ed importanza nelle applicazioni, non è certo il caso di ricordare qui.

Data una funzione $f(x)$, periodica in $[0, 2\pi]$, in n nodi equidistanti x_i $(i = 1, 2, ..., n)$ si vuole determinare il polinomio trigonometrico

$$F_m(x) = \frac{a_0}{2} + \sum_{j=0}^{m} a_j \cos(jx) + \sum_{j=0}^{m} b_j \sin(jx)$$

che minimizza

$$\begin{aligned} J(a_j, b_j) &= \sum_{i=1}^{n} [f(x_i) - F_m(x_i)]^2 = \\ &= \sum_{i=1}^{n} \left[f(x_i) - \left(\frac{a_0}{2} + \sum_{j=0}^{m} a_j \cos(jx) + \sum_{j=0}^{m} b_j \sin(jx) \right) \right]^2 . \end{aligned}$$

Per quanto precisato nell'osservazione 4.2.1 il calcolo dei coefficienti a_j e b_j richiede la risoluzione, nel senso dei minimi quadrati, del sistema lineare

$$F^T F \underline{c} = F^T \underline{y} \tag{4.15}$$

dove F, $\underline{c}$ ed $\underline{y}$ sono dati da

$$F = \begin{bmatrix} 1 & \cos(x_1) & \cdots & \cos(mx_1) & \sin(x_1) & \cdots & \sin(mx_1) \\ 1 & \cos(x_2) & \cdots & \cos(mx_2) & \sin(x_2) & \cdots & \sin(mx_2) \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ 1 & \cos(x_{n-1}) & \cdots & \cos(mx_{n-1}) & \sin(x_{n-1}) & \cdots & \sin(mx_{n-1}) \\ 1 & \cos(x_n) & \cdots & \cos(mx_n) & \sin(x_n) & \cdots & \sin(mx_n) \end{bmatrix}$$

$$\underline{c} = \begin{bmatrix} \frac{a_0}{2} \\ a_1 \\ \vdots \\ a_m \\ b_1 \\ \vdots \\ b_m \end{bmatrix} ; \qquad \underline{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{n-1} \\ y_n \end{bmatrix} .$$

La risoluzione del sistema (4.15), se i nodi x_i sono equidistanti, non richiede, come ci si potrebbe aspettare, un numero di operazioni dell'ordine $n^3 + mn$ flops, ma solo mn, in quanto la matrice F risulta "*quasi-ortogonale*" per cui basta calcolare il vettore $F^T \underline{y}$.

Osservazione 4.2.2 *Si verifichi, utilizzando il programma MATLAB, che*

$$F^T F = diag(m, \frac{m}{2}, ..., \frac{m}{2}).$$

Osservazione 4.2.3 *Se si cerca il polinomio di Fourier interpolante ($m = n$) la matrice F risulta quadrata per cui basta risolvere il sistema*

$$F\underline{c} = \underline{y},$$

che, sfruttando le proprietà di F può essere risolto in soli $n \log_2 n$ flops (trasformata veloce di Fourier FFT).

4.3 Riepilogo

Sintetizzando quanto visto nel presente capitolo possiamo dire che

1. L'interpolazione polinomiale di Lagrange è utile quando si hanno pochi punti d'interpolazione.

2. Le differenze divise sono più efficienti computazionalmente quando si devono aggiungere dei nodi (si salvano i calcoli già fatti).

3. L'nterpolazione d'Hermite è utile quando si hanno informazioni anche sulle derivate della funzione.

4. Le spline sono preferibili quando si devono ricostruire linee "lisce" (computer graphics, CAD ecc..); sono molto accurate se la funzione d'approssimare è regolare.

5. La derivazione numerica è pericolosa in quanto gli errori d'arrotondamento crescono al tendere a zero della distanza fra i nodi.

6. I minimi quadrati sono utili per individuare "modelli" quando i dati sono affetti da errore. La scelta delle funzioni di base dipendono dal modello, mentre i coefficienti lo caratterizzano.

7. L'utilizzo dell'approssimazione mediante polinomi di Fourier, indipendentemente dalle innumerevoli applicazioni pratiche, è anche computazionalmete poco oneroso grazie alle peculiarità delle funzioni di base e all'algoritmo FFT.

4.3.1 Esercizi

Esercizio 4.3.1 *Costruire il polinomio interpolante i dati*

x_i	1	2	3	4
y_i	0.84	0.90	0.14	-0.75

.

Esercizio 4.3.2 *Sapendo che un certo fenomeno si evolve nel tempo con la seguente legge*

$$f(t) = at + b \log t$$

determinare, nel senso dei minimi quadrati, i parametri a e b sapendo che

t	2	2.5	3.5	4
$f(t)$	12.54	16.08	22.27	25.09

.

Esercizio 4.3.3 *Dati tre punti nel piano, non allineati, costruire la cubica passante per il primo ed il terzo ed ivi tangente alle 2 rette uscenti dal secondo punto. E' unica ? Perchè ?*

Esercizio 4.3.4 *Scrivere un programma in MATLAB che risolva l'esercizio precedente e disegni i 3 punti, le 2 rette e la cubica. (Meglio se il programma prevede l'input grafico dei 3 punti).*

Esercizio 4.3.5 *Ricavare la formula di derivazione (4.11).*

Capitolo 5

Formule di Quadratura (M.Frontini)

Data una funzione $f(x)$ continua nell'intervallo $[a, b]$ dovendo calcolare

$$I(f) = \int_a^b f(x)dx \tag{5.1}$$

è ben noto (teorema fondamentale del calcolo integrale) che esiste $F(x) \in C^1[a, b]$ funzione primitiva di $f(x)$ ($F'(x) \equiv f(x)$), ed inoltre

$$I(f) = F(b) - F(a). \tag{5.2}$$

E' interessante osservare che mentre la (5.1) rappresenta un *numero reale,* la (5.2) afferma che tale numero può essere determinato se si conosce una delle infinite *funzioni primitive* di $f(x)$. E' chiara la differenza di "quantità di informazioni" in gioco ? (La conoscenza di un numero contro la conoscenza di una, anzi infinite, funzioni!!).

Non sempre la funzione primitiva, che pur esiste se $f(x)$ è continua, è nota o è facilmente computabile, per cui ci si propone il compito di calcolare $I(f)$ utilizzando solo le informazioni date in (5.1) (intervallo d'integrazione $[a, b]$ e funzione integranda $f(x)$).

Le formule di quadratura che considereremo, ricavabili dall'interpolazione di Lagrange, sono della forma

$$I(f) = Q_n(f) + R(Q_n; f) = \sum_{j=0}^{n} A_j^{(n)} f(x_j) + R(Q_n; f) \tag{5.3}$$

dove gli n *pesi* $A_j^{(n)}$ e gli n *nodi* $x_j \in [a,b]$, della *formula di quadratura* $Q_n(f)$, sono scelti in modo che il *resto* $R(Q_n;f)$ sia "piccolo" (in dipendenza della regolarità di $f(x)$ e dal numero di nodi n).

5.1 Formule di Newton-Cotes (M.F.)

Dall'interpolazione di Lagrange abbiamo che, fissati i nodi $x_i \in [a,b]$, è possibile costruire il polinomio interpolante $p_n(x)$ tale per cui

$$f(x) = p_n(x) + e_n(x) \tag{5.4}$$

dove $e_n(x)$ è l'errore d'interpolazione.

Integrando la (5.4) si ottiene

$$\int_a^b f(x)dx = \int_a^b p_n(x)dx + \int_a^b e_n(x)dx$$

essendo

$$p_n(x) = \sum_{j=0}^{n} f(x_j)l_j(x)$$

dove $l_j(x)$ sono i polinomi di Lagrange sui nodi x_i, si ha

$$Q_n(f) = \sum_{j=0}^{n} f(x_j) \int_a^b l_j(x)dx = \sum_{j=0}^{n} A_j^{(n)} f(x_j)$$

dove

$$A_j^{(n)} = \int_a^b l_j(x)dx \tag{5.5}$$

sono i pesi della formula di quadratura e

$$R(Q_n;f) = \int_a^b e_n(x)dx$$

è l'errore che si commette sostituendo ad $I(f)$ il valore $Q_n(f)$.

Osservazione 5.1.1 *Dalla (5.5) è immediato osservare che i pesi dipendono dai nodi d'interpolazione (zeri della formula di quadratura) ed inoltre la formula sarà esatta se la $f(x)$ è un polinomio di grado al più n.*

Definizione 5.1.1 *Una formula del tipo (5.3) si dice di ordine n se è esatta per polinomi di grado al più n.*

Dall'osservazione 5.1.1 e dalla precedente definizione segue che le formule di quadratura interpolatorie si possono costruire utilizzando il concetto di ordine di precisione determinando i pesi in modo che l'ordine risulti massimo.

Posto

$$Q_n(f) = \sum_{j=0}^{n} A_j^{(n)} f(x_j)$$

consideriamo i monomi x^k $(k = 0, 1, ..., n)$ ed imponiamo che

$$I(x^k) = \int_a^b x^k dx = \frac{1}{k+1}(b^{k+1} - a^{k+1}) = \sum_{j=0}^{n} A_j^{(n)} x_j^k = Q_n(x^k)$$

otteniamo il sistema lineare, nelle incognite $A_j^{(n)}$,

$$\sum_{j=0}^{n} A_j^{(n)} x_j^k = \frac{1}{k+1}(b^{k+1} - a^{k+1}). \tag{5.6}$$

Esempio 5.1.1 *Dato l'intervallo* $[0, h]$, *posto* $x_0 = 0$ *e* $x_1 = h$, *costruiamo la formula del primo ordine.*

Avremo

$$Q_1(f) = A_0 f(x_0) + A_1 f(x_1)$$

da cui

$$\begin{cases} \int_0^h dx = h = A_0 + A_1 \\ \int_0^h x dx = \frac{h^2}{2} = A_0 0 + A_1 h \end{cases}$$

e quindi

$$\begin{cases} A_0 + A_1 = h \\ A_1 h = \frac{h^2}{2} \end{cases} \Longrightarrow A_0 = A_1 = \frac{h}{2}$$

che è la formula dei Trapezi

$$Q_1(f) = \frac{h}{2}\left[f(x_0) + f(x_1)\right].\square \tag{5.7}$$

Analogamente è possibile costruire la *formula di Simpson* su tre nodi equidistanti nell'intervallo $[-h, h]$ ($x_0 = -h$, $x_1 = 0$, $x_2 = h$)

$$Q_2(f) = \frac{h}{3}\left[f(x_0) + 4f(x_1) + f(x_2)\right].\square \tag{5.8}$$

La formula dei trapezi poteva essere ottenuta, mediante l'interpolazione di Lagrange, costruendo la retta passante per i punti $(0, f(0))$ e $(h, f(h))$.

Ricordando che, in questo caso, l'errore d'interpolazione è

$$e_1(x) = f(x) - p_1(x) = \frac{x(x-h)}{2} f''(\xi); \quad \xi = \xi(x) \in [0, h]$$

abbiamo

$$R(Q_1; f) = \int_0^h e(x)dx = \int_0^h \frac{x(x-h)}{2} f''(\xi)dx.$$

Applicando il teorema della media (essendo $x(x-h)$ di segno costante in $[0, h]$) si ha

$$\begin{aligned} R(Q_1; f) &= \int_0^h \frac{x(x-h)}{2} f''(\xi)dx = f''(\eta) \int_0^h \frac{x(x-h)}{2} dx = \\ &= \left[\frac{x^3}{6} - \frac{hx^2}{4}\right]_0^h f''(\eta) = -\frac{h^3}{12} f''(\eta), \end{aligned}$$

che ci fornisce l'errore per la formula dei trapezi.

Osservazione 5.1.2 *La presenza, nella formula dell'errore, della derivata seconda della funzione integranda riconferma che la formula è esatta per polinomi di primo grado.*

Per ottenere l'errore nella formula di Simpson osserviamo che

$$\int_{-h}^h x^3 dx = 0 = \frac{h}{3}\left[-h^3 + 4 \cdot 0 + h^3\right]$$

per cui la formula risulta esatta (sorpresa!) anche per polinomi di terzo grado.

Se applichiamo la formula di Simpson al monomio x^4, abbiamo

$$\int_{-h}^h x^4 dx = \frac{2}{5}h^5 \neq \frac{h}{3}\left[(-h)^4 + 4 \cdot 0 + h^4\right] = \frac{2}{3}h^5$$

per cui, in questo caso, si commette un errore pari a $-\frac{4}{15}h^5$.

