

**GIOVANNI MONEGATO**

# **Metodi e algoritmi per il CALCOLO NUMERICO**

**CLUT**

I diritti di elaborazione, di traduzione o l'adattamento anche parziale in qualsiasi forma, di memorizzazione anche digitale, su supporti di qualsiasi tipo, di riproduzione e di adattamento totale o parziale con qualsiasi mezzo (compresi i microfilm e le copie fotostatiche) sono riservati per tutti i Paesi.  
Fotocopie per uso personale (cioè privato ed individuale) nei limiti del 15% di ciascun volume possono essere effettuate negli esercizi che aderiscono all'accordo S.I.A.E. - S.N.S. e C.N.A. Confartigianato, C.A.S.A., Confcommercio del 18 Dicembre 2000, dietro pagamento del compenso previsto in tale accordo, conformemente alla legge n. 633 del 23.04.1941.  
Per riproduzioni ad uso non personale l'Editore potrà concedere a pagamento l'autorizzazione a riprodurre un numero di pagine non superiore al 15% delle pagine del presente volume. Le richieste per tale tipo di riproduzione vanno inoltrate esclusivamente all'indirizzo dell'Editore.

La messa a punto di un libro è un'operazione complessa ed articolata, che necessita di studi, progettualità grafica, nonché di numerosi controlli di testo, immagine, stili grafici e di stampa. È praticamente impossibile pubblicare un libro scevro da errori. La C.L.U.T. ringrazia sin d'ora i lettori che vorranno segnalare all'indirizzo dell'Editore eventuali errori riscontrati nella lettura del libro.

© 2008 C.L.U.T. Editrice  
Proprietà letteraria riservata  
Stampato in Italia da STAMPATRE - Torino  
Copyright C.L.U.T. - Torino - Settembre 2008

ISBN 978-88-7992-265-4

Edizioni C.L.U.T. - Torino  
Corso Duca degli Abruzzi, 24 - 10129 Torino  
Tel. 011.5647980 - Fax 011.542192  
e-mail: clut@inrete.it - [www.clut.it](http://www.clut.it)

*A Paola*



# Indice

<b>Prefazione</b>	<b>ix</b>
<b>1 Aritmetica, errori</b>	<b>1</b>
1.1 Sistemi di numerazione . . . . .	1
1.2 Rappresentazione dei numeri in un calcolatore . . . . .	3
1.3 Errori di arrotondamento, operazioni di macchina . . . . .	6
1.4 Cancellazione numerica . . . . .	10
1.5 Problemi numerici e algoritmi . . . . .	12
1.6 Condizionamento di un problema, stabilità numerica di un algoritmo . . . . .	13
Bibliografia . . . . .	18
Esercizi proposti . . . . .	19
<b>2 Richiami sulle matrici</b>	<b>23</b>
2.1 Preliminari . . . . .	23
2.2 Operazioni tra matrici . . . . .	24
2.3 Matrici con proprietà particolari . . . . .	27
2.4 Matrici non singolari . . . . .	28
2.5 Autovalori di una matrice . . . . .	30
2.6 Norme di vettore e di matrice . . . . .	31
Bibliografia . . . . .	34
Esercizi proposti . . . . .	35
<b>3 Sistemi lineari</b>	<b>37</b>
3.1 Preliminari . . . . .	37
3.2 Metodi diretti . . . . .	41
3.2.1 Il metodo di eliminazione di Gauss . . . . .	41
3.2.2 Pivoting e scaling . . . . .	45

3.2.3	Decomposizione di Gauss e fattorizzazione LU . . . . .	48
3.2.4	Raffinamento iterativo . . . . .	60
3.2.5	Sistemi complessi . . . . .	62
3.2.6	Matrice inversa . . . . .	63
3.3	Metodi iterativi . . . . .	66
3.3.1	Metodo di Jacobi . . . . .	68
3.3.2	Metodo di Gauss-Seidel . . . . .	69
3.3.3	Metodo di sovrarilassamento (SOR) . . . . .	73
3.3.4	Metodo del gradiente coniugato . . . . .	76
Bibliografia . . . . .		81
Esercizi proposti . . . . .		83
<b>4 Autovalori di matrici</b>		<b>87</b>
4.1	Preliminari . . . . .	87
4.2	Metodo delle potenze . . . . .	91
4.3	Metodo delle potenze inverse . . . . .	98
4.4	Trasformazioni di similitudine e trasformazioni di Householder . . . . .	99
4.5	Applicazioni delle trasformazioni di Householder . . . . .	103
4.5.1	Fattorizzazione $QR$ di una matrice . . . . .	103
4.5.2	Riduzione di una matrice alla forma simile tridiagonale oppure di Hessenberg . . . . .	105
4.5.3	Decomposizione ai valori singolari . . . . .	110
4.6	Calcolo degli autovalori di una matrice tridiagonale simmetrica . . . . .	112
4.7	Cenni sul metodo $QR$ . . . . .	114
Bibliografia . . . . .		116
Esercizi proposti . . . . .		117
<b>5 Approssimazione di dati e di funzioni</b>		<b>121</b>
5.1	Preliminari . . . . .	121
5.2	Interpolazione polinomiale . . . . .	126
5.2.1	Formula di interpolazione di Lagrange . . . . .	126
5.2.2	Formula di interpolazione di Hermite . . . . .	134
5.3	Formula di Newton alle differenze divise . . . . .	135
5.4	Formule di Newton alle differenze finite . . . . .	142
5.5	Interpolazione trigonometrica . . . . .	145
5.6	Algoritmo FFT . . . . .	148
5.7	Interpolazione con funzioni polinomiali a tratti. Funzioni spline . . . . .	153
5.8	Interpolazione di funzioni di più variabili . . . . .	163
5.9	Metodo dei minimi quadrati. Caso lineare discreto . . . . .	165
5.10	Cenni sul caso continuo del metodo dei minimi quadrati . . . . .	170
5.11	Derivazione numerica . . . . .	173
Bibliografia . . . . .		175

---

Esercizi proposti . . . . .	176
<b>6 Equazioni non lineari</b>	<b>181</b>
6.1 Preliminari . . . . .	181
6.2 Radici reali di equazioni non lineari . . . . .	183
6.2.1 Metodo di bisezione . . . . .	183
6.2.2 Metodi delle secanti, delle tangenti (Newton-Raphson) e altri . . . . .	186
6.2.3 Test di convergenza . . . . .	195
6.2.4 Metodi iterativi in generale . . . . .	196
6.2.5 Metodo di accelerazione $\Delta^2$ di Aitken . . . . .	199
6.3 Sistemi di equazioni non lineari . . . . .	201
6.3.1 Metodo di Newton e sue varianti . . . . .	202
6.3.2 Metodi iterativi in generale . . . . .	204
6.4 Equazioni algebriche (a coefficienti reali) . . . . .	205
6.4.1 Radici reali . . . . .	205
6.4.2 Metodo di Bairstow . . . . .	208
6.5 Ottimizzazione . . . . .	211
Bibliografia . . . . .	214
Esercizi proposti . . . . .	215
<b>7 Calcolo di integrali</b>	<b>219</b>
7.1 Preliminari. Formule di quadratura interpolatorie . . . . .	219
7.2 Formule di Newton-Cotes . . . . .	223
7.3 Polinomi ortogonali . . . . .	225
7.4 Formule di quadratura Gaussiane . . . . .	229
7.5 Costruzione delle formule Gaussiane . . . . .	233
7.6 Stima dell'errore $R_n(f)$ . . . . .	235
7.7 Formule composte . . . . .	238
7.8 Routine automatiche . . . . .	240
7.9 Integrazione su intervalli infiniti . . . . .	242
7.10 Alcune applicazioni delle formule di quadratura . . . . .	242
Bibliografia . . . . .	246
Esercizi proposti . . . . .	247
<b>8 Equazioni differenziali ordinarie</b>	<b>251</b>
8.1 Preliminari . . . . .	251
8.2 Metodi one-step espliciti. Metodi Runge-Kutta . . . . .	259
8.2.1 Comportamento locale dei metodi one-step . . . . .	260
8.2.2 Esempi di metodi one-step espliciti . . . . .	262
8.2.3 Convergenza dei metodi one-step espliciti . . . . .	267
8.2.4 Stima dell'errore locale di troncamento e scelta del passo di integrazione . . . . .	268
8.3 Metodi multistep lineari . . . . .	271

8.3.1	Comportamento locale dei metodi multistep lineari . . . . .	272
8.3.2	Metodi multistep di Adams . . . . .	273
8.3.3	Convergenza dei metodi multistep . . . . .	276
8.3.4	Metodi previsore-correttore . . . . .	277
8.3.5	Metodi multistep a passo variabile . . . . .	278
8.4	Stabilità dei metodi numerici . . . . .	279
8.5	Sistemi stiff . . . . .	287
8.6	Problemi con valori ai limiti . . . . .	291
8.6.1	Metodo delle differenze finite . . . . .	293
8.6.2	Metodo shooting (o di puntamento) . . . . .	296
8.6.3	Metodo di collocazione . . . . .	299
Bibliografia . . . . .		302
Esercizi proposti . . . . .		304
<b>9 Equazioni alle derivate parziali</b>		<b>307</b>
9.1	Preliminari . . . . .	307
9.2	Caratteristiche. Classificazione delle equazioni quasi-lineari di ordine 2 . . . . .	311
9.3	Metodi alle differenze finite . . . . .	316
9.3.1	Generalità . . . . .	316
9.3.2	Equazioni di tipo iperbolico . . . . .	321
9.3.3	Equazioni di tipo parabolico . . . . .	334
9.3.4	Consistenza, stabilità e convergenza degli schemi alle differenze finite per problemi a valori iniziali . . . . .	342
9.3.5	Equazioni di tipo ellittico . . . . .	346
9.4	Metodi dei residui pesati . . . . .	350
9.5	Formulazione debole e elementi finiti . . . . .	355
Bibliografia . . . . .		378
Esercizi proposti . . . . .		380
<b>Bibliografia Generale</b>		<b>385</b>
<b>Indice analitico</b>		<b>387</b>

# Prefazione

In questi ultimi decenni l'uso della simulazione matematica, quale strumento per lo studio di fenomeni fisici e non, ha subito una crescita vertiginosa e assunto un ruolo diventato fondamentale nello sviluppo delle scienze applicate e di nuove tecnologie. Sempre più spesso la simulazione matematica sostituisce la sperimentazione fisica, con notevoli vantaggi in termini di costi, tempi e sicurezza. Anche nella vita di tutti i giorni la *modellistica matematica* svolge un ruolo determinante: si pensi, ad esempio, alle previsioni atmosferiche, alla tomografia assiale computerizzata o alla risonanza magnetica, alla ricostruzione di immagini trasmesse tramite segnali, al controllo di un aeromobile durante il volo, al controllo di produzioni industriali.

Tale sviluppo è reso possibile non solo dall'avvento di elaboratori elettronici sempre più potenti, ma anche dalla continua evoluzione di nuovi e sempre più sofisticati *metodi numerici* che consentono di risolvere i modelli matematici proposti, nei tempi desiderati e con la precisione richiesta.

La risoluzione di modelli matematici non banali raramente può prescindere dall'uso di metodi di calcolo numerico approssimato. Anche quando il modello può essere affrontato con tecniche esclusivamente analitiche, spesso la rappresentazione della soluzione cercata risulta inutilizzabile in pratica, o semplicemente troppo "costosa" in termini di operazioni aritmetiche e quindi di tempi di calcolo; più efficiente potrebbe invece rivelarsi un approccio numerico che permetta di ottenere direttamente un'approssimazione, eventualmente discreta, della soluzione incognita. Va inoltre ricordato che spesso la via analitica non è assolutamente percorribile.

La stessa costruzione di modelli matematici è spesso influenzata dall'esistenza o meno di metodi analitici e numerici che ne consentano la risoluzione, quasi sempre approssimata. Talvolta, alcune semplificazioni o idealizzazioni vengono introdotte nel modello proprio per consentirne la risoluzione, anche se approssimata, con i requisiti di efficienza e precisione richiesti. Altre volte, è la presenza di un nuovo modello matematico a suggerire la costruzione di tecniche numeriche originali. Diventa pertanto indispensabile

conoscere i metodi numerici che la comunità scientifica ha elaborato, acquisendo al tempo stesso la capacità di modificarli e adattarli a possibili nuove situazioni o esigenze.

Certamente la risoluzione di problemi scientifici non è la sola motivazione, anche se è senz'altro la principale, per lo studio e lo sviluppo di metodi numerici. L'analisi di tali metodi è anche un'attività matematica vasta e stimolante, il cui obiettivo primario deve comunque essere la effettiva costruzione di approssimazioni numeriche con specifiche caratteristiche, tenendo presente che l'aritmetica in cui si opera è quella dei calcolatori.

Questo testo è rivolto principalmente a studenti che affrontano per la prima volta lo studio del Calcolo Numerico. Volutamente sono state omesse le dimostrazioni di molti teoremi e risultati di carattere matematico, e ciò per dare maggior rilievo alla costruzione e alle proprietà dei metodi numerici presentati. Gli enunciati dei teoremi sono spesso inseriti solo per poter esaminare le caratteristiche dei metodi proposti e giustificare quindi le scelte effettuate; per altro le dimostrazioni mancanti possono essere facilmente reperite nei libri citati nelle bibliografie di fine capitolo. Tema centrale di questo testo è pertanto l'esposizione delle problematiche e delle idee che ispirano la costruzione di quei metodi numerici che vengono considerati di base per il Calcolo Numerico e che spesso si presentano quali passi intermedi nella risoluzione di modelli matematici più complessi.

Dopo una breve introduzione sull'aritmetica dei calcolatori e le sue implicazioni (capitolo 1), e alcuni richiami di algebra lineare (capitolo 2), esamineremo metodi numerici per la risoluzione di sistemi di equazioni lineari (capitolo 3), per il calcolo di autovalori di matrici (capitolo 4), per l'approssimazione di dati e di funzioni (capitolo 5), per il calcolo di radici di equazioni non lineari e per la soluzione di sistemi di equazioni non lineari (capitolo 6), per la valutazione numerica di integrali (capitolo 7), per la risoluzione di equazioni differenziali ordinarie (capitolo 8) e alle derivate parziali (capitolo 9).

Tutti i risultati numerici presentati in questo testo sono stati ottenuti utilizzando l'aritmetica floating-point (Standard IEEE) con precisione semplice, ovvero con errore relativo  $\simeq 5.96E - 8$ .

Concludiamo questa breve prefazione ricordando che sino agli anni 70 l'implementazione dei suddetti metodi numerici avveniva a livello artigianale. Spesso le case costruttrici di calcolatori dotavano le loro macchine di *librerie software* contenenti alcuni dei programmi di calcolo più comuni, mentre alcune riviste specializzate di calcolo numerico (Communications of the Association for Computing Machinery, The Computer Journal, Numerische Mathematik, Computing, ACM Transactions on Mathematical Software, ecc.) pubblicavano anche *routines* che implementavano i metodi proposti dagli autori. Questo tipo di risposta era tuttavia insufficiente a soddisfare le sempre più crescenti esigenze di calcolo della comunità scientifica. La necessità di collezioni omogenee di routine, con elevati standards di qualità e affidabilità, ha ben presto favorito la costituzione di organizzazioni internazionali, quali il NAG, l'IMSL e l'ABACI(<sup>†</sup>), capaci di produrre e diffondere sofisticate librerie di programmi di calcolo per la risoluzione dei problemi

---

(<sup>†</sup>) NAG: The Numerical Algorithms Group Ltd., Oxford, Inghilterra; IMSL: International Mathematical and Statistical Libraries, Houston, Texas, USA; ABACI: The Scientific Desk, Raleigh, North Carolina, USA.

numerici cosiddetti di base. Ciò nonostante, a nostro avviso, un corretto ed efficiente utilizzo di tali programmi non può prescindere dalla conoscenza dei metodi numerici su cui essi si basano e delle loro caratteristiche. Inoltre, non sempre tali collezioni contengono una risposta idonea a tutti i problemi che un utente si trova a dover risolvere; non di rado il metodo numerico risolutivo deve essere costruito ex-novo dallo stesso utente.

Questo testo è una revisione del volume *Fondamenti di Calcolo Numerico*, pubblicato nel 1998 dalla stessa casa editrice. In questa nuova edizione sono stati introdotti alcuni miglioramenti, nuovi risultati e riferimenti bibliografici, e soprattutto rivista la presentazione del metodo degli elementi finiti.

Le figure riportate in copertina, da sinistra a destra partendo dall'alto, rappresentano le seguenti prove di simulazione numerica al calcolatore:

- comportamento aerodinamico dell'ala anteriore di una vettura di Formula Uno
- irradiazione di un'antenna a parabola per il collegamento satellitare
- simulazione idrodinamica per il miglioramento dell'opera viva di un'imbarcazione a vela
- interazione meccanica tra uno strato di cellule endoteliali e un substrato elastico (ad es. un tessuto muscolare)

Un ringraziamento particolare alla collega Letizia Scuderi per le utili osservazioni, a Giuseppe Ghibò per la composizione L<sup>A</sup>T<sub>E</sub>X, ai colleghi Renato Orta e Luigi Preziosi del Politecnico di Torino e Antonio Strozzi dell'Università di Modena per le immagini inserite in copertina.

Torino, giugno 2008



# Capitolo 1

## Aritmetica, errori

### 1.1 Sistemi di numerazione

Il sistema da noi comunemente usato per rappresentare i numeri è quello *decimale*. Il numero 31.57, per esempio, viene da noi interpretato nel modo seguente:

$$31.57 = 3 \times 10^1 + 1 \times 10^0 + 5 \times 10^{-1} + 7 \times 10^{-2}$$

Il numero 10 è la *base del sistema di numerazione* decimale. Ogni numero reale ha una rappresentazione (decimale) unica, eccetto quando la parte frazionaria contiene una sequenza infinita di 9 consecutivi (oppure è finita):

$$0.31999\dots 9\dots \quad \text{e} \quad 0.32$$

rappresentano lo stesso numero.

Non esiste tuttavia una ragione fondamentale per assumere 10 come base del sistema di numerazione; qualunque intero  $N > 1$  può essere scelto come base, ed ogni numero reale  $a$  può essere scritto nella forma

$$a = \pm(a_m N^m + a_{m-1} N^{m-1} + \dots + a_1 N + a_0 + a_{-1} N^{-1} + \dots)$$

ovvero

$$a = \pm a_m a_{m-1} \dots a_1 a_0 . a_{-1} \dots$$

dove i coefficienti  $a_i$  sono interi compresi tra 0 e  $N - 1$  ( $0 \leq a_i \leq N - 1$ ). Questa rappresentazione è unica, tranne quando la parte frazionaria contiene una successione di infinite cifre  $a_{-k} = N - 1$  consecutive. In questo caso esiste una seconda<sup>(†)</sup> rappresentazione alternativa, che possiamo però definire equivalente alla prima e che otteniamo sopprimendo la predetta successione e aggiungendo una unità all'ultima cifra rimasta:

$$N = 8 \quad a = 370.25777\dots 7\dots = 370.26$$

---

(†) E questa sola.

Noi sceglieremo sempre questa seconda rappresentazione.

Vediamo alcuni esempi di rappresentazioni di numeri<sup>(†)</sup>:

$$\begin{aligned}(17)_{10} &= (11)_{16} = (10001)_2 = (10000.111\dots1\dots)_2 \\ (0.25)_{10} &= (0.24999\dots9\dots)_{10} = (0.01)_2 = (0.00111\dots1\dots)_2 \\ (0.1)_{10} &= (0.063146314\dots)_8 = (0.0001100110011\dots)_2\end{aligned}$$

Più piccola è la base scelta, più semplici risultano le regole per le operazioni aritmetiche. In particolare, nel caso del sistema in base 2, detto anche *sistema binario*, i coefficienti  $a_i$  possono assumere solamente i valori 0 e 1, e le operazioni somma e prodotto<sup>(††)</sup> sono definite dalle regole seguenti:

$$\begin{array}{ll} 0 + 0 = 0 & 0 \cdot 0 = 0 \\ 0 + 1 = 1 + 0 = 1 & 0 \cdot 1 = 1 \cdot 0 = 0 \\ 1 + 1 = 10 & 1 \cdot 1 = 1 \end{array}$$

### ► Esempi.

$$\begin{array}{rcl} 10110.011 & + & 1011.11 & \times \\ 1010.11 & = & 100.1 & = \\ \hline 100001.001 & & 101111 & \\ & & 000000 & \\ & & 000000 & \\ & & 101111 & \\ \hline & & 110100.111 & \end{array}$$



Osserviamo tuttavia che più piccola è la base del sistema più lunga è (in genere) la rappresentazione di un numero.

Il sistema binario risulta particolarmente interessante perché permette di rappresentare ogni numero reale mediante una sequenza di 0 e 1; quindi per rappresentare la generica cifra binaria (che in seguito chiameremo *bit*) è sufficiente un “alfabeto” di due soli simboli, e quest’ultimo può essere realizzato con qualsiasi oggetto capace di assumere due stati diversi (ad esempio stato di magnetizzazione, e non, di un nucleo di ferrite; condutività, e non, di un diodo).

Osserviamo ancora che, nota la rappresentazione binaria di un numero, per ottenere le corrispondenti rappresentazioni in base 4, 8, 16 (in generale  $2^n$ ) è sufficiente suddividere il numero binario in gruppi di 2, 3, 4 ( $n$ ) bit partendo dal punto “decimale”, proseguendo verso sinistra per la parte intera e verso destra per quella frazionaria, e interpretare ogni singolo gruppo come rappresentazione binaria di un numero intero in base rispettivamente

<sup>(†)</sup> Con la notazione (...)<sub>N</sub> intendiamo ricordare che la rappresentazione contenuta nella parentesi si riferisce al sistema di numerazione in base  $N$ .

<sup>(††)</sup> Ricordiamo che le operazioni binarie di sottrazione, prodotto e divisione sono riconducibili a operazioni di somma (vedasi [1.6]).

4, 8, 16 ( $2^n$ ). Nel caso in cui i gruppi estremi (il primo di sinistra della parte intera e l'ultimo di destra della parte frazionaria) hanno meno di 2, 3, 4, ( $n$ ) bit, occorre completarli aggiungendo degli zeri. Per esempio, se

$$a = (1100101.00100001)_2$$

per ottenere la rappresentazione in base  $N = 8$  operiamo nel modo seguente:

$$a = (\underbrace{001}_{1} \underbrace{100}_{4} \underbrace{101}_{5} . \underbrace{001}_{1} \underbrace{000}_{0} \underbrace{010}_{2})_8$$

Ovviamente vale anche il viceversa: nota la rappresentazione di un numero in base  $N = 2^n$ , per ottenere la corrispondente espressione binaria è sufficiente sostituire ogni singola cifra della rappresentazione in base  $N$  con la sua rappresentazione binaria (formata esattamente da  $n$  cifre binarie).

Nel sistema di numerazione in base 16 i valori 10, 11, ..., 15 vengono sostituiti, per evitare ambiguità, con le lettere  $A, B, \dots, F$ .

#### ► Esempi.

$$\begin{aligned} (17)_{10} &= (10001)_2 = (101)_4 = (21)_8 = (11)_{16} \\ (13.25)_{10} &= (1101.01)_2 = (31.1)_4 = (15.2)_8 = (D.4)_{16} \\ (0.1)_{10} &= (0.00011001100\dots)_2 = (0.01212\dots)_4 = (0.063146314\dots)_8 \\ &\quad = (0.1999\dots 9\dots)_{16} \end{aligned}$$

◀

Ricordiamo infine che un numero reale con rappresentazione decimale non finita ha necessariamente quella binaria composta da infiniti bit; inoltre, un numero con rappresentazione decimale finita (vedi ad esempio 0.1) può avere quella binaria (oppure in base 4, 8, 16) infinita.

## 1.2 Rappresentazione dei numeri in un calcolatore

In un calcolatore siamo costretti, per ovvie ragioni, a riservare alla rappresentazione di ogni numero reale uno “spazio” finito di *memoria*. Pertanto possiamo rappresentare esattamente solo quei numeri contenuti nello spazio previsto per ognuno di essi; questi numeri sono chiamati *numeri di macchina*.

Ricordiamo che, qualunque sia il sistema di numerazione scelto, le singole cifre dei numeri di macchina verranno rappresentate nel calcolatore secondo il sistema binario, e quindi con una sequenza di bit. Tuttavia, ciò non significa che i numeri siano poi trattati come se fossero binari. Nel caso  $N = 2^n$ ,  $n > 1$ , per esempio, quando il calcolatore dovrà operare su un numero di macchina non agirà sul singolo bit, ma su gruppi di  $n$  bit consecutivi; ognuno di questi gruppi rappresenta una cifra  $a_i$  nella base  $N$ , secondo regole e

modalità prestabilite. L’aritmetica è effettivamente quella in base  $N$ ; la rappresentazione binaria delle singole cifre  $a_i$  è solo un espediente per poterle rappresentare con sequenze di  $n$  oggetti capaci di assumere ciascuno due stati diversi.

Esaminiamo ora le convenzioni (due) che nei moderni calcolatori vengono usate per la rappresentazione dei numeri di macchina.

**Numeri interi.** Alla rappresentazione di ogni numero intero viene riservato uno spazio di memoria di lunghezza costante (32 o 64 bit), corrispondente ad un numero massimo di cifre, per esempio  $l$ , nel sistema di numerazione scelto. Risulteranno pertanto rappresentabili solamente quei numeri interi che, nel sistema di numerazione scelto (normalmente è quello binario), non hanno più di  $l$  cifre (binarie). Per poter rappresentare i numeri interi con più di  $l$  cifre dovremo considerarli come numeri reali e ricorrere alla rappresentazione che segue.

**Numeri reali.** (rappresentazione *floating-point*) Ogni numero reale  $a$  può essere scritto nella forma

$$a = pN^q$$

dove  $p$  è reale,  $N$  è la base del sistema di numerazione scelto e  $q$  è un intero positivo, negativo o nullo. Questa rappresentazione non è unica; infatti abbiamo

$$a = pN^q = p'N^{q-1} = p''N^{q+1} = \dots, \quad p' = pN, \quad p'' = \frac{p}{N}$$

La rappresentazione di  $a \neq 0$  si dice *normalizzata* quando

$$N^{-1} \leq |p| < 1$$

ossia quando la prima cifra di  $p$  (dopo il punto “decimale”) è diversa da zero. Per esempio, supponendo  $N = 10$ , le rappresentazioni normalizzate di

$$a = (123.715)_{10} \quad \text{e} \quad b = (0.000718)_{10}$$

sono

$$a = 0.123715 \times 10^3 \quad \text{e} \quad b = 0.718000 \times 10^{-3}$$

Quella dell’intero

$$a = (225478)_{10}$$

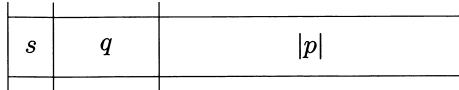
è

$$a = 0.225478 \times 10^6$$

In questo caso chiamiamo  $p$  e  $q$  rispettivamente *mantissa* e *caratteristica* (o esponente) del numero  $a$ . Fissata la base  $N$ , ogni numero reale  $a \neq 0$  è univocamente definito dalla coppia  $(p, q)$ ; per cui è sufficiente memorizzare quest’ultima. Non esistendo la rappresentazione normalizzata del numero  $a = 0$ , per convenzione potremmo individuare tale numero con la coppia  $(0, 0)$ .

In un calcolatore lo spazio riservato alla rappresentazione di ogni numero reale può essere pensato suddiviso nel modo seguente:

$$s = \text{segno di } a$$



Come numeri reali di macchina definiamo quindi quei numeri le cui mantisse e caratteristiche sono rappresentabili esattamente negli spazi a loro riservati. In particolare, se lo spazio dedicato alla rappresentazione di  $|p|$  corrisponde a  $t$  cifre nel sistema di numerazione scelto, saranno rappresentabili solamente quelle mantisse che non hanno più di  $t$  cifre. La caratteristica invece dovrà soddisfare una disuguaglianza del tipo

$$m \leq q \leq M$$

dove  $m < 0$  e  $M > 0$  sono interi che possono variare da calcolatore a calcolatore. Per la “memorizzazione” di  $q$  di solito si suggerisce di operare nel modo seguente: fissato  $m$ , memorizzare la nuova quantità  $q^* = q - m$  che risulta sempre non negativa, “ricordando” che ogni numero viene così rappresentato a meno del fattore costante  $N^m$ .

▷ **Osservazione.** Nella rappresentazione binaria il primo bit della mantissa è sempre 1, mentre nelle rappresentazioni con base  $N > 2$  ciò non è sempre vero. Per esempio nel sistema esadecimale ( $N = 16$ ) quando  $a_{-1} = 1$  abbiamo i primi tre bit nulli:

$$(p)_{16} = (0.191523D\ldots)_{16} = 0.\underbrace{0001}_{1} \underbrace{1001}_{9} \underbrace{0001}_{1} \underbrace{0101}_{5} \underbrace{0010}_{2} \underbrace{0011}_{3} \underbrace{1101}_{D}\ldots$$

In questo caso la memorizzazione dei primi tre bit nulli comporta uno “spreco” di spazio, ovvero una minor precisione nella rappresentazione della mantissa  $p$  nel calcolatore. Infatti, se supponiamo che lo spazio di memoria riservato alla mantissa sia costituito da 24 bit, nel caso  $N = 2$  riusciamo a memorizzare i primi 24 bit della rappresentazione normalizzata

$$(p)_2 = 0.110010001010100100011110\dot{1}\ldots$$

Inoltre, poiché il primo bit della mantissa binaria è sempre uguale a 1, possiamo fare a meno di memorizzare quest’ultimo e guadagnare così un bit, che potremmo per esempio aggiungere a quelli riservati alla caratteristica  $q^*$ . ◁

Nella maggior parte dei moderni calcolatori, per la memorizzazione dei numeri reali in aritmetica floating-point esiste la possibilità di riservare a ciascun numero una lunghezza complessiva di 32 bit (in questo caso l’aritmetica viene definita in *semplice precisione*), oppure di 64 bit (aritmetica in *doppia precisione*). Nel caso della semplice precisione, e sistema di numerazione binario, i 32 bit vengono suddivisi nel modo seguente: il primo

bit è riservato al segno del numero (0 se positivo e 1 se negativo), nei successivi 8 viene memorizzato l'esponente normalizzato

$$q^* = q + 127, \quad 0 \leq q^* \leq 255 \quad (m = -127, M = 128)$$

mentre nei restanti vengono memorizzati i 23 bit della mantissa che seguono l'1 iniziale.

Lo standard IEEE (IEEE Standard for Binary Floating-Point Arithmetic 754-1985), ormai adottato dalla maggior parte delle aziende produttrici delle unità di calcolo (CPU) degli elaboratori, prescrive l'uso dell'aritmetica binaria sopra descritta, con le seguenti modifiche. La mantissa viene normalizzata nella forma  $p = \pm 1.a_{-1}a_{-2}\dots a_{-23}$  e i bit  $a_{-1}, a_{-2}, \dots, a_{-23}$  sono memorizzati nello spazio (23 bit) ad essa riservato. L'esponente  $q^* = 0$  ( $q = -127$ ) viene usato per la codifica dello zero (in questo caso  $p = 0$ ) e di eventuali numeri denormalizzati con mantisse del tipo  $p = \pm 0.a_{-1}a_{-2}\dots a_{-23}$ , dove le prime cifre possono essere tutte nulle. L'esponente  $q^* = 255$  viene riservato per rappresentare un *non numero*, quale il simbolo  $\infty$  (in questo caso  $p = 0$ ), oppure risultati di operazioni impossibili o non valide (ponendo  $p \neq 0$ ), come  $0/0$ ,  $\infty/\infty$ , o la radice quadrata di un numero negativo. Questi ultimi vengono denotati con il simbolo NaN (acronimo di “Not a Number”).

Il generico numero di macchina ha quindi la forma

$$\begin{aligned} & (-1)^s |p| 2^{q^*-127}, \quad 1 \leq q^* \leq 254 \\ & |p| = 1.a_{-1}a_{-2}\dots a_{-23} \end{aligned}$$

Nella doppia precisione, dei 32 bit aggiuntivi tre vengono assegnati all'esponente ( $-1023 \leq q \leq 1024$ ) e 29 alla mantissa.

Dall'anno 2000 è in corso un progetto di revisione del predetto standard, denominato IEEE 754r. Il miglioramento più evidente è l'aggiunta dello standard a 128 bit (*quadrupla precisione*) e dell'aritmetica floating-point in base  $N = 10$  (particolarmente utile nelle applicazioni di tipo finanziario e commerciale). La definizione del nuovo standard dovrebbe essere completata entro la fine del 2008 ([1.9]).

### 1.3 Errori di arrotondamento, operazioni di macchina

Ricordiamo preliminarmente che se  $x$  rappresenta un valore esatto e  $\bar{x}$  una sua approssimazione, gli errori assoluto e relativo associati a  $\bar{x}$  sono definiti rispettivamente dalle quantità

$$|x - \bar{x}| \quad \text{e} \quad \left| \frac{x - \bar{x}}{x} \right|, \quad x \neq 0$$

inoltre, possiamo scrivere

$$\bar{x} = x(1 + \varepsilon), \quad \varepsilon = \frac{\bar{x} - x}{x}$$

Come abbiamo visto nel paragrafo precedente, in un elaboratore non tutti i numeri reali possono venire rappresentati. Nel caso di un sistema floating-point con  $t$  cifre

(nella base  $N$ ) per la mantissa, tutti i numeri che nella base scelta ammettono una rappresentazione (normalizzata) con un numero di cifre nella mantissa superiore a  $t$  (e, ovviamente, esponente  $m \leq q \leq M$ ) dovranno in qualche modo venire “accorciati”, o meglio, *arrotondati* a  $t$  cifre.

Le tecniche di arrotondamento usate sono essenzialmente due:

- (i) si esclude la parte a destra della  $t$ -esima cifra,
- (ii) si aggiunge  $1/2 \cdot N^{-t}$  alla mantissa in questione e poi si tronca quest’ultima alla  $t$ -esima cifra.

► **Esempio.** Supponendo  $t = 6$  e  $N = 10$ , la mantissa

$$p = 0.724325643$$

verrebbe arrotondata a

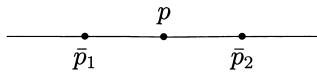
$$\bar{p} = 0.724325$$

adottando la tecnica (i), e a

$$\bar{p} = 0.724326$$

con la (ii). ◀

Le mantisse  $\bar{p}$  dei numeri di macchina,  $N^{-1} \leq |\bar{p}| < 1$ , non hanno più di  $t$  cifre, e la distanza tra due mantisse di macchina consecutive è esattamente  $N^{-t}$ . Con la tecnica (i), tutte le mantisse  $p$  comprese tra le due mantisse di macchina consecutive  $\bar{p}_1$  e  $\bar{p}_2 = \bar{p}_1 + N^{-t}$ , positive per esempio,



vengono sostituite da  $\bar{p}_1$ ; in questo caso abbiamo  $|p - \bar{p}_1| < N^{-t}$ . Con la tecnica (ii) invece, tutte le mantisse comprese nell’intervallo  $(\bar{p}_1, \bar{p}_1 + 1/2 \cdot N^{-t})$  vengono sostituite con  $\bar{p}_1$ , mentre le mantisse situate in  $[\bar{p}_1 + 1/2 \cdot N^{-t}, \bar{p}_2)$  vengono approssimate con  $\bar{p}_2$ , così che, denotando con  $\bar{p}$  la mantissa di macchina che il criterio di arrotondamento associa a  $p$ , risulta  $|p - \bar{p}| \leq 1/2 \cdot N^{-t}$ .

Dei due metodi, il primo è certamente il peggiore (ma il meno oneroso), non solo perché può provocare un errore maggiore (doppio rispetto al secondo), ma anche perché il segno di  $(p - \bar{p})/p$  è costante (positivo). Lo standard IEEE ( $N = 2$ ) prescrive l’adozione della tecnica di arrotondamento (ii), con una modifica nella definizione di  $\bar{p}$  quando si ha  $p = \bar{p}_1 + 1/2 \cdot N^{-t}$  (vedi [1.7]): tra le mantisse  $\bar{p}_1$  e  $\bar{p}_2$  viene scelta quella delle due che ha l’ultimo bit (il  $t$ -esimo) uguale a 0, cosicché la probabilità che il segno del corrispondente errore sia ‘+’ o ‘-’ è la stessa (50%) per entrambi i segni.

Sia  $a = pN^q$ ,  $N^{-1} \leq |p| < 1$ , un numero reale, e sia  $\bar{a} = \bar{p}N^q$  il corrispondente numero di macchina ottenuto mediante una delle due tecniche di arrotondamento introdotte. Poiché dalle precedenti definizioni delle due tecniche di arrotondamento delle mantisse non di macchina seguono le seguenti maggiorazioni:

$$(1.1) \quad |p - \bar{p}| \begin{cases} < N^{-t} & \text{con la (i)} \\ \leq \frac{1}{2}N^{-t} & \text{con la (ii)} \end{cases}$$

per gli errori, assoluto e relativo, associati ad  $\bar{a}$  abbiamo rispettivamente

$$|a - \bar{a}| \begin{cases} < N^{q-t} & \text{(i)} \\ \leq \frac{1}{2}N^{q-t} & \text{(ii)} \end{cases}$$

e

$$\frac{|a - \bar{a}|}{|a|} \leq \frac{|a - \bar{a}|}{N^{q-1}} \begin{cases} < N^{1-t} & \text{(i)} \\ \leq \frac{1}{2}N^{1-t} & \text{(ii)} \end{cases}$$

La quantità  $\text{eps} = N^{1-t}$  nel caso (i), e  $\text{eps} = 1/2 \cdot N^{1-t}$  in (ii), definisce la cosiddetta *precisione di macchina*. Essa è una costante caratteristica di ogni aritmetica floating-point (e tecnica di arrotondamento), e rappresenta la massima precisione di calcolo raggiungibile con il calcolatore su cui tale aritmetica è implementata. Non ha pertanto senso cercare di determinare approssimazioni con precisione relativa inferiore alla quantità  $\text{eps}$ .

Supponiamo, per esempio, che l'arrotondamento dei numeri reali venga effettuato con la tecnica (ii). Data un'approssimazione  $\bar{x}$  di un numero  $x$ , se risulta

$$|x - \bar{x}| \leq \frac{1}{2}N^{-k}, \quad k \geq 1$$

allora diciamo che l'approssimazione  $\bar{x}$  ha almeno  $k$  “decimali” corretti (nella base  $N$ ), e definiamo *significative* le cifre che precedono il  $(k+1)$ -esimo decimale (escludendo gli eventuali zeri iniziali). Più in generale, le cifre significative di una generica approssimazione  $\bar{x}$  coincidono con i decimali corretti presenti nella mantissa  $\bar{p}$  della rappresentazione (normalizzata)  $\bar{x} = \bar{p}N^q$ . Se la mantissa  $\bar{p}$  ha  $k$  decimali corretti possiamo scrivere

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

quindi, ricordando che  $|p| < 1$ , se vale una diseguaglianza del tipo

$$\frac{|x - \bar{x}|}{|x|} \leq \frac{1}{2}N^{-k}$$

possiamo senz’altro affermare che l’approssimazione  $\bar{x}$  ha almeno  $k$  cifre significative. Infatti, dall’ultima relazione deduciamo

$$|p - \bar{p}| \leq \frac{1}{2}N^{-k}|p| < \frac{1}{2}N^{-k}$$

Diciamo “almeno  $k$  cifre” perché in realtà secondo la nostra definizione potrebbero essere anche  $k+1$ .

► **Esempio.** Sia  $x = 15.2000$ ,  $\bar{x} = 15.1997$  e  $N = 10$ . L'approssimazione  $\bar{x}$  ha 3 decimali corretti e 5 cifre significative. Esaminando l'errore relativo

$$\frac{|x - \bar{x}|}{|x|} = 0.197\ldots \times 10^{-4} < \frac{1}{2} 10^{-4}$$

dedurremmo che  $\bar{x}$  ha almeno 4 cifre significative.

Se invece  $x = 99.2000$  e  $\bar{x} = 99.1997$ , pur avendo lo stesso errore assoluto del caso precedente, la relazione

$$\frac{|x - \bar{x}|}{|x|} = 0.302\ldots \times 10^{-5} < \frac{1}{2} 10^{-5}$$

ci assicura che le cifre significative di  $\bar{x}$  sono almeno 5. ◀

Qualora l'arrotondamento dei numeri reali fosse effettuato con la tecnica (i), nelle disuguaglianze che definiscono i decimali corretti, ovvero le cifre significative, occorrerebbe sostituire  $\leq 1/2 \cdot N^{-k}$  con  $< N^{-k}$ .

► **Osservazione.** La relazione (1.1) ci assicura che l'arrotondamento provocato dall'e-laboratore genera una mantissa  $\bar{p}$  le cui cifre sono tutte corrette (rispetto a  $p$ ). ◇

Finora abbiamo definito i numeri di macchina e visto come arrotondare un generico numero reale a numero di macchina. Osserviamo tuttavia che i risultati di operazioni aritmetiche tra i numeri di macchina generalmente non sono numeri di macchina; pertanto in un calcolatore risulterà impossibile implementare esattamente le operazioni aritmetiche. Dovremo accontentarci di realizzare le cosiddette “operazioni di macchina”. L'operazione di macchina associa a due numeri di macchina un terzo numero di macchina, ottenuto arrotondando (con la tecnica (i) o (ii)) l'esatto risultato dell'operazione aritmetica in questione.

Se con  $\bar{a} = \text{fl}(a)$  indichiamo l'operazione di arrotondamento, in aritmetica floating-point, di  $a$  a numero di macchina  $\bar{a}$ <sup>(†)</sup>, e con  $\oplus$ ,  $\ominus$ ,  $\odot$ ,  $\oslash$  denotiamo le operazioni di macchina corrispondenti a quelle aritmetiche  $+$ ,  $-$ ,  $\cdot$ ,  $/$ , abbiamo

$$\begin{aligned}\bar{a} \oplus \bar{b} &= \text{fl}(\bar{a} + \bar{b}) = (\bar{a} + \bar{b})(1 + \varepsilon_1) \\ \bar{a} \ominus \bar{b} &= \text{fl}(\bar{a} - \bar{b}) = (\bar{a} - \bar{b})(1 + \varepsilon_2) \\ \bar{a} \odot \bar{b} &= \text{fl}(\bar{a} \cdot \bar{b}) = (\bar{a} \cdot \bar{b})(1 + \varepsilon_3) \\ \bar{a} \oslash \bar{b} &= \text{fl}(\bar{a}/\bar{b}) = (\bar{a}/\bar{b})(1 + \varepsilon_4)\end{aligned}$$

dove  $|\varepsilon_i| \leq \text{eps}$ . L'errore relativo introdotto dalle quattro operazioni aritmetiche di macchina, prescindendo dagli eventuali errori presenti nei due operandi  $\bar{a}$  e  $\bar{b}$ , non supera mai la precisione di macchina  $\text{eps}$ .

---

(†) Ricordiamo che  $\bar{a} = a(1 + \varepsilon)$ ,  $|\varepsilon| \leq \text{eps}$ .

Una domanda cruciale cui dobbiamo ora rispondere è senz’altro la seguente: per le operazioni di macchina valgono ancora le note proprietà (commutativa, associativa, distributiva, ecc.) delle quattro operazioni aritmetiche? Purtroppo la risposta è spesso negativa. In particolare, mentre la proprietà commutativa

$$\bar{a} \oplus \bar{b} = \bar{b} \oplus \bar{a}, \quad \bar{a} \odot \bar{b} = \bar{b} \odot \bar{a}$$

risulta ancora valida, le seguenti non lo sono più:

$$\begin{aligned}\bar{a} \oplus (\bar{b} \oplus \bar{c}) &= (\bar{a} \oplus \bar{b}) \oplus \bar{c}, & \bar{a} \odot (\bar{b} \odot \bar{c}) &= (\bar{a} \odot \bar{b}) \odot \bar{c} \\ \bar{a} \odot (\bar{b} \oplus \bar{c}) &= (\bar{a} \odot \bar{b}) \oplus (\bar{a} \odot \bar{c}) & & \\ (\bar{a} \odot \bar{b}) \odot \bar{b} &= \bar{a}, & (\bar{a} \odot \bar{b}) \odot \bar{b} &= \bar{a} \\ (\bar{a} \odot \bar{c}) \odot \bar{b} &= (\bar{a} \odot \bar{b}) \odot \bar{c}\end{aligned}$$

Un’ulteriore relazione anomala è la seguente:

$$\bar{a} \oplus \bar{b} = \bar{a} \quad \text{quando } 0 < |\bar{b}| \ll |\bar{a}|$$

Possiamo pertanto concludere con l’affermazione:

*le espressioni che sono equivalenti in aritmetica esatta non risultano generalmente tali nelle aritmetiche con precisione finita.*

Ciononostante due espressioni (non nulle) saranno definite “equivalenti” dal punto di vista del calcolo numerico quando, valutate in un calcolatore, forniscono risultati che differiscono per una tolleranza relativa dell’ordine della precisione di macchina.

▷ **Osservazione.** Anche se i dati iniziali e il risultato finale di un processo di calcolo sono numeri di macchina, o comunque arrotondabili a numeri di macchina, le operazioni intermedie possono dare origine a fenomeni di *overflow* (numeri con esponente  $q > M$ ) o di *underflow* (numeri con  $q < m$ ); quando ciò succede, i risultati di tali operazioni non sono rappresentabili ed il calcolatore invia una segnalazione di errore. ◁

### ► Esempio.

$$\begin{aligned}y &= \sqrt{a \odot b}, & a &= p_1 N^{q_1}, & b &= p_2 N^{q_2} \\ \text{overflow se } q_1 + q_2 &> M + 1 \\ \text{underflow se } q_1 + q_2 &< m - 1\end{aligned}$$



## 1.4 Cancellazione numerica

La conseguenza più grave della rappresentazione con precisione finita dei numeri reali è senza dubbio il fenomeno della cancellazione numerica, ovvero la *perdita di cifre significative* dovuta ad operazioni di sottrazione quando il risultato è più piccolo di ciascuno dei

due operandi; questo fenomeno si verifica quando i due operandi sono “quasi uguali”. Per meglio illustrare quanto accade, supponiamo di avere due numeri floating-point  $a = p_1 N^q$  e  $b = p_2 N^q$ , dove le mantisse  $p_1$  e  $p_2$ , pur avendo più di  $t$  cifre, sono rappresentabili solo con  $t$  cifre. Supponiamo inoltre che le prime tre cifre, per esempio, di  $p_1$  coincidano con le corrispondenti di  $p_2$ . Nella mantissa  $\bar{p}$  della differenza (normalizzata)  $\bar{a} \ominus \bar{b} = \bar{p} N^{q-3}$ , solamente le prime  $t - 3$  cifre provengono dalle mantisse  $p_1$  e  $p_2$ ; le restanti 3 (poste uguali a zero) non hanno alcun significato. Consideriamo il seguente esempio numerico:

$N = 10$ ,  $t = 6$  e tecnica di arrotondamento (ii)

$$\begin{array}{ll} p_1 = 0.147554326 & \bar{p}_1 = 0.147554 \\ p_2 = 0.147251742 & \bar{p}_2 = 0.147252 \end{array}$$

La mantissa  $\bar{p}$  della differenza di macchina, che si ottiene normalizzando il risultato di tale operazione, assume il valore:

$$\bar{p} = (\bar{p}_1 \ominus \bar{p}_2) \times 10^3 = 0.302000$$

mentre la vera mantissa del risultato (normalizzato) prodotto dall’operazione non di macchina è

$$p = (p_1 - p_2) \times 10^3 = 0.302584$$

L’operazione di sottrazione in sé, anche quella di macchina, non introduce alcuna perdita di precisione; come abbiamo visto nel paragrafo precedente, le quattro operazioni aritmetiche di macchina possono provocare un errore relativo che non supera mai la precisione di macchina. La perdita di cifre significative descritta nell’esempio precedente (dove peraltro la sottrazione di macchina non introduce alcun errore) ha la sua origine negli errori presenti nei due operandi; o meglio, l’operazione di sottrazione ha amplificato detti errori. Se i due operandi sono privi di errori, il risultato non presenta alcuna perdita di precisione e gli zeri finali aggiunti sono esatti.

Nei tre esempi che seguono descriviamo tre diverse situazioni che possono comportare una perdita di precisione, e presentiamo formulazioni alternative dove il fenomeno della cancellazione numerica è scomparso.

► **Esempio 1.1.** Di solito le radici dell’equazione di secondo grado

$$x^2 - 2ax + \varepsilon = 0$$

vengono determinate con le espressioni

$$x_1 = a + \sqrt{a^2 - \varepsilon}, \quad x_2 = a - \sqrt{a^2 - \varepsilon}$$

Quando  $|\varepsilon| \ll |a|$ , una delle due formule, e precisamente  $a - \text{sgn}(a)\sqrt{a^2 - \varepsilon}$  (†), com-

---

(†) Si ha

$$\text{sgn}(a) = \begin{cases} 1 & \text{se } a \geq 0 \\ -1 & \text{se } a < 0 \end{cases}$$

porta la differenza di due numeri quasi uguali. L'utilizzazione delle formule alternative

$$x_1 = a + \operatorname{sgn}(a)\sqrt{a^2 - \varepsilon}, \quad x_2 = \frac{\varepsilon}{x_1}$$

ci consente di evitare la possibile cancellazione numerica. ◀

Nei due successivi esempi quando  $|\delta| \ll |x|$  le espressioni a primo membro causano perdite di cifre significative. Per non avere tale perdita è sufficiente utilizzare i secondi membri.

► **Esempio 1.2.**

$$\sqrt{x + \delta} - \sqrt{x} = \frac{\delta}{\sqrt{x + \delta} + \sqrt{x}}$$

► **Esempio 1.3.**

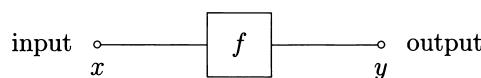
$$\cos(x + \delta) - \cos(x) = -2 \sin\left(\frac{\delta}{2}\right) \sin\left(x + \frac{\delta}{2}\right)$$

Purtroppo non sempre è possibile superare l'ostacolo con una semplice manipolazione delle espressioni coinvolte. A volte, comunque si cerchi di aggirare la difficoltà, la sottrazione che provoca cancellazione numerica (nella stessa circostanza) si ripropone in un punto diverso del procedimento di calcolo. Ciò è quanto succede nei primi due esempi, quando  $a^2 \approx \varepsilon$  nel primo e  $x \approx -\delta$  nel secondo. In questi casi la cancellazione è insita nel problema in esame: qualunque sia l'espressione matematica che utilizziamo per rappresentarlo, essa si manifesterà nella stessa forma e con la stessa intensità.

Nel terzo esempio invece, l'espressione a secondo membro manifesta ancora il fenomeno della cancellazione numerica, ma in una situazione diversa da quella del primo membro: quando  $x \approx -\delta/2$ . Pertanto, utilizzando una delle due, potremo sempre eliminare la perdita di precisione. In questo caso la cancellazione numerica non è propria del problema, ma della particolare espressione utilizzata per rappresentarlo.

## 1.5 Problemi numerici e algoritmi

Per *problema numerico* intendiamo una descrizione chiara e non ambigua di una connessione funzionale tra i dati (input) del problema (che costituiscono le variabili indipendenti) e i risultati desiderati (output)



I dati  $x$  e i risultati  $y$  devono essere rappresentabili da vettori (di dimensione finita) di numeri. Spesso il problema numerico scaturisce da un'approssimazione o *discretizzazione* di un modello matematico, per esempio un'equazione differenziale, la cui soluzione è una funzione.

La connessione funzionale  $f$  può essere espressa sia in forma esplicita  $y = f(x)$ , che implicita  $f(x, y) = 0$ . Essa in generale ammetterà infinite rappresentazioni alternative, formalmente diverse ma equivalenti nell'aritmetica esatta, nel senso che tutte associano al dato  $x$  lo stesso risultato  $y$ .

Per *algoritmo* di un problema numerico intendiamo invece una descrizione completa e ben definita di operazioni che permetta di trasformare (in un numero finito di passi) ogni vettore di dati (permissibili)  $x$  nel corrispondente output  $y^*$ , non necessariamente uguale a  $y$ .

Ad ogni problema numerico possiamo associare più algoritmi che in generale, pur risultando equivalenti nell'aritmetica esatta, utilizzando l'aritmetica del calcolatore forniranno risultati con precisione diversa.

Il problema numerico è caratterizzato dalla sola connessione funzionale (nell'aritmetica esatta) esistente tra input ed output. In un algoritmo invece, ogni singolo passo (operazione) deve essere definito chiaramente ed inequivocabilmente, dai dati sino alla determinazione effettiva dei risultati, e l'aritmetica è quella di macchina.

► **Esempio.** La determinazione della radice reale più grande dell'equazione algebrica

$$a_0x^n + a_1x^{n-1} + \cdots + a_{n-1}x + a_n = 0 \quad n \text{ dispari}$$

con coefficienti reali  $a_0, a_1, \dots, a_n$ , costituisce un problema numerico. L'input è il vettore  $(a_0, a_1, \dots, a_n)$ , mentre l'output è costituito dalla radice cercata  $t$ , definita in modo隐式 dalloquazione precedente (con la condizione di essere la radice reale più grande).

Un algoritmo per la soluzione di questo problema numerico è, per esempio, il metodo delle tangenti (vedi cap. 6), integrato però da una regola per la scelta dell'approssimazione iniziale e da un criterio per l'arresto del processo iterativo. ◀

Nei capitoli che seguono parleremo di problemi e di algoritmi; in particolare esamineremo algoritmi per il calcolo della soluzione di problemi numerici spesso ottenuti *discretizzando* modelli matematici.

## 1.6 Condizionamento di un problema, stabilità numerica di un algoritmo

Per valutare la bontà della risposta fornita da un algoritmo quando viene utilizzato per la risoluzione di un problema numerico, che per semplicità supporremo rappresentato da un'espressione di tipo esplicito  $y = f(x)$ , è necessario conoscere anche la “reazione” di quest'ultima all'introduzione di perturbazioni  $\delta x$  nei dati iniziali  $x$ . Occorre cioè sapere come gli errori (inevitabili) presenti nei dati vengono propagati dal problema

(la funzione  $f$ ), in assenza di errori di calcolo e di rappresentazione dei numeri. Questo studio ci consente di stabilire la massima precisione raggiungibile da un “buon” algoritmo (implementato su di un elaboratore, e quindi operante con l’aritmetica di macchina). Infatti, poiché in generale non avremo i dati iniziali  $x$ , ma una loro approssimazione  $\bar{x} = x + \delta x$ , per esempio  $\bar{x} = \text{fl}(x)$ , la funzione  $f$  come risposta ci fornirà (nell’aritmetica con precisione infinita) il valore  $f(\bar{x})$ . Partendo con i dati  $\bar{x}$  l’obiettivo dell’algoritmo “diventa” quindi  $f(\bar{x})$  e non più  $f(x)$ .

Pertanto, per giudicare la bontà di un algoritmo proposto per il calcolo di  $f(x)$  dobbiamo confrontare la risposta  $y^*$ , fornita dall’algoritmo (nell’aritmetica di macchina), con  $f(\bar{x})$ . Diremo che l’algoritmo proposto è *numericamente stabile* quando la quantità<sup>(†)</sup>

$$\frac{\|f(\bar{x}) - y^*\|}{\|f(\bar{x})\|}, \quad f(\bar{x}) \neq 0$$

è dell’ordine di grandezza della precisione di macchina, e *instabile* altrimenti.

Il concetto di *stabilità numerica* di un algoritmo è stato introdotto in [1.2]. Esso tiene conto della sola propagazione degli errori di arrotondamento provocati dall’aritmetica di macchina. Tale stabilità viene da noi denominata *numerica* per distinguerla da altri concetti di stabilità che vedremo, per esempio, nel capitolo 8. Una tecnica generale per esaminare la stabilità numerica di un algoritmo è la cosiddetta *analisi all’indietro* degli errori (*backward error analysis*), introdotta in [1.1] con lo scopo di studiare la propagazione degli errori negli algoritmi di algebra lineare.

L’esame dell’errore relativo

$$\frac{\|f(x) - f(\bar{x})\|}{\|f(x)\|}, \quad f(x) \neq 0$$

quale funzione delle perturbazioni relative  $\|\delta x\|/\|x\|$  ( $x \neq 0$ ), introdotte nei dati iniziali  $x$ , conduce invece alla definizione del concetto di *condizionamento del problema*. Quando a piccole perturbazioni (relative) nei dati  $x$  corrispondono, nell’aritmetica esatta, perturbazioni (relative) su  $f(x)$  dello stesso ordine di grandezza, il problema  $y = f(x)$  è definito *ben condizionato*; altrimenti è detto *mal condizionato*. Ovviamente il condizionamento del problema è una funzione dei dati  $x$ : il problema può risultare ben condizionato per un certo insieme di dati e mal condizionato per un altro. Inoltre, il condizionamento non dipende dall’espressione scelta per rappresentare  $f(x)$ , poiché nell’aritmetica esatta le espressioni equivalenti assumono tutte lo stesso valore  $y$ .

Quando  $y = f(x) \in \mathbb{R}$  e la funzione  $f(x)$  è derivabile almeno due volte, con derivate seconde continue, rispetto alle variabili  $x_1, x_2, \dots, x_n$ , il comportamento del problema potrebbe essere studiato ricorrendo alla formula di Taylor

$$\begin{aligned} \delta y &= f(\bar{x}) - f(x) = \\ &= \frac{\partial f(x)}{\partial x_1} \delta x_1 + \frac{\partial f(x)}{\partial x_2} \delta x_2 + \cdots + \frac{\partial f(x)}{\partial x_n} \delta x_n + \text{termini ordine 2} \end{aligned}$$

---

<sup>(†)</sup>  $\|\cdot\|$  denota una norma di vettore; vedi pag. 31.

se  $y \neq 0$  e  $x_i \neq 0$  otteniamo

$$\left| \frac{\delta y}{y} \right| \leq \left( \left| \frac{\partial f(x)}{\partial x_1} \right| \left| \frac{x_1}{y} \right| \right) \left| \frac{\delta x_1}{x_1} \right| + \cdots + \left( \left| \frac{\partial f(x)}{\partial x_n} \right| \left| \frac{x_n}{y} \right| \right) \left| \frac{\delta x_n}{x_n} \right| + \text{termini ordine 2}$$

I fattori  $K_i = |\partial f(x)/\partial x_i| |x_i/y|$  rappresentano i coefficienti di amplificazione degli errori relativi  $|\delta x_i/x_i|$ . Più in generale, dato il problema

$$y = f(x), \quad x \in \mathbb{R}^n, \quad y \in \mathbb{R}^m$$

anziché considerare i singoli coefficienti di amplificazione, ottenibili per esempio sviluppando in serie di Taylor la funzione  $f(x)$ , potremmo conseguire delle disuguaglianze del tipo

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

e definire il fattore  $K = K(x)$  numero di condizionamento del problema. Osserviamo tuttavia che, causa le maggiorazioni effettuate, le indicazioni fornite da questi coefficienti possono risultare troppo pessimistiche.

Per lo studio del condizionamento di alcuni problemi classici del Calcolo Numerico, vedere il testo alle pagine 39, 130 e 181.

Il concetto di condizionamento è proprio del problema numerico (o della funzione  $f$ ). Poiché esso è associato all'aritmetica esatta, le espressioni alternative ma equivalenti della funzione  $f(x)$  hanno tutte lo stesso condizionamento. Pertanto esso non ha alcun legame né con gli errori di arrotondamento delle operazioni di macchina né con il particolare procedimento di calcolo seguito per determinare la risposta  $f(x)$ . Solo quando il problema è ben condizionato, e l'algoritmo proposto per la sua risoluzione è stabile, gli errori (relativi) introdotti nei dati e nelle operazioni di macchina non vengono amplificati sul risultato.

Dopo quanto è stato finora detto sul condizionamento di un problema e sulla stabilità di un algoritmo appare naturale porre la domanda seguente: quale delle quattro operazioni aritmetiche può provocare una perdita di precisione? Un semplice esame dell'errore relativo

$$\left| \frac{(x_1 \text{ op } x_2) - (\bar{x}_1 \text{ op } \bar{x}_2)}{x_1 \text{ op } x_2} \right| = \left| \frac{(x_1 \text{ op } x_2) - \{[x_1(1 + \varepsilon_1)] \text{ op } [x_2(1 + \varepsilon_2)]\}}{x_1 \text{ op } x_2} \right|$$

dove con ‘op’ denotiamo la generica operazione aritmetica esatta, ci consente di affermare che è la somma di due numeri di segno opposto a poter causare un’amplificazione degli errori presenti nei due operandi  $\bar{x}_1$  e  $\bar{x}_2$ . Infatti, mentre per il prodotto e la divisione abbiamo

$$\frac{x_1 x_2 - x_1(1 + \varepsilon_1) \cdot x_2(1 + \varepsilon_2)}{x_1 x_2} = -\varepsilon_1 - \varepsilon_2 - \varepsilon_1 \varepsilon_2 \cong -\varepsilon_1 - \varepsilon_2$$

e

$$\frac{\frac{x_1}{x_2} - \frac{x_1(1 + \varepsilon_1)}{x_2(1 + \varepsilon_2)}}{\frac{x_1}{x_2}} = \frac{-\varepsilon_1 + \varepsilon_2}{1 + \varepsilon_2} \cong -\varepsilon_1 + \varepsilon_2$$

nel caso della somma algebrica risulta

$$\frac{(x_1 + x_2) - [x_1(1 + \varepsilon_1) + x_2(1 + \varepsilon_2)]}{x_1 + x_2} = -\frac{x_1}{x_1 + x_2}\varepsilon_1 - \frac{x_2}{x_1 + x_2}\varepsilon_2$$

da cui segue

$$K_i = \left| \frac{x_i}{x_1 + x_2} \right| \rightarrow \infty \quad \text{quando} \quad x_1 + x_2 \rightarrow 0$$

Il mal condizionamento di una somma algebrica non è altro che una interpretazione alternativa del fenomeno della cancellazione numerica descritto nel paragrafo 1.4.

Concludiamo quest'ultimo paragrafo elencando gli errori principali che intervengono nella risoluzione numerica di un problema (numerico).

Consideriamo, per semplicità, un generico problema descritto da una funzione esplicita

$$y = f(x)$$

e supponiamo che nei dati sia presente un errore  $x - \bar{x}$ , al quale corrisponde un errore finale, dovuto unicamente alla funzione  $f(x)$  (condizionamento),

$$e_1 = f(x) - f(\bar{x})$$

Spesso  $f$  viene approssimata con una funzione più semplice  $f_1$  (algoritmo); in questo caso abbiamo un ulteriore errore, detto di troncamento o di *discretizzazione*,

$$e_2 = f(\bar{x}) - f_1(\bar{x})$$

Come sappiamo però, le operazioni eseguite dal calcolatore nella valutazione di  $f_1(\bar{x})$  non sono esatte; invece di  $f_1(\bar{x})$  otterremo un valore  $f_2(\bar{x})$ . L'errore

$$e_3 = f_1(\bar{x}) - f_2(\bar{x})$$

è dovuto alla propagazione degli errori di arrotondamento nel calcolo numerico di  $f_1(\bar{x})$ . Responsabile di  $e_3$  è la stabilità numerica dell'algoritmo.

Il risultato finale fornito dall'algoritmo è quindi  $f_2(\bar{x})$ , e l'errore complessivo ad esso associato è

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

Allo scopo di illustrare con un esempio molto semplice il significato dei precedenti errori, consideriamo il seguente problema numerico:

$$y = f(x) \equiv \sum_{k=0}^{\infty} \frac{\left(-\frac{1}{4}x^2\right)^k}{(k!)^2}, \quad x \in \mathbb{R}$$

Il legame funzionale tra il dato  $x$  e il risultato  $y$  è in questo caso rappresentato da una serie convergente. Poiché non possiamo sommare infiniti termini, siamo costretti ad approssimare la serie predetta con una sua ridotta che indichiamo con

$$f_1(x) = \sum_{k=0}^{N-1} \frac{\left(-\frac{1}{4}x^2\right)^k}{(k!)^2}$$

Essendo la serie a termini di segno alterno, l'errore di troncamento che così commettiamo risulterà (in valore assoluto) inferiore al valore assoluto del primo termine trascurato, che, per  $N$  sufficientemente grande, possiamo rendere (in aritmetica con precisione infinita) piccolo a piacere. In questo caso l'algoritmo di calcolo coinciderà con la sequenza di operazioni aritmetiche effettuate per determinare la somma finita  $f_1(x)$ . Tale sequenza non è unica. Per esempio, potremmo sommare i termini secondo l'ordine naturale, oppure sommare dapprima tutti i termini positivi e poi tutti i termini negativi. Se potessimo operare con precisione infinita tutti gli algoritmi fornirebbero lo stesso risultato finale; con precisione finita invece, in alcuni casi essi producono risultati sensibilmente diversi. Avremo pertanto:

$$\begin{aligned}\varepsilon_1 &= f(x) - f(\bar{x}) \\ \varepsilon_2 &= f(\bar{x}) - f_1(\bar{x}) \quad \text{con } |\varepsilon_2| < \frac{x^{2N}}{4^N(N!)^2} \\ \varepsilon_3 &= f_1(\bar{x}) - f_2(\bar{x})\end{aligned}$$

Se  $f(x)$  fosse stata rappresentata da una somma finita non avremmo avuto l'errore  $\varepsilon_2$ .

Dopo quanto è stato finora detto appare chiaro che il nostro obiettivo nei prossimi capitoli sarà soprattutto la costruzione, con errori di discretizzazione nulli oppure arbitrariamente piccoli, di algoritmi numericamente stabili.

**Conclusione.** In questi paragrafi abbiamo presentato una breve descrizione dell'aritmetica in un calcolatore e ricordato alcune conseguenze di quest'ultima nel calcolo numerico. L'esposizione è lungi dall'essere completa; per esempio, l'implementazione effettiva delle operazioni di macchina meriterebbe di essere illustrata con maggiori dettagli tecnici, necessariamente propri del calcolatore preso in esame. Ciò nonostante abbiamo preferito limitare questa prima parte, introduttiva al calcolo automatico, a quegli elementi che risulteranno essenziali nelle motivazioni delle problematiche dei capitoli successivi. Una descrizione più approfondita e completa richiederebbe necessariamente il coinvolgimento di altre tematiche (proprie delle componenti hardware e software di un calcolatore) e porterebbe inevitabilmente ad uno stravolgimento dello spirito che ha guidato la stesura di questo testo.

Il lettore è tuttavia invitato ad approfondire gli argomenti introdotti in questo capitolo, consultando, per esempio, i testi suggeriti nella bibliografia.

## Bibliografia

- [1.1] J. H. Wilkinson, *Rounding errors in algebraic processes*, Prentice-Hall, Englewood Cliffs, N.J., 1963.
- [1.2] F. L. Bauer, J. Heinhold, K. Samelson, R. Sauer, *Moderne Rechenanlagen*, Stuttgart, Teubner, 1965.
- [1.3] D. E. Knuth, *The art of computer programming*, vol. II, Addison-Wesley, 1969.
- [1.4] W. Gear, *Computer organization and programming*, McGraw-Hill, 1974.
- [1.5] P. H. Sterbenz, *Floating-point computation*, Prentice-Hall, 1974.
- [1.6] J. F. Cavanagh, *Digital computer arithmetic*, McGraw-Hill, 1983.
- [1.7] *IEEE Standards for Binary floating-point arithmetic*, ANSI/IEEE 754-1985, New York, 1985.
- [1.8] M. Overton, *Numerical computing with IEEE floating point arithmetic*, SIAM, Philadelphia, 2001.
- [1.9] P. Markstein, *The new IEEE-754 standard for floating point arithmetic*, Dagstuhl Seminar Proceedings: Numerical validation in current hardware architectures 2008, Schloss Dagstuhl - Leibnitz Center for Informatics, Wadern, 2008.