Se la funzione $f(x)$ è sufficientemente regolare possiamo considerarne lo sviluppo di Mac-Laurin arrestato al quarto ordine, per cui

$$f(x) = f(0) + xf'(0) + \frac{x^2}{2}f''(0) + \frac{x^3}{3!}f'''(0) + \frac{x^4}{4!}f^{iv}(\xi)$$

ed integrando termine a termine si commetterà errore solo sull'ultimo addendo (i primi 4 termini sono polinomi di grado minore od uguale al terzo), per cui

$$R(Q_2; f) = -\frac{4}{15}h^5\frac{1}{4!}f^{iv}(\xi) = -\frac{1}{90}h^5 f^{iv}(\xi).$$

Osservazione 5.1.3 *Il guadagno di un ordine di precisione (con 3 nodi si ha ordine 3 e non 2) non è peculiare del metodo di Simpson ma è proprio di tutte le formule di Newton-Cotes su un numero dispari di nodi equidistanti.*

A sostegno della precedente osservazione ricordiamo che per la *formula dei rettangoli* (si ricordano gli "scaloidi " nella definizione di integrabilità secondo Reimann?) sull'intervallo $[-h, h]$

$$Q_0(f) = 2hf(0)$$

comporta un errore

$$R(Q_0; f) = +\frac{h^3}{3}f''(\xi) \tag{5.9}$$

per cui risulta esatta non solo per costanti, ma anche per rette. Una chiara interpretazione geometrica di questo risultato è dato dalla seguente figura 5.1. La scelta, del tutto arbitraria, dell'intervallo $[0, h]$ piuttosto che $[-h, h]$, nella determinazione delle formule precedenti, è giustificata dal fatto che è sempre possibile passare, con un opportuno cambio di variabile, da un intervallo d'integrazione ad un altro. Analogamente è possibile, nota una formula di quadratura (nodi e pesi) su un certo intervallo, ottenere l'equivalente formula su un altro intervallo.

Supponiamo di dover calcolare

$$I(f) = \int_a^b f(x)dx$$

avendo a disposizione la formula di quadratura

$$Q_n(f) = \sum_{j=0}^{n} A_j f(t_j) \simeq \int_{-1}^{+1} f(t)dt. \tag{5.10}$$

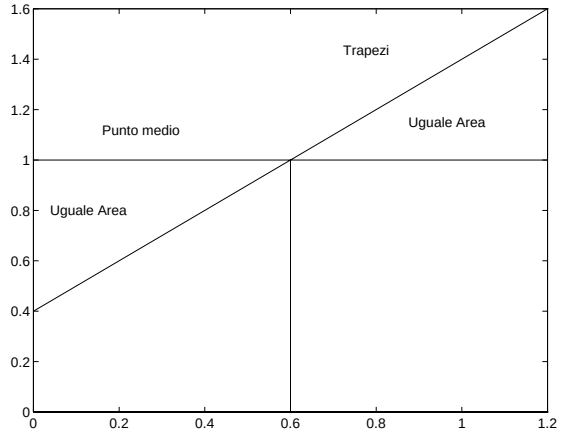


Figura 5.1:

Come è possibile ottenere la formula

$$Q_n^*(f) = \sum_{j=0}^{n} A_j^* f(x_j) \simeq \int_a^b f(x)dx \tag{5.11}$$

dalla (5.10) ? Basta considerare il cambiamento di variabile

$$x = \frac{b-a}{2}t + \frac{b+a}{2} \tag{5.12}$$

per cui si ha

$$\begin{aligned} \int_a^b f(x)dx &= \int_{-1}^{+1} f(\frac{b-a}{2}t + \frac{b+a}{2})\frac{b-a}{2}dt \\ &\simeq \frac{b-a}{2}\sum_{j=0}^{n} A_j f(t_j) = \sum_{j=0}^{n} A_j^* f(x_j) \end{aligned}$$

e quindi i pesi della $Q_n^*(f)$ sono dati da

$$A_j^* = \frac{b-a}{2}A_j$$

mentre i nodi x_j, ottenuti dalla (5.12) sono

$$x_j = \frac{b-a}{2}t_j + \frac{b+a}{2}.\blacksquare$$

Esistono due tipi di formule di Newton-Cotes

1. formule di *tipo chiuso*: comprendono fra i nodi anche gli estremi dell'intervallo d'integrazione (vedi Trapezi e Simpson);

2. formule di *tipo aperto*: utilizzano solo nodi interni all'intervallo d'integrazione (vedi punto medio).

Le formule di tipo aperto sono utili, per esempio, quando si hanno singolarità agli estremi.

Per formule di Newton-Cotes di ordine più elevato si rimanda alla bibliografia.

5.1.1 Formule composite

Le stime dell'errore fornite sono state ottenute considerando, fissato l'intervallo d'integrazione ed il numero di nodi, il polinomio *interpolante globale*. Se è vero che, al crescere del numero dei nodi, cresce l'ordine della derivata della $f(x)$ che compare nella formula dell'errore (maggiore *ordine d'accuratezza*), è altresì vero che si ha anche una potenza crescente dell'intervallo d'integrazione (che è però fissato, quindi *costante*). Come nell'interpolazione si è passati, per motivi di convergenza, dal polinomio interpolante globale ai *polinomi a tratti* (spline) così ora converrà suddividere l'intervallo d'integrazione $[a, b]$ (additività dell'integrale) in tanti sotto intervalli di ampiezza h ed applicare, su ogni sotto intervallo, la formula prescelta (convergenza per $h \to 0$). E' l'applicazione del vecchio principio di Cesare Augusto "*divide et impera*".

Si ottengono così le *formule composite* (o generalizzate), di cui proveremo la convergenza nel caso dei trapezi. Consideriamo

$$I(f) = \int_a^b f(x)dx \overset{additività}{=} \sum_{k=0}^{m-1} \int_{x_k}^{x_{k+1}} f(x)dx = \sum_{k=0}^{m-1} I_k(f)$$

dove

$$x_0 = a; \quad x_m = b; \quad h = \frac{b-a}{m}$$

e

$$I_k(f) = \int_{x_k}^{x_{k+1}} f(x)dx \simeq \sum_{j=0}^{n} A_j f(x_j) = Q_n(f; I_k),$$

con $Q_n(f; I_k)$ si è indicata la formula $Q_n(f)$, di ordine n, applicata sull'intervallo I_k.

Per la formula dei trapezi, applicata a I_k, abbiamo

$$\int_{x_k}^{x_{k+1}} f(x)dx = \frac{h}{2}\left[f(x_k) + f(x_{k+1})\right] - \frac{h^3}{12}f''(\xi_k)$$

da cui la formula composta

$$T_m(f) = \frac{h}{2}\sum_{k=0}^{m-1}\left[f(x_k) + f(x_{k+1})\right].$$

Per l'errore abbiamo quindi

$$E_T = I(f) - T_m(f) = -\sum_{k=0}^{m-1}\frac{h^3}{12}f''(\xi_k), \tag{5.13}$$

che può essere semplificata nella forma

$$E_T = -f''(\eta)\sum_{k=0}^{m-1}\frac{h^3}{12} = -m\frac{h^3}{12}f''(\eta) = -(b-a)\frac{h^2}{12}f''(\eta).\blacksquare \tag{5.14}$$

Per ottenere dalla (5.13) la (5.14), abbiamo fatto uso del seguente

Lemma 5.1.1 *Data una funzione $g(x) \in C^0[a,b]$ e dei coefficienti di segno costante a_k, allora esiste un $\eta \in [a,b]$ tale per cui, $\forall x_k \in [a,b]$, si ha*

$$\sum_{k=0}^{m-1} a_k g(x_k) = g(\eta)\sum_{k=0}^{m-1} a_k.\square$$

Osservazione 5.1.4 *Si osservi che l'errore nella formula dei trapezi composita è quindi della forma ch^2 dove*

$$c = -\frac{(b-a)}{12}f''(\eta)$$

e va a zero, quando $h \to 0$, come h^2 (convergenza quadratica).

Analogamente per la formula di Simpson composita si può dimostrare che l'errore è della forma Ch^4, infatti, posto

$$\int_{x_k}^{x_{k+2}} f(x)dx = \frac{h}{3}\left[f(x_k) + 4f(x_{k+1}) + f(x_{k+2})\right] - \frac{h^5}{90}f^{iv}(\xi_k)$$

si ha

$$S_m(f) = \sum_{k=0}^{m-1} \frac{h}{3}\left[f(x_{2k}) + 4f(x_{2k+1}) + f(x_{2k+2})\right]$$

dove

$$x_0 = a; \quad x_{m+1} = b; \quad h = \frac{b-a}{2m}$$

per cui, sempre in virtù del Lemma 5.1.1,

$$E_S = I(f) - S_m(f) = -\sum_{k=0}^{m-1} \frac{h^5}{90}f^{iv}(\xi_k) = -(b-a)\frac{h^4}{180}f^{iv}(\eta).\blacksquare \qquad (5.15)$$

Osservazione 5.1.5 *Dalle (5.14,5.15) si sarebbe tentati di prendere n "infinitamente grande" per avere h "infinitamente piccolo", ed ottenere quindi un errore "trascurabile". Nella pratica NON ha senso prendere n "troppo grande" in quanto, oltre ad aumentare il tempo di calcolo (troppe valutazioni della funzione integranda), si rischia che, da un certo punto in poi, gli errori di calcolo (epsilon macchina) non facciano guadagnare in precisione (si può anzi peggiorare il risultato!).*

5.2 Formule adattive (M.F.)

In tutte le stime dell'errore fornite è implicitamente sottointesa una certa regolarità della funzione integranda $f(x)$ (devono almeno esistere le derivate in gioco nelle varie formule) ma non si dice quanto valgono le costanti c e C che dipendono dal punto η, per cui alla domanda: *"quanti punti si devono prendere per avere un errore prefissato"* ? Non possiamo dare una risposta a priori, senza conoscere η.

Sovente non è possibile determinare preventivamente η (è difficile, se non impossibile, maggiorare la derivata di $f(x)$ che compare nella formula dell'errore), in questi casi sono utili le *formule adattive* (note anche come *integratori automatici* o programmi con controllo automatico dell'errore).

Chiariamo con un semplice esempio di cosa si tratta.

L'idea di base è quella di fornire, se possibile, noti la funzione integranda $f(x)$ e l'intervallo d'intregrazione $[a,b]$, il valore dell'integrale $I(f)$ con una prefissata precisione (*toll*). Cercheremo di fare questo seguendo le seguenti due linee

1. usare l'*errore locale* per controllare l'*errore globale*;
2. ridurre al minimo le valutazioni della funzione integranda.

Esemplifichiamo tale processo mediante la ben nota formula di Simpson. Utilizzando il metodo di Simpson sull'intero intervallo $[a,b]$, si ha

$$\int_a^b f(x)dx - S_1(f) = -\frac{1}{90}\left(\frac{b-a}{2}\right)^5 f^{iv}(\xi) \tag{5.16}$$

mentre, dividendo in due parti l'intervallo, si ha

$$\int_a^b f(x)dx - S_2(f) = -2\frac{1}{90}\left(\frac{b-a}{4}\right)^5 f^{iv}(\eta) \tag{5.17}$$

dove ξ è, in generale, diverso da η. Supponendo, e questa è una condizione forte se $[a,b]$ è grande, mentre è "ragionevole", se $f(x)$ è abbastanza regolare e $[a,b]$ è "piccolo", che

$$f^{iv}(x) \simeq \text{costante in } [a,b],$$

posto

$$I(f) = \int_a^b f(x)dx$$

si ottiene, dalle (5.16,5.17), che

$$I(f) - S_2(f) \simeq \frac{1}{16}\left(I(f) - S_1(f)\right)$$

per cui

$$16\left(I(f) - S_2(f)\right) \simeq I(f) - S_1(f).$$

Sottraendo ad entrambi i menbri la quantità $I(f) - S_2(f)$, si ha

$$15\left(I(f) - S_2(f)\right) \simeq S_2(f) - S_1(f)$$

e quindi

$$I(f) - S_2(f) \simeq \frac{S_2(f) - S_1(f)}{15}. \tag{5.18}$$

Nella (5.18) l'errore commesso intregrando con la formula di Simpson su 2 sotto intervalli è controllato dalla quantità a secondo membro che, essendo legata alla differenza fra due integrazioni numeriche, è computabile.

Osservazione 5.2.1 *Volendo essere pessimisti, e quindi salvaguardarsi nella stima dell'errore, la costante* $\frac{1}{15}$ *che compare nella (5.18) può essere sostituita da* $\frac{1}{2}$.

Un algoritmo che sintetizza quanto detto può essere così formulato:

1. si parte dall'intervallo $[a, b]$ e con la tolleranza *toll*;
2. si calcolano $S_1(f)$ e $S_2(f)$;
   (a) se $\left|\frac{S_2(f)-S_1(f)}{2}\right| \leq$ tolleranza, allora $S_2(f)$ è uguale all'integrale nella precisione richiesta.
3. altrimenti si dimezzano l'intervallo e la tolleranza;
   (a) se NON si sono fatte "troppe" sottodivisioni, si opera analogamente sul nuovo intervallo;
   (b) altrimenti l'integrale non può essere calcolato con la tolleranza *toll* richiesta.
4. calcolato, nella nuova precisione, l'integrale sul sotto intervallo, si torna al punto 2 con l'intervallo restante e ridotta tolleranza.□

Quanto visto per la formula di Simpson può essere esteso, con le dovute varianti, ad altre formule. Non potendoci dilungare oltre si rimanda il lettore interessato alla bibliografia specifica.

Quando la funzione $f(x)$ ha derivata seconda di segno costante in $[a, b]$ (supponiamola per esempio negativa) si può fornire una stima dell'errore, che si commette integrando numericamente la $f(x)$ in $[a, b]$, utilizzando le formule del punto medio e dei trapezi. E' facile osservare (si confronti la figura 5.2) che

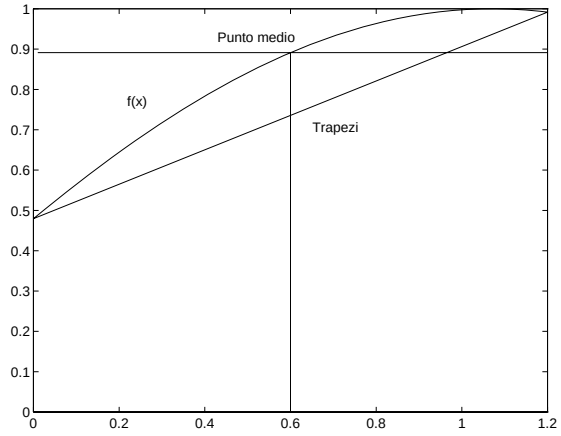


Figura 5.2:

$$I_T(f) \leq I(f) \leq I_{MP}(f),$$

dove con $I_T(f)$ si è indicato il valore ottenuto con la formula dei trapezi e con $I_{MP}(f)$ quello ottenuto con la formula del punto medio.

Utilizzando le formule composite, suddividendo l'intervallo $[a, b]$ in sotto intervalli di ampiezza h, si ha, con ovvio significato dei simboli,

$$I_T^h(f) \leq I(f) \leq I_{MP}^h(f)$$

da cui

$$0 = I_T^h(f) - I_T^h(f) \leq I(f) - I_T^h(f) \leq I_{MP}^h(f) - I_T^h(f)$$

ed inoltre

$$-\left[I_{MP}^h - I_T^h(f)\right] \leq I(f) - I_{MP}^h(f) \leq I_{MP}^h(f) - I_{MP}^h(f) = 0$$

sommando, termine a termine, la prima relazione e la seconda, si ha

$$-\left[I_{MP}^h - I_T^h(f)\right] \leq 2I(f) - \left(I_T^h(f) + I_{MP}^h(f)\right) \leq I_{MP}^h(f) - I_T^h(f)$$

per cui

$$\left| I(f) - \frac{I_T^h(f) + I_{MP}^h(f)}{2} \right| \leq \frac{I_{MP}^h(f) - I_T^h(f)}{2},$$

dove la quantità a secondo membro è computabile.

5.3 Formule gaussiane (M.F.)

Se nella formula (5.3) i nodi x_j non sono fissati a priori, (equidistanti nelle formule di Newton-Cotes) imponendo che la formula abbia il massimo ordine di accuratezza si ottiene un sistema non lineare nelle $2n+2$ incognite x_j e A_j. Gauss fu il primo a dimostrare che, sotto opportune ipotesi, tale sistema ammette una ed una sola soluzione, dando origine ad uno dei più prolifici settori della matematica.

Alla base delle formule di *quadratura gaussiane* sta la teoria dei *polinomi ortogonali.* Non potendo qui dedicarle il dovuto spazio ci limiteremo a riassumere solo alcuni degli aspetti più significativi relativi alle formule di quadratura.

Definita con

$$Q_{2n}(f) = \sum_{j=0}^{n} A_j^{(n)} f(x_j) \tag{5.19}$$

la formula di quadratura con n nodi liberi, vale il seguente

Teorema 5.3.1 *Il sistema non lineare (5.6) dove i nodi x_j sono incogniti, con $[a,b] \equiv [-1,1]$, ammette una ed una sola soluzione ed inoltre nella $Q_{2n}(f)$*

1. *i pesi $A_j^{(n)}$ sono positivi;*
2. *i nodi x_j sono gli zeri del polinomio ortogonale di Legendre di grado n ($\mathcal{L}_n(x)$);*
3. *tale formula di quadratura gaussiana è esatta per ogni polinomio di grado minore o uguale a $2n+1$.*■

Che la formula $Q_{2n}(f)$ non possa essere di grado maggiore di $2n+1$ è facilmente dimostrabile considerando il polinomio

$$\pi_{2n}(x) = \prod_{j=0}^{n} (x - x_j)^2; \quad (x_j \text{ zeri di } \mathcal{L}_n(x))$$

in quanto il suo integrale

$$\int_{-1}^{1} \pi_{2n}(x)dx = \int_{-1}^{1} \prod_{j=0}^{n} (x - x_j)^2 dx > 0$$

è una quantità strettamente maggiore di zero, mentre verrà valutato uguale a zero dalla (5.19). (Si osservi che è nullo il valore del polinomio $\pi_{2n}(x)$ in tutti i nodi della formula di quadratura).

La positività dei pesi $A_j^{(n)}$ è importante perchè permette d'assicurare la convergenza, al crescere del numero dei nodi ($n \to \infty$), delle formule gaussiane in virtù del seguente

Teorema 5.3.2 *Se $f(x) \in C^0[a,b]$ data una formula di quadratura della forma*

$$Q_n(f) = \sum_{j=0}^{n} A_j^{(n)} f(x_j)$$

se esiste una costante k tale per cui

$$\sum_{j=0}^{n} \left| A_j^{(n)} \right| < k; \quad \forall n$$

allora

$$\lim_{n\to\infty} Q_n(f) = I(f).\blacksquare$$

Osservazione 5.3.1 *Essendo $\sum_{j=0}^{n} A_j^{(n)} = b - a$ e $A_j^{(n)} > 0$ per ogni j, basterà prendere, nel teorema 5.3.2, $k = b - a$.*

Osservazione 5.3.2 *Il teorema 5.3.2 non può essere applicato nel caso delle formule di Newton-Cotes perchè, per $n > 7$, compaiono pesi $A_j^{(n)}$ negativi.*

L'esistenza delle formule gaussiane è legata all'esistenza di una famiglia di polinomi ortogonali sull'intervallo (non necessariamente finito) $[a,b]$ rispetto ad una opportuna funzione peso $\omega(x)$, non negativa in $[a,b]$.

Definizione 5.3.1 *Si dicono ortogonali in $[a,b]$ rispetto alla funzione peso $\omega(x)$, non negativa in $[a,b]$ i polinomi $p_n(x)$ che verificano*

$$\int_a^b p_i(x)p_j(x)\omega(x)dx = c_i\delta_{ij}; \quad c_i > 0.\square$$

Se $c_i = 1$ per ogni i, i polinomi si dicono *ortonormali*.

Osservazione 5.3.3 *Nel caso dei polinomi $\mathcal{L}_n(x)$ di Legendre, $[a,b] \equiv [-1,1]$ e $\omega(x) = 1$.*

Vale il seguente

Teorema 5.3.3 *Se esiste la famiglia di polinomi ortogonali in $[a,b]$ rispetto alla funzione peso non negativa $\omega(x)$ allora, dato*

$$I(f) = \int_a^b f(x)\omega(x)dx \tag{5.20}$$

esiste la formula di quadratura gaussiana

$$Q_{2n}(f) = \sum_{j=0}^{n} A_j^{(n)} f(x_j) \tag{5.21}$$

dove

$$A_j^{(n)} = \int_a^b l_j^2(x)\omega(x)dx$$

(dove $l_j^2(x)$ è il j-esimo polinomio di Lagrange di grado n sui nodi x_k $k = 0,1,...,n$) ed i nodi x_k sono gli zeri (interni all'intervallo $[a,b]$) del polinomio ortogonale di grado n.■

Osservazione 5.3.4 *Si fa osservare che mentre la funzione integranda in (5.20) comprende anche la funzione peso $\omega(x)$, nella (5.21) compare solo la $f(x)$ valutata nei nodi. Questo ci permette di rimuovere eventuali singolarità nella funzione integranda, scaricandole sulla funzione peso $\omega(x)$.*

5.3.1 Integrali impropri

Per concludere il capitolo faremo solo un breve cenno al problema del calcolo degli integrali della forma

$$I_1(f) = \int_0^{+\infty} f(x)dx; \quad I_2(f) = \int_0^1 \frac{f(x)}{\sqrt{x}}dx$$

(*integrali impropri*) dove $f(x)$ e $g(x) = \frac{f(x)}{\sqrt{x}}$ sono funzioni integrabili rispettivamente in R e in $[0,1]$. Vedremo che in questi casi si possono ancora utilizzare le formule di Newton-Cotes (eventualmente aperte), anche se le formule di Gauss risultano in generale più efficienti.

Per calcolare l'integrale I_1 con prefissata precisone (ε), si può operare, sfruttando l'additività dell'integrale, nel modo seguente, si pone

$$I_1(f) = \int_0^{+\infty} f(x)dx = \int_0^a f(x)dx + \int_a^{+\infty} f(x)dx = I_a(f) + I_\infty(f)$$

e si sceglie a in modo che

$$\left| \int_a^{+\infty} f(x)dx \right| \leq \varepsilon_1$$

(tale a deve esistere per l'integrabilità di $f(x)$) e si calcola $I_a(f)$ con una opportuna formula di Newton-Cotes $Q_n(f)$ in modo che

$$|I_a(f) - Q_n(f)| \leq \varepsilon_2$$

dove ε_1 e ε_2 sono stati scelti in modo che $\varepsilon_1 + \varepsilon_2 = \varepsilon$.

Per calcolare l'integrale I_2 è possibile utilizzare la formula del *punto medio* composita in modo da evitare il calcolo della funzione integranda in $x = 0$ (singolarità). Se la funzione $g(x)$ è sufficientemente regolare la formula del punto medio converge all'integrale I_2, anche se tale convergenza può essere assai lenta.

Osservazione 5.3.5 *Dall'osservazione 5.3.4 segue che basta costruire la famiglia di polinomi ortogonali in* $[0, 1]$ *rispetto alla funzione pesi* $\omega(x) = \frac{1}{\sqrt{x}}$ *per ottenere una formula di quadratura gaussiana che permette di calcolare in modo esatto* I_2 *ogniqualvolta* $f(x)$ *è un polinomio. (Di fatto tale formula esiste ed è correlata ai polinomi di Chebyshev).*

Per una più esauriente trattazione dell'uso delle formule gaussiane si rimanda alla bibliografia specifica.

5.4 Riepilogo (M.F.)

Sintetizzando quanto esposto nel presente capitolo abbiamo

1. Se l'intervallo $[a, b]$ è finito, e $f(x)$ è regolare le formule di Newton-Cotes composite funzionano in generale bene.

2. Si possono avere problemi coi tempi di calcolo se si richiede molta accuratezza (numero eccesivo di sottodivisioni dell'intervallo).

3. La convergenza, per le formule di Newton-Cotes, è garantita al crescere del numero di nodi (composite) non al crescere dell'ordine (grado del polinomio interpolante).

4. Volendo ottenere un risultato "con una prefissata precisione" è meglio utilizzare formule adattive (integratori automatici), anche se in generale i programmi sono più complessi.

5. Se l'intervallo $[a, b]$ è infinito si possono ancora usare le Newton-Cotes pur di spezzare opportunamente l'integrale (additività).

6. Le formule gaussiane sono più accurate, convergenti e permettono di trattare, in alcuni casi, gli integrali singolari. (Solo cenni nel corso).