## Esercizi proposti

**1.1.** Scrivere le rappresentazioni floating-point normalizzate dei seguenti numeri decimali:

$$\begin{aligned} 125.375 \\ 0.0075343 \\ 1.47512 \times 10^2 \end{aligned}$$

**1.2.** Siano  $a = 0.97524372 \times 10^2$  e  $b = 0.2$  due numeri in base 10. Siano  $\bar{a} = 0.975239 \times 10^2$  e  $\bar{b} = 0.199999$  due loro approssimazioni. Quante cifre significative sono presenti in  $\bar{a}$  e  $\bar{b}$ ?

**1.3.** I seguenti numeri reali normalizzati vengono introdotti in un calcolatore operante con rappresentazione in base 10, aritmetica floating-point con mantissa di 4 cifre:

$$a = 0.4523 \times 10^4 \quad b = 0.2115 \times 10^{-3} \quad c = 0.2583 \times 10^1$$

Supponendo di arrotondare i risultati con la tecnica (ii), eseguire le seguenti operazioni aritmetiche e indicare gli errori presenti nei risultati:

(i) $a + b + c$	(iv) $a - b - c$
(ii) $a/c$	(v) $a \cdot b/c$
(iii) $a - b$	(vi) $b/c \cdot a$

**1.4.** I seguenti numeri reali sono introdotti in un calcolatore nel quale essi vengono rappresentati in aritmetica floating-point, con base  $N = 10$  e  $t = 7$  cifre riservate alla mantissa (tecnica di arrotondamento (i)):

$$\begin{aligned} a &= 123.54337624 \\ b &= 123.1111111 \end{aligned}$$

Determinare il risultato  $\bar{c} = \bar{a} \ominus \bar{b} = \text{fl}(\text{fl}(a) - \text{fl}(b))$ . Confrontare  $\bar{c}$  con  $c = a - b$  e dire quante sono le cifre significative presenti in  $\bar{c}$ .

Commentare la risposta.

**1.5.** Consideriamo un elaboratore con aritmetica floating-point con  $t = 8$  cifre decimali e tecnica di arrotondamento (i) di pagina 7. Dati i seguenti tre numeri di macchina

$$a = 0.23371258 \times 10^{-4}, \quad b = 0.33678429 \times 10^2, \quad c = -0.33677811 \times 10^2$$

calcolare le somme  $x = a \oplus (b \oplus c)$  e  $y = (a \oplus b) \oplus c$  e confrontare i due risultati ottenuti con la somma esatta  $s = a + b + c = 0.641371258 \times 10^{-3}$ .

Spiegare il diverso comportamento delle due somme di macchina.

**1.6.** Supponendo di lavorare in aritmetica floating-point con mantissa di 4 cifre (decimali), sommare i numeri seguenti prima in ordine ascendente (dal più piccolo al più grande) e poi in ordine discendente, arrotondando le somme parziali:

$$\begin{array}{lll} 0.1580 & 0.4298 \times 10^1 & 0.7767 \times 10^3 \\ 0.2653 & 0.6266 \times 10^2 & 0.7889 \times 10^3 \\ 0.2581 \times 10^1 & 0.7555 \times 10^2 & 0.8999 \times 10^4 \end{array}$$

Confrontare i risultati ottenuti con il valore esatto  $0.107101023 \times 10^5$ .

**1.7.** Denotiamo con  $F$  l'insieme di tutti i numeri floating-point di macchina. La precisione di macchina `eps` può venire definita nel modo seguente:

$$\text{eps} = \min\{\varepsilon \in F, \varepsilon > 0 : 1 \oplus \varepsilon > 1\}$$

Se il vostro calcolatore opera con un sistema di numerazione in base  $N = 2^n$ , `eps` dovrebbe essere individuata dall'algoritmo seguente:

- 1:  $\varepsilon \leftarrow 1$
- 2:  $\varepsilon \leftarrow \varepsilon/2$
- 3: **se**  $1 + \varepsilon > 1$  **Allora vai al punto 2**
- 4: **altrimenti**  $\text{eps} \leftarrow 2\varepsilon$ ; **stop**

Applicare l'algoritmo al vostro calcolatore e verificarne la correttezza.

**1.8.** Determinare le radici dell'equazione

$$x^2 - 0.4002x + 0.8 \times 10^{-4} = 0$$

usando un'aritmetica floating-point con 4 cifre decimali di mantissa.

**1.9.** Proporre un algoritmo per valutare  $y = \sqrt{a^2 + b^2}$  qualunque siano i numeri di macchina  $a$  e  $b$ , eliminando i possibili fenomeni di underflow o di overflow.

**1.10.** Generalizzare l'algoritmo proposto nell'esercizio precedente al fine di valutare la quantità

$$y = \left( \sum_{i=1}^n x_i^2 \right)^{\frac{1}{2}}$$

qualunque siano i numeri di macchina  $\{x_i\}$ .

**1.11.** Illustrare con un esempio numerico il fenomeno della cancellazione numerica, precisando qual è l'origine della perdita di precisione.

**1.12.** Supponendo  $0 \leq x \leq 2\pi$ , quali delle seguenti espressioni possono generare il fenomeno della cancellazione numerica?

- (i)  $y = 1 + \sin \frac{x}{2}$
- (ii)  $y = 1 + \cos \frac{x}{2}$
- (iii)  $y = \cos x + \sin x$

**1.13.** Valutare le seguenti funzioni in un generico punto  $x$

$$f(x) = \frac{x - \sin x}{\tan x}, \quad f(x) = x - \sqrt{x^2 - a}, \quad f(x) = \frac{1 - \cos x}{x^2}$$

evitando il possibile fenomeno di cancellazione numerica.

**1.14.** Supponiamo di utilizzare un calcolatore che opera con un'aritmetica floating-point in base  $N$  e con mantisse di  $t = 7$  cifre, per sommare 5 numeri reali non di macchina. Se almeno uno degli addendi ha una rappresentazione normalizzata con esponente  $q = 3$ , mentre la somma determinata dal calcolatore ha  $q = -2$ , qual è il numero massimo di cifre corrette che la mantissa della somma potrà avere?

**1.15.** La radice quadrata  $\pm(s + it)$  di un numero complesso  $x + iy$ , con  $y \neq 0$ , può essere calcolata con le formule

$$s = \pm \sqrt{\frac{x + \sqrt{x^2 + y^2}}{2}}$$

$$t = \frac{y}{2s}$$

Esaminare i casi  $x \geq 0$  e  $x < 0$  dal punto di vista della cancellazione numerica e modificare, se necessario, le formule in modo da rendere stabile il calcolo di  $s$  e  $t$ .

**1.16.** Data la funzione lineare  $f(x) = ax + b$ ,  $a \neq 0$  e  $b \neq 0$  numeri non di macchina, vogliamo approssimare  $f'(0)$  con il seguente rapporto incrementale

$$f[-h, h] = \frac{f(h) - f(-h)}{2h}, \quad h = 2^{-k}, \quad k > 0$$

operando con aritmetica binaria floating-point a  $t$  cifre (per la mantissa).

Dare una maggiorazione dell'errore relativo presente nel valore calcolato di  $f[-h, h]$  ed esaminare il comportamento di tale maggiorazione quando  $k \rightarrow \infty$ .

**1.17.** Studiare il condizionamento del seguente problema:

$$y = f(x, a) = x - \sqrt{x^2 - a}$$

**1.18.** Utilizzando la tecnica delle sviluppo in serie di Taylor, studiare il condizionamento dei problemi numerici definiti negli esempi 1.1 e 1.2 di p.12.

**1.19.** Si vuole risolvere un problema numerico il cui numero di condizionamento è uguale a  $1E10$ , avendo a disposizione un algoritmo perfettamente stabile e desiderando determinare la soluzione con tolleranza relativa non superiore a  $1E - 3$ . Con quale precisione devono essere inseriti i dati ed effettuati i calcoli? Motivare la risposta.

**1.20.** Fissati un valore della variabile  $x$  ed una tolleranza relativa  $\varepsilon > \text{eps}$ , costruire un algoritmo che consenta di ottenere un'approssimazione della quantità

$$J_0(x) = \sum_{k=0}^{\infty} \frac{\left(-\frac{1}{4}x^2\right)^k}{(k!)^2}$$

con errore relativo non superiore a  $\varepsilon$ .

# Capitolo 2

## Richiami sulle matrici

### 2.1 Preliminari

In questo capitolo ricordiamo brevemente quelle definizioni e proprietà delle matrici che risultano indispensabili per poter affrontare lo studio dei capitoli successivi.

Per *matrice* intendiamo una tabella rettangolare ordinata di numeri, in generale complessi, che chiamiamo *elementi* della matrice. Generalmente per indicare una matrice useremo una notazione del tipo

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}$$

Ogni matrice è composta da righe e da colonne; per esempio, quella da noi considerata ha  $m$  righe ed  $n$  colonne. In questo caso diciamo che la matrice  $A$  è di tipo  $(m, n)$ , oppure, se per esempio gli elementi di  $A$  sono reali, che  $A \in \mathbb{R}^{m \times n}$ <sup>(†)</sup>. Con  $a_{ij} = (A)_{ij}$  denotiamo l'elemento di  $A$  situato all'intersezione della  $i$ -esima riga con la  $j$ -esima colonna. La matrice  $A$  si riduce ad un *vettore riga* quando  $m = 1$ , e ad un *vettore colonna* quando  $n = 1$ .

Spesso conviene considerare la matrice  $A$  come tabella i cui elementi sono a loro volta sottomatrici della tabella iniziale di numeri. Per esempio  $A$  potrebbe venire rappresentata in una delle forme seguenti:

$$A = \begin{pmatrix} a_{11} & b \\ a & A_1 \end{pmatrix} = (a_1, a_2, \dots, a_n) = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}$$

---

<sup>(†)</sup> Con  $\mathbb{R}^n$  indichiamo l'usuale spazio vettoriale reale di dimensione  $n$ .

dove

$$a = \begin{pmatrix} a_{21} \\ a_{31} \\ \vdots \\ a_{m1} \end{pmatrix} \quad a_i = \begin{pmatrix} a_{1i} \\ a_{2i} \\ \vdots \\ a_{mi} \end{pmatrix} \quad b = (a_{12}, a_{13}, \dots, a_{1n}) \quad b_i = (a_{i1}, a_{i2}, \dots, a_{in})$$

e

$$A_1 = \begin{pmatrix} a_{22} & \dots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{m2} & \dots & a_{mn} \end{pmatrix}$$

## 2.2 Operazioni tra matrici

Le operazioni somma e sottrazione sono definite *solo* tra matrici dello stesso tipo. Se  $A$  e  $B$  sono due matrici di tipo  $(m, n)$  allora la matrice  $C = A \pm B$  è ancora di tipo  $(m, n)$  ed è così definita:

$$(C)_{ij} = (A)_{ij} \pm (B)_{ij}$$

Dato un numero  $\lambda$ , reale o complesso, possiamo definire la nuova matrice  $\lambda A$ :

$$(\lambda A)_{ij} = \lambda (A)_{ij}$$

Sia  $A$  una matrice di tipo  $(m, p)$  e  $B$  una matrice di tipo  $(p, n)$ ; la matrice prodotto  $C = AB$ , di tipo  $(m, n)$ , è definita nel modo seguente:

$$(C)_{ij} = \sum_{k=1}^p (A)_{ik} (B)_{kj}$$

Osserviamo che il prodotto è definito solamente quando il numero di colonne di  $A$  è uguale al numero di righe di  $B$ .

In generale  $AB \neq BA$ , cioè la proprietà commutativa del prodotto non è valida; anzi, il prodotto  $BA$  può non essere definito, o se lo è, può dare una matrice di tipo diverso. Anche nel caso in cui  $A$  e  $B$  sono *quadrate*, cioè  $m = p = n$ , il prodotto non risulta commutativo:

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \neq \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

Ad ogni trasformazione lineare  $f : \mathbb{C}^n \rightarrow \mathbb{C}^m$ <sup>(†)</sup>, cioè tale che  $f(\alpha x + \beta y) = \alpha f(x) + \beta f(y)$  per ogni coppia  $x, y \in \mathbb{C}^n$  e qualunque siano i coefficienti  $\alpha, \beta \in \mathbb{C}$ , possiamo associare un'unica matrice  $A_f \in \mathbb{C}^{m \times n}$  tale che  $f(x) = A_f x$  per ogni  $x \in \mathbb{C}^n$ . Viceversa, se  $A_f \in \mathbb{C}^{m \times n}$  allora la funzione  $f(x) = A_f x$  è una trasformazione lineare da  $\mathbb{C}^n$  a  $\mathbb{C}^m$ .

---

(†) Con  $\mathbb{C}^n$  indichiamo l'usuale spazio vettoriale complesso di dimensione  $n$ .

Date le matrici  $A, B, C, D, E, F$ , su cui siano definibili le operazioni in esame, e due costanti arbitrarie, reali o complesse,  $\lambda$  e  $\mu$ , le seguenti proprietà sono valide:

$$\begin{aligned} A + B &= B + A \\ (A + B) + C &= A + (B + C) \\ (\lambda\mu)A &= \lambda(\mu A) \\ (\lambda + \mu)A &= \lambda A + \mu A \\ (DE)F &= D(EF) \\ D(E + F) &= DE + DF \\ (D + E)F &= DF + EF \\ \lambda(DE) &= (\lambda D)E = D(\lambda E) \end{aligned}$$

La matrice  $O$  i cui elementi sono tutti nulli viene chiamata *matrice nulla*. Per ogni matrice  $A$  esiste una ed una sola matrice, che denotiamo con  $-A$ , tale che  $A + (-A) = O$ .

Le proprietà riguardanti la somma di matrici e il prodotto di uno scalare per una matrice ci consentono di affermare che l'insieme di tutte le matrici dello stesso tipo,  $A \in \mathbb{C}^{m \times n}$  per esempio, con le operazioni di somma e di prodotto per uno scalare, costituisce uno spazio vettoriale. Le  $m \times n$  matrici  $E_{ij}$ , i cui elementi sono tutti nulli tranne quello che si trova in posizione  $(i, j)$  che assume il valore 1, formano una base di tale spazio. Per ogni  $A \in \mathbb{C}^{m \times n}$  abbiamo

$$A = \sum_{i=1}^m \sum_{j=1}^n a_{ij} E_{ij}$$

La dimensione di questo spazio è ovviamente  $m \times n$ .

Il prodotto di due matrici non nulle può dare come risultato una matrice nulla:

$$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

Ciò significa che l'uguaglianza

$$AX = AY \quad \text{con } A \neq O$$

non implica

$$X = Y$$

La matrice quadrata di ordine  $n$

$$I = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix} \quad (I)_{ij} = \delta_{ij}(\dagger)$$

è tale che  $IA = AI = A$  per ogni  $A \in \mathbb{C}^{m \times n}$ ; essa viene chiamata *matrice unità o identità*.

Definiamo ora la *trasposta* di una matrice  $A$ , che denotiamo con  $A^T$ :

$$(A^T)_{ij} = (A)_{ji}$$

Le righe di  $A^T$  coincidono con le colonne di  $A$ . Inoltre,

$$(A^T)^T = A \quad (\lambda A)^T = \lambda A^T \quad (A + B)^T = A^T + B^T \quad (AB)^T = B^T A^T$$

Con  $A^*$  indichiamo la matrice coniugata di  $A$ , ossia quella matrice i cui elementi sono i coniugati dei corrispondenti elementi di  $A$ . Ovviamente, quando  $A$  è reale abbiamo  $A^* = A$ .

Infine introduciamo l'operazione

$$A^H = (A^*)^T = (A^T)^*$$

Le operazioni predette, e in particolare quelle di somma e di prodotto di matrici, possono essere facilmente estese alle matrici cosiddette “a blocchi”, ovvero a matrici i cui elementi sono a loro volta matrici, purché siano rispettate le dimensioni. Per esempio, date le due matrici a blocchi

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \quad \text{e} \quad B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}$$

la somma

$$A + B = \begin{pmatrix} A_{11} + B_{11} & A_{12} + B_{12} \\ A_{21} + B_{21} & A_{22} + B_{22} \end{pmatrix}$$

è definita solo se le matrici  $A_{ij}$  e  $B_{ij}$  hanno le stesse dimensioni (cioè sono dello stesso tipo), mentre il prodotto

$$AB : (AB)_{ij} = \sum_{k=1}^2 A_{ik} B_{kj}$$

è definito solo se il numero di colonne di ogni  $A_{ik}$  è uguale al numero di righe della corrispondente  $B_{kj}$ .

Ricordiamo infine l'estensione della definizione di matrice trasposta:

$$A^T = \begin{pmatrix} A_{11}^T & A_{21}^T \\ A_{12}^T & A_{22}^T \end{pmatrix}$$

---

(†) Il simbolo di Kronecker  $\delta_{ij}$  ha il seguente significato:

$$\delta_{ij} = \begin{cases} 0 & \text{per } i \neq j \\ 1 & \text{per } i = j \end{cases}$$

## 2.3 Matrici con proprietà particolari

Ricordiamo dapprima le seguenti definizioni:

$$\text{se } \begin{cases} A^H = A \\ A^T = A \\ A^H = -A \\ A^T = -A \end{cases} \quad A \text{ viene detta} \quad \begin{cases} \text{hermitiana} \\ \text{simmetrica} \\ \text{antihermitiana} \\ \text{antisimmetrica} \end{cases}$$

Elenchiamo poi alcune forme speciali di matrici che incontreremo in seguito:

$\begin{pmatrix} & & 0 \\ & & \\ 0 & & \end{pmatrix} \quad a_{ij} = 0 \quad i \neq j$ <p>diagonale</p>	$\begin{pmatrix} & & \\ & & \\ & & 0 \\ & & \\ & & \end{pmatrix} \quad a_{ij} = 0 \quad i > j$ <p>triangolare superiore</p>	$\begin{pmatrix} & & \\ & & \\ & & 0 \\ & & \\ & & \end{pmatrix} \quad a_{ij} = 0 \quad i < j$ <p>triangolare inferiore</p>
$\begin{pmatrix} & & \\ & & \\ & & 0 \\ & & \\ & & \end{pmatrix} \quad a_{ij} = 0 \quad  i - j  > 1$ <p>tridiagonale</p>	$\begin{pmatrix} & & \\ & & \\ & & 0 \\ & & \\ & & \end{pmatrix} \quad a_{ij} = 0 \quad  i - j  > k$ <p>a banda di ampiezza <math>2k + 1</math></p>	
$\begin{pmatrix} & & \\ & & \\ & & 0 \\ & & \\ & & \end{pmatrix} \quad a_{ij} = 0 \quad i > j + 1$ <p>Hessenberg superiore</p>	$\begin{pmatrix} & & \\ & & \\ & & 0 \\ & & \\ & & \end{pmatrix} \quad a_{ij} = 0 \quad j > i + 1$ <p>Hessenberg inferiore</p>	

Nei capitoli successivi tratteremo esclusivamente matrici reali quadrate; anche se quanto esporremo risulta valido nella situazione più generale delle matrici complesse, eventualmente sostituendo  $A^T$  con  $A^H$ , noi limiteremo la nostra descrizione al caso reale. L'estensione dei risultati al caso complesso non dovrebbe presentare difficoltà.

Una matrice simmetrica  $A \in \mathbb{R}^{n \times n}$  è *definita positiva* se

$$x^T Ax > 0$$

per ogni vettore non nullo  $x \in \mathbb{R}^n$ . È *semidefinita positiva* se

$$x^T Ax \geq 0$$

Spesso la quantità  $x^T Ax$  ha un significato fisico ben preciso, quale ad esempio quello di una energia cinetica, oppure di una energia di deformazione. In questi casi è ovviamente noto a priori che la quantità in questione è sempre positiva. Altrimenti esistono dei

criteri numerici che permettono di stabilire se una data matrice  $A$  è definita positiva o meno. Ne elenchiamo alcuni, senza riportare le relative dimostrazioni.

**Criterio di Sylvester.** *Una matrice simmetrica  $A \in \mathbb{R}^{n \times n}$  è definita positiva se e solo se*

$$\det(A_k) > 0 \quad k = 1, \dots, n$$

*dove  $\det(A_k)$  rappresenta il determinante della matrice di ordine  $k$  formata dalle intersezioni delle prime  $k$  righe e  $k$  colonne di  $A$ .*

Ricordiamo che ad ogni matrice quadrata  $A \in \mathbb{R}^{n \times n}$  possiamo associare un numero reale, che chiamiamo determinante e che denotiamo con  $\det(A)$ . Le definizioni e le proprietà relative ai determinanti dovrebbero essere già note al lettore; pertanto qui riportiamo solo la formula di Binet per il determinante del prodotto di matrici  $C = AB$ :

$$\det(C) = \det(A) \det(B)$$

**Corollario 2.1.** *Gli elementi diagonale di una matrice simmetrica definitiva positiva sono positivi.*

**Corollario 2.2.** *Se  $A \in \mathbb{R}^{n \times n}$  è simmetrica definita positiva allora*

$$|a_{ij}|^2 < a_{ii}a_{jj} \quad i \neq j$$

*inoltre, l'elemento di  $A$  di modulo massimo giace sulla diagonale.*

I risultati contenuti nei due precedenti corollari scaturiscono da due particolari applicazioni del criterio di Sylvester. Lasciamo la verifica al lettore.

Una matrice  $A \in \mathbb{R}^{n \times n}$  è detta *a diagonale dominante per righe*, o semplicemente *a diagonale dominante*, se

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

e *a diagonale dominante per colonne* quando

$$|a_{jj}| > \sum_{\substack{i=1 \\ i \neq j}}^n |a_{ij}| \quad j = 1, \dots, n$$

È possibile dimostrare che una matrice simmetrica e a diagonale dominante, con elementi diagonale tutti positivi, è necessariamente definita positiva.

## 2.4 Matrici non singolari

Supponiamo di avere  $k$  vettori (riga o colonna)  $a_1, a_2, \dots, a_k$ , ognuno con  $n$  componenti. Se la relazione

$$\alpha_1 a_1 + \alpha_2 a_2 + \cdots + \alpha_k a_k = o$$

con  $\alpha_i$  costanti arbitrarie, è soddisfatta solo quando  $\alpha_1 = \alpha_2 = \dots = \alpha_k = 0$  allora diciamo che i vettori  $a_1, a_2, \dots, a_k$  sono *linearmente indipendenti*.

Data una matrice  $A \in \mathbb{R}^{n \times n}$ , quando  $\det(A) \neq 0$  i vettori colonna (riga) risultano linearmente indipendenti, e viceversa. Pertanto,  $\det(A) = 0$  se e solo se i vettori colonna (riga) sono linearmente dipendenti, cioè se e solo se almeno una delle colonne (righe) può essere espressa come combinazione lineare delle altre.

Quando  $\det(A) \neq 0$  ( $\det(A) = 0$ ) la matrice  $A$  è detta *non singolare* (*singolare*).

Se  $\det(A) \neq 0$  allora esiste una ed una sola matrice (non singolare)  $B \in \mathbb{R}^{n \times n}$  tale che

$$AB = BA = I$$

La matrice  $B$ , che denoteremo sempre con il simbolo  $A^{-1}$ , viene chiamata *matrice inversa* di  $A$ .

Prese due matrici non singolari  $C, D \in \mathbb{R}^{n \times n}$ , le seguenti regole risultano valide:

$$\begin{aligned}(CD)^{-1} &= D^{-1}C^{-1} \\ (C^T)^{-1} &= (C^{-1})^T\end{aligned}$$

È possibile dimostrare che una matrice  $A \in \mathbb{R}^{n \times n}$  a diagonale dominante, per righe oppure per colonne, è necessariamente non singolare. Anche quando  $A$  è simmetrica definita positiva esiste la matrice inversa  $A^{-1}$  ed è anch'essa simmetrica definita positiva.

Consideriamo ora una generica matrice  $A$  di ordine  $n$ . Per ogni intero  $p = 0, 1, \dots$  esaminiamo tutte le sottomatrici di ordine  $n - p$  ottenute eliminando da  $A$   $p$  righe e  $p$  colonne in ogni modo possibile. Se per  $p = 0, \dots, k$  tutti i determinanti associati a tali matrici sono nulli, e se esiste almeno una sottomatrice di ordine  $n - k - 1$  non singolare, allora diciamo che la matrice  $A$  ha *rango*  $r = n - k - 1$ . Quando  $\det(A) \neq 0$  il rango è  $n$ .

Se il rango di  $A$  è  $r$ , tra gli  $n$  vettori colonna (riga) di  $A$  solamente  $r$  di essi sono linearmente indipendenti. Il rango di una matrice coincide quindi con il numero di vettori colonna (riga) linearmente indipendenti.

Dati due vettori (colonna)  $x, y \in \mathbb{R}^n$ ,

$$x = (x_1, x_2, \dots, x_n)^T \quad y = (y_1, y_2, \dots, y_n)^T$$

e definito il loro *prodotto scalare*

$$x^T y = \sum_{i=1}^n x_i y_i$$

diciamo che  $x$  e  $y$  sono tra di loro *ortogonali* quando

$$x^T y = 0$$

Un sistema di vettori  $\{a_1, a_2, \dots, a_k\}$ ,  $a_i \in \mathbb{R}^n$ , è detto *ortonormale* quando

$$a_i^T a_j = \delta_{ij}$$

La matrice  $A \in \mathbb{R}^{n \times n}$  è definita *ortogonale* se le sue colonne (righe) formano un sistema ortonormale; in questo caso

$$A^T A = A A^T = I$$

cioè  $A^{-1} = A^T$ . Se inoltre  $A$  è simmetrica abbiamo  $A^{-1} = A$  e la matrice viene chiamata *involutiva*.

Quando i vettori  $x$  e  $y$  sono complessi, il loro prodotto scalare viene definito dalla quantità (in generale complessa)

$$x^H y = \sum_{i=1}^n x_i^* y_i$$

Inoltre, essi risultano ortogonali tra di loro se  $x^H y = 0$ .

Ricordiamo infine che una matrice complessa  $A$  con colonne (righe) ortonormali viene definita *unitaria* e

$$A^H A = A A^H = I$$

## 2.5 Autovalori di una matrice

Un numero  $\lambda$ , reale o complesso, è detto *autovalore* di una matrice (quadrata)  $A$  quando esiste un vettore, reale o complesso, non nullo  $x$  tale che

$$Ax = \lambda x$$

Il vettore  $x$  viene chiamato *autovettore* di  $A$  corrispondente all'autovalore  $\lambda$ . Ogni autovettore è definito a meno di una costante moltiplicativa.

Dalla definizione precedente non è difficile osservare che  $\lambda$  è un autovalore di  $A$  se e solo se la matrice  $A - \lambda I$  è singolare. Gli autovalori di  $A$  coincidono pertanto con le  $n$  radici dell'*equazione caratteristica* (algebrica di grado  $n$ )

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

Siano  $x_1, x_2, \dots, x_k$  autovettori di  $A$  corrispondenti a  $k$  autovalori  $\lambda_1, \lambda_2, \dots, \lambda_k$  distinti; i vettori  $\{x_i\}$  sono necessariamente linearmente indipendenti. Gli autovettori linearmente indipendenti corrispondenti ad un autovalore  $\lambda$  di  $A$  sono esattamente  $n - r$ , dove  $r$  è il rango della matrice  $A - \lambda I$ .

Quando  $A \in \mathbb{R}^{n \times n}$  è simmetrica, i suoi autovalori sono reali e i corrispondenti autovettori (anch'essi reali) formano un sistema ortogonale. Se inoltre  $A$  è definita positiva, gli autovalori risultano positivi; infatti,

$$\lambda = \frac{x^H A x}{x^H x} = \frac{x^T A x}{x^T x} > 0$$

La definizione stessa di autovalore-autovettore

$$Ax_i = \lambda_i x_i \quad i = 1, \dots, n$$

ci consente di scrivere l'identità

$$(Ax_1, Ax_2, \dots, Ax_n) = (\lambda_1 x_1, \lambda_2 x_2, \dots, \lambda_n x_n)$$

ovvero

$$AX = X\Lambda$$

con

$$X = (x_1, x_2, \dots, x_n) \quad \text{e} \quad \Lambda = \begin{pmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_n \end{pmatrix}$$

Inoltre, quando gli autovettori  $x_1, x_2, \dots, x_n$  di  $A$  sono linearmente indipendenti abbiamo

$$X^{-1}AX = \Lambda$$

In questo caso diciamo che la matrice  $A$  è *diagonalizzabile*.

Una matrice con autovalori distinti è certamente diagonalizzabile.

Ricordiamo infine il concetto di raggio spettrale di una matrice (quadrata)  $A$ , che denotiamo con  $\rho(A)$ :

$$\rho(A) = \max\{|\lambda_i| : \lambda_i \text{ è autovalore di } A\}$$

## 2.6 Norme di vettore e di matrice

Una norma di vettore su  $\mathbb{R}^n$  è una funzione che ad ogni vettore  $x \in \mathbb{R}^n$  associa un numero reale, che indichiamo con  $\|x\|$ , con le seguenti proprietà:

- (i)  $\|x\| > 0$  per ogni  $x \neq o$  e  $\|x\| = 0$  se e solo se  $x = o$ ;
- (ii)  $\|cx\| = |c| \|x\|$  qualunque sia  $c \in \mathbb{R}$ ;
- (iii)  $\|x + y\| \leq \|x\| + \|y\|$ .

Le relazioni (ii) e (iii) ci permettono di dedurre che per ogni norma vale la disegualanza

$$\|x - y\| \geq |\|x\| - \|y\||, \quad \forall x, y \in \mathbb{R}^n$$

Ogni norma su  $\mathbb{R}^n$  è una funzione uniformemente continua. Inoltre, poiché si suppone  $n$  fisso, tutte le norme sono equivalenti, nel senso che per ogni coppia di norme  $\|\cdot\|^{(1)}$  e  $\|\cdot\|^{(2)}$  esistono due costanti positive  $m$  e  $M$  tali che

$$m\|x\|^{(2)} \leq \|x\|^{(1)} \leq M\|x\|^{(2)}$$

per tutti i vettori  $x \in \mathbb{R}^n$ .

Le norme più frequentemente usate sono le seguenti:

(a)  $\|x\|_\infty = \max_{1 \leq i \leq n} |x_i|$  norma infinito,

(b)  $\|x\|_1 = \sum_{i=1}^n |x_i|$

(c)  $\|x\|_2 = \left( \sum_{i=1}^n |x_i|^2 \right)^{\frac{1}{2}} = \sqrt{x^T x}$ <sup>(†)</sup> norma euclidea.

Esse sono casi particolari della più generale norma, chiamata *norma p*, definita dalla relazione

$$\|x\|_p = \left( \sum_{i=1}^n |x_i|^p \right)^{\frac{1}{p}}, \quad p \geq 1$$

Infatti la (a) corrisponde al caso limite  $p = \infty$ , la (b) a  $p = 1$  e la (c) a  $p = 2$ .

Noi useremo quasi sempre la norma euclidea, per la quale risulta valida la disegualanza di Cauchy-Schwarz

$$|x^T y| \leq \|x\|_2 \|y\|_2$$

Inoltre, quando  $x$  e  $y$  sono ortogonali, cioè  $x^T y = 0$ ,abbiamo

$$\|x + y\|_2^2 = \|x\|_2^2 + \|y\|_2^2$$

Nel capitolo successivo (pagina 78) utilizzeremo tuttavia anche una norma di vettore diversa dalle precedenti, definita dalla relazione:

$$\|x\|_A = \sqrt{x^T A x}$$

dove  $A \in \mathbb{R}^{n \times n}$  è una matrice simmetrica definita positiva. Nel caso particolare  $A \equiv I$  otteniamo la norma euclidea (c). Essa può quindi essere interpretata come una generalizzazione di quest'ultima.

La definizione di norma di matrice<sup>(††)</sup> su  $\mathbb{R}^{n \times n}$  è analoga a quella di vettore: è una funzione di  $\mathbb{R}^{n \times n}$  in  $\mathbb{R}$ , che denotiamo sempre con  $\|\cdot\|$ , tale che

- (i)  $\|A\| > 0$  per ogni  $A \neq O$  e  $\|A\| = 0$  se e solo se  $A = O$ ;
- (ii)  $\|cA\| = |c| \|A\|$  qualunque sia il reale  $c$ ;
- (iii)  $\|A + B\| \leq \|A\| + \|B\|$ .

Nelle nostre applicazioni utilizzeremo esclusivamente norme che godono della ulteriore proprietà

- (iv)  $\|AB\| \leq \|A\| \|B\|$ .

---

<sup>(†)</sup> Nel caso di vettori complessi abbiamo  $\|x\|_2 = \sqrt{x^H x}$ .

<sup>(††)</sup> La definizione vale, ovviamente, anche per matrici non quadrate.

Anche la norma di matrice è uniformemente continua su  $\mathbb{R}^{n \times n}$ . Inoltre, per ogni coppia di norme  $\|\cdot\|$  e  $\|\cdot\|'$  esistono due costanti positive  $m$  e  $M$  tali che

$$m\|A\| \leq \|A\|' \leq M\|A\| \quad \forall A \in \mathbb{R}^{n \times n}$$

In questo senso tutte le norme di matrice su  $\mathbb{R}^{n \times n}$  sono equivalenti.

Le norme più note sono:

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

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

$$(c) \|A\|_F = \left( \sum_{i=1}^n \sum_{j=1}^n |a_{ij}|^2 \right)^{\frac{1}{2}} \quad \text{norma di Frobenius}$$

$$(d) \|A\|_2 = \sqrt{\rho(A^T A)} \quad \text{norma spettrale}$$

Poiché spesso abbiamo espressioni che coinvolgono sia matrici che vettori, è utile aggiungere alla definizione di norma di matrice una ulteriore condizione (detta di compatibilità) che permetta di introdurre un legame tra una norma di matrice e una di vettore. Pertanto, data una norma di matrice ed una di vettore, diciamo che le due norme sono *compatibili* se

$$\|Ax\| \leq \|A\| \|x\|$$

per ogni  $A \in \mathbb{R}^{n \times n}$  e  $x \in \mathbb{R}^n$ . Una norma di vettore può risultare compatibile con più norme di matrice.

Ad ogni norma di vettore possiamo associare una norma di matrice nel modo seguente:

$$\|A\| = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|} = \max_{\|x\|=1} \|Ax\|$$

Una norma così definita viene denominata *naturale*, o indotta da quella di vettore. Dalla definizione stessa risulta  $\|Ax\| \leq \|A\| \|x\|$ , e quindi la relazione di compatibilità tra le due norme è automaticamente soddisfatta. Per le norme naturaliabbiamo inoltre

$$\|I\| = \max_{\|x\|=1} \|Ix\| = 1$$

Le norme di matrice più note sono naturali? La risposta è data dai risultati seguenti:

$$\begin{array}{lll} \|x\|_1 & & \|A\|_1 \\ \|x\|_\infty & \text{induce} & \|A\|_\infty \\ \|x\|_2 & & \|A\|_2 \end{array}$$

La norma di Frobenius invece non è naturale in quanto  $\|I\|_F = n^{1/2}$ .

Infine ricordiamo che per ogni norma di matrice compatibile con una norma di vettore abbiamo

$$\rho(A) \leq \|A\|$$

## Bibliografia

- [2.1] A. S. Householder, *The theory of matrices in numerical analysis*, Blaisdell, Boston, 1964.
- [2.2] J. H. Wilkinson, *The algebraic eigenvalue problem*, Clarendon Press, Oxford, 1965.
- [2.3] S. Lang, *Algebra lineare*, Boringhieri, Torino, 1968.
- [2.4] G. W. Stewart, *Introduction to matrix computation*, Academic Press, New York, 1973.
- [2.5] G. Strang., *Algebra lineare e sue applicazioni*, Liguori Editore, Napoli, 1981.
- [2.6] P. Lancaster, M. Tismenetsky, *The theory of matrices*, Academic Press, New York, 1985.
- [2.7] J. M. Ortega, *Matrix theory, a second course*, Plenum Press, New York, 1987.

## Esercizi proposti

**2.1.** Calcolare i seguenti prodotti:

$$(X, Y)^T A(X, Y), \quad (\xi, x^T) \begin{pmatrix} \alpha & a^T \\ a & A \end{pmatrix} \begin{pmatrix} \xi \\ x \end{pmatrix}, \quad \begin{pmatrix} \alpha & a_2^T \\ a_1 & A \end{pmatrix} \begin{pmatrix} \beta & b_2^T \\ b_1 & B \end{pmatrix}$$

**2.2.** Definiamo le matrici

$$P_1 = \left( \begin{array}{ccc|cc} 1 & & & & 0 \\ & \ddots & & & \\ & & 1 & & \\ \hline & & & \lambda & \\ 0 & & & & 1 \\ \hline & & & i & \end{array} \right) i \quad P_2 = \left( \begin{array}{cc|cc|cc} 1 & & & & & 0 \\ & \ddots & & & & \\ & & 1 & & & \\ \hline & & & 0 & & 1 \\ & & & & 1 & \\ \hline & & & i & & j \\ & & & & & \ddots \\ & & & & & 1 \end{array} \right) j$$

$$P_3 = \left( \begin{array}{cc|cc} 1 & & & 0 \\ & \ddots & & \\ & & 1 & \\ \hline & & & \\ & & \lambda & 1 \\ \hline & & & \ddots \\ & & & 1 \\ & & i & j \end{array} \right)$$

Data una generica matrice quadrata  $A$  calcolare i prodotti  $P_i A$  e  $AP_i$ ,  $i = 1, 2, 3$ .

**2.3.** Sia data una matrice  $A$  di ordine  $n$  ed una diagonale

$$D = \begin{pmatrix} d_1 & & & 0 \\ & d_2 & & \\ & & \ddots & \\ 0 & & & d_n \end{pmatrix}$$

costruire i prodotti  $DA$  e  $AD$ .

**2.4.** Verificare che il prodotto di due matrici triangolari superiori (inferiori) è ancora una matrice triangolare superiore (inferiore).

**2.5.** Il prodotto di due matrici simmetriche non è necessariamente una matrice simmetrica. Costruire un esempio.

**2.6.** Dimostrare che la matrice  $B = A^T A$ ,  $A \in \mathbb{R}^{m \times n}$ , è simmetrica semidefinita positiva. La matrice  $B$  è non singolare se e solo se le colonne di  $A$  sono linearmente indipendenti; in questo caso  $B$  è definita positiva.

**2.7.** Calcolare l'inversa di una matrice diagonale. Quand'è che una matrice diagonale o triangolare risulta singolare? Verificare che l'inversa di una matrice triangolare superiore (inferiore) è ancora una matrice triangolare superiore (inferiore).

**2.8.** Dimostrare che se l'elemento  $a_{kk}$  di una matrice triangolare superiore (o inferiore)  $A \in \mathbb{R}^{n \times n}$  è nullo, allora le prime  $k$  colonne (o righe) di  $A$  risultano linearmente dipendenti.

**2.9.** Verificare che se  $A \in \mathbb{R}^{n \times n}$  è triangolare superiore e  $B \in \mathbb{R}^{n \times n}$  è di Hessenberg superiore, la matrice prodotto  $C = AB$  è di Hessenberg superiore.

**2.10.** Sia  $A$  simmetrica a diagonale dominante con tutti gli elementi diagonale positivi. Dimostrare che  $A$  è definita positiva.

**2.11.** Dimostrare che in una matrice simmetrica definita positiva, suddivisa a blocchi nella forma

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

con  $A_{11} \in \mathbb{R}^{p \times p}$  e  $A_{22} \in \mathbb{R}^{q \times q}$ , anche i blocchi diagonale  $A_{11}$  e  $A_{22}$  risultano simmetrici definiti positivi.

**2.12.** Verificare che una matrice ortogonale triangolare è necessariamente diagonale.

**2.13.** Dimostrare che la matrice  $A = I - 2ww^T$ , dove  $w$  è un vettore con  $\|w\|_2 = 1$ , è simmetrica e ortogonale.

**2.14.** Dimostrare che le matrici  $A$  e  $A^T$  hanno gli stessi autovalori.

**2.15.** Dimostrare che gli autovalori di  $A^{-1}$  coincidono con i reciproci degli autovalori di  $A$ .

**2.16.** Dimostrare che se la matrice  $U$  è ortogonale allora  $|\det(U)| = 1$  e gli autovalori di  $U$  hanno tutti modulo 1.

**2.17.** Sia  $U$  una matrice ortogonale. Dimostrare che:

$$\begin{aligned} \|UA\|_F &= \|AU\|_F = \|A\|_F \\ \|Ux\|_2 &= \|x\|_2 \text{ e } \|Ux - Uy\|_2 = \|x - y\|_2 \\ \|UA\|_2 &= \|AU\|_2 = \|A\|_2 \end{aligned}$$

**2.18.** Dimostrare che

$$\|A\|_2 \leq \sqrt{\|A\|_1 \|A\|_\infty}$$

**2.19.** Sia  $U \in \mathbb{R}^{m \times n}$  una matrice con colonne ortonormali, e  $V \in \mathbb{R}^{n \times n}$  una matrice ortogonale. Verificare che la matrice prodotto  $UV$  ha colonne ortonormali.

**2.20.** Date una successione di vettori  $\{x_n\}$  ed una norma di vettore  $\|\cdot\|$ , sia  $\|x_n\| \rightarrow 0$  per  $n \rightarrow \infty$ . Dimostrare che la convergenza a zero della successione delle norme è indipendente dalla norma scelta.

# Capitolo 3

## Sistemi lineari

### 3.1 Preliminari

Il problema della risoluzione di sistemi di equazioni lineari si presenta in moltissime applicazioni, sia esplicitamente nel modello matematico associato al fenomeno fisico in esame, sia come passo intermedio o finale nella risoluzione numerica del modello in questione, rappresentato, per esempio, da equazioni differenziali. Come vedremo nei capitoli successivi, la stessa costruzione di metodi numerici per il calcolo di autovalori di matrici, per la risoluzione di sistemi di equazioni non lineari, di equazioni differenziali, per l'approssimazione di dati e di funzioni, può richiedere la risoluzione di sistemi lineari.

In questo capitolo ci proponiamo pertanto di costruire metodi efficienti per la determinazione della soluzione di sistemi lineari di  $n$  equazioni in  $n$  incognite (il vettore  $x$ )

$$(3.1) \quad Ax = b, \quad A \in \mathbb{R}^{n \times n}, \quad b, x \in \mathbb{R}^n$$

È noto che il sistema (3.1) ammette soluzione se e solo se il vettore  $b$  appartiene allo spazio (lineare) generato dalle colonne di  $A$ . Infatti, posto  $A = (a_1, a_2, \dots, a_n)$ ,  $a_i \in \mathbb{R}^n$ , e  $x = (x_1, x_2, \dots, x_n)^T$ ,  $x_i \in \mathbb{R}$ , la scrittura  $Ax = b$  può essere interpretata nella forma

$$b = x_1 a_1 + x_2 a_2 + \cdots + x_n a_n$$

Inoltre, la soluzione è unica se e solo se la matrice  $A$  è non singolare, e in questo caso possiamo scrivere

$$x = A^{-1}b$$

Un metodo di risoluzione che certamente tutti conosciamo è la regola di Cramer:

$$x_i = \frac{\det(A_i)}{\det(A)}, \quad i = 1, 2, \dots, n$$

dove  $A_i$  denota la matrice ottenuta da  $A$  sostituendo la colonna  $i$ -esima con il vettore  $b$ . I determinanti coinvolti potrebbero essere calcolati utilizzando la regola di Laplace, scritta nella forma

$$\det(A) = \sum_{j=1}^n (-1)^{j+1} a_{1j} \det(A_{1j})$$

dove  $A_{1j}$  rappresenta la matrice di ordine  $n - 1$  ottenuta da  $A$  eliminando la prima riga e la  $j$ -esima colonna. Il numero di operazioni aritmetiche<sup>(†)</sup> richieste da questo metodo è superiore a  $(n + 1)!$ ; troppo elevato<sup>(††)</sup> perché possa competere con altri metodi che più avanti vedremo.

Anche l'uso dell'espressione  $x = A^{-1}b$  risulta troppo oneroso in termini di operazioni aritmetiche. Anzi, come vedremo alla fine di questo capitolo, la costruzione della matrice inversa  $A^{-1}$  nella sua essenza comporta la risoluzione del sistema stesso.

Di norma i metodi numerici per la risoluzione di sistemi lineari vengono suddivisi in due classi: metodi *diretti* e metodi *iterativi*. Con i metodi diretti l'esatta soluzione viene costruita, in assenza di errori di arrotondamento, in un numero finito di passi. Per sistemi  $Ax = b$  con matrici  $A$  *dense*, cioè con la maggior parte degli elementi  $(A)_{ij}$  non nulli, i metodi diretti sono di solito più efficienti.

I metodi iterativi invece sono generalmente utilizzati per la risoluzione di sistemi con matrici  $A$  *sparse*, cioè con molti elementi  $(A)_{ij}$  nulli, e di ordine elevato. Sistemi sparsi sono presenti in numerose applicazioni. A volte, come infatti succede nella risoluzione numerica di equazioni alle derivate parziali con metodi alle differenze finite, gli elementi non nulli costituiscono delle successioni regolari di pochi numeri distinti; per esempio,

$$A = \begin{pmatrix} 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 4 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 4 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 \end{pmatrix}$$

A causa dell'elevato ordine delle matrici coinvolte in problemi di questo tipo (da alcune migliaia sino a  $10^6$  e oltre) i metodi diretti non sempre sono utilizzabili. Infatti questi ultimi ottengono la soluzione  $x$  in un numero finito di passi mediante una successione (finita) di trasformazioni del problema iniziale in problemi equivalenti, cioè con la stessa

(†) D'ora in avanti per operazione aritmetica intendiamo la coppia somma-prodotto.

(††) Ricordiamo che  $10! \cong 3.6 \times 10^6$ ,  $20! \cong 2.4 \times 10^{18}$ ,  $50! \cong 3.0 \times 10^{64}$ .

soluzione  $x$ , ma con matrici dei coefficienti diverse; anzi, con il procedere del metodo di risoluzione il numero di elementi non nulli presenti in queste matrici generalmente cresce, e può ben presto saturare lo spazio disponibile nella memoria centrale del calcolatore. In questi casi è utile, e spesso indispensabile, utilizzare metodi iterativi, i quali, operando sempre e solo con gli elementi della matrice iniziale  $A$ , generano una successione infinita di vettori convergenti, sotto opportune condizioni, alla soluzione cercata. Poiché il processo iterativo lascia inalterata la matrice  $A$ , è sufficiente memorizzare gli elementi non nulli di  $A$ .

Osserviamo infine che, mentre i metodi diretti in assenza di errori nella rappresentazione dei numeri forniscono la soluzione esatta del sistema, indipendentemente da quella che potrebbe essere la precisione richiesta dall'utente, i metodi iterativi, se convergenti, consentono invece di arrestare il processo iterativo non appena la precisione desiderata è stata raggiunta. Va tuttavia sottolineato che un metodo iterativo risulterà efficiente solo se consentirà di raggiungere la precisione desiderata con un numero accettabile di iterazioni, ovvero di operazioni aritmetiche.

Prima di illustrare alcuni dei metodi più importanti per la risoluzione di sistemi lineari, esaminiamo il condizionamento del problema.

Per semplicità supponiamo di introdurre nel solo termine noto  $b$  una perturbazione  $\delta b$ . La soluzione che troveremo, supponendo tutte le operazioni aritmetiche fatte esattamente e i numeri rappresentabili con infinite cifre, non sarà quella del problema iniziale, cioè  $x$ , bensì  $\bar{x} = x + \delta x$ ; avremo allora

$$A(x + \delta x) = b + \delta b$$

da cui segue, ricordando che  $Ax = b$ ,

$$A\delta x = \delta b \quad \text{e} \quad \delta x = A^{-1}\delta b$$

Dopo aver scelto una norma di vettore ed una di matrice ad essa compatibile, scriviamo

$$(3.2) \quad \|\delta x\| = \|A^{-1}\delta b\| \leq \|A^{-1}\| \|\delta b\|$$

La (3.2) e la diseguaglianza

$$\|b\| = \|Ax\| \leq \|A\| \|x\|$$

ci consentono infine di scrivere

$$\frac{\|\delta x\|}{\|A\| \|x\|} \leq \|A^{-1}\| \frac{\|\delta b\|}{\|b\|}$$

e quindi

$$(3.3) \quad \frac{\|\delta x\|}{\|x\|} \leq \|A\| \|A^{-1}\| \frac{\|\delta b\|}{\|b\|}$$

Se supponiamo che anche la matrice  $A$  subisca una perturbazione  $\delta A$ , l'esame dell'effetto delle perturbazioni  $\delta b$  e  $\delta A$  sulla soluzione  $x$  diventa ben più complesso; anche

perché la matrice perturbata  $A + \delta A$  potrebbe risultare singolare. Tuttavia quando  $A + \delta A$  è nonsingolare e  $\|\delta A\| < 1/\|A^{-1}\|$  possiamo stabilire la seguente maggiorazione ([3.20, pag. 137])

$$(3.4) \quad \frac{\|\delta x\|}{\|x\|} \leq \frac{K(A)}{1 - K(A) \frac{\|\delta A\|}{\|A\|}} \left( \frac{\|\delta A\|}{\|A\|} + \frac{\|b\|}{\|A\|} \right)$$

dove  $K(A) = \|A\| \|A^{-1}\|$ . Anche in questa seconda situazione, più realistica,  $K(A)$  rappresenta sostanzialmente il fattore di amplificazione delle perturbazioni (relative) introdotte in  $A$  e  $b$ ; infatti, se, per esempio,  $\|\delta A\| \leq \frac{1}{2\|A^{-1}\|}$  abbiamo

$$\frac{K(A)}{1 - K(A) \frac{\|\delta A\|}{\|A\|}} \leq 2K(A)$$

In ogni caso, la quantità  $K(A)$  viene definita *numero di condizionamento* del sistema  $Ax = b$ .

Osserviamo che per le norme di matrice naturali risulta

$$K(A) = \|A\| \|A^{-1}\| \geq \|AA^{-1}\| = \|I\| = 1$$

Generalmente si considera  $K_2(A) = \|A\|_2 \|A^{-1}\|_2$  (numero di condizionamento spettrale) oppure  $K_\infty(A) = \|A\|_\infty \|A^{-1}\|_\infty$ .

Il seguente teorema precisa meglio il significato del numero  $K(A)$ :

**Teorema 3.1.** ([10, pag. 177]). *Per ogni matrice non singolare  $A \in \mathbb{R}^{n \times n}$  e per ogni norma di matrice compatibile con una di vettore, la quantità  $1/K(A)$  rappresenta la distanza relativa di  $A$  dall'insieme di tutte le matrici singolari di ordine  $n$ , cioè*

$$\frac{1}{K(A)} = \min \left\{ \frac{\|A - B\|}{\|A\|} : B \in \mathbb{R}^{n \times n} \text{ singolare} \right\}$$

Un esempio classico di sistema mal condizionato è quello associato alla matrice di Hilbert

$$H_n = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{n+1} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \cdots & \frac{1}{n+2} \\ \dots & \dots & \dots & \dots & \dots \\ \frac{1}{n} & \frac{1}{n+1} & \frac{1}{n+2} & \cdots & \frac{1}{2n-1} \end{pmatrix}$$

Nella successiva tabella riportiamo alcuni suoi numeri di condizionamento spettrale.

$n$	2	3	4	5	6	7	8	9	10
$K_2(H_n)$	$1.9 \times 10^1$	$5.2 \times 10^2$	$1.6 \cdot 10^4$	$4.8 \cdot 10^5$	$1.5 \cdot 10^7$	$4.8 \cdot 10^8$	$1.5 \cdot 10^{10}$	$4.9 \cdot 10^{11}$	$1.6 \cdot 10^{13}$

Anche i sistemi con matrici di Vandermonde(<sup>†</sup>)

$$V_n = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ x_1 & x_2 & x_3 & \dots & x_n \\ x_1^2 & x_2^2 & x_3^2 & \dots & x_n^2 \\ \dots & \dots & \dots & \dots & \dots \\ x_1^{n-1} & x_2^{n-1} & x_3^{n-1} & \dots & x_n^{n-1} \end{pmatrix}, \quad x_i \neq x_j \quad \text{per } i \neq j$$

risultano mal condizionati.

Osserviamo infine che la quantità  $K(A)$  che consideriamo come misura del condizionamento del problema, ovvero come fattore di amplificazione degli errori (relativi), può in alcuni casi rivelarsi troppo pessimistica. Attribuire tale significato al numero  $K(A)$  significa supporre di trovarci nelle condizioni peggiori, ossia supporre che tutte le disegualanze  $\leq$  introdotte nello studio del condizionamento siano in realtà delle uguaglianze; e ciò può non sempre esser vero.

## 3.2 Metodi diretti

### 3.2.1 Il metodo di eliminazione di Gauss

Il metodo diretto più noto e più utilizzato è senza dubbio quello delle eliminazioni successive di Gauss.

Osserviamo preliminarmente che la soluzione di un sistema non singolare di forma triangolare, superiore o inferiore, è pressoché immediata. Infatti, nel caso di un sistema triangolare superiore

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \dots \dots \dots \\ a_{nn}x_n = b_n \end{array} \right.$$

con elementi diagonale  $a_{ii} \neq 0$ ,  $i = 1, 2, \dots, n$ , abbiamo

$$(3.5) \quad \left\{ \begin{array}{l} x_n = \frac{b_n}{a_{nn}} \\ x_k = \left( b_k - \sum_{j=k+1}^n a_{kj}x_j \right) / a_{kk}, \quad k = n-1, n-2, \dots, 1 \end{array} \right.$$

con sole  $n^2/2$  operazioni aritmetiche. Questa considerazione ci suggerisce pertanto di esaminare la possibilità di trasformare un generico sistema non singolare in un sistema *equivalente*, cioè con la stessa soluzione, di forma triangolare.

---

(<sup>†</sup>) Che incontreremo, per esempio, nello studio del problema dell'interpolazione, capitolo 5.

È noto che, quando ad un'equazione del sistema sostituiamo una combinazione lineare dell'equazione con un'altra dello stesso sistema, il nuovo sistema risulta equivalente al precedente. Con il metodo di Gauss che tra poco presenteremo dimostreremo che è sempre possibile trasformare, mediante un numero finito di combinazioni lineari del tipo predetto ed eventuali permutazioni di equazioni, un generico sistema non singolare in un sistema equivalente di forma triangolare.

Per meglio illustrare il processo delle eliminazioni di Gauss scriviamo il sistema  $Ax = b$  esplicitamente:

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \dots \dots \dots \dots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{array} \right.$$

Supponiamo  $a_{11} \neq 0$  (se  $a_{11} = 0$  è sufficiente, come vedremo più avanti in 3.2.2, permutare le equazioni). Possiamo eliminare l'incognita  $x_1$  dalle ultime  $(n - 1)$  equazioni, cioè dalla 2<sup>a</sup>, 3<sup>a</sup>, ...,  $n$ -esima, sommando alla  $i$ -esima equazione,  $i = 2, 3, \dots, n$ , la prima moltiplicata per

$$m_{i1} = -\frac{a_{i1}}{a_{11}}, \quad i = 2, 3, \dots, n$$

Dopo queste operazioni, il nuovo sistema, equivalente al precedente, assume la forma

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ \boxed{a_{22}^{(2)}x_2 + \cdots + a_{2n}^{(2)}x_n = b_2^{(2)}} \\ \dots \dots \dots \dots \\ a_{n2}^{(2)}x_2 + \cdots + a_{nn}^{(2)}x_n = b_n^{(2)} \end{array} \right.$$

dove

$$i = 2, \dots, n : \left\{ \begin{array}{l} a_{ij}^{(2)} = a_{ij} + m_{i1}a_{1j}, \quad j = 2, \dots, n \\ b_i^{(2)} = b_i + m_{i1}b_1 \end{array} \right.$$

Riapplichiamo il procedimento (di eliminazione) alle ultime  $(n - 1)$  equazioni. Se  $a_{22}^{(2)} \neq 0$  possiamo eliminare l'incognita  $x_2$  dalla 3<sup>a</sup>, 4<sup>a</sup>, ...,  $n$ -esima equazione; è sufficiente porre

$$m_{i2} = -\frac{a_{i2}^{(2)}}{a_{22}^{(2)}}, \quad i = 3, \dots, n$$

e sommare alla  $i$ -esima equazione,  $i = 3, \dots, n$ , la 2<sup>a</sup> moltiplicata per  $m_{i2}$ . Avremo un

nuovo sistema equivalente a quello di partenza:

$$\left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n = b_1 \\ a_{22}^{(2)}x_2 + a_{23}^{(2)}x_3 + \cdots + a_{2n}^{(2)}x_n = b_2^{(2)} \\ \boxed{a_{33}^{(3)}x_3 + \cdots + a_{3n}^{(3)}x_n = b_3^{(3)}} \\ \cdots \cdots \cdots \\ a_{n3}^{(3)}x_3 + \cdots + a_{nn}^{(3)}x_n = b_n^{(3)} \end{array} \right.$$

dove

$$i = 3, \dots, n : \begin{cases} a_{ij}^{(3)} = a_{ij}^{(2)} + m_{i2}a_{2j}^{(2)}, & j = 3, \dots, n \\ b_i^{(3)} = b_i^{(2)} + m_{i2}b_2^{(2)} \end{cases}$$

Gli elementi  $a_{11}, a_{22}^{(2)}, a_{33}^{(3)}, \dots$ , che compaiono durante le successive eliminazioni vengono chiamati *elementi pivot*.

Dopo  $(n-1)$  passi arriveremo, supponendo tutti gli elementi pivot non nulli, al seguente sistema triangolare

$$\left\{ \begin{array}{l} a_{11}^{(1)}x_1 + a_{12}^{(1)}x_2 + a_{13}^{(1)}x_3 + \cdots + a_{1n}^{(1)}x_n = b_1^{(1)} \\ a_{22}^{(2)}x_2 + a_{23}^{(2)}x_3 + \cdots + a_{2n}^{(2)}x_n = b_2^{(2)} \\ a_{33}^{(3)}x_3 + \cdots + a_{3n}^{(3)}x_n = b_3^{(3)} \\ \cdots \cdots \cdots \\ a_{nn}^{(n)}x_n = b_n^{(n)} \end{array} \right.$$

dove per convenienza di notazione abbiamo posto  $a_{1j}^{(1)} = a_{1j}$ ,  $j = 1, \dots, n$ , e  $b_1^{(1)} = b_1$ , la cui risoluzione è, come rilevato all'inizio del paragrafo, pressoché immediata. Osserviamo che il termine noto  $b$  viene trasformato esattamente come se fosse un'ulteriore colonna di  $A$ .

Lo schema di calcolo seguente riassume la descrizione del metodo di Gauss:

- l'eliminazione delle variabili viene eseguita in  $(n-1)$  passi; al passo  $k$ -esimo,  $k = 1, 2, \dots, n-1$ , gli elementi  $a_{ij}^{(k)}$ ,  $i$  e  $j > k$ , e  $b_i^{(k)}$  vengono trasformati in accordo con le formule

$$i = k+1, \dots, n : \begin{cases} m_{ik} = -a_{ik}^{(k)} / a_{kk}^{(k)} \\ a_{ij}^{(k+1)} = a_{ij}^{(k)} + m_{ik}a_{kj}^{(k)}, & j = k+1, \dots, n \\ b_i^{(k+1)} = b_i^{(k)} + m_{ik}b_k^{(k)} \end{cases}$$

2. la soluzione del sistema triangolare finale risulta

$$\begin{cases} x_n = \frac{b_n^{(n)}}{a_{nn}^{(n)}} \\ x_k = \left( b_k^{(k)} - \sum_{j=k+1}^n a_{kj}^{(k)} x_j \right) / a_{kk}^{(k)}, \quad k = n-1, \dots, 1 \end{cases}$$

► **Esempio.**

$$\begin{aligned} & \begin{cases} 2x_1 - x_2 + x_3 - 2x_4 = 0 \\ 2x_2 - x_4 = 1 \\ x_1 - 2x_3 + x_4 = 0 \\ 2x_2 + x_3 + x_4 = 4 \end{cases} \implies \begin{cases} 2x_1 - x_2 + x_3 - 2x_4 = 0 \\ 2x_2 - x_4 = 1 \\ \frac{1}{2}x_2 - \frac{5}{2}x_3 + 2x_4 = 0 \\ 2x_2 + x_3 + x_4 = 4 \end{cases} \\ & \implies \begin{cases} 2x_1 - x_2 + x_3 - 2x_4 = 0 \\ 2x_2 - x_4 = 1 \\ -\frac{5}{2}x_3 + \frac{9}{4}x_4 = -\frac{1}{4} \\ x_3 + 2x_4 = 3 \end{cases} \implies \begin{cases} 2x_1 - x_2 + x_3 - 2x_4 = 0 \\ 2x_2 - x_4 = 1 \\ -\frac{5}{2}x_3 + \frac{9}{4}x_4 = -\frac{1}{4} \\ \frac{29}{10}x_4 = \frac{29}{10} \end{cases} \end{aligned}$$

Partendo dall'ultima equazione (del sistema triangolare finale) otteniamo

$$\begin{aligned} x_4 &= 1 \\ x_3 &= \left( -\frac{1}{4} - \frac{9}{4}x_4 \right) \left( -\frac{2}{5} \right) = 1 \\ x_2 &= (1 + x_4)/2 = 1 \\ x_1 &= (x_2 - x_3 + 2x_4)/2 = 1 \end{aligned}$$



Denotando con  $A_k$  la matrice di ordine  $k$  formata dagli elementi  $a_{ij}$ ,  $1 \leq i, j \leq k$  possiamo affermare (vedi [3, teorema 5.3.1, pag. 156]) che *il procedimento delle eliminazioni di Gauss può essere portato a termine senza permutare l'ordine iniziale delle equazioni, ovvero i successivi elementi pivot  $a_{kk}^{(k)}$  sono tutti non nulli, se e solo se  $\det(A_k) \neq 0$ ,  $k = 1, 2, \dots, n$ .* Ciò è senz'altro vero, per esempio, quando la matrice  $A$  è a diagonale dominante per righe o per colonne oppure è simmetrica definita positiva. In quest'ultimo caso (vedi l'esercizio 3.7) il numero di operazioni necessario per la triangolarizzazione può essere dimezzato.

Se al generico passo  $k$ -esimo il processo di eliminazione non viene effettuato solo sulle righe successive alla  $k$ -esima, ma anche sulle precedenti, allora dopo  $n$  passi otteniamo

un sistema diagonale

$$\left\{ \begin{array}{lll} \bar{a}_{11}^{(1)} x_1 & = \bar{b}_1^{(1)} \\ \bar{a}_{22}^{(2)} x_2 & = \bar{b}_2^{(2)} \\ \bar{a}_{33}^{(3)} x_3 & = \bar{b}_3^{(3)} \\ \vdots & \vdots \\ \bar{a}_{nn}^{(n)} x_n & = \bar{b}_n^{(n)} \end{array} \right.$$

Questa variante del metodo di Gauss è generalmente nota con il nome di *metodo di Jordan*.

Confrontiamo il numero di operazioni aritmetiche richieste dai due metodi per il calcolo della soluzione del sistema  $Ax = b$ ,  $A \in \mathbb{R}^{n \times n}$ :

	$+, -$	$\times$	$/$
Gauss	$\frac{n(n-1)(2n+5)}{6}$	$\frac{n(n-1)(2n+5)}{6}$	$\frac{n(n+1)}{2}$
Jordan	$\frac{n(n-1)(n+1)}{2}$	$\frac{n(n-1)(n+1)}{2}$	$\frac{n(n+1)}{2}$

Il numero di operazioni è essenzialmente  $n^3/3$  per Gauss e  $n^3/2$  per Jordan, e quindi il primo è da preferirsi. Per avere un'idea più precisa della mole di calcolo richiesta dai due metodi ricordiamo che il prodotto di due matrici di ordine  $n$ , eseguito applicando direttamente la definizione di prodotto, richiede  $n^3$  operazioni!

### 3.2.2 Pivoting e scaling

Se nello svolgimento del processo delle eliminazioni di Gauss, al passo  $k$ -esimo troviamo  $a_{kk}^{(k)} = 0$ , allora il metodo così come è stato descritto non può proseguire. Supposto il sistema non singolare, il rimedio consiste nello scambiare di posizione due equazioni (la  $k$ -esima con una delle successive) in modo che il nuovo  $a_{kk}^{(k)}$  sia diverso da zero. Infatti, se  $a_{kk}^{(k)} = 0$  necessariamente qualche altro elemento  $a_{ik}^{(k)}$ ,  $i = k+1, \dots, n$ , della colonna  $k$ -esima della matrice dei coefficienti deve essere non nullo, altrimenti il sistema risulta singolare. Supponiamo per esempio che  $a_{rk}^{(k)} \neq 0$ ; in questo caso basta scambiare l'equazione  $k$ -esima con la  $r$ -esima e poi procedere con le eliminazioni. Ne segue che *ogni sistema non singolare può sempre essere ricondotto alla forma triangolare* (superiore) con il metodo di Gauss (più eventuali scambi di equazioni). Se nell'esempio presentato

a pagina 44 il sistema iniziale fosse stato

$$\begin{cases} 2x_2 & - x_4 = 1 \\ 2x_1 - x_2 + x_3 - 2x_4 = 0 \\ x_1 & - 2x_3 + x_4 = 0 \\ 2x_2 + x_3 + x_4 = 4 \end{cases}$$

prima di proseguire con il processo delle eliminazioni avremmo dovuto scambiare la prima equazione con la seconda o con la terza.

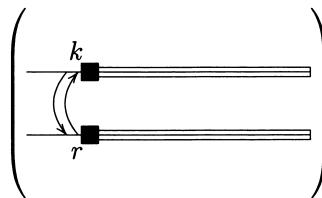
Per assicurare una migliore stabilità numerica al metodo è spesso necessario permutare l'ordine delle equazioni anche quando l'elemento pivot non è esattamente zero, ma è molto piccolo (in valore assoluto) rispetto agli altri elementi. Infatti, in quest'ultimo caso l'elemento pivot potrebbe essere stato generato come differenza di due numeri quasi uguali e quindi essere stato contaminato dal fenomeno della cancellazione numerica.

Per cercare di evitare catastrofiche propagazioni di errori è di solito necessario scegliere, al generico passo  $k$ -esimo, l'elemento pivot seguendo una delle due seguenti strategie:

- (i) *Pivoting parziale*: scegliere  $r$  uguale al più piccolo intero  $\geq k$  tale che

$$|a_{rk}^{(k)}| = \max_{k \leq i \leq n} |a_{ik}^{(k)}|$$

e, se  $r \neq k$ , scambiare l'equazione  $k$ -esima con la  $r$ -esima.



**Figura 3.1**

- (ii) *Pivoting completo*: scegliere una coppia  $(r, s)$ ,  $r$  e  $s \geq k$ , (la più vicina a  $(k, k)$  per esempio) tale che

$$|a_{rs}^{(k)}| = \max_{k \leq i, j \leq n} |a_{ij}^{(k)}|$$

e scambiare l'equazione  $k$ -esima con la  $r$ -esima e l'incognita  $k$ -esima (con il suo coefficiente) con la  $s$ -esima (fig. 3.2).