5.4.1 Esercizi

Esercizio 5.4.1 *Dimostrare che il sistema (5.6) è non singolare se i nodi x_i sono distinti.*

Esercizio 5.4.2 *Costruire la formula (5.8).*

Esercizio 5.4.3 *Dimostrare la formula (5.9). (Suggerimento: costruito il polinomio di Lagrange su un punto, integrando l'errore d'interpolazione,.....)*

Esercizio 5.4.4 *Verificare che la formula del punto medio è gaussiana (Gauss-Legendre con $n = 1$).*

Esercizio 5.4.5 *Calcolare, se possibile, con errore relativo minore di 10^{-3} i seguenti integrali*

$$I_1 = \int_0^1 e^{-x^2+1} dx; \quad I_2 = \int_0^1 \frac{e^{-x^2+1}}{x} dx; \quad I_3 = \int_0^{+\infty} \frac{\cos x}{x^5 + 3} dx.$$

Capitolo 6

Equazioni differenziali ordinarie (M.Frontini)

Molti problemi della fisica, dell'ingegneria e delle discipline scientifiche in generale sono governati da equazioni differenziali ordinarie; come esempio si pensi alle equazioni

$$m\,\ddot{x} = f \qquad \text{(equazione della dinamica)}$$

$$Ri + L\frac{di}{dt} + v_c = V_s \qquad \text{(equazione circuito RLC)}$$

che permettono rispettivamente di studiare, date le opportune condizioni iniziale, il moto di un corpo di massa m soggetto ad una forza f, ed il variare, nel tempo, dell'intensità di corrente (i) e del voltaggio del condensatore (v_c) in un circuito elettrico con una resistenza (R), un'induttanza (L), un condensatore (di capacità C, con $i = C\frac{dv_c}{dt}$) ed un generatore (V_s).

Non sempre è possibile determinare in "forma chiusa", mediante i metodi forniti nei corsi di Analisi Matematica, la soluzione di tali problemi, ed anche quando ciò è possibile tale soluzione può risultare tanto complicata da non essere facilmente utilizzabile. Va notato inoltre che spesso in molte applicazioni non è necessario conoscere la soluzione ovunque ma basta averne una "buona stima" solo in alcuni punti. Per questi motivi i metodi numerici rivestono un ruolo sempre più importante nel calcolo scientifico.

Prima di addentrarci nella presentazione dei metodi numerici richiamiamo alcuni risultati classici nella teoria delle equazioni differenziali ordinarie.

Dato un problema ai valori iniziali (problema di Cauchy) nella forma

$$\begin{cases} y' = f(x,y) \\ y(a) = y_0 \end{cases} \tag{6.1}$$

dove $f(x,y)$ esprime il legame fra una funzione e la sua derivata prima, mentre la condizione iniziale $y(a) = y_0$ impone il passaggio della soluzione per il punto (a, y_0), vale il seguente

Teorema 6.0.1 *Se la funzione $f(x,y)$ del problema (6.1) è definita e continua in $a \leq x \leq b$ per ogni y ed inoltre verifica la condizione (di Lipschitz)*

$$|f(x,u) - f(x,v)| \leq L\,|u - v|$$

$$a \leq x \leq b, \quad \forall u, v \text{ ed } L = \text{costante},$$

allora esiste ed è unica la soluzione del problema, cioè

$$\exists! y(x) | y' = f(x,y) \text{ con } y(a) = y_0 . \blacksquare$$

Corollario 6.0.1 *Condizione sufficiente per l'esistenza ed unicità della soluzione di (6.1) è che esista la derivata parziale di f rispetto ad y ed inoltre,*

$$|f_y(x,y)| \leq L, \quad a \leq x \leq b, \quad \forall y . \blacksquare$$

Oltre a richiedere l'esistenza e l'unicità della soluzione del problema (6.1) è importante che il problema (6.1) sia ben posto (nel senso di *Hadamard*) ovvero che: *la soluzione, che esiste ed è unica, dipenda con continuità dai dati.* Il concetto di *problema ben posto* è importante nelle applicazioni perché ci permette di risolvere un problema partendo anche da condizioni affette da errori ("non troppo grandi") ed ottenere comunque una soluzione "abbastanza vicina" alla soluzione vera del problema.

Esempio 6.0.1 *Come esempio di problema ben posto si consideri l'equazione differenziale*

$$y' = y$$

che ammette come soluzione la famiglia di curve

$$y(x) = ce^x, \quad c \in R.$$

Se si considera il corrispondente problema di Cauchy con la condizione iniziale

$$y(0) = 1$$

si ottiene l'unica soluzione

$$y(x) = e^x.$$

Il problema è ben posto, infatti, se si considera il problema perturbato

$$\begin{cases} y' = y \\ y(0) = 1 + \epsilon \end{cases}$$

si ottiene la soluzione

$$y(x) = (1 + \epsilon)\, e^x$$

che, per ϵ piccoli, poco si discosta, dalla soluzione $y(x) = e^x$ (si valuti l'errore relativo).□

Per trattare equazioni differenziali di ordine superiore al primo, ricordiamo che un'equazione di ordine m

$$y^{(m)} = f(x, y, y', y'', ..., y^{(m-1)}),$$

può essere scritta sotto forma di sistema di m equazioni del primo ordine

$$\begin{cases} z_1'(x) = z_2(x) \\ z_2'(x) = z_3(x) \\ \vdots \\ z_{m-1}'(x) = z_m(x) \\ z_m'(x) = f(x, z_1, z_2, ..., z_m) \end{cases},$$

dove si è posto

$$z_1(x) = y(x);\ z_2(x) = y'(x);\ z_3(x) = y''(x); ...; z_m(x) = y^{(m-1)}(x)$$

per cui, date le opportune condizioni iniziali, siamo ricondotti ad un problema di Cauchy

$$\begin{cases} \underline{z}' = \underline{f}(x, \underline{z}) \\ \underline{z}(a) = \underline{z}_0 \end{cases}$$

formalmente equivalente al sistema (6.1).

Fatte salve le ipotesi dei teoremi precedenti, ci proponiamo ora di fornire metodi numerici che, partendo dal valore iniziale dato nel problema (6.1), forniscano la soluzione in prefissati punti x_i di un dato intervallo $[a, b]$.

6.1 Metodi ad un passo (M.F.)

I metodi ad un passo sono metodi numerici che, fissato un passo d'integrazione h, determinano la soluzione nel punto $x_1 = a+ h$, $x_2 = a+ 2h$,... partendo dal valore noto in $x_0 = a$, mediante relazioni della forma

$$y_{n+1} = y_n + h\phi(x_n, y_n; h); \quad n = 0, 1, 2, ...$$

dove con y_n si è indicata la *stima numerica* di $y(\ x_n)$, mentre $\phi(x_n, y_n; h)$ è una funzione che dipenderà da $f(x_n, y_n)$ ed h. (I valori trovati al passo $n+1$ dipendono *solo* dai valori ottenuti al passo precedente, *metodi ad un passo*).

Esistono due grandi famiglie di metodi ad un passo di cui parleremo nel seguito

1. i metodi di Taylor,
2. i metodi di Runge-Kutta.

6.1.1 Metodi di Taylor

I metodi di Taylor si basano sull'utilizzo dello sviluppo in serie di Taylor della funzione incognita $y(x)$, più precisamente dato il problema

$$\begin{cases} y' = f(x, y) \\ y(x_0) = y_0 \end{cases}, \tag{6.2}$$

supposta $y(x)$ "sufficientemente regolare" si ha

$$y(x_0 + h) = y(x_1) = y(x_0) + hy'(x_0) + \frac{h^2}{2}y''(x_0) + ... + \frac{h^k}{k!}y^{(k)}(\xi) \tag{6.3}$$

$$\xi \in [x_0, x_1], \quad h = x_1 - x_0.$$

Arrestandoci, nella (6.3), alla derivata prima, tenuto conto della (6.2), si ha

$$y(x_1) \simeq y(x_0) + hy'(x_0) = y(x_0) + hf(x_0, y_0) \tag{6.4}$$

che, dopo n passi, fornisce

$$y_{n+1} = y_n + hf(x_n, y_n). \tag{6.5}$$

La (6.5) prende il nome di *metodo di Eulero;* in questo caso $\phi(x_n, y_n; h) \equiv f(x_n, y_n)$.

Dalle (6.3) e (6.5) con $n = 0$, è immediato ottenere l'*errore di troncamento* che si commette ad ogni passo, precisamente si ha

$$E_e = y(x_1) - y_1 = \frac{h^2}{2} y''(\xi).$$

Osservazione 6.1.1 *Il metodo di Eulero gode di una chiara interpretazione geometrica (cfr. la figura 6.1): il valore assunto dalla soluzione nel punto x_1 ($y(x_1)$) è approssimato con il valore (y_1) assunto, nello stesso punto, dalla retta tangente la linea integrale $y(x)$ nel punto x_0.*

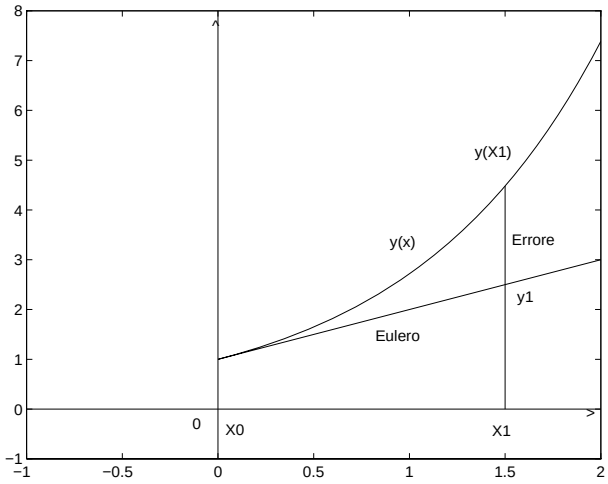


Figura 6.1:

Il metodo di Eulero è esatto (errore di troncamento nullo) se la soluzione $y(x)$ del problema (6.2) è una retta; diremo quindi che il metodo di Eulero è un *metodo del primo ordine.*

Volendo ottenere un metodo d'ordine superiore (maggiore accuratezza) sarà necessario introdurre un altro termine dello sviluppo (6.3) contenente $y''(x_0)$. Dal problema (6.2) è facile ricavare tale termine osservando che

$$\begin{aligned} y''(x) &= \frac{d}{dx}\left(y'(x)\right) = \frac{d}{dx}\left(f(x, y(x))\right) = \\ &= f_x(x, y(x)) + f_y(x, y(x)) y'(x) = \\ &= f_x(x, y(x)) + f_y(x, y(x)) f(x, y(x)). \end{aligned}$$

Si ottiene così un *metodo del secondo ordine* ponendo

$$y_{n+1} = y_n + hf(x_n, y_n) + \frac{h^2}{2}\left[f_x(x_n, y_n) + f_y(x_n, y_n)f(x_n, y_n)\right]$$

che comporta, verificarlo per esercizio, un errore di troncamento

$$E_2 = y(x_1) - y_1 = \frac{h^3}{6}y'''(\xi).$$

In generale un metodo di Taylor di ordine k si ottiene ponendo

$$\phi(x_n, y_n; h) = T_k(x_n, y_n; h)$$

dove

$$T_k(x_n, y_n; h) = f(x_n, y_n) + \frac{h}{2}y''(x_n) + ... + \frac{h^{k-1}}{k!}y^{(k)}(x_n)$$

e le derivate $y^{(j)}(x)$ $(j = 2, 3, ..., k)$ sono calcolate derivando successivamente $y'(x) = f(x, y)$. Il metodo diviene quindi

$$y_{n+1} = y_n + hT_k(x_n, y_n; h)$$

con errore di troncamento

$$E_k = y(x_1) - y_1 = \frac{h^{k+1}}{(k+1)!}y^{(k+1)}(\xi).$$

Osservazione 6.1.2 *Nei metodi di Taylor (Eulero escluso) si ha l'inconveniente che devono essere calcolate preventivamente le derivate $y^{(j)}(x)$ derivando "simbolicamente" l'equazione $y'(x) = f(x, y)$. Questo può portare, per k elevati, ad espressioni di $T_k(x_n, y_n; h)$ molto complesse anche quando $f(x, y)$ è semplice, rendendo il metodo poco efficiente benchè accurato.*

Prima di procedere nella determinazione di altri metodi è opportuno soffermarci sulla seguente domanda: *cosa vuole dire risolvere un problema di Cauchy?* Se la variabile indipendente x rappresenta il tempo possiamo dire che risolvere un problema di Cauchy vuole dire, essere in grado di prevedere l'evolvere nel tempo di un fenomeno (rappresentato dalla $y(x)$) che all'istante $a = x_0$ si trovava nella configurazione y_0. Sarà quindi cruciale riuscire ad ottenere, con il metodo numerico scelto, una buona stima della soluzione del problema, non solo vicino, ma anche "lontano" dalla condizione iniziale.

Nella determinazione dei precedenti metodi abbiamo supposto d'eseguire un solo passo d'integrazione (dalla condizione iniziale in x_0 al punto $x_1 = x_0 + h$). Nella pratica essendo interessati a conoscere la soluzione in tutto un intervallo $[a, b]$ (dalla condizione iniziale in $a = x_0$ al punto $b = x_n = x_0 + nh$) dovremo eseguire, fissato h, n passi d'integrazione. E' chiaro che, essendo $[a, b]$ fissato, se si riduce l'ampiezza del passo h si dovrà aumentare il numero n di passi in modo che $nh = b - a$. Questa banale constatazione è cruciale per far soffermare il lettore sul fatto che, contrariamente a quanto avviene per i metodi iterativi per il calcolo degli zeri (metodi di punto fisso), l'indice d'iterazione nei metodi per equazioni differenziali fa spostare da un "istante" (n) al "successivo" (n+1) e non "convergere" verso un punto fisso.

Per quanto sopra detto introduciamo le seguenti definizioni

Definizione 6.1.1 ***Errore di troncamento globale****: è la differenza fra la soluzione $y(x_{n+1})$ (soluzione "vera" del problema nel punto x_{n+1}) e y_{n+1} (soluzione "calcolata" dopo $n+1$ passi). Con ovvio significato dei simboli abbiamo*

$$e_{n+1} = y(x_{n+1}) - y_{n+1}.$$

Definizione 6.1.2 ***Errore di troncamento locale****: è la differenza fra la soluzione $y(x_{n+1})$ (soluzione "vera" del problema nel punto x_{n+1}) e u_{n+1} (soluzione "calcolata" partendo dalla condizione $y(x_n)$ eseguendo un solo passo d'integrazione). Per cui*

$$\tau_{n+1} = y(x_{n+1}) - u_{n+1} \tag{6.6}$$

dove

$$u_{n+1} = y(x_n) + h\phi(x_n, y(x_n); h)$$

è la soluzione del problema di Cauchy

$$\begin{cases} u' = f(x, u) \\ u(x_n) = y(x_n) \end{cases}$$

calcolata numericamente nel punto $x_n + h$.

Definizione 6.1.3 ***Errore di troncamento locale unitario****: è l'errore τ^h_{n+1} di troncamento locale rapportato al passo d'integrazione h:*

$$\tau^h_{n+1} = \frac{\tau_{n+1}}{h}.$$

Definizione 6.1.4 ***Ordine****: un metodo di integrazione numerica si dice di ordine p se l'errore di troncamento locale unitario va a zero come* h^p

$$\tau_{n+1}^h = O(h^p).$$

Definizione 6.1.5 ***Consistente****: un metodo di integrazione numerica è consistente se è del primo ordine o, equivalentemente, se*

$$\lim_{h\to 0} T_k(x_n, y_n; h) = f(x_n, y_n). \tag{6.7}$$

Osservazione 6.1.3 *E' facile verificare l'equivalenza fra la (6.7) e il fatto che*

$$\tau_{n+1}^h = O(h)$$

dalla (6.6) si ha

$$\tau_{n+1}^h = \frac{y(x_{n+1}) - y(x_n)}{h} - \phi(x_n, y(x_n); h) \tag{6.8}$$

e passando al limite nella (6.8), essendo

$$\lim_{h\to 0} \tau_{n+1}^h = 0$$

si ha

$$\lim_{h\to 0} \frac{y(x_{n+1}) - y(x_n)}{h} = y'(x_n) = f(x_n, y_n) = \phi(x_n, y(x_n); 0).\square$$

Definizione 6.1.6 ***Convergenza****: un metodo di integrazione numerica è convergente se*

$$\lim_{\substack{h\to 0 \\ nh=x}} y_n = y(x)$$

(riducendo il passo h, tenendo fisso il punto d'arrivo x, la soluzione numerica stima sempre meglio la soluzione analitica).

Una chiara interpretazione delle precedenti definizioni è data in figura (6.2)

Per quanto detto la sola consistenza non ci basta per garantire la bontà dei risultati ottenuti con il metodo numerico ma è necessario garantire la convergenza. D'altro canto mentre è, in generale, relativamente semplice verificare la consistenza (basta considerare un passo d'integrazione) assai più complesso è verificare la convergenza ($n \to \infty$ passi d'integrazione!!).

Dimostriamo il seguente

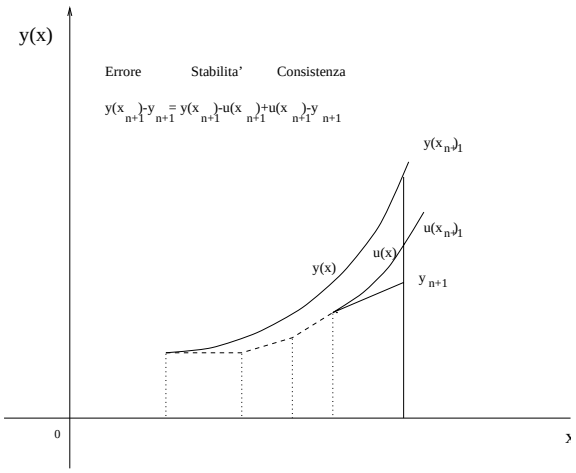


Figura 6.2:

Teorema 6.1.1 *Sotto le ipotesi del teorema d'esistenza ed unicità il metodo di Eulero è convergente.*

Dimostrazione 6.1.1 *Dalla (6.5) si ha*

$$y_{n+1} = y_n + hf(x_n, y_n)$$

dallo sviluppo in serie di Taylor segue

$$y(x_{n+1}) = y(x_n) + hf(x_n, y(x_n)) + \frac{h^2}{2}y''(\xi).$$

Per l'errore al passo $(n+1)$*-esimo si avrà*

$$e_{n+1} = y(x_{n+1}) - y_{n+1} = e_n + h\left[f(x_n, y(x_n)) - f(x_n, y_n)\right] + \frac{h^2}{2}y''(\xi)$$

per le condizioni di esistenza ed unicità

$$|f(x_n, y(x_n)) - f(x_n, y_n)| \leq L\,|y(x_n) - y_n|$$

per cui

$$e_{n+1} \leq e_n + hL\,|y(x_n) - y_n| + \frac{h^2}{2}y''(\xi)$$

ovvero

$$|e_{n+1}| \leq (1 + hL)\,|e_n| + \frac{h^2}{2}\,|y''(\xi)|\,.$$

Posto $n = 0, 1, ..., (n+1)$, *si ha*

$$\begin{aligned} |e_1| &\leq (1+hL)\,|e_0| + \frac{h^2}{2}\,|y''(\xi)| \qquad (6.9)\\ |e_2| &\leq (1+hL)^2\,|e_0| + (1+hL)\frac{h^2}{2}\,|y''(\xi)| + \frac{h^2}{2}\,|y''(\xi)| \\ &\cdots \\ |e_{n+1}| &\leq (1+hL)^{n+1}\,|e_0| + \sum_{j=0}^{n}(1+hL)^j\frac{h^2}{2}\,|y''(\xi)|\,. \end{aligned}$$

Supposto $e_0 = 0$ *(errore su condizione iniziale nullo), moltiplicando e dividendo l'ultima delle (6.9) per* $[(1+hL)-1]$, *si ottiene*

$$|e_{n+1}| \leq \frac{(1+hL)^{n+1}-1}{hL}\frac{h^2}{2}\,|y''(\xi)|$$

ricordando che $e^{hL} \geq 1+hL$ *(si ricordi lo sviluppo di Taylor ...) si ha*

$$|e_{n+1}| \leq \frac{e^{(n+1)hL}-1}{hL}\frac{h^2}{2}\,|y''(\xi)| \leq \frac{e^{(n+1)hL}-1}{2L}\,|y''(\xi)|\,h = Ch \qquad (6.10)$$

per cui l'errore globale va a zero come il passo d'integrazione h (convergenza).■

Il risultato appena dimostrato per il metodo di Eulero può essere generalizzato a tutti i metodi ad un passo in virtù del seguente

Teorema 6.1.2 *Dato un metodo numerico di tipo Taylor (ad un passo) se la* $T_k(x, y; h)$ *è "lipschitziana" ed il metodo è consistente allora il metodo è convergente.*■

Osservazione 6.1.4 *La consistenza, se la* $T_k(x, y; h)$ *è "lipschitziana", è quindi condizione necessaria e sufficiente per la convergenza.*

Osservazione 6.1.5 *Essendo, nel metodo di Eulero ed in tutti i metodi ad un passo (di tipo Taylor)*

$$\lim_{h\to 0} T_k(x, y; h) = f(x, y)$$

la "lipschitzianeità" è garantita dal teorema di esistenza ed unicità, per cui si ha garantita, in virtù del teorema precedente, la convergenza.

Osservazione 6.1.6 *Nella (6.10) l'errore va a zero come h, si è persa una potenza di h rispetto all'errore locale di discretizzazione. Questo fatto non deve stupire in quanto, esattamente come per le formule di quadratura composte, l'aver diviso l'intervallo d'integrazione in n parti ($nh = x_n - x_0$), ci obbliga ad eseguire n passi d'integrazione. Questa osservazione giustifica anche l'introduzione della definizione di errore di troncamento locale unitario (la divisione per h ci fa perdere una potenza nell'ordine di convergenza).*

Nella deduzione della (6.10) abbiamo trascurato gli errori d'arrotondamento che si commettono ad ogni singolo passo. Volendo tener conto anche di questi si avrebbe, detto η l'errore d'arrotondamento,

$$\begin{aligned} |e_{n+1}| &\leq \frac{e^{(n+1)hL} - 1}{hL} \left(\frac{h^2}{2} |y''(\xi)| + \eta \right) \\ &\leq \left(\frac{h}{2L} |y''(\xi)| + \frac{\eta}{hL} \right) \left(e^{(n+1)hL} - 1 \right) \end{aligned} \tag{6.11}$$

per cui, mentre l'errore di discretizzazione va a zero con h, l'errore d'arrotondamento cresce come h^{-1}. Nasce quindi, come nel caso della derivazione numerica, la necessità di scegliere, fissata la precisione di macchina η, un h ottimo. Derivando rispetto ad h il secondo membro della (6.11), ignorando le costanti, ed uguagliando a zero, si ha

$$\frac{d}{dh} \left(\frac{h}{2} |y''(\xi)| + \frac{\eta}{h} \right) = \frac{1}{2} |y''(\xi)| + \frac{\eta}{h^2} = 0$$

per cui, indicato con $y''_M = \max_{\xi} |y''(\xi)|$, si ha

$$h_{ott} \simeq \sqrt{\frac{2\eta}{y''_M}} = c\eta^{\frac{1}{2}}. \tag{6.12}$$

La (6.12) ci fornisce un legame fra h ed η che può essere così parafrasato: più è piccolo η (maggiore è l'accuratezza nei calcoli) più si può prendere h piccolo (indipendentemente dalla costante c), purtroppo l'esponente $\frac{1}{2}$ controlla male la riduzione del passo in quanto, per avere h dell'ordine del 10^{-3}, è necessario prendere η dell'ordine del 10^{-6}.

6.1.2 Metodi Runge-Kutta

I metodi di Rung-Kutta di ordine p sono metodi ad un passo in cui $\phi(x, y; h)$ viene determinata in modo d'ottenere, senza effettuare il calcolo delle derivate d'ordine superiore della $y(x)$, un ordine d'accuratezza p. Per fare questo si usa una combinazione di r (metodi di Runge-Kutta a r stadi) derivate prime di $y(x)$ in opportuni punti dell'intervallo $[x, x + h]$.

Osservazione 6.1.7 *Il metodo d'Eulero è un metodo di Runge-Kutta del primo ordine ad uno stadio (si usa solo la* $y'(x_n) = f(x_n, y_n)$*).*

Vediamo come è possibile costruire un metodo del *secondo ordine* a *due stadi*. Scriviamo $\phi(x_n, y_n; h)$ nella forma

$$\phi(x_n, y_n; h) = a_1 K_1 + a_2 K_2$$

dove a_1 e a_2 sono opportuni coefficienti e

$$\begin{aligned} K_1 &= f(x_n, y_n) \\ K_2 &= f(x_n + \mu h, y_n + \mu h K_1) \end{aligned}$$

con $0 \leq \mu \leq 1$, sono approssimazioni dei valori di $y'(x)$ nei punti x_n e $x_n + \mu h$ ottenute mediante $f(x_n, y_n)$ e $f(x_n + \mu h, y_n + \mu h K_1)$ rispettivamente. Il metodo diviene quindi

$$y_{n+1} = y_n + h\left[a_1 f(x_n, y_n) + a_2 f(x_n + \mu h, y_n + \mu h K_1)\right].$$

Una chiara interpretazione geometrica è fornita in figura (6.3).