Poiché il pivoting parziale risulta generalmente soddisfacente, quello completo, a causa dell'eccessivo lavoro di ricerca dell'elemento massimo, è poco usato.

Nel caso di matrici dei coefficienti simmetriche e a diagonale dominante è possibile dimostrare che l'introduzione della strategia del pivoting parziale è del tutto superflua in quanto essa non provoca alcun scambio di equazioni. Ricordiamo infine che quando

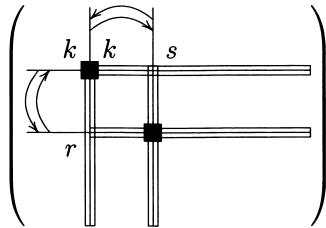


Figura 3.2

la matrice  $A$  è simmetrica definita positiva, l'algoritmo di Gauss senza pivoting risulta numericamente stabile.

Riprendiamo in esame la tecnica di pivoting e osserviamo che moltiplicando una singola equazione per una costante arbitraria la soluzione (teorica) del sistema rimane inalterata, mentre la scelta dell'elemento pivot può venire fortemente influenzata da un'operazione di questo tipo. Evidentemente la tecnica di pivoting da sola potrebbe non risultare efficace nel prevenire eventuali propagazioni di errori, soprattutto quando gli elementi della matrice iniziale sono molto diversi tra di loro come ordine di grandezza. Per tentare di ovviare a questo inconveniente, prima di applicare il metodo di Gauss con pivoting potremmo pensare di *equilibrare* la matrice del sistema nel modo seguente:

$$D_1 A D_2 = B$$

con  $D_1$  e  $D_2$  matrici diagonali da scegliere in modo “opportuno”. Il metodo di Gauss verrebbe poi applicato al nuovo sistema

$$By = c$$

dove  $c = D_1 b$  e  $x = D_2 y$ . L'efficienza di questa operazione di *scaling* non è tuttavia garantita, ed in ogni caso in generale non è nota una scelta ottimale delle matrici  $D_1$  e  $D_2$ .

La tecnica di scaling forse più usata ricorre all'equilibratura delle sole righe ( $D_2 = I$ ) della matrice  $A$ , rendendole tutte di lunghezza unitaria mediante la scelta, per esempio,

$$(D_1)_{ii} = \frac{1}{\|a_i^T\|_\infty} = \frac{1}{\max_{1 \leq j \leq n} |a_{ij}|}$$

dove con  $a_i^T$  denotiamo la  $i$ -esima riga di  $A$ . Poiché lo scopo di questa operazione è una scelta più efficace dell'elemento pivot, alcuni preferiscono operare un'equilibratura implicita nella stessa fase di scelta dell'elemento pivot (pivoting parziale)  $a_{rk}^{(k)}$ :

$$\frac{|a_{rk}^{(k)}|}{s_r} = \max_{k \leq i \leq n} \frac{|a_{ik}^{(k)}|}{s_i}$$

con

$$s_i = \max_{1 \leq j \leq n} |a_{ij}|, \quad i = 1, \dots, n$$

oppure

$$s_i = s_i^{(k)} = \max_{k \leq j \leq n} |a_{ij}^{(k)}|, \quad i = k, \dots, n$$

Tuttavia spesso questa operazione di scaling non comporta alcun miglioramento e, anzi, a priori non possiamo escludere che in qualche caso non peggiori addirittura la stabilità dell'algoritmo. Pertanto gli autori di quella che è ritenuta la miglior collezione di routine di metodi diretti per sistemi lineari (LINPACK) adottano la semplice strategia del pivoting parziale senza introdurre alcuna operazione di scaling.

Prima di concludere questo paragrafo, consideriamo il seguente sistema lineare di ordine  $n = 18$ :

$$\begin{aligned} Ax &= b \\ a_{ij} &= \cos((j-1)\theta_i), \quad \theta_i = \frac{2i-1}{2n}\pi \\ b_i &= \sum_{j=1}^n a_{ij} \end{aligned}$$

la cui esatta soluzione è  $x = (1, 1, \dots, 1)^T$ . Risolviamolo dapprima con il semplice metodo di Gauss, e poi inserendo nel metodo stesso la strategia di pivoting parziale (senza scaling). I risultati ottenuti sono riportati nella tabella 3.1.

La colonna pivot riporta le successive permutazioni effettuate con la strategia di pivoting parziale. Essa va interpretata nel modo seguente: poiché  $\text{pivot}(1) = 1$ , l'equazione numero 1 è rimasta al suo posto;  $\text{pivot}(2) = 18$  significa invece che al passo  $k = 2$  (ovvero dopo aver introdotto gli zeri nella 1<sup>a</sup> colonna e prima di introdurre gli zeri nella 2<sup>a</sup> colonna) l'equazione numero 2 è stata scambiata con la 18<sup>a</sup>; successivamente, al passo  $k = 3$ , essendo  $\text{pivot}(3) = 9$  la 3<sup>a</sup> equazione è stata scambiata con la 9<sup>a</sup>; e così di seguito.

L'introduzione dello scaling comporterebbe una diversa scelta degli elementi pivot, ma la soluzione finale non risulterebbe diversa da quella da noi ottenuta senza tale variante (peraltro onerosa in termini di operazioni aritmetiche).

D'ora in avanti con il termine *metodo di Gauss* intenderemo il metodo con pivoting parziale.

### 3.2.3 Decomposizione di Gauss e fattorizzazione LU

In questo paragrafo ci proponiamo di interpretare il metodo di Gauss come successione finita di trasformazioni della matrice dei coefficienti  $A$  e del termine noto  $b$ , ovvero come moltiplicazione di  $A$  e di  $b$  per un numero finito di opportune matrici. Questa particolare interpretazione ci consentirà poi di riformulare l'algoritmo di Gauss in due parti distinte: una, la più onerosa in termini di operazioni aritmetiche, determinerà una matrice non singolare  $G$  tale che  $GA = U$  è di forma triangolare superiore; l'altra, utilizzando la matrice  $G$ , ci consentirà di risolvere il sistema  $Ax = b$ :

$$GAx = Gb \quad \rightarrow \quad Ux = \bar{b}$$

Gauss semplice	Gauss + pivoting parziale	pivot
$-0.1066476 \cdot 10^1$	$0.1000001 \cdot 10^1$	1
$0.4751467 \cdot 10^1$	$0.1000000 \cdot 10^1$	18
$-0.1786051 \cdot 10^1$	$0.9999998 \cdot 10^0$	9
$0.2645315 \cdot 10^1$	$0.9999994 \cdot 10^0$	13
$0.2965986 \cdot 10^0$	$0.1000000 \cdot 10^1$	5
$0.1126027 \cdot 10^1$	$0.1000000 \cdot 10^1$	15
$0.1119538 \cdot 10^1$	$0.9999997 \cdot 10^0$	7
$0.8452096 \cdot 10^0$	$0.9999999 \cdot 10^0$	11
$0.1095097 \cdot 10^1$	$0.9999996 \cdot 10^0$	9
$0.9925374 \cdot 10^0$	$0.9999999 \cdot 10^0$	16
$0.9083479 \cdot 10^0$	$0.9999998 \cdot 10^0$	15
$0.1211428 \cdot 10^1$	$0.9999998 \cdot 10^0$	12
$0.6517138 \cdot 10^0$	$0.9999999 \cdot 10^0$	18
$0.1463063 \cdot 10^1$	$0.9999998 \cdot 10^0$	17
$0.4999179 \cdot 10^0$	$0.1000000 \cdot 10^1$	15
$0.1434433 \cdot 10^1$	$0.1000000 \cdot 10^1$	16
$0.7034890 \cdot 10^0$	$0.1000000 \cdot 10^1$	18
$0.1142145 \cdot 10^1$	$0.9999998 \cdot 10^0$	

Tabella 3.1

Questa interpretazione si rivelerà particolarmente utile, in quanto comporterà una notevole riduzione di operazioni aritmetiche in importanti applicazioni, quali ad esempio la determinazione della matrice inversa  $A^{-1}$  (vedi pagina 63) e il metodo delle potenze inverse (vedi pagina 98).

Osserviamo preliminarmente che lo scambio di due equazioni del sistema  $Ax = b$ , per esempio la  $i$ -esima con la  $j$ -esima, può essere interpretato come prodotto (da sinistra) di entrambi i membri del sistema per la matrice

$$P_{ij} = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 & \dots & 0 & \dots & 0 \\ \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & \dots & 1 & \dots & 0 \\ \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & \dots & 0 & \dots & 0 \\ \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & \dots & 0 & \dots & 1 \end{pmatrix} \quad \begin{matrix} i \\ j \end{matrix}$$

ottenuta dalla matrice identità  $I$  scambiando tra di loro le righe  $i$ -esima e  $j$ -esima; cioè

$$P_{ij}Ax = P_{ij}b$$

Analogamente, la sostituzione nel sistema dell'equazione  $i$ -esima con la medesima più la  $j$ -esima moltiplicata per il coefficiente  $m_{ij}$  può essere ottenuta moltiplicando (da sinistra) entrambi i membri dell'equazione  $Ax = b$  per la matrice

$$M_{ij} = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 & \dots & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 & \dots & 0 & \dots & 0 \\ \dots & \dots \\ 0 & 0 & 0 & \dots & 1 & \dots & 0 & \dots & 0 \\ \dots & \dots \\ 0 & 0 & 0 & \dots & m_{ij} & \dots & 1 & \dots & 0 \\ \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & \dots & 0 & \dots & 1 \end{pmatrix} = I + m_{ij}e_i e_j^T$$

dove

$$e_k = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

$(e_k)_i = \delta_{ki}$  e  $e_i e_j^T$  è la matrice di ordine  $n$  con elementi tutti nulli tranne quello in posizione  $(i, j)$  che assume il valore 1. Verifichiamo inoltre che  $P_{ij}^{-1} = P_{ij}$  e

$$M_{ij}^{-1} = \begin{pmatrix} 1 & & & & & & & & \\ & 1 & & & & & & & \\ & & 1 & & & & & & \\ & & & \ddots & & & & & \\ & & & & 1 & & & & \\ & & & & & \ddots & & & \\ 0 & & & & & & 1 & & \\ & & & & & & & \ddots & \\ & & & & & & & & 1 \end{pmatrix} = I - m_{ij}e_i e_j^T$$

Infatti,

$$\begin{aligned}
 P_{ij}P_{ij} &= \left( e_1, \dots, e_{i-1}, e_j, e_{i+1}, \dots, e_{j-1}, e_i, e_{j+1}, \dots, e_n \right) \cdot \begin{pmatrix} e_1^T \\ \vdots \\ e_{i-1}^T \\ e_j^T \\ e_i^T \\ e_{i+1}^T \\ \vdots \\ e_{j-1}^T \\ e_i^T \\ e_{j+1}^T \\ \vdots \\ e_n^T \end{pmatrix} \\
 &= e_1e_1^T + \cdots + e_{i-1}e_{i-1}^T + e_je_j^T + e_{i+1}e_{i+1}^T + \cdots \\
 &\quad + e_{j-1}e_{j-1}^T + e_ie_i^T + e_{j+1}e_{j+1}^T + \cdots + e_ne_n^T = I
 \end{aligned}$$

e

$$(I - m_{ij}e_i e_j^T)(I + m_{ij}e_i e_j^T) = I + m_{ij}Ie_i e_j^T - m_{ij}e_i e_j^T I - m_{ij}^2 e_i(e_j^T e_i)e_j^T = I, \quad i \neq j$$

Pertanto, con il metodo di Gauss in realtà determiniamo implicitamente delle matrici  $P_1, P_2, \dots, P_{n-1}$ , di tipo  $I$  quando non avvengono scambi di equazioni e  $P_{ij}$  altrimenti, e delle matrici  $M_1, M_2, \dots, M_{n-1}$ , con

$$\begin{aligned}
 M_j &= M_{nj} \dots M_{j+2,j} M_{j+1,j} \\
 (3.6) \quad &= \begin{pmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & \ddots & & & 0 & \\ & & & & 1 & & & \\ & & & & & m_{j+1,j} & 1 & \\ & 0 & & m_{j+2,j} & & & \ddots & \\ & & & \vdots & & & & \ddots \\ & & & m_{nj} & & & & 1 \end{pmatrix} \\
 &= I + \sum_{i=j+1}^n m_{ij}e_i e_j^T
 \end{aligned}$$

tali che il nuovo sistema

$$M_{n-1}P_{n-1} \dots M_2P_2M_1P_1Ax = M_{n-1}P_{n-1} \dots M_2P_2M_1P_1b$$

assuma la forma triangolare superiore  $Ux = \bar{b}$ , ossia

$$M_{n-1}P_{n-1} \dots M_2P_2M_1P_1A = U$$

Posto  $G = M_{n-1}P_{n-1} \dots M_2P_2M_1P_1$ , in seguito denoteremo questa decomposizione con  $GA = U$ . Ricordiamo che la sua realizzazione richiede  $n^3/3$  operazioni aritmetiche.

▷ **Osservazioni.** (i) Nelle applicazioni è del tutto superfluo costruire esplicitamente la matrice  $G$ . Poiché  $G$  verrà utilizzata solo per trasformare vettori e matrici, è sufficiente memorizzare i moltiplicatori  $m_{ij}$  e le permutazioni effettuate. I moltiplicatori  $m_{ij}$  ( $i > j$ ) verranno memorizzati nelle corrispondenti posizioni della matrice  $A(a_{ij} \leftarrow m_{ij})$ , e al termine della triangolarizzazione al posto della matrice iniziale  $A$  avremo

$$\begin{pmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{1n} \\ m_{21} \searrow u_{22} & u_{23} & \cdots & u_{2n} \\ m_{31} & m_{32} \searrow u_{33} & \cdots & u_{3n} \\ \cdots & \cdots & \cdots & \cdots \\ m_{n1} & m_{n2} & m_{n3} & \cdots & u_{nn} \end{pmatrix}$$

(ii) Anche la costruzione delle matrici  $P_i$  è superflua. Poiché la loro unica funzione è quella di provocare scambi di righe, possiamo riprodurre tali azioni semplicemente memorizzando gli scambi effettuati in un vettore (che in seguito chiameremo pivot) di  $n - 1$  componenti intere. Per esempio, se al passo  $k$ -esimo del processo di triangolarizzazione viene effettuata la permutazione tra le righe  $k$ -esima ed  $r$ -esima, è sufficiente porre  $\text{pivot}(k) \leftarrow r$ ; se invece la riga  $k$ -esima rimane al suo posto,  $\text{pivot}(k) \leftarrow k$ .

(iii) Poiché

$$\det(M_{n-1}) \det(P_{n-1}) \dots \det(M_2) \det(P_2) \det(M_1) \det(P_1) \det(A) = \det(U)$$

e

$$\det(M_j) = 1 \quad \det(P_i) = \begin{cases} 1 & \text{se } P_i = I \\ -1 & \text{altrimenti} \end{cases} \quad \det(U) = \prod_{i=1}^n u_{ii}$$

abbiamo

$$\det(A) = (-1)^s \prod_{i=1}^n u_{ii}$$

dove  $s$  denota il numero complessivo di scambi effettuati. □

► **Esempio.** Consideriamo la matrice

$$A = \begin{pmatrix} 0 & 2 & 0 & -1 \\ 2 & -1 & 1 & -2 \\ 1 & 0 & -2 & 1 \\ -1 & 3 & 1 & 1 \end{pmatrix}$$

Nelle successive fasi del procedimento di Gauss essa subisce le seguenti trasformazioni:

$$k = 1$$

$$\Rightarrow \begin{pmatrix} 2 & -1 & 1 & -2 \\ 0 & 2 & 0 & -1 \\ 1 & 0 & -2 & 1 \\ -1 & 3 & 1 & 1 \end{pmatrix}$$

$$\text{pivot} = \begin{pmatrix} 2 \\ 2 \end{pmatrix} \Rightarrow \begin{pmatrix} 2 & -1 & 1 & -2 \\ 0 & 2 & 0 & -1 \\ -\frac{1}{2} & \frac{1}{2} & -\frac{5}{2} & 2 \\ \frac{1}{2} & \frac{5}{2} & \frac{3}{2} & 0 \end{pmatrix}$$

$$k = 2$$

$$\Rightarrow \begin{pmatrix} 2 & -1 & 1 & -2 \\ 0 & \frac{5}{2} & \frac{3}{2} & 0 \\ -\frac{1}{2} & \frac{1}{2} & -\frac{5}{2} & 2 \\ \frac{1}{2} & 2 & 0 & -1 \end{pmatrix}$$

$$\text{pivot} = \begin{pmatrix} 2 \\ 4 \end{pmatrix} \Rightarrow \begin{pmatrix} 2 & -1 & 1 & -2 \\ 0 & \frac{5}{2} & \frac{3}{2} & 0 \\ -\frac{1}{2} & -\frac{1}{5} & -\frac{14}{5} & 2 \\ \frac{1}{2} & -\frac{4}{5} & -\frac{6}{5} & -1 \end{pmatrix}$$

$$k = 3$$

$$\Rightarrow \begin{pmatrix} 2 & -1 & 1 & -2 \\ 0 & \frac{5}{2} & \frac{3}{2} & 0 \\ -\frac{1}{2} & -\frac{1}{5} & -\frac{14}{5} & 2 \\ \frac{1}{2} & -\frac{4}{5} & -\frac{6}{5} & -1 \end{pmatrix}$$

$$\text{pivot} = \begin{pmatrix} 2 \\ 4 \\ 3 \end{pmatrix} \Rightarrow \begin{pmatrix} 2 & -1 & 1 & -2 \\ 0 & \frac{5}{2} & \frac{3}{2} & 0 \\ -\frac{1}{2} & -\frac{1}{5} & -\frac{14}{5} & 2 \\ \frac{1}{2} & -\frac{4}{5} & -\frac{3}{7} & -\frac{13}{7} \end{pmatrix}$$

Il contenuto del vettore pivot va interpretato nel modo seguente: all'inizio del passo  $k = 1$  (introduzione degli zeri nella 1<sup>a</sup> colonna), poiché  $\text{pivot}(1) = 2$ , la prima riga è stata scambiata con la 2<sup>a</sup>; successivamente, dopo aver operato le combinazioni lineari che hanno “introdotto” gli zeri nella 1<sup>a</sup> colonna, poiché  $\text{pivot}(2) = 4$ , la nuova 2<sup>a</sup> riga è stata scambiata con la 4<sup>a</sup>; infine, la 3<sup>a</sup> non è stata rimossa ( $\text{pivot}(3) = 3$ ). Inoltre,

$$\det(A) = (-1)^2 \times 2 \times \frac{5}{2} \times \left(-\frac{14}{5}\right) \times \left(-\frac{13}{7}\right) = 26$$



Concludiamo questa prima parte con la descrizione dell'algoritmo (Factor) che determina la composizione di Gauss  $GA = U$  di una matrice non singolare  $A$  di ordine  $n$ .

---

**Algoritmo 1:** Factor( $n, A, \text{pivot}, \det, \text{ier}$ )
 

---

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

La variabile  $\det$  contiene il valore  $\det(A)$ .

La variabile  $\text{ier}$  è un indicatore di errore. Se  $\text{ier} = 0$  il processo di Gauss è stato portato a termine ed in  $A$  troviamo le matrici  $G$  e  $U$ ; se invece  $\text{ier} = 1$  la matrice  $A$  risulta singolare.

*Parametri.* **Input:**  $n, A$

**Output:**  $A, \text{pivot}, \det, \text{ier}$

- 1:  $\det \leftarrow 1$
  - 2: **ciclo 1:**  $k = 1, \dots, n - 1$
  - 3:    $a_{\max} \leftarrow \max_{k \leq i \leq n} |a_{ik}|$ ; sia  $i_0$  il più piccolo indice  $i \geq k$  tale che  $|a_{i_0k}| = a_{\max}$
  - 4:    $\text{pivot}(k) \leftarrow i_0$
  - 5:   **se**  $a_{\max} = 0$  **allora**  $\det \leftarrow 0$ ,  $\text{ier} \leftarrow 1$ ; **esci**
  - 6:   **se**  $i_0 = k$  **allora** vai al punto 9
  - 7:    $a_{kj} \leftrightarrow a_{i_0j}$ ,  $j = k, \dots, n$
  - 8:    $\det \leftarrow -\det$
  - 9: **ciclo 2:**  $i = k + 1, \dots, n$
  - 10:    $a_{ik} \leftarrow m_{ik} = -a_{ik}/a_{kk}$
  - 11:    $a_{ij} \leftarrow a_{ij} + a_{ik}a_{kj}$ ,  $j = k + 1, \dots, n$
  - 12: **fine ciclo 2**
  - 13:  $\det \leftarrow \det \cdot a_{kk}$
  - 14: **fine ciclo 1**
  - 15: **se**  $a_{nn} = 0$  **allora**  $\det \leftarrow 0$ ,  $\text{ier} \leftarrow 1$ ; **esci**
  - 16:  $\det \leftarrow \det \cdot a_{nn}$
  - 17:  $\text{ier} \leftarrow 0$
  - 18: **esci**
-

Note le matrici  $G(\dagger)$  e  $U$ , per risolvere il sistema  $Ax = b$  è sufficiente porre

$$GAx = Gb$$

cioè formare il vettore

$$\bar{b} = Gb = M_{n-1}P_{n-1} \dots M_2P_2M_1P_1b$$

e risolvere il sistema triangolare superiore

$$Ux = \bar{b}$$

L'algoritmo che segue (Solve) utilizza i risultati forniti da Factor e determina, con  $n^2$  operazioni aritmetiche, la soluzione  $x$ .

---

**Algoritmo 2:** Solve( $n, A, \text{pivot}, b$ )
 

---

*Commento.* L'algoritmo risolve il sistema non singolare, di ordine  $n$ ,  $Ux = \bar{b}$ ,  $\bar{b} = M_{n-1}P_{n-1} \dots M_2P_2M_1P_1b$ , con  $(U)_{ij} = (A)_{ij}$ ,  $i \leq j$ , e  $m_{ij} = (A)_{ij}$ ,  $i > j$ .

La matrice input  $A$  è stata ottenuta dall'algoritmo Factor.

Il vettore pivot contiene gli scambi di riga effettuati da Factor.

Al termine il vettore  $b$  contiene la soluzione  $x$ .

*Parametri.* **Input:**  $n, A, \text{pivot}, b$

**Output:**  $b$

- 1: **ciclo 1:**  $k = 1, \dots, n - 1$
  - 2:    $j \leftarrow \text{pivot}(k)$
  - 3:   se  $j \neq k$  allora  $b_j \leftrightarrow b_k$
  - 4:   **ciclo 2:**  $i = k + 1, \dots, n$
  - 5:      $b_i \leftarrow b_i + a_{ik}b_k$
  - 6:   **fine ciclo 2**
  - 7: **fine ciclo 1**
  - 8:  $b_n \leftarrow b_n/a_{nn}$
  - 9: **ciclo 3:**  $i = n - 1, \dots, 1$
  - 10:    $b_i \leftarrow (b_i - \sum_{l=i+1}^n a_{il}b_l) / a_{ii}$
  - 11: **fine ciclo 3**
  - 12: **esci**
- 

La riformulazione del procedimento di Gauss in due fasi distinte (nella prima, la più onerosa dal punto di vista delle operazioni aritmetiche, trasformiamo la sola matrice  $A$ , mentre nella seconda costruiamo il vettore  $\bar{b} = Gb$  e risolviamo il sistema triangolare

(<sup>†</sup>) Ovvero note le matrici  $M_i$  e  $P_i$  che compongono  $G$ .

$Ux = \bar{b}$ ) ci consente di risolvere in modo efficiente sistemi del tipo

$$\begin{cases} Ax_1 = b_1 \\ Ax_2 = b_2 \\ \vdots \\ Ax_p = b_p \end{cases}$$

dove  $b_k = f(x_{k-1})$ ,  $k = 2, \dots, p$ . Infatti, poiché tutti i sistemi predetti hanno la stessa matrice  $A$ , possiamo operare la decomposizione  $GA = U$  una sola volta, riducendo il costo complessivo da  $pn^3/3$  operazioni aritmetiche a  $n^3/3 + pn^2$ :

- 1: Determina la composizione  $GA = U$  (Factor)
- 2: **ciclo 1:**  $k = 1, \dots, p - 1$
- 3:  $\bar{b}_k \leftarrow Gb_k$
- 4:  $Ux_k = \bar{b}_k \Rightarrow x_k$  (Solve)
- 5:  $b_{k+1} \leftarrow f(x_k)$
- 6: **fine ciclo 1**
- 7:  $\bar{b}_p \leftarrow Gb_p$
- 8:  $Ux_p = \bar{b}_p \Rightarrow x_p$  (Solve)

Con la notazione  $Ux_k = \bar{b}_k \Rightarrow x_k$  intendiamo: “risolvi il sistema nell’incognita  $x_k$ ”.

Supponiamo di aver già determinato la decomposizione di Gauss. La conoscenza delle matrici  $M_i, P_i$  ci consente di riordinare le successive trasformazioni di  $A$ :

$$M_{n-1}P_{n-1} \dots M_2P_2M_1P_1A = (\bar{M}_{n-1} \dots \bar{M}_2\bar{M}_1)(P_{n-1} \dots P_2P_1)A$$

così che, posto

$$\bar{M} = \bar{M}_{n-1} \dots \bar{M}_2\bar{M}_1 \quad \text{e} \quad P = P_{n-1} \dots P_2P_1$$

possiamo scrivere

$$\bar{M}PA = U$$

ovvero

$$PA = \bar{M}^{-1}U$$

Per dimostrare che la nostra osservazione è vera, occorre ricordare che  $P_i^{-1} = P_i$  e quindi utilizzare la proprietà

$$P_iB = B_1P_i \quad \text{con } B_1 = P_iBP_i$$

dove  $B$  è una generica matrice di ordine  $n$ . Infatti, esaminando a titolo di esempio il caso  $n = 5$ , abbiamo

$$\begin{aligned}
 M_4 P_4 M_3 P_3 M_2 P_2 M_1 P_1 A &= M_4 P_4 M_3 P_3 M_2 M_1^{(1)} P_2 P_1 A & M_1^{(1)} &= P_2 M_1 P_2 \\
 &= M_4 P_4 M_3 M_2^{(1)} P_3 M_1^{(1)} P_2 P_1 A & M_2^{(1)} &= P_3 M_2 P_3 \\
 &= M_4 P_4 M_3 M_2^{(1)} M_1^{(2)} P_3 P_2 P_1 A & M_1^{(2)} &= P_3 M_1^{(1)} P_3 \\
 &= M_4 M_3^{(1)} P_4 M_2^{(1)} M_1^{(2)} P_3 P_2 P_1 A & M_3^{(1)} &= P_4 M_3 P_4 \\
 &= M_4 M_3^{(1)} M_2^{(2)} P_4 M_1^{(2)} P_3 P_2 P_1 A & M_2^{(2)} &= P_4 M_2^{(1)} P_4 \\
 &= M_4 M_3^{(1)} M_2^{(2)} M_1^{(3)} P_4 P_3 P_2 P_1 A & M_1^{(3)} &= P_4 M_1^{(2)} P_4
 \end{aligned}$$

In generale avremo

$$M_{n-1} P_{n-1} M_{n-2} P_{n-2} \dots M_2 P_2 M_1 P_1 A = M_{n-1} M_{n-2}^{(1)} \dots M_2^{(n-3)} M_1^{(n-2)} P_{n-1} P_{n-2} \dots P_2 P_1 A$$

con

$$\bar{M}_i = M_i^{(n-1-i)} = P_{n-1} P_{n-2} \dots P_{i+1} M_i P_{i+1} \dots P_{n-2} P_{n-1}, \quad i = 1, \dots, n-2$$

$$\text{e } \bar{M}_{n-1} = M_{n-1}.$$

Le matrici  $M_i$  e  $\bar{M}_i$  sono esattamente dello stesso tipo e differiscono solo per un diverso ordinamento (nell'ambito della stessa colonna e sotto l'elemento diagonale) dei moltiplicatori  $m_{ij}$ , conseguenza degli scambi di riga effettuati in precedenza.

Posto

$$\bar{M}_j \dots \bar{M}_2 \bar{M}_1 = \left( I + \sum_{i=j+1}^n \bar{m}_{ij} e_i e_j^T \right) \dots \left( I + \sum_{i=3}^n \bar{m}_{i2} e_i e_2^T \right) \left( I + \sum_{i=2}^n \bar{m}_{i1} e_i e_1^T \right)$$

e osservato che

$$\left( I + \sum_{i=l+1}^n \bar{m}_{il} e_i e_l^T \right)^{-1} = I - \sum_{i=l+1}^n \bar{m}_{il} e_i e_l^T$$

un calcolo tutt'altro che difficile ci permette di verificare l'uguaglianza

$$(\bar{M}_j \dots \bar{M}_2 \bar{M}_1)^{-1} = I - \sum_{k=1}^j \sum_{i=k+1}^n \bar{m}_{ik} e_i e_k^T$$

e quindi di ottenere l'espressione

$$L = \bar{M}^{-1} = I - \sum_{k=1}^{n-1} \sum_{i=k+1}^n \bar{m}_{ik} e_i e_k^T = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ -\bar{m}_{21} & 1 & 0 & \dots & 0 \\ -\bar{m}_{31} & -\bar{m}_{32} & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\bar{m}_{n1} & -\bar{m}_{n2} & -\bar{m}_{n3} & \dots & 1 \end{pmatrix}$$

dove con  $(-\bar{m}_{j+1,j}, -\bar{m}_{j+2,j}, \dots, -\bar{m}_{nj})^T$  denotiamo un possibile diverso ordinamento degli elementi del vettore  $(-m_{j+1,j}, -m_{j+2,j}, \dots, -m_{nj})^T$ . Pertanto possiamo affermare che il metodo di Gauss può essere utilizzato anche per determinare una matrice di permutazione  $P$ , una matrice triangolare inferiore con diagonale unitaria  $L$  ed una matrice triangolare superiore  $U$ , tali che

$$(3.7) \quad PA = LU$$

La costruzione della matrice  $L$  è più semplice di quanto possa apparire a prima vista. Infatti, se memorizziamo i moltiplicatori  $-m_{ij}$  nelle corrispondenti posizioni  $(A)_{ij}$ , per ottenere  $L$  è sufficiente che ogni qual volta il procedimento di Gauss richiede lo scambio di due righe, tale scambio venga effettuato anche sui moltiplicatori memorizzati su tali righe. In particolare, l'algoritmo Factor di pagina 54 produrrà gli elementi essenziali di  $L$  ( $(L)_{ij}, i > j$ ) nella parte inferiore di  $A$  se, dopo aver posto al punto 10  $a_{ik} \leftarrow -m_{ik} = a_{ik}/a_{kk}$  e al punto 11  $a_{ij} \leftarrow a_{ij} - a_{ik}a_{kj}, j = k+1, \dots, n$ , al punto 7 abbiamo  $a_{kj} \leftarrow a_{ij}, j = 1, \dots, n$ . La matrice  $U$  in (3.7) invece è la stessa matrice della decomposizione  $GA = U$ .

Riprendendo l'esempio di pagina 53 ed applicando le predette varianti, abbiamo

$$\begin{aligned} \left( \begin{array}{cccc} 2 & -1 & 1 & -2 \\ 0 & 2 & 0 & -1 \\ 1 & 0 & -2 & 1 \\ -1 & 3 & 1 & 1 \end{array} \right) &\Rightarrow \left( \begin{array}{ccccc|ccccc} 2 & -1 & 1 & -2 & & & & & \\ 0 & 2 & 0 & -1 & & & & & \\ 1 & 1 & \frac{1}{2} & -\frac{5}{2} & 2 & & & & \\ \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & 2 & & & & & \\ -\frac{1}{2} & \frac{5}{2} & \frac{3}{2} & 0 & & & & & \end{array} \right) \Rightarrow \left( \begin{array}{ccccc|ccccc} 2 & -1 & 1 & -2 & & & & & \\ -\frac{1}{2} & \frac{5}{2} & \frac{3}{2} & 0 & & & & & \\ \frac{1}{2} & \frac{1}{2} & -\frac{5}{2} & 2 & & & & & \\ 0 & 2 & 0 & -1 & & & & & \end{array} \right) \\ &\Rightarrow \left( \begin{array}{ccccc|ccccc} 2 & -1 & 1 & -2 & & & & & \\ -\frac{1}{2} & \frac{5}{2} & \frac{3}{2} & 0 & & & & & \\ \frac{1}{2} & \frac{1}{2} & -\frac{14}{5} & 2 & & & & & \\ 0 & \frac{4}{5} & -\frac{6}{5} & -1 & & & & & \end{array} \right) \Rightarrow \left( \begin{array}{ccccc|ccccc} 2 & -1 & 1 & -2 & & & & & \\ -\frac{1}{2} & \frac{5}{2} & \frac{3}{2} & 0 & & & & & \\ \frac{1}{2} & \frac{1}{5} & -\frac{14}{5} & 2 & & & & & \\ 0 & \frac{4}{5} & \frac{3}{7} & -\frac{13}{7} & & & & & \end{array} \right) \end{aligned}$$

e quindi

$$L = \left( \begin{array}{cccc} 1 & 0 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 & 0 \\ \frac{1}{2} & \frac{1}{5} & 1 & 0 \\ 0 & \frac{4}{5} & \frac{3}{7} & 1 \end{array} \right), \quad U = \left( \begin{array}{cccc} 2 & -1 & 1 & -2 \\ 0 & \frac{5}{2} & \frac{3}{2} & 0 \\ 0 & 0 & -\frac{14}{5} & 2 \\ 0 & 0 & 0 & -\frac{13}{7} \end{array} \right)$$

Ricordiamo infine che la matrice  $P$  è univocamente individuata dal vettore pivot dello stesso algoritmo Factor.

Nota la fattorizzazione  $PA = LU$ , per determinare la soluzione del sistema  $Ax = b$  è sufficiente risolvere i seguenti due sistemi triangolari:

$$\begin{cases} Ly = Pb \\ Ux = y \end{cases}$$

che deduciamo immediatamente dalla relazione  $PAx = Pb$ , ovvero  $LUX = Pb$ .

Riesaminando quanto abbiamo finora detto sulla fattorizzazione  $LU$  possiamo affermare che quando il procedimento di Gauss non richiede scambi di righe abbiamo  $A = LU^{(\dagger)}$ ; inoltre questa decomposizione è unica e può venire riscritta nella forma

$$(3.8) \quad A = LDU_1$$

dove ora sia  $L$  che  $U_1$  sono triangolari con diagonali unitarie e  $D$  è diagonale con  $(D)_{ii} = (U)_{ii}$ ,  $i = 1, 2, \dots, n$ . Quando  $A$  è simmetrica abbiamo  $U_1 = L^T$ ; inoltre, gli elementi  $(D)_{ii}$  sono tutti positivi se e solo se  $A$  è definita positiva. Questo fatto ci consente di concludere con il seguente risultato:

**Teorema 3.2.** *Se  $A$  è simmetrica definita positiva, esiste un'unica matrice triangolare inferiore  $L_1$ , con elementi diagonale positivi, tale che*

$$(3.9) \quad A = L_1 L_1^T$$

Infatti, è sufficiente porre in (3.8)  $L_1 = LD^{1/2}$ , dove con  $D^{1/2}$  denotiamo la matrice diagonale che ha come elementi diagonale  $(D^{1/2})_{ii} = \sqrt{(D)_{ii}}$ ,  $i = 1, 2, \dots, n$ . Gli elementi della matrice

$$L_1 = \begin{pmatrix} l_{11} & 0 & 0 & \dots & 0 \\ l_{21} & l_{22} & 0 & \dots & 0 \\ l_{31} & l_{32} & l_{33} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ l_{n1} & l_{n2} & l_{n3} & \dots & l_{nn} \end{pmatrix}, \quad l_{ii} > 0$$

la cui esistenza e unicità è assicurata dal teorema precedente, possono venire determinati con le formule

$$i = 2, \dots, n : \quad \begin{cases} l_{11} = \sqrt{a_{11}} \\ l_{ij} = \left( a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk} \right) / l_{jj}, \quad j = 1, \dots, i-1 \\ l_{ii} = \left( a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2 \right)^{\frac{1}{2}} \end{cases}$$

---

<sup>(†)</sup> In questo caso  $L = G^{-1}$ .

facilmente deducibili dall'identità (3.9).

Questa decomposizione rappresenta il noto *metodo (o fattorizzazione) di Choleski*. Per la sua determinazione sono richieste  $n^3/6$  operazioni.

### 3.2.4 Raffinamento iterativo

Quando il sistema  $Ax = b$ , cui abbiamo applicato il metodo di Gauss (decomposizione  $GA = U$ ), non è troppo mal condizionato, il processo di raffinamento seguente elimina l'eventuale propagazione di errori dovuta ad una possibile non perfetta stabilità dell'algoritmo di Gauss (ma non elimina la propagazione dovuta al condizionamento del problema).

---

**Algoritmo 3:** Refine( $n, A, A_{\text{new}}, b, \text{pivot}, \tilde{x}, k_{\max}, \text{ier}$ )

---

*Commento.* Questo algoritmo può migliorare l'approssimazione  $\tilde{x} = x^{(0)}$ , della soluzione del sistema  $Ax = b$  di ordine  $n$ , fornita dagli algoritmi Factor e Solve.

$A_{\text{new}}$  e pivot sono le variabili output di Factor ( $A_{\text{new}}$  “contiene” le matrici  $G$  e  $U$  della decomposizione di Gauss).

Il numero massimo di iterazioni è  $k_{\max}$ . Se la precisione di macchina (eps) viene raggiunta, la variabile ier assume il valore 0; altrimenti ier = 1.

Con  $x^{(i)}$  e  $r^{(i)}$  denotiamo gli  $i$ -esimi “aggiornamenti” dei vettori  $\tilde{x}$  e  $r$ .

*Parametri.* **Input:**  $n, A, A_{\text{new}}, b, \text{pivot}, \tilde{x} = x^{(0)}, k_{\max}$   
**Output:**  $\tilde{x} = x^{(i+1)}, \text{ier}$

- 1:  $e \leftarrow x^{(0)}$
  - 2: **ciclo 1:**  $i = 0, \dots, k_{\max}$
  - 3:    $\delta \leftarrow b - Ax^{(i)}$  (in doppia precisione)
  - 4:   **richiama** Solve( $n, A_{\text{new}}, \text{pivot}, \delta$ )
  - 5:    $x^{(i+1)} \leftarrow x^{(i)} + \delta$
  - 6:   **se**  $\|\delta\| \|x^{(i)}\| \geq \|e\| \|x^{(i+1)}\|$  **allora**  $\text{ier} \leftarrow 1$ ; **esci**
  - 7:   **se**  $\|\delta\| \leq \text{eps} \|x^{(i+1)}\|$  **allora**  $\text{ier} \leftarrow 0$ ; **esci**
  - 8:    $e = \delta$
  - 9: **fine ciclo 1**
  - 10:  $\text{ier} \leftarrow 1$
  - 11: **esci**
- 

L'algoritmo scaturisce dall'osservazione seguente. Sia  $x = x^{(i)} + \delta x^{(i)}$  la soluzione esatta del sistema, così che

$$A(x^{(i)} + \delta x^{(i)}) = b$$

La correzione  $\delta x^{(i)}$  è a sua volta soluzione del sistema

$$A\delta x^{(i)} = r^{(i)}, \quad r^{(i)} = b - Ax^{(i)}$$

che ha come matrice dei coefficienti ancora  $A$ . Risolvendo quest'ultimo sistema (utilizzando la decomposizione  $GA = U$  precedentemente determinata con Factor nella fase di calcolo di  $x^{(0)}$ ), non troveremo  $\delta x^{(i)}$ , bensì una sua approssimazione  $\bar{\delta}x^{(i)}$ . Posto quindi  $x^{(i+1)} = x^{(i)} + \bar{\delta}x^{(i)}$ , ripetiamo l'operazione considerando  $x^{(i+1)}$ . In generale, se il sistema non è “troppo” mal condizionato, per esempio  $K_\infty(A) < 0.1/(n \cdot \text{eps})$ , il procedimento iterativo converge molto rapidamente; spesso già  $x^{(1)}$  o  $x^{(2)}$  danno la massima precisione raggiungibile. Se invece il sistema è “eccessivamente” mal condizionato il procedimento può addirittura divergere.

Ricordiamo che in realtà il problema cui applichiamo il metodo di Gauss, e quindi il raffinamento finale, è  $\bar{A}\bar{x} = \bar{b}$ , dove  $\bar{A}$  e  $\bar{b}$  sono le rappresentazioni di macchina (per esempio in precisione semplice) di  $A$  e  $b$ . Pertanto  $\bar{x}$  rappresenta la massima precisione raggiungibile partendo con i dati  $\bar{A}$  e  $\bar{b}$ . La successione  $x^{(i)}$  convergerà a  $\bar{x}$  e non a  $x$ . Quando il metodo di Gauss non si rivela perfettamente stabile, il raffinamento iterativo può essere utile per depurare l'approssimazione  $x^{(0)}$ , ottenuta con Gauss (utilizzando l'aritmetica di macchina), dagli errori introdotti dallo stesso algoritmo di Gauss.

Poiché il calcolo dei residui  $r^{(i)} = b - Ax^{(i)}$  comporta della cancellazione numerica, perché il processo di raffinamento abbia l'effetto auspicato è indispensabile che la differenza  $b - Ax^{(i)}$  sia valutata con aritmetica in doppia precisione (se, per esempio, tutte le altre operazioni aritmetiche, inclusa la determinazione di  $x^{(0)}$ , sono fatte in precisione semplice).

► **Esempi.** Risolviamo dapprima il sistema

$$\begin{pmatrix} 5 & 7 & 6 & 5 \\ 7 & 10 & 8 & 7 \\ 6 & 8 & 10 & 9 \\ 5 & 7 & 9 & 10 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 23 \\ 32 \\ 33 \\ 31 \end{pmatrix}$$

con il metodo di Gauss; otteniamo l'approssimazione

$$x^{(0)} = \begin{pmatrix} 0.9999940 \cdot 10^0 \\ 0.1000004 \cdot 10^1 \\ 0.1000002 \cdot 10^1 \\ 0.9999991 \cdot 10^0 \end{pmatrix}$$

Dopo una sola iterazione l'algoritmo Refine ci fornisce l'esatta soluzione del sistema  $x = (0.1000000 \cdot 10^1, \dots, 0.1000000 \cdot 10^1)^T$ .

Successivamente consideriamo la matrice di Wilkinson ([1.1, pag. 132])

$$A = \begin{pmatrix} 0.932165 & 0.443126 & 0.417632 \\ 0.712345 & 0.915312 & 0.887652 \\ 0.632165 & 0.514217 & 0.493909 \end{pmatrix}$$

e quindi il sistema

$$Ax = b$$

con  $b$  scelto in modo che risulti  $x = (1, 1, 1)^T$ . Il metodo di Gauss ci dà

$$x^{(0)} = \begin{pmatrix} 0.1000264 \cdot 10^1 \\ 0.9873875 \cdot 10^0 \\ 0.1012794 \cdot 10^1 \end{pmatrix}$$

La prima iterazione di Refine apporta un lieve miglioramento

$$x^{(1)} = \begin{pmatrix} 0.1000170 \cdot 10^1 \\ 0.9918556 \cdot 10^0 \\ 0.1008262 \cdot 10^1 \end{pmatrix}$$

mentre le successive lasciano inalterata quest'ultima approssimazione. Il vettore  $x^{(1)}$  rappresenta la corretta soluzione del sistema perturbato  $\bar{A}\bar{x} = \bar{b}$ , dove con  $\bar{A}$  e  $\bar{b}$  denotiamo le rappresentazioni di macchina (precisione semplice) di  $A$  e  $b$ . La perturbazione presente in  $x^{(1)}$  non è dovuta all'algoritmo di calcolo, ma al cattivo condizionamento del sistema; infatti abbiamo  $K_\infty(A) \cong 3 \times 10^5$ . ◀

### 3.2.5 Sistemi complessi

Un sistema del tipo

$$(3.10) \quad Cz = w$$

con  $C = A + iB$ ,  $z = x + iy$ ,  $w = u + iv$ ,  $A, B \in \mathbb{R}^{n \times n}$  e  $x, y, u, v \in \mathbb{R}^n$ , può essere risolto con il metodo di Gauss, sostituendo però le operazioni aritmetiche reali con le corrispondenti complesse. Ricordando che ogni numero complesso per la sua memorizzazione richiede 2 locazioni di memoria<sup>(†)</sup>, e che una moltiplicazione complessa comporta 4 moltiplicazioni e due addizioni reali, la soluzione del sistema in questione con il metodo di Gauss in aritmetica complessa richiede  $2n^2$  locazioni di memoria e  $4n^3/3$  operazioni reali.

Quando non si ha a disposizione l'aritmetica complessa è indispensabile risolvere il sistema (3.10) utilizzando la sola aritmetica reale. In questa situazione è necessario ricondurre dapprima il sistema alla seguente forma reale

$$(3.11) \quad \begin{cases} Ax - By = u \\ Bx + Ay = v \end{cases} \quad \text{ovvero} \quad \begin{pmatrix} A & -B \\ B & A \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} u \\ v \end{pmatrix}$$

e poi applicare il metodo di Gauss a quest'ultima. Tale modo di procedere risulta tuttavia più oneroso del precedente, sia dal punto di vista dello spazio di memoria che da quello del numero di operazioni; infatti sono richieste  $4n^2$  locazioni di memoria e  $8n^3/3$  operazioni aritmetiche.

---

(†) Per locazione di memoria intendiamo lo spazio di memoria necessario per la memorizzazione di un numero reale in aritmetica floating-point, cioè, nella maggioranza dei calcolatori, 32 bit (precisione semplice).

### 3.2.6 Matrice inversa

Esaminiamo dapprima il problema dell'inversione di una generica matrice (nonsingolare) di forma triangolare inferiore oppure superiore. Denotiamo con  $L$  la matrice

$$\begin{pmatrix} l_{11} & 0 & 0 & \dots & 0 \\ l_{21} & l_{22} & 0 & \dots & 0 \\ l_{31} & l_{32} & l_{33} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ l_{n1} & l_{n2} & l_{n3} & \dots & l_{nn} \end{pmatrix}, \quad l_{ii} \neq 0, \quad i = 1, \dots, n$$

e con  $Y$  la sua inversa, cioè  $Y = L^{-1}$ . Dalla teoria sappiamo che  $Y$  è anch'essa triangolare inferiore:

$$Y = \begin{pmatrix} y_{11} & 0 & 0 & \dots & 0 \\ y_{21} & y_{22} & 0 & \dots & 0 \\ y_{31} & y_{32} & y_{33} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ y_{n1} & y_{n2} & y_{n3} & \dots & y_{nn} \end{pmatrix}$$

Ricordando la definizione di prodotto di due matrici, e l'identità  $LY = I$ , per ogni intero  $j$ ,  $j = 1, \dots, n - 1$ , moltiplichiamo la  $i$ -esima riga di  $L$ ,  $i = j + 1, \dots, n$ , per la  $j$ -esima colonna di  $Y$ ; otteniamo

$$\left\{ \begin{array}{l} l_{jj}y_{jj} = 1 \\ l_{j+1,j}y_{jj} + l_{j+1,j+1}y_{j+1,j} = 0 \\ \dots \\ l_{nj}y_{jj} + l_{n,j+1}y_{j+1,j} + \dots + l_{nn}y_{nj} = 0 \end{array} \right.$$

ovvero

$y_{jj} \leftarrow 1/l_{jj}, \quad j = 1, \dots, n$   
**ciclo 1:**  $j = 1, \dots, n - 1$   
 $y_{ij} \leftarrow -(\sum_{k=j}^{i-1} l_{ik}y_{kj})/l_{ii}, \quad i = j + 1, \dots, n$   
**fine ciclo 1**

Il caso della matrice triangolare superiore è del tutto analogo al precedente. Posto

$$U = \begin{pmatrix} u_{11} & u_{12} & u_{13} & \dots & u_{1n} \\ 0 & u_{22} & u_{23} & \dots & u_{2n} \\ 0 & 0 & u_{33} & \dots & u_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & u_{nn} \end{pmatrix} \quad \text{e} \quad U^{-1} = Z = \begin{pmatrix} z_{11} & z_{12} & z_{13} & \dots & z_{1n} \\ 0 & z_{22} & z_{23} & \dots & z_{2n} \\ 0 & 0 & z_{33} & \dots & z_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & z_{nn} \end{pmatrix}, \quad UZ = I$$

per ogni intero  $j$ ,  $j = 1, \dots, n$ , moltiplichiamo la  $i$ -esima riga di  $U$ ,  $i = 1, \dots, j$ , per la  $j$ -esima colonna di  $Z$ ; otteniamo

$$\left\{ \begin{array}{l} u_{11}z_{1j} + u_{12}z_{2j} + \cdots + u_{1j}z_{jj} = 0 \\ \cdots \cdots \cdots \cdots \cdots \\ u_{j-1,j-1}z_{j-1,j} + u_{j-1,j}z_{jj} = 0 \\ u_{jj}z_{jj} = 1 \end{array} \right.$$

ovvero

$$z_{jj} \leftarrow 1/u_{jj}, \quad j = 1, \dots, n$$

**ciclo 1:**  $j = 2, \dots, n$

$$z_{ij} \leftarrow -\left(\sum_{k=i+1}^j u_{ik}z_{kj}\right)/u_{ii}, \quad i = j-1, \dots, 1$$

**fine ciclo 1**

L'inversione di una matrice triangolare, inferiore o superiore, di ordine  $n$  coinvolge  $n^3/6$  operazioni aritmetiche.

Consideriamo ora una generica matrice (quadrata di ordine  $n$ ) nonsingolare  $A$ , e osserviamo che la conoscenza della sua inversa  $A^{-1}$  ridurrebbe la soluzione del sistema  $Ax = b$  al semplice prodotto

$$x = A^{-1}b$$

il cui costo computazionale è, in generale, pari a  $n^2$  operazioni aritmetiche. Ciò sembra quindi suggerire il calcolo di  $A^{-1}$ , specialmente quando si presenta la necessità di dover risolvere più sistemi del tipo

$$Ax_i = b_i, \quad i = 1, 2, \dots, p$$

tutti con la stessa matrice  $A$ . Tuttavia, la soluzione  $x$  può essere determinata con un numero minore di operazioni aritmetiche, e generalmente con una precisione superiore, utilizzando la decomposizione di Gauss oppure la fattorizzazione  $LU$ . Infatti, come vedremo tra poco, il numero minimo di operazioni richiesto per il calcolo di  $A^{-1}$  è  $n^3$ , mentre le decomposizioni predette costano solamente  $n^3/3$  operazioni aritmetiche; inoltre, il prodotto  $A^{-1}b$  implica altre  $n^2$  operazioni, ossia lo stesso numero di operazioni necessarie per la risoluzione di

$$Ux = Gb \quad \text{oppure di} \quad \begin{cases} Ly = Pb \\ Ux = y \end{cases}$$

Osserviamo infine che l'inversa di una matrice sparsa è generalmente densa. Per esempio, nel caso della matrice tridiagonale simmetrica che segue:

$$\begin{pmatrix} 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 2 \end{pmatrix}$$

abbiamo

$$A^{-1} = \frac{1}{10} \begin{pmatrix} 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \\ 8 & 16 & 14 & 12 & 10 & 8 & 6 & 4 & 2 \\ 7 & 14 & 21 & 18 & 15 & 12 & 9 & 6 & 3 \\ 6 & 12 & 18 & 24 & 20 & 16 & 12 & 8 & 4 \\ 5 & 10 & 15 & 20 & 25 & 20 & 15 & 10 & 5 \\ 4 & 8 & 12 & 16 & 20 & 24 & 18 & 12 & 6 \\ 3 & 6 & 9 & 12 & 15 & 18 & 21 & 14 & 7 \\ 2 & 4 & 6 & 8 & 10 & 12 & 14 & 16 & 8 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{pmatrix}$$

Se la determinazione dell'inversa  $A^{-1}$  è veramente indispensabile, un metodo efficiente (ed ottimale dal punto di vista del numero di operazioni) può essere facilmente dedotto dalle stesse decomposizioni  $GA = U$  e  $PA = LU$ ; infatti, nel caso della decomposizione di Gauss abbiamo

$$A^{-1} = U^{-1}G = U^{-1}M_{n-1}P_{n-1} \dots M_2P_2M_1P_1$$

mentre dalla fattorizzazione  $LU$  deduciamo la formula

$$A^{-1} = U^{-1}L^{-1}P = (U^{-1}L^{-1})P_{n-1} \dots P_2P_1$$

In entrambi i casi il costo complessivo del calcolo di  $A^{-1}$  è di  $n^3$  operazioni aritmetiche.

Ovviamente quando le due formule suddette verranno implementate in un calcolatore esse forniranno solo un'approssimazione  $X^{(0)}$  dell'inversa  $A^{-1}$ . Supponendo il problema non troppo mal condizionato, cioè  $K(A)$  non eccessivamente grande, possiamo migliorare  $X^{(0)}$  con il seguente procedimento iterativo<sup>(†)</sup>. Sia  $X^{(k)}$  un'approssimazione di  $A^{-1}$ , e definiamo  $E^{(k)} = A^{-1} - X^{(k)}$ ; dall'identità  $AA^{-1} = I$  otteniamo

$$A(X^{(k)} + E^{(k)}) = I$$

e quindi

$$AE^{(k)} = I - AX^{(k)} \equiv R^{(k)}$$

---

<sup>(†)</sup> Simile al raffinamento iterativo introdotto a pagina 60 per migliorare la soluzione di  $Ax = b$  fornita dal metodo di Gauss.

Quest'ultima relazione ci consente dapprima di scrivere

$$E^{(k)} = A^{-1}R^{(k)} \cong X^{(k)}R^{(k)}$$

e poi

$$X^{(k+1)} = X^{(k)} + X^{(k)}R^{(k)}$$

Quando l'approssimazione iniziale  $X^{(0)}$  è sufficientemente buona, nel senso che  $\|I - AX^{(0)}\| < 1$ , è possibile dimostrare che, in assenza di errori nella rappresentazione dei numeri, la successione  $\{X^{(k)}\}$  converge a  $A^{-1}$ . Tuttavia, anche in questo caso valgono considerazioni del tutto analoghe a quelle fatte a pagina 60 per il raffinamento iterativo della “soluzione” di sistemi lineari. In particolare, nell'implementazione dell'algoritmo, con aritmetica in precisione semplice per esempio, è indispensabile che il calcolo del “residuo”  $I - AX^{(k)}$  sia effettuato con aritmetica in doppia precisione.

### 3.3 Metodi iterativi

In alcune situazioni, per esempio nella soluzione di equazioni alle derivate parziali di tipo ellittico con metodi alle differenze finite o agli elementi finiti, i sistemi da risolvere sono sparsi e di dimensioni tali ( $n = 10^3 \div 10^6$ ) da rendere inutilizzabile, o quanto meno inefficiente, il metodo di Gauss anche con i moderni calcolatori. Infatti, mentre in questi casi la matrice iniziale ha un numero di elementi non nulli  $p \ll n^2$ , il processo delle eliminazioni successive del metodo di Gauss cambia le equazioni del sistema ad ogni passo, cosicché la matrice dei coefficienti può diventare sempre meno sparsa e richiedere quindi la memorizzazione di un numero eccessivo di elementi.

I metodi iterativi invece non alterano mai la matrice iniziale  $A$ . Partendo da un'approssimazione iniziale  $x^{(0)}$  e utilizzando sempre e solo gli elementi non nulli di  $A$ , essi definiscono una successione di approssimazioni  $x^{(1)}, x^{(2)}, \dots$  convergente, sotto opportune ipotesi, alla soluzione  $x$  del sistema non singolare<sup>(†)</sup>

$$(3.12) \quad Ax = b$$

Una nota classe di metodi iterativi può venire definita decomponendo dapprima la matrice  $A$  nella forma

$$(3.13) \quad A = D + C$$

(questo “sdoppiamento” può essere fatto in una infinità di modi), e riscrivendo poi il sistema (3.13) come segue:

$$Dx = -Cx + b$$

Partendo da un generico vettore  $x^{(0)}$  possiamo costruire la successione  $x^{(1)}, x^{(2)}, \dots$  innescando il procedimento iterativo

$$(3.14) \quad Dx^{(k+1)} = d^{(k)} \quad k = 0, 1, \dots$$

---

<sup>(†)</sup> O meglio, alla soluzione  $\bar{x}$  del sistema perturbato  $\bar{A}\bar{x} = \bar{b}$ .

dove  $d^{(k)} = -Cx^{(k)} + b$ .

Poiché il calcolo di  $x^{(k+1)}$  comporta la soluzione del sistema (3.14), è chiaro che la scelta della matrice  $D$  deve essere fatta in modo che  $\det(D) \neq 0$  e, soprattutto, il sistema (3.14) risulti facilmente risolvibile.

Esaminiamo dapprima la convergenza della successione  $x^{(1)}, x^{(2)}, \dots$  così costruita. Dalla relazione

$$Dx^{(k+1)} = -Cx^{(k)} + b$$

otteniamo

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

Ponendo

$$B = -D^{-1}C = I - D^{-1}A$$

la relazione precedente assume la nuova forma

$$(3.15) \quad x^{(k+1)} = Bx^{(k)} + D^{-1}b$$

Consideriamo  $x^{(k+1)}$  e l'errore assoluto  $e^{(k+1)}$  ad esso associato:

$$\begin{aligned} e^{(k+1)} &= x - x^{(k+1)} = (Bx + D^{-1}b) - (Bx^{(k)} + D^{-1}b) \\ &= B(x - x^{(k)}) = Be^{(k)}, \quad k = 0, 1, 2, \dots \end{aligned}$$

Possiamo allora scrivere

$$\begin{aligned} e^{(0)} &= x - x^{(0)} \\ e^{(1)} &= Be^{(0)} \\ e^{(2)} &= Be^{(1)} = BB^{(0)} = B^2e^{(0)} \\ e^{(3)} &= Be^{(2)} = BB^2e^{(0)} = B^3e^{(0)} \\ &\dots \\ e^{(k+1)} &= Be^{(k)} = BB^ke^{(0)} = B^{k+1}e^{(0)} \end{aligned}$$

A questo punto ricordiamo il seguente teorema:

**Teorema 3.3.** ([3.20, pag. 232]). *Sia  $B$  una matrice quadrata di ordine  $n$ . Allora*

$$\lim_{m \rightarrow \infty} B^m = O \quad (\text{matrice nulla})$$

se e solo se  $\rho(B) < 1$ , dove  $\rho(B)$  denota il raggio spettrale di  $B$ .

Quest'ultimo risultato ci permette pertanto di affermare che:

il processo iterativo (3.15) è convergente, cioè  $\lim_{k \rightarrow \infty} e^{(k)} = o$ , se e solo se

$$\rho(I - D^{-1}A) < 1$$

Ricordiamo inoltre che *per ogni norma di matrice compatibile con una di vettore abbiamo*

$$\rho(B) \leq \|B\|$$

Pertanto, il metodo (3.15) è senz'altro convergente quando  $\|I - D^{-1}A\| < 1$ . Per esempio, il metodo risulta convergente se  $\|B\|_1$  oppure  $\|B\|_\infty$  (in qualche caso facilmente calcolabili) è minore di 1.

L'idea di sdoppiare la matrice  $A$  nella forma  $A = D + C$  conduce alla costruzione di un metodo iterativo, definito dalla (3.15), che comporta ad ogni passo la soluzione del sistema non singolare (3.14). Ovviamente, affinché questo procedimento possa presentare dei vantaggi, il sistema (3.14) deve risultare più “semplice” del sistema di partenza  $Ax = b$ . Inoltre, come abbiamo visto sopra, è necessario che  $\rho(I - D^{-1}A) < 1$ . Tra le infinite scelte di  $D$  ci limiteremo pertanto a quelle che rispondono ai seguenti requisiti:

- (i)  $\det(D) \neq 0$ ;
- (ii)  $D$  diagonale o triangolare;
- (iii) l'insieme delle matrici  $A$  per cui  $\|I - D^{-1}A\| < 1$ , almeno per una norma naturale, non è vuoto.

### 3.3.1 Metodo di Jacobi

Scegliamo

$$D = \begin{pmatrix} a_{11} & 0 & 0 & \dots & 0 \\ 0 & a_{22} & 0 & \dots & 0 \\ 0 & 0 & a_{33} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & a_{nn} \end{pmatrix}, \quad C = A - D$$

supponendo  $a_{ii} \neq 0$ ,  $i = 0, 1, 2, \dots, n$ . Se qualche  $a_{ii}$  risulta nullo, allora prima di procedere allo sdoppiamento dobbiamo permutare le righe di  $A$  in modo che i nuovi elementi diagonale siano tutti non nulli. Quando  $A$  è non singolare ciò è sempre possibile. Ovviamente tali scambi devono essere fatti anche sui corrispondenti elementi di  $b$ .

Una condizione sufficiente per la convergenza facile da verificare è la seguente:

$$(3.16) \quad \|I - D^{-1}A\|_\infty = \max_{1 \leq i \leq n} \sum_{\substack{j=1 \\ j \neq i}}^n \left| \frac{a_{ij}}{a_{ii}} \right| < 1$$

ovvero

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

Per meglio capire il metodo di Jacobi, scriviamo il sistema  $Ax = b$  esplicitamente

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

con le equazioni ordinate in modo che  $a_{ii} \neq 0$ ,  $i = 1, 2, \dots, n$ . Tale sistema può anche essere riscritto nella forma

$$x_i = \frac{b_i - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij}x_j}{a_{ii}}, \quad i = 1, 2, \dots, n$$

Il metodo di Jacobi consiste nel calcolare, nota un'approssimazione iniziale  $x^{(0)}$  (oppure prendendo  $x^{(0)} = o$ ), le approssimazioni successive  $x^{(k+1)} = (x_1^{(k+1)}, x_2^{(k+1)}, \dots, x_n^{(k+1)})^T$ ,  $k = 0, 1, 2, \dots$ , per mezzo della relazione

$$(3.17) \quad x_i^{(k+1)} = \frac{b_i - \sum_{\substack{j=1 \\ j \neq i}}^n a_{ij}x_j^{(k)}}{a_{ii}}, \quad i = 1, 2, \dots, n$$

### 3.3.2 Metodo di Gauss-Seidel

Nel metodo di Jacobi ogni singola componente di  $x^{(k+1)}$  dipende unicamente dall'approssimazione precedente  $x^{(k)}$ . Calcolata  $x_1^{(k+1)}$ , potremmo già utilizzare questo nuovo valore nella determinazione di  $x_2^{(k+1)}$ , e poi utilizzare  $x_1^{(k+1)}$  e  $x_2^{(k+1)}$  (più  $x_4^{(k)}, \dots, x_n^{(k)}$ ) nel calcolo di  $x_3^{(k+1)}$ , e così via. Questo procedimento rappresenta il metodo di Gauss-Seidel

$$(3.18) \quad x_i^{(k+1)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)}}{a_{ii}}, \quad i = 1, 2, \dots, n(\dagger)$$

---

#### Algoritmo 4: Gseidel( $n, A, b, \text{tol}, k_{\max}, x, \text{ier}$ )

---

*Commento.* L'algoritmo, utilizzando il processo iterativo di Gauss-Siedel, migliora l'approssimazione iniziale  $x = x^{(0)}$  della soluzione del sistema non singolare, di ordine  $n$ ,  $Ax = b$ . Le successive approssimazioni vengono memorizzate nel vettore  $x$ . Se la precisione richiesta (errore relativo)  $\text{tol} (> \text{eps})$  è raggiunta con un numero di iterazioni  $\leq k_{\max}$ , la variabile  $\text{ier}$  assume il valore 0; altrimenti  $\text{ier} = 1$ .

(†) I simboli  $\sum_{j=1}^0$  e  $\sum_{j=n+1}^n$  vanno intesi come “sommatorie vuote”.

*Parametri.* **Input:**  $n, A, b, \text{toll}, k_{\max}, x$   
**Output:**  $x, \text{ier}$

---

```

1: ciclo 1:  $k = 1, \dots, k_{\max}$ 
2:    $y \leftarrow x_1$ 
3:    $x_1 \leftarrow (b_1 - \sum_{j=2}^n a_{1j}x_j)/a_{11}$ 
4:    $x_{\max} \leftarrow |x_1|$ 
5:    $\text{ermax} \leftarrow |y - x_1|$ 
6:   ciclo 2:  $i = 2, \dots, n$ 
7:      $y \leftarrow x_i$ 
8:      $x_i \leftarrow (b_i - \sum_{j=1}^{i-1} a_{ij}x_j - \sum_{j=i+1}^n a_{ij}x_j)/a_{ii}$ 
9:      $\text{er} \leftarrow |y - x_i|$ 
10:    se  $x_{\max} < |x_i|$  allora  $x_{\max} \leftarrow |x_i|$ 
11:    se  $\text{ermax} < \text{er}$  allora  $\text{ermax} \leftarrow \text{er}$ 
12:   fine ciclo 2
13:   se  $\text{ermax} < \text{toll} \cdot x_{\max}$  allora  $\text{ier} \leftarrow 0$ ; esci
14: fine ciclo 1
15:  $\text{ier} \leftarrow 1$ 
16: esci

```

---

Un rapido esame di questo secondo metodo ci consente di concludere che quanto abbiamo fatto è, sostanzialmente, uno sdoppiamento di tipo (3.13) della matrice  $A$ , dove come  $D$  abbiamo scelto la matrice seguente:

$$D = \begin{pmatrix} a_{11} & 0 & 0 & \dots & 0 \\ a_{21} & a_{22} & 0 & \dots & 0 \\ a_{31} & a_{32} & a_{33} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{pmatrix}, \quad a_{ii} \neq 0, \quad i = 1, \dots, n$$

Anche in questo caso, in mancanza di un'approssimazione iniziale prendiamo  $x^{(0)} = o$ .

Una condizione sufficiente per la convergenza del processo iterativo è la stessa vista per il metodo di Jacobi, cioè

$$(3.19) \quad \|I - D^{-1}A\|_{\infty} \leq \max_{1 \leq i \leq n} \sum_{\substack{j=1 \\ j \neq i}}^n \left| \frac{a_{ij}}{a_{ii}} \right| < 1$$

Le condizioni (3.16) e (3.19) ci garantiscono che i metodi di Jacobi e di Gauss-Seidel sicuramente convergono quando  $A$  è una matrice a diagonale dominante. Entrambi i metodi potrebbero però convergere anche quando  $A$  non è a diagonale dominante.

La convergenza di Jacobi e di Gauss-Seidel è assicurata anche quando  $A$  è a *diagonale dominante per colonne*, ossia

$$|a_{kk}| > \sum_{\substack{i=1 \\ i \neq k}}^n |a_{ik}|, \quad k = 1, \dots, n$$

Per la classe di matrici che ora definiamo, le condizioni di convergenza predette possono essere lievemente attenuate.

Una matrice  $A$  è detta *irriducibile* se non esiste alcuna matrice di permutazione  $P$  tale che

$$P^T A P = \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \\ O & \bar{A}_{22} \end{pmatrix}$$

dove  $\bar{A}_{11}$  e  $\bar{A}_{12}$  sono entrambe quadrate. Quando  $A$  è irriducibile, la convergenza dei metodi di Jacobi e di Gauss-Seidel è ancora assicurata se

$$|a_{ii}| \geq \sum_{\substack{k=1 \\ k \neq i}}^n |a_{ik}|, \quad i = 1, \dots, n$$

oppure

$$|a_{kk}| \geq \sum_{\substack{i=1 \\ i \neq k}}^n |a_{ik}|, \quad k = 1, \dots, n$$

con diseguaglianza stretta valida almeno per un valore degli indici  $i$  (per la prima) o  $k$  (per la seconda).