Per determinare i parametri a_1, a_2 e μ imponiamo la massima accuratezza possibile per il metodo, il che equivale ad eguagliare, nello sviluppo in serie di Taylor di $y(x_n)$ e di $\phi(x_n, y_n; h)$ in un intorno di x_n, più termini possibili. Arrestando lo sviluppo ai termini del secondo ordine si ha, per la $\phi(x_n, y_n; h)$

$$\begin{aligned} \phi(x_n, y_n; h) &= a_1 f(x_n, y_n) + a_2 f(x_n + \mu h, y_n + \mu h K_1) \\ &= a_1 f(x_n, y_n) + a_2[f(x_n, y_n) + \mu h f_x(x_n, y_n) + \\ &\quad + \mu h f(x_n, y_n) f_y(x_n, y_n)] + O(h^2) \\ &= (a_1 + a_2) f(x_n, y_n) + \mu h a_2[f_x(x_n, y_n) + \\ &\quad + f(x_n, y_n) f_y(x_n, y_n)] + O(h^2), \end{aligned} \tag{6.13}$$

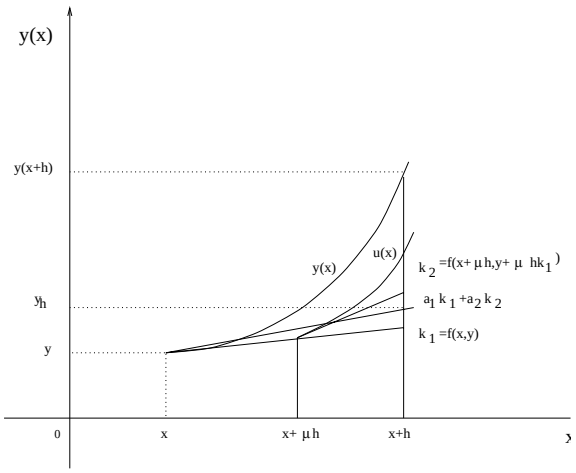


Figura 6.3:

mentre per $y(x_n)$ si ha

$$\frac{y(x_n+h)-y(x_n)}{h} = f(x_n,y_n) + \frac{h}{2}\left[f_x(x_n,y_n) + f(x_n,y_n)f_y(x_n,y_n)\right] + O(h^2). \tag{6.14}$$

Eguagliando i coefficienti dei termini corrispondenti nei due sviluppi si ottiene il seguente sistema non lineare

$$\begin{cases} a_1 + a_2 &= 1 \\ \mu a_2 &= \frac{1}{2} \end{cases} \tag{6.15}$$

che, fra le infinite soluzioni, ammette anche

1. *metodo di Eulero modificato*

$$\begin{cases} a_1 = 0 \\ a_2 = 1 \\ \mu = \frac{1}{2} \end{cases}$$

che è della forma

$$y_{n+1} = y_n + hf\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}f(x_n,y_n)\right);$$

2. *metodo di Heun*

$$\begin{cases} a_1 = \frac{1}{2} \\ a_2 = \frac{1}{2} \\ \mu = 1 \end{cases}$$

che è della forma

$$y_{n+1} = y_n + \frac{h}{2}\left[f(x_n, y_n) + f\left(x_n + h, y_n + hf(x_n, y_n)\right)\right].$$

Dalle (6.13) e (6.14) è immediato osservare che i metodi risultano del secondo ordine essendo l'*errore di troncamento locale unitario* $O(h^2)$.

In figura (6.4) è riportata l'interpretazione geometrica del metodo di Heun.

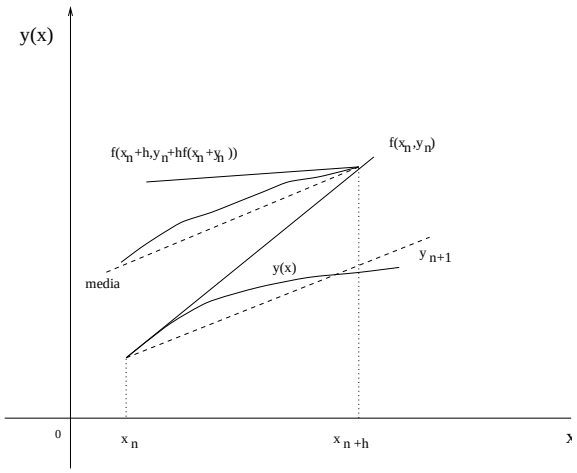


Figura 6.4:

Osservazione 6.1.8 *Nella (6.15) la condizione* $a_1 + a_2 = 1$ *implica la consistenza del metodo, per cui, sotto le consuete ipotesi di esistenza ed unicità, tutti i metodi ricavati dalla (6.15) risultano convergenti.*

Quanto visto per due stadi può essere esteso (non con facili conti!) al caso generale di r stadi, per cui otteniamo

$$\phi(x_n, y_n; h) = \sum_{i=1}^{r} a_i K_i(x, y; h)$$

dove

$$\begin{aligned} K_1 &= f(x_n, y_n) \\ K_i &= f(x_n + \mu_i h, y_n + h \sum_{j=1}^{i-1} \lambda_{ij} K_j); \quad i = 2, 3, ..., r \end{aligned} \tag{6.16}$$

che prendono il nome di metodi Runge-Kutta a r stadi *espliciti*, dove l'aggettivo esplicito evidenzia il fatto che ogni K_i *dipende solo* dai K_j precedenti. Nella (6.16) i parametri a_i, μ_i e λ_{ij} verranno scelti in modo d'ottenere un metodo il più accurato possibile.

Per i metodi espliciti, detto p l'ordine del metodo ed r il numero di stadi, vale la seguente regola: "al crescere del numero degli stadi non si guadagna sempre parimenti in accuratezza", si ha infatti che

- esistono metodi con $p = r$ solo per $r = 1, 2, 3, 4$
- per $r = 5, 6, 7$ si hanno solo metodi di ordine $p = r - 1$
- per $r = 8, 9$ si hanno solo metodi di ordine $p = r - 2$
- per $r \geq 10$ l'ordine è $p < r - 2$.

Per concludere ricordiamo il classico metodo di Runge-Kutta del quarto ordine a quattro stadi

$$\phi(x_n, y_n; h) = \frac{1}{6} [K_1 + 2K_2 + 2K_3 + K_4]$$

dove

$$\begin{aligned} K_1 &= f(x_n, y_n) \\ K_2 &= f(x_n + \frac{h}{2}, y_n + \frac{h}{2} K_1) \\ K_3 &= f(x_n + \frac{h}{2}, y_n + \frac{h}{2} K_2) \\ K_4 &= f(x_n + h, y_n + h K_3). \end{aligned}$$

Se nella (6.16) l'indice j della sommatoria può andare fino ad r (ogni K_i dipende da tutti gli altri) si hanno i metodi di *Runge-Kutta impliciti*, se l'indice j della sommatoria può andare fino ad i (ogni K_i dipende dai precedenti K_j e da se stesso) si hanno i metodi di *Runge-Kutta semi-impliciti*, per i quali si rimanda alla letteratura specifica.

6.1.3 Sistemi del primo ordine

Quanto visto per un'equazione del primo ordine può essere facilmente esteso sia a sistemi del primo ordine che ad equazioni di grado superiore al primo, come di seguito illustrato per un'equazione del secondo ordine.

Dato il seguente problema del secondo ordine

$$\begin{cases} y'' = f(x, y, y') \\ y(a) = \alpha \\ y'(a) = \beta \end{cases}$$

abbiamo visto che è possibile trasformarlo nel sistema equivalente

$$\begin{cases} y' = z \\ z' = f(x, y, z) \end{cases} \quad con \quad \begin{cases} y(a) = \alpha \\ z(a) = \beta \end{cases}$$

ed i metodi precedenti possono essere generalizzati al nuovo sistema utilizzando, con ovvio significato dei simboli, la seguente notazione vettoriale

$$\underline{u}' = \underline{f}(x, \underline{u})$$

ove

$$\underline{u} = \begin{bmatrix} y \\ z \end{bmatrix}, \quad \underline{u}' = \begin{bmatrix} y' \\ z' \end{bmatrix}, \quad \underline{f} = \begin{bmatrix} z \\ f(x, y, z) \end{bmatrix}$$

con le condizioni

$$\underline{u}(a) = \underline{\gamma}, \quad ove \quad \underline{\gamma} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}.$$

Per cui, dato il metodo numerico

$$y_{n+1} = y_n + h\phi(x_n, y_n; h)$$

si potrà scrivere, per il sistema precedente

$$\underline{u}_{n+1} = \underline{u}_n + h\underline{\phi}(x_n, \underline{u}_n; h).$$

Esemplifichiamo quanto sopra nel caso del metodo di Heun

$$\begin{aligned} y_{n+1} &= y_n + \frac{h}{2}\left[f(x_n y_n) + f\left(x_{n+1}, y_n + hf(x_n, y_n)\right)\right] = \\ &= y_n + \frac{h}{2}[K_1 + K_2] \end{aligned}$$

ed all'equazione

$$y''' = x^2 y'' + y^2 y';$$

posto

$$\begin{cases} y = u_1 \\ y' = u_2 \\ y'' = u_3 \end{cases}$$

si ha, in forma vettoriale:

$$\begin{cases} u_1' = u_2 \\ u_2' = u_3 \\ u_3' = x^2 u_3 + u_1^2 u_2 \end{cases} .$$

Un semplice programma in Matlab, per risolvere il problema precedente, può essere il seguente

```
function uprime=uprim(x,u)
uprime=[u(2);u(3);u(3)*x^2+u(2)*u(1)^2];
return
end

.....% definizioni iniziali
h=...
x=...
uold=..
while x < xfin
  k1=uprim(x,uold);
  x=x+h;
  utemp=uold+h*k1;
  k2=uprim(x,utemp);
  unew=uold+h*(k1+k2)/2;
  uold=unew;
...%continua fino a che x < xfin
end;...%fine while
```

Come si può osservare dall'attenta lettura del listato del programma, per trattare, con il metodo di Heun, altri sistemi differenziali basta modificare opportunamente solo la `function uprim`.

6.1.4 Stabilità numerica

Per i metodi precedenti abbiamo garantito un certo *ordine di accuratezza* e la *convergenza* per $h \to 0$. Quando si deve integrare numericamente, in un intervallo prefissato $[a, b]$, un'equazione differenziale, il passo h deve essere, ovviamente, preso diverso da zero. Nasce spontanea la domanda: *per ogni passo h la soluzione numerica sarà una buona stima di quella reale ?* O equivalentemente: *gli errori introdotti in ogni singolo passo d'integrazione non fanno "esplodere" la soluzione numerica* ? Questi interrogativi permettono di introdurre il concetto di *stabilità numerica.* Diamo le seguenti

Definizione 6.1.7 *La soluzione di un problema di Cauchy è detta inerentemente stabile se*

$$|y(x_{n+1})| \leq C\,|y(x_n)|\,; \quad C < 1, \quad \forall n.$$

Definizione 6.1.8 *Un metodo numerico si dice assolutamente stabile se*

$$|y_{n+1}| \leq c\,|y_n|\,; \quad c < 1, \quad \forall n,$$

(dove y_n è la soluzione numerica nel punto x_n).

Per studiare la stabilità numerica considereremo (per semplicità d'esposizione) l'*equazione test*

$$\begin{cases} y' = \lambda y \\ y(0) = y_0 \end{cases} \quad ; \quad \boxed{\mathrm{Re}(\lambda)<0} \tag{6.17}$$

che ammette come soluzione

$$y(x) = y_0 e^{\lambda x} \tag{6.18}$$

che, essendo $\mathrm{Re}(\lambda) < 0$, è inerentemente stabile.

Se consideriamo il metodo di Eulero, otteniamo la soluzione

$$y_{n+1} = y_n + h\lambda y_n = (1 + h\lambda)\, y_n$$

e quindi

$$y_{n+1} = (1 + h\lambda)^{n+1}\, y_0. \tag{6.19}$$

La soluzione (6.19) si comporta, per $n \to \infty$, come la soluzione (6.18) se e solo se

$$|1 + h\lambda| < 1. \tag{6.20}$$

La (6.20) rappresenta la *regione di stabilità assoluta* del metodo di Eulero, se non stiamo all'interno di questa regione (cerchio del piano complesso con centro in $(-1, 0)$ e raggio unitario) la soluzione numerica crescerà in modulo allontanandosi sempre più dalla soluzione vera $y(x)$ (che tende a zero per $n \to \infty$).

Analogamente si potrebbe calcolare la regione di stabilità assoluta per i metodi di Heun e di Runge-Kutta del quarto ordine, per i quali si rimanda alla bibliografia specifica.

Per tutti i metodi ad un passo visti esiste quindi una condizione sul passo d'integrazione h che, se violata, produce una soluzione inattendibile. Questo fatto si traduce dicendo che i metodi Runge-Kutta e di Taylor sono *condizionatamente stabili*.

6.2 Metodi a più passi (M.F.)

I metodi a p passi sono metodi che approssimano la soluzione in un punto utilizzando i valori trovati in p passi precedenti. Si possono scrivere nella forma

$$\begin{aligned} y_{n+p} &= -\sum_{j=0}^{p-1} \alpha_j y_{n+j} + h \sum_{j=0}^{p} \beta_j y'_{n+j} = \\ &= -\sum_{j=0}^{p-1} \alpha_j y_{n+j} + h \sum_{j=0}^{p} \beta_j f(x_{n+j}, y_{n+j}) \end{aligned} \tag{6.21}$$

dove i $2p + 1$ parametri α_j e β_j individuano il singolo metodo. Se $\beta_p = 0$ il metodo si dirà *esplicito*, mentre se $\beta_p \neq 0$ il metodo sarà detto *implicito*. Per determinare i parametri α_j e β_j si possono seguire differenti approcci

1. *utilizzando le formule di derivazione numerica*; l'equazione differenziale fornisce un legame fra la funzione $y(x)$ e la sua derivata prima;

2. *utilizzando le formule di quadratura*; il problema (6.1) è equivalente al problema

$$\int_{x_0}^{x_0+h} y'(x)dx = \int_{x_0}^{x_0+h} f(x, y)dx$$

ovvero

$$y(x_0 + h) = y(x_0) + \int_{x_0}^{x_0+h} f(x, y)dx, \tag{6.22}$$

in cui compare il calcolo di un integrale;

3. *utilizzando il concetto di ordine d'accuratezza*; imponendo che la formula sia esatta ogniqualvolta la soluzione della (6.1) è un polinomio di grado al più $2p$.

Esempio 6.2.1 *dalle formula di derivazione: consideriamo la formula*

$$y'(x) = \frac{y(x+h) - y(x-h)}{2h} + O(h^2) \tag{6.23}$$

si ottiene il metodo

$$y_{n+2} = y_n + 2hy'_{n+1} \tag{6.24}$$

per cui $\alpha_0 = -1, \alpha_1 = 0, \beta_0 = 0, \beta_1 = 2, \beta_2 = 0$ *che è un LMM (Linear Multy step Method) a due passi. Il passaggio dalla (6.23) alla (6.24) si giustifica facilmente considerando, dovendo integrare la (6.1) nell'intervallo* $[x-h, x+h]$*, il seguente schema:*

$$\overset{y_n}{\underset{y(x-k)}{+}} \quad - - - - \quad \overset{y_{n+1}}{\underset{y(x)}{+}} \quad - - - - \quad \overset{y_{n+2}}{\underset{y(x+k)}{+}} \tag{6.25}$$

Esempio 6.2.2 *dalle formule di quadratura: consideriamo la formula del punto medio*

$$\int_{x_0-h}^{x_0+h} f(x)dx = 2hf(x_0) + \frac{h^3}{3}f''(\xi)$$

applichiamola, tenuto conto del cambio d'intervallo e dello schema (6.25), alla (6.22) ed otteniamo ancora

$$y_{n+2} = y_n + 2hy'_{n+1}$$

che è lo stesso metodo ottenuto nell'esempio precedente.

Esempio 6.2.3 *imponendo l'ordine d'accuratezza: imponiamo che il metodo numerico a 2 passi*

$$y_{n+2} = -\left(\alpha_0 y_n + \alpha_1 y_{n+1}\right) + h\left(\beta_0 y'_n + \beta_1 y'_{n+1} + \beta_2 y'_{n+2}\right) \tag{6.26}$$

sia esatto (errore di troncamento nullo) quando la soluzione è un polinomio di grado minore od uguale a 2. Ricordando che i monomi 1*,* x *ed* x^2 *verificano rispettivamente i problemi di Cauchy*

$$\begin{cases} y' = 0 \\ y(-h) = 1 \end{cases} ; \quad \begin{cases} y' = 1 \\ y(-h) = -h \end{cases} ; \quad \begin{cases} y' = 2x \\ y(-h) = h^2 \end{cases}$$

le soluzioni ottenute con il metodo (6.26) nell'intervallo $[-h, h]$ *(si confronti lo schema seguente)*

$$\overset{-h}{\underset{n}{+}} \; - - - - \; \overset{0}{\underset{n+1}{+}} \; - - - - \; \overset{h}{\underset{n+2}{+}}$$

portano al seguente sistema lineare

$$\begin{cases} 1 = & -(\alpha_0 + \alpha_1) + h \cdot 0 \\ h = & -(\alpha_0(-h) + \alpha_1 \cdot 0) + h\,(\beta_0 + \beta_1 + \beta_2) \\ h^2 = & -(\alpha_0 h^2 + \alpha_1 \cdot 0) + h\,(-2h\beta_0 + \beta_1 \cdot 0 + 2h\beta_2) \end{cases} .$$

Imponendo al metodo di essere esplicito ($\beta_2 = 0$) *si ha*

$$\begin{cases} 1 = & -\alpha_0 - \alpha_1 \\ 1 = & \alpha_0 + \beta_0 + \beta_1 \\ 1 = & -\alpha_0 - 2\beta_0 \end{cases} ,$$

avendo ancora un grado di libertà (3 equazioni e 4 parametri) possiamo porre $\alpha_1 = 0$, *ottenendo così* $\alpha_0 = -1$, $\alpha_1 = 0$, $\beta_0 = 0$, $\beta_1 = 2$, $\beta_2 = 0$ *che è ancora il metodo (6.24).*

Dall'ultimo esempio considerato si ottiene che il *metodo del punto medio*

$$y_{n+2} = y_n + 2hy'_{n+1}$$

è un metodo *LMM a 2 passi esplicito* e del *secondo ordine.*

Osservazione 6.2.1 *Che l'errore di troncamento locale unitario sia* $O(h^2)$ *lo si evince dalla (6.23), che poi il metodo non sia esatto per polinomi di grado 3 è facilmente verificabile sul problema di Cauchy*

$$\begin{cases} y' = 3x^2 \\ y(-h) = -h^3 \end{cases} .$$

I metodi della forma (6.21) devono appoggiarsi, nei primi $p-1$ passi, ad un metodo di integrazione ad un passo (Taylor o Runge-Kutta) che fornisca le prime $p-1$ valutazioni necessarie ad innescare il metodo. Se il metodo è esplicito procederà poi automaticamente nelle iterazioni sucessive mentre, se è implicito, bisognerà risolvere ad ogni passo un'equazione implicita. A fronte di questi due inconvegnenti si ha il vantaggio che ad ogni singolo passo si ha

un metodo di ordine p ($p+1$ se implicito) con una sola nuova valutazione della $f(x_n, y_n)$.

L'equazione implicita da risolvere ad ogni passo si presenta nella forma (cfr. (6.21))

$$y_{n+p} = h\beta_p f(x_{n+p}, y_{n+p}) + g_n \tag{6.27}$$

dove

$$g_n = -\sum_{j=0}^{p-1} \alpha_j y_{n+j} + h\sum_{j=0}^{p-1} \beta_j f(x_{n+j}, y_{n+j})$$

non dipende da y_{n+p}. Per risolvere la (6.27) si può utilizzare un metodo di punto fisso nella forma

$$y_{n+p}^{(i+1)} = h\beta_p f(x_{n+p}, y_{n+p}^{(i)}) + g_n; \quad i = 0, 1, ... \tag{6.28}$$

partendo da un punto iniziale $y_{n+p}^{(0)}$ (per esempio $y_{n+p}^{(0)} \equiv y_{n+p-1}$). Ricordando la condizione di convergenza per i metodi del primo ordine (cfr. teorema 3.2.3) si ha il seguente

Teorema 6.2.1 *Se $f(x,y)$ è lipschitziana*

$$\|f(x, y^*) - f(x, y)\| \leq L \|y^* - y\|; \quad \forall a \leq x \leq b,\ y, y^* \in R$$

ed inoltre

$$h\left|\beta_p\right| L < 1 \tag{6.29}$$

allora la successione (6.28) converge.■

Osservazione 6.2.2 *La condizione (6.29) afferma che, pur di prendere h sufficientemente piccolo, l'equazione implicita (6.27) può essere risolta con il metodo di punto fisso (6.28).*

Come per i metodi ad un passo possiamo dare le definizioni di consistenza ed ordine per i LMM;

Definizione 6.2.1 *Un LMM della forma (6.21) è consistente se l'errrore di troncamento locale unitario va a zero con h, ovvero:*

$$\lim_{h\to 0} \tau_n^h = \lim_{h\to 0} \left[\frac{y(x_n + h) + \sum_{j=0}^{p-1} \alpha_j y(x_n + jh)}{h} - \sum_{j=0}^{p} \beta_j y'(x_n + jh) \right] = 0.$$

Definizione 6.2.2 *Un LMM della forma (6.21) è di ordine r se*

$$\tau_n^h = O(h^r).$$

Osservazione 6.2.3 *E' interessante osservare che mentre un LMM di ordine r integra esattamente (a meno degli errori di arrotondamento) un qualunque problema di Cauchy la cui soluzione è un polinomio di grado minore od ugual ad r, questo non è più vero, in generale, se si usa un metodo di Runge-Kutta di ordine r. (Si veda l'esercizio 6.3.1).*

Per quanto riguarda la convergenza nei metodi LMM la consistenza (data la lischipzianità di $f(x,y)$) è solo condizione necessaria ma non più sufficiente. Infatti, mentre nei metodi ad un passo l'equazione differenziale del primo ordine viene sostituita con un' equazione alle differenze anch' essa del primo ordine, la cui *unica soluzione* è un' approssimazione della soluzione del problema differenziale, nei LMM l'equazione alle differenze, di ordine $p > 1$, può ammettere fino a p *soluzioni distinte*, di cui una sola (*soluzione principale*) approssima correttamente la soluzione del problema differenziale, mentre le rimanenti (*soluzioni parassite*) introducono solo dell'errore. E' chiaro quindi che, per avere convergenza, al tendere di h a zero, le soluzioni parassite dovranno essere ininfluenti.

Ricordando che data l'equazione alle differenze (a coefficienti costanti)

$$y_{n+p} = -\sum_{j=0}^{p-1} \alpha_j y_{n+j} \tag{6.30}$$

(ottenuta, guarda caso, dalla (6.21) per $h = 0$) ammette la soluzione generale

$$y_m = \sum_{j=0}^{p-1} \gamma_j (r_j)^m$$

dove le r_j sono le p radici distinte del polinomio di grado p, associato alla (6.30) (*polinomio caratteristico*):

$$\lambda^p + \sum_{j=0}^{p-1} \alpha_j \lambda^j = 0$$

e le γ_j sono p costanti arbitrarie (determinabili una volta dati i primi p valori $y_0, y_1, ..., y_{p-1}$), possiamo dare la seguente

Definizione 6.2.3 *Un LMM della forma (6.21) è zero stabile se*

$$|r_j| \leq 1; \ \forall j \tag{6.31}$$

e se $|r_j| = 1$ allora r_j è semplice. La (6.31) è detta anche root condition (o condizione delle radici).

Si può dimostrare il seguente

Teorema 6.2.2 *Un metodo LMM è convergente se e solo se è consistente e zero stabile.*■

Dipendendo le radici del polinomio caratteristico dai coefficienti α_j questi potranno essere scelti in modo opportuno, non solo per garantire la consistenza e l'ordine ma anche la zero stabilità.

6.2.1 Metodi di Adams

I metodi di Adams-Bashforth e Adams-Moulton sono della forma

$$y_{n+p} = y_{n+p-1} + h\sum_{j=0}^{p-1} \beta_j y'_{n+j}; \quad (Adams - Bashforth)$$

$$y_{n+p} = y_{n+p-1} + h\sum_{j=0}^{p} \beta_j y'_{n+j}; \quad (Adams - Moulton)$$

dove i coefficienti β_j dipendono dalla particolare formula di quadratura utilizzata per approssimare, nell'identità

$$y_{n+p} = y_{n+p-1} + \int_{x_{n+p-1}}^{x_{n+p}} y'(x)dx$$

l'integrale a secondo membro. Se nella formula di quadratura si usa il nodo x_{n+p} (si è costruito il polinomio interpolante su tutti i $p+1$ nodi $x_n, x_{n+1}, ..., x_{n+p}$) si hanno i metodi di Adams-Moulton che sono impliciti, se invece nella formula di quadratura non si usa il nodo x_{n+p} (si è costruito il polinomio interpolante solo sui p nodi $x_n, x_{n+1}, ..., x_{n+p-1}$) si hanno i metodi di Adams-Bashforth che sono espliciti.