È possibile dimostrare che quando  $A$  è simmetrica definita positiva il metodo di Gauss-Seidel risulta convergente.

La convergenza del metodo di Gauss-Seidel non implica quella di Jacobi, e viceversa. Tuttavia, quando entrambi convergono, la velocità di convergenza di Gauss-Seidel è generalmente superiore.

Nei due esempi che seguono mettiamo a confronto le prestazioni fornite dai metodi di Jacobi e di Gauss-Seidel quando come approssimazione iniziale si prende il vettore nullo.

### ► Esempio 3.1.

$$A = \begin{pmatrix} 5 & 7 & 6 & 5 \\ 7 & 10 & 8 & 7 \\ 6 & 8 & 10 & 9 \\ 5 & 7 & 9 & 10 \end{pmatrix}, \quad b = \begin{pmatrix} 23 \\ 32 \\ 33 \\ 31 \end{pmatrix}$$

La matrice  $A$  è simmetrica definita positiva, e la soluzione del sistema  $Ax = b$  è  $x = (1, 1, 1, 1)^T$ .

Il metodo di Jacobi non converge, mentre quello di Gauss-Seidel, la cui convergenza è assicurata dalle proprietà di  $A$ , produce le approssimazioni riportate in tabella 3.2.

$k$	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$x_4^{(k)}$
1	4.5999999	$-2.0000076 \cdot 10^{-2}$	0.5560001	0.3136000
2	3.6471999	$-1.7360115 \cdot 10^{-2}$	0.8433282	0.5295566
3	3.0827537	$-3.2796864 \cdot 10^{-3}$	0.9763706	0.6821854
4	2.7507613	$1.5840912 \cdot 10^{-2}$	1.0229034	0.7929176
5	2.5574210	$3.6440276 \cdot 10^{-2}$	1.0227696	0.8752887
6	2.4463713	$5.6622312 \cdot 10^{-2}$	0.9991196	0.9379711
7	2.3838141	$7.5454712 \cdot 10^{-2}$	0.9651737	0.9866183
8	2.3495364	$9.2552759 \cdot 10^{-2}$	0.9282795	1.0249932
9	2.3314974	0.1078331	0.8923413	1.0556611
10	2.3225634	0.1213698	0.8592713	1.0804154
11	2.3185415	0.1333132	0.8298508	1.1005443
12	2.3169961	0.1438412	0.8042397	1.1169975
13	2.3165371	0.1531340	0.7822728	1.1304922
14	2.3163929	0.1613623	0.7636315	1.1415818
15	2.3161533	0.1686802	0.7479405	1.1507008
16	2.3156185	0.1752241	0.7348191	1.1581967
17	2.3147068	0.1811125	0.7239090	1.1643498
18	2.3134017	0.1864470	0.7148865	1.1693884
19	2.3117218	0.1913138	0.7074665	1.1734997
20	2.3097012	0.1957861	0.7014004	1.1768389
:				
50	2.2032538	0.2767151	0.6946560	1.1794821
:				
100	2.0301588	0.3808355	0.7384396	1.1537402
:				
200	1.7550234	0.5462017	0.8082982	1.1126787
:				
400	1.4055747	0.7562342	0.8970236	1.0605276
:				
1000	1.0628636	0.9622166	0.9840385	1.0093820

Tabella 3.2



► **Esempio 3.2.**

$$A = \begin{pmatrix} 3 & 1 & -1 \\ 2 & 6 & 2 \\ 1 & 2 & 4 \end{pmatrix}, \quad b = \begin{pmatrix} 3 \\ 10 \\ 7 \end{pmatrix}$$

Le approssimazioni prodotte sono riportate in tabella 3.3.

Jacobi				Gauss-Siedel			
$k$	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$k$	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
1	1.0000000	1.6666666	1.7500000	1	1.0000000	1.3333334	0.8333333
2	1.0277778	0.7500000	0.6666668	2	0.8333333	1.1111112	0.9861111
3	0.9722223	1.1018518	1.1180556	3	0.9583333	1.0185186	1.0011574
4	1.0054013	0.9699073	0.9560185	4	0.9942129	1.0015432	1.0006752
5	0.9953704	1.0128601	1.0136960	5	0.9997107	0.9998714	1.0001366
6	1.0002786	0.9969778	0.9947274	6	1.0000885	0.9999250	1.0000154
7	0.9992498	1.0016646	1.0014415	7	1.0000302	0.9999848	1.0000000
8	0.9999256	0.9997695	0.9993552	8	1.0000050	0.9999983	0.9999996
9	0.9998619	1.0002397	1.0001338	9	1.0000005	1.0000000	0.9999999
10	0.9999647	1.0000014	0.9999147	10	0.9999999	1.0000001	0.9999999
11	0.9999711	1.00000402	1.6000081	11	0.9999999	1.0000000	1.0000000
12	0.9999893	1.0000069	1.9999871	12	1.0000000	1.0000000	1.0000000
13	0.9999934	1.0000079	0.9999992				
14	0.9999971	1.0000025	0.9999977				
15	0.9999984	1.0000017	0.9999995				
16	0.9999993	1.0000007	0.9999996				
17	0.9999996	1.0000004	0.9999998				
18	0.9999998	1.0000001	0.9999999				
19	0.9999999	1.0000001	1.0000000				
20	0.9999999	1.0000000	0.9999999				
21	1.0000000	1.0000000	1.0000000				

Tabella 3.3



### 3.3.3 Metodo di sovrarilassamento (SOR)

Riprendiamo il metodo di Gauss-Seidel. Dalla relazione (3.18), sottraendo  $x_i^{(k)}$  da ambo i membri, otteniamo

$$(3.20) \quad r_i^{(k)} = x_i^{(k+1)} - x_i^{(k)} = \frac{1}{a_{ii}} \left[ b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i}^n a_{ij}x_j^{(k)} \right], \quad i = 1, \dots, n$$

dove  $r_i^{(k)}$  rappresenta la correzione da apportare a  $x_i^{(k)}$  per ottenere la nuova approssimazione

$$(3.21) \quad x_i^{(k+1)} = x_i^{(k)} + r_i^{(k)}, \quad k = 0, 1, 2, \dots$$

Le relazioni (3.20) e (3.21) definiscono ancora il metodo di Gauss-Seidel; tuttavia la forma (3.21) ci suggerisce l'introduzione di un parametro  $\omega$

$$(3.22) \quad x_i^{(k+1)} = x_i^{(k)} + \omega r_i^{(k)}$$

al fine di migliorare la correzione da effettuare su  $x_i^{(k)}$ . Occorrerà scegliere  $\omega$  in modo da accelerare il più possibile la convergenza della successione  $\{x^{(k)}\}$ . La (3.22), con  $r_i^{(k)}$  definito dalla (3.24), definisce un nuovo metodo, detto di *rilassamento*. Quando  $\omega = 1$  il metodo si identifica con quello di Gauss-Seidel.

Prima di proseguire osserviamo che la formula (3.18) può essere scritta in forma più compatta

$$x^{(k+1)} = D^{-1}(b - Lx^{(k+1)} - Ux^{(k)}), \quad L + D + U = A$$

dove

$$L = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ a_{21} & 0 & 0 & \dots & 0 & 0 \\ a_{31} & a_{32} & 0 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n-1,1} & a_{n-1,2} & a_{n-1,3} & \dots & 0 & 0 \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{n,n-1} & 0 \end{pmatrix}$$

$$D = \begin{pmatrix} a_{11} & & & & & \\ & a_{22} & & & & \\ & & a_{33} & & & \\ & 0 & & \ddots & & \\ & & & & a_{n-1,n-1} & \\ & & & & & a_{nn} \end{pmatrix}, \quad U = \begin{pmatrix} 0 & a_{12} & a_{13} & \dots & a_{1n} \\ 0 & 0 & a_{23} & \dots & a_{2n} \\ 0 & 0 & 0 & \dots & a_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix}$$

Con queste nuove notazioni le (3.20) e (3.22) possono essere riscritte nel modo seguente:

$$(3.23) \quad r^{(k)} = D^{-1}[b - Lx^{(k+1)} - (D + U)x^{(k)}]$$

$$(3.24) \quad (D + \omega L)x^{(k+1)} = [(1 - \omega)D - \omega U]x^{(k)} + \omega b$$

Ricordando la rappresentazione (3.15) e il successivo teorema, possiamo affermare che il nuovo metodo risulterà convergente se e solo se

$$\rho(B_{\text{SOR}}) < 1, \quad B_{\text{SOR}} = (D + \omega L)^{-1}[(1 - \omega)D - \omega U]$$

inoltre, la convergenza sarà tanto più rapida quanto più piccolo risulterà il raggio spettrale  $\rho(B_{SOR})$ . Dovendo scegliere il parametro  $\omega$ , converrà pertanto prendere quel valore  $\omega_{opt}$  che rende  $\rho(B_{SOR})$  minimo. Il calcolo di  $\omega_{opt}$  non è affatto semplice. Per certe classi di matrici che si presentano nella risoluzione numerica di equazioni alle derivate parziali di tipo ellittico con metodi alle differenze finite è possibile determinare delle stime aprioristiche del valore ottimo di  $\omega$ .

Per esempio, nel caso di matrici  $A$  simmetriche definite positive e tridiagonali a blocchi, di forma

$$A = \begin{pmatrix} D_1 & U_1 & & & \\ L_1 & D_2 & U_2 & & \\ & L_2 & D_3 & U_3 & \\ & & L_3 & \ddots & \ddots \\ & & & \ddots & \ddots & U_{n-1} \\ & & & & L_{n-1} & D_n \end{pmatrix}$$

dove le matrici  $D_i$  sono diagonali, per il valore ottimo del parametro  $\omega$  abbiamo la seguente rappresentazione (vedi [3.1])

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - \rho(B_{GS})}}, \quad \rho(B_{GS}) < 1$$

dove  $\rho(B_{GS})$  denota il raggio spettrale della matrice di iterazione  $B = I - D^{-1}A$  relativa al metodo di Gauss-Seidel. Inoltre abbiamo

$$\rho(B_{GS}) = \rho^2(B_J)$$

essendo  $\rho(B_J)$  il raggio spettrale della matrice  $B$  di iterazione del metodo di Jacobi, e

$$\min_{\omega} \rho(B_{SOR}) = \omega_{opt} - 1$$

Per la determinazione di  $\omega_{opt}$ , anche in altre situazioni, vedasi [3.8] e [9.1].

È stato dimostrato (vedere ad esempio [3.1]) che quando  $\omega \leq 0$  oppure  $\omega \geq 2$  abbiamo  $\rho(B_{SOR}) \geq 1$  e quindi il metodo (3.22) non può convergere. Nel caso di matrici  $A$  simmetriche definite positive il metodo converge per ogni valore  $0 < \omega < 2$  (vedere [3.20]).

Il metodo è detto di *sottilassamento* se  $0 < \omega < 1$ , e di *sovralassamento* (SOR) se  $1 < \omega < 2$ .

Nell'esempio che segue applichiamo il metodo SOR al sistema  $Ax = b$ , dove  $A$  è la matrice di pagina 38 e  $b = (2, 1, 1, 2, 1, 0, 0, 1, 2, 1, 1, 2)^T$ , prendendo prima  $\omega = 1$  e poi  $\omega = \omega_{opt} \simeq 1.235$ , e riportiamo i corrispondenti errori  $\|x - x^{(k)}\|_\infty$ .

$k$	$\omega = 1$	$\omega = 1.235$
1	$8.7 \cdot 10^{-1}$	$7.6 \cdot 10^{-1}$
2	$5.6 \cdot 10^{-1}$	$3.6 \cdot 10^{-1}$
3	$3.6 \cdot 10^{-1}$	$1.6 \cdot 10^{-1}$
4	$2.1 \cdot 10^{-1}$	$6.2 \cdot 10^{-2}$
5	$1.3 \cdot 10^{-1}$	$2.2 \cdot 10^{-2}$
6	$7.2 \cdot 10^{-2}$	$2.2 \cdot 10^{-3}$
7	$4.2 \cdot 10^{-2}$	$6.8 \cdot 10^{-4}$
8	$2.4 \cdot 10^{-2}$	$1.1 \cdot 10^{-4}$
9	$1.4 \cdot 10^{-2}$	$3.0 \cdot 10^{-5}$
10	$8.0 \cdot 10^{-3}$	$1.1 \cdot 10^{-5}$
11	$4.6 \cdot 10^{-3}$	$2.2 \cdot 10^{-6}$
12	$2.6 \cdot 10^{-3}$	$2.4 \cdot 10^{-7}$
13	$1.5 \cdot 10^{-3}$	$1.2 \cdot 10^{-7}$
14	$8.6 \cdot 10^{-4}$	
15	$5.0 \cdot 10^{-4}$	
20	$3.1 \cdot 10^{-5}$	
30	$1.2 \cdot 10^{-7}$	

Tabella 3.4

Osserviamo infine che poiché i metodi iterativi vengono applicati a sistemi sparsi, nelle formule (3.17), (3.18) e (3.22) molti degli elementi  $a_{ij}$  sono nulli. Pertanto, se per esempio la matrice  $A$  (di ordine  $n$ ) ha solo  $p \ll n^2$  elementi diversi da zero, ogni singola iterazione richiede  $n$  divisioni,  $p - 1$  moltiplicazioni e  $p - 1$  addizioni. Denotando con  $k$  il numero di iterazioni necessarie per raggiungere la precisione richiesta, il processo iterativo risulterà competitivo<sup>(†)</sup> con il metodo di Gauss quando

$$k < \frac{n^3}{3(p-1)}$$

### 3.3.4 Metodo del gradiente coniugato

Nel metodo SOR presentato nel paragrafo precedente la scelta del parametro di accelerazione  $\omega$  in generale è tutt'altro che semplice. Il metodo iterativo del gradiente coniugato che ci accingiamo a descrivere non presenta questa difficoltà e ci consente di risolvere in modo efficiente sistemi simmetrici definiti positivi.

Osserviamo preliminarmente che la soluzione del sistema  $Ax = b$ , dove la matrice  $A$  è supposta simmetrica definita positiva, è equivalente alla minimizzazione del funzionale

---

(†) In termini di operazioni aritmetiche. Dal punto di vista dell'occupazione di memoria a volte l'approccio iterativo è indispensabile.

quadratico

$$(3.25) \quad \Phi(x) = \frac{1}{2}x^T Ax - x^T b$$

Infatti il gradiente di  $\Phi$  è  $\nabla\Phi(x) = Ax - b$ , ed esso risulta nullo se e solo se  $Ax = b$ . Il minimo di  $\Phi$  è quindi unico e viene ottenuto ponendo  $x = A^{-1}b$ .

Possiamo pertanto risolvere il sistema  $Ax = b$  determinando il valore di  $x$  che minimizza il funzionale (3.25). La strategia più semplice per calcolare il minimo di  $\Phi$  consiste nel seguire localmente la direzione di massima pendenza, definita dal vettore  $-\nabla\Phi$ , lungo la quale la funzione  $\Phi$  decresce più rapidamente.

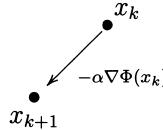
Dato pertanto un punto  $x_k$ <sup>(†)</sup>, se  $\nabla\Phi(x_k) = o$  allora  $x_k$  è la soluzione cercata. Se invece  $\nabla\Phi(x_k) \neq o$ , definito il residuo generato da  $x_k$

$$r_k = b - Ax_k = -\nabla\Phi(x_k)$$

esiste certamente un reale positivo  $\alpha$  tale che  $\Phi(x_k + \alpha r_k) < \Phi(x_k)$ . Con il *metodo della massima pendenza* definiamo

$$x_{k+1} = x_k - \alpha \nabla\Phi(x_k) = x_k + \alpha r_k$$

e scegliamo il parametro  $\alpha$  in modo che  $\Phi(x_{k+1})$  sia minimo:



Il valore ottimo di  $\alpha$  è dato dall'espressione:

$$\alpha = \alpha_k = \frac{r_k^T r_k}{r_k^T A r_k} > 0$$

Poiché

$$r_{k+1} = b - Ax_{k+1} = b - A(x_k + \alpha_k r_k) = r_k - \alpha_k A r_k$$

risulta

$$r_{k+1}^T r_k = 0$$

il che significa che  $r_{k+1}$  è ortogonale a  $r_k$ .

---

(†) Per semplificare le notazioni, in questo paragrafo, con i simboli  $x_k, r_k, p_k, z_k, k = 0, 1, \dots$ , denotiamo dei vettori.

Il metodo può quindi essere riassunto dal seguente algoritmo:

- 1:  $x_0 \leftarrow o$
- 2:  $r_0 \leftarrow b$
- 3: **ciclo 1:**  $k = 0, \dots, k_{\max}$
- 4:    $\alpha_k \leftarrow r_k^T r_k / r_k^T A r_k$
- 5:    $x_{k+1} \leftarrow x_k + \alpha_k r_k$
- 6:    $r_{k+1} \leftarrow r_k - \alpha_k A r_k$
- 7:   **se**  $r_{k+1} = o$  **allora esci**,  $x_{k+1}$  è la soluzione cercata
- 8: **fine ciclo 1**
- 9: **esci**

La convergenza del metodo è assicurata, ma la sua velocità diminuisce al crescere del numero di condizionamento spettrale  $K_2(A) = \lambda_1(A)/\lambda_n(A)$ , dove con  $\lambda_1(A) \geq \lambda_2(A) \geq \dots \geq \lambda_n(A)$  denotiamo gli autovalori (reali e positivi) della matrice  $A$ . Infatti, introdotta la norma

$$\|x\|_A = \sqrt{x^T A x}$$

vale la seguente stima:

$$\|x - x_{k+1}\|_A \leq \left( \frac{K_2(A) - 1}{K_2(A) + 1} \right)^{k+1} \|x - x_0\|_A$$

Conviene quindi procedere alla minimizzazione di  $\Phi$  scegliendo delle direzioni migliori  $\{p_1, p_2, \dots\}$ , in generale non coincidenti con quelle di massima pendenza locale  $\{r_0, r_1, \dots\}$ , e definendo

$$x_{k+1} = x_k + \alpha p_{k+1}, \quad k = 0, 1, \dots$$

Anche in questo caso sceglieremo il parametro  $\alpha$  in modo che  $\Phi(x_k + \alpha p_{k+1})$  sia minimo. Il valore ottimo è dato da

$$\alpha = \alpha_k = \frac{p_{k+1}^T r_k}{p_{k+1}^T A p_{k+1}}$$

Affinché risulti  $\alpha \neq 0$ , e si abbia quindi una effettiva diminuzione del valore di  $\Phi$ , è necessario che  $p_{k+1}^T r_k \neq 0$ , cioè che la direzione  $p_{k+1}$  non sia ortogonale al residuo  $r_k$ . Ma come scegliamo le direzioni  $\{p_k\}$  in modo che tale condizione sia soddisfatta e al tempo stesso la velocità di convergenza risulti sufficientemente elevata? Un esame del problema mostra (vedi [3.19, pag. 519]) come convenga scegliere, quando ovviamente  $r_k \neq o$ , la direzione  $p_{k+1} \neq o$  (la cui esistenza è certamente assicurata) con le seguenti proprietà:

$$(3.26) \quad \begin{aligned} p_j^T A p_{k+1} &= 0, \quad j = 1, 2, \dots, k \\ p_{k+1}^T r_k &\neq 0 \end{aligned}$$

Ricordiamo che due vettori non nulli  $p_i$  e  $p_j$  che soddisfano la relazione

$$p_j^T A p_i = 0, \quad i \neq j$$

sono definiti *A-coniugati*. Pertanto la prima condizione in (3.26) implica che la direzione  $p_{k+1}$  sia A-coniugata alle precedenti  $p_j$ ,  $j = 1, \dots, k$ . Conseguenza immediata di tale proprietà è la indipendenza lineare dei vettori  $\{p_1, \dots, p_k\}$ , per  $k \leq n$ .

La scelta ritenuta ottimale dei vettori  $\{p_k\}$  è la seguente:

$$\begin{aligned} p_1 &= r_0 \\ p_{k+1} &= r_k + \beta_k p_k, \quad k = 1, 2, \dots \\ \beta_k &= \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}} > 0 \end{aligned}$$

Come in precedenza, per calcolare il residuo  $r_{k+1}$  conviene (dal punto di vista della propagazione degli errori) utilizzare la rappresentazione

$$r_{k+1} = b - Ax_{k+1} = b - A(x_k + \alpha_k p_{k+1}) = r_k - \alpha_k Ap_{k+1}$$

Anche per questi residui vale la proprietà:  $r_{k+1}^T r_k = 0$ .

Poiché da quest'ultima relazione segue che  $p_k^T r_k = 0$ , al coefficiente  $\alpha_k$  possiamo dare la nuova espressione:

$$\alpha_k = \frac{r_k^T r_k}{p_{k+1}^T A p_{k+1}} > 0$$

Tutto ciò ci consente di dare una prima formulazione del ben noto *metodo del gradiente coniugato*:

- 1:  $x_0 \leftarrow o$
- 2:  $r_0 \leftarrow b$
- 3:  $p_1 \leftarrow r_0$
- 4:  $\alpha_0 \leftarrow r_0^T r_0 / p_1^T A p_1$
- 5:  $x_1 \leftarrow x_0 + \alpha_0 p_1$
- 6:  $r_1 \leftarrow r_0 - \alpha_0 A p_1$
- 7: **ciclo 1:**  $k = 1, \dots, k_{\max}$
- 8:   **se**  $r_k = o$  **allora esci**,  $x = x_k$
- 9:    $\beta_k \leftarrow r_k^T r_k / r_{k-1}^T r_{k-1}$
- 10:    $p_{k+1} \leftarrow r_k + \beta_k p_k$
- 11:    $\alpha_k \leftarrow r_k^T r_k / p_{k+1}^T A p_{k+1}$
- 12:    $x_{k+1} \leftarrow x_k + \alpha_k p_{k+1}$
- 13:    $r_{k+1} \leftarrow r_k - \alpha_k A p_{k+1}$
- 14: **fine ciclo 1**
- 15: **esci**

Per quanto riguarda la convergenza del metodo è possibile dimostrare che, in aritmetica con precisione infinita, certamente risulta  $Ax_n = b$ ; ossia, dopo  $n$  iterazioni troviamo la soluzione del sistema. Il metodo del gradiente coniugato pur essendo un metodo iterativo, che quindi utilizza sempre e solo i dati iniziali  $A$  e  $b$  senza mai modificarli, termina esattamente dopo al più  $n$  iterazioni.

Occorre tuttavia rilevare che questo risultato non è del tutto soddisfacente per i seguenti motivi. Innanzi tutto a causa della precisione finita di calcolo, i vettori  $\{p_k\}$  non godranno più delle proprietà matematiche (3.26) che li avevano caratterizzati e il metodo dopo  $n$  passi fornirà quindi solo un'approssimazione della soluzione, non necessariamente con la precisione di macchina. Inoltre, anche se la precisione richiesta venisse raggiunta con  $n$  iterazioni, quando la dimensione di  $A$  è molto grande tale numero risulterebbe ancora eccessivo. Per questi motivi il metodo deve essere considerato a tutti gli effetti un metodo iterativo, a cui dobbiamo pertanto imporre un criterio di arresto.

Per quanto riguarda poi la sua velocità di convergenza è possibile ottenere la seguente stima:

$$\|x - x_{k+1}\|_A \leq 2 \left( \frac{\sqrt{K_2(A)} - 1}{\sqrt{K_2(A)} + 1} \right)^{k+1} \|x - x_0\|_A$$

Esso converge molto rapidamente quando la matrice  $A$  o è di forma  $I + B$  con rango  $(B) = r \ll n$ <sup>(†)</sup>, oppure ha un numero di condizionamento  $K_2(A) \approx 1$ . Quando non ci troviamo in queste condizioni esistono delle tecniche, dette di *precondizionamento*, che ci consentono di trasformare un sistema simmetrico definito positivo  $Ax = b$  in un altro  $\bar{A}\bar{x} = \bar{b}$  con le stesse caratteristiche e in più con  $K_2(\bar{A}) \approx 1$ , dalla cui soluzione  $\bar{x}$  è possibile dedurre  $x$ .

L'idea generale per ottenere il nuovo sistema è la seguente. Data una matrice simmetrica non singolare  $C$ , da  $Ax = b$  deduciamo

$$C^{-1}AC^{-1}Cx = C^{-1}b$$

Posto quindi

$$\bar{A} = C^{-1}AC^{-1}, \quad \bar{x} = Cx \quad \text{e} \quad \bar{b} = C^{-1}b$$

otteniamo

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

È facile osservare che la nuova matrice  $\bar{A}$  è ancora simmetrica definita positiva. Dobbiamo pertanto scegliere  $C$  in modo che  $K_2(\bar{A}) \approx 1$  e quindi applicare il metodo del gradiente coniugato al nuovo sistema  $\bar{A}\bar{x} = \bar{b}$ . Dalla sua soluzione  $\bar{x}$  dedurremo poi la soluzione originaria  $x$  risolvendo il sistema  $Cx = \bar{x}$ .

In pratica, applicando l'algoritmo di pagina 79 al nuovo sistema e rielaborando oculatamente le espressioni coinvolte, è possibile scrivere l'algoritmo in modo da ottenere direttamente la soluzione  $x$  senza dover mai calcolare o utilizzare  $C^{-1}$ . Si osserva inoltre che l'algoritmo dipende dalla sola matrice  $M = C^2$ , simmetrica definita positiva, che viene chiamata *precondizionatore*. La matrice  $C$  deve essere tale da rendere il sistema  $Mz = r$  di facile soluzione.

I precondizionatori proposti in letteratura sono diversi. Uno dei più noti e importanti è quello che scaturisce dalla cosiddetta *fattorizzazione incompleta di Choleski* della matrice  $A$ . Esso assume la forma  $M = \bar{L}_1 \bar{L}_1^T$ , dove, denotando con  $L_1$  la matrice triangolare

(†) Ossia  $A$  ha pochi autovalori distinti.

(inferiore) della classica decomposizione di Choleski (vedi pagina 60), definiamo  $(\bar{L}_1)_{ij} = (L_1)_{ij}$  quando  $(A)_{ij} \neq 0$  e  $(\bar{L}_1)_{ij} = 0$  se  $(A)_{ij} = 0$ .

```

1:  $r_0 \leftarrow b$ 
2:  $Mz_0 = r_0 \Rightarrow z_0$ 
3:  $p_1 \leftarrow z_0$ 
4:  $\alpha_0 \leftarrow r_0^T z_0 / p_1^T A p_1$ 
5:  $x_1 \leftarrow \alpha_0 p_1$ 
6:  $r_1 \leftarrow r_0 - \alpha_0 A p_1$ 
7: ciclo 1:  $k = 1, \dots, k_{\max}$ 
8:   se  $r_k = o$  allora esci,  $x = x_k$ 
9:    $Mz_k = r_k \Rightarrow z_k$ 
10:   $\beta_k \leftarrow r_k^T z_k / r_{k-1}^T z_{k-1}$ 
11:   $p_{k+1} \leftarrow z_k + \beta_k p_k$ 
12:   $\alpha_k \leftarrow r_k^T z_k / p_{k+1}^T A p_{k+1}$ 
13:   $x_{k+1} \leftarrow x_k + \alpha_k p_{k+1}$ 
14:   $r_{k+1} \leftarrow r_k - \alpha_k A p_{k+1}$ 
15: fine ciclo 1
16: esci
```

## Bibliografia

- [3.1] R.S. Varga, *Matrix iterative analysis*, Prentice-Hall, Englewood Cliffs, N.J., 1962.
- [3.2] D. K Faddeev, V. N. Faddeeva, *Computational methods of linear algebra*, Freeman, San Francisco, 1963.
- [3.3] L. Fox, *Introduction to numerical linear algebra*, Oxford University Press, New York, 1965.
- [3.4] G.E. Forsythe, C. Moler, *Computer solution of linear algebraic systems*, Prentice-Hall, Englewood Cliffs, N.J., 1967.
- [3.5] B. Noble, *Applied linear algebra*, Prentice-Hall, Englewood Cliffs, N.J., 1969.
- [3.6] N. Gastinel, *Linear numerical analysis*, Prentice-Hall, Hermann, Paris, 1970.
- [3.7] J. H. Wilkinson, C. Reinsch, *Handbook for automatic computation, Vol. II: Linear algebra*, Springer Verlag, New York, 1971.
- [3.8] D.M. Young, *Iterative solution of large linear systems*, Academic Press, New York, 1971.
- [3.9] G.W. Stewart, *Introduction to matrix computation*, Academic Press, New York, 1973.
- [3.10] H. R. Schwarz, H. Rutishauser, E. Stiefel, *Numerical analysis of simmetric matrices*, Prentice-Hall, Englewood Cliffs, N.J., 1973.
- [3.11] C. G. Barozzi, *Introduzione agli algoritmi dell'algebra lineare*, Zanichelli, Bologna, 1976.

- [3.12] A. Jennings, *Matrix computation for engineers and scientists*, John Wiley & Sons, New York, 1977.
- [3.13] J.C Nash, *Compact numerical methods for computers: linear algebra and function minimization*, Adam Hilger Ltd., Bristol, 1979.
- [3.14] J.J. Dongarra, et. al., *LINPACK User's guide*, Society for Industrial and Applied Mathematics, Philadelphia, 1979.
- [3.15] I.S. Duff, G.W. Stewart (Eds.), *Sparse matrix proceedings 1978*, SIAM, Philadelphia, 1979.
- [3.16] L.A. Hageman, D. M Young, *Applied iterative methods*, Academic Press, New York, 1981.
- [3.17] A. George, J. W. Liu, *Computer solution of large sparse positive definite systems*, Prentice-Hall, Englewood Cliff, N.J., 1981.
- [3.18] I. S. Duff, *Sparse matrices and their uses*, Academic Press, London, 1982.
- [3.19] G. H. Golub, C. Van Loan, *Matrix computations*, Hohn Hopkins University Press, Baltimore, 1983.
- [3.20] D. Bini, M. Capovani, O. Menchi, *Metodi numerici per l'algebra lineare*, Zanichelli, 1988.
- [3.21] P.E. Gill, W. Murray, M. H. Wright, *Numerical linear algebra and optimization*, Addison-Wesley, Redwood City, 1991.
- [3.22] W. Hackbusch, *Iterative solution of large sparse systems of equations*, Springer-Verlag, New York, 1994.
- [3.23] J.W. Demmel, *Applied numerical linear algebra*, SIAM Publications, Philadelphia, 1997.
- [3.24] Y. Saad, *Iterative methods for sparse linear systems*, SIAM Publications, Philadelphia, 2003.

## Esercizi proposti

**3.1.** Risolvere il seguente sistema lineare con il metodo di Gauss (e strategia di pivoting parziale)

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

**3.2.** Consideriamo il sistema

$$\begin{cases} x_1 + \frac{2}{3}x_2 + \frac{4}{3}x_3 = 3 \\ 3x_1 + 2x_2 + x_3 = 6 \\ x_1 + 2x_2 + x_3 = 4 \end{cases}$$

la cui soluzione è  $x_1 = x_2 = x_3 = 1$ .

Supponendo di operare nel sistema di numerazione decimale, con aritmetica floating-point con  $t = 4$  cifre per la mantissa, e tecnica di arrotondamento (i) di pagina 7, risolvere il sistema con il metodo di Gauss (a) senza pivoting, (b) con pivoting parziale. Commentare i risultati.

**3.3.** Se un sistema lineare ha un numero di condizionamento  $K(A) = 10^6$  e desideriamo determinare, con un metodo numerico stabile, la sua soluzione con almeno tre cifre significative, come occorre procedere?

**3.4.** Determinare la fattorizzazione  $LU$  di una matrice  $A$  simmetrica tridiagonale a diagonale dominante. Quali sono le strutture di  $L$  ed  $U$ ? E se  $A$  non è a diagonale dominante?

**3.5.** Scrivere l'algoritmo che risolve un sistema triangolare inferiore.

**3.6.** Sia data una matrice simmetrica tridiagonale a diagonale dominante (oppure definita positiva). Supponiamo di memorizzare la diagonale nel vettore  $d$  e la codiagonale nel vettore  $c$ . Costruire l'algoritmo di Gauss utilizzando solo vettori.

**3.7.** Sia  $A$  una matrice simmetrica definita positiva. Ad  $A$  applichiamo il metodo di Gauss (senza pivoting). Dopo la prima eliminazione  $A$  è ridotta alla forma

$$A^{(1)} = \begin{pmatrix} \alpha & a^T \\ o & A_{22}^{(1)} \end{pmatrix}$$

dove  $\alpha \in \mathbb{R}$  e  $A_{22}^{(1)} \in \mathbb{R}^{(n-1) \times (n-1)}$ . Dimostrare che  $A_{22}^{(1)}$  è ancora simmetrica definita positiva. Più in generale, dopo  $k$  eliminazioni abbiamo

$$A^{(k)} = \begin{pmatrix} A_{11}^{(k)} & A_{12}^{(k)} \\ O & A_{22}^{(k)} \end{pmatrix}$$

dove  $A_{11}^{(k)} \in \mathbb{R}^{k \times k}$  è triangolare superiore e  $A_{22}^{(k)} \in \mathbb{R}^{(n-k) \times (n-k)}$ . Dimostrare che per ogni  $k = 1, 2, \dots, n-1$   $A_{22}^{(k)}$  è simmetrica definita positiva.

**3.8.** Semplificare il processo di eliminazioni di Gauss adattandolo al caso in cui la matrice  $A$  è simmetrica e definita positiva (ricordare il risultato dell'esercizio 3.7).

**3.9.** Dimostrare che quando la matrice dei coefficienti di un sistema lineare è simmetrica e a diagonale dominante la strategia di pivoting parziale non altera l'ordine iniziale delle equazioni.

**3.10.** Sia  $H_{10}$  la matrice di Hilbert di ordine 10. Risolvere il sistema

$$H_{10}x = b$$

con  $b$  scelto in modo che risulti  $x = (1, 1, \dots, 1)^T$ , e stampare il residuo  $r = b - H_{10}x$ . Commentare i risultati ottenuti.

**3.11.** “Azzerare” gli elementi non diagonale della prima colonna della matrice

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 & -1 \end{pmatrix}$$

e osservare la propagazione degli elementi non nulli.

**3.12.** Sia data la matrice

$$A = \begin{pmatrix} U_{11} & U_{12} \\ v^T & w^T \end{pmatrix} \in \mathbb{R}^{(k+1) \times n}$$

dove  $U_{11} \in \mathbb{R}^{k \times k}$  è triangolare superiore non singolare e  $v^T \in \mathbb{R}^{1 \times k}$ . Verificare dapprima che esiste un vettore  $y \in \mathbb{R}^k$  tale che la matrice prodotto

$$\begin{pmatrix} I & o \\ y^T & 1 \end{pmatrix} \begin{pmatrix} U_{11} & U_{12} \\ v^T & w^T \end{pmatrix}$$

risulta triangolare superiore. Utilizzare poi questo risultato per risolvere il sistema  $Ax = b$ ,  $A \in \mathbb{R}^{n \times n}$ , leggendo una riga di  $A$  per volta (e il corrispondente elemento di  $b$ ) e facendo intervenire  $n^2/2$  locazioni di memoria anziché  $n^2$  (come richiederebbe la memorizzazione di tutta la matrice  $A$ ).

**3.13.** Risolvere il sistema lineare  $(A^T A)x = b$ ,  $A \in \mathbb{R}^{n \times n}$  non singolare, senza costruire esplicitamente la matrice  $B = A^T A$ .

**3.14.** Siano dati  $a, b \in \mathbb{R}^n$ ,  $A \in \mathbb{R}^{n \times n}$  non singolare, e  $h \in \mathbb{R}$ . Come determinereste il vettore incognito

$$x = a + hA^{-1}b?$$

**3.15.** Calcolare l'inversa della matrice di Hilbert di ordine 5 ( $H_5$ ) e stampare il residuo  $R = I - H_5 H_5^{-1}$ . Commentare il risultato ottenuto.

**3.16.** Sia data una matrice  $A$  simmetrica definita positiva. Come operereste per calcolare  $A^{-1}$ ?

**3.17.** Dimostrare le condizioni (3.16) e (3.19).

**3.18.** Costruire un algoritmo che implementi il metodo di Jacobi nel caso di un sistema lineare non necessariamente sparso. Imporre un numero massimo di iterazioni ed un test sull'errore relativo.

**3.19.** Il sistema  $Ax = b$ , con

$$A = \begin{pmatrix} 4 & -1 & 0 & -1 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & -1 & 4 & 0 & 0 & -1 \\ -1 & 0 & 0 & 4 & -1 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & -1 & 0 & -1 & 4 \end{pmatrix} \quad \text{e} \quad b = \begin{pmatrix} 2 \\ 1 \\ 2 \\ 2 \\ 1 \\ 2 \end{pmatrix}$$

ha soluzione  $x = (1, 1, 1, 1, 1, 1)^T$ . Risolvere il sistema usando dapprima il metodo di Jacobi e successivamente il metodo di Gauss-Siedel. Prendere come approssimazione iniziale  $x^{(0)} = o$  e valutare la soluzione con precisione relativa  $\text{tol} = 0.0001$ . Osservare la velocità di convergenza dei due metodi.

**3.20.** Nel problema precedente usare il metodo SOR con  $\omega = 1 + 0.2k$ ,  $k = 0, 1, 2, 3, 4, 5$ , e osservare la variazione del numero di iterazioni richieste (nei casi  $k < 5$ ) per raggiungere la precisione desiderata.

**3.21.** Sia dato un sistema lineare sparso di ordine  $N$ . Memorizzare solamente gli elementi non nulli e costruire l'algoritmo di Gauss-Siedel utilizzando un solo vettore per memorizzare le approssimazioni della soluzione.

**3.22.** Sia  $A \in \mathbb{R}^{n \times n}$  una matrice simmetrica definita positiva. Dimostrare che i vettori A-coniugati  $\{p_1, p_2, \dots, p_k\}$ , con  $k \leq n$ , del metodo del gradiente coniugato sono necessariamente linearmente indipendenti.



# Capitolo 4

## Autovalori di matrici

### 4.1 Preliminari

Alcuni problemi di ingegneria e di fisica vengono descritti con modelli matematici del tipo

$$(4.1) \quad \begin{cases} f_1(\xi_1, \dots, \xi_n; \lambda) = 0 \\ f_2(\xi_1, \dots, \xi_n; \lambda) = 0 \\ \dots \dots \dots \\ f_n(\xi_1, \dots, \xi_n; \lambda) = 0 \end{cases}$$

cioè da un sistema di  $n$  equazioni nelle  $n$  incognite  $\xi_1, \dots, \xi_n$ , dove le funzioni  $f_i$  dipendono anche da un parametro  $\lambda$ . Normalmente il sistema ammette soluzioni non nulle solamente in corrispondenza di determinati valori  $\{\lambda_i\}$  del parametro  $\lambda$ . Questi valori speciali  $\{\lambda_i\}$  vengono chiamati *autovalori* del sistema, mentre le corrispondenti soluzioni non nulle sono chiamate *autosoluzioni*.

In questo capitolo esaminiamo il caso in cui le funzioni  $f_i$  in (4.1) sono lineari e il sistema (4.1) assume la forma

$$(4.2) \quad (\lambda I - A)x = o, \quad \text{ossia} \quad Ax = \lambda x$$

Il problema che affrontiamo è pertanto il seguente: *data una matrice  $A$  di ordine  $n$ , trovare dei numeri  $\lambda_i$  (reali o complessi) in corrispondenza dei quali il sistema (4.2) ammetta soluzioni  $x_i$  non nulle.* Tali soluzioni  $x_i = (\xi_1^{(i)}, \dots, \xi_n^{(i)})^T$  vengono chiamate *autovettori*.

Le matrici  $A$  in generale possono essere reali o complesse, sparse o dense, e spesso sono simmetriche. Noi circoscriveremo la nostra esposizione alle matrici reali; la descrizione dei metodi che presenteremo può essere facilmente generalizzata per le matrici a elementi complessi.

Dalla prima equazione in (4.2) segue che gli autovalori  $\lambda_i$  coincidono con le  $n$  radici dell'equazione caratteristica

$$(4.3) \quad \det(\lambda I - A) = 0$$

dove  $\det(\lambda I - A) = \lambda^n + \alpha_1 \lambda^{n-1} + \cdots + \alpha_{n-1} \lambda + \alpha_n$  è un polinomio (denominato *polinomio caratteristico*) di grado  $n$  nella variabile  $\lambda$ . Gli autovalori sono esattamente  $n$ , non necessariamente tutti distinti, e possono venire calcolati indipendentemente dagli autovettori. Una volta che un autovalore  $\lambda_i$  è noto, un corrispondente autovettore  $x_i$  in *teoria*(<sup>†</sup>) potrebbe venire determinato risolvendo il sistema lineare omogeneo (4.2). Se invece è l'autovettore  $x_i$  ad essere noto, allora l'autovalore  $\lambda_i$  ad esso associato è fornito dall'espressione

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

denominata *quoziente di Rayleigh*.

La relazione (4.3) sembra suggerire la determinazione degli autovalori quali radici della stessa equazione (4.3) utilizzando i metodi del capitolo 6(<sup>††</sup>). Tuttavia questo approccio si rivela generalmente poco efficiente rispetto ai metodi che presenteremo nei paragrafi che seguono. Per una visione completa dei metodi numerici e del software attualmente utilizzati consigliamo le letture [4.5] e [4.6].

Per scegliere un metodo numerico efficiente è opportuno rispondere preliminarmente alle seguenti domande:

- (i) è solamente richiesto l'autovalore più grande in modulo (oppure il più piccolo, o comunque uno ben determinato) e il corrispondente autovettore?
- (ii) sono richiesti tutti gli autovalori e gli autovettori corrispondenti, oppure solo gli autovalori?
- (iii) la matrice ha proprietà speciali (simmetrica, tridiagonale, sparsa)?

Enunciamo dapprima un criterio semplice per la localizzazione degli autovalori di una matrice  $A$ .

**Teorema 4.1.** (Gershgorin) [4.10, pag. 76]. *Definiamo le somme*

$$r_i = \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|, \quad i = 1, 2, \dots, n$$

$$c_j = \sum_{\substack{i=1 \\ i \neq j}}^n |a_{ij}|, \quad j = 1, 2, \dots, n$$

*Le affermazioni seguenti risultano vere:*

---

(<sup>†</sup>) Ma generalmente non in pratica.

(<sup>††</sup>) Senza costruire però esplicitamente il polinomio caratteristico, altrimenti potremmo ritrovarci ad affrontare un problema malcondizionato.

(i) ogni autovalore di  $A$  appartiene all'insieme

$$R = \bigcup_{i=1}^n R_i, \quad R_i = \{z : |z - a_{ii}| \leq r_i\}$$

(ii) ogni autovalore di  $A$  appartiene all'insieme

$$C = \bigcup_{j=1}^n C_j, \quad C_j = \{z : |z - a_{jj}| \leq c_j\}$$

e quindi anche all'insieme intersezione  $R \cap C$ ;

(iii) ogni componente di  $R$  o di  $C$ , cioè ogni unione connessa massimale di cerchi  $R_i$  o  $C_j$ , contiene tanti autovalori di  $A$  quanti sono i cerchi della componente, tenendo conto della molteplicità di ogni autovalore e di ogni cerchio.

► **Esempio.** Applichiamo il teorema alla matrice

$$A = \begin{pmatrix} 4 & -1 & 1 & 0 & 0 \\ 1 & 3 & -1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 1 & 8 \end{pmatrix}$$

i cui autovalori sono  $\lambda_1 = 5 + \sqrt{10}$ ,  $\lambda_2 = \lambda_3 = 3$ ,  $\lambda_4 = 2$ ,  $\lambda_5 = 5 - \sqrt{10}$ . Tutti gli autovalori appartengono all'insieme  $R$  unione dei cerchi seguenti:

$$\begin{aligned} R_1 &= \{z : |z - 4| \leq 2\} \\ R_2 &= \{z : |z - 3| \leq 2\} \\ R_3 &= \{z : |z - 1| \leq 1\} \\ R_4 &= \{z : |z - 2| \leq 1\} \\ R_5 &= \{z : |z - 8| \leq 1\} \end{aligned}$$

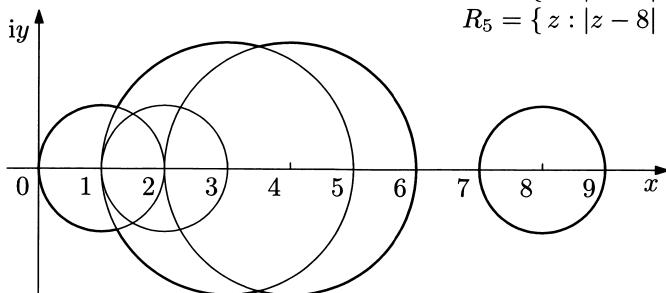
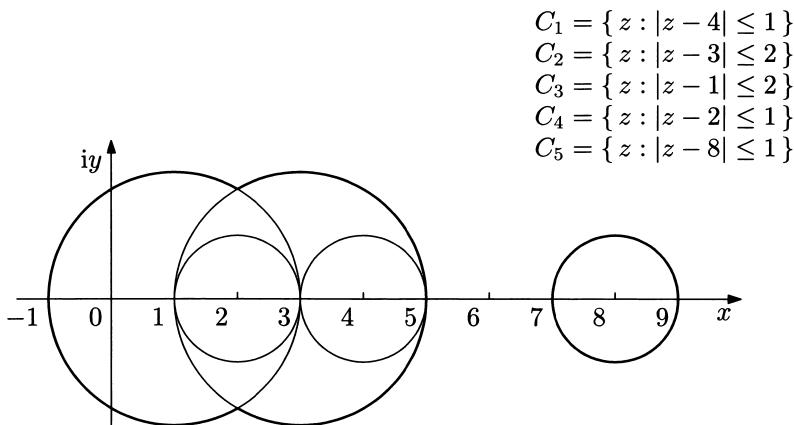


Figura 4.1

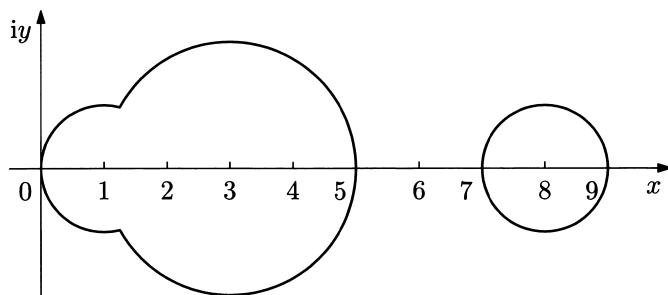
La regione  $R$  è formata dalle due componenti segnate in neretto. In una vi sono 4 cerchi e quindi 4 autovalori; l'altra, essendo formata da 1 solo cerchio, contiene 1 autovalore. Tutti gli autovalori appartengono però anche alla regione  $C$  unione dei cerchi  $C_j$ :



**Figura 4.2**

La regione  $C$  (segnata in neretto) ha due componenti: la prima contiene 4 autovalori, la seconda 1.

Possiamo pertanto concludere che tutti gli autovalori devono giacere nella regione intersezione  $R \cap C$ .



**Figura 4.3**

Ovviamente, quando la matrice  $A$  è simmetrica è sufficiente considerare l'intersezione delle regioni suddette con l'asse reale; in questo caso individueremo segmenti contenenti

gli autovalori.

Teoremi di questo tipo possono essere utili, per esempio, per ottenere delle limitazioni sull'autovalore più grande (in modulo). Un altro risultato che ci permette di conseguire delle maggiorazioni del raggio spettrale della matrice  $A$  è il seguente:

$$\rho(A) \leq \|A\|$$

dove  $\|\cdot\|$  denota una qualsiasi norma di matrice compatibile con una norma di vettore.

## 4.2 Metodo delle potenze

Denotiamo con  $\lambda_1, \lambda_2, \dots, \lambda_n$  gli  $n$  autovalori della matrice  $A$ , e supponiamo che tra essi ve ne sia uno solo di modulo massimo, ossia  $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n|$ . Indichiamo con  $x_1, x_2, \dots, x_n$   $n$  autovettori associati ai suddetti autovalori, e supponiamo che i primi siano linearmente indipendenti. Un generico vettore  $v_0$  può essere allora espresso nella forma

$$v_0 = \alpha_1 x_1 + \alpha_2 x_2 + \dots + \alpha_n x_n$$

Scegliamo il vettore iniziale  $v_0$  in modo che  $\alpha_1 \neq 0$ .

Osserviamo subito che poiché  $A$  è reale, il polinomio caratteristico è a coefficienti reali e quindi le sue eventuali radici (autovalori) complesse compaiono a coppie coniugate. Ne segue che  $\lambda_1$  deve necessariamente essere reale e le componenti di  $x_1$  possono essere scelte reali.

Successivamente costruiamo il nuovo vettore

$$\begin{aligned} v_1 &= Av_0 = \alpha_1 Ax_1 + \alpha_2 Ax_2 + \dots + \alpha_n Ax_n = \alpha_1 \lambda_1 x_1 + \alpha_2 \lambda_2 x_2 + \dots + \alpha_n \lambda_n x_n = \\ &= \lambda_1 \left[ \alpha_1 x_1 + \left( \frac{\lambda_2}{\lambda_1} \right) \alpha_2 x_2 + \dots + \left( \frac{\lambda_n}{\lambda_1} \right) \alpha_n x_n \right] \end{aligned}$$

e in generale

$$\begin{aligned} v_m &= Av_{m-1} = A^m v_0 = \alpha_1 A^m x_1 + \alpha_2 A^m x_2 + \dots + \alpha_n A^m x_n \\ (4.4) \quad &= \alpha_1 \lambda_1^m x_1 + \alpha_2 \lambda_2^m x_2 + \dots + \alpha_n \lambda_n^m x_n \\ &= \lambda_1^m \left[ \alpha_1 x_1 + \alpha_2 \left( \frac{\lambda_2}{\lambda_1} \right)^m x_2 + \dots + \alpha_n \left( \frac{\lambda_n}{\lambda_1} \right)^m x_n \right] \end{aligned}$$

Poiché per ipotesi  $|\lambda_i/\lambda_1| < 1$ ,  $i = 2, 3, \dots, n$ , avremo

$$(4.5) \quad \lim_{m \rightarrow \infty} \frac{1}{\lambda_1^m} v_m = \alpha_1 x_1$$

Indicando con  $(v_m)_k$  la  $k$ -esima componente del vettore  $v_m$ , e supponendo  $(x_1)_k \neq 0$ , dalla (4.4) segue che

$$(4.6) \quad \lambda_1 = \lim_{m \rightarrow \infty} \frac{(v_{m+1})_k}{(v_m)_k} = \lim_{m \rightarrow \infty} \frac{e_k^T A v_m}{e_k^T v_m}, \quad k = 1, 2, \dots, n$$

dove  $(e_k)_i = \delta_{ki}$ . La convergenza della successione  $\{(v_{m+1})_k / (v_m)_k\}$  al limite  $\lambda_1$  dipende dalla potenza  $|\lambda_2/\lambda_1|^m$ : essa è tanto più rapida quanto più piccolo è il rapporto  $|\lambda_2/\lambda_1|$ .

Osserviamo che per  $m \rightarrow \infty$  la potenza  $|\lambda_1|^m$  in (4.4) tende a zero se  $|\lambda_1| < 1$  e a infinito se  $|\lambda_1| > 1$ . Pertanto, poiché siamo interessati al rapporto delle  $k$ -esime componenti di due vettori successivi  $v_m$  e  $v_{m+1}$ , e gli autovettori sono definiti a meno di una costante moltiplicativa, possiamo “normalizzare” la successione in modo da lasciare inalterato il predetto rapporto. Costruiamo allora la nuova successione  $\{y_m\}$  definita dall'algoritmo:

$$(4.7) \quad \begin{aligned} 1: \quad & y_0 \leftarrow (1, 1, \dots, 1)^T \\ 2: \quad & \textbf{ciclo 1: } m = 0, \dots, m_{\max} \\ 3: \quad & w_{m+1} \leftarrow Ay_m \\ 4: \quad & \lambda_1^{(m+1)} \leftarrow (w_{m+1})_{k_0} / (y_m)_{k_0} \\ 5: \quad & y_{m+1} \leftarrow w_{m+1} / \|w_{m+1}\|_\infty \\ 6: \quad & \textbf{fine ciclo 1} \end{aligned}$$

Le quantità  $\lambda_1^{(m+1)}$  e  $y_{m+1}$  rappresentano le approssimazioni di  $\lambda_1$  e di un corrispondente autovettore di norma (infinito) unitaria. Anzi,  $\lambda_1^{(m+1)}$  coincide ancora con il rapporto  $(v_{m+1})_k / (v_m)_k$  presente in (4.6); infatti abbiamo:

$$\begin{aligned} v_1 &= Av_0 = Ay_0 = w_1, \quad \text{con } y_0 = v_0 \text{ e } w_1 = Ay_0 \\ v_2 &= Av_1 = Aw_1 = \|w_1\|Ay_1 = \|w_1\|w_2, \quad \text{con } y_1 = w_1 / \|w_1\| \text{ e } w_2 = Ay_1 \\ v_3 &= Av_2 = \|w_1\|Aw_2 = \|w_1\| \|w_2\|Ay_2 = \|w_1\| \|w_2\|w_3, \\ &\quad \text{con } y_2 = w_2 / \|w_2\| \text{ e } w_3 = Ay_2 \\ &\vdots \\ v_{m+1} &= Av_m = \|w_1\| \|w_2\| \dots \|w_m\|Ay_m = \|w_1\| \|w_2\| \dots \|w_m\|w_{m+1}, \\ &\quad \text{con } y_m = w_m / \|w_m\| \text{ e } w_{m+1} = Ay_m \end{aligned}$$

da cui segue

$$\frac{(v_{m+1})_k}{(v_m)_k} = \frac{\|w_1\| \|w_2\| \dots \|w_{m-1}\| \|w_m\| (w_{m+1})_k}{\|w_1\| \|w_2\| \dots \|w_{m-1}\| (w_m)_k} = \frac{(w_{m+1})_k}{\left(\frac{w_m}{\|w_m\|}\right)_k} = \frac{(w_{m+1})_k}{(y_m)_k}$$

In (4.7), per garantire che  $(y_m)_{k_0} \neq 0$  è sufficiente scegliere l'indice  $k_0$  in modo che  $|(w_m)_{k_0}| = \|w_m\|_\infty$ , ovvero  $|(y_m)_{k_0}| = 1$ . Poiché in questo caso  $k_0$  potrebbe non rimanere costante per tutte le iterazioni, sarebbe improprio applicare la relazione (4.6). È tuttavia possibile dimostrare che la convergenza della successione  $(w_{m+1})_{k_0} / (y_m)_{k_0}$  a  $\lambda_1$  è garantita ed è analoga alla (4.6).

▷ **Osservazioni.** 1. Se  $\alpha_1 = 0$  e  $|\lambda_2| > |\lambda_3|$  il processo iterativo definito dalla (4.6) dovrebbe, teoricamente, convergere a  $\lambda_2$ . In pratica però, causa gli errori di arrotondamento, abbiamo “sempre”  $\alpha_1 \neq 0$ .

2. Quando  $|\lambda_2| \cong |\lambda_1|$  la convergenza della successione  $\{\lambda_1^{(m+1)}\}$  a  $\lambda_1$  può risultare eccessivamente lenta. In questa situazione il metodo viene utilizzato per ottenere solo una stima iniziale, da migliorare successivamente con un metodo più veloce, quale, ad esempio, il *metodo delle potenze inverse* che tra poco presenteremo.
3. Se  $\lambda_1$  è reale ed ha molteplicità  $k$  (e  $|\lambda_1| > |\lambda_{k+1}| \geq \dots \geq |\lambda_n|$ ) il metodo converge ancora a  $\lambda_1$  e ad un suo autovettore (anche se in questo caso a  $\lambda_1$  possono corrispondere più autovettori linearmente indipendenti, sino ad un massimo di  $k$ ). La convergenza risulta tuttavia generalmente lenta.
4. Ovviamente, il metodo, così come è stato formulato, non può convergere ad autovettori ed autovalori complessi. Inoltre, neppure la successione  $|(w_{m+1})_{k_0}| / |(y_m)_{k_0}|$  può convergere a  $|\lambda_1| = |\lambda_2|$ , essendo  $\lambda_1$  e  $\lambda_2$  complessi coniugati. Anche quando  $\lambda_1$  e  $\lambda_2$ , pur essendo reali e distinti hanno lo stesso modulo, cioè  $\lambda_1 = -\lambda_2$ , il metodo generalmente non converge (vedere tuttavia l'esercizio 4.10).

□

► **Esempi.** L'algoritmo (4.7) applicato alla matrice

$$(4.8) \quad \begin{pmatrix} -7 & -9 & 9 \\ 11 & 13 & -9 \\ -16 & -16 & 20 \end{pmatrix}$$

i cui autovalori sono  $\lambda_1 = 20$ ,  $\lambda_2 = 4$  e  $\lambda_3 = 2$ , con  $\lambda_2/\lambda_1 = 0.2$ , produce i risultati riportati in tabella 4.1.

$m$	$\lambda_1^{(m)}$
1	-7.00000
2	15.06667
3	21.39130
4	20.26016
5	20.05136
6	20.01025
7	20.00205
8	20.00041
9	20.00009
10	20.00002
11	20.00000

**Tabella 4.1**

Quando invece applichiamo lo stesso algoritmo alla matrice

$$(4.9) \quad \begin{pmatrix} -4 & -5 & 4 \\ 14 & 15 & -5 \\ -1 & -1 & 11 \end{pmatrix}$$

che ha autovalori  $\lambda_1 = (21 + \sqrt{5})/2 = 11.61803398\dots$ ,  $\lambda_2 = (21 - \sqrt{5})/2$ ,  $\lambda_3 = 1$ , con rapporto  $\lambda_2/\lambda_1 \cong 0.81$ , otteniamo la successione riportata in tabella 4.2.

$m$	$\lambda_1^{(m)}$	$m$	$\lambda_1^{(m)}$	$m$	$\lambda_1^{(m)}$
1	-5.000000	15	11.169128	29	11.590217
2	12.799999	16	11.240958	30	11.595517
3	10.140628	17	11.303317	$\vdots$	$\vdots$
4	10.020030	18	11.356814	50	11.617717
5	10.098723	19	11.402236	$\vdots$	$\vdots$
6	10.204273	20	11.440474	60	11.617796
7	10.317976	21	11.472422	$\vdots$	$\vdots$
8	10.435890	22	11.498954	70	11.618030
9	10.555276	23	11.520877	71	11.618031
10	10.673409	24	11.538915	72	11.618032
11	10.787706	25	11.553703	73	11.618032
12	10.895905	26	11.565797	74	11.618032
13	10.996241	27	11.575660	75	11.618033
14	11.087523	28	11.583692	76	11.618033

Tabella 4.2

Quando la matrice  $A$  è simmetrica e  $\lambda_1$ , di molteplicità  $k \geq 1$ , è l'unico autovalore di modulo massimo, cioè  $-\lambda_1$  non è autovalore di  $A$ , conviene definire in (4.7)

$$y_{m+1} = \frac{w_{m+1}}{\|w_{m+1}\|_2}$$

e considerare come approssimazione  $\lambda_1^{(m+1)}$  il quoziente di Rayleigh

$$\frac{y_m^T A y_m}{y_m^T y_m} = y_m^T w_{m+1}$$

Infatti, quando  $A$  è simmetrica abbiamo  $n$  autovettori  $\{x_i\}$  ortonormali, ovvero possiamo scrivere

$$y_m = \frac{1}{k_m} (\alpha_1 \lambda_1^m x_1 + \alpha_2 \lambda_2^m x_2 + \cdots + \alpha_n \lambda_n^m x_n), \quad k_m = \|w_1\|_2 \|w_2\|_2 \dots \|w_m\|_2$$

con  $x_i^T x_j = \delta_{ij}$ ; inoltre

$$y_m^T w_{m+1} = \frac{y_m^T A y_m}{y_m^T y_m} = \frac{\sum_{i=1}^n \alpha_i^2 \lambda_i^{2m+1}}{\sum_{i=1}^n \alpha_i^2 \lambda_i^{2m}} = \lambda_1 \frac{\alpha_1^2 + \sum_{i=2}^n \alpha_i^2 \left(\frac{\lambda_i}{\lambda_1}\right)^{2m+1}}{\alpha_1^2 + \sum_{i=2}^n \alpha_i^2 \left(\frac{\lambda_i}{\lambda_1}\right)^{2m}}$$

e quindi  $\lim_{m \rightarrow \infty} \lambda_1^{(m+1)} = \lambda_1$  con velocità di convergenza pari a quella con cui  $|\lambda_2/\lambda_1|^{2m}$  tende a zero.

► **Esempio.**

$$(4.10) \quad A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 0 \\ 3 & 4 & 1 & 2 \\ 4 & 0 & 2 & 3 \end{pmatrix}$$

$m$	Potenze	Quoziente di Rayleigh
1	$0.1000000 \cdot 10^2$	$0.9500000 \cdot 10^1$
2	$0.9400000 \cdot 10^1$	$0.9519337 \cdot 10^1$
3	$0.9553191 \cdot 10^1$	$0.9520695 \cdot 10^1$
4	$0.9512250 \cdot 10^1$	$0.9520789 \cdot 10^1$
5	$0.9523062 \cdot 10^1$	$0.9520797 \cdot 10^1$
6	$0.9520198 \cdot 10^1$	$0.9520798 \cdot 10^1$
7	$0.9520956 \cdot 10^1$	$0.9520798 \cdot 10^1$
8	$0.9520756 \cdot 10^1$	
9	$0.9520808 \cdot 10^1$	
10	$0.9520795 \cdot 10^1$	
11	$0.9520798 \cdot 10^1$	
12	$0.9520798 \cdot 10^1$	

Tabella 4.3



Se la matrice  $A$  non è simmetrica, il quoziente di Rayleigh converge ancora a  $\lambda_1$  (purché non vi siano altri autovalori diversi con lo stesso modulo), ma in quest'ultimo caso la velocità dipende dalla potenza  $|\lambda_2/\lambda_1|^m$ ; esattamente come nell'algoritmo (4.7).

Quando la matrice  $A$  ha un solo autovalore (reale)  $\lambda_n$  di modulo minimo (eventualmente con molteplicità  $k \geq 1$ ) il metodo delle potenze (4.7) può essere utilizzato per costruire una successione di approssimazioni  $\{\lambda_n^{(m)}\}$  convergente a  $\lambda_n$ . È infatti sufficiente ricordare che se  $\lambda_1, \lambda_2, \dots, \lambda_n$  sono gli autovalori di  $A$ , con corrispondenti autovettori

$x_1, x_2, \dots, x_n$ , le quantità  $\lambda_1^{-1}, \lambda_2^{-1}, \dots, \lambda_n^{-1}$  rappresentano gli autovalori di  $A^{-1}$ , e

$$A^{-1}x_i = \lambda_i^{-1}x_i, \quad i = 1, 2, \dots, n$$

La verifica è immediata:

$$Ax_i = \lambda_i x_i \implies x_i = \lambda_i A^{-1}x_i \implies \lambda_i^{-1}x_i = A^{-1}x_i$$

Inoltre, gli autovettori di  $A$  sono anche autovettori di  $A^{-1}$ .

Pertanto, la determinazione dell'autovalore di modulo minimo di  $A$  (e del corrispondente autovettore) è riconducibile alla valutazione dell'autovalore di modulo massimo di  $A^{-1}$ . Questa constatazione ci suggerisce di modificare l'algoritmo (4.7) nel modo seguente:

- 1:  $GA = U$  (Factor)
- 2:  $y_0 \leftarrow (1, 1, \dots, 1)^T$
- 3: **ciclo 1:**  $m = 0, \dots, m_{\max}$
- 4:  $Uw_{m+1} = Gy_m \Rightarrow w_{m+1}$  (Solve)
- 5:  $\lambda_n^{(m+1)} \leftarrow (y_m)_{k_0} / (w_{m+1})_{k_0}$
- 6:  $y_{m+1} \leftarrow w_{m+1} / \|w_{m+1}\|_\infty$
- 7: **fine ciclo 1**

Supponiamo che gli autovalori della matrice  $A$  siano ordinati nel modo seguente:  $|\lambda_1| > |\lambda_2| > |\lambda_3| \geq \dots \geq |\lambda_n|$ ; supponiamo inoltre di aver già calcolato l'autovalore  $\lambda_1$  e l'autovettore  $x_1$  (con il metodo delle potenze, per esempio). Vogliamo determinare l'autovalore successivo  $\lambda_2$  e il corrispondente autovettore  $x_2$ . Nelle righe che seguono illustriamo un procedimento che ci consente di risolvere questo problema utilizzando, dopo una trasformazione iniziale della matrice  $A$ , ancora il metodo delle potenze.

Come vedremo nel paragrafo 4.4, è sempre possibile costruire una matrice ortogonale  $U$  tale che

$$(4.11) \quad Ux_1 = -\sigma e_1, \quad e_1 = (1, 0, \dots, 0)^T, \quad |\sigma| = \|x_1\|_2$$

Con questo risultato, dalla definizione  $Ax_1 = \lambda_1 x_1$ <sup>(†)</sup> deduciamo allora l'identità

$$(UAU^T)Ux_1 = \lambda_1 Ux_1$$

e quindi

$$(4.12) \quad (UAU^T)e_1 = \lambda_1 e_1$$

Le matrici  $A$  e  $(UAU^T)$  hanno gli stessi autovalori (vedi paragrafo 4.4); inoltre, la seconda deve necessariamente avere la forma seguente:

$$UAU^T = \begin{pmatrix} \lambda_1 & a^T \\ o & A_1 \end{pmatrix}$$

---

<sup>(†)</sup> In realtà noi possediamo solo delle approssimazioni di  $\lambda_1$  e  $x_1$ .

dove  $A_1$  è una matrice di ordine  $n-1$  che ha gli stessi autovalori di  $A$  tranne, ovviamente,  $\lambda_1$ . Infatti, posto

$$UAU^T = \begin{pmatrix} \alpha & a^T \\ b & A_1 \end{pmatrix}, \quad \alpha \in \mathbb{R}, \quad a, b \in \mathbb{R}^{n-1}, \quad A_1 \in \mathbb{R}^{(n-1) \times (n-1)}$$

dalla relazione (4.12) otteniamo

$$\begin{pmatrix} \alpha & a^T \\ b & A_1 \end{pmatrix} \begin{pmatrix} 1 \\ o \end{pmatrix} = \begin{pmatrix} \lambda_1 \\ o \end{pmatrix}$$

donde

$$\begin{pmatrix} \alpha \\ b \end{pmatrix} = \begin{pmatrix} \lambda_1 \\ o \end{pmatrix}$$

Pertanto, per determinare  $\lambda_2$  è sufficiente applicare il metodo delle potenze alla nuova matrice  $A_1$  ( $\lambda_2$  è l'autovalore di modulo massimo di  $A_1$ ). Inoltre, dall'autovettore  $\bar{x}_2$  di  $A_1$  corrispondente a  $\lambda_2$  possiamo facilmente risalire all'autovettore  $x_2$  di  $A$ . Infatti, considerando la relazione

$$Ax_2 = \lambda_2 x_2$$

e quindi, utilizzando la matrice ortogonale  $U$  definita in (4.11), la trasformazione

$$UAU^T Ux_2 = \lambda_2 Ux_2$$

ovvero

$$(4.13) \quad \begin{pmatrix} \lambda_1 & a^T \\ o & A_1 \end{pmatrix} \begin{pmatrix} \beta \\ \bar{x}_2 \end{pmatrix} = \lambda_2 \begin{pmatrix} \beta \\ \bar{x}_2 \end{pmatrix}$$

con

$$(4.14) \quad \begin{pmatrix} \beta \\ \bar{x}_2 \end{pmatrix} = Ux_2$$

otteniamo

$$A_1 \bar{x}_2 = \lambda_2 \bar{x}_2$$

e

$$\lambda_1 \beta + a^T \bar{x}_2 = \lambda_2 \beta$$

Da queste ultime due relazioni segue che è sufficiente formare il vettore

$$\begin{pmatrix} \beta \\ \bar{x}_2 \end{pmatrix}, \quad \beta = \frac{a^T \bar{x}_2}{\lambda_2 - \lambda_1}$$

e, ricordando la (4.14), moltiplicarlo a sinistra per la matrice  $U^T$ .

La tecnica descritta ci consente di calcolare i primi autovalori (reali e distinti in modulo) e autovettori della matrice  $A$ . In teoria, quando le ipotesi richieste sono soddisfatte, il metodo potrebbe venire usato per il calcolo di tutti gli autovalori (e autovettori). In

pratica però, nel calcolo degli autovalori successivi al primo si può avere una perdita progressiva di precisione dovuta al fatto che noi possediamo solo delle approssimazioni degli autovettori (che definiscono le matrici  $U$  in (4.11)).

Quando occorre determinare tutti gli autovalori di  $A$ , si rivelano più efficienti metodi basati sulle trasformazioni che descriveremo tra breve nel paragrafo 4.4.

### 4.3 Metodo delle potenze inverse

Il metodo delle potenze può anche venire generalizzato per il calcolo di un autovalore particolare, del quale si ha, per esempio, un'approssimazione iniziale; ovvero, data una buona approssimazione  $p$  di un autovalore di  $A$  possiamo usare il metodo delle potenze non solo per migliorare  $p$ , ma anche per determinare un'approssimazione di un autovettore corrispondente. Posto in questa forma il metodo viene denominato *metodo delle potenze inverse*.

Per dimostrare la validità della nostra asserzione è necessario osservare preliminarmente che quando  $\lambda$  è autovalore di  $A$ , con autovettore  $x$ , cioè

$$Ax = \lambda x$$

possiamo scrivere

$$(A - pI)x = Ax - px = \lambda x - px = (\lambda - p)x$$

e quindi affermare che  $(\lambda - p)$  è autovalore della matrice  $(A - pI)$ , con autovettore  $x$ . Conseguentemente  $(\lambda - p)^{-1}$  è autovalore di  $(A - pI)^{-1}$ , e se  $p$  è sufficientemente vicino ad un autovalore  $\lambda$  di  $A$ , il numero  $\mu = (\lambda - p)^{-1}$  è l'autovalore di modulo massimo di  $(A - pI)^{-1}$ . Pertanto, il calcolo di  $\mu$ , e quindi di  $\lambda = p + 1/\mu$ , può essere effettuato applicando il metodo delle potenze (4.7) alla nuova matrice  $(A - pI)^{-1}$ .

---

#### Algoritmo 5: Invpow( $n, A, \text{toll}, m_{\max}, p, x, \text{ier}$ )

---

*Commento.* Questo algoritmo utilizza il metodo delle potenze inverse per determinare l'autovalore della matrice  $A$  di ordine  $n$  più vicino al numero  $p$  e il corrispondente autovettore  $x$ . Se la precisione relativa richiesta `toll` viene raggiunta con un numero di iterazioni  $\leq m_{\max}$  la variabile `ier` assume il valore 0; altrimenti `ier` = 2. Se la matrice  $A - pI$  risulta singolare `ier` = 1.

*Parametri.* **Input:**  $n, A, \text{toll}, m_{\max}, p$

**Output:**  $p, x, \text{ier}$

- 1:  $(A)_{ii} \leftarrow (A)_{ii} - p, \quad i = 1, \dots, n$
- 2:  $GA = U$  (**richiama** l'algoritmo Factor)
- 3: **se** `ier` = 1 **allora** **esci**
- 4:  $y_0 \leftarrow (1, 1, \dots, 1)^T$
- 5:  $\lambda_p^{(0)} \leftarrow p$

- 
- 6: **ciclo 1:**  $m = 0, \dots, m_{\max}$   
 7:  $Uw_{m+1} = Gy_m \Rightarrow w_{m+1}$  (**richiama** l'algoritmo Solve)  
 8:  $\alpha = \|w_{m+1}\|_\infty$ ; sia  $k_0$  la posizione della prima componente di  $w_{m+1}$  di modulo massimo  
 9:  $y_{m+1} \leftarrow w_{m+1}/\alpha$   
 10:  $\lambda_p^{(m+1)} \leftarrow p + (y_m)_{k_0}/(w_{m+1})_{k_0}$   
 11:  $\text{er} \leftarrow |\lambda_p^{(m)} - \lambda_p^{(m+1)}|$   
 12: **se**  $\text{er} \leq \text{toll} |\lambda_p^{(m+1)}|$  (†) **allora**  $\text{ier} \leftarrow 0$ ; **vai al punto** 15  
 13: **fine ciclo 1**  
 14:  $\text{ier} \leftarrow 2$   
 15:  $x \leftarrow y_{m+1}$   
 16:  $p \leftarrow \lambda_p^{(m+1)}$   
 17: **esci**
- 

Se l'approssimazione iniziale  $p$  non è sufficientemente buona, la convergenza del metodo risulta assai lenta; in questo caso conviene aggiornare ogni  $l \geq 1$  iterazioni la matrice  $A - pI$  con  $A - \lambda_p^{(jl)} I$ ,  $j = 1, 2, \dots$ , e determinare le decomposizioni corrispondenti. Ovviamente nella valutazione della velocità di convergenza bisogna tener conto della mole di calcolo che ogni iterazione comporta. Pertanto, di solito non risulta efficiente prendere, per esempio,  $l = 1, 2$ , in quanto il numero eccessivo di determinazioni di nuove decomposizioni di Gauss potrebbe compromettere l'efficienza del metodo stesso.

## 4.4 Trasformazioni di similitudine e trasformazioni di Householder

Sia  $S$  una matrice non singolare di ordine  $n$ . Si definisce (*trasformazione*) di *similitudine* la trasformazione che associa alla generica matrice  $A$  (di ordine  $n$ ) la matrice  $B$  così definita:

$$B = SAS^{-1}$$

Le matrici  $A$  e  $B$  sono dette *simili*.

La proprietà che rende queste trasformazioni particolarmente utili ai nostri scopi è l'invarianza degli autovalori. Inoltre, se  $Ax = \lambda x$ ,  $y = Sx$  è un autovettore della nuova matrice  $B$  corrispondente all'autovalore  $\lambda$ . Infatti da  $Ax = \lambda x$  otteniamo

$$SAS^{-1}Sx = \lambda Sx$$

e quindi

$$(SAS^{-1})y = \lambda y$$

L'impiego delle trasformazioni di similitudine ci viene anche suggerito dai due risultati seguenti.

---

(†) oppure  $\text{er} \leq \text{toll}$ .

**Teorema 4.2.** *Data una matrice  $A$  reale e simmetrica, esiste una matrice ortogonale  $V_1$  tale che*

$$D = V_1 A V_1^T$$

*è diagonale.*

**Teorema 4.3.** *Data una matrice reale  $A$  di ordine  $n$ , esiste una matrice ortogonale  $V_2$  tale che*

$$H = V_2 A V_2^T$$

*è del tipo di Hessenberg (tridiagonale se  $A$  è simmetrica).*

La dimostrazione del primo teorema è immediata. Infatti, una matrice reale e simmetrica è certamente diagonalizzabile, e la matrice ( $V_1$ ) che ha come colonne gli autovettori normalizzati di  $A$  risulta ortogonale. Il secondo invece si deduce facilmente dal teorema 3.9 in [4.3].

Questi due teoremi ci assicurano l'esistenza di trasformazioni di similitudine capaci di ridurre una generica matrice  $A$  quadrata ad una matrice di forma particolare (diagonale, tridiagonale, di Hessenberg). Tuttavia nulla ci dicono sulla effettiva possibilità di costruire le matrici  $V_1$  e  $V_2$  citate. Anzi, in generale non è possibile trasformare una matrice reale simmetrica in una matrice di forma diagonale attraverso una successione finita di trasformazioni di similitudine.

La possibilità di utilizzare matrici ortogonali per ridurre la matrice  $A$  ad una simile di forma più semplice è, dal punto di vista computazionale, particolarmente interessante, in quanto le inverse delle matrici  $S$  richieste nelle successive trasformazioni vengono a coincidere con le trasposte delle matrici stesse.

Il risultato del teorema 4.3, pur essendo più debole di quello del teorema 4.2, presenta il vantaggio che la matrice  $V_2$  può essere costruita in un numero finito di passi. Le matrici che tra poco definiremo verranno da noi utilizzate proprio per costruire esplicitamente una matrice  $V_2$  con le proprietà indicate nel teorema 4.3.

I metodi per il calcolo degli autovalori che più avanti descriveremo risulterebbero troppo onerosi se applicati a matrici dense. Nasce quindi la necessità di riuscire a trasformare preliminarmente la matrice iniziale  $A$  in una matrice simile di forma tale da rendere poi efficiente (e meno onerosa) la determinazione degli autovalori di quest'ultima. Le forme convenienti per il calcolo degli autovalori, ottenibili con un numero finito di trasformazioni, sono proprio la tridiagonale (per le matrici simmetriche) e quella di Hessenberg (per le matrici non simmetriche), la cui esistenza è peraltro garantita dal teorema 4.3.

**Definizione 4.1.** *Un riflettore elementare è una matrice di forma*

$$U = I - 2uu^T$$

*dove  $u$  è un vettore di lunghezza  $\|u\|_2 = \sqrt{u^T u} = 1$ .*

Le trasformazioni lineari associate a queste matrici sono anche note con il nome di *trasformazioni di Householder*.

La verifica della validità delle proprietà seguenti è pressoché immediata:

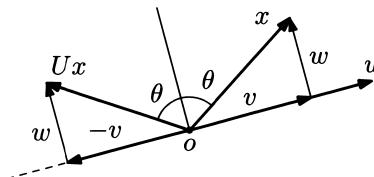
- (i)  $U$  è simmetrica:  $U^T = U$
- (ii)  $U$  è ortogonale:  $U^T U = I$
- (iii)  $U$  è involutoria:  $U^2 = I$

Esaminiamo il significato geometrico della trasformazione  $y = Ux$ . A tale fine prendiamo un generico vettore non nullo  $x$  e denotiamo con  $v$  la sua componente parallela al vettore  $u$  e con  $w$  la sua componente ortogonale a  $u$ , così che  $v = \alpha u$ ,  $\alpha$  scalare, e  $u^T w = 0$ . Posto quindi  $x = v + w$ , calcoliamo

$$Ux = Uv + Uw = v - 2uu^T v + w - 2uu^T w = v + w - 2uu^T v$$

Poiché  $v = \alpha u$ , abbiamo  $u(u^T v) = \alpha u = v$  e

$$Ux = -v + w$$



**Figura 4.4**

Pertanto, geometricamente la matrice  $U$  opera una riflessione del vettore  $x$  rispetto all'asse perpendicolare al vettore  $u$  e passante per l'origine  $o$ .

Il prossimo teorema ci farà vedere come sia possibile utilizzare i riflettori elementari per “introdurre zeri” in un vettore.

**Teorema 4.4.** *Dato un generico vettore non nullo  $x$ , il riflettore elementare*

$$U = I - \frac{1}{\pi} uu^T$$

*con  $u = x + \sigma e_1$ ,  $e_1 = (1, 0, \dots, 0)^T$ ,  $\sigma = \pm \|x\|_2$  e  $\pi = 1/2\|u\|_2^2$ , produce l'effetto seguente:*

$$(4.15) \quad Ux = -\sigma e_1$$

La verifica non è difficile: è sufficiente considerare dapprima il vettore

$$Ux = \left( I - \frac{1}{\pi} uu^T \right) x = x - \frac{u^T x}{\pi} u$$

e successivamente le quantità

$$u^T x = (x + \sigma e_1)^T x = (x^T + \sigma e_1^T)x = x^T x + \sigma e_1^T x = \sigma^2 + \sigma x_1 = \sigma(\sigma + x_1)$$

e

$$\pi = \frac{1}{2}u^T u = \frac{1}{2}(x + \sigma e_1)^T(x + \sigma e_1) = \frac{1}{2}(x^T x + 2\sigma x_1 + \sigma^2) = \sigma(\sigma + x_1)$$

per ottenere

$$Ux = x - u = x - (x + \sigma e_1) = -\sigma e_1$$

Osserviamo che la coppia  $(u, \pi)$ <sup>(†)</sup>, formata da  $n + 1$  numeri reali, è sufficiente per individuare univocamente la matrice  $U$  che ha invece  $n^2$  elementi. Pertanto, dato un vettore non nullo  $x = (\xi_1, \xi_2, \dots, \xi_n)^T$ , costruiamo un algoritmo efficiente che fornisca le quantità  $u$  (che sovrapporremo a  $x$ ) e  $\pi$  (e anche  $\sigma$ ):

```

1:  $\eta \leftarrow \max\{ |\xi_i|, i = 1, \dots, n \}$ 
2:  $\sigma \leftarrow 0$ 
3: ciclo 1:  $i = 1, \dots, n$ 
4:   se  $|\xi_i| \geq \sqrt{\text{eps}} \eta$  allora  $\sigma \leftarrow \sigma + (\xi_i / \eta)^2$ 
5: fine ciclo 1
6:  $\sigma = \text{sgn}(\xi_1) \sqrt{\sigma \eta}$ 
7:  $\xi_1 \leftarrow \xi_1 + \sigma$ 
8:  $\pi \leftarrow \sigma \xi_1$ 
```

- ▷ **Osservazioni.** (i) Il controllo e la normalizzazione  $\xi_i/\eta$  al punto 4 dell'algoritmo ci consentono di trascurare a priori quei numeri il cui contributo in ogni caso non inciderebbe sulla rappresentazione di macchina del risultato, e di eliminare possibili fenomeni di overflow e underflow.
- (ii) L'arbitrarietà del segno di  $\sigma$  in (4.15) viene da noi utilizzata al punto 6 ( $\text{sgn}(\sigma) = \text{sgn}(\xi_1)$ ) per eliminare possibili fenomeni di cancellazione numerica nella “somma”  $\xi_1 + \sigma$  (punto 7).

□

La coppia  $(u, \pi)$  determinata con il precedente algoritmo è sufficiente per costruire esplicitamente prodotti del tipo

$$(4.16) \quad UA = U(a_1, a_2, \dots, a_n) = (Ua_1, Ua_2, \dots, Ua_n)$$

infatti abbiamo

$$Ua_i = \left( I - \frac{1}{\pi}uu^T \right) a_i = a_i - \left( \frac{u^T a_i}{\pi} \right) u$$

Questa osservazione è importante in quanto le matrici  $U$  verranno da noi utilizzate proprio per operazioni di forma (4.16), oppure di tipo  $AU = (UA^T)^T$ .

---

(†) Ovvero il solo vettore  $u$ , dato che  $\pi = 1/2\|u\|_2^2$ .

Dati i vettori  $u = (v_1, v_2, \dots, v_n)^\top$ ,  $a = (\alpha_1, \alpha_2, \dots, \alpha_n)^\top$  ed il numero  $\pi \neq 0$ , la sostituzione di  $a$  con il vettore  $Ua$  può essere effettuata nel modo seguente:

$$\begin{aligned} 1: \tau &\leftarrow \frac{1}{\pi} \sum_{i=1}^n v_i \alpha_i \\ 2: \alpha_i &\leftarrow \alpha_i - \tau v_i, \quad i = 1, \dots, n \end{aligned}$$

Mentre in generale il prodotto di una matrice di ordine  $n$  per un vettore coinvolge  $n^2$  operazioni, la trasformazione  $Ua$  ne richiede solamente  $2n$  (più una divisione).

## 4.5 Applicazioni delle trasformazioni di Householder

### 4.5.1 Fattorizzazione $QR$ di una matrice

In questo paragrafo ci proponiamo di generalizzare l'azione dei riflettori elementari, al fine di poter operare utili trasformazioni di matrici. In particolare, iniziamo ponendoci il seguente problema: dato un vettore  $x = (x_1, x_2, \dots, x_n)^\top$  è possibile determinare un riflettore elementare  $U_k \equiv U_k^{(n)}$  tale che

$$U_k x = (\bar{x}_1, \dots, \bar{x}_k, 0, \dots, 0)^\top?$$

La risposta è affermativa. Infatti, ricordando la caratterizzazione del riflettore elementare  $U \equiv U_1^{(n)}$  definito dal teorema 4.4, possiamo costruire l' $U_k$  cercato nel modo seguente:

$$\begin{aligned} U_k &= \begin{pmatrix} I_{k-1} & O \\ O & U_1^{(n+1-k)} \end{pmatrix} = I - \frac{1}{\pi_k} u_k u_k^\top \\ u_k &= \begin{pmatrix} o \\ u'_k \end{pmatrix}, \quad u'_k \in \mathbb{R}^{n+1-k}, \quad \pi_k = \pi'_k = \frac{1}{2} \|u'_k\|_2^2 \end{aligned}$$

dove  $I_{k-1}$  è la matrice identità di ordine  $(k-1)$  e  $U_1^{(n+1-k)} = I_{n+1-k} - (1/\pi'_k) u'_k (u'_k)^\top$  è il riflettore elementare di ordine  $n+1-k$  definito dalla relazione

$$U_1^{(n+1-k)}(x_k, x_{k+1}, \dots, x_n)^\top = -\sigma_k e_1$$

La matrice  $U_k$  proposta ha anche il pregio di non alterare le prime  $k-1$  componenti del vettore  $x$ <sup>(†)</sup>; ciò significa che del nuovo vettore  $U_k x$  è sufficiente calcolare la  $k$ -esima componente. Osserviamo infine che, fissato l'intero  $k$ , la matrice  $U_k \in \mathbb{R}^{n \times n}$  è univocamente individuata dalla coppia  $(u'_k, \pi'_k)$ , con  $u'_k \in \mathbb{R}^{n+1-k}$ ; inoltre,

$$U_k a = \begin{pmatrix} I_{k-1} & O \\ O & U_1^{(n+1-k)} \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} a_1 \\ U_1^{(n+1-k)} a_2 \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 - \frac{(u'_k)^\top a_2}{\pi'_k} u'_k \end{pmatrix}$$

---

<sup>(†)</sup> Le componenti  $(U_k x)_i$ ,  $i = k+1, \dots, n$ , non devono essere calcolate; conosciamo già a priori il loro valore!

La possibilità di costruire riflettori elementari  $U_k$  con le proprietà suddette potrebbe essere vantaggiosamente utilizzata per ridurre una generica matrice  $A$  alla forma triangolare superiore, e ottenere una decomposizione analoga a quella di Gauss ovvero alla fattorizzazione  $LU$  del paragrafo 3.2.3. Infatti, data una matrice  $A \in \mathbb{R}^{n \times n}$  possiamo costruire  $n - 1$  riflettori elementari  $U_1, U_2, \dots, U_{n-1}$  tali che la nuova matrice

$$U_{n-1} \dots U_2 U_1 A = R$$

assuma la forma triangolare superiore. Definendo poi la matrice  $Q = (U_{n-1} \dots U_2 U_1)^T = U_1 U_2 \dots U_{n-1}$ , che, essendo prodotto di matrici ortogonali, risulta ortogonale, possiamo scrivere

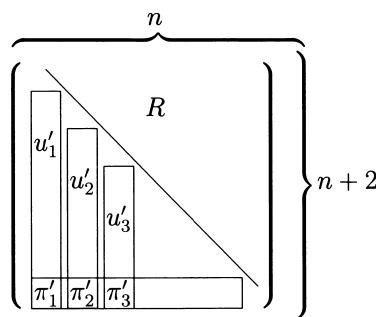
$$(4.17) \quad Q^T A = R$$

ovvero

$$A = QR$$

Il numero operazioni aritmetiche necessarie per costruire queste due ultime decomposizioni è  $2n^3/3$ .

Come per la decomposizione di Gauss, anche in questo caso la matrice  $Q^T$ , ovvero  $Q$ , non verrà mai costruita esplicitamente. Poiché l'unico suo ruolo è quello di trasformare vettori, è sufficiente, come abbiamo già ricordato in 4.4, memorizzare i singoli riflettori  $U_k$ , ovvero le singole coppie  $(u'_k, \pi'_k)$  secondo lo schema descritto dalla seguente tabella di dimensione  $(n + 2) \times n$ :



▷ **Osservazione.** La fattorizzazione  $QR$ , così come la decomposizione di Gauss, può essere conseguita anche quando la matrice  $A$  è rettangolare di tipo  $m \times n$ ; in questo caso abbiamo  $A = QR$  con  $Q \in \mathbb{R}^{m \times m}$  e  $R \in \mathbb{R}^{m \times n}$ . □

Possiamo pertanto affermare:

---

<sup>(†)</sup> La fattorizzazione è ottenuta con  $r = \min\{m - 1, n\}$  riflettori elementari  $U_1, U_2, \dots, U_r$ , e, quando  $m \geq n$  per esempio, con  $mn^2 - 1/3n^3$  operazioni aritmetiche.

**Teorema 4.5.** *Data una matrice  $A \in \mathbb{R}^{m \times n}$  esiste una matrice ortogonale  $Q \in \mathbb{R}^{m \times m}$  tale che*

$$(4.18) \quad A = QR, \quad R \in \mathbb{R}^{m \times n}$$

dove  $(R)_{ij} = 0$ ,  $i > j$ .

Questa fattorizzazione verrà successivamente (vedi paragrafo 4.7) utilizzata nella costruzione del *metodo QR* per il calcolo di tutti gli autovalori di una matrice. La (4.17) ci consentirebbe inoltre di costruire un algoritmo stabile per la risoluzione del sistema  $Ax = b$ ; infatti, otterremmo

$$Rx = Q^T b$$

Tuttavia tale procedimento richiederebbe  $2n^3/3$  operazioni aritmetiche (il metodo di Gauss ne richiede solamente  $n^3/3$ ).

#### 4.5.2 Riduzione di una matrice alla forma simile tridiagonale oppure di Hessenberg

I riflettori elementari  $U_k$  possono anche venire utilizzati per trasformare una matrice simmetrica  $A$  in una matrice simile, e quindi con gli stessi autovalori, di forma tridiagonale. Per realizzare questo obiettivo costruiamo un primo riflettore  $U_2$  che “azzeri” gli elementi  $a_{31}, a_{41}, \dots, a_{n1}$  della prima colonna di  $A$ , e formiamo la nuova matrice, simile ad  $A$ ,

$$A_2 = U_2 A_1 U_2, \quad A_1 \equiv A$$

Posto

$$A_1 = \begin{pmatrix} a_{11} & a_1^T \\ a_1 & A_{11} \end{pmatrix}$$

abbiamo

$$U_2 A_1 U_2 = \begin{pmatrix} 1 & o^T \\ o & U_1^{(n-1)} \end{pmatrix} \begin{pmatrix} a_{11} & a_1^T \\ a_1 & A_{11} \end{pmatrix} \begin{pmatrix} 1 & o^T \\ o & U_1^{(n-1)} \end{pmatrix} = \begin{pmatrix} a_{11} & a_1^T U_1^{(n-1)} \\ U_1^{(n-1)} a_1 & U_1^{(n-1)} A_{11} U_1^{(n-1)} \end{pmatrix}$$

Ma poiché  $U_1^{(n-1)}$  è stato costruito proprio per produrre la trasformazione  $U_1^{(n-1)} a_1 = (x, 0, 0, \dots, 0)^T$ , dove con il simbolo  $x$  denotiamo un numero reale generalmente non nullo, avremo anche  $a_1^T U_1^{(n-1)} = (U_1^{(n-1)} a_1)^T = (x, 0, 0, \dots, 0)$ . Pertanto  $A_2$  assumerà la forma (simmetrica<sup>(†)</sup>)

$$A_2 = \begin{pmatrix} x & x & 0 & 0 & \dots & 0 \\ x & x & x & x & \dots & x \\ 0 & x & x & x & \dots & x \\ 0 & x & x & x & \dots & x \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & x & x & x & \dots & x \end{pmatrix}$$

---

<sup>(†)</sup> Ricordiamo che quando  $A_1$  è simmetrica, cioè  $A_1^T = A_1$ , abbiamo  $(U_2 A_1 U_2)^T = U_2^T A_1^T U_2^T = U_2 A_1 U_2$ .

Successivamente determiniamo il riflettore  $U_3$  in modo che

$$A_3 = U_3 A_2 U_3$$

sia di forma (simmetrica)

$$A_3 = \begin{pmatrix} x & x & 0 & 0 & \dots & 0 \\ x & x & x & 0 & \dots & 0 \\ 0 & x & x & x & \dots & x \\ 0 & 0 & x & x & \dots & x \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & x & x & \dots & x \end{pmatrix}$$

e così proseguiamo sino a raggiungere la matrice finale

$$A_{n-1} = U_{n-1} A_{n-2} U_{n-1}$$

simmetrica, tridiagonale, e simile alla matrice iniziale  $A$ .

► **Esempio.**

$$A_1 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 1 \\ 3 & 4 & 5 & 1 & 2 \\ 4 & 5 & 1 & 2 & 3 \\ 5 & 1 & 2 & 3 & 4 \end{pmatrix}$$

$$U_2 A_1 = \begin{pmatrix} 1.000 & 2.000 & 3.000 & 4.000 & 5.000 \\ 7.348 & -5.852 & -5.035 & -4.899 & -5.443 \\ 0 & 1.160 & 2.101 & -2.177 & -6.771 \cdot 10^{-2} \\ 0 & 1.213 & -2.866 & -2.236 & 2.431 \cdot 10^{-1} \\ 0 & -3.734 & -2.832 & -2.294 & 0.554 \end{pmatrix}$$

$$A_2 = U_2 A_1 U_2 = \begin{pmatrix} 1.000 & 7.348 & 0 & 0 & 0 \\ 7.348 & 1.002 \cdot 10^1 & 5.778 \cdot 10^{-2} & 1.891 & 3.045 \\ 0 & 5.778 \cdot 10^{-2} & 1.747 & -2.648 & 6.569 \cdot 10^{-1} \\ 0 & 1.891 & -2.648 & -1.945 & 6.061 \cdot 10^{-1} \\ 0 & 3.045 & -6.569 \cdot 10^{-1} & 6.061 \cdot 10^{-1} & 4.180 \end{pmatrix}$$

$$U_3 A_2 = \begin{pmatrix} 1.000 & 7.348 & 0 & 0 & 0 \\ 7.348 & 1.002 \cdot 10^1 & 5.778 \cdot 10^{-2} & 1.891 & 3.045 \\ 0 & 3.585 & 1.923 & 5.541 \cdot 10^{-1} & -3.859 \\ 0 & 0 & -2.555 & -2.823 \cdot 10^{-1} & -1.057 \\ 0 & 0 & -5.065 \cdot 10^{-1} & 3.283 & 1.503 \end{pmatrix}$$

$$A_3 = U_3 A_2 U_3 = \begin{pmatrix} 1.000 & 7.348 & 0 & 0 & 0 \\ 7.348 & 1.002 \cdot 10^1 & 3.585 & 0 & 0 \\ 0 & 3.585 & 2.954 & 1.087 & -3.000 \\ 0 & 0 & 1.087 & 1.609 & 1.988 \\ 0 & 0 & -3.000 & 1.988 & -5.815 \cdot 10^{-1} \end{pmatrix}$$

$$U_4 A_3 = \begin{pmatrix} 1.000 & 7.348 & 0 & 0 & 0 \\ 7.348 & 1.002 \cdot 10^1 & 3.585 & 0 & 0 \\ 0 & 3.585 & 2.954 & 1.087 & -3.000 \\ 0 & 0 & 3.191 & 1.321 & -1.224 \\ 0 & 0 & 0 & 2.190 & 1.671 \end{pmatrix}$$

$$A_4 = U_4 A_3 U_4 = \begin{pmatrix} 1.000 & 7.348 & 0 & 0 & 0 \\ 7.348 & 1.002 \cdot 10^1 & 3.585 & 0 & 0 \\ 0 & 3.585 & 2.954 & 3.191 & 0 \\ 0 & 0 & 3.191 & -1.601 & 8.244 \cdot 10^{-1} \\ 0 & 0 & 0 & 8.244 \cdot 10^{-1} & 2.628 \end{pmatrix}$$

◀

Se la matrice iniziale  $A$  non è simmetrica il procedimento descritto conduce ad una matrice finale  $A_{n-1}$  di tipo Hessenberg superiore.

Riassumendo, con le precedenti trasformazioni di similitudine otteniamo

$$A_{n-1} = U_{n-1} \dots U_3 U_2 A U_2 U_3 \dots U_{n-1} = VAV^T$$

con  $V = U_{n-1} \dots U_3 U_2$  matrice ortogonale. Quest'ultima matrice, che abbiamo costruito mediante il prodotto di  $n-2$  riflettori elementari, definisce una matrice  $V_2$  del teorema 4.3. Osserviamo tuttavia che la matrice  $V$  non verrà mai costruita esplicitamente. Come già abbiamo rilevato in precedenza, nelle applicazioni di queste trasformazioni è sufficiente, e più efficiente, memorizzare i singoli fattori  $U_k$ , cioè le singole coppie  $(u'_k, \pi'_k)$ .

L'algoritmo che segue trasforma una matrice simmetrica  $A$  in una matrice simile di forma tridiagonale (simmetrica) ed illustra la memorizzazione dei singoli fattori  $U_k$ ,  $k = 2, 3, \dots, n-1$ .

---

**Algoritmo 6:** Tridig( $n, A, d, c$ )
 

---

*Commento.*  $A$  è una matrice reale simmetrica di ordine  $n$ .

L'algoritmo determina dei riflettori elementari  $U_2, U_3, \dots, U_{n-1}$  che rendono la nuova matrice  $A_{n-1} = U_{n-1} \dots U_3 U_2 A U_2 U_3 \dots U_{n-1}$  di forma tridiagonale simmetrica. Gli elementi diagonale di  $A_{n-1}$  sono memorizzati nel vettore  $d$ , mentre quelli della codiagonale nel vettore  $c$ .

Le  $(n+1-k)$  componenti del vettore  $u'_k$  necessarie per definire il riflettore  $U_k$ ,  $k = 2, \dots, n-1$ , vengono sovrapposte alle ultime  $n+1-k$  componenti della  $(k-1)$ -esima colonna di  $A$ ; il numero  $\pi'_k$  è sovrapposto a  $a_{k-1,k-1} = (A)_{k-1,k-1}$ . Gli elementi  $a_{ij}$ ,  $i < j$ , vengono lasciati inalterati.

Se al passo  $k$ -esimo non è richiesta alcuna trasformazione viene posto  $\pi'_k = 0$ .

*Parametri.* **Input:**  $n, A$

**Output:**  $A, d, c$

- 1: **ciclo 1:**  $k = 1, \dots, n - 2$
- 2:    $d_k \leftarrow a_{kk}$
- \*:   *determinazione del riflettore  $U_{k+1}$*
- 3:    $\eta \leftarrow \max\{|a_{ik}|, i = k + 1, \dots, n\}$
- 4:   **se**  $\eta = 0$  **allora**  $a_{kk} \leftarrow 0, c_k \leftarrow 0$ ; **vai al punto 21**
- 5:    $\sigma \leftarrow 0$
- 6:   **ciclo 2:**  $i = k + 1, \dots, n$
- 7:     **se**  $|a_{ik}| \geq \sqrt{\text{eps}} \eta$  **allora**  $\sigma \leftarrow \sigma + (a_{ik}/\eta)^2$
- 8:   **fine ciclo 2**
- 9:    $\sigma \leftarrow \text{sgn}(a_{k+1,k})\sqrt{\sigma} \eta$
- 10:    $a_{k+1,k} \leftarrow a_{k+1,k} + \sigma$
- 11:    $a_{k,k} \leftarrow \sigma a_{k+1,k}$
- 12:    $c_k \leftarrow -\sigma$
- \*:   *trasformazione  $U_{k+1}A_k = A'_k$*
- 13:   **ciclo 3:**  $j = k + 1, \dots, n$
- 14:      $\tau \leftarrow \left(\sum_{i=k+1}^n a_{ik}a_{ij}\right) / a_{kk}$
- 15:      $a_{ij} \leftarrow a_{ij} - \tau a_{ik}, \quad i = k + 1, \dots, n$
- 16:   **fine ciclo 3**
- \*:   *trasformazione  $A'_k U_{k+1} = A_{k+1}$*
- 17:   **ciclo 4:**  $i = k + 1, \dots, n$
- 18:      $\tau \leftarrow \left(\sum_{j=k+1}^n a_{ij}a_{jk}\right) / a_{kk}$
- 19:      $a_{ij} \leftarrow a_{ij} - \tau a_{jk}, \quad j = k + 1, \dots, n$
- 20:   **fine ciclo 4**
- 21: **fine ciclo 1**
- 22:  $d_{n-1} \leftarrow a_{n-1,n-1}$
- 23:  $d_n \leftarrow a_{n,n}$
- 24:  $c_{n-1} \leftarrow a_{n,n-1}$
- 25: **esci**

Illustriamo la sequenza delle trasformazioni operate dall'algoritmo Tridig applicando quest'ultimo alla matrice dell'esempio precedente:

$$U_2 A_1 = \begin{pmatrix} \pi'_2 \\ 6.870 \cdot 10^1 & 2.000 & 3.000 & 4.000 & 5.000 \\ 9.348 & -5.852 & -5.035 & -4.899 & -5.443 \\ 3.000 & 1.159 & 2.101 & -2.177 & -6.771 \cdot 10^{-2} \\ 4.000 & 1.213 & -2.866 & -2.236 & 2.431 \cdot 10^{-1} \\ 5.000 & u'_2 & -3.734 & -2.832 & -2.294 & 5.538 \cdot 10^{-1} \end{pmatrix}$$

$$D = \begin{pmatrix} 1.000 \\ \\ \\ \\ \end{pmatrix} \quad C = \begin{pmatrix} -7.348 \\ \\ \\ \end{pmatrix}$$

$$A_2 = \begin{pmatrix} 6.870 \cdot 10^1 & 2.000 & 3.000 & 4.000 & 5.000 \\ 9.348 & 1.002 \cdot 10^1 & 5.778 \cdot 10^{-2} & 1.891 & 3.045 \\ 3.000 & 5.778 \cdot 10^{-2} & 1.747 & -2.648 & -6.569 \cdot 10^{-1} \\ 4.000 & 1.891 & -2.648 & -1.945 & 6.061 \cdot 10^{-1} \\ 5.000 & 3.045 & -6.569 \cdot 10^{-1} & 6.061 \cdot 10^{-1} & 4.180 \end{pmatrix}$$

$$U_3 A_2 = \begin{pmatrix} 6.870 \cdot 10^1 & 2.000 & 3.000 & 4.000 & 5.000 \\ 9.348 & 1.306 \cdot 10^1 & 5.778 \cdot 10^{-2} & 1.891 & 3.045 \\ 3.000 & 3.643 & \pi'_3 & 1.927 & 5.541 \cdot 10^{-1} -3.859 \\ 4.000 & 1.891 & -2.555 & -2.823 \cdot 10^{-1} & -1.057 \\ 5.000 & 3.045 & u'_3 & -5.065 \cdot 10^{-1} & 3.283 & 1.503 \end{pmatrix}$$

$$D = \begin{pmatrix} 1.000 \\ 1.002 \cdot 10^1 \\ \\ \\ \end{pmatrix} \quad C = \begin{pmatrix} -7.348 \\ \\ -3.585 \end{pmatrix}$$

$$A_3 = \begin{pmatrix} 6.870 \cdot 10^1 & 2.000 & 3.000 & 4.000 & 5.000 \\ 9.348 & 1.306 \cdot 10^1 & 5.778 \cdot 10^{-2} & 1.891 & 3.045 \\ 3.000 & 3.643 & 2.954 & 1.087 & -3.000 \\ 4.000 & 1.891 & 1.087 & 1.609 & 1.988 \\ 5.000 & 3.045 & -3.000 & 1.988 & -5.815 \cdot 10^{-1} \end{pmatrix}$$

$$U_4 A_3 = \begin{pmatrix} 6.870 \cdot 10^1 & 2.000 & 3.000 & 4.000 & 5.000 \\ 9.348 & 1.306 \cdot 10^1 & 5.778 \cdot 10^{-2} & 1.891 & 3.045 \\ 3.000 & 3.643 & \pi'_4 & 1.366 \cdot 10^1 & 1.087 -3.000 \\ 4.000 & 1.891 & 4.279 & 1.321 & -1.224 \\ 5.000 & 3.045 & u'_4 & -3.000 & 2.190 & 1.671 \end{pmatrix}$$

$$D = \begin{pmatrix} 1.000 \\ 1.002 \cdot 10^1 \\ 2.954 \end{pmatrix} \quad C = \begin{pmatrix} -7.348 \\ -3.585 \\ -3.191 \end{pmatrix}$$

$$A_4 = \begin{pmatrix} 6.870 \cdot 10^1 & 2.000 & 3.000 & 4.000 & 5.000 \\ 9.348 & 1.306 \cdot 10^1 & 5.778 \cdot 10^{-2} & 1.891 & 3.045 \\ 3.000 & 3.643 & 1.366 \cdot 10^1 & 1.087 & -3.000 \\ 4.000 & 1.891 & 4.279 & -1.601 & 8.244 \cdot 10^{-1} \\ 5.000 & 3.045 & -3.000 & 8.244 \cdot 10^{-1} & 2.628 \end{pmatrix}$$

$$D = \begin{pmatrix} 1.000 \\ 1.002 \cdot 10^1 \\ 2.954 \\ -1.601 \\ 2.628 \end{pmatrix} \quad C = \begin{pmatrix} -7.348 \\ -3.585 \\ -3.191 \\ -8.244 \cdot 10^{-1} \end{pmatrix}$$

### 4.5.3 Decomposizione ai valori singolari

I riflettori elementari sono alla base anche di un'altra importante fattorizzazione di una generica matrice  $A \in \mathbb{R}^{m \times n}$ , alternativa a quelle finora viste. Essa è denominata *decomposizione ai valori singolari* e viene di solito indicata con l'acronimo *SVD*, dai termini inglesi Singular Value Decomposition.

Si ha infatti il seguente risultato, la cui dimostrazione è reperibile nei testi [4.9] e [5.11].

**Teorema 4.6.** *Data  $A \in \mathbb{R}^{m \times n}$  esistono due matrici ortogonali  $U = (u_1, \dots, u_m) \in \mathbb{R}^{m \times m}$  e  $V = (v_1, \dots, v_n) \in \mathbb{R}^{n \times n}$  tali che*

$$(4.19) \quad U^T A V = S, \quad A = U S V^T$$

dove  $S \in \mathbb{R}^{m \times n}$  è diagonale, ovvero

$$(S)_{ij} = \begin{cases} 0, & i \neq j \\ \sigma_i, & i = j \end{cases}$$

con  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0$ ,  $p = \min\{m, n\}$ .

Le matrici  $U$  e  $V$  non sono uniche. Algoritmi per l'effettiva costruzione di una coppia di matrici  $U$  e  $V$  che realizzino la predetta decomposizione sono riportati in [4.9] e [5.11]. Il costo computazionale della costruzione delle matrici  $U$ ,  $V$  e  $S$ , con l'algoritmo presentato in [4.9], è di  $2m^2n + 4mn^2 + 14n^3/3$  operazioni aritmetiche.

Gli elementi diagonale  $\{\sigma_i\}$  della matrice  $S$  sono detti *valori singolari* della matrice  $A$ . Essi coincidono con la radice quadrata degli autovalori (tutti non negativi) della matrice

$A^T A$ . I vettori  $\{u_i\}$  e  $\{v_i\}$  sono invece definiti rispettivamente *vettori singolari sinistri* e *vettori singolari destri* della matrice  $A$ ; essi soddisfano le relazioni seguenti:

$$\begin{aligned} Av_i &= \sigma_i u_i & i = 1, \dots, p \\ A^T u_i &= \sigma_i v_i \end{aligned}$$

I valori singolari di  $A$  hanno un importante significato geometrico: essi rappresentano le lunghezze dei semiassi dell'iperellissoido

$$E = \{ y : y = Ax, \|x\|_2 = 1 \}$$

Ricordiamo inoltre che la SVD caratterizza appieno la trasformazione lineare definita dalla matrice  $A$ , come si evince dal seguente risultato, dove con  $N(A)$  denotiamo il *nucleo* della trasformazione e con  $R(A)$  lo spazio immagine.

**Teorema 4.7.** *Se in (4.19) risulta*

$$\sigma_1 \geq \dots \geq \sigma_r > \sigma_{r+1} = \dots = \sigma_p = 0, \quad r < p$$

*allora*

- (i) *il rango di  $A$  è  $r$ ;*
- (ii) *i vettori  $u_1, \dots, u_r$  sono una base di  $R(A)$ ;*
- (iii) *i vettori  $v_{r+1}, \dots, v_n$  sono una base di  $N(A)$ ;*

$$(iv) \quad A = \sum_{i=1}^r \sigma_i u_i v_i^T = U_r S_r V_r^T, \text{ con}$$

$$U_r = (u_1, \dots, u_r), \quad V_r = (v_1, \dots, v_r), \quad (S_r)_{ij} = \begin{cases} 0, & i \neq j \\ \sigma_i, & i = j \end{cases}$$

$$(v) \quad \|A\|_2 = \sigma_1$$

Si osservi che dalla (v) si deduce, per esempio, che quando  $A \in \mathbb{R}^{n \times n}$  è non singolare, il numero di condizionamento spettrale associato alla matrice  $A$  è definito a pagina 40 ammette la seguente espressione:

$$K_2(A) = \frac{\sigma_1}{\sigma_n}$$

**Teorema 4.8.** *Sia data la SVD di una matrice  $A \in \mathbb{R}^{m \times n}$ , e supponiamo che il rango di  $A$  sia  $r$ . Se, fissato un intero positivo  $k < r$ , definiamo*

$$A_k = \sum_{i=1}^k \sigma_i u_i v_i^T$$

e

$$\mathcal{B} = \{ B \in \mathbb{R}^{m \times n} : \text{rango}(B) = k \}$$

allora risulta

$$\min_{B \in \mathcal{B}} \|A - B\|_2 = \|A - A_k\|_2 = \sigma_{k+1}$$

Quest'ultimo risultato ci consente di quantificare esattamente la distanza, in norma 2, della matrice  $A$  dall'insieme delle matrici di rango  $k$ . Inoltre, la matrice  $A_k$  rappresenta la migliore approssimazione (in norma 2) di rango  $k$  della matrice  $A$ .

La decomposizione *SVD* trova importanti applicazioni. Per esempio essa costituisce lo strumento più efficace per la determinazione del rango di una matrice ([4.3], [4.9]). Viene inoltre utilizzata per risolvere il problema lineare dei minimi quadrati (caso discreto) quando quest'ultimo risulta mal condizionato ([4.9], [5.5]).

## 4.6 Calcolo degli autovalori di una matrice tridiagonale simmetrica.

In questo paragrafo presentiamo uno dei primi metodi che la letteratura specializzata ha proposto per il calcolo degli autovalori di una matrice tridiagonale simmetrica

$$(4.20) \quad \begin{pmatrix} \alpha_1 & \beta_1 & 0 & \dots & 0 & 0 \\ \beta_1 & \alpha_2 & \beta_2 & \dots & 0 & 0 \\ 0 & \beta_2 & \alpha_3 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \alpha_{n-1} & \beta_{n-1} \\ 0 & 0 & 0 & \dots & \beta_{n-1} & \alpha_n \end{pmatrix}$$

Esso consiste nella ricerca degli zeri del polinomio caratteristico  $\det(\lambda I - B)$ , che nel caso di una matrice tridiagonale simmetrica può essere “individuato” con notevoli semplificazioni di calcolo rispetto al caso generale.

Non perdiamo nulla in generalità nel supporre sin dall'inizio che la matrice  $B$  abbia tutti gli elementi  $\beta_i$  non nulli. Infatti, quando qualche  $\beta_i$  è nullo possiamo decomporre  $\det(\lambda I - B)$  nel prodotto di determinanti  $\det(\lambda I_j - B_j)$  associati a sottomatrici  $B_j$  tridiagonali simmetriche con tutti i  $\beta_i$  diversi da zero, e applicare il metodo ad ogni singolo fattore. Per esempio, nel caso della matrice

$$B = \left( \begin{array}{ccc|cc} \alpha_1 & \beta_1 & 0 & 0 & 0 \\ \beta_1 & \alpha_2 & \beta_2 & 0 & 0 \\ 0 & \beta_2 & \alpha_3 & 0 & 0 \\ \hline 0 & 0 & 0 & \alpha_4 & \beta_4 \\ 0 & 0 & 0 & \beta_4 & \alpha_5 \end{array} \right) \quad (\beta_3 = 0)$$

possiamo porre

$$\det(\lambda I - B) = \det(\lambda I_1 - B_1) \cdot \det(\lambda I_2 - B_2)$$

dove ora le singole sottomatrici

$$B_1 = \begin{pmatrix} \alpha_1 & \beta_1 & 0 \\ \beta_1 & \alpha_2 & \beta_2 \\ 0 & \beta_2 & \alpha_3 \end{pmatrix} \quad \text{e} \quad B_2 = \begin{pmatrix} \alpha_4 & \beta_4 \\ \beta_4 & \alpha_5 \end{pmatrix}$$

hanno tutti gli elementi  $\beta_i \neq 0$ , e applicare il metodo prima a  $B_1$  e poi a  $B_2$ .

Proseguiamo pertanto supponendo in (4.20) tutti i  $\beta_i \neq 0$  e scriviamo

$$\lambda I - B = \left( \begin{array}{ccccc|cc} (\lambda - \alpha_1) & -\beta_1 & 0 & 0 & \dots & 0 \\ -\beta_1 & (\lambda - \alpha_2) & -\beta_2 & 0 & \dots & 0 \\ 0 & -\beta_2 & (\lambda - \alpha_3) & -\beta_3 & \dots & 0 \\ 0 & 0 & -\beta_3 & (\lambda - \alpha_4) & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & (\lambda - \alpha_n) \end{array} \right)$$

consideriamo poi la seguente successione di funzioni (anzi, polinomi in  $\lambda$ ):

$$(4.21) \quad \begin{cases} p_0(\lambda) = 1 \\ p_1(\lambda) = \lambda - \alpha_1 \\ \vdots \\ p_i(\lambda) = (\lambda - \alpha_i)p_{i-1}(\lambda) - \beta_{i-1}^2 p_{i-2}(\lambda), \quad i = 2, 3, \dots, n \end{cases}$$

Esaminiamo il significato di questi polinomi  $p_i(\lambda)$ , di grado  $i$  nella variabile  $\lambda$ :  $p_1(\lambda) = \lambda - \alpha_1$  può essere interpretato come il determinante associato alla sottomatrice principale di ordine 1 di  $(\lambda I - B)$ ,  $p_2(\lambda) = (\lambda - \alpha_1)(\lambda - \alpha_2) - \beta_1^2$  come il determinante della sottomatrice principale di ordine 2 di  $(\lambda I - B)$ , ecc. L'ultimo di questi polinomi,  $p_n(\lambda)$ , rappresenta il determinante di  $(\lambda I - B)$ , ovvero il polinomio caratteristico che stiamo cercando.

Il risultato seguente è strumento indispensabile per la localizzazione degli autovalori di  $B$ .

**Teorema 4.9.** ([4.10, pag. 343]). *Sia  $B$  una matrice tridiagonale, reale e simmetrica, con  $\beta_i \neq 0$ ,  $i = 1, \dots, n-1$ . Gli zeri di ogni polinomio  $p_i(\lambda)$ ,  $i = 2, \dots, n$ , definito dalla (4.21), sono tutti reali e distinti; inoltre gli zeri di  $p_i(\lambda)$  si alternano con quelli di  $p_{i-1}(\lambda)$ .*

*Fissato un reale  $\gamma$ , se  $p_n(\gamma) \neq 0$  il numero di zeri di  $p_n(\lambda)$ , quindi autovalori di  $B$ , che si trovano alla destra di  $\gamma$  è uguale al numero di variazioni di segno nella successione (di numeri)*

$$(4.22) \quad 1, p_1(\gamma), p_2(\gamma), \dots, p_{n-1}(\gamma), p_n(\gamma)$$

*ignorando gli eventuali zeri.*

Supponiamo di aver individuato un intervallo  $[a_0, b_0]$  contenente tutti gli autovalori di  $B$  (per esempio con il teorema di Gershgorin). Prendiamo  $\gamma = (a_0 + b_0)/2$  e costruiamo la successione (4.22). Il teorema 4.9 ci permette di conoscere il numero di zeri (autovalori) di  $p_n(\lambda)$  appartenenti all'intervallo  $(\gamma, b_0)$ , quindi il numero di zeri in  $(a_0, \gamma)$ . Suddividiamo poi ognuno di questi due intervalli in due parti uguali

$$\overbrace{\phantom{aaaaaaa}}^{a_0} \quad \overbrace{\phantom{aaaaaaa}}^{\gamma_1} \quad \overbrace{\phantom{aaaaaaa}}^{\gamma} \quad \overbrace{\phantom{aaaaaaa}}^{\gamma_2} \quad \overbrace{\phantom{aaaaaaa}}^{b_0}$$

esaminando i segni della successione (4.22) nei due nuovi punti  $\gamma_1$  e  $\gamma_2$  siamo in grado di determinare il numero esatto di autovalori contenuti in ognuno dei quattro sottointervalli. Procedendo con questa tecnica di *bisezione* è possibile individuare degli intervalli di ampiezza sempre più piccola (anzi, tendente a zero), ognuno dei quali contiene esattamente un autovalore. Come approssimazione di ogni singolo autovalore  $\lambda_i$  conviene prendere il punto medio  $m_i = (a_i + b_i)/2$  dell'intervallo  $(a_i, b_i)$  contenente  $\lambda_i$ ; in questo caso abbiamo

$$|\lambda_i - m_i| < \frac{b_i - a_i}{2}$$

Per una più dettagliata analisi del metodo di bisezione vedi pagina 183.

Nella costruzione della successione  $\{p_i(\lambda)\}$  può presentarsi, anche nel caso di matrici di ordine modesto, il fenomeno di overflow o di underflow. Per evitare tale inconveniente è sufficiente sostituire la successione  $\{p_i(\lambda)\}$  con  $\{q_i(\lambda)\}$ , dove

$$q_i(\lambda) = \frac{p_i(\lambda)}{p_{i-1}(\lambda)}, \quad i = 1, \dots, n$$

Il numero di autovalori situati a destra del numero  $\gamma$  è ora data dal numero di segni negativi presenti nella nuova successione  $\{q_i(\gamma)\}$ . Le quantità  $q_i(\lambda)$  vengono determinate con la formula ricorsiva

$$(4.23) \quad \begin{cases} q_1(\lambda) = \lambda - \alpha_1 \\ q_i(\lambda) = (\lambda - \alpha_i) - \beta_{i-1}^2/q_{i-1}(\lambda), \quad i = 2, \dots, n \end{cases}$$

facilmente deducibile dalla (4.21). Se  $q_{i-1}(\lambda)$  risulta uguale a zero, è sufficiente sostituire  $q_{i-1}(\lambda)$  con un numero sufficientemente piccolo.

Poiché la velocità di convergenza di tale metodo è modesta, individuate delle approssimazioni sufficientemente buone degli autovalori possiamo utilizzare il metodo delle potenze inverse per migliorare tali approssimazioni e, contemporaneamente, determinare i corrispondenti autovettori.

## 4.7 Cenni sul metodo $QR$

Il metodo generale più efficiente e più usato per il calcolo di tutti gli autovalori, ed eventualmente autovettori, di una matrice, simmetrica e non, è attualmente il metodo

*QR*. La sua descrizione è assai complessa, per cui ci limiteremo alla presentazione delle idee principali che ne stanno alla base.

Il nucleo del metodo è costituito dalla fattorizzazione *QR* di una matrice e dalla costruzione della seguente successione:

$$\left. \begin{array}{l} A_1 = A \\ A_i = Q_i R_i \\ A_{i+1} = R_i Q_i \end{array} \right\} \quad i = 1, 2, \dots$$

Tutte le matrici  $A_i$  risultano simili ad  $A$ ; infatti

$$A_{i+1} = (Q_1 Q_2 \dots Q_i)^T A (Q_1 Q_2 \dots Q_i)$$

Inoltre, l'eventuale simmetria di  $A$  è preservata in tutte le matrici  $A_i$ . Ogni iterazione del metodo comporta la determinazione di una fattorizzazione *QR*. Quando la matrice  $A$  è densa, la decomposizione *QR* risulta eccessivamente onerosa ed il metodo non appare competitivo. Conviene invece ridurre dapprima la matrice  $A$ , mediante trasformazioni ortogonali di similitudine, alla forma tridiagonale (se  $A$  è simmetrica) oppure di Hessenberg superiore (se  $A$  non è simmetrica), e poi applicare il metodo *QR*. Le successive matrici  $A_i$  conservano la forma di  $A$ ; inoltre, esse convergono alla forma triangolare superiore (con elementi diagonale coincidenti con gli autovalori di  $A$ ), oppure (per esempio nel caso di autovalori complessi) alla forma *quasi-triangolare*<sup>(†)</sup> con blocchi diagonale  $2 \times 2$  associati alle coppie di autovalori complessi coniugati. Per accelerare la convergenza vengono introdotti dei parametri  $t_i$  “opportunamente” scelti, dando quindi al metodo la nuova forma:

$$\left\{ \begin{array}{l} (A_i - t_i I) = Q_i R_i \\ A_{i+1} = R_i Q_i + t_i I \end{array} \right.$$

---

(†) Una matrice quasi-triangolare è una matrice triangolare a blocchi, con blocchi diagonale quadrati di ordine al più 2.

## Bibliografia

- [4.1] J. H. Wilkinson, *The algebraic eigenvalue*, Oxford University Press, New York, 1965.
- [4.2] J. H. Wilkinson, C. Reinsch, *Handbook for automatic computation, Vol II: Linear algebra*, Springer-Verlag, New Yourk, 1971.
- [4.3] G. W. Stewart, *Introduction to matrix computation*, Academic Press, New York, 1973.
- [4.4] A. R. Gourlay, G. A. Watson, *Computations methods for matrix eigenproblems*, John Wile & Sons, New York, 1973.
- [4.5] B. T. Smith, et. al., *Matrix eigensystems routines – EISPACK Guide*, Lecture Notes in Computer Science 6, Springer-Verlag, Heidelberg, 1976.
- [4.6] B. S. Garbow, et. al., *Matrix eigensystems routines – EISPACK Guide extension*, Lecture Notes in Computer Science 51, Springer-Verlag, Heidelberg, 1977.
- [4.7] A. Jennings, *Matrix computation for engineers and scientists*, John Wiley & Sons, New York, 1977.
- [4.8] B. N. Parlett, *The symmetric eigenvalue problem*, Prentice Hall, Englewood Cliffs, New Jersey, 1980.
- [4.9] G. H. Golub, C. Van Loan, *Matrix computations*, John Hopkins Univ. Press, Baltimore, 1983.
- [4.10] D. Bini, M. Capovani, O. Menchi, *Metodi numerici per l'algebra lineare*, Zanichelli, Bologna, 1988.
- [4.11] F. Chatelin, *Eigenvalues of matrices*, John Wiley & Sons, Chichester, 1993.
- [4.12] D. S. Watkins, *Fundamentals of matrix computation*, John Wiley & Sons, Chichester, 2002.

## Esercizi proposti

**4.1.** Localizzare gli autovalori della matrice seguente:

$$A = \begin{pmatrix} 1 & 2 & -1 \\ 2 & 7 & 0 \\ -1 & 0 & -5 \end{pmatrix}$$

**4.2.** Ad ogni polinomio  $p_n(t) = t^n + a_1t^{n-1} + \dots + a_n$  possiamo associare la matrice

$$A = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 0 & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \dots & -a_2 & -a_1 \end{pmatrix}$$

che ha come autovalori proprio le radici di  $p_n(t)$ . Verificare che  $p_n(t)$  è il polinomio caratteristico di  $A$ . Quali sono gli autovettori?

**4.3.** Siano dati una matrice  $A \in \mathbb{R}^{n \times n}$  e i numeri reali  $a_0, a_1, \dots, a_n$ . Costruiamo la nuova matrice  $P(A) = a_0I + a_1A + \dots + a_nA^n$ . Dimostrare che se  $\lambda$  è autovalore di  $A$  con autovettore  $x$  allora  $\mu = P(\lambda)$  è autovalore di  $P(A)$  con autovettore  $x$ .

**4.4.** Supponiamo di avere una routine capace di calcolare tutti gli autovalori di una matrice. Volendo determinare gli autovalori della matrice

$$B = 2I + A + 3A^2 - A^3$$

come conviene procedere?

**4.5.** Una matrice  $T$  è detta *triangolare superiore a blocchi* se può essere posta nella forma

$$T = \begin{pmatrix} T_{11} & T_{12} & \dots & T_{1n} \\ O & T_{22} & \dots & T_{2n} \\ \dots & \dots & \dots & \dots \\ O & O & \dots & T_{nn} \end{pmatrix}$$

dove ogni blocco diagonale  $T_{ii}$  è quadrato. Dimostrare che gli autovalori di  $T$  coincidono con gli autovalori di tutti i blocchi diagonale  $T_{ii}$ ,  $i = 1, \dots, n$ .

**4.6.** Sia data una matrice  $A$  di forma

$$A = \begin{pmatrix} D & \dots & \dots & \dots & \dots \\ O & D & \dots & \dots & \dots \\ O & O & D & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ O & O & O & \dots & D \end{pmatrix}$$

dove  $D \in \mathbb{R}^{m \times m}$ . Come conviene procedere per determinare gli autovalori di  $A$ ?

**4.7.** Dimostrare che il quoziente di Rayleigh  $v_m^T A v_m / (v_m^T v_m)$ , con il vettore  $v_m$  definito dalla relazione  $v_m = A^m v_0$ , quando  $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots$  e gli autovettori  $x_1, x_2, \dots, x_n$  sono linearmente indipendenti converge a  $\lambda_1 (m \rightarrow \infty)$ .

**4.8.** Applicare il metodo delle potenze alla matrice

$$\begin{pmatrix} -4 & -5 & 5 \\ 14 & 15 & -5 \\ -1 & -1 & 11 \end{pmatrix}$$

i cui autovalori sono 11, 10, 1, e calcolare l'autovalore di massimo modulo. Nell'algoritmo delle potenze di pagina 92 scegliere dapprima  $k_0 = 1$ , poi  $k_0 = 2$ . Spiegare la diversa velocità del metodo nei due casi.

**4.9.** Applicare il metodo delle potenze (4.7), con l'indice  $k_0$  scelto in modo che  $|(\omega_m)_{k_0}| = \|\omega_m\|_\infty$ , alla matrice

$$A = \begin{pmatrix} -1 & 1 & -1 \\ 5 & 3 & 1 \\ 8 & 8 & -4 \end{pmatrix}$$

i cui autovalori sono  $\lambda_1 = 4$ ,  $\lambda_2 = -4$ ,  $\lambda_3 = -2$ . Successivamente applicare il metodo prendendo  $k_0 = 1$  e spiegare i risultati.

**4.10.** Riprendendo la dimostrazione della convergenza del metodo delle potenze e supponendo  $\lambda_1 = -\lambda_2$ , considerare il rapporto  $(v_{m+2})_k / (v_m)_k$  e dimostrare che esso converge a  $\lambda_1^2$ .

**4.11.** Calcolare l'autovalore di modulo massimo della matrice  $A^T A$  senza costruire esplicitamente il prodotto.

**4.12.** Nell'esercizio precedente supporre  $A \in \mathbb{R}^{n \times n}$  non singolare e determinare l'autovalore di modulo minimo.

**4.13.** In quale situazione il metodo delle potenze inverse non converge?

**4.14.** Costruire un algoritmo che determini la fattorizzazione  $QR$  di una generica matrice  $A \in \mathbb{R}^{m \times n}$ ,  $m \leq n$ .

**4.15.** Richiamando la decomposizione  $QR$  di una matrice  $A \in \mathbb{R}^{m \times n}$ ,  $m \geq n$ , dimostrare che le colonne di  $A$  sono linearmente indipendenti se e solo se tutti gli elementi diagonale di  $R$  sono diversi da zero.

**4.16.** Verificare che quando la matrice  $A$  è simmetrica e  $U$  è ortogonale, la trasformazione di similitudine  $U^T A U$  lascia inalterata la simmetria.

**4.17.** Costruire un algoritmo che trasformi una generica matrice reale  $A$  di ordine  $n$  in una matrice simile con forma di Hessenberg.

**4.18.** Verificare che, operando come in 4.5.2, non è possibile ridurre una matrice  $A$  simmetrica alla forma (simile) diagonale.

**4.19.** Applicare il teorema di Gershgorin alla matrice  $A_4$  di pagina 107. Successivamente determinare, con il metodo descritto in 4.6, gli autovalori di  $A_4$ .

**4.20.** Nella discretizzazione di alcuni problemi agli autovalori descritti da equazione alle derivate parziali, perveniamo a problemi agli autovalori generalizzati del tipo

$$Ax = \lambda Bx$$

dove  $A \in \mathbb{R}^{n \times n}$  è simmetrica e  $B \in \mathbb{R}^{n \times n}$  è simmetrica definita positiva. Proporre un algoritmo che permetta di determinare gli autovalori e autovettori di tale problema.



# Capitolo 5

## Approssimazione di dati e di funzioni

### 5.1 Preliminari

In molti problemi matematici, e nella costruzione stessa di alcuni metodi numerici di base, emerge l'esigenza di dover approssimare una funzione  $f(x)$ , definita mediante una sua rappresentazione analitica oppure nota solo in alcuni punti  $\{x_i\}$ , ovvero soluzione di un problema matematico, con un'altra  $f_n(x)$  di forma più semplice su cui si possa facilmente operare (per esempio derivare, integrare). Esaminiamo brevemente due diverse situazioni.

Nella prima, dopo aver eseguito delle misurazioni  $\{y_i\}$ , corrispondenti a valori prefissati  $\{x_i\}$  della cosiddetta variabile indipendente, relative ad un determinato fenomeno (funzione) sottoposto al nostro esame, vogliamo costruire un “modello” matematico  $f_n(x)$  che descriveva “sufficientemente” bene il fenomeno in questione e ci permetta quindi di “fare previsioni” attendibili in punti  $x$  diversi dai nodi  $\{x_i\}$ .

Nella seconda invece, supponiamo di dover operare su di una funzione  $f(x)$  nota analiticamente; per esempio calcolare l'integrale

$$\int_a^b f(x) \, dx$$

Supponiamo però che l'espressione di  $f(x)$  sia tale da non permetterci di ottenere il valore incognito con i soli strumenti dell'Analisi Matematica. Conviene allora approssimare  $f(x)$ , nell'intervallo di interesse, con un'altra funzione  $f_n(x)$  di forma più semplice su cui sia poi possibile operare analiticamente e dedurre un'approssimazione, nei limiti di tolleranza richiesti, del risultato incognito.

Prima di affrontare un qualsiasi problema di approssimazione è indispensabile:

- (i) individuare la classe  $\mathbb{F}_n = \{f_n(x)\}$  delle funzioni approssimanti;
- (ii) fissata la classe  $\mathbb{F}_n$ , adottare un criterio per la scelta di un suo particolare elemento  $f_n(x)$ .

Elenchiamo dapprima le classi  $\mathbb{F}_n$  più usate.

1.  $\mathbb{P}_n = \{ f_n(x) = a_0 + a_1x + \cdots + a_nx^n \}$ , polinomi algebrici di grado  $n$ .

Ricordiamo che per ogni funzione  $f(x)$  continua in un intervallo chiuso e limitato  $[a, b]$ , e per ogni tolleranza  $\varepsilon > 0$ , il teorema di Weierstrass ci garantisce l'esistenza di un polinomio  $P_N(x)$  tale che

$$|f(x) - P_N(x)| < \varepsilon$$

per tutti gli  $x \in [a, b]$ .

Per individuare un elemento di  $\mathbb{P}_n$  è necessario fissare gli  $n + 1$  parametri  $a_0, a_1, \dots, a_n$ .

2.  $\mathbb{T}_n(\omega) = \{ f_n(x) = a_0 + \sum_{k=1}^n (a_k \cos k\omega x + b_k \sin k\omega x) \}$ , polinomi trigonometrici di grado  $n$  e frequenza  $\omega$ .

Anche in questo caso l'Analisi Matematica ci fornisce un teorema (analogo al precedente di Weierstrass) che ci assicura che per ogni funzione  $f(x)$  continua e periodica, con periodo  $2\pi/\omega$ , e per ogni tolleranza  $\varepsilon > 0$ , esiste un polinomio trigonometrico  $T_N(\omega; x)$  di frequenza circolare  $\omega$  tale che

$$|f(x) - T_N(\omega; x)| < \varepsilon$$

per tutti gli  $x \in \mathbb{R}$ .

Per individuare un elemento della classe  $\mathbb{T}_n(\omega)$  occorre fissare  $2n + 1$  parametri  $\{a_k, b_k\}$ .

3.  $\mathbb{S}_{n,d}([a, b]) = \{ f_n(x) = \text{funzione polinomiale a tratti di grado locale } d \text{ definita nell'intervallo limitato } [a, b] \}$ .

Il generico elemento  $f_n(x)$  di questa nuova classe è costituito da un'unione di tratti contigui di polinomi algebrici, ciascuno di grado (locale)  $d$ , definita su una suddivisione dell'intervallo  $[a, b]$  in  $n$  parti. Esso contiene  $n(d + 1)$  coefficienti.

La classe  $\mathbb{S}_{n,d}$  costituisce un'alternativa alle approssimazioni di tipo 1. e 2., soprattutto quando queste ultime, per raggiungere la precisione  $\varepsilon$  desiderata, richiedono l'uso di polinomi di grado troppo elevato o eccessivamente oscillanti.

4.  $\mathbb{R}_{n,d} = \{ f_n(x) = P_n(x)/P_d(x), \quad P_n(x) \in \mathbb{P}_n, \quad P_d(x) \in \mathbb{P}_d \}$ , funzioni razionali.

Mentre le classi precedenti possono rivelarsi valide per l'approssimazione di funzioni continue su intervalli chiusi e limitati, questa terza classe si dimostra utile quando occorre simulare singolarità (poli) o rappresentare fenomeni non periodici su intervalli infiniti.

I parametri essenziali presenti in  $f_n(x)$  sono  $n + d + 1$ ; infatti, poiché il rapporto  $P_n(x)/P_d(x)$  non cambia se dividiamo entrambi i polinomi per una costante, possiamo sempre normalizzare uno dei due polinomi, per esempio  $P_d(x)$ , in modo che il coefficiente di  $x^d$  sia 1.