Per l'ordine di accuratezza si può dimostrare che, a parità di passi p, gli Adams-Bashforth hanno ordine p, mentre gli Adams-Moulton hanno ordine $p+1$.

Questi metodi fanno tutti parte della famiglia dei metodi LMM con la caratteristica di essere sicuramente zero stabili in quanto il loro polinomio caratteristico, essendo $\alpha_{p-1} = -1$ e $\alpha_j = 0$ per $j = 0, 1, ..., p-2$, è

$$\lambda^p - \lambda^{p-1} = 0$$

e quindi ha radici tutte nulle tranne una uguale ad 1 (root condition verificata).

Fra i metodi di Adams ricordiamo:

1. Adams-Bashforth

   (a) $y_{n+1} = y_n + hf(x_n, y_n)$; metodo di Eulero ($p = 1 \rightarrow$ ordine 1);

   (b) $y_{n+2} = y_{n+1} + \frac{h}{2}\left[3f(x_{n+1}, y_{n+1}) - f(x_n, y_n)\right]$; ($p = 2 \rightarrow$ ordine 2).

2. Adams-Moulton

   (a) $y_{n+1} = y_n + \frac{h}{2}\left[f(x_{n+1}, y_{n+1}) + f(x_n, y_n)\right]$; metodo dei Trapezi ($p = 1 \rightarrow$ ordine 2)

   (b) $y_{n+2} = y_{n+1} + \frac{h}{12}\left[5f(x_{n+2}, y_{n+2}) + 8f(x_{n+1}, y_{n+1}) - f(x_n, y_n)\right]$; ($p = 2 \rightarrow$ ordine 3)

6.2.2 Stabilità numerica

Per i metodi LMM oltre che di *stabilità assoluta* (come per i metodi ad un passo) si può parlare anche di *stabilità relativa*, in quanto, anche quando la soluzione è inerentemente instabile non è garantito che $\forall h$ la soluzione numerica, che dipende da tutte le radici del polinomio caratteristico, "segua" bene la soluzione analitica. Analizzeremo brevemente questi concetti su due classici metodi, il metodo dei trapezi ed il metodo del punto medio.

Stabilità assoluta

Come per i metodi ad un passo ci proponiamo di studiare, fissato il passo h d'integrazione, come si comporta la soluzione numerica calcolata con i LMM. Ci limiteremo, per motivi di tempo, ad analizzare la formula dei Trapezi (per una trattazione di carattere generale si rimanda ai testi in bibliografia).

Consideriamo la solita equazione test (6.17), il metodo dei Trapezi ad essa applicato produce

$$y_{n+1} = y_n + \frac{h}{2}\left[\lambda y_{n+1} + \lambda y_n\right]$$

che, risolta rispetto a y_{n+1}, fornisce

$$y_{n+1} = \frac{1 + \frac{h\lambda}{2}}{1 - \frac{h\lambda}{2}} y_n$$

ed, essendo $\mathrm{Re}(\lambda) < 0$, si ha

$$\left| \frac{1 + \frac{h\lambda}{2}}{1 - \frac{h\lambda}{2}} \right| < 1, \tag{6.32}$$

per cui la soluzione numerica è assolutamente stabile per ogni scelta del passo h.

Osservazione 6.2.4 *La regione di stabilità del metodo dei Trapezi è quindi tutto il semipiano negativo del piano complesso. Questa maggiore "libertà" nella scelta del passo h è stata pagata con la necessità di dover risolvere, ad ogni passo, un'equazione implicita. In generale i metodi impliciti sono "più stabili" degli espliciti.*

Nella (6.32) la quantità $\frac{1+\frac{h\lambda}{2}}{1-\frac{h\lambda}{2}}$ è l'unica radice del polinomio associato al metodo dei Trapezi applicato all'equazione test (6.17), in generale, dette σ_i le p radici (supposte distinte) del polinomio associato al generico LMM applicato all'equazione test (6.17), la condizione di *assoluta stabilità* si traduce in

$$|\sigma_i| < 1; \quad i = 1, 2, ..., p.$$

Stabilità relativa

Se nel problema test (6.17) abbiamo $\mathrm{Re}(\lambda) > 0$ la soluzione tenderà, per $n \to \infty$, ad infinito. Per i LMM si può quindi introdurre il concetto di *stabilità relativa*, nel senso che, è ragionevole richiedere che la soluzione numerica (che dovrà anch'essa crescere) si mantenga "vicino" alla soluzione vera (errore relativo limitato). Poichè la soluzione generale dell'equazione alle differenze dipende da tutte le p radici σ_i dovremo avere

$$|\sigma_i| < |\sigma_1|; \quad i = 2, 3, ..., p$$

dove con σ_1 si è indicata la *radice principale* e con σ_i, $i = 2, 3, ..., p$ le *radici parassite* (le radici parassite devono essere controllate dalla radice principale).

Vediamo di chiarire le ultime affermazioni fatte studiando la stabilità del metodo del *punto medio* (6.24). Dal metodo

$$y_{n+2} = y_n + 2hy'_{n+1}$$

e dal problema test (6.17) si ottiene l'equazione alle differenze

$$y_{n+2} - 2h\lambda y_{n+1} - y_n = 0$$

il cui polinomio caratteristico è

$$p(r) = r^2 - 2h\lambda r - 1 \tag{6.33}$$

se $\lambda < 0$ allora si hanno due radici distinte, $0 < r_1 < 1$ e $r_2 < -1$, come è facile verificare osservando che

$$p(-1) = -p(1) = 2h\lambda < 0; \quad p(0) = -1.$$

La soluzione generale dell'equazione alle differenze

$$y_n = Ar_2^n + Br_1^n$$

(dove A e B sono costanti da determinarsi), si comporta, per $n \to \infty$, come $y_n \simeq Ar_2^n$ in quanto il termine in r_1 tende a zero. Essendo $r_2 < -1$ la soluzione cresce in modulo per cui il metodo non è assolutamente stabile. Se venisse utilizzato per risolvere un problema di Cauchy con soluzione inerentemente stabile produrrebbe, al crescere di n, una soluzione numerica inattendibile.

Se nel problema test $\lambda > 0$ (soluzione inerentemente instabile) le radici del polinomio (6.33) divengono $-1 < r_2 < 0$ e $r_1 > 1$, come è facile verificare osservando che

$$p(-1) = -p(1) = 2h\lambda > 0; \quad p(0) = -1.$$

La soluzione generale dell'equazione alle differenze si comporta, in questo caso come $y_n \simeq Br_1^n$, poichè, per $n \to \infty$, il termine in r_2 tende a zero. Il metodo è quindi relativamente stabile e fornirà una soluzione numerica con errore relativo limitato (la radice parassita r_2 al crescere di n influenza sempre meno la soluzione).

Osservazione 6.2.5 *Garantita la stabilità relativa bisognerà scegliere il passo d'integrazione h sufficientemente piccolo per garantire l'accuratezza.*

Osservazione 6.2.6 *Nei metodi ad un passo si parla solo di stabilità assoluta in quanto non ha senso parlare di stabilità relativa (si osservi al riguardo che il polinomio caratteristico ha una sola radice).*

6.3 Riepilogo (M.F.)

Sintetizzando quanto esposto nel presente capitolo abbiamo

1. I metodi ad un passo (Taylor e Runge-Kutta), possono essere utilizzati sfruttando la sola condizione iniziale fornita nel problema di Cauchy, mentre, per utilizzare i metodi LMM a p passi, è necessario preventivamente ottenere le prime $p-1$ valutazioni.

2. Per aumentare l'accuratezza nei metodi di Taylor e Runge-Kutta è necessario aumentare il numero di valutazioni funzionali ad ogni passo, mentre, per i LMM a p passi basta sempre una sola valutazione pur avendo ordine p (espliciti) o $p+1$ (impliciti).

3. Il concetto di consistenza permette di legare l'errore di troncamento locale alla convergenza e precisamente

   (a) la consistenza è condizione necessaria e sufficiente per la convergenza nei metodi a un passo;

   (b) la consistenza (condizione necessaria) + la zero stabilità (root condition) garantiscono la convergenza nei metodi a p passi.

4. Il concetto di zero stabilità si introduce solo per i metodi a p passi in quanto, con questi metodi, si sostituisce al problema continuo, dipendente da una sola condizione iniziale, un problema discreto dipendente da p condizioni (per fissare le p costanti arbitrarie della soluzione generale dell'equazione alle differenze associata al metodo numerico).

5. Per lo stesso motivo solo per i LMM si può parlare di stabilità relativa.

6. Oltre all'accuratezza è cruciale il concetto di assoluta stabilità che può vincolare il passo h d'integrazione (regione di assoluta stabilità) tanto da rendere praticamente inutilizzabile un metodo (se h è troppo piccolo non si procede nell'integrazione).

7. I metodi impliciti, pur essendo più laboriosi (equazione da risolvere ad ogni passo), hanno regioni di stabilità più ampie (permettono passi h più grandi).

6.3.1 Esercizi

Esercizio 6.3.1 *Dato il problema di Cauchy*

$$\begin{cases} y' = y - x^2 + 2x - 1 \\ y(0) = 1 \end{cases}$$

la cui soluzione è $y(x) = x^2 + 1$, *calcolare con passo* $h = 0.05$ *la soluzione in* $[0, 1]$ *con*

1. *il metodo di Heun;*

2. *il metodo dei Trapezi (esplicitando* y_{n+1} *in funzione di* y_n*).*

Giustificare i risultati ottenuti.

Esercizio 6.3.2 *Integrare nell'intervallo* $[0, 1]$ *il seguente problema di Cauchy*

$$\begin{cases} y' = -15y \\ y(0) = 1 \end{cases}$$

come passi d'integrazione $h = 0.01$ *e* $h = 0.1$ *utilizzando*

1. *la formula del punto medio;*

2. *la formula dei trapezi.*

Giustificare i risultati ottenuti.

Esercizio 6.3.3 *Dato il metodo numerico*

$$y_{n+3} + \frac{3}{2}y_{n+2} - 3y_{n+1} + \frac{1}{2}y_n = 3hf(x_{n+2}, y_{n+2});$$

1. *è consistente ?*

2. *E' convergente?*

Giustificare le risposte date.

Esercizio 6.3.4 *Scrivere una function in MATLAB richiamabile con*

```
[xn2,yn2]=adabas2(xn,xn1,h,yn,yn1,f)
```

che esegua un passo del metodo d'integrazione di Adams-Bashforth di ordine 2.

Esercizio 6.3.5 *Integrare nell'intervallo* $[0, 2\pi]$, *utilizzando diversi passi d'integrazione, il sistema*

$$\begin{cases} y' = x(t) \\ x' = -y(t) \end{cases} ; \quad \begin{cases} y(0) = 0 \\ x(0) = 1 \end{cases} ,$$

mediante

1. *il metodo di Eulero;*

2. *il metodo di Heun.*

Dire qual'è la linea i cui punti hanno coordinate $P(t) \equiv (x(t), y(t))$ *e tracciarne il grafico.*

Bibliografia

[1] K.E.Atkinson, *An Introduction to Numerical Analysis,* John Wiley & Sons, New York, 1978

[2] J.C.Butcher, *The Numerical Analysis od Ordinary Differential Equations: Runge-Kutta and General Linear Methods,* John Wiley & Sons, Chichester, 1987

[3] S.D.Conte e C.de Boor, *Elementary Numerical Analysis,* McGraw-Hill Kogakusha, Ltd., Tokio, 1980

[4] G.Dahlquist e A.Bjorck, *Numerical Methods,* Prentice-Hall, Englewood Cliffs, NJ, 1074

[5] F.Fontanella e A.Pasquali, *Calcolo Numerico. Metodi e algoritmi,* Vol.1,2, Pitagora Editrice, Bologna, 1982

[6] G.Gambolati, *Metodi Numerici*, Cortina, Padova, 1994

[7] L.Gotusso, *Calcolo Numerico,* Clup, Milano, 1978

[8] G.H.Golub e C..Van Loan, *Matrix Computation,* The John Hopkins Press, Baltimore, 1989

[9] J.D.Lambert, *Numerical Methods for Ordinary Differential Systems. The Initial Value Problem*, John Wiley & Sons, Chichester, 1991

[10] A.M.Ostrowski, *Solution of equations and systems of equations,* Academic Press, New York, 1966

[11] J.R.Rice, *The Approximation of functions,* Vol I,II, Addison-Wesley, New York, 1969

[12] J.Stoer e R.Bulish, *Introduction to Numerical Analysis,* Springer-Verlag, New York, 1980

[13] G.Strang, *Linear algebra and its applications,* Academic Press, New York, 1980

[14] A.Stroud, *Numerical Quadrature and Solutio of Ordinary Differential Equations,* Springer-Verlag, New York, 1974