5.  $\mathbb{E}_n = \{ f_n(x) = \sum_{k=1}^n a_k e^{-b_k x} \}$ , somme esponenziali di ordine  $n$ .

Questa classe viene proposta per l'approssimazione di quei fenomeni che presentano un comportamento di tipo esponenziale, male approssimabile con dei polinomi algebrici. I coefficienti  $\{b_k\}$  possono essere assegnati a priori, e in questo caso i parametri sono solo  $n$  e il generico elemento  $f_n(x)$  è lineare negli  $\{a_k\}$ ; oppure sono liberi tutti i  $2n$  parametri  $\{a_k, b_k\}$ .

Nelle pagine che seguono supporremo di avere a nostra disposizione, oppure di voler utilizzare, solo  $m+1$  dati numerici  $(x_i, y_i)$ ,  $i = 0, 1, \dots, m$ , del fenomeno o della funzione  $y = f(x)$  in esame. Dopo aver individuato la classe  $\mathbb{F}_n$  più appropriata per l'approssimazione del suddetto fenomeno o funzione, dovremo scegliere un elemento  $f_n(x) \in \mathbb{F}_n$ . Generalmente tale scelta viene fatta utilizzando uno dei seguenti tre criteri.

- a. *Interpolazione*. Come funzione  $f_n(x)$  prendiamo l'elemento (possibilmente unico) di  $\mathbb{F}_n$  che soddisfa le condizioni

$$f_n(x_i) = y_i, \quad i = 0, 1, \dots, m$$

Il numero dei parametri presenti in  $f_n(x)$  è generalmente uguale al numero dei punti  $\{x_i\}$ , cioè  $m+1$ .

Questo criterio ha senso soltanto quando i valori  $\{y_i\}$  sono accurati.

- b. *Minimi quadrati* (caso discreto). Come funzione  $f_n(x)$  scegliamo l'elemento di  $\mathbb{F}_n$  che minimizza la quantità

$$(5.1) \quad \sum_{i=0}^m [f_n(x_i) - y_i]^2$$

ovvero

$$(5.2) \quad \sum_{i=0}^m w_i [f_n(x_i) - y_i]^2, \quad w_i > 0$$

Nel secondo caso ai valori  $y_i$  più accurati o significativi associamo coefficienti  $w_i$  più grandi di quelli con cui invece “pesiamo” gli  $y_i$  meno precisi.

Quando si applica questo criterio il numero dei parametri presenti in  $f_n(x)$  è inferiore al numero dei punti  $\{x_i\}$ .

- c. *Minimax* (caso discreto). Con il criterio b. imponiamo la condizione di minimo alla somma dei quadrati dei singoli errori. Ciò non esclude la possibilità che per qualche indice  $i$  l'errore  $|f_n(x_i) - y_i|$  sia eccessivamente grande (rispetto alle esigenze del problema). Con il criterio minimax invece, scegliendo in  $\mathbb{F}_n$  l'elemento  $f_n(x)$  che minimizza la quantità

$$(5.3) \quad \max_{0 \leq i \leq m} |f_n(x_i) - y_i|$$

“controlliamo” proprio l’errore massimo.

Anche questo criterio viene applicato a modelli  $f_n(x)$  con un numero di parametri inferiore a  $m + 1$ .

Finora abbiamo usato il termine, un po’ vago, “classe di funzioni”. In realtà le funzioni reali  $f(x)$  che noi dovremo approssimare appartengono a *spazi lineari* (o *vettoriali*)  $\mathbb{F}$ ; ciò significa che se  $f, g \in \mathbb{F}$ , e  $\alpha, \beta \in \mathbb{R}$ , il nuovo elemento  $\alpha f + \beta g$  appartiene ancora a  $\mathbb{F}$ . Per esempio  $\mathbb{F} = C[a, b]$  è lo spazio lineare delle funzioni continue in  $[a, b]$ , e  $\mathbb{F} = C^k[a, b]$  è lo spazio lineare delle funzioni derivabili  $k$  volte in  $[a, b]$  con derivata  $k$ -esima continua.

Gli  $n + 1$  elementi di  $\mathbb{F}$ :  $\varphi_0(x), \varphi_1(x), \dots, \varphi_n(x)$  sono *linearmente indipendenti* quando

$$\sum_{i=0}^n c_i \varphi_i(x) \equiv 0$$

se e solo se  $c_0 = c_1 = \dots = c_n = 0$ . In questo caso, se tali elementi permettono di generare (con combinazioni lineari) tutto lo spazio  $\mathbb{F}$ , diciamo che essi formano una *base* di  $\mathbb{F}$  e che lo spazio  $\mathbb{F}$  ha *dimensione* finita  $n + 1$ . Gli spazi  $C[a, b]$  e  $C^k[a, b]$  hanno entrambi dimensione infinita, mentre  $\mathbb{P}_n$ , lo spazio dei polinomi algebrici di grado  $\leq n$ , ha dimensione finita ( $n + 1$ ) e una sua base è, per esempio,

$$\varphi_i(x) = x^i, \quad i = 0, 1, \dots, n$$

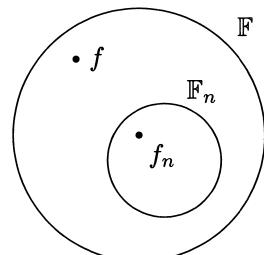
Nei paragrafi che seguono le funzioni approssimanti  $f_n(x)$  verranno prese in sottospazi lineari  $\mathbb{F}_n \subset \mathbb{F}$  di dimensione finita, così che, scelta una base dell’ $\mathbb{F}_n$  prescelto, ogni elemento di  $\mathbb{F}_n$  può essere espresso come combinazione lineare di tale base. Particolarmente importante, per motivi di stabilità numerica e di efficienza della determinazione dell’elemento  $f_n(x)$ , è la scelta di una base  $\{\varphi_i(x)\}$  “idonea”.

I problemi di cui ci occuperemo nei prossimi paragrafi possono pertanto essere formulati nel modo seguente: data una funzione  $f(x)$  appartenente allo spazio lineare di dimensione infinita  $\mathbb{F}$ , e scelto un sottospazio  $\mathbb{F}_n \subset \mathbb{F}$  di dimensione finita, determinare l’elemento (se unico)  $f_n(x)$  di  $\mathbb{F}_n$  che meglio approssima, secondo il criterio di scelta adottato, la funzione  $f(x)$ .

Determinata un’approssimazione  $f_n(x)$  della funzione  $f(x)$ , è importante poter misurare la bontà di  $f_n(x)$ , ovvero la “distanza” di quest’ultima da  $f(x)$ . A tale scopo occorre introdurre in  $\mathbb{F}$ , e quindi in  $\mathbb{F}_n$ , il concetto di *norma*, del tutto analogo a quello definito nel caso dei vettori in  $\mathbb{R}^n$ .

Un funzionale  $\|\cdot\| : \mathbb{F} \rightarrow \mathbb{R}$  è definito norma se:

1.  $\|f\| \geq 0$  per ogni  $f \in \mathbb{F}$
2.  $\|f\| = 0$  se e solo se  $f \equiv 0$



**Figura 5.1**

3.  $\|\alpha f\| = |\alpha| \|f\|$  per ogni  $f \in \mathbb{F}$  e  $\alpha \in \mathbb{R}$
4.  $\|f + g\| \leq \|f\| + \|g\|$  per ogni coppia  $f, g \in \mathbb{F}$

Essa è invece una *seminorma* quando viene a cadere la condizione 2, ossia quando esistono funzioni  $f \in \mathbb{F}$  non identicamente nulle per le quali  $\|f\| = 0$ .

Gli spazi  $\mathbb{F}$  che noi utilizzeremo in questo capitolo sono principalmente  $C[a, b]$  e  $C^k[a, b]$ ,  $-\infty < a < b < \infty$ , e in essi le norme più comuni sono certamente

$$\begin{aligned}\|f\|_\infty &= \max_{a \leq x \leq b} |f(x)| \\ \|f\|_1 &= \int_a^b |f(x)| dx \quad \text{oppure} \quad \|f\|_{1,w} = \int_a^b w(x)|f(x)| dx \\ \|f\|_2 &= \left( \int_a^b [f(x)]^2 dx \right)^{\frac{1}{2}} \quad \text{oppure} \quad \|f\|_{2,w} = \left( \int_a^b w(x)[f(x)]^2 dx \right)^{\frac{1}{2}}\end{aligned}$$

dove  $w(x)$  è una funzione non negativa in  $(a, b)$ , con al più un numero finito di zeri, e tale che  $\int_a^b w(x) dx < \infty$ . Osserviamo la validità delle diseguaglianze

$$(5.4) \quad \begin{aligned}\|f\|_{1,w} &\leq \|f\|_\infty \left[ \int_a^b w(x) dx \right] \\ \|f\|_{2,w} &\leq \|f\|_\infty \left[ \int_a^b w(x) dx \right]^{\frac{1}{2}}\end{aligned}$$

Le (5.3), (5.1) e (5.2)(<sup>†</sup>) rappresentano l'analogo discreto delle norme  $\|f\|_\infty$ ,  $\|f\|_2$  e  $\|f\|_{2,w}$ . Esse sono tuttavia solo delle seminorme.

Spesso, scelto lo spazio  $\mathbb{F}$  ed in esso una norma  $\|\cdot\|$ , è importante poter individuare una successione di sottospazi  $\mathbb{F}_n$ ,  $n = 1, 2, \dots$ , ciascuno di dimensione  $n + 1$ , che ci consenta di raggiungere qualsiasi tolleranza prefissata. È dunque necessario che, scelta in ogni  $\mathbb{F}_n$  la funzione  $f_n(x)$ ,

$$\lim_{n \rightarrow \infty} \|f - f_n\| = 0$$

Quando ciò si verifica, diciamo che la successione  $\{f_n(x)\}$  converge in norma a  $f(x)$ . Nel caso della *norma infinito* la convergenza viene denominata *uniforme*.

Le (5.4) ci permettono di affermare che la convergenza uniforme implica la convergenza sia in norma  $\|\cdot\|_{1,w}$  che in norma  $\|\cdot\|_{2,w}$ . Tuttavia, a differenza di quanto accade per le norme di vettore e di matrice (vedere il paragrafo 2.6), le norme  $\|\cdot\|_\infty$ ,  $\|\cdot\|_{1,w}$ ,  $\|\cdot\|_{2,w}$  non sono equivalenti tra di loro; per esempio la convergenza in norma  $\|\cdot\|_{1,w}$  non implica la convergenza in norma  $\|\cdot\|_\infty$  o  $\|\cdot\|_{2,w}$ .

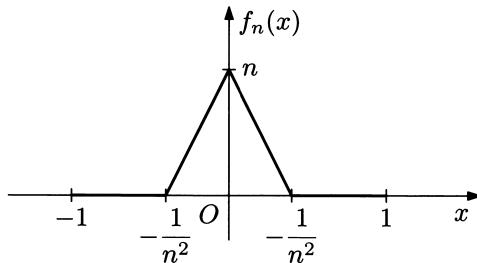
Il significato di questa nostra ultima asserzione può forse essere meglio compreso considerando la funzione identicamente nulla  $f \equiv 0$  in  $\mathbb{F} = C[-1, 1]$ , e prendendo  $f_n \in C[-1, 1]$

---

(<sup>†</sup>) O meglio, nel caso delle ultime due le radici quadrate di tali espressioni.

definita nel modo seguente:

$$f_n(x) = \begin{cases} n(1 + n^2x), & -\frac{1}{n^2} \leq x \leq 0 \\ n(1 - n^2x), & 0 < x \leq \frac{1}{n^2} \\ 0 & \text{altrimenti} \end{cases}$$



**Figura 5.2**

Con facili calcoli otteniamo

$$\begin{aligned}\|f - f_n\|_1 &= \frac{1}{n} \\ \|f - f_n\|_2 &= \sqrt{2/3} \\ \|f - f_n\|_\infty &= n\end{aligned}$$

e quindi

$$\begin{aligned}\lim_{n \rightarrow \infty} \|f - f_n\|_1 &= 0 \\ \lim_{n \rightarrow \infty} \|f - f_n\|_2 &= \sqrt{2/3} \\ \lim_{n \rightarrow \infty} \|f - f_n\|_\infty &= \infty\end{aligned}$$

## 5.2 Interpolazione polinomiale

### 5.2.1 Formula di interpolazione di Lagrange

In questo primo paragrafo esaminiamo il problema dell'approssimazione di dati numerici o di funzioni in un intervallo limitato  $[a, b]$ , mediante polinomi scelti con il criterio dell'interpolazione.

Siano dati  $n + 1$  punti  $(x_i, y_i)$ ,  $i = 0, 1, \dots, n$ ,  $y_i = f(x_i)$ , con  $x_i \neq x_j$  per  $i \neq j$ . Ci proponiamo di determinare il polinomio (eventualmente unico) di grado minimo che passa per i punti assegnati.

Poiché i punti sono  $n + 1$ , è sufficiente considerare il generico polinomio (di grado  $n$ ) che ha  $n + 1$  coefficienti

$$(5.5) \quad P_n(x) = a_0 + a_1x + \cdots + a_nx^n$$

ed imporre le condizioni

$$\begin{cases} a_0 + a_1x_0 + \cdots + a_nx_0^n = y_0 \\ a_0 + a_1x_1 + \cdots + a_nx_1^n = y_1 \\ \dots \\ a_0 + a_1x_n + \cdots + a_nx_n^n = y_n \end{cases}$$

I parametri incogniti  $a_0, a_1, \dots, a_n$  sono soluzione del seguente sistema quadrato di ordine  $n + 1$

$$(5.6) \quad \begin{pmatrix} 1 & x_0 & \dots & x_0^n \\ 1 & x_1 & \dots & x_1^n \\ \dots & \dots & \dots & \dots \\ 1 & x_n & \dots & x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_n \end{pmatrix}$$

La matrice del sistema è quella classica di Vandermonde, che risulta non singolare se e solo se  $x_i \neq x_j$  per  $i \neq j$ . Infatti, denotando con  $V$  tale matrice, è noto che

$$(5.7) \quad \det(V) = \prod_{i>j} (x_i - x_j) = \prod_{j=0}^{n-1} \left[ \prod_{i=j+1}^n (x_i - x_j) \right]$$

Possiamo quindi affermare:

**Teorema 5.1.** *Esiste uno ed un sol polinomio di grado  $n$  che assume valori  $y_i$ ,  $i = 0, 1, \dots, n$ , in corrispondenza di  $n + 1$  punti distinti  $x_i$ ,  $i = 0, 1, \dots, n$ .*

Osserviamo subito che il mal condizionamento dei sistemi con matrici di Vandermonde già denunciato al termine del paragrafo 3.1 non consiglia certo di procedere risolvendo (5.6). Inoltre, anche dal punto di vista del numero di operazioni aritmetiche necessarie, la determinazione della rappresentazione (5.5) mediante la soluzione del sistema (5.6) non appare conveniente. L'esame del sistema (5.6) ci è stato utile per stabilire l'esistenza e unicità del polinomio di interpolazione  $P_n(x)$  quando le  $x_i$  sono distinte; tuttavia, per l'effettiva costruzione di tale polinomio occorre cercare rappresentazioni alternative ben condizionate e, possibilmente, meno costose in termini di operazioni aritmetiche.

Pertanto, dopo aver provato l'esistenza e unicità di  $P_n(x)$ , esaminiamo il problema della sua rappresentazione. E a tale scopo, consideriamo preliminarmente, per ogni  $j = 0, 1, \dots, n$ , il polinomio di interpolazione  $l_j(x)$ , univocamente definito, associato al seguente insieme di dati:

$$(5.8) \quad (x_i, \delta_{ij}), \quad i = 0, 1, \dots, n$$

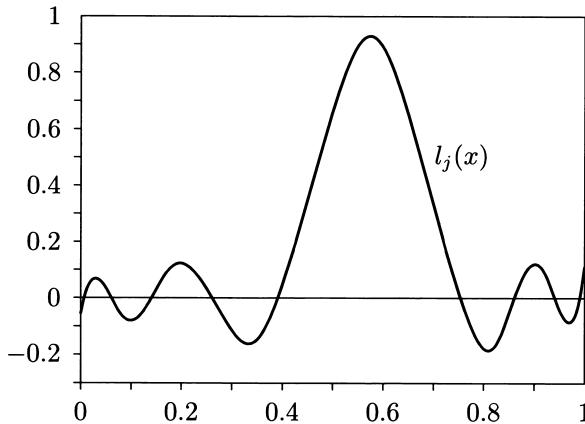


Figura 5.3

Tale polinomio deve quindi annullarsi in tutti i nodi  $x_i \neq x_j$ , ovvero contenere i fattori  $(x - x_i)$ ,  $i \neq j$ , ed assumere il valore 1 in  $x_j$ . Pertanto esso deve necessariamente ammettere la seguente rappresentazione:

$$(5.9) \quad l_j(x) = \frac{\prod_{\substack{i=0 \\ i \neq j}}^n (x - x_i)}{\prod_{\substack{i=0 \\ i \neq j}}^n (x_j - x_i)} = \frac{\omega_{n+1}(x)}{(x - x_j)\omega'_{n+1}(x_j)}$$

dove

$$\omega_{n+1}(x) = \prod_{i=0}^n (x - x_i) \quad \text{e} \quad \omega'_{n+1}(x_j) = \left. \frac{d\omega_{n+1}(x)}{dx} \right|_{x=x_j} = \prod_{\substack{i=0 \\ i \neq j}}^n (x_j - x_i)$$

La conoscenza dell'insieme di polinomi, ciascuno di grado  $n$ ,  $\{l_j(x), j = 0, 1, \dots, n\}$  associato ai nodi (distinti)  $\{x_i, i = 0, 1, \dots, n\}$ , ci consente di individuare immediatamente il polinomio di grado  $n$  che in corrispondenza delle ascisse  $x_i = 0, 1, \dots, n$ , assume valori assegnati  $y_i, i = 0, 1, \dots, n$ . Infatti non è difficile verificare la validità della rappresentazione

$$(5.10) \quad P_n(x) = \sum_{j=0}^n y_j l_j(x)$$

Il polinomio di interpolazione  $P_n(x)$  definito dal sistema (5.6) viene così espresso come combinazione lineare dei polinomi  $\{l_j(x)\}$ , e i coefficienti di tale combinazione sono proprio le ordinate  $\{y_j\}$ . La formula (5.10) è attribuita a Lagrange, e i polinomi base  $\{l_j(x)\}$  vengono denominati *polinomi fondamentali* di Lagrange associati ai nodi  $\{x_i\}$ .

Data una funzione continua  $f(x)$ , e costruito il polinomio di interpolazione  $P_n(x)$  di grado  $n$  associato ai punti  $(x_i, f(x_i))$ ,  $i = 0, 1, \dots, n$ , è possibile stimare l'errore  $f(x) - P_n(x)$  quando  $x \neq x_i$ ? Ovviamente il problema non può avere una risposta affermativa quando non vengono fornite altre informazioni sulla funzione  $f(x)$ . Infatti ad ogni  $f(x)$  che nei nodi  $\{x_i\}$  assume i valori  $\{y_i\}$  viene associato lo stesso polinomio di interpolazione. Quando invece la funzione  $f(x)$  è sufficientemente *regolare* possiamo conseguire qualche risultato.

Supponiamo che  $f(x)$  sia derivabile  $n+1$  volte in un intervallo  $[a, b]$  contenente tutti i nodi  $\{x_i\}$ , e osserviamo che l'errore

$$E_n(x) = f(x) - P_n(x)$$

è nullo quando  $x = x_i$ ,  $i = 0, 1, \dots, n$ . Ha quindi senso cercare in  $[a, b]$  una rappresentazione di  $E_n(x)$  del tipo

$$E_n(x) = \omega_{n+1}(x)R_n(x)$$

così che

$$f(x) = P_n(x) + \omega_{n+1}(x)R_n(x)$$

Successivamente definiamo una funzione ausiliaria

$$G(t) = f(t) - P_n(t) - \omega_{n+1}(t)R_n(x)$$

e supponiamo  $x \neq x_i$ . La funzione  $G(t)$  si annulla negli  $n+2$  punti  $t = x, x_0, x_1, \dots, x_n$ .

Il teorema di Rolle ci assicura l'esistenza di  $n+1$  punti distinti  $\xi_1^{(1)}, \xi_2^{(1)}, \dots, \xi_{n+1}^{(1)}$  di  $(a, b)$  nei quali  $G'(\xi_i^{(1)}) = 0$ . Lo stesso teorema, applicato questa volta a  $G'(t)$ , ci permette di affermare che esistono  $n$  punti distinti di  $(a, b)$ ,  $\xi_1^{(2)}, \dots, \xi_n^{(2)}$  nei quali  $G''(\xi_i^{(2)}) = 0$ . Applicando ripetutamente il teorema di Rolle alle funzioni  $G(t), G'(t), \dots, G^{(n)}(t)$  giungiamo alla conclusione che esiste un punto  $\xi = \xi(x) \in (a, b)$  in cui

$$G^{(n+1)}(\xi) = 0$$

così che

$$G^{(n+1)}(\xi) = f^{(n+1)}(\xi) - (n+1)! R_n(x) = 0$$

Quest'ultima espressione ci permette di affermare che nelle ipotesi fatte sulla  $f(x)$

$$(5.11) \quad R_n(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi)$$

dove  $\xi = \xi(x)$  è un punto (non noto) dell'intervallo  $(a, b)$ , e quindi

$$(5.12) \quad f(x) = \sum_{j=0}^n f(x_j)l_j(x) + \frac{\omega_{n+1}(x)}{(n+1)!} f^{(n+1)}(\xi)$$

La rappresentazione di  $E_n(x)$  che abbiamo costruito riveste un'importanza soprattutto teorica. Infatti, il suo utilizzo richiede non solo la determinazione di  $f^{(n+1)}(x)$ , ma anche la valutazione di quest'ultima in un punto  $\xi$  non noto.

Quando la funzione  $f(x)$  è un polinomio di grado non superiore a  $n$ , la derivata  $f^{(n+1)}(x)$  è identicamente nulla e dalla (5.12) deduciamo l'identità  $f(x) \equiv P_n(x)$ . Osserviamo tuttavia come questa proprietà poteva essere ottenuta senza ricorrere alla rappresentazione (5.12), ma semplicemente invocando l'unicità del polinomio di interpolazione.

In particolare nel caso  $f(x) = 1$  otteniamo l'interessante relazione

$$\sum_{j=0}^n l_j(x) \equiv 1$$

Una seconda conseguenza della rappresentazione (5.12) è la seguente. Scelti  $n + 1$  punti distinti  $\{x_i\}$  in un intervallo  $I = [\bar{x}, \bar{x} + h]$ , e costruito il corrispondente polinomio di interpolazione  $P_n(x)$ , supponiamo di voler usare  $P_n(x)$ , con  $n$  fissato, per approssimare una funzione  $f(x) \in C^{(n+1)}(I)$  nel solo intervallo  $I$ . L'espressione (5.11) ci consente di dedurre la seguente maggiorazione

$$(5.13) \quad \max_{x \in I} |f(x) - P_n(x)| < \frac{M_{n+1}}{(n+1)!} h^{n+1} (\dagger)$$

dove  $M_{n+1}$  rappresenta il massimo di  $f^{(n+1)}(x)$  in  $I$ . Tale maggiorazione, che descrive il comportamento dell'errore di interpolazione nell'intervallo  $I$  quale funzione dell'ampiezza  $h$ , giustifica l'interesse per la costruzione delle cosiddette *funzioni polinomiali a tratti* che introdurremo nel paragrafo 5.7.

Proseguiamo lo studio del problema di interpolazione polinomiale esaminando la reazione del valore numerico  $P_n(x)$  alla presenza di perturbazioni nelle ordinate  $f(x_j)$   $j = 0, 1, \dots, n$ . Introdotti i valori perturbati  $\bar{f}_j = f(x_j) + \varepsilon_j$ , definiamo

$$\bar{P}_n(x) = \sum_{j=0}^n \bar{f}_j l_j(x)$$

Per l'errore

$$P_n(x) - \bar{P}_n(x) = \sum_{j=0}^n l_j(x)[f(x_j) - \bar{f}_j]$$

abbiamo la maggiorazione

$$|P_n(x) - \bar{P}_n(x)| \leq \left( \sum_{j=0}^n |l_j(x)| \right) \max_{0 \leq j \leq n} |f(x_j) - \bar{f}_j|$$

Poiché

$$\|P_n(x)\|_\infty = \max_{x \in [a,b]} |P_n(x)| \geq \max_{0 \leq j \leq n} |P_n(x_j)|, \quad P_n(x_j) = f(x_j)$$

possiamo scrivere

$$\frac{\|P_n(x) - \bar{P}_n(x)\|_\infty}{\|P_n(x)\|_\infty} \leq \Lambda_n \frac{\|\varepsilon\|_\infty}{\|f\|_\infty}$$

( $\dagger$ ) Ovvero  $\max_{x \in I} |f(x) - P_n(x)| = O(h^{n+1})$ ,  $h \rightarrow 0$ .

dove

$$(5.14) \quad \Lambda_n = \left\| \sum_{j=0}^n |l_j(x)| \right\|_\infty$$

$$\varepsilon = (\varepsilon_0, \varepsilon_1, \dots, \varepsilon_n)^T \quad \text{e} \quad f = (f(x_0), f(x_1), \dots, f(x_n))^T$$

Il numero reale  $\Lambda_n$  viene chiamato *costante di Lebesgue* associata ai nodi  $x_0, x_1, \dots, x_n$  e all'intervallo  $[a, b]$ ; esso rappresenta una maggiorazione del coefficiente di amplificazione degli errori nei dati  $\{\varepsilon_j\}$  sul risultato  $P_n(x)$ . Appare quindi interessante esaminare il comportamento di  $\Lambda_n$  quando  $n \rightarrow \infty$ .

L'importanza della costante  $\Lambda_n$  viene ribadita dal risultato seguente:

**Teorema 5.2.** *Sia  $f(x) \in C[a, b]$ . Con  $P_n(x)$  denotiamo il polinomio di interpolazione associato ai nodi  $x_0, x_1, \dots, x_n$ . La seguente diseguaglianza risulta valida*

$$\|f(x) - P_n(x)\|_\infty \leq (1 + \Lambda_n) E_n(f)$$

dove<sup>(†)</sup>

$$E_n(f) = \min_{q \in \mathbb{P}_n} \|f(x) - q(x)\|_\infty$$

La dimostrazione di questo risultato non è difficile. Supposta nota l'esistenza del polinomio  $q(x)$  di grado  $n$  che minimizza la quantità  $\|f(x) - q(x)\|_\infty$ , è sufficiente osservare che

$$q(x) = \sum_{j=0}^n l_j(x) q(x_j)$$

e quindi

$$(5.15) \quad \begin{aligned} f(x) - P_n(x) &= [f(x) - q(x)] - [P_n(x) - q(x)] \\ &= [f(x) - q(x)] - \sum_{j=0}^n l_j(x) [f(x_j) - q(x_j)] \end{aligned}$$

Supponiamo  $n$  fissato, e scegliamo i punti  $\{x_i\}$  (distinti) in modo che la corrispondente costante  $\Lambda_n$  assuma il più piccolo valore possibile. Denotiamo quest'ultimo con  $\bar{\Lambda}_n$ . È stato dimostrato che

$$\bar{\Lambda}_n \sim \frac{2}{\pi} \log n, \quad n \rightarrow \infty$$

ovvero

$$\lim_{n \rightarrow \infty} \frac{\bar{\Lambda}_n}{\log n} = \frac{2}{\pi}$$

---

(†) Con  $\mathbb{P}_n$  denotiamo lo spazio lineare dei polinomi di grado  $\leq n$ .

La determinazione dei nodi di interpolazione associati alla costante  $\bar{\Lambda}_n$  risulta eccessivamente onerosa. Dal punto di vista del comportamento di  $\Lambda_n$ , le seguenti scelte, relative all'intervallo  $[-1, 1]$ , risultano pressoché ottimali:

$$\begin{aligned}x_i &= \frac{b-a}{2} \cos \frac{(2i+1)\pi}{2(n+1)} + \frac{b+a}{2}, & i = 0, 1, \dots, n \\x_i &= \frac{b-a}{2} \cos \frac{i\pi}{n} + \frac{b+a}{2}, & i = 0, 1, \dots, n\end{aligned}$$

Per entrambe abbiamo

$$\Lambda_n \sim \frac{2}{\pi} \log n \quad n \rightarrow \infty$$

Quando invece i nodi  $x_i$  sono scelti equidistanti

$$\Lambda_n \geq e^{\frac{n}{2}}$$

Consideriamo una successione infinita di insiemi di nodi distinti dell'intervallo  $[a, b]$ :

$$(5.16) \quad \begin{array}{ccccccc}x_0^{(0)} & & & & & & \\x_0^{(1)} & x_1^{(1)} & & & & & \\x_0^{(2)} & x_1^{(2)} & x_2^{(2)} & & & & \dots \\& & & & & & \dots \\x_0^{(n)} & x_1^{(n)} & x_2^{(n)} & \dots & x_n^{(n)} & & \dots\end{array}$$

Data una funzione  $f(x)$  definita in  $[a, b]$ , costruiamo la successione di polinomi di interpolazione di Lagrange  $\{P_n(x)\}$ , dove con  $P_n(x)$  denotiamo il polinomio di grado  $n$  associato ai nodi (distinti)  $x_0^{(n)}, x_1^{(n)}, \dots, x_n^{(n)}$ . La successione  $\{P_n(x)\}$  converge a  $f(x)$  in tutti i punti di  $[a, b]$ ? Una risposta, seppure parziale, ci viene data dal risultato seguente:

**Teorema 5.3.** ([2, pag. 156]). *Quando  $f(z)$  è una funzione analitica in un dominio  $D$  (del piano complesso) contenente  $[a, b]$ , e la distanza delle singolarità di  $f(z)$  dall'intervallo  $[a, b]$  è maggiore di  $(b-a)$ , abbiamo*

$$\lim_{n \rightarrow \infty} \|P_n(x) - f(x)\|_\infty = 0$$

La presenza di singolarità della funzione  $f(z)$  “troppo vicine” all'intervallo  $[a, b]$  può compromettere la convergenza della successione  $\{P_n(x)\}$ . Un esempio classico, dovuto a Runge, è costituito dalla funzione  $f(x) = 1/(1+x^2)$  interpolata su nodi equidistanti nell'intervallo  $[-5, 5]$  (vedere figura 5.4). In questo caso la predetta successione di polinomi di interpolazione non converge alla funzione  $f(x)$ . Se invece consideriamo lo stesso problema nell'intervallo  $[1, 2]$  (figura 5.5), per esempio, la convergenza è assicurata dal precedente teorema.

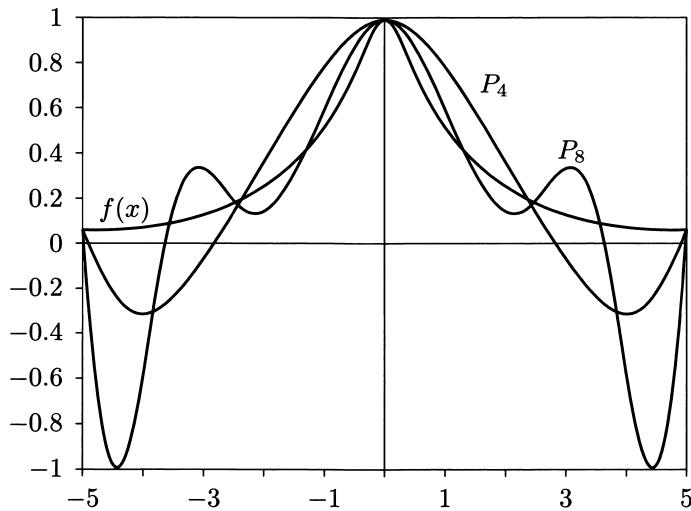


Figura 5.4

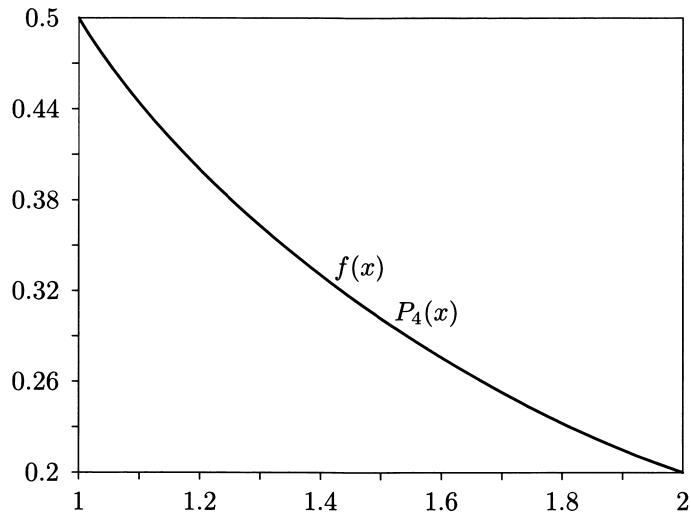


Figura 5.5

I limiti dell'interpolazione polinomiale quale metodo per l'approssimazione uniforme di funzioni continue su intervalli chiusi e limitati vengono messi in luce dal teorema che segue.

**Teorema 5.4.** (vedi [6, v. 1, pag 45]). *Data una qualunque successione di nodi distinti (5.16), tutti situati in  $[a, b]$ , esiste sempre una funzione  $f(x) \in C[a, b]$  che genera una successione di polinomi di interpolazione  $\{P_n(x)\}$  non convergente uniformemente a  $f(x)$  in  $[a, b]$ .*

Per esempio, il polinomio  $P_n(x)$  associato alla funzione  $f(x) = |x|$ ,  $x \in [-1, 1]$ , e ai nodi  $x_i = -1 + 2i/n$ ,  $i = 0, 1, \dots, n$ , quando  $n \rightarrow \infty$  converge a  $f(x)$  solamente nei punti  $x = -1, 0, 1$ .

Il teorema di Weierstrass ci assicura che per ogni funzione  $f(x) \in C[a, b]$  esiste sempre una successione di polinomi  $\{p_n(x)\}$  tale che

$$\lim_{n \rightarrow \infty} \|f(x) - p_n(x)\|_\infty = 0$$

Sfortunatamente questo risultato in generale non risulta valido quando  $p_n(x)$  viene definito da un processo di interpolazione.

### 5.2.2 Formula di interpolazione di Hermite

Con uno studio del tutto simile a quello che ha condotto alla formula di Lagrange (5.10) possiamo affrontare il seguente problema: scelti  $n+1$  punti distinti  $x_0, x_1, \dots, x_n$  in  $[a, b]$ , associare ad ogni funzione  $f(x)$  derivabile in  $[a, b]$  il polinomio (unico)  $P_{2n+1}(x)$ , di grado  $2n+1$ , tale che

$$\begin{aligned} P_{2n+1}(x_i) &= f(x_i) \\ P'_{2n+1}(x_i) &= f'(x_i) \end{aligned} \quad i = 0, 1, \dots, n$$

La seguente rappresentazione (vedere, per esempio, [1]) è attribuita ad Hermite:

$$(5.17) \quad P_{2n+1}(x) = \sum_{j=0}^n [1 - 2l'_j(x_j)(x - x_j)] l_j^2(x) f(x_j) + \sum_{j=0}^n (x - x_j) l_j^2(x) f'(x_j)$$

Inoltre, quando  $f(x) \in C^{2n+2}[a, b]$  abbiamo

$$f(x) - P_{2n+1}(x) = \frac{\omega_{n+1}^2(x)}{(2n+2)!} f^{(2n+2)}(\xi)$$

Al lettore proponiamo ora il seguente esercizio. Assegnati due punti distinti  $\alpha < \beta$ , e noti i valori che una funzione e la sua derivata prima assumono in  $\alpha$  e  $\beta$ , costruiamo il polinomio di Hermite  $P_3(x)$  definito dalle condizioni

$$\begin{cases} P_3(\alpha) = f(\alpha) \\ P'_3(\alpha) = f'(\alpha) \end{cases} \quad \begin{cases} P_3(\beta) = f(\beta) \\ P'_3(\beta) = f'(\beta) \end{cases}$$

Successivamente, consideriamo  $n+1$  punti distinti  $x_0 < x_1 < \dots < x_n$  e supponiamo di conoscere i corrispondenti valori  $f(x_i)$  e  $f'(x_i)$ ,  $i = 0, 1, \dots, n$ . Costruiamo la funzione

polinomiale a tratti<sup>(†)</sup>  $f_n(x)$  che in ogni intervallo  $[x_i, x_{i+1}]$  coincide con il corrispondente polinomio di Hermite  $P_{3,i}(x)$ ,

$$f_n(x) \equiv P_{3,i}(x), \quad x \in [x_i, x_{i+1}], \quad i = 0, 1, \dots, n-1$$

definito dalle condizioni

$$\begin{cases} P_{3,i}(x_i) = f(x_i) \\ P'_{3,i}(x_i) = f'(x_i) \end{cases} \quad \begin{cases} P_{3,i}(x_{i+1}) = f(x_{i+1}) \\ P'_{3,i}(x_{i+1}) = f'(x_{i+1}) \end{cases}$$

Avremo

$$\begin{aligned} f_n(x_i) &= f(x_i) & i = 0, 1, \dots, n \\ f'_n(x_i) &= f'(x_i) \end{aligned}$$

Quali altre proprietà possiede la funzione  $f_n(x)$ ? È continua, derivabile?

Il problema di Hermite può essere a sua volta generalizzato nella forma seguente: dati gli insiemi di valori

$$\begin{aligned} \{y_0^{(i)}\}_{i=0}^{\alpha_0-1} &\quad \text{in } x_0 \\ \{y_1^{(i)}\}_{i=0}^{\alpha_1-1} &\quad \text{in } x_1 \\ \cdots & \\ \{y_n^{(i)}\}_{i=0}^{\alpha_n-1} &\quad \text{in } x_n \end{aligned}$$

determinare il polinomio  $P_N(x)$ , di grado  $N = \alpha_0 + \alpha_1 + \dots + \alpha_n - 1$ , tale che

$$\begin{aligned} P_N^{(i)}(x_0) &= y_0^{(i)}, & i = 0, 1, \dots, \alpha_0 - 1 \\ P_N^{(i)}(x_1) &= y_1^{(i)}, & i = 0, 1, \dots, \alpha_1 - 1 \\ \cdots & \\ P_N^{(i)}(x_n) &= y_n^{(i)}, & i = 0, 1, \dots, \alpha_n - 1 \end{aligned}$$

Questo polinomio esiste ed è unico.

### 5.3 Formula di Newton alle differenze divise

Per costruire questa nuova rappresentazione del polinomio di interpolazione occorre preliminarmente introdurre delle nuove quantità denominate *differenze divise*.

Siano  $\{x_0, x_1, \dots, x_n\}$   $n+1$  punti o nodi assegnati, che inizialmente supponiamo distinti. Definiamo la differenza divisa di ordine 1 di  $f(x)$ :

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

---

<sup>(†)</sup> Vedi paragrafo 5.7.

Successivamente definiamo la differenza divisa di ordine 2:

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

e in generale quella di ordine  $n$ :

$$(5.18) \quad f[x_0, x_1, \dots, x_n] = \frac{f[x_1, x_2, \dots, x_n] - f[x_0, x_1, \dots, x_{n-1}]}{x_n - x_0}$$

Per induzione matematica non è difficile dimostrare che

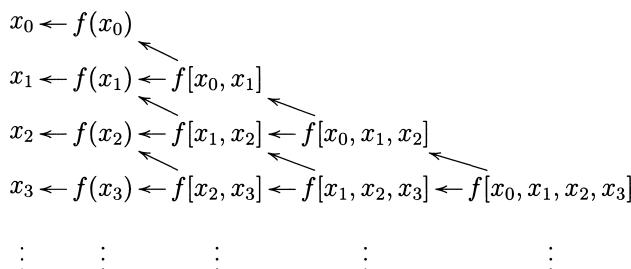
$$(5.19) \quad f[x_0, x_1, \dots, x_n] = \sum_{k=0}^n \frac{f(x_k)}{\prod_{\substack{j=0 \\ j \neq k}}^n (x_k - x_j)}$$

Quest'ultima espressione ci permette di affermare che  $f[x_0, x_1, \dots, x_n]$  è una funzione invariante a permutazioni dei suoi argomenti, cioè

$$f[x_0, x_1, \dots, x_n] = f[x_{i_0}, x_{i_1}, \dots, x_{i_n}]$$

dove  $(i_0, i_1, \dots, i_n)$  denota una qualsiasi permutazione di  $(0, 1, \dots, n)$ .

La costruzione delle differenze divise associate ai nodi  $x_0, x_1, x_2, \dots$  può essere schematizzata nel modo seguente:



dove, per esempio,

$$\begin{aligned} f[x_1, x_2, x_3] &= \frac{f[x_2, x_3] - f[x_1, x_2]}{x_3 - x_1} \\ f[x_0, x_1, x_2, x_3] &= \frac{f[x_1, x_2, x_3] - f[x_0, x_1, x_2]}{x_3 - x_0} \end{aligned}$$

Quando due argomenti risultano coincidenti possiamo ugualmente dare un significato alla corrispondente differenza divisa di ordine 1, purché  $f'(x)$  esista in quel punto:

$$f[x_0, x_0] = \lim_{x \rightarrow x_0} f[x_0, x] = \lim_{x \rightarrow x_0} \frac{f(x) - f(x_0)}{x - x_0} = f'(x_0)$$

Più in generale, definiamo

$$(5.20) \quad f[\underbrace{x_0, x_0, \dots, x_0}_{k+1}] = \frac{f^{(k)}(x_0)}{k!}$$

Il seguente risultato, la cui dimostrazione è reperibile in [1, pag. 252], stabilisce un utile rapporto tra le differenze divise e le derivate di  $f(x)$ .

**Teorema 5.5.** *Se  $f(x) \in C^k[a, b]$  e i nodi  $\{x_i\}$ , non necessariamente distinti, appartengono tutti all'intervallo  $[a, b]$ , allora esiste un punto  $\xi$ , con  $\min\{x_i\} \leq \xi \leq \max\{x_i\}$ , tale che*

$$f[x_0, x_1, \dots, x_k] = \frac{f^{(k)}(\xi)}{k!}$$

In particolare

$$f[\underbrace{x, x, \dots, x}_{k+1}] = \frac{f^{(k)}(x)}{k!}$$

Conseguenza immediata di questo teorema è che

$$(5.21) \quad f[x_0, x_1, \dots, x_k] = 0$$

ogniqualvolta  $f(x)$  è un polinomio di grado  $N$  e  $k > N$ .

Riprendiamo la definizione di differenza divisa e scriviamo

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

donde

$$(5.22) \quad f(x) = f(x_0) + (x - x_0)f[x, x_0]$$

Il termine  $P_0(x) = f(x_0)$  rappresenta il polinomio di grado 0 interpolante la funzione  $f(x)$  nel nodo  $x = x_0$ , mentre  $(x - x_0)f[x, x_0]$  denota l'errore  $f(x) - P_0(x)$ . Successivamente consideriamo

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

ovvero

$$f[x, x_0] = f[x_0, x_1] + (x - x_1)f[x, x_0, x_1]$$

e sostituiamo quest'espressione in (5.22); troviamo

$$(5.23) \quad f(x) = f(x_0) + (x - x_0)f[x_0, x_1] + (x - x_0)(x - x_1)f[x, x_0, x_1]$$

Il polinomio  $P_1(x) = f(x_0) + (x - x_0)f[x_0, x_1]$ , di grado 1, interpola la funzione  $f(x)$  nei nodi  $x_0$  e  $x_1$  e rappresenta pertanto il corrispondente polinomio di interpolazione,

con  $f(x) - P_1(x) = (x - x_0)(x - x_1)f[x, x_0, x_1]$ . Per avere l'espressione del polinomio di interpolazione  $P_2(x)$  sui nodi  $x_0, x_1, x_2$  è sufficiente considerare la differenza divisa

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

da cui otteniamo

$$f[x, x_0, x_1] = f[x_0, x_1, x_2] + (x - x_2)f[x, x_0, x_1, x_2]$$

e quindi, inserendo quest'ultima in (5.23),

$$\begin{aligned} f(x) &= \underbrace{f(x_0) + (x - x_0)f[x_0, x_1] + (x - x_0)(x - x_1)f[x_0, x_1, x_2]}_{P_2(x)} \\ &\quad + (x - x_0)(x - x_1)(x - x_2)f[x, x_0, x_1, x_2] \end{aligned}$$

Così proseguendo, sino ad includere tutti i nodi  $x_0, x_1, \dots, x_n$ , perveniamo alla rappresentazione

$$\begin{aligned} f(x) &= f(x_0) + (x - x_0)f[x_0, x_1] + (x - x_0)(x - x_1)f[x_0, x_1, x_2] + \dots \\ &\quad + (x - x_0)(x - x_1) \dots (x - x_{n-1})f[x_0, x_1, \dots, x_n] \\ &\quad + (x - x_0)(x - x_1) \dots (x - x_n)f[x, x_0, x_1, \dots, x_n] \end{aligned}$$

ossia

$$f(x) = P_n(x) + E_n(x)$$

con

$$\begin{aligned} (5.24) \quad P_n(x) &= f(x_0) + (x - x_0)f[x_0, x_1] + \dots \\ &\quad + (x - x_0)(x - x_1) \dots (x - x_{n-1})f[x_0, x_1, \dots, x_n] \\ E_n(x) &= (x - x_0)(x - x_1) \dots (x - x_n)f[x, x_0, x_1, \dots, x_n] \end{aligned}$$

Questa è la *formula di interpolazione di Newton* alle differenze divise. Il polinomio  $P_n(x)$ , di grado  $n$ , così trovato soddisfa le condizioni di interpolazione  $P_n(x_i) = f(x_i)$ ,  $i = 0, 1, \dots, n$ . Per verificare la correttezza di questa affermazione è sufficiente applicare la formula di Newton al polinomio di interpolazione di Lagrange associato agli stessi dati  $\{x_i, f(x_i)\}$ : per la proprietà (5.21) risulta  $E_n(x) \equiv 0$ .

Abbiamo così ottenuto una seconda rappresentazione del polinomio di interpolazione, alternativa a quella di Lagrange. Osserviamo subito che la formula di Newton è particolarmente interessante dal punto di vista computazionale: la sua costruzione richiede solo  $n^2/2$  divisioni e  $n^2$  sottrazioni; inoltre il polinomio  $P_{n+1}(x)$  costruito sui nodi  $x_0, x_1, \dots, x_n$  e su un nuovo punto  $x_{n+1}$  è legato a  $P_n(x)$  dalla semplice relazione

$$P_{n+1}(x) = P_n(x) + (x - x_0)(x - x_1) \dots (x - x_n)f[x_0, x_1, \dots, x_{n+1}]$$

Osserviamo infine che i coefficienti  $f[x_0, x_1, \dots, x_k]$  sono indipendenti da  $x$ .

Anche il polinomio di interpolazione di Hermite (5.17) può essere espresso nella forma di Newton. Per provare quest'ultima affermazione è sufficiente ricordare che  $f[x_i, x_i] = f'(x_i)$  e costruire il seguente schema alle differenze divise:

$$\begin{array}{ll} x_0 & f(x_0) \\ x_0 & f(x_0) \quad f'(x_0) \\ x_1 & f(x_1) \quad f[x_0, x_1] \\ x_1 & f(x_1) \quad f'(x_1) \\ \vdots & \vdots \quad \vdots \\ x_n & f(x_n) \quad f[x_{n-1}, x_n] \\ x_n & f(x_n) \quad f'(x_n) \end{array}$$

Analogamente si procede nel caso dell'interpolazione di Hermite generalizzata; nella tabella delle differenze divise ogni coppia  $\{x_i, f(x_i)\}$  viene ripetuta (consecutivamente) tante volte quante sono le condizioni imposte su  $x_i$ :

$$\begin{aligned} \alpha_0 & \left\{ \begin{array}{ll} x_0 & f(x_0) \\ \vdots & \vdots \\ x_0 & f(x_0) \end{array} \right. \\ \alpha_1 & \left\{ \begin{array}{ll} x_1 & f(x_1) \\ \vdots & \vdots \\ x_1 & f(x_1) \end{array} \right. \\ & \vdots \quad \vdots \\ \alpha_n & \left\{ \begin{array}{ll} x_n & f(x_n) \\ \vdots & \vdots \\ x_n & f(x_n) \end{array} \right. \end{aligned}$$

e le differenze divise del tipo  $f[\overbrace{x_i, x_i, \dots, x_i}^{k+1}]$  vengono sostituite dalle quantità  $f^{(k)}(x_i)/k!$ .

► **Esempio 5.1.** Costruire il polinomio di grado  $n = 4$  passante per i punti  $(-4, 1245)$ ,  $(-1, 33)$ ,  $(0, 5)$ ,  $(2, 9)$ ,  $(5, 1335)$ .

Costruiamo dapprima la tabella delle differenze divise:

$x_i$	$f(x_i)$	$f[x_i, x_{i+1}]$	$f[x_i, x_{i+1}, x_{i+2}]$	$\dots$
-4	1245			
-1	33	-404		
0	5	-28	94	
2	9	2	10	-14
5	1335	442	88	13 3

e quindi la rappresentazione di Newton del polinomio di interpolazione:

$$\begin{aligned} P_4(x) = & 1245 - 404(x + 4) + 94(x + 4)(x + 1) - 14(x + 4)(x + 1)x \\ & + 3(x + 4)(x + 1)x(x - 2) = 3x^4 - 5x^3 + 6x^2 - 14x + 5 \end{aligned}$$

► **Esempio 5.2.** Trovare il polinomio di grado 3 passante per il punto  $(0, 10)$  con coefficiente angolare della retta tangente uguale a 1, e per i punti  $(1, 15)$  e  $(2, 5)$ .

Tabella differenze divise:

$x_i$	$f(x_i)$			
0	10			
0	10	1		
1	15	5	4	
2	5	-10	$-\frac{15}{2}$	$-\frac{23}{4}$

Rappresentazione di Newton del polinomio di interpolazione:

$$P_3(x) = 10 + x + 4x^2 - \frac{23}{4}x^2(x - 1)$$

Concludiamo le nostre osservazioni rilevando che la formula di Newton, nella sua formulazione più generale, costituisce una generalizzazione della ben nota formula di Taylor per lo sviluppo di una funzione nell'intorno di un punto. Infatti per ottenere quest'ultima in un punto  $\bar{x}_0$  è sufficiente prendere nella formula di Newton tutti i punti  $\{x_i\}$  coincidenti con  $\bar{x}_0$ .

Presentiamo infine due algoritmi: il primo (Difdiv) costruisce la tabella (triangolare) delle differenze divise associata ad un insieme di punti  $\{x_i, f(x_i)\}$  con ascisse  $x_i$  distinte, mentre il secondo (Interp) utilizza i risultati prodotti dal primo per valutare il polinomio di interpolazione di Newton in un punto assegnato.

---

**Algoritmo 7:** Difdiv( $n, x, f$ )

---

*Commento.* I vettori  $x$  e  $f$ , di lunghezza  $n + 1$ , inizialmente contengono i dati  $x_i$  e  $f(x_i)$ ,  $i = 0, 1, \dots, n$ . Le colonne della tabella alle differenze divise vengono successivamente determinate e memorizzate nel vettore  $f$ . Alla fine il vettore  $f$  contiene gli elementi della diagonale della tabella.

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

- 1: **ciclo 1:**  $i = 1, \dots, n$
  - 2:   **ciclo 2:**  $j = n, \dots, i$
  - 3:      $f_j \leftarrow (f_j - f_{j-1})/(x_j - x_{j-i})$
  - 4:   **fine ciclo 2**
  - 5: **fine ciclo 1**
  - 6: **esci**
- 

---

**Algoritmo 8:** Interp( $n, x, f, t, p$ )

---

*Commento.* I vettori  $x$  e  $f$ , di dimensione  $n + 1$ , inizialmente contengono i dati  $\{x_i\}$  e gli elementi diagonale della tabella delle differenze divise (l'output di Difdiv).

Assegnato un valore  $t$ , l'algoritmo determina il valore che il polinomio di Newton assume in  $t$  e lo memorizza in  $p$ .

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

- 1:  $p \leftarrow f_n$
  - 2: **ciclo 1:**  $i = n - 1, \dots, 0$
  - 3:    $p \leftarrow p(t - x_i) + f_i$
  - 4: **fine ciclo 1**
  - 5: **esci**
- 

Come abbiamo già osservato in precedenza, la costruzione della tabella delle differenze divise associata ai dati  $(x_i, f(x_i))$ ,  $i = 0, 1, \dots, n$ , richiede  $n^2$  somme o sottrazioni e  $n^2/2$  divisioni. Inoltre, per ogni ascissa  $\bar{x}$ , la valutazione di  $P_n(\bar{x})$  mediante l'algoritmo 8, che scaturisce dalla seguente interpretazione della (5.24)

$$P_n(x) = ((\dots((f[x_0, x_1, \dots, x_n](x - x_{n-1}) + f[x_0, x_1, \dots, x_{n-1}])(x - x_{n-2}) \\ + f[x_0, x_1, \dots, x_{n-2}])(x - x_{n-3}) + \dots)(x - x_1) + f[x_0, x_1])(x - x_0) + f(x_0)$$

richiede solamente  $n$  moltiplicazioni e  $2n$  somme.

## 5.4 Formule di Newton alle differenze finite

In questo paragrafo consideriamo il problema dell'interpolazione polinomiale nel caso in cui i punti  $\{x_i\}$  siano distinti ed equidistanti, ossia  $x_{i+1} = x_i + h, h > 0$ . In particolare diamo una rappresentazione del polinomio di interpolazione che evidenzia l'equidistanza dei nodi  $x_i$ . A tale scopo è necessario introdurre preliminarmente alcune nozioni.

Definiamo dapprima gli operatori lineari  $E, \Delta, \nabla$  associati al passo  $h$ :

$$\begin{aligned} E f(x) &= f(x + h) \\ \Delta f(x) &= f(x + h) - f(x) \\ \nabla f(x) &= f(x) - f(x - h) \end{aligned}$$

Se gli operatori suddetti vengono associati ai nodi  $x_{i+1} = x_i + h$ , posto  $f_i = f(x_i)$  le definizioni precedenti assumono il nuovo aspetto:

$$\begin{aligned} E f_i &= f_{i+1} \\ \Delta f_i &= f_{i+1} - f_i \\ \nabla f_i &= f_i - f_{i-1} \end{aligned}$$

Dalle definizioni degli operatori  $E, \Delta$  e  $\nabla$  seguono immediatamente le identità:

$$\begin{aligned} E &= I + \Delta \\ \Delta &= E \nabla \end{aligned}$$

dove nella prima  $I$  denota l'operatore identità.

Successivamente definiamo

$$\begin{array}{ll} E^0 f_i = f_i, & E^n f_i = E(E^{n-1} f_i) = E^{n-1} f_{i+1} \\ \Delta^0 f_i = f_i, & \Delta^n f_i = \Delta(\Delta^{n-1} f_i) = \Delta^{n-1} f_{i+1} - \Delta^{n-1} f_i \\ \nabla^0 f_i = f_i, & \nabla^n f_i = \nabla(\nabla^{n-1} f_i) = \nabla^{n-1} f_i - \nabla^{n-1} f_{i-1} \end{array}$$

Chiameremo  $\Delta^n f_i$  differenza finita progressiva (o in avanti) di ordine  $n$  in  $x_i$  e  $\nabla^n f_i$  differenza finita regressiva (o all'indietro) di ordine  $n$  in  $x_i$ .

Un esame più attento delle definizioni suddette ci consente di osservare che le differenze  $\nabla^k f_i$ , per esempio, possono venire espresse tramite le  $\Delta^k f_i$ ; risulta infatti:

$$(5.25) \quad \nabla^k f_i = \Delta^k f_{i-k}$$

All'insieme di dati  $\{f_i\}$  possiamo associare, per esempio, la seguente tabella triangolare alle differenze finite progressive:

$$\begin{array}{ccccccc} & & & \vdots & & & \\ & & f_0 & \vdots & & & \\ & f_1 & \Delta f_0 & \vdots & & & \\ f_2 & \Delta f_1 & \Delta^2 f_0 & \vdots & & & \\ f_3 & \Delta f_2 & \Delta^2 f_1 & \Delta^3 f_0 & \vdots & & \\ f_4 & \Delta f_3 & \Delta^2 f_2 & \Delta^3 f_1 & \Delta^4 f_0 & \vdots & \\ f_5 & \Delta f_4 & \Delta^2 f_3 & \Delta^3 f_2 & \Delta^4 f_1 & \Delta^5 f_0 & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{array}$$

Il teorema che ora presentiamo (e dimostriamo) stabilisce un'utile relazione tra le differenze divise introdotte nel paragrafo precedente e le differenze progressive  $\Delta^k f_0$ .

**Teorema 5.6.** *Per  $k \geq 0$  abbiamo*

$$(5.26) \quad f[x_0, x_1, \dots, x_k] = \frac{1}{k! h^k} \Delta^k f_0$$

*Dimostrazione.* Nel caso  $k = 0$  la verifica della validità della (5.26) è immediata. Supponiamo che l'identità (5.26) sia vera per tutti gli interi  $k \leq r$ ; per  $k = r + 1$  richiamando la definizione di differenza divisa possiamo scrivere

$$f[x_0, x_1, \dots, x_{r+1}] = \frac{f[x_1, \dots, x_{r+1}] - f[x_0, \dots, x_r]}{x_{r+1} - x_0}$$

Ma l'ipotesi fatta ci consente di scrivere

$$f[x_0, x_1, \dots, x_{r+1}] = \frac{1}{(r+1)h} \left[ \frac{1}{r! h^r} \Delta^r f_1 - \frac{1}{r! h^r} \Delta^r f_0 \right] = \frac{1}{(r+1)! h^{r+1}} \Delta^{r+1} f_0$$

Invocando infine il principio di induzione matematica possiamo affermare la validità della (5.26) per tutti gli interi  $k \geq 0$ .  $\square$

Per ottenere una rappresentazione del polinomio di interpolazione che evidenzi l'equidistanza dei nodi (e quindi il passo  $h$ ) è sufficiente considerare la formula di Newton e sostituire le differenze divise ivi presenti con le corrispondenti differenze finite definite dalla (5.26). Introducendo inoltre la nuova variabile  $t = \frac{1}{h}(x - x_0)$ , così che

$$(5.27) \quad (x - x_0)(x - x_1) \dots (x - x_k) = t(t - 1) \dots (t - k)h^{k+1}$$

otteniamo

$$P_n(x_0 + th) = f_0 + th \frac{\Delta f_0}{h} + t(t-1)h^2 \frac{\Delta^2 f_0}{2! h^2} + \cdots + t(t-1)\dots(t-n+1)h^n \frac{\Delta^n f_0}{n! h^n}$$

Definendo poi i coefficienti binomiali generalizzati

$$\binom{t}{0} = 1, \quad \binom{t}{k} = \frac{t(t-1)\dots(t-k+1)}{k!}, \quad k > 0$$

al polinomio  $P_n(x)$  possiamo dare la forma più compatta

$$(5.28) \quad P_n(x_0 + th) = \sum_{k=0}^n \binom{t}{k} \Delta^k f_0$$

Questa è la *formula di interpolazione di Newton alle differenze progressive*.

Le differenze divise possono anche venire espresse mediante le differenze finite regressive  $\nabla^k f_i$ . Infatti, osservando che  $\Delta^k f_0 = \nabla^k f_k$ , l'identità (5.26) in questo caso assume la forma

$$f[x_0, x_1, \dots, x_k] = \frac{1}{k! h^k} \nabla^k f_k$$

Conviene poi ordinare i nodi di interpolazione nel modo seguente

$$x_{-n}, \dots, x_{-1}, x_0, \quad x_{-k} = x_0 - kh$$

così che

$$f[x_0, x_{-1}, \dots, x_{-k}] = f[x_{-k}, \dots, x_{-1}, x_0] = \frac{1}{k! h^k} \nabla^k f_0$$

Con l'aiuto di queste ultime relazioni non è difficile ottenere la *formula di Newton alle differenze regressive*

$$P_n(x_0 - uh) = \sum_{k=0}^n \binom{-u+k-1}{k} \nabla^k f_0$$

Ricordiamo infine due risultati (le cui dimostrazioni lasciamo al lettore come esercizio) sulle differenze  $\Delta^k f_i$ .

**Teorema 5.7.** Se  $f(x) \in C^r[a, b]$ , allora per ogni intero  $k \leq r$  abbiamo

$$\Delta^k f_i = h^k f^{(k)}(\xi_{ki})$$

dove  $\xi_{ki}$  è un punto, non noto, dell'intervallo  $(x_i, x_{i+k})$ .

**Corollario 5.1.** Quando  $f(x) = a_0 x^m + a_1 x^{m-1} + \cdots + a_m$  è un polinomio di grado  $m$  tutte le differenze finite di ordine  $m$ ,  $\Delta^m f_i$ , sono uguali e valgono

$$\Delta^m f_i = m! h^m a_0$$

mentre le differenze di ordine superiore sono tutte nulle.

Prima di concludere il paragrafo esaminiamo la propagazione di eventuali perturbazioni  $\varepsilon_i$ ,  $|\varepsilon_i| \leq \varepsilon$ , presenti nei dati  $f_i$ , sulle differenze  $\Delta^k f_i$ . La definizione di  $\Delta^k$  ci permette di scrivere

$$\Delta^k f_i - \Delta^k(f_i + \varepsilon_i) = -\Delta^k \varepsilon_i = -\sum_{j=0}^k \binom{k}{j} (-1)^{k-j} \varepsilon_{i+j}$$

e quindi di dedurre

$$|\Delta^k \varepsilon_i| \leq \varepsilon \sum_{j=0}^k \binom{k}{j} = 2^k \varepsilon$$

Come fattore di amplificazione degli errori  $\varepsilon_i$  possiamo pertanto prendere la quantità  $2^k$ .

## 5.5 Interpolazione trigonometrica

Consideriamo il generico polinomio trigonometrico di grado  $n$  e periodo  $2\pi$

$$T_n(x) = \frac{a_0}{2} + \sum_{j=1}^n [a_j \cos jx + b_j \sin jx]$$

L'esame di polinomi di periodo  $2\pi$  non toglie nulla in generalità ai risultati che otterremo. Infatti, nel caso di una funzione  $f(u)$  periodica con periodo  $\tau$ , cioè  $f(u + \tau) = f(u)$  per ogni  $u$ , è sufficiente considerare

$$g(x) = f\left(\frac{\tau x}{2\pi}\right)$$

funzione periodica con periodo  $2\pi$ ; se  $T_n(x)$  è il polinomio trigonometrico scelto per approssimare  $g(x)$ , il corrispondente polinomio  $t_n(u)$  associato a  $f(u)$  avrà la seguente rappresentazione:

$$t_n(u) = T_n\left(\frac{2\pi u}{\tau}\right)$$

La non singolarità delle matrici

$$\begin{pmatrix} 1 & \cos x_0 & \sin x_0 & \cos 2x_0 & \sin 2x_0 & \dots & \cos nx_0 & \sin nx_0 \\ 1 & \cos x_1 & \sin x_1 & \cos 2x_1 & \sin 2x_1 & \dots & \cos nx_1 & \sin nx_1 \\ \dots & \dots \\ 1 & \cos x_{2n} & \sin x_{2n} & \cos 2x_{2n} & \sin 2x_{2n} & \dots & \cos nx_{2n} & \sin nx_{2n} \end{pmatrix}$$

e

$$\begin{pmatrix} 1 & \cos x_0 & \sin x_0 & \dots & \cos(n-1)x_0 & \sin(n-1)x_0 & \cos nx_0 \\ 1 & \cos x_1 & \sin x_1 & \dots & \cos(n-1)x_1 & \sin(n-1)x_1 & \cos nx_1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & \cos x_{2n-1} & \sin x_{2n-1} & \dots & \cos(n-1)x_{2n-1} & \sin(n-1)x_{2n-1} & \cos nx_{2n-1} \end{pmatrix}$$

per  $0 \leq x_0 < x_1 < \dots < x_{2n-1} < x_{2n} < 2\pi$ ,

$$\begin{pmatrix} 1 & \cos x_0 & \cos 2x_0 & \dots & \cos nx_0 \\ 1 & \cos x_1 & \cos 2x_1 & \dots & \cos nx_1 \\ \dots & \dots & \dots & \dots & \dots \\ 1 & \cos x_n & \cos 2x_n & \dots & \cos nx_n \end{pmatrix}$$

per  $0 \leq x_0 < x_1 < \dots < x_n \leq \pi$ , e

$$\begin{pmatrix} \sin x_0 & \sin 2x_0 & \dots & \sin nx_0 \\ \sin x_1 & \sin 2x_1 & \dots & \sin nx_1 \\ \dots & \dots & \dots & \dots \\ \sin x_{n-1} & \sin 2x_{n-1} & \dots & \sin nx_{n-1} \end{pmatrix}$$

per  $0 < x_0 < x_1 < \dots < x_{n-1} < \pi$ , garantisce l'esistenza e unicità di polinomi di interpolazione di grado  $n$  del tipo

$$(5.29) \quad \begin{aligned} T_n(x) &= \frac{a_0}{2} + \sum_{j=1}^n [a_j \cos jx + b_j \sin jx] \\ T_n(x) &= \frac{a_0}{2} + \sum_{j=1}^{n-1} [a_j \cos jx + b_j \sin jx] + \frac{a_n}{2} \cos nx \\ T_n(x) &= \frac{a_0}{2} + \sum_{j=1}^n a_j \cos jx \\ T_n(x) &= \sum_{j=1}^n b_j \sin jx \end{aligned}$$

Osserviamo inoltre che le formule di Eulero

$$\cos x = \frac{e^{ix} + e^{-ix}}{2}, \quad \sin x = \frac{e^{ix} - e^{-ix}}{2i}$$

ci consentono di trasformare l'interpolazione trigonometrica in un problema di interpolazione (nel campo complesso) con polinomi algebrici.

Nel caso, per esempio, del polinomio

$$T_n(x) = \frac{a_0}{2} + \sum_{j=1}^n [a_j \cos jx + b_j \sin jx]$$

otteniamo

$$T_n(x) = \sum_{j=-n}^n c_j e^{ijx}$$

con

$$c_0 = \frac{a_0}{2}, \quad c_j = \frac{1}{2}(a_j - ib_j) \quad \text{e} \quad c_{-j} = \frac{1}{2}(a_j + ib_j) \quad j = 1, \dots, n$$

La determinazione dei coefficienti  $\{a_j\}, \{b_j\}$  tali che  $T_n(x_k) = f(x_k)$ ,  $k = 0, 1, \dots, 2n$ , viene ricondotta alla risoluzione del seguente problema di interpolazione: determinare i coefficienti  $c_j$ ,  $j = -n, \dots, 0, \dots, n$ , del sistema

$$\sum_{l=0}^{2n} c_{l-n} z_k^l = z_k^n f(x_k), \quad z_k = e^{ix_k}, \quad k = 0, 1, \dots, 2n$$

Ovviamente i coefficienti  $a_j = c_j + c_{-j}$  e  $b_j = i(c_j - c_{-j})$  sono reali se e solo se  $c_{-j} = \bar{c}_j$ ; in questo caso

$$a_j = 2 \operatorname{Re}(c_j) \quad \text{e} \quad b_j = -2 \operatorname{Im}(c_j)$$

Pertanto, quando i dati  $\{f(x_k)\}$  sono reali, per individuare i coefficienti  $\{a_j\}$  e  $\{b_j\}$  del polinomio trigonometrico  $T_n(x)$  è sufficiente conoscere  $c_j$ ,  $j = 0, 1, \dots, n$ .

La situazione più interessante, dal punto di vista della determinazione dei coefficienti del polinomio trigonometrico, si ha certamente quando i punti  $\{x_k\}$  sono equidistanti:

$$x_k = k \frac{2\pi}{N}, \quad k = 0, 1, \dots, N-1$$

Infatti, ricordando la validità delle relazioni

(5.30)

$$\sum_{k=0}^{N-1} \cos jx_k \cos lx_k = \begin{cases} 0 & j \neq l \\ N/2 & j = l \neq 0 \text{ se } N \text{ è dispari, e } j = l \neq 0, N/2 \text{ se } N \text{ è pari} \\ N & j = l = 0 \text{ se } N \text{ è dispari, e } j = l = 0, N/2 \text{ se } N \text{ è pari} \end{cases}$$

$$\sum_{k=0}^{N-1} \cos jx_k \sin lx_k = 0$$

$$\sum_{k=0}^{N-1} \sin jx_k \sin lx_k = \begin{cases} 0 & j = l = 0 \text{ oppure } j \neq l \\ 0 & N \text{ è pari e } j = l = N/2 \\ N/2 & j = l \neq 0 \text{ se } N \text{ è dispari, e } j = l \neq 0, N/2 \text{ se } N \text{ è pari} \end{cases}$$

per  $0 \leq j, l \leq \lfloor N/2 \rfloor$ , e considerando le quantità

$$\begin{aligned} \sum_{k=0}^{N-1} T_n(x_k) \cos lx_k &= \sum_{k=0}^{N-1} f(x_k) \cos lx_k \\ \sum_{k=0}^{N-1} T_n(x_k) \sin lx_k &= \sum_{k=0}^{N-1} f(x_k) \sin lx_k \end{aligned}$$

dove per esempio  $T_n(x)$  assume una delle espressioni in (5.29), otteniamo

$$(5.31) \quad \begin{aligned} a_j &= \frac{2}{N} \sum_{k=0}^{N-1} f(x_k) \cos jx_k \\ b_j &= \frac{2}{N} \sum_{k=0}^{N-1} f(x_k) \sin jx_k \end{aligned}$$

ovvero

$$c_j = \frac{1}{2}(a_j - i b_j) = \frac{1}{N} \sum_{k=0}^{N-1} f(x_k) e^{-ijkx_k}$$

Assegnati i valori  $\{f(x_k)\}$ , il calcolo di ogni singolo coefficiente  $a_j$  e  $b_j$  richiede  $N$  moltiplicazioni e  $N - 1$  addizioni<sup>(†)</sup>. La determinazione di una delle rappresentazioni di  $T_n(x)$  in (5.29) mediante le (5.31) coinvolge pertanto  $O(N^2)$  operazioni aritmetiche. Tuttavia, quando scegliamo  $N = b^m$ ,  $b > 1$  intero, possiamo costruire un algoritmo (vedi il paragrafo 5.6) che ci consente di calcolare tutti i coefficienti di  $T_n(x)$  con sole  $O(N \log_2 N)$  operazioni.

## 5.6 Algoritmo FFT

Come già abbiamo visto in 5.5, nell'interpolazione trigonometrica in  $[0, 2\pi]$  su nodi equidistanti  $x_k = k \frac{2\pi}{N}$ ,  $k = 0, 1, \dots, N - 1$ , o più in generale nell'interpolazione con polinomi complessi del tipo

$$P_{N-1}(x) = c_0 + c_1 z + \dots + c_{N-1} z^{N-1}$$

e nodi  $z_k = e^{ix_k}$ ,  $k = 0, 1, \dots, N - 1$ , si presenta il problema di determinare gli  $N$  coefficienti

$$(5.32) \quad c_j = \frac{1}{N} \sum_{k=0}^{N-1} f_k e^{-ikj\frac{2\pi}{N}}, \quad j = 0, 1, \dots, N - 1$$

corrispondenti agli  $N$  valori, reali o complessi,  $f_0, f_1, \dots, f_{N-1}$ . La trasformazione (5.32), che associa alla successione  $\{f_k\}$  quella delle  $c_j$ , rappresenta la ben nota *Analisi di Fourier Discreta*, e viene solitamente denominata *Trasformata di Fourier Discreta* (DFT). Le sue applicazioni sono molteplici; vedasi ad esempio [5.16], [5.17], [5.19].

Una trasformazione analoga è l'inversa della (5.32), ovvero la cosiddetta *Sintesi Discreta di Fourier*, che associa all'insieme  $\{c_j\}$  i valori

$$(5.33) \quad f_k = \sum_{j=0}^{N-1} c_j e^{ikj\frac{2\pi}{N}}, \quad k = 0, 1, \dots, N - 1$$

---

<sup>(†)</sup> Nel conteggio trascuriamo il fattore  $2/N$ , perché comune a tutti i coefficienti, e il calcolo dei valori  $\cos jx_k$  e  $\sin jx_k$ .

L'utilizzo dell'espressione (5.32) o (5.33) per determinare la corrispondente trasformata discreta richiede complessivamente  $N^2$  operazioni aritmetiche. Per alcune applicazioni, ove  $N > 10^3$ , tale mole di calcolo risulta eccessiva. Si pone pertanto il problema di costruire algoritmi alternativi che consentano di determinare le quantità incognite con un numero di operazioni aritmetiche di ordine inferiore a  $N^2$ . La soluzione ci viene data dall'algoritmo FFT (*Fast Fourier Transform*), che illustreremo nel caso particolare, ma molto utilizzato,  $N = 2^l$ .

Consideriamo la (5.32). Posto  $\omega = e^{-i\frac{2\pi}{N}}$ , e quindi

$$(5.34) \quad c_j^{(N)} = c_j = \frac{1}{N} \sum_{k=0}^{N-1} f_k \omega^{jk}, \quad j = 0, 1, \dots, N-1$$

suddividiamo quest'ultima sommatoria in due parti:

$$\begin{aligned} c_j^{(N)} &= \frac{1}{N} \sum_{k=0}^{N/2-1} [f_k \omega^{jk} + f_{N/2+k} \omega^{j(N/2+k)}] \\ &= \frac{1}{N} \sum_{k=0}^{N/2-1} [f_k + f_{N/2+k} \omega^{jN/2}] \omega^{jk} \end{aligned}$$

Successivamente, osservando che

$$\omega^{jN/2} = e^{-ij\pi} = (-1)^j$$

scriviamo

$$\begin{aligned} c_{2m_1}^{(N)} &= \frac{1}{2} \left[ \frac{1}{N/2} \sum_{k=0}^{N/2-1} (f_k + f_{N/2+k}) (\omega^2)^{m_1 k} \right] & m_1 = 0, 1, \dots, N/2 - 1 \\ c_{2m_1+1}^{(N)} &= \frac{1}{2} \left[ \frac{1}{N/2} \sum_{k=0}^{N/2-1} (f_k - f_{N/2+k}) \omega^k (\omega^2)^{m_1 k} \right] \end{aligned}$$

ovvero, definendo

$$(5.35) \quad \begin{cases} f_k^{(1)} = f_k + f_{N/2+k} & k = 0, 1, \dots, N/2 - 1 \\ f_{N/2+k}^{(1)} = (f_k - f_{N/2+k}) \omega^k & \end{cases}$$

$$\begin{aligned} (5.36) \quad c_{2m_1}^{(N)} &= \frac{1}{2} \left[ \frac{1}{N/2} \sum_{k=0}^{N/2-1} f_k^{(1)} \omega_1^{m_1 k} \right] & m_1 = 0, 1, \dots, N/2 - 1 \\ c_{2m_1+1}^{(N)} &= \frac{1}{2} \left[ \frac{1}{N/2} \sum_{k=0}^{N/2-1} f_{N/2+k}^{(1)} \omega_1^{m_1 k} \right] \end{aligned}$$

con  $\omega_1 = e^{-i\frac{2\pi}{N/2}}$ .

La (5.36) ci consente di affermare che *la determinazione di una DFT su N punti può essere effettuata mediante due DFT, ciascuna su N/2 punti*. Ovviamenete possiamo applicare questo risultato alle due sommatorie presenti in (5.36) e ottenere quattro DFT su N/4 punti; ovvero, definite

$$c_{1,m_1}^{(N/2)} = \frac{1}{N/2} \sum_{k=0}^{N/2-1} f_k^{(1)} \omega_1^{m_1 k}$$

e

$$c_{2,m_1}^{(N/2)} = \frac{1}{N/2} \sum_{k=0}^{N/2-1} f_{N/2+k}^{(1)} \omega_1^{m_1 k}$$

scrivere

$$\begin{aligned} c_{1,2m_2}^{(N/2)} &= \frac{1}{2} \left[ \frac{1}{N/4} \sum_{k=0}^{N/4-1} f_k^{(2)} \omega_2^{m_2 k} \right] \\ c_{1,2m_2+1}^{(N/2)} &= \frac{1}{2} \left[ \frac{1}{N/4} \sum_{k=0}^{N/4-1} f_{N/4+k}^{(2)} \omega_2^{m_2 k} \right] \\ c_{2,2m_2}^{(N/2)} &= \frac{1}{2} \left[ \frac{1}{N/4} \sum_{k=0}^{N/4-1} f_{N/2+k}^{(2)} \omega_2^{m_2 k} \right] \\ c_{2,2m_2+1}^{(N/2)} &= \frac{1}{2} \left[ \frac{1}{N/4} \sum_{k=0}^{N/4-1} f_{3N/4+k}^{(2)} \omega_2^{m_2 k} \right] \end{aligned} \quad m_2 = 0, 1, \dots, N/4 - 1$$

con

$$\begin{aligned} \omega_2 &= e^{-i\frac{2\pi}{N/4}} \\ f_k^{(2)} &= f_k^{(1)} + f_{N/4+k}^{(1)} \\ f_{N/4+k}^{(2)} &= (f_k^{(1)} - f_{N/4+k}^{(1)}) \omega_1^k \quad k = 0, 1, \dots, N/4 - 1 \\ f_{N/2+k}^{(2)} &= f_{N/2+k}^{(1)} + f_{3N/4+k}^{(1)} \\ f_{3N/4+k}^{(2)} &= (f_{N/2+k}^{(1)} - f_{3N/4+k}^{(1)}) \omega_1^k \end{aligned}$$

Il processo di sdoppiamento delle singole sommatorie può essere portato avanti esattamente  $l$  volte, sino ad ottenere  $N$  DFT, ciascuna su un solo punto.

► **Esempio.** Consideriamo il caso  $N = 8 = 2^3$  e procediamo costruendo i successivi vettori  $\{f_i^{(m)}\}$ ,  $m = 1, \dots, l$ :

$$\begin{pmatrix} f_0^{(1)} \\ f_1^{(1)} \\ f_2^{(1)} \\ f_3^{(1)} \\ f_4^{(1)} \\ f_5^{(1)} \\ f_6^{(1)} \\ f_7^{(1)} \end{pmatrix} = \begin{pmatrix} f_0 + f_4 \\ f_1 + f_5 \\ f_2 + f_6 \\ f_3 + f_7 \\ f_0 - f_4 \\ (f_1 - f_5)\omega \\ (f_2 - f_6)\omega^2 \\ (f_3 - f_7)\omega^3 \end{pmatrix} \quad \begin{aligned} c_{2m_1} &= c_{2m_1}^{(8)} = \frac{1}{2}c_{1,m_1}^{(4)} \\ c_{1,m_1}^{(4)} &= \frac{1}{4} \sum_{k=0}^3 f_k^{(1)} \omega^{m_1 k} \\ c_{2m_1+1} &= c_{2m_1+1}^{(8)} = \frac{1}{2}c_{2,m_1}^{(4)} \quad m_1 = 0, 1, 2, 3 \\ c_{2,m_1}^{(4)} &= \frac{1}{4} \sum_{k=0}^3 f_{4+k}^{(1)} \omega^{m_1 k} \end{aligned}$$

$$\begin{pmatrix} f_0^{(2)} \\ f_1^{(2)} \\ f_2^{(2)} \\ f_3^{(2)} \\ f_4^{(2)} \\ f_5^{(2)} \\ f_6^{(2)} \\ f_7^{(2)} \end{pmatrix} = \begin{pmatrix} f_0^{(1)} + f_2^{(1)} \\ f_1^{(1)} + f_3^{(1)} \\ f_0^{(1)} - f_2^{(1)} \\ (f_1^{(1)} - f_3^{(1)})\omega_1 \\ f_4^{(1)} + f_6^{(1)} \\ f_5^{(1)} + f_7^{(1)} \\ f_4^{(1)} - f_6^{(1)} \\ (f_5^{(1)} - f_7^{(1)})\omega_1 \end{pmatrix} \quad \begin{aligned} c_{2(2m_2)} &= \frac{1}{4} \left[ \frac{1}{2} \sum_{k=0}^1 f_k^{(2)} \omega_1^{m_2 k} \right] = \frac{1}{4} c_{1,1,m_2}^{(2)} \\ c_{2(2m_2+1)} &= \frac{1}{4} \left[ \frac{1}{2} \sum_{k=0}^1 f_{2+k}^{(2)} \omega_1^{m_2 k} \right] = \frac{1}{4} c_{1,2,m_2}^{(2)} \quad m_2 = 0, 1 \\ c_{2(2m_2)+1} &= \frac{1}{4} \left[ \frac{1}{2} \sum_{k=0}^1 f_{4+k}^{(2)} \omega_1^{m_2 k} \right] = \frac{1}{4} c_{2,1,m_2}^{(2)} \\ c_{2(2m_2+1)+1} &= \frac{1}{4} \left[ \frac{1}{2} \sum_{k=0}^1 f_{6+k}^{(2)} \omega_1^{m_2 k} \right] = \frac{1}{4} c_{2,2,m_2}^{(2)} \end{aligned}$$

$$\begin{pmatrix} f_0^{(3)} \\ f_1^{(3)} \\ f_2^{(3)} \\ f_3^{(3)} \\ f_4^{(3)} \\ f_5^{(3)} \\ f_6^{(3)} \\ f_7^{(3)} \end{pmatrix} = \begin{pmatrix} f_0^{(2)} + f_1^{(2)} \\ f_0^{(2)} - f_1^{(2)} \\ f_2^{(2)} + f_3^{(2)} \\ f_2^{(2)} - f_3^{(2)} \\ f_4^{(2)} + f_5^{(2)} \\ f_4^{(2)} - f_5^{(2)} \\ f_6^{(2)} + f_7^{(2)} \\ f_6^{(2)} - f_7^{(2)} \end{pmatrix} \quad \begin{aligned} c_{4(2m_3)} &= c_0 = \frac{1}{8}f_0^{(3)} \\ c_{4(2m_3+1)} &= c_4 = \frac{1}{8}f_1^{(3)} \\ c_{2(4m_3+1)} &= c_2 = \frac{1}{8}f_2^{(3)} \\ c_{2(2(2m_3+1)+1)} &= c_6 = \frac{1}{8}f_3^{(3)} \quad m_3 = 0 \\ c_{2(4m_3+1)+1} &= c_1 = \frac{1}{8}f_4^{(3)} \\ c_{2(2(2m_3+1))+1} &= c_5 = \frac{1}{8}f_5^{(3)} \\ c_{2(4m_3+1)+1} &= c_3 = \frac{1}{8}f_6^{(3)} \\ c_{2(2(2m_3+1)+1)+1} &= c_7 = \frac{1}{8}f_7^{(3)} \end{aligned}$$

Come possiamo osservare, i coefficienti  $c_j$  che abbiamo ottenuto trasformando<sup>(†)</sup> i successivi vettori  $\{f_i^{(m)}\}$  non sono posti nell'ordine naturale. Esiste tuttavia una regola assai semplice per individuare tale ordine: *considera le rappresentazioni binarie degli interi  $0, 1, \dots, N - 1$ , inverti le successioni di bit delle singole rappresentazioni e determina i corrispondenti numeri decimali*. Nel caso del nostro esempio otteniamo

(†) Tale trasformazione è lineare, del tipo  $\{f_i^{(m+1)}\} = W_m \{f_i^{(m)}\}$  dove  $W_m$  è una matrice con una struttura a blocchi particolarmente semplice; vedi ad esempio [5.16, §4.2].

$j$	rappresentazione binaria	inversione bit	numero decimale
0	0 0 0	0 0 0	0
1	0 0 1	1 0 0	4
2	0 1 0	0 1 0	2
3	0 1 1	1 1 0	6
4	1 0 0	0 0 1	1
5	1 0 1	1 0 1	5
6	1 1 0	0 1 1	3
7	1 1 1	1 1 1	7

Tabella 5.1

**Algoritmo 9:** FFT( $f, l$ )

*Commento.* Assegnati  $N = 2^l$  dati reali o complessi  $f_0, f_1, \dots, f_{N-1}$ , memorizzati nel vettore  $f$ , l'algoritmo ne determina la trasformata discreta di Fourier (non ordinata) a meno del fattore  $1/N$ . Quest'ultima viene memorizzata nello stesso vettore  $f$ .

*Parametri.* **Input:**  $f, l$   
**Output:**  $f$

- 1: **ciclo 1:**  $m = 1, \dots, l$
- 2:     $m_2 \leftarrow 2^{m-1}$
- 3:     $m_3 \leftarrow N/2^m$
- 4:    **ciclo 2:**  $i = 0, \dots, 2(m_2 - 1)(2)$  (†)
- 5:     **ciclo 3:**  $k = 0, \dots, m_3 - 1$
- 6:        $g \leftarrow f_{i \cdot m_3 + k} + f_{(i+1)m_3 + k}$
- 7:        $f_{(i+1)m_3 + k} \leftarrow (f_{i \cdot m_3 + k} - f_{(i+1)m_3 + k})\omega^{km_2}$
- 8:        $f_{i \cdot m_3 + k} \leftarrow g$
- 9:     **fine ciclo 3**
- 10:   **fine ciclo 2**
- 11: **fine ciclo 1**
- 12: **esci**

Ogni trasformazione del vettore  $f = \{f_j^{(m)}\}$  richiede  $N$  addizioni (o sottrazioni) e  $N/2 - 2^{m-1}$  moltiplicazioni,  $m = 1, \dots, l$ . Il costo complessivo dell'algoritmo, supponendo di aver già precalcolato le potenze  $\omega^k$  necessarie, è di  $N \log_2 N$  addizioni e di  $(N/2) \log_2(N/2)$  moltiplicazioni complesse.

(†) Ovvero  $i = 0, 2, 4, 6, \dots, 2(m_2 - 1)$

È possibile riformulare l'algoritmo prendendo  $N = b^l$ ,  $b$  intero  $> 1$ , oppure  $N$  intero non primo scomponibile nel prodotto di numeri interi, preferibilmente piccoli, non necessariamente distinti. Per un'esposizione più completa dell'algoritmo, delle sue proprietà e applicazioni, nonché delle sue possibili varianti, consigliamo la lettura dei testi [5.16] e [5.19].

▷ **Osservazione.** Ricordando la periodicità delle funzioni  $\cos(x)$  e  $\sin(x)$ , possiamo costruire l'algoritmo FFT anche per il calcolo di trasformate discrete di forma

$$a_j = \frac{2}{N} \sum_{k=0}^{N-1} f_k \cos \left( j \frac{k2\pi}{N} \right), \quad j = 0, 1, \dots, N-1$$

oppure

$$b_j = \frac{2}{N} \sum_{k=0}^{N-1} f_k \sin \left( j \frac{k2\pi}{N} \right), \quad j = 1, \dots, N-1$$

□

## 5.7 Interpolazione con funzioni polinomiali a tratti. Funzioni spline

Supponiamo di avere  $n + 1$  punti  $(x_i, y_i)$ ,  $i = 0, 1, \dots, n$ , con

$$(5.37) \quad a \equiv x_0 < x_1 < \dots < x_{n-1} < x_n \equiv b$$

Sappiamo che esiste uno ed un sol polinomio di grado  $n$  passante per tali punti; tuttavia, il comportamento a volte eccessivamente oscillatorio di questo polinomio, quando  $n$  non è piccolo (per esempio  $n \geq 8$ ), può non essere accettabile (vedi ad esempio la figura 5.6). In questi casi preferiamo agire diversamente, operando con tratti di polinomi di grado basso (per esempio  $d = 1, 2, 3$ ).

In particolare, nel caso più semplice  $d = 1$ , uniamo ciascuna coppia di punti consecutivi  $(x_i, y_i)$ ,  $(x_{i+1}, y_{i+1})$  con un segmento, ovvero prendiamo come funzione interpolante una poligonale quale quella disegnata in figura 5.7, definita dall'espressione

$$f_n(x) = \frac{(x_{i+1} - x)y_i + (x - x_i)y_{i+1}}{x_{i+1} - x_i}, \quad x_i \leq x \leq x_{i+1}$$

Quando  $y_i = f(x_i)$  con  $f \in C^2[x_0, x_n]$ , dalla (5.13) deduciamo la stima

$$\|f(x) - f_n(x)\|_\infty = O(h^2), \quad h = \max_i(x_{i+1} - x_i)$$

Se supponiamo  $n$  pari e dividiamo l'insieme dei nodi  $\{x_i\}$  in terne consecutive  $\{(x_{2i-2}, x_{2i-1}, x_{2i}), i = 1, \dots, \frac{n}{2}\}$ , possiamo prendere  $d = 2$ , ovvero definire in ogni intervallo

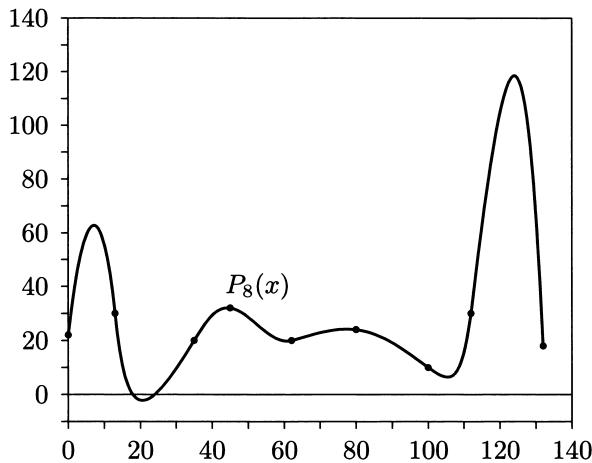


Figura 5.6

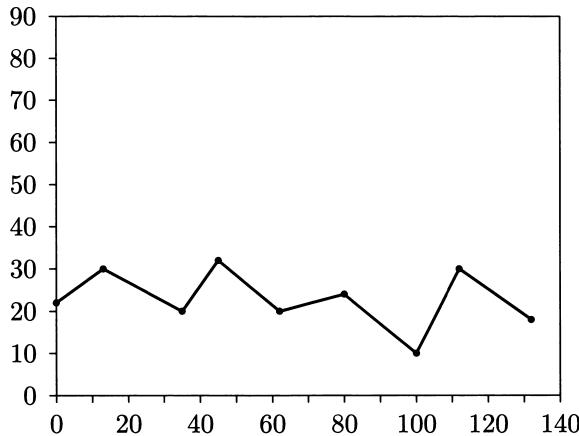


Figura 5.7

$[x_{2i-2}, x_{2i}]$  il polinomio di interpolazione di grado 2 passante per i punti  $(x_{2i-2}, y_{2i-2})$ ,  $(x_{2i-1}, y_{2i-1})$ ,  $(x_{2i}, y_{2i})$ . In questo caso la funzione  $f_n(x)$  che interpola i dati  $(x_i, y_i)$ ,  $i = 0, \dots, n$ , è un'unione di archi di parabola (vedi figura 5.8). L'idea può essere facilmente estesa ad unioni di tratti di polinomi di grado  $d > 2$ .

Le funzioni costruite seguendo questa strategia vengono generalmente chiamate *funzioni polinomiali a tratti*; esse sono continue in tutto l'intervallo  $[a, b]$ , ma di solito la loro derivata  $f'_n(x)$  è discontinua nei nodi di unione di due archi distinti, come facilmente si evince dai grafici precedenti. Sul comportamento dell'errore  $f(x) - f_n(x)$  ricordiamo il risultato (5.13).

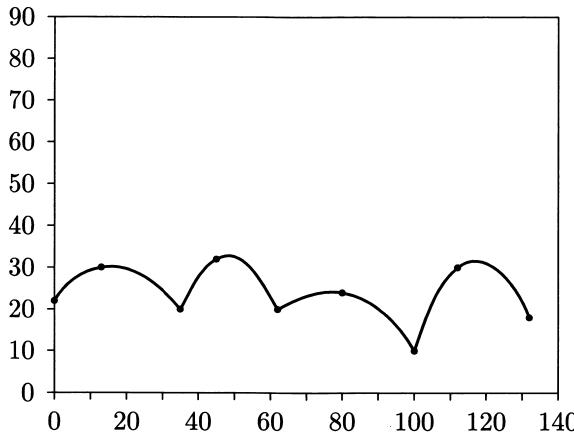


Figura 5.8

Le funzioni  $f_n(x)$  predette non sono le sole alternative ai polinomi di interpolazione; per esempio potremmo utilizzare la funzione definita nell'esercizio di pag. 134. Tuttavia, quest'ultima richiede la conoscenza della derivata prima  $f'(x)$  nei nodi di interpolazione  $\{x_i\}$ .

Il metodo di approssimazione che ora esaminiamo costituisce un ulteriore esempio di come il problema dell'interpolazione con funzioni polinomiali a tratti possa essere affrontato.

**Definizione 5.1.** Fissato un intero  $d \geq 1$ ,  $S_d(x)$  è una funzione spline di ordine  $d$  associata alla suddivisione (5.37) dell'intervallo  $[a, b]$  se:

- (i)  $S_d(x)$  è un polinomio di grado  $d$  in ogni intervallo  $[x_{i-1}, x_i]$ ,  $i = 1, \dots, n$ , ovvero, in tale intervallo assume la forma:  $S_d(x) \equiv a_0^i s^d + a_1^i x^{d-1} + \dots + a_d^i$ ;
- (ii)  $S_d^{(k)}(x) = d^k S_d(x)/dx^k$  è una funzione (globalmente) continua in  $[a, b]$  per ogni  $k = 0, 1, \dots, d-1$ .

Dalla definizione stessa di  $S_d(x)$  segue che la derivata di una spline di ordine  $m$  è un'altra spline di ordine  $m-1$ , mentre la primitiva di  $S_m(x)$  è una spline di ordine  $m+1$ .

Le spline  $S_d$  che abbiamo definito sono quindi particolari funzioni polinomiali a tratti di grado locale  $d$  (condizione (i) della definizione 5.1), che godono delle ulteriori proprietà di regolarità stabilite dalla condizione (ii).

Le funzioni spline più usate sono certamente quelle cubiche. La determinazione di una spline cubica che interpola la funzione  $y = f(x)$  nei nodi (5.37), cioè tale che

$$(a) \quad S_3(x_i) = y_i, \quad i = 0, 1, \dots, n$$

risulta assai semplice. In questo caso la definizione 5.1 può essere semplificata nella forma seguente:

- (b)  $S_3(x) = a_i + b_i x + c_i x^2 + d_i x^3, \quad x \in [x_{i-1}, x_i], \quad i = 1, 2, \dots, n$   
(c)  $S_3^{(k)}(x_i^+) = S_3^{(k)}(x_i^-), \quad i = 1, 2, \dots, n-1, \quad k = 0, 1, 2$

Con il simbolo  $S_3^{(k)}(x_i^+)$  denotiamo il valore che la derivata  $k$ -esima del polinomio di terzo grado  $a_{i+1} + b_{i+1}x + c_{i+1}x^2 + d_{i+1}x^3$ , che definisce  $S_3(x)$  nell'intervallo  $[x_i, x_{i+1}]$ , assume nel nodo  $x_i$ .  $S_3^{(k)}(x_i^-)$  rappresenta invece il corrispondente valore assunto dalla derivata  $k$ -esima di  $a_i + b_i x + c_i x^2 + d_i x^3$ .

Le condizioni (a) e (c) conducono ad un sistema lineare di  $4n - 2$  equazioni nelle  $4n$  incognite  $\{a_i, b_i, c_i, d_i\}$ ,  $i = 1, 2, \dots, n$ , definite in (b). Rimane ancora a nostra disposizione la possibilità di imporre due ulteriori condizioni. Tuttavia, anziché risolvere un sistema di ordine  $4n$ , possiamo costruire  $S_3(x)$ , univocamente se imponiamo due condizioni aggiuntive opportune, risolvendo un sistema lineare di ordine al più  $n + 1$ , con struttura particolarmente semplice.

A tale fine introduciamo le nuove incognite

$$M_i = S_3''(x_i), \quad i = 0, 1, \dots, n$$

Poiché  $S_3(x)$  è in ogni intervallo  $[x_{i-1}, x_i]$  un polinomio di grado 3, la derivata seconda  $S_3''(x)$  è a sua volta, in tali intervalli, una funzione lineare di forma

$$S_3''(x) = \frac{(x_i - x)M_{i-1} + (x - x_{i-1})M_i}{h_i}, \quad i = 1, 2, \dots, n$$

dove  $h_i = x_i - x_{i-1}$ . Osserviamo che la  $S_3''(x)$  così definita è continua su tutto  $[a, b]$ .

Per individuare  $S_3(x)$  dobbiamo dapprima integrare due volte  $S_3''(x)$ , ottenendo

$$(5.38) \quad S_3(x) = \frac{(x_i - x)^3 M_{i-1} + (x - x_{i-1})^3 M_i}{6h_i} + C_i(x - x_{i-1}) + D_i, \quad x \in [x_{i-1}, x_i]$$

dove  $C_i$  e  $D_i$  sono due costanti arbitrarie. Tuttavia, dovendo  $S_3(x)$  soddisfare le condizioni di interpolazione (a), alle costanti  $C_i$  e  $D_i$  assegnamo i valori

$$C_i = \frac{y_i - y_{i-1}}{h_i} - \frac{h_i(M_i - M_{i-1})}{6} \quad \text{e} \quad D_i = y_{i-1} - \frac{h_i^2}{6} M_{i-1}$$

che deduciamo imponendo  $S_3(x_{i-1}) = y_{i-1}$  e  $S_3(x_i) = y_i$ ; in questo caso otteniamo

$$(5.39) \quad S_3(x) = \frac{(x_i - x)^3 M_{i-1} + (x - x_{i-1})^3 M_i}{6h_i} + \left[ \frac{y_i - y_{i-1}}{h_i} - \frac{h_i}{6}(M_i - M_{i-1}) \right] (x - x_{i-1}) + y_{i-1} - \frac{h_i^2}{6} M_{i-1} \quad x \in [x_{i-1}, x_i]$$

La funzione  $S_3(x)$  definita dalla (5.39), dove i valori  $M_0, M_1, \dots, M_n$  sono per ora arbitrari, non è ancora una spline. Finora abbiamo imposto la continuità solo a  $S_3''(x)$  e  $S_3(x)$ . Perché  $S_3(x)$  sia una spline cubica è sufficiente determinare le quantità  $\{M_i\}$  in modo che le condizioni

$$(5.40) \quad \lim_{x \rightarrow x_i^-} S'_3(x) = \lim_{x \rightarrow x_i^+} S'_3(x), \quad i = 1, 2, \dots, n-1$$

siano soddisfatte, cosicché anche  $S'_3(x)$  risulti continua in  $[a, b]$ . La funzione  $S'_3(x)$  è rappresentata dall'espressione

$$S'_3(x) = \frac{(x - x_{i-1})^2 M_i - (x_i - x)^2 M_{i-1}}{2h_i} + \frac{y_i - y_{i-1}}{h_i} - \frac{h_i}{6}(M_i - M_{i-1})$$

nell'intervallo  $[x_{i-1}, x_i]$ , e da

$$S'_3(x) = \frac{(x - x_i)^2 M_{i+1} - (x_{i+1} - x)^2 M_i}{2h_{i+1}} + \frac{y_{i+1} - y_i}{h_{i+1}} - \frac{h_{i+1}}{6}(M_{i+1} - M_i)$$

in  $[x_i, x_{i+1}]$ . Sostituendo queste due espressioni in (5.40) non è difficile dedurre il seguente sistema lineare

$$(5.41) \quad \begin{aligned} h_i M_{i-1} + 2(h_i + h_{i+1}) M_i + h_{i+1} M_{i+1} &= \frac{6}{h_{i+1}}(y_{i+1} - y_i) \\ &\quad - \frac{6}{h_i}(y_i - y_{i-1}), \quad i = 1, 2, \dots, n-1 \end{aligned}$$

di  $(n-1)$  equazioni nelle  $(n+1)$  incognite  $M_0, M_1, \dots, M_n$ .

Dobbiamo infine imporre altre due condizioni. Esaminiamo tre possibili scelte, non le sole<sup>(†)</sup>, ma certamente le più note.

*Scelta 1.* Imponiamo le condizioni

$$(5.42) \quad S''_3(x_0) = f'(x_0) \quad \text{e} \quad S'_3(x_n) = f'(x_n)$$

che si trasformano nelle seguenti due equazioni aggiuntive:

$$(5.43) \quad \begin{aligned} 2h_1 M_0 + h_1 M_1 &= 6 \left[ \frac{y_1 - y_0}{h_1} - f'(x_0) \right] \\ h_n M_{n-1} + 2h_n M_n &= 6 \left[ f'(x_n) - \frac{y_n - y_{n-1}}{h_n} \right] \end{aligned}$$

---

(†) Per altre condizioni vedere ad esempio [5.14] e [15].

L'unione delle (5.43) alle (5.41) produce il sistema

$$(5.44) \quad \begin{pmatrix} 2h_1 & h_1 & & \\ h_1 & 2(h_1 + h_2) & h_2 & \\ \ddots & \ddots & \ddots & \\ h_{n-1} & 2(h_{n-1} + h_n) & h_n & 2h_n \end{pmatrix} \begin{pmatrix} M_0 \\ M_1 \\ \vdots \\ M_{n-1} \\ M_n \end{pmatrix} = 6 \begin{pmatrix} \frac{y_1 - y_0}{h_1} - f'(x_0) \\ \frac{y_2 - y_1}{h_2} - \frac{y_1 - y_0}{h_1} \\ \vdots \\ \frac{y_n - y_{n-1}}{h_n} - \frac{y_{n-1} - y_{n-2}}{h_{n-1}} \\ f'(x_n) - \frac{y_n - y_{n-1}}{h_n} \end{pmatrix}$$

tridiagonale simmetrico, a diagonale dominante. Tale sistema è non singolare e viene risolto con il metodo delle eliminazioni di Gauss senza pivoting né scaling.

*Scelta 2.*

$$(5.45) \quad M_0 = f''(x_0) \quad \text{e} \quad M_n = f''(x_n)$$

Anche in questo caso perveniamo ad un sistema tridiagonale simmetrico a diagonale dominante (di ordine  $n - 1$ ):

$$(5.46) \quad \begin{pmatrix} 2(h_1 + h_2) & h_2 & & \\ h_2 & 2(h_2 + h_3) & h_3 & \\ \ddots & \ddots & \ddots & \\ h_{n-2} & 2(h_{n-2} + h_{n-1}) & h_{n-1} & 2(h_{n-1} + h_n) \end{pmatrix} \begin{pmatrix} M_1 \\ M_2 \\ \vdots \\ M_{n-2} \\ M_{n-1} \end{pmatrix} = \\ 6 \begin{pmatrix} \frac{y_2 - y_1}{h_2} - \frac{y_1 - y_0}{h_1} - \frac{h_1}{6} f''(x_0) \\ \frac{y_3 - y_2}{h_3} - \frac{y_2 - y_1}{h_2} \\ \vdots \\ \frac{y_{n-1} - y_{n-2}}{h_{n-1}} - \frac{y_{n-2} - y_{n-3}}{h_{n-2}} \\ \frac{y_n - y_{n-1}}{h_n} - \frac{y_{n-1} - y_{n-2}}{h_{n-1}} - \frac{h_n}{6} f''(x_n) \end{pmatrix}$$

Quando invece poniamo  $M_0 = M_n = 0$ , indipendentemente dai valori  $f''(x_0)$  e  $f''(x_n)$ , le spline cubiche corrispondenti vengono denominate *naturali*.

*Scelta 3.* Quando il fenomeno in questione è periodico in  $[a, b]$ , dopo aver osservato che le condizioni di interpolazione automaticamente implicano  $S_3(x_0) = S_3(x_n)$  imponiamo le seguenti ulteriori condizioni di periodicità alla spline:

$$(5.47) \quad \begin{aligned} S'_3(x_0) &= S'_3(x_n) \\ S''_3(x_0) &= S''_3(x_n) \end{aligned}$$

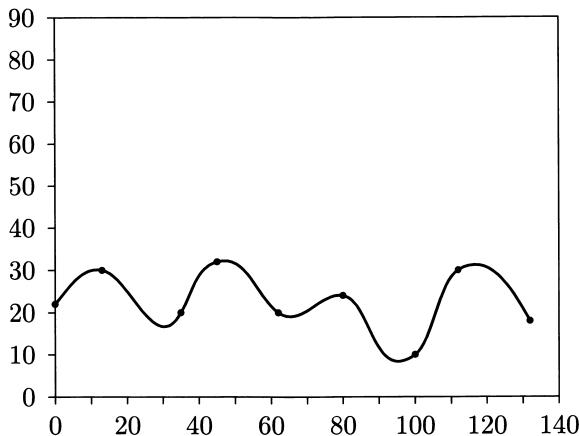
Il sistema corrispondente<sup>(†)</sup>

$$(5.48) \quad \begin{aligned} & \left( \begin{array}{cccccc} 2(h_1 + h_2) & & h_2 & & h_1 & \\ h_2 & 2(h_2 + h_3) & h_3 & & & \\ & \ddots & \ddots & \ddots & & \\ & h_{n-1} & 2(h_{n-1} + h_n) & h_n & & \\ h_1 & h_n & & 2(h_1 + h_n) & & \end{array} \right) \begin{pmatrix} M_1 \\ M_2 \\ \vdots \\ M_{n-1} \\ M_n \end{pmatrix} = \\ & = 6 \begin{pmatrix} \frac{y_2 - y_1}{h_2} - \frac{y_1 - y_n}{h_1} \\ \frac{y_3 - y_2}{h_3} - \frac{y_2 - y_1}{h_2} \\ \vdots \\ \frac{y_n - y_{n-1}}{h_n} - \frac{y_{n-1} - y_{n-2}}{h_{n-1}} \\ \frac{y_1 - y_n}{h_1} - \frac{y_n - y_{n-1}}{h_n} \end{pmatrix} \end{aligned}$$

non è più esattamente tridiagonale, ma è ancora simmetrico e a diagonale dominante.

In tutti e tre i casi esaminati la spline cubica cercata *esiste* ed è *unica*. Inoltre, per ognuna delle scelte proposte la generica spline cubica associata ai nodi  $\{x_i\}$  (che supponiamo fissati) dipende da  $n+3$  parametri: le  $(n+1)$  ordinate  $\{y_i\}$  e le due condizioni aggiuntive agli estremi.

Nella figura 5.9 viene riportata la funzione spline cubica naturale che interpola i dati numerici già considerati nelle figure 5.6, 5.7 e 5.8.



**Figura 5.9**

(†) Ricordiamo che in questo caso  $y_0 = y_n$ .

**Algoritmo 10:** Spline( $n, x, y, z$ )

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

*Parametri.* **Input:**  $n, x, y$

**Output:**  $z$

- 1: **ciclo 1:**  $i = 1, \dots, n - 2$
  - 2:    $d_i \leftarrow 2(x_{i+1} - x_{i-1})$
  - 3:    $c_i \leftarrow x_{i+1} - x_i$
  - 4:    $b_i \leftarrow 6 \left( \frac{y_{i+1} - y_i}{x_{i+1} - x_i} - \frac{y_i - y_{i-1}}{x_i - x_{i-1}} \right)$
  - 5: **fine ciclo 1**
  - 6:  $d_{n-1} \leftarrow 2(x_n - x_{n-2})$
  - 7:  $b_{n-1} \leftarrow 6 \left( \frac{y_n - y_{n-1}}{x_n - x_{n-1}} - \frac{y_{n-1} - y_{n-2}}{x_{n-1} - x_{n-2}} \right)$
  - \*: *Processo di eliminazione di Gauss per il sistema tridiagonale*
  - 8: **ciclo 2:**  $i = 2, \dots, n - 1$
  - 9:    $d_i \leftarrow d_i - c_{i-1}^2 / d_{i-1}$
  - 10:    $b_i \leftarrow b_i - (c_{i-1} / d_{i-1}) b_{i-1}$
  - 11: **fine ciclo 2**
  - \*: *Soluzione sistema bidiagonale*
  - 12:  $z_{n-1} \leftarrow b_{n-1} / d_{n-1}$
  - 13: **ciclo 3:**  $i = 2, \dots, n - 1$
  - 14:    $z_{n-i} \leftarrow (b_{n-i} - c_{n-i} z_{n+1-i}) / d_{n-i}$
  - 15: **fine ciclo 3**
  - 16: **esci**
- 

**Algoritmo 11:** Valspl( $n, x, y, z, t, s$ )

*Commento.* Noti i coefficienti  $M_0 = 0, M_1, M_2, \dots, M_{n-1}, M_n = 0$ , contenuti nel vettore  $z$ , della spline cubica naturale passante per i punti  $(x_i, y_i)$ ,  $i = 0, 1, \dots, n$ , Valspl valuta il valore  $s$  che tale spline assume nel punto  $t$ ,  $x_0 \leq t \leq x_n$ .

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

**Output:**  $s$

- 1: **ciclo 1:**  $i = 1, \dots, n - 1$

- 
- 2: se  $t \leq x_i$  allora vai al punto 5  
 3: fine ciclo 1  
 4:  $i \leftarrow n$   
 5:  $h \leftarrow x_i - x_{i-1}$   
 6:  $s \leftarrow \frac{(x_i - t)^3 z_{i-1} + (t - x_{i-1})^3 z_i}{6h} + \left[ \frac{y_i - y_{i-1}}{h} - \frac{h}{6}(z_i - z_{i-1}) \right] (t - x_{i-1}) + y_{i-1} - \frac{h^2}{6} z_{i-1}$   
 7: esci
- 

Il risultato seguente caratterizza le spline cubiche nei tre casi esaminati.

**Teorema 5.8.** ([5.6, cap. 2]). *Tra tutte le funzioni  $f(x) \in C^2[a, b]$  che assumono valori  $\{y_i\}$  nei nodi  $\{x_i\}$ , con (i)  $f'(x_0) = y'_0$ ,  $f'(x_n) = y'_n$ , oppure (ii)  $f''(x_0) = 0$  e  $f''(x_n) = 0$ , o (iii)  $f^{(k)}(x_0) = f^{(k)}(x_n)$ ,  $k = 0, 1, 2$ , dove  $y'_0$ ,  $y'_n$  sono numeri da noi assegnati, le spline cubiche corrispondenti sono le sole funzioni che minimizzano l'integrale*

$$(5.49) \quad E(f) = \int_{x_0}^{x_n} [f''(x)]^2 dx$$

Particolarmente interessante è la proprietà di minimo assoluto delle spline cubiche naturali: esse sono le uniche funzioni dell'insieme  $\{f(x) \in C^2[a, b], f(x_i) = y_i, i = 0, 1, \dots, n\}$  che minimizzano l'integrale (5.49). Per meglio apprezzare i risultati suddetti, ricordiamo che l'espressione  $f''(x)[1 + f'(x)^2]^{-3/2}$  definisce la curvatura della funzione  $f(x)$  nel punto  $x$ . Se  $|f'(x)|$  è sufficientemente piccola, la quantità  $E(f)$  rappresenta una misura (approssimata) della curvatura totale della  $f(x)$  nell'intervallo  $[a, b]$ .

Concludiamo l'argomento delle spline riportando alcuni risultati di convergenza che dimostrano l'efficacia di queste funzioni.

**Teorema 5.9.** ([5.6, cap. 2]). *Sia  $S_3(x)$  la spline cubica associata ai dati  $\{x_i, f(x_i)\}$ , con condizioni agli estremi di tipo (5.42), oppure (5.45), oppure (5.47) se  $f(x)$  è periodica. Definiamo  $h = \max_{1 \leq i \leq n} h_i$ . Quando  $f(x) \in C^2[a, b]$  abbiamo*

$$\|f^{(p)}(x) - S_3^{(p)}(x)\|_\infty = o(h^{2-p}), \quad p = 0, 1, 2, \quad h \rightarrow 0$$

*Se invece  $f(x) \in C^k[a, b]$ ,  $k = 3, 4$ , ed inoltre esiste una costante  $\gamma$  tale che  $h/h_i \leq \gamma < \infty$  per  $h \rightarrow 0$ ,*

$$\|f^{(p)}(x) - S_3^{(p)}(x)\|_\infty = \begin{cases} o(h^{3-p}), & k = 3 \\ O(h^{4-p}), & k = 4 \end{cases} \quad p = 0, 1, 2, 3$$

È stato tuttavia dimostrato (vedi [5.6, pag. 96]) che  $O(h^4)$  è il massimo ordine di convergenza che possiamo avere con le spline cubiche considerate, ovvero  $\|f(x) - S_3(x)\|_\infty = O(h^4)$  anche quando  $f(x) \in C^k[a, b]$  con  $k > 4$ .

Le spline cubiche naturali godono di interessanti proprietà teoriche; tuttavia se  $f''(a)$  e  $f''(b)$  non sono entrambe nulle l'ordine di approssimazione che esse forniscono non

è quello ottimale. Per esempio, quando  $f(x) \in C^4[a, b]$  l'errore di approssimazione è sempre  $O(h^4)$  in ogni intervallo  $[c, d] \subset (a, b)$ , ma nelle vicinanze degli estremi  $a$  e  $b$  esso si riduce a  $O(h^2)$ .

Quando vogliamo interpolare una funzione non periodica  $f(x) \in C^4[a, b]$ , di cui non si conoscono i valori di  $f'(x)$  o  $f''(x)$  nei due estremi  $a$  e  $b$ , per mantenere l'ordine di convergenza definito nel teorema 5.9 conviene imporre le seguenti condizioni, denominate *not-a-knot*:

$$\begin{aligned} S_3^{(3)}(x_1^+) &= S_3^{(3)}(x_1^-) \\ S_3^{(3)}(x_{n-1}^+) &= S_3^{(3)}(x_{n-1}^-) \end{aligned}$$

che garantiscono la continuità della derivata terza di  $S_3(x)$  in  $x_1$  e  $x_{n-1}$ . Non è difficile dimostrare che imponendo tali condizioni, i due tratti (contigui) di polinomi (di terzo grado) che definiscono la spline cubica nell'intervallo  $(x_0, x_2)$  (oppure in  $(x_{n-2}, x_n)$ ) appartengono ad uno stesso polinomio di terzo grado. Ovvero, in questo caso, i nodi (knot)  $x_1$  e  $x_{n-1}$  sono nodi di interpolazione ma non più di partizione, nel senso che non separano più tratti di polinomi diversi. Come nei casi precedenti, la costruzione delle spline not-a-knot può essere ricondotta alla risoluzione di un sistema lineare tridiagonale.

I risultati presentati non solo ci assicurano la convergenza uniforme in  $[a, b]$  di  $S_3(x)$  a  $f(x)$ , ma ci suggeriscono anche che le spline cubiche sono uno strumento efficace per l'approssimazione uniforme, in tutto  $[a, b]$ , delle stesse derivate di  $f(x)$ .

Dopo aver definito le spline cubiche di interpolazione nei tre casi considerati, ed aver stabilito la loro esistenza e unicità, ci proponiamo ora di esaminare la possibilità di rappresentarle con una formula analoga a quella di Lagrange per i polinomi.

Nel caso della scelta 1 per esempio, consideriamo le  $(n + 3)$  spline cubiche  $S_{3,-1}(x)$ ,  $S_{3,0}(x), \dots, S_{3,n}(x), S_{3,n+1}(x)$ , univocamente definite dalle condizioni

$$\begin{aligned} j = 0, 1, \dots, n : \quad &\begin{cases} S_{3,j}(x_i) = \delta_{ij} & i = 0, 1, \dots, n \\ S'_{3,j}(x_0) = 0 \\ S'_{3,j}(x_n) = 0 \end{cases} \\ &\begin{cases} S_{3,-1}(x_i) = 0, & i = 0, 1, \dots, n \\ S'_{3,-1}(x_0) = 1 \\ S'_{3,-1}(x_n) = 0 \end{cases} \\ &\begin{cases} S_{3,n+1}(x_i) = 0, & i = 0, 1, \dots, n \\ S'_{3,n+1}(x_0) = 0 \\ S'_{3,n+1}(x_n) = 1 \end{cases} \end{aligned}$$

Esse ci permettono di rappresentare la generica spline cubica  $S_3(x)$  di tipo (5.42) passante per i punti  $\{x_i, f(x_i)\}$ ; infatti abbiamo

$$S_3(x) = \sum_{j=0}^n f(x_j) S_{3,j}(x) + f'(x_0) S_{3,-1}(x) + f'(x_n) S_{3,n+1}(x)$$

Per le spline cubiche naturali è sufficiente considerare le  $(n+1)$  spline cubiche naturali  $\{S_{3,j}(x), j = 0, 1, \dots, n\}$  associate all'insieme di dati  $\{x_i, \delta_{ij}\}$  per verificare che

$$S_3(x) = \sum_{j=0}^n f(x_j) S_{3,j}(x)$$

Lasciamo al lettore l'esame dei casi (5.45) e (5.47).

Le funzioni  $S_{3,j}(x)$  vengono denominate *spline cubiche cardinali*. Esse dipendono unicamente dai nodi  $\{x_i\}$ ; non dai particolari dati  $\{y_i\}$  assegnati.

## 5.8 Interpolazione di funzioni di più variabili

Il problema dell'interpolazione polinomiale di funzioni di più variabili generalmente non ha soluzione unica come nel caso monodimensionale. Per esempio, assegnati  $n$  punti  $\{(x_i, y_i)\}$  del piano  $(x, y)$ , ed  $n$  valori  $\{f(x_i, y_i)\}$ , non esiste un unico polinomio, di grado opportuno,  $P_{r,s}(x, y)$  tale che

$$P_{r,s}(x_i, y_i) = f(x_i, y_i), \quad i = 1, \dots, n$$

Per convincerci di ciò basta considerare il caso di  $n$  punti distinti allineati nello spazio: esistono addirittura infiniti piani, ovvero polinomi di grado 1,  $z = \alpha x + \beta y + \gamma$ , passanti per tali punti.

In questo paragrafo ci limitiamo ad osservare come alcuni risultati presentati nel caso di funzioni di una sola variabile possono essere utilizzati per dare una risposta al problema dell'interpolazione di funzioni di due o più variabili. Per non appesantire troppo le notazioni circoscriviamo la nostra presentazione alle funzioni  $f(x, y)$  di due variabili. Supponiamo inoltre che  $f(x, y)$  sia definita in un rettangolo  $R = \{(x, y) : a \leq x \leq b, c \leq y \leq d\}$ .

Esaminiamo dapprima il caso dell'interpolazione polinomiale. Supponiamo di conoscere i valori  $f(x_i, y_j)$  corrispondenti ai nodi  $(x_i, y_j)$  del reticolo definito dalle discretizzazioni  $a \leq x_0 < x_1 < \dots < x_{n-1} < x_n \leq b$  e  $c \leq y_0 < y_1 < \dots < y_{m-1} < y_m \leq d$ . Vogliamo determinare un polinomio in due variabili  $P_{n,m}(x, y)$ , di grado  $n$  nella variabile  $x$  e di grado  $m$  nella variabile  $y$ , tale che

$$P_{n,m}(x_i, y_j) = f(x_i, y_j), \quad i = 0, 1, \dots, n; j = 0, 1, \dots, m$$

Con  $\{l_i(x), i = 0, 1, \dots, n\}$  denotiamo i polinomi fondamentali di Lagrange associati ai nodi  $\{x_i\}$ , e con  $\{\bar{l}_j(y), j = 0, 1, \dots, m\}$  quelli associati a  $\{y_j\}$ . Successivamente consideriamo il polinomio prodotto

$$l_{i,j}(x, y) = l_i(x) \bar{l}_j(y)$$

e osserviamo che

$$l_{i,j}(x_i, y_j) = 1$$

e

$$l_{i,j}(x_k, y_l) = 0 \quad \text{per } k \neq i \quad \text{o} \quad l \neq j$$

Quest'ultimo risultato ci permette di affermare che il polinomio

$$(5.50) \quad P_{n,m}(x, y) = \sum_{i=0}^n \sum_{j=0}^m f(x_i, y_j) l_{i,j}(x, y) = \sum_{i=0}^n \sum_{j=0}^m f(x_i, y_j) l_i(x) \bar{l}_j(y)$$

soddisfa le condizioni di interpolazione richieste. Lasciamo al lettore come esercizio (non banale) la dimostrazione dell'unicità di tale polinomio.

Al polinomio  $P_{n,m}(x, y)$  possiamo anche dare una rappresentazione alle differenze divise (parziali) nella variabile  $x$

$$\begin{aligned} f[x; y] &= f(x, y) \\ f[x_0, x_1; y] &= \frac{f[x_1; y] - f[x_0; y]}{x_1 - x_0} \\ &\vdots \\ f[x_0, \dots, x_i; y] &= \frac{f[x_1, \dots, x_i; y] - f[x_0, \dots, x_{i-1}; y]}{x_i - x_0} \end{aligned}$$

e considerare la seguente formula di Newton

$$(5.51) \quad f(x, y) = \sum_{i=0}^n \omega_i(x) f[x_0, \dots, x_i; y] + \omega_{n+1}(x) f[x, x_0, \dots, x_n; y]$$

dove

$$\omega_0(x) = 1 \quad \text{e} \quad \omega_i(x) = \prod_{k=0}^{i-1} (x - x_k), \quad i \geq 1$$

D'altra parte, alla generica  $f[x_0, \dots, x_i; y]$ , considerata ora come funzione della sola variabile  $y$ , possiamo associare la corrispondente formula di Newton con nodi  $\{y_j\}$ :

$$f[x_0, \dots, x_i; y] = \sum_{j=0}^m \omega_j(y) f[x_0, \dots, x_i; y_0, \dots, y_j] + \omega_{m+1}(y) f[x_0, \dots, x_i; y, y_0, \dots, y_m]$$

Sostituendo quest'ultima espressione nella (5.51) otteniamo infine

$$f(x, y) = P_{n,m}(x, y) + E_{n,m}(x, y)$$

con

$$P_{n,m}(x, y) = \sum_{i=0}^n \sum_{j=0}^m \omega_i(x) \omega_j(y) f[x_0, \dots, x_i; y_0, \dots, y_j]$$

$$E_{n,m}(x, y) = \omega_{n+1}(x) f[x, x_0, \dots, x_n; y] + \omega_{m+1}(y) \sum_{i=0}^n \omega_i(x) f[x_0, \dots, x_i; y, y_0, \dots, y_m]$$

In modo perfettamente analogo possiamo estendere l'interpolazione con funzioni spline cubiche, che nel caso bidimensionale chiameremo *spline bicubiche*  $S_3(x, y)$ . La spline bicubica è una spline cubica sia nella variabile  $x$  ( $y$  fissata) che nella variabile  $y$  ( $x$  fissata).

Consideriamo il caso delle spline cubiche naturali. Scelti i nodi  $x_0 < x_1 < \dots < x_n$ , sappiamo che la spline cubica naturale  $S_3(x)$  interpolante una funzione  $g(x)$  può essere espressa nella forma

$$S_3(x) = \sum_{i=0}^n g(x_i) S_{3,i}(x)$$

dove con  $\{S_{3,i}\}$  denotiamo le spline cubiche naturali cardinali associate ai punti  $\{x_i\}$ . Nota questa rappresentazione di  $S_3(x)$ , la costruzione delle spline bicubiche naturali risulta pressoché immediata. Infatti, determinate le spline naturali cardinali associate ai nodi  $\{x_i\}$ ,  $\{S_{3,i}(x)\}$ , e ai nodi  $\{y_i\}$ ,  $\{\bar{S}_{3,j}(y)\}$ , la funzione

$$S_3(x, y) = \sum_{i=0}^n \sum_{j=0}^m f(x_i, y_j) S_{3,i}(x) \bar{S}_{3,j}(y)$$

rappresenta la spline bicubica naturale che interpola  $f(x, y)$  nei nodi  $\{(x_i, y_j)\}$ .

I risultati presentati possono venire facilmente estesi al caso di funzioni di più variabili.

## 5.9 Metodo dei minimi quadrati. Caso lineare discreto

La necessità di adottare il criterio di approssimazione dei minimi quadrati si presenta in non poche situazioni; per esempio nell'“approssimazione” di dati sperimentali, e in particolare in connessione con il problema della *regressione lineare* (vedere ad esempio [5.13]).

Supponiamo che in corrispondenza di  $(m+1)$  nodi  $\{x_i\}$ , non necessariamente tutti distinti, siano stati rilevati i valori  $\{y_i\}$ . Supponiamo inoltre di aver scelto  $n+1$  ( $\ll m+1$ ) funzioni base  $\{\varphi_0(x), \varphi_1(x), \dots, \varphi_n(x)\}$  e di voler approssimare il fenomeno in esame, rappresentato dai dati  $\{(x_i, y_i)\}$ , con una combinazione lineare (a coefficienti costanti) delle funzioni  $\{\varphi_k(x)\}$

$$(5.52) \quad f_n(x) = c_0 \varphi_0(x) + c_1 \varphi_1(x) + \dots + c_n \varphi_n(x)$$

individuata con il criterio dei minimi quadrati, ossia con i coefficienti  $\{c_k\}$  determinati in modo che il residuo

$$(5.53) \quad \varepsilon^2 = \sum_{i=0}^n \left[ y_i - \sum_{k=0}^n c_k \varphi_k(x_i) \right]^2$$

risulti minimo

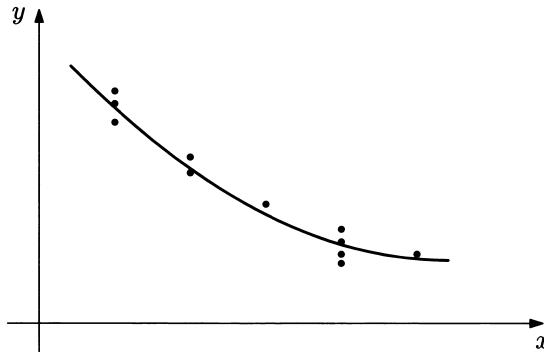


Figura 5.10

Perché il metodo dia risultati utili è importante scegliere bene il modello (5.52), ovvero le funzioni  $\{\varphi_k(x)\}$ . Tale scelta è in genere guidata dalle possibili informazioni note sul comportamento del fenomeno in esame, oppure semplicemente dalla distribuzione dei dati stessi. Una delle scelte più frequenti è certamente  $\varphi_k(x) = x^k$ , ma non sempre essa si rivela idonea.

Introducendo gli errori  $r_i = y_i - f_n(x_i)$  il problema può essere formulato nella forma seguente: determinare i coefficienti  $c_0, c_1, \dots, c_n$  definiti dal sistema

$$\begin{pmatrix} \varphi_0(x_0) & \varphi_1(x_0) & \dots & \varphi_n(x_0) \\ \varphi_0(x_1) & \varphi_1(x_1) & \dots & \varphi_n(x_1) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi_0(x_{m-1}) & \varphi_1(x_{m-1}) & \dots & \varphi_n(x_{m-1}) \\ \varphi_0(x_m) & \varphi_1(x_m) & \dots & \varphi_n(x_m) \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix} + \begin{pmatrix} r_0 \\ r_1 \\ \vdots \\ r_{m-1} \\ r_m \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{m-1} \\ y_m \end{pmatrix}$$

in modo che la quantità

$$\varepsilon^2 = \sum_{i=0}^m r_i^2$$

assuma il valore minimo. È ovvio che se  $m > n$  vi sono più equazioni che incognite, e quindi in generale l'eventuale soluzione ottima non potrà dare  $r_i = 0, i = 0, 1, \dots, m$ . L'aspetto essenziale della funzione  $f_n(x)$  (5.52) è la linearità nei parametri incogniti  $\{c_k\}$ ; per questo motivo il modello viene denominato *lineare*. Il modello

$$f_1(x) = c_0 + c_1 e^{c_2 x}$$

invece, è *non lineare*. Per il caso non lineare si veda ad esempio [6].

Al problema lineare dei minimi quadrati possiamo sempre dare la forma seguente: *dati una matrice  $A \in \mathbb{R}^{(m+1) \times (n+1)}$  ed un vettore  $b \in \mathbb{R}^{m+1}$ , determinare un vettore  $z = (c_0, \dots, c_n)^T \in \mathbb{R}^{n+1}$  tale che*

$$(5.54) \quad \|b - Az\|_2^2 = \text{minimo}$$

L'Analisi Matematica ci suggerisce di esaminare il sistema (quadrato)

$$\frac{\partial \varepsilon^2}{\partial c_k} = 0, \quad k = 0, 1, \dots, n$$

ossia

$$(5.55) \quad \begin{pmatrix} \sum_{i=0}^m \varphi_0^2(x_i) & \sum_{i=0}^m \varphi_0(x_i)\varphi_1(x_i) & \dots & \sum_{i=0}^m \varphi_0(x_i)\varphi_n(x_i) \\ \sum_{i=0}^m \varphi_1(x_i)\varphi_0(x_i) & \sum_{i=0}^m \varphi_1^2(x_i) & \dots & \sum_{i=0}^m \varphi_1(x_i)\varphi_n(x_i) \\ \dots & \dots & \dots & \dots \\ \sum_{i=0}^m \varphi_n(x_i)\varphi_0(x_i) & \sum_{i=0}^m \varphi_n(x_i)\varphi_1(x_i) & \dots & \sum_{i=0}^m \varphi_n^2(x_i) \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} \sum_{i=0}^m y_i \varphi_0(x_i) \\ \sum_{i=0}^m y_i \varphi_1(x_i) \\ \vdots \\ \sum_{i=0}^m y_i \varphi_n(x_i) \end{pmatrix}$$

che, come è ben noto, rappresenta una condizione necessaria per l'esistenza di un "punto"  $(c_0^*, c_1^*, \dots, c_n^*)^T$  di minimo per la funzione  $\varepsilon^2$ . Le equazioni del sistema (5.55), in seguito rappresentate nella forma  $Bz = d$ , vengono chiamate *equazioni normali* del problema. Osserviamo che gli elementi della matrice dei coefficienti  $B$  dipendono unicamente dalle funzioni base  $\{\varphi_k(x)\}$ , oltre che dai nodi  $\{x_i\}$ , e non dai valori  $\{y_i\}$ .

Un esame più attento della matrice  $B$  rivela la seguente struttura:

$$B = A^T A, \quad A = \begin{pmatrix} \varphi_0(x_0) & \varphi_1(x_0) & \dots & \varphi_n(x_0) \\ \varphi_0(x_1) & \varphi_1(x_1) & \dots & \varphi_n(x_1) \\ \dots & \dots & \dots & \dots \\ \varphi_0(x_{m-1}) & \varphi_1(x_{m-1}) & \dots & \varphi_n(x_{m-1}) \\ \varphi_0(x_m) & \varphi_1(x_m) & \dots & \varphi_n(x_m) \end{pmatrix}$$

Inoltre, per il termine noto  $d$  del sistema (5.55) abbiamo

$$d = A^T b, \quad b = (y_0, y_1, \dots, y_m)^T$$

Queste ultime espressioni ci permettono di rappresentare il sistema delle equazioni normali (5.55) nella forma più compatta

$$(5.56) \quad A^T A z = A^T b$$

e ci rivelano che il sistema è sempre simmetrico semidefinito positivo; anzi, se le colonne della matrice  $A$  sono linearmente indipendenti allora  $A^T A$  è nonsingolare e definita positiva. In questo secondo caso la soluzione del problema è unica e può essere posta nella forma

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

La matrice

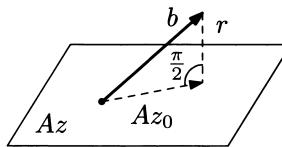
$$A^+ = (A^T A)^{-1} A^T$$

viene chiamata *pseudo-inversa* o *inversa generalizzata* di Moore-Penrose; quando  $\text{rango}(A) = n + 1 = m + 1$  abbiamo  $A^+ = A^{-1}$ .

Se  $z_0$  è soluzione del sistema delle equazioni normali, posto  $r = b - Az_0$  otteniamo la relazione

$$A^T r = A^T b - A^T A z_0 = 0$$

che caratterizza il residuo prodotto dalla (o da una) soluzione ottima: esso deve essere ortogonale alle colonne della matrice  $A$  e quindi allo spazio lineare generato dalle colonne stesse.



**Figura 5.11**

Sappiamo che ogni soluzione del problema (5.54) è anche soluzione del sistema (5.56). È pure vero il viceversa: ogni soluzione del sistema delle equazioni normali è soluzione del problema (5.54) ([6, §4.8.1.3]).

Il problema dei minimi quadrati (5.54) con  $m$  e  $n$  generici ammette sempre soluzione? In quali circostanze essa è unica? La risposta a queste due domande ci viene data dal teorema seguente.

**Teorema 5.10.** (vedi [5.12]). *Il problema  $\|b - Az\|_2^2 = \min$  ammette sempre soluzione. La soluzione è unica se e solo se le colonne della matrice  $A$  sono linearmente indipendenti. Se invece le colonne di  $A$  sono linearmente dipendenti allora il problema ha infinite soluzioni; tuttavia quella di lunghezza (euclidea) minima è unica.*

Limitiamoci al caso  $m \geq n$ , descriviamo ora brevemente due possibili metodi per risolvere il problema dei minimi quadrati. Per una visione più completa dei risultati conseguiti sull'argomento e dei metodi di risoluzione proposti consigliamo la lettura del testo [5.12].

**Metodo 1.** L'approccio più ovvio consiste nel costruire la matrice  $B = A^T A$  ed il vettore  $d = A^T b$ , e risolvere successivamente il sistema  $Bz = d$  con il metodo Choleski. L'implementazione numerica di questo metodo comporta tuttavia alcune difficoltà:

- (i) il sistema  $Bz = d$  può risultare mal condizionato anche quando il problema nella sua forma originale non lo è;
- (ii) la matrice  $B$  determinata numericamente può, causa gli errori dovuti alla precisione finita dei calcolatori, non risultare più definita positiva;
- (iii) se il rango di  $A$  è minore di  $n + 1$  allora la fattorizzazione  $LL^T$  non è unica.

Ciononostante questi inconvenienti possono essere limitati con opportuni accorgimenti. Il numero complessivo di operazioni aritmetiche richieste è dell'ordine di  $mn^2/2 + n^3/6$ .

**Metodo 2.** Ricordando la decomposizione  $QR$  di una matrice descritta nel paragrafo 4.5.1, e supponendo per semplicità le colonne di  $A$  linearmente indipendenti, trasformiamo la matrice  $A^{(0)} \equiv A$  ed il vettore  $b^{(0)} \equiv b$  con una successione di al più  $n+1$  riflettori elementari di Householder  $\{U_i\}$

$$A^{(i)} = U_i A^{(i-1)}, \quad b^{(i)} = U_i b^{(i-1)}, \quad i = 1, 2, \dots, n+1$$

sino ad ottenere una matrice finale  $A^{(n+1)}$  di forma

$$A^{(n+1)} = \begin{pmatrix} R \\ O \end{pmatrix}, \quad R \in \mathbb{R}^{(n+1) \times (n+1)}, \quad O \in \mathbb{R}^{(m-n) \times (n+1)}$$

con

$$R = \begin{pmatrix} r_{11} & r_{12} & \dots & r_{1,n+1} \\ 0 & r_{22} & \dots & r_{2,n+1} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & r_{n+1,n+1} \end{pmatrix}$$

La matrice triangolare superiore  $R$  è non singolare se e solo se le colonne di  $A$  sono linearmente indipendenti.

Le trasformazioni eseguite possono essere sintetizzate nell'espressione

$$A^{(n+1)} = Q^T A, \quad b^{(n+1)} = Q^T b$$

dove  $Q^T = U_{n+1} U_n \dots U_1$  è una matrice ortogonale.

A questo punto, dopo aver ricordato che  $\|Q^T y\|_2 = \|y\|_2$  e suddiviso il vettore  $b^{(n+1)}$  nella forma

$$b^{(n+1)} = \begin{pmatrix} b_1^{(n+1)} \\ b_2^{(n+1)} \end{pmatrix}, \quad b_1^{(n+1)} \in \mathbb{R}^{n+1}, \quad b_2^{(n+1)} \in \mathbb{R}^{m-n}$$

scriviamo

$$\|b - Az\|_2 = \|Q^T(b - Az)\|_2 = \|b^{(n+1)} - A^{(n+1)}z\|_2 = \left\| \begin{pmatrix} b_1^{(n+1)} - Rz \\ b_2^{(n+1)} \end{pmatrix} \right\|_2$$

La quantità  $\|b - Az\|_2$  viene pertanto minimizzata quando

$$(5.57) \quad Rz = b_1^{(n+1)}$$

e in questo caso il suo valore minimo è  $\|b_2^{(n+1)}\|_2$ . Il problema dei minimi quadrati è cosicché condotto alla soluzione di un sistema triangolare superiore, non singolare quando le colonne di  $A$  sono linearmente indipendenti.

Le operazioni aritmetiche necessarie per la determinazione della soluzione sono complessivamente dell'ordine di  $mn^2 - n^3/3$ . Ricordiamo inoltre che dal punto di vista della propagazione degli errori le trasformazioni di Householder risultano stabili.

Questo secondo approccio al problema rappresenta generalmente un buon metodo di risoluzione.

La descrizione del metodo 2. nel caso di matrici con colonne linearmente dipendenti, oppure “non sufficientemente indipendenti”, è ben più complessa e delicata. Ricordiamo tuttavia che in questa situazione un ottimo metodo di risoluzione è quello che si avvale della decomposizione SVD (della matrice  $A$ ) descritta nel paragrafo 4.5.3 ([4.9], [5.11]).

## 5.10 Cenni sul caso continuo del metodo dei minimi quadrati

Consideriamo lo spazio  $\mathbb{F} = C[a, b]$ ,  $-\infty < a < b < \infty$ , e definiamo il *prodotto interno* (o scalare)

$$(5.58) \quad \langle f, g \rangle = \int_a^b w(x)f(x)g(x) dx, \quad f, g \in \mathbb{F}$$

dove  $w(x)$  è una funzione non negativa in  $(a, b)$  (con al più un numero finito di zeri), e tale che  $\int_a^b w(x) dx$  esista. Ricordiamo che il prodotto interno  $\langle \cdot, \cdot \rangle$  è una funzione di  $\mathbb{F} \times \mathbb{F}$  in  $\mathbb{R}$  che gode delle seguenti proprietà:

1.  $\langle f, g \rangle = \langle g, f \rangle$
2.  $\langle \alpha f + \beta g, h \rangle = \alpha \langle f, h \rangle + \beta \langle g, h \rangle$ ,  $f, g, h \in \mathbb{F}$  e  $\alpha, \beta \in \mathbb{R}$
3.  $\langle f, f \rangle \geq 0$
4.  $\langle f, f \rangle = 0$  se e solo se  $f \equiv 0$ .

Inoltre,  $\langle f, f \rangle$  definisce in  $\mathbb{F}$  una norma. Per esempio nel caso della (5.58) abbiamo

$$\langle f, f \rangle = \|f\|_{2,w}^2$$

Due funzioni  $f, g \in \mathbb{F}$  sono ortogonali (tra di loro) se  $\langle f, g \rangle = 0$ . Le funzioni  $\varphi_0(x), \varphi_1(x), \dots, \varphi_n(x)$ , tutte in  $\mathbb{F}$ , costituiscono un *sistema ortogonale* se

$$\begin{aligned} \langle \varphi_i, \varphi_j \rangle &= 0, & i &\neq j \\ &\neq 0, & i &= j \end{aligned}$$

il sistema è invece detto *ortonormale* quando  $\langle \varphi_i, \varphi_j \rangle = \delta_{i,j}$ . Ovviamente, in un sistema ortogonale gli elementi  $\{\varphi_i\}$  sono linearmente indipendenti. Ricordiamo inoltre che da una base di un sottospazio  $\mathbb{F}_n \subset \mathbb{F}$  è sempre possibile, mediante il noto *processo di ortogonalizzazione* di Gram-Schmidt, dedurre un’altra base formata da elementi ortogonali tra di loro. In molte applicazioni le basi ortogonali assumono un ruolo importante per le notevoli semplificazioni di calcolo che esse comportano.

Osserviamo che qualunque sia la base  $\varphi_0(x), \varphi_1(x), \dots, \varphi_n(x)$  di  $\mathbb{F}_n$  abbiamo

$$(5.59) \quad \left\| \sum_{i=0}^n c_i \varphi_i(x) \right\|_{2,w} = 0$$

se e solo se  $c_0 = c_1 = \dots = c_n = 0$ . Infatti, per la definizione di norma la (5.59) risulta vera se e solo se  $\sum_{i=0}^n c_i \varphi_i(x) \equiv 0$ , cioè se e solo se  $c_0 = c_1 = \dots = c_n = 0$ . Quando la suddetta base è ortogonale

$$\left\| \sum_{i=0}^n c_i \varphi_i(x) \right\|_{2,w}^2 = \sum_{i=0}^n c_i^2 \|\varphi_i\|_{2,w}^2$$

se poi è anche ortonormale,

$$\left\| \sum_{i=0}^n c_i \varphi_i(x) \right\|_{2,w}^2 = \sum_{i=0}^n c_i^2$$

Supponiamo di aver scelto lo spazio  $\mathbb{F}_n$  delle approssimazioni, per esempio  $\mathbb{F}_n \equiv \mathbb{P}_n$ , e una sua base  $\{\varphi_i\}$ . Generalizzando il caso discreto che abbiamo esaminato nel paragrafo precedente, ci proponiamo di risolvere il seguente problema: *data una funzione  $f \in \mathbb{F}$  (e  $f \notin \mathbb{F}_n$ ), determinare l'elemento (se unico)  $f_n^* \in \mathbb{F}_n$  che rende minimo il “residuo”*

$$\varepsilon^2 = \varepsilon^2(c_0, c_1, \dots, c_n) = \|f_n - f\|_{2,w}^2, \quad f_n = \sum_{i=0}^n c_i \varphi_i$$

Procedendo come nel caso discreto, osserviamo che una condizione necessaria (che risulterà poi anche sufficiente) per l'esistenza di una soluzione ottima  $\{c_0^*, c_1^*, \dots, c_n^*\}$  è la seguente:

$$(5.60) \quad \frac{\partial \varepsilon^2}{\partial c_k}(c_0^*, \dots, c_n^*) = 2 \left\langle \sum_{i=0}^n c_i^* \varphi_i - f, \varphi_k \right\rangle = 0, \quad k = 0, 1, \dots, n$$

ovvero

$$(5.61) \quad \begin{pmatrix} \langle \varphi_0, \varphi_0 \rangle & \langle \varphi_1, \varphi_0 \rangle & \dots & \langle \varphi_n, \varphi_0 \rangle \\ \langle \varphi_0, \varphi_1 \rangle & \langle \varphi_1, \varphi_1 \rangle & \dots & \langle \varphi_n, \varphi_1 \rangle \\ \dots & \dots & \dots & \dots \\ \langle \varphi_0, \varphi_n \rangle & \langle \varphi_1, \varphi_n \rangle & \dots & \langle \varphi_n, \varphi_n \rangle \end{pmatrix} \begin{pmatrix} c_0^* \\ c_1^* \\ \vdots \\ c_n^* \end{pmatrix} = \begin{pmatrix} \langle f, \varphi_0 \rangle \\ \langle f, \varphi_1 \rangle \\ \vdots \\ \langle f, \varphi_n \rangle \end{pmatrix}$$

Quest'ultimo viene denominato *sistema delle equazioni normali*.

Osserviamo subito che la soluzione della (5.61) è unica. Infatti, se per assurdo non lo fosse, la matrice dei coefficienti dovrebbe risultare singolare, e il sistema omogeneo associato

$$\sum_{i=0}^n c_i \langle \varphi_i, \varphi_k \rangle = 0 \quad k = 0, 1, \dots, n$$

dovrebbe ammettere soluzioni non nulle. Ma quest'ultima eventualità comporterebbe la seguente diseguaglianza assurda:

$$0 < \left\| \sum_{i=0}^n c_i \varphi_i \right\|_{2,w}^2 = \left\langle \sum_{i=0}^n c_i \varphi_i, \sum_{k=0}^n c_k \varphi_k \right\rangle = \sum_{k=0}^n \left[ \sum_{i=0}^n c_i \langle \varphi_i, \varphi_k \rangle \right] c_k = \sum_{k=0}^n 0 \cdot c_k = 0$$

È poi possibile verificare che la soluzione è effettivamente un punto di minimo.

La relazione (5.60) ci dice che la funzione residuo  $(\sum_{i=0}^n c_i^* \varphi_i - f)$  prodotta dalla soluzione ottima risulta ortogonale a tutte le funzioni base  $\{\varphi_k\}$ . Quando la base scelta è ortogonale, il sistema (5.61) assume la forma diagonale e la sua risoluzione è immediata:

$$c_i^* = \frac{\langle f, \varphi_i \rangle}{\langle \varphi_i, \varphi_i \rangle}, \quad i = 0, 1, \dots, n$$

ogni singolo coefficiente  $c_i^*$  dipende unicamente dalla funzione  $\varphi_i(x)$ .

Inoltre, quando consideriamo il nuovo sottospazio  $\mathbb{F}_{n+1}$  generato dalla base  $\varphi_0, \varphi_1, \dots, \varphi_n, \varphi_{n+1}$ , la soluzione ottima  $f_{n+1}^* \in \mathbb{F}_{n+1}$  è legata a quella  $(f_n^*)$  di  $\mathbb{F}_n$  dalla relazione

$$f_{n+1}^* = f_n^* + c_{n+1}^* \varphi_{n+1}, \quad c_{n+1}^* = \frac{\langle f, \varphi_{n+1} \rangle}{\langle \varphi_{n+1}, \varphi_{n+1} \rangle}$$

I coefficienti  $\{c_i^*\}$  vengono spesso chiamati *coefficienti di Fourier generalizzati*.

È possibile dimostrare che quando  $\mathbb{F}_n \equiv \mathbb{P}_n$  la convergenza in norma di  $f_n^*$  a  $f$  è assicurata, ossia

$$\lim_{n \rightarrow \infty} \|f_n^* - f\|_{2,w} = 0$$

inoltre, quando il sistema  $\{\varphi_i\}$  è ortonormale,

$$\sum_{i=0}^{\infty} c_i^* = \|f\|_{2,w}^2 \quad \text{e} \quad \|f_n^* - f\|_{2,w}^2 = \sum_{i=n+1}^{\infty} (c_i^*)^2$$

La convergenza uniforme invece non è garantita; tuttavia, quando in (5.58) scegliamo  $w(x) \equiv 1$  oppure  $w(x) = (x-a)^{-\frac{1}{2}}(b-x)^{-\frac{1}{2}}$ , e  $f(x) \in C^2[a, b] \subset \mathbb{F}$ , abbiamo ([1, §3.4])

$$\lim_{n \rightarrow \infty} \|f_n^* - f\|_{\infty} = 0$$

Pertanto nelle applicazioni, dopo aver scelto lo spazio  $\mathbb{F}_n$ , occorre prendere in (5.58) una funzione peso  $w(x)$  “idonea”. Infatti quest’ultima sarà poi la diretta responsabile della velocità di convergenza (in norma) di  $f_n^*$  a  $f$ .

Supponiamo di aver scelto  $\mathbb{F}_n \equiv \mathbb{P}_n$ . La base ortogonale necessaria per individuare i  $\{c_i^*\}$  è costituita dai polinomi  $P_0(x), P_1(x), \dots, P_n(x)$  ortogonali in  $(a, b)$  rispetto alla funzione peso  $w(x)$ . Come vedremo nel paragrafo 7.3 tali polinomi sono univocamente definiti, a meno di fattori di normalizzazione, dall’intervallo  $(a, b)$  e da  $w(x)$ . Costruita la base, i coefficienti  $\{c_i^*\}$  vengono infine determinati calcolando gli integrali  $\langle f, P_i \rangle$ <sup>(†)</sup> con formule di quadratura. I polinomi ortogonali e le formule di quadratura per il calcolo approssimato di integrali saranno argomento del capitolo 7.

---

(†) La costante  $\langle P_i, P_i \rangle$  dipende dalla normalizzazione scelta; se il sistema è ortonormale,  $\langle P_i, P_i \rangle = 1$ .

## 5.11 Derivazione numerica

Data una funzione  $f(x)$ , nota o incognita, supponiamo di voler approssimare la sua derivata in un punto  $x$  mediante una combinazione lineare di alcuni suoi valori  $\{f(x_i)\}$ . Un'approssimazione di questo tipo può essere certamente costruita prendendo dapprima, per esempio, il polinomio  $P_n(x)$  che interpola la funzione  $f(x)$  nell'insieme di nodi  $\{x_i\}$  e considerando poi il valore  $P'_n(x) \simeq f'(x)$ .

Sceglieremo  $n+1$  punti distinti  $x_0, x_1, \dots, x_n$  nell'intervallo  $[a, b]$  di interesse, e consideriamo il polinomio di interpolazione, nella forma di Newton per esempio,

$$P_n(x) = f(x_0) + (x - x_0)f[x_0, x_1] + \dots + (x - x_0)(x - x_1)\dots(x - x_{n-1})f[x_0, x_1, \dots, x_n]$$

Per l'errore  $E_n(x) = f(x) - P_n(x)$  abbiamo (vedi pagina 138) la seguente rappresentazione

$$(5.62) \quad E_n(x) = \omega_{n+1}(x)f[x_0, x_1, \dots, x_n, x]$$

dove  $\omega_{n+1}(x) = \prod_{i=0}^n (x - x_i)$ . Dalla relazione

$$f'(x) = P'_n(x) + \omega'_{n+1}(x)f[x_0, x_1, \dots, x_n, x] + \omega_{n+1}(x) \frac{d}{dx} f[x_0, x_1, \dots, x_n, x]$$

con(<sup>†</sup>)

$$\begin{aligned} \frac{d}{dx} f[x_0, x_1, \dots, x_n, x] &= \lim_{h \rightarrow 0} \frac{f[x_0, \dots, x_n, x+h] - f[x_0, \dots, x_n, x]}{h} \\ &= \lim_{h \rightarrow 0} \frac{f[x_0, \dots, x_n, x+h] - f[x, x_0, \dots, x_n]}{h} \\ &= \lim_{h \rightarrow 0} f[x, x_0, \dots, x_n, x+h] \\ &= f[x, x_0, \dots, x_n, x] = f[x_0, \dots, x_n, x, x] \end{aligned}$$

deduciamo l'espressione

$$f'(x) - P'_n(x) = \omega'_{n+1}(x)f[x_0, \dots, x_n, x] + \omega_{n+1}(x)f[x_0, \dots, x_n, x, x]$$

dalla quale, supponendo  $f \in C^{n+2}[a, b]$  e ricordando il teorema 5.5, segue

$$f'(x) - P'_n(x) = \omega'_{n+1}(x) \frac{f^{(n+1)}(\xi)}{(n+1)!} + \omega_{n+1}(x) \frac{f^{(n+2)}(\eta)}{(n+2)!}$$

Generalizzando quanto esposto sopra possiamo approssimare  $f^{(k)}(x)$ ,  $k \geq 1$ , con  $P_n^{(k)}(x)$ , purché  $n \geq k$ .

---

(†) Ricordiamo l'invarianza delle differenze divise a permutazioni degli argomenti.

Come esercizio, al lettore lasciamo il compito di verificare la validità delle formule seguenti:

$$(5.63) \quad \begin{cases} f'(x_0 - h) = \frac{1}{2h}[-f(x_0 + h) + 4f(x_0) - 3f(x_0 - h)] + \frac{h^2}{3}f^{(3)}(\xi_1) \\ f'(x_0) = \frac{1}{2h}[f(x_0 + h) - f(x_0 - h)] - \frac{h^2}{6}f^{(3)}(\xi_2) \\ f'(x_0 + h) = \frac{1}{2h}[3f(x_0 + h) - 4f(x_0) + f(x_0 - h)] + \frac{h^2}{3}f^{(3)}(\xi_3) \\ f''(x_0 - h) = \frac{1}{h^2}[f(x_0 + h) - 2f(x_0) + f(x_0 - h)] - hf^{(3)}(\xi_1) + \frac{h^2}{6}f^{(4)}(\eta_1) \\ f''(x_0) = \frac{1}{h^2}[f(x_0 + h) - 2f(x_0) + f(x_0 - h)] - \frac{h^2}{12}f^{(4)}(\eta_2) \\ f''(x_0 + h) = \frac{1}{h^2}[f(x_0 + h) - 2f(x_0) + f(x_0 - h)] + hf^{(3)}(\xi_3) + \frac{h^2}{6}f^{(4)}(\eta_3) \end{cases}$$

Le formule costruibili con il suddetto procedimento vengono di solito utilizzate *solo* per la discretizzazione di equazioni differenziali (vedere i capitoli 8 e 9). Il comportamento eccessivamente oscillatorio del polinomio di interpolazione  $P_n(x)$ , soprattutto al crescere di  $n$ , non rende quest'ultimo strumento proponibile per l'approssimazione di  $f'(x)$  in tutto un intervallo.

Quando l'obiettivo principale è l'approssimazione di  $f'(x)$ , oppure  $f''(x)$ , in un intervallo  $[a, b]$  possiamo utilizzare le spline cubiche; il teorema 5.9 ci assicura la convergenza uniforme.

Prima di concludere ritorniamo alle formule (5.63), e in particolare analizziamo la propagazione delle perturbazioni nella formula

$$f'(x_0) = \frac{1}{2h}[f(x_0 + h) - f(x_0 - h)] - \frac{h^2}{6}f^{(3)}(\xi)$$

A tale scopo supponiamo di avere a nostra disposizione “solamente” i valori perturbati  $\bar{f}(x_0 + h) = f(x_0 + h) + \varepsilon_1$  e  $\bar{f}(x_0 - h) = f(x_0 - h) + \varepsilon_2$ , così che l'approssimazione da noi determinata risulta<sup>(†)</sup>

$$\bar{P}'_2(x_0) = \frac{1}{2h}[\bar{f}(x_0 + h) - \bar{f}(x_0 - h)]$$

con un errore globale

$$f'(x_0) - \bar{P}'_2(x_0) = \frac{\varepsilon_2 - \varepsilon_1}{2h} - \frac{h^2}{6}f^{(3)}(\xi)$$

Quando  $h \rightarrow 0$  l'errore di discretizzazione (o di troncamento) tende a zero, ma quello provocato dalle perturbazioni  $\varepsilon_i$  va a infinito. L'errore globale ha un comportamento di

---

(†) Trascurando gli errori dovuti alla rappresentazione di  $h$  e alla corrispondente divisione

tipo convesso: decresce al diminuire di  $h$  fintantoché  $h$  non raggiunge il punto di minimo  $h_{\text{opt}}$ , oltre il quale l'errore globale cresce (come  $h^{-1}$  per  $h \rightarrow 0$ ).

## Bibliografia

- [5.1] P. J. Davis, *Interpolation and approximation*, Blaisdell, Waltham, Mass., 1963.
- [5.2] I. P. Natanson, *Constructive function theory*, Vol. III, Frederick Ungar, New York, 1965.
- [5.3] J. W. Cooley, J. W. Tukey, *An algorithm for the machine calculation of complex Fourier series*, Math. Comput., v. 19, 1965, pp. 297–301.
- [5.4] J. R. Rice, *The approximation of functions: linear theory*, vol. I, Addison-Wesley, Reading, Mass., 1964.
- [5.5] G.G. Lorentz, *Approximation of functions*, Holt Rinehart & Winston, New York, 1966.
- [5.6] J. Ahlberg, E. Nilson, J. Walsh, *The theory of splines and their applications*, Academic Press, New York, 1967.
- [5.7] E. W. Cheney, *Introduction to approximation theory*, McGraw-Hill, New York, 1966.
- [5.8] J. R. Rice, *The approximation of functions: nonlinear and multivariate theory*, vol. II, Addison-Wesley, Reading, Mass., 1969.
- [5.9] T.J. Rivlin, *An Introduction to the approximation of functions*, Blaisdell, Waltham, Mass. 1969.
- [5.10] M. H. Schultz, *Spline analysis*, Prentice-Hall, Englewood Cliffs, N.J., 1973.
- [5.11] C. L. Lawson, R. J. Hanson, *Solving least squares problems*, Prentice-Hall, Englewood Cliffs, N.J., 1974.
- [5.12] G. A. F. Seber, *Linear regression analysis*, John Wiley & Sons, New York, 1977.
- [5.13] C. de Boor, *A Practical guide to splines*, Springer-Verlag, New York, 1978.
- [5.14] G. A. Watson, *Approximation theory and numerical methods*, John Wiley & Sons, Chichester, 1980.
- [5.15] M. J. D. Powell, *Approximation theory and methods*, Cambridge University Press, Cambridge, 1981.
- [5.16] D. F. Elliot, K. R. Rao, *Fast transforms, algorithms, analysis, applications*, Academic Press, New York, 1982.
- [5.17] M. Pickering, *An introduction to fast Fourier transform methods for partial differential equations, with applications*, Research Studies Press, John Wiley & Sons, New York, 1986.
- [5.18] R. Bartels, J. Beatty, B. Barsky, *An introduction to splines for use in computer graphics and geometric modeling*, Morgan Kaufmann, Los Altos, 1987.
- [5.19] E. O. Brigham, *The Fast Fourier Transform and its applications*, Prentice-Hall, Englewood Cliffs, New Jersey, 1988.

## Esercizi proposti

**5.1.** Costruire la tabella delle differenze finite progressive associata ai dati seguenti:

$x_i$	$f(x_i)$
1.8	1.341641
1.9	1.378405
2.0	1.414214
2.1	1.449138
2.2	1.483240

e, successivamente, la corrispondente formula di interpolazione di Newton. Usare tale formula per approssimare la funzione  $f(x) = \sqrt{x}$  nell'intervallo  $[1.8, 2.2]$ . Esaminare il comportamento dell'errore nei punti  $x = 1.85, 1.95, 2.05, 2.15$ . Utilizzare infine il polinomio di Newton precedente per determinare un'approssimazione dell'integrale

$$\int_{1.8}^{2.2} \sqrt{x} \, dx$$

**5.2.** Siano assegnati i valori

$$\begin{cases} f(0) = 0 & f(1) = 1 & f(2) = 1 \\ f'(0) = 0 & f'(1) = 1 \end{cases}$$

Costruire la tabella delle differenze divise e il corrispondente polinomio di interpolazione di Newton. Verificare che tale polinomio soddisfa effettivamente le condizioni imposte.

**5.3.** Costruire il polinomio  $P_m(x)$  di grado minimo che soddisfa le seguenti condizioni:

$$\begin{aligned} P_m(0) &= 1 \\ P'_m(0) &= 0 \\ P''_m(0) &= 2 \\ P_m(-1) &= 2 \\ P_m(1) &= 0 \\ P'_m(1) &= 2 \end{aligned}$$

Successivamente determinare il nuovo polinomio che, oltre a soddisfare le condizioni precedenti, assume il valore  $P_m(2) = 1$ .

**5.4.** Disegnare (utilizzando un plotter) i grafici relativi ai polinomi di interpolazione di grado  $n - 1$  individuati dalla funzione  $f(x) = 1/(25x^2 + 1)$ ,  $-1 \leq x \leq 1$ , e dai seguenti insiemi di nodi:

- (i)  $x_i = -1 + \frac{2i}{(n-1)}$        $i = 0, 1, \dots, n-1$ ,       $n = 5, 10, 20$
- (ii)  $x_i = \cos\left(\frac{2i+1}{2n}\pi\right)$        $i = 0, 1, \dots, n-1$ ,       $n = 5, 10, 20$

Commentare i risultati.

**5.5.** Dimostrare il teorema 5.7 e il relativo corollario.

**5.6.** Siano dati  $n + 1$  nodi distinti  $x_0, x_1, \dots, x_n$ . Consideriamo la funzione

$$f_n(x) = \sum_{j=0}^n c_j e^{jx}$$

Determinare i coefficienti  $\{c_j\}$  in modo che  $f_n(x_i) = y_i$ ,  $i = 0, 1, \dots, n$ , dove gli  $\{y_i\}$  sono  $n + 1$  numeri assegnati, e dimostrare che la soluzione  $\{c_j\}$  è unica.

**5.7.** Implementare l'algoritmo FFT di pagina 152 in un linguaggio di programmazione.

**5.8.** Costruire l'algoritmo FFT che determini i coefficienti  $a_j$  e  $b_j$  definiti a pagina 153.

**5.9.** Siano dati  $M$  punti  $\{(x_i, y_i), i = 1, \dots, M\}$  del piano, con le ascisse ordinate come segue:  $x_1 < x_2 < \dots < x_M$ . Costruire un algoritmo che, lette le coordinate dei predetti punti, e letta un'ascissa  $t$ , verifichi se tale ascissa risulta  $\geq x_1$  e  $\leq x_M$ , e, in caso affermativo, determini il valore che la poligonale passante per i punti assegnati assume in  $t$ .

**5.10.** Generalizzare l'esercizio precedente come segue. Data una partizione  $a = x_1 < x_2 < \dots < x_N = b$  di  $[a, b]$ , e, in ciascun intervallo  $[x_i, x_{i+1}]$ , assegnate  $d + 1$  ascisse

$$x_i = x_{i1} < x_{i2} < \dots < x_{i,d+1} = x_{i+1}$$

supponiamo di conoscere le corrispondenti ordinate  $y_{ij}$ ,  $j = 1, \dots, d + 1$ . Utilizzando gli algoritmi Difdiv e Interp di pagina 141, costruire un algoritmo che, letta un'ascissa  $a \leq t \leq b$ , determini l'ordinata corrispondente che la funzione polinomiale a tratti, di grado locale  $d$ , associata alla partizione  $\{x_i\}$  assume.

**5.11.** Costruire le spline cubiche  $S_3(x)$ , naturali prima e con condizioni (5.45) poi, che interpolano la funzione  $f(x) = (1-x^2)^{9/2} + x^2$  nei nodi  $x_i = -1 + i2/n$ ,  $i = 0, 1, \dots, n$ ,  $n = 2^k$ ,  $k = 2, 3, \dots, 6$ . Osservare le approssimazioni fornite dalle due spline e commentare i risultati.

Successivamente approssimare in  $[-1, 1]$   $f'(x)$  con  $S'_3(x)$ .

**5.12.** Ripetere l'esercizio 5.4 utilizzando come funzioni interpolanti delle spline cubiche e confrontare i risultati.

**5.13.** Disegnare la spline cubica naturale che interpola la funzione  $f(x) = (\ln(x + 0.5))/(x + 0.5)$ ,  $0 \leq x \leq 2$ , nei nodi  $x_i = 2i/(n-1)$ ,  $i = 0, 1, \dots, n-1$ ,  $n = 5, 10, 20$ , e commentare i risultati.

**5.14.** Con  $S_3(f; x)$  denotiamo la spline cubica naturale di interpolazione associata ai nodi  $x_0 < x_1 < \dots < x_n$  e alla funzione  $f(x)$ . Dimostrare che l'operatore  $f \rightarrow S_3$  è lineare, ossia

$$S_3(\alpha f + \beta g; x) = \alpha S_3(f; x) + \beta S_3(g; x)$$

**5.15.** Consideriamo lo spazio delle spline cubiche naturali associate ai nodi  $x_0 < x_1 < \dots < x_n$  e tali che  $S_3(x_i) = y_i$ ,  $i = 0, 1, \dots, n$ . Verificare che al variare di  $y = (y_0, y_1, \dots, y_n)^T$  in  $\mathbb{R}^{n+1}$   $S_3(x)$  descrive uno spazio lineare. Qual è la dimensione di tale spazio? Proporre una base.

**5.16.** Data una partizione  $a = x_0 < x_1 < \dots < x_n = b$  dell'intervallo  $[a, b]$ , introduciamo sei nodi ausiliari  $x_{-3} < x_{-2} < x_{-1} < x_0$ ,  $x_{n+3} > x_{n+2} > x_{n+1} > x_n$ . Definiamo poi  $n + 3$  funzioni spline cubiche  $\{B_{3,j}, j = -1, 0, 1, \dots, n + 1\}$  che soddisfano le seguenti condizioni:

$$\begin{cases} B_{3,j}(x) = 0 & \text{per } x \leq x_{j-2} \text{ e } x \geq x_{j+2} \\ B_{3,j}(x_j) = 1 & \\ B'_{3,j}(x) = B''_{3,j}(x) = 0 & \text{per } x = x_{j-2} \text{ e } x = x_{j+2} \end{cases}$$

Dimostrare l'esistenza e unicità di tali funzioni e verificare che esse costituiscono una base per lo spazio lineare delle spline cubiche associate ai nodi  $x_0 < x_1 < \dots < x_n$ .

Le funzioni  $B_{3,j}(x)$  vengono denominate  $B$ -splime cubiche. Disegnare  $B_{3,j}(x)$ .

**5.17.** Supponiamo di aver effettuato i seguenti rilevamenti

$t$ (h)	0	1	2	3	4	5
$H$ (m)	0.5	0.8	0.7	0.3	0.1	0.4

Determinare i parametri  $h_0, a_1, a_2$  del modello

$$H(t) = h_0 + a_1 \sin \frac{2\pi t}{6} + a_2 \cos \frac{2\pi t}{6}$$

con il metodo dei minimi quadrati.

**5.18.** Supponiamo di aver effettuato le seguenti misurazioni

$$\begin{array}{ll} \text{in } x_0 = 1 & y_0 = 1.1 \\ \text{in } x_1 = 1 & y_1 = 1.05 \\ \text{in } x_2 = 1 & y_2 = 1.12 \\ \text{in } x_3 = 1 & y_3 = 1.07 \\ \text{in } x_4 = 1 & y_4 = 2 \\ \text{in } x_5 = 2 & y_5 = 2.1 \\ \text{in } x_6 = 3 & y_6 = 3.05 \\ \text{in } x_7 = 4 & y_7 = 3.7 \\ \text{in } x_8 = 5 & y_8 = 4.0 \\ \text{in } x_9 = 5 & y_9 = 4.8 \\ \text{in } x_{10} = 5 & y_{10} = 5.2 \end{array}$$

Determinare con il metodo dei minimi quadrati i coefficienti della retta  $y = ax + b$ .

**5.19.** Richiamando il paragrafo 7.3, determinare il polinomio algebrico di grado  $N$  che meglio approssima la funzione  $e^x$  nell'intervallo  $[0, 1]$ , secondo il criterio dei minimi quadrati.

**5.20.** Ricavare le formule (5.63).



# Capitolo 6

## Equazioni non lineari

### 6.1 Preliminari

Il problema che ci accingiamo ad affrontare in questo capitolo è la ricerca di soluzioni di equazioni (e di sistemi di equazioni) non lineari

$$f(x) = 0$$

La funzione  $f(x)$  spesso è data mediante un'espressione analitica, altre volte come soluzione di un altro problema (per esempio di un'equazione differenziale); oppure essa è valutabile, per ogni valore assegnato dell'argomento  $x$ , tramite un algoritmo (routine). Purtroppo, come vedremo nei prossimi paragrafi, la non linearità della  $f(x)$  introduce particolari difficoltà, soprattutto in relazione al problema della convergenza dei metodi di risoluzione proposti, necessariamente di tipo iterativo.

Incominciamo con l'esame del caso di una singola equazione, e prima di affrontare la costruzione di metodi numerici analizziamo il condizionamento del problema almeno quando  $f(x)$  è un polinomio algebrico

$$f(x) = x^n + a_1x^{n-1} + \dots a_{n-1}x + a_n$$

Sia  $r = r(a_1, a_2, \dots, a_n)$  una radice (reale o complessa) semplice dell'equazione  $f(x) = 0$ ; ovvero, dati i coefficienti  $a_1, a_2, \dots, a_n$ , sia  $f(r) = f(a_1, a_2, \dots, a_n; r(a_1, a_2, \dots, a_n)) \equiv 0$ . Ricordando il teorema sulla derivazione delle funzioni composte, e quindi derivando l'identità  $g(a_1, a_2, \dots, a_n) = f(a_1, a_2, \dots, a_n; r(a_1, a_2, \dots, a_n)) = 0$  rispetto alla variabile  $a_k$ , otteniamo la relazione

$$\frac{\partial g}{\partial a_k} = f'(r) \frac{\partial r}{\partial a_k} + r^{n-k} = 0, \quad k = 1, 2, \dots n$$

dalla quale, supponendo  $f'(r) \neq 0$  e sostituendo le derivate parziali con rapporti incrementali, possiamo dedurre l'uguaglianza approssimata

$$\frac{\Delta r}{r} \simeq -a_k \frac{r^{n-k-1}}{f'(r)} \frac{\Delta a_k}{a_k}$$

Quest'ultima relazione rappresenta la variazione relativa di  $r$  provocata dalla perturbazione (relativa)  $\Delta a_k/a_k$  presente nel solo coefficiente  $a_k$  (supponendo cioè esatti tutti gli altri coefficienti  $a_i, i \neq k$ ); come fattore di amplificazione dell'errore consideriamo la quantità  $A_k = |a_k \frac{r^{n-k-1}}{f'(r)}|$ . Un esame attento dell'espressione di  $A_k$  ci permette di osservare che quando  $|f'(r)|$  è piccola rispetto alla quantità  $|a_k r^{n-k-1}|$ , il problema del calcolo della radice  $r$  è mal condizionato. Una situazione di questo tipo si ha, per esempio, quando la distanza (relativa) di due radici è piccolissima, cioè quando  $r$  è “quasi” una radice doppia.

Un esempio molto noto [6.1] che pone in evidenza il fenomeno del mal condizionamento è il seguente:

$$(x - 1)(x - 2) \dots (x - 20) = x^{20} - 210x^{19} + \dots + 20! = 0$$

Supponiamo di introdurre in  $a_1 = 210$  una perturbazione  $\Delta a_1 = 2^{-23}$ . Mentre gli errori  $\Delta r_i$  associati alle radici  $r_i = 1, 2, \dots, 8$  sono molto piccoli, troviamo, per esempio,  $13.99 \pm i2.52$  invece di  $r = 14$  e  $15$ ,  $16.73 \pm i2.81$  invece di  $16$  e  $17$ , e  $20.85$  al posto di  $20$ . La previsione di perdita di cifre, nel calcolo della radice  $r = 16$  per esempio, fatta dalla predetta espressione di  $A_k$  risulta

$$A_1 = 210 \frac{16^{18}}{4! 15!} = 3.2 \times 10^{10}$$

Ciò significa che la determinazione della radice  $r = 16$  comporta una perdita di circa 10 cifre significative. Pertanto, se  $a_1$  viene dato per esempio con 16 cifre significative (mentre tutti gli altri coefficienti si suppongono noti esattamente), lo zero  $r = 16$  può essere calcolato al più con circa 6 cifre significative. E questo supponendo di operare con precisione di calcolo infinita!

Queste prime semplici osservazioni ci danno un'indicazione delle variazioni degli zeri causate da perturbazioni presenti nei coefficienti  $\{a_i\}$  della rappresentazione standard

$$x^n + a_1 x^{n-1} + \dots + a_{n-1} x + a_n$$

Tuttavia, nelle applicazioni spesso i polinomi risultano facilmente rappresentabili anche in altre forme, più convenienti dal punto di vista del condizionamento; per esempio come combinazioni lineari di *polinomi ortogonali*  $\{p_0(x), p_1(x), \dots, p_n(x), \dots\}$ <sup>(†)</sup>

$$c_0 p_n(x) + c_1 p_{n-1}(x) + \dots + c_n p_0(x)$$

---

(†) Vedere il paragrafo 7.3.

In questo caso gli zeri del polinomio sono funzione dei dati  $\{c_0, c_1, \dots, c_n\}$ . Mentre ogni singolo zero può essere molto sensibile alle perturbazioni dei coefficienti  $\{a_i\}$ , può non esserlo, o esserlo in misura minore, a quelle dei  $\{c_i\}$ .

Le radici di un'equazione non lineare  $f(x) = 0$  non possono in generale venire espresse in “forma chiusa”; e anche quando ciò è possibile, la corrispondente espressione può risultare troppo complessa, o comunque non competitiva con altri modi di procedere. Pertanto, per risolvere numericamente equazioni non lineari siamo costretti a ricorrere a dei metodi approssimati. Questi ultimi sono necessariamente di tipo iterativo; partendo da una o più approssimazioni iniziali essi producono una successione  $x_0, x_1, x_2, \dots$  convergente, quando le ipotesi richieste sono soddisfatte, alla radice incognita.

Nella prima parte di questo capitolo affrontiamo la costruzione di metodi per il calcolo di sole radici reali di una generica equazione non lineare. Gli stessi vengono successivamente generalizzati ai sistemi di equazioni non lineari. Infine, nell'ultima parte, esamineremo brevemente il problema dell'approssimazione di zeri reali e complessi di polinomi a coefficienti reali.

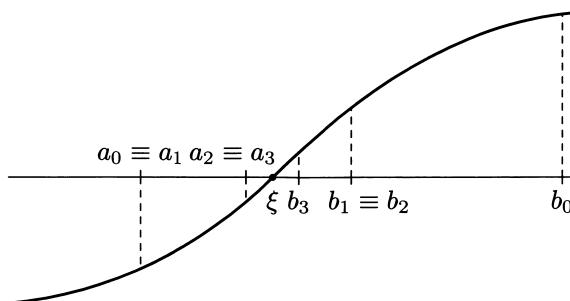
## 6.2 Radici reali di equazioni non lineari

### 6.2.1 Metodo di bisezione

Supponiamo che la funzione  $f(x)$  sia continua nell'intervallo  $[a_0, b_0]$  e che  $f(a_0)f(b_0) < 0$ . Queste ipotesi ci garantiscono l'esistenza di una radice nell'intervallo suddetto. Utilizzando tali informazioni ci proponiamo di costruire una successione di intervalli encapsulati  $(a_0, b_0) \supset (a_1, b_1) \supset (a_2, b_2) \supset \dots$ , tutti contenenti una radice dell'equazione  $f(x) = 0$ , con  $(b_n - a_n) \rightarrow 0$  per  $n \rightarrow \infty$ .

Per esempio, sia  $f(a_0) < 0$  e  $f(b_0) > 0$ . Gli intervalli successivi  $(a_n, b_n)$ ,  $n = 1, 2, \dots$ , vengono individuati con la strategia seguente (vedi figura 6.1). Dato  $(a_{n-1}, b_{n-1})$ , determiniamo il punto medio

$$m_n = \frac{1}{2}(a_{n-1} + b_{n-1})$$



**Figura 6.1**

Supponiamo  $f(m_n) \neq 0$ , altrimenti  $m_n$  è radice. Esaminiamo il segno di  $f(m_n)$  e poniamo

$$(a_n, b_n) = \begin{cases} (m_n, b_{n-1}) & \text{se } f(m_n) < 0 \\ (a_{n-1}, m_n) & \text{se } f(m_n) > 0 \end{cases}$$

Dopo  $n$  passi giungiamo all'intervallo  $(a_n, b_n)$ , contenente la radice  $\xi$  cercata, di ampiezza

$$b_n - a_n = \frac{b_{n-1} - a_{n-1}}{2} = \frac{b_{n-2} - a_{n-2}}{2^2} = \dots = \frac{b_0 - a_0}{2^n}$$

Come stima della radice  $\xi$  prendiamo

$$m_{n+1} = \frac{1}{2}(a_n + b_n)$$

così che

$$(6.1) \quad \xi = m_{n+1} + e_{n+1}, \quad |e_{n+1}| < \frac{b_0 - a_0}{2^{n+1}}$$

La convergenza del metodo di bisezione è lenta: ad ogni passo “guadagnamo” una cifra binaria. Poiché  $10^{-1} \simeq 2^{-3.3}$ , in media ogni 3.3 passi guadagnamo una cifra decimale. Va però sottolineato che il metodo richiede solo la continuità della funzione  $f(x)$  e la conoscenza del segno di  $f(x)$  in  $[a_0, b_0]$ .

Ricordiamo infine che il metodo di bisezione era già stato da noi introdotto in occasione del calcolo degli autovalori di una matrice tridiagonale simmetrica (paragrafo 4.6).

---

#### Algoritmo 12: Bisez( $a_0, b_0, f, \text{toll}, x, \text{ier}$ )

---

*Commento.* L'algoritmo determina con tolleranza relativa  $\text{toll}$  una radice  $x$  dell'equazione  $f(x) = 0$ , contenuta nell'intervallo  $(a_0, b_0)$ , mediante la tecnica di bisezione. Se  $f(x_0)f(b_0) < 0$ , la variabile  $\text{ier}$  assume il valore 0, e il calcolo di  $x$  viene effettuato; altrimenti l'algoritmo si arresta e segnala l'inconveniente ponendo  $\text{ier} = 1$ .

*Parametri.* **Input:**  $a_0, b_0, f, \text{toll}$

**Output:**  $x, \text{ier}$

- 1: **se**  $f(a_0)f(b_0) > 0$  **allora**  $\text{ier} \leftarrow 1$ ; **esci**
- 2:  $\text{ier} \leftarrow 0$
- 3:  $n \leftarrow 0$
- 4:  $n \leftarrow n + 1$
- 5:  $m_n = (a_{n-1} + b_{n-1})/2$
- 6: **se**  $|b_{n-1} - a_{n-1}| < 2\text{toll}|m_n|(\dagger)$  **allora**  $x \leftarrow m_n$ ; **esci**
- 7: **se**  $f(b_0)f(m_n) < 0$  **allora**  $a_n \leftarrow m_n, b_n \leftarrow b_{n-1}$ ; **vai al punto 4**
- 8:  $a_n \leftarrow a_{n-1}$
- 9:  $b_n \leftarrow m_n$
- 10: **vai al punto 4**

## 11: esce

Nella tabella che segue riportiamo i successivi intervalli incapsulati  $(a_n, b_n)$  e i relativi punti medi  $m_n$  generati dall'algoritmo Bisez nel caso  $f(x) = \sinh(x) - 1/x$  e  $(a_0, b_0) \equiv (0.5, 2)$ .

$n$	$a_n$	$b_n$	$m_n$	$(b_n - a_n)/2$	$\text{sgn}(f(b_0)f(m_n))$
0	0.5000000	2.0000000	1.2500000	0.75	+
1	0.5000000	1.2500000	0.8750000	0.38	-
2	0.8750000	1.2500000	1.0625000	0.19	+
3	0.8750000	1.0625000	0.9687500	$0.94 \cdot 10^{-1}$	+
4	0.8750000	0.9687500	0.9218750	$0.47 \cdot 10^{-1}$	-
5	0.9218750	0.9687500	0.9453125	$0.23 \cdot 10^{-1}$	+
6	0.9218750	0.9453125	0.9335937	$0.12 \cdot 10^{-1}$	+
7	0.9218750	0.9335937	0.9277344	$0.59 \cdot 10^{-2}$	-
8	0.9277344	0.9335937	0.9306641	$0.29 \cdot 10^{-2}$	-
9	0.9306641	0.9335937	0.9321289	$0.15 \cdot 10^{-2}$	+
10	0.9306641	0.9321289	0.9313965	$0.73 \cdot 10^{-3}$	-
11	0.9313965	0.9321289	0.9317627	$0.37 \cdot 10^{-3}$	-
12	0.9317627	0.9321289	0.9319458	$0.18 \cdot 10^{-3}$	-
13	0.9319458	0.9321289	0.9320374	$0.92 \cdot 10^{-4}$	+
14	0.9319458	0.9320374	0.9319916	$0.46 \cdot 10^{-4}$	+
15	0.9319916	0.9320374	0.9320145	$0.23 \cdot 10^{-4}$	-
16	0.9320145	0.9320374	0.9320259	$0.11 \cdot 10^{-4}$	+
17	0.9320145	0.9320259	0.9320202	$0.57 \cdot 10^{-5}$	+
18	0.9320145	0.9320202	0.9320173	$0.29 \cdot 10^{-5}$	-
19	0.9320173	0.9320202	0.9320188	$0.14 \cdot 10^{-5}$	-
20	0.9320188	0.9320202	0.9320195	$0.72 \cdot 10^{-6}$	-
21	0.9320195	0.9320202	0.9320198	$0.36 \cdot 10^{-6}$	-
22	0.9320198	0.9320202	0.9320200	$0.18 \cdot 10^{-6}$	-
23	0.9320200	0.9320202	0.9320201	$0.89 \cdot 10^{-7}$	-
24	0.9320201	0.9320202	0.9320201		

Tabella 6.1

Prima di proseguire nella costruzione di metodi alternativi più efficienti, introduciamo una misura della velocità di convergenza di una successione numerica.

**Definizione 6.1.** Sia  $x_0, x_1, x_2, \dots, x_n, \dots$  una successione convergente al valore  $\xi$ ; poniamo  $e_n = \xi - x_n$ . Se esiste un numero reale  $p \geq 1$  e una costante (reale) positiva

(<sup>†</sup>) oppure  $|b_{n-1} - a_{n-1}| < 2\text{tol1}$ .

$c(< \infty)$  tali che

$$\lim_{n \rightarrow \infty} \frac{|e_{n+1}|}{|e_n|^p} = c$$

allora diciamo che la successione  $\{x_n\}$  ha ordine di convergenza  $p$ .

Quando  $p = 1, 2, 3$  la convergenza viene denominata lineare, quadratica, cubica. Nel caso della convergenza lineare è necessario che  $c \leq 1$  (generalmente  $c < 1$ ). La convergenza viene definita superlineare se  $1 < p < 2$ .

### 6.2.2 Metodi delle secanti, delle tangenti (Newton-Raphson) e altri

Dopo aver proposto nel paragrafo precedente un metodo con convergenza lineare (bisezione), nelle pagine che seguono presentiamo la costruzione di metodi iterativi con ordine di convergenza superiore a 1.

Partendo da un'approssimazione iniziale  $x_0$ , graficamente possiamo pensare di generare i valori successivi  $x_1, x_2, \dots, x_n$  nel modo seguente:

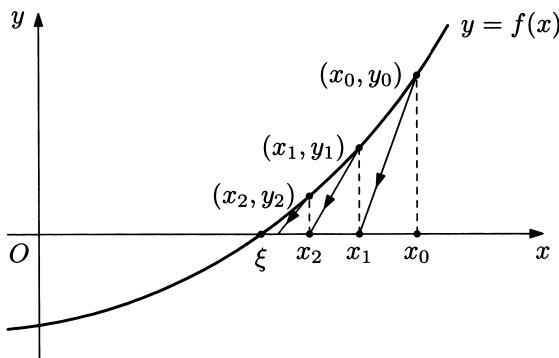


Figura 6.2

Ossia, conduciamo dal punto iniziale  $(x_0, y_0)$ ,  $y_0 = f(x_0)$ , sulla curva  $y = f(x)$  una retta con pendenza  $k_0$ , e prendiamo come nuova (e migliore) approssimazione  $x_1$  l'intersezione di questa retta con l'asse  $x$ . Ripartiamo poi dal nuovo punto  $(x_1, y_1)$ ,  $y_1 = f(x_1)$ , con una seconda retta avente pendenza  $k_1$ , e determiniamo l'intersezione  $x_2$  di quest'ultima con l'asse  $x$ ; e così via. In altri termini, ad ogni passo linearizziamo localmente il problema iniziale  $f(x) = 0$ , e come nuova approssimazione della radice  $\xi$  prendiamo la radice dell'equazione lineare

$$y_n + k_n(x - x_n) = 0, \quad n = 0, 1, 2, \dots$$

cioè

$$x_{n+1} = x_n - \frac{f(x_n)}{k_n}, \quad n = 0, 1, 2, \dots$$

Le "direzioni"  $k_0, k_1, k_2, \dots$  possono essere scelte in molti modi. Vediamo alcuni esempi.

$$1. \text{ Regula falsi: } k_n = \frac{y_n - y_0}{x_n - x_0}$$

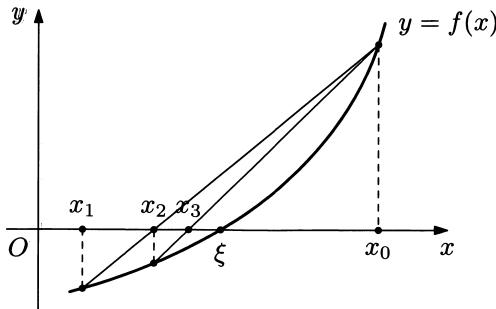


Figura 6.3

$$2. \text{ Metodo delle secanti: } k_n = \frac{y_n - y_{n-1}}{x_n - x_{n-1}}$$

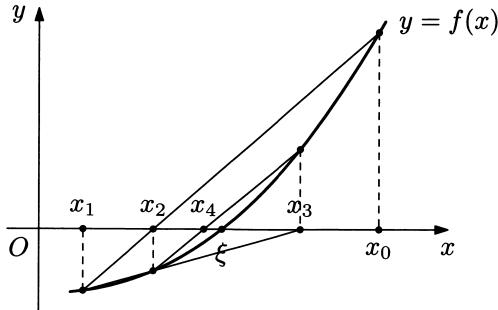


Figura 6.4

Questi primi due metodi richiedono la conoscenza di due approssimazioni iniziali.

$$3. \text{ Metodo delle tangenti o di Newton-Raphson: } k_n = f'(x_n) \text{ (vedi figura 6.5).}$$

In questo caso è indispensabile l'esistenza di  $f'(x_n)$  per ogni  $n$ .

Esaminiamo i metodi proposti e poniamoci le seguenti domande: la successione  $x_0, x_1, x_2, \dots, x_n, \dots$  converge alla radice  $\xi$ ? Se sì, qual è la velocità di convergenza? Ovvero, qual è l'ordine del metodo?

Supponiamo che la radice  $\xi$  sia semplice, cioè  $f'(\xi) \neq 0$ , e scriviamo la formula di interpolazione di Newton costruita su due nodi  $\{a, b\}$ , non necessariamente distinti:

$$f(x) = f(a) + (x - a)f[a, b] + (x - a)(x - b)f[x, a, b]$$

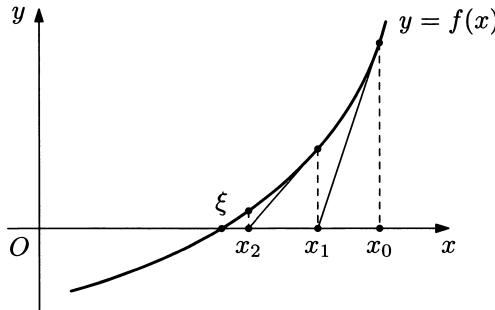


Figura 6.5

In particolare abbiamo

$$f(\xi) = 0 = f(a) + (\xi - a)f[a, b] + (\xi - a)(\xi - b)f[\xi, a, b]$$

dalla quale otteniamo

$$\xi = a - \frac{f(a)}{f[a, b]} - (\xi - a)(\xi - b) \frac{f[\xi, a, b]}{f[a, b]}$$

Come approssimazione della radice  $\xi$  prendiamo l'ascissa

$$(6.2) \quad c = a - \frac{f(a)}{f[a, b]}$$

cui è associato l'errore

$$(6.3) \quad e = \xi - c = -(\xi - a)(\xi - b) \frac{f[\xi, a, b]}{f[a, b]}$$

Ricordiamo che quando  $f''(x)$  è continua in un intervallo contenente  $a, b$  e  $\xi$ , si ha

$$f[a, b] = f'(\alpha) \quad \text{e} \quad f[\xi, a, b] = \frac{1}{2}f''(\beta)$$

dove  $\alpha$  e  $\beta$  sono due punti (non noti) del suddetto intervallo.

Scegliendo convenientemente in (6.2) i nodi  $a$  e  $b$  otteniamo i metodi 1, 2 e 3. Infatti ponendo in (6.2)<sup>(†)</sup>  $a = x_n$  e  $b = x_0$  abbiamo la regula falsi, con  $a = x_n$  e  $b = x_{n-1}$  otteniamo il metodo delle secanti, mentre la scelta  $a = b = x_n$  produce il metodo delle tangenti.

Sia  $f(\xi) = 0$  e  $f'(\xi) \neq 0$ . Supponiamo inoltre che la funzione  $f(x)$  sia di classe  $C^2$  in un intorno  $I$  della radice  $\xi$  in cui, qualunque sia la coppia  $a, b$  di punti scelti, si abbia

$$\left| \frac{f[\xi, a, b]}{f[a, b]} \right| = \frac{|f''(\beta)|}{2|f'(\alpha)|} \leq M < \infty$$

(†) Dove, ovviamente  $c = x_{n+1}$ .

Ricordando la (6.3) possiamo pertanto scrivere

$$|e| \leq M|\xi - a||\xi - b|$$

In particolare, per la regula falsi abbiamo

$$|e_{n+1}| \leq (M|e_0|)|e_n|$$

per il metodo delle secanti

$$(6.4) \quad |e_{n+1}| \leq M|e_{n-1}e_n|$$

mentre per quello delle tangenti

$$|e_{n+1}| \leq M|e_n|^2$$

avendo definito  $e_n = \xi - x_n$ . Supponendo quindi che la funzione  $f(x)$  sia tale per cui tutte le successive approssimazioni prodotte dal metodo numerico scelto appartengano al predetto intervallo  $I$ , la convergenza dei tre metodi è certamente assicurata se  $|e_0|$  (ed  $|e_1|$  nel caso del metodo delle secanti) è sufficientemente piccolo. Infatti, per la regula falsi risulta

$$|e_{n+1}| \leq k|e_n| \leq k^n|e_0|, \quad k = M|e_0|$$

e  $\lim_{n \rightarrow \infty} e_{n+1} = 0$  quando  $k < 1$ ; per il metodo delle secanti invece, se  $|e_0|$  e  $|e_1|$  sono entrambi minori di  $M^{-1}$ , ovvero

$$|e_0|M \leq k \quad \text{e} \quad |e_1|M \leq k \quad \text{con } k < 1$$

dalla (6.4) otteniamo, per esempio,

$$\begin{aligned} |e_2| &\leq k|e_1| \\ |e_3| &\leq k|e_2| \\ |e_4| &\leq k^2|e_3| \\ |e_5| &\leq k^3|e_4| \\ &\vdots \\ |e_{n+1}| &\leq k^{m_n}|e_n|, \quad \text{con } \lim_{n \rightarrow \infty} m_n = +\infty (\dagger) \end{aligned}$$

onde

$$\lim_{n \rightarrow \infty} e_{n+1} = 0$$

Inoltre,

$$\lim_{n \rightarrow \infty} \frac{|e_{n+1}|}{|e_n|} = \lim_{n \rightarrow \infty} k^{m_n} = 0$$

---

(†) La successione  $\{m_n\}$  coincide con quella dei numeri di Fibonacci, che possiamo definire con la relazione  $m_0 = 0$ ,  $m_1 = 1$ ,  $m_{i+2} = m_{i+1} + m_i$ ,  $i = 0, 1, 2, \dots$ .

per cui in questo caso la convergenza risulta almeno superlineare. Lasciamo al lettore l'esame della convergenza del metodo delle tangenti.

Determiniamo infine gli ordini dei tre metodi.

1. *Regula falsi*. Poiché

$$e_{n+1} = -e_n e_0 \frac{f[\xi, x_n, x_0]}{f[x_n, x_0]}$$

abbiamo

$$\lim_{n \rightarrow \infty} \frac{|e_{n+1}|}{|e_n|} = |e_0| \left| \frac{f[\xi, \xi, x_0]}{f[\xi, x_0]} \right|$$

inoltre, essendo  $f'(\xi) \neq 0$ , se  $x_0$  è sufficientemente vicino a  $\xi$  risulta  $f[\xi, x_0] \neq 0$ . Il metodo ha pertanto ordine di convergenza almeno  $p = 1$ . Quando  $f''(\xi) \neq 0$  possiamo affermare che l'ordine è esattamente 1.

2. *Secanti*. In questo caso

$$e_{n+1} = -e_n e_{n-1} \frac{f[\xi, x_n, x_{n-1}]}{f[x_n, x_{n-1}]}$$

e

$$\lim_{n \rightarrow \infty} \frac{e_{n+1}}{e_n e_{n-1}} = -\frac{f''(\xi)}{2f'(\xi)}$$

Tuttavia, per determinare l'ordine del metodo occorre individuare il reale  $p \geq 1$  che permette di scrivere la relazione

$$\lim_{n \rightarrow \infty} \frac{|e_{n+1}|}{|e_n|^p} = c, \quad 0 < c < \infty$$

È possibile dimostrare che

$$\left[ \lim_{n \rightarrow \infty} \frac{|e_{n+1}|}{|e_n e_{n-1}|} \right]^{1/p} = \lim_{n \rightarrow \infty} \frac{|e_{n+1}|}{|e_n|^p}, \quad p = \frac{\sqrt{5} + 1}{2} \quad (\text{sezione aurea})$$

e quindi concludere che il metodo delle secanti ha ordine  $p \approx 1.618$ .

3. *Tangenti o Newton-Raphson*. Dalla relazione

$$e_{n+1} = -e_n^2 \frac{f[\xi, x_n, x_n]}{f[x_n, x_n]}$$

deduciamo

$$\lim_{n \rightarrow \infty} \frac{e_{n+1}}{e_n^2} = -\frac{f''(\xi)}{2f'(\xi)}$$

L'ordine di questo metodo è pertanto  $p = 2$  (assumendo  $f''(\xi) \neq 0$ , altrimenti  $p > 2$ ).

Osserviamo tuttavia che mentre i metodi 1 e 2 richiedono ad ogni passo<sup>(†)</sup> la valutazione di  $f(x)$  in un solo punto (l'ultimo trovato), il metodo di Newton per determinare  $x_{n+1}$  richiede la valutazione sia di  $f(x_n)$  che di  $f'(x_n)$ .

I tre metodi che finora abbiamo esaminato sono stati applicati all'equazione  $\sinh(x) - 1/x = 0$ , già considerata nel paragrafo 6.2.1. Le approssimazioni ottenute, prendendo  $x_0 = 0.5$  e  $x_1 = 2$  per i metodi della regula falsi e delle secanti, e  $x_0 = 0.5$  per quello delle tangenti, sono riportate nella tabella che segue.

$n$	$x_n$		
	regula falsi	secanti	tangenti
0	0.5000000	0.5000000	0.5000000
1	2.0000000	2.0000000	0.7884190
2	0.9816479	0.9816479	0.9231899
3	0.9431759	0.9380459	0.9319985
4	0.9346181	0.9319488	0.9320200
5	0.9326298	0.9320201	
6	0.9321634	0.9320200	
7	0.9320537	0.9320200	
8	0.9320279		
9	0.9320219		
10	0.9320205		
11	0.9320201		
12	0.9320201		

Tabella 6.2

Esaminiamo il significato del concetto di ordine di convergenza. Prima di tutto osserviamo che quando al passo  $n$ -esimo l'errore assoluto di  $x_n$  è  $|e_n| \leq \frac{1}{2}10^{-k}$ , cioè  $x_n$  ha  $k$  decimali corretti, abbiamo

$$|e_{n+1}| \simeq c \left( \frac{1}{2}10^{-k} \right)^p = \frac{c}{2^p} 10^{-pk}$$

il numero di decimali corretti presenti in  $x_{n+1}$  è dunque dell'ordine di  $pk$ . La relazione precedente è in realtà una relazione limite; ciò significa che il numero di decimali corretti tende ad essere moltiplicato per  $p$  ad ogni passo solo per  $n \rightarrow \infty$ . Per valori finiti di  $n$  (e soprattutto nei primi passi) l'aumento di cifre corrette dipende anche dalla costante moltiplicativa  $c_n$  presente nella relazione

$$|e_{n+1}| = c_n |e_n|^p, \quad \lim_{n \rightarrow \infty} c_n = c$$

(†) Tranne quello iniziale.

Ritorniamo al metodo delle secanti e osserviamo che le espressioni

$$x_{n+1} = x_n - y_n \frac{x_n - x_{n-1}}{y_n - y_{n-1}} = \frac{x_{n-1}y_n - x_n y_{n-1}}{y_n - y_{n-1}}$$

non sono equivalenti nell'aritmetica di un calcolatore. Nell'ultima, quando  $x_n \simeq x_{n-1}$  (e ciò si verifica senz'altro quando  $x_n$  è prossima a  $\xi$ ) e  $y_n y_{n-1} > 0$  il fenomeno della cancellazione numerica può limitare seriamente la massima precisione raggiungibile. Anche nella prima espressione abbiamo della cancellazione numerica, ma nel termine correttivo, non su  $x_{n+1}$ . Più  $x_n$  si avvicina a  $x_{n+1}$ , più piccolo sarà il contributo della correzione  $y_n \frac{x_n - x_{n-1}}{y_n - y_{n-1}}$ ; solo le primissime cifre di quest'ultimo termine risultano "significative" nella determinazione (con l'aritmetica di macchina) di  $x_{n+1}$ .

I metodi finora presentati in questo paragrafo sono stati tutti esaminati supponendo che la radice incognita sia semplice, cioè  $f'(\xi) \neq 0$ . Che cosa succede quando la radice è *doppia*? O più in generale quando essa ha molteplicità  $\mu > 1$ , cioè quando  $f(\xi) = f'(\xi) = \dots = f^{(\mu-1)}(\xi) = 0$  e  $f^{(\mu)}(\xi) \neq 0$ ? Esaminiamo il comportamento del metodo di Newton-Raphson:

$$\begin{aligned} e_{n+1} &= e_n + \frac{f(\xi - e_n)}{f'(\xi - e_n)} = \\ &= e_n + \frac{f(\xi) - e_n f'(\xi) + \dots + \frac{(-1)^{\mu-1}}{(\mu-1)!} e_n^{\mu-1} f^{(\mu-1)}(\xi) + \frac{(-1)^\mu}{\mu!} e_n^\mu f^{(\mu)}(\eta_n)}{f'(\xi) - e_n f''(\xi) + \dots + \frac{(-1)^{\mu-2}}{(\mu-2)!} e_n^{\mu-2} f^{(\mu-1)}(\xi) + \frac{(-1)^{\mu-1}}{(\mu-1)!} e_n^{\mu-1} f^{(\mu)}(\delta_n)} = \\ &= e_n - \frac{\frac{e_n^\mu}{\mu!} f^{(\mu)}(\eta_n)}{\frac{e_n^{\mu-1}}{(\mu-1)!} f^{(\mu)}(\delta_n)} = e_n \left[ 1 - \frac{1}{\mu} \frac{f^{(\mu)}(\eta_n)}{f^{(\mu)}(\delta_n)} \right] \end{aligned}$$

donde

$$\lim_{n \rightarrow \infty} \frac{e_{n+1}}{e_n} = \frac{\mu - 1}{\mu}$$

Pertanto, quando  $\mu > 1$  il metodo risulta ancora convergente, ma l'ordine scende a  $p = 1$ . Per ristabilire la convergenza quadratica è sufficiente modificare la formula di Newton-Raphson come segue:

$$(6.5) \quad x_{n+1} = x_n - \mu \frac{f(x_n)}{f'(x_n)}$$

$n$	$x_n$	
	$\mu = 1$	$\mu = 2$
0	-0.5000000	-0.5000000
1	-0.7553419	-1.0106838
2	-0.8782848	-0.9999999
3	-0.9392176	-1.0000000
4	-0.9696182	
5	-0.9848103	
6	-0.9924053	
7	-0.9962027	
8	-0.9981014	
9	-0.9990507	
10	-0.9995254	
11	-0.9997627	
12	-0.9998814	
13	-0.9999407	
14	-0.9999704	
15	-0.9999852	
16	-0.9999926	
17	-0.9999963	
18	-0.9999982	
19	-0.9999991	
20	-0.9999996	
21	-0.9999998	
22	-0.9999999	
23	-0.9999999	
24	-1.0000000	

Tabella 6.3

Come esempio numerico, consideriamo l'equazione  $\cos(x + 1) - 1 = 0$  e proponiamoci di determinare la radice (doppia) di cui supponiamo di conoscere un'approssimazione iniziale  $x_0 = -0.5$ . Alla suddetta equazione applichiamo sia il metodo di Newton-Raphson nella forma originale ( $\mu = 1$ ) sia quello modificato (6.5) con  $\mu = 2$ . I risultati ottenuti, che riportiamo nella tabella 6.3, confermano le nostre precedenti affermazioni sull'ordine di convergenza delle successioni prodotte dalle due formule.

La formula (6.5) è utilizzabile quando la molteplicità della radice è nota. Se invece la molteplicità non è nota possiamo considerare la nuova funzione  $g(x) = f(x)/f'(x)$ , per la quale  $g(\xi) = 0$  e  $g'(\xi) \neq 0$ , e applicare ad essa il metodo delle tangenti:

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

La convergenza della successione  $\{x_n\}$  è nuovamente quadratica; tuttavia, per poter applicare questa formula è necessario valutare anche  $f''(x_n)$ .

La convergenza dei metodi delle secanti e delle tangenti è garantita quando, supposte soddisfatte le condizioni di regolarità sulla  $f(x)$  richieste per la convergenza, l'approssimazione iniziale  $x_0$  (e  $x_1$  nel caso delle secanti) è “sufficientemente” vicina alla radice. Pertanto, tali metodi spesso si rivelano efficienti soprattutto per migliorare un'approssimazione “sufficientemente buona” ottenuta con un metodo di ordine 1 la cui convergenza è assicurata. Per esempio potremmo utilizzare il metodo di bisezione per determinare un'approssimazione con 1-2 cifre significative, e poi applicare il metodo di Newton-Raphson oppure quello delle secanti per ottenere con pochissime iterazioni la precisione desiderata.

I metodi iterativi introdotti finora in questo paragrafo sono stati costruiti approssimando localmente la funzione  $f(x)$  con una retta, e assumendo come nuova approssimazione della radice incognita l'intersezione di questa retta con l'asse  $x$ . Poiché anche le radici di una generica parabola sono immediatamente determinabili, possiamo pensare di migliorare l'approssimazione locale di  $f(x)$  utilizzando una parabola anziché una retta. Vediamo due esempi.

1. Consideriamo tre approssimazioni  $x_{n-2}, x_{n-1}, x_n$  della radice  $\xi$ , e costruiamo la parabola  $y = ax^2 + bx + c$  passante per i punti  $(x_{n-2}, f(x_{n-2})), (x_{n-1}, f(x_{n-1})), (x_n, f(x_n))$ . Come nuova approssimazione  $x_{n+1}$  di  $\xi$  prendiamo la radice (reale) della parabola più vicina a  $x_n$ . Questo metodo è attribuito a *Muller* e ha ordine di convergenza  $p \simeq 1.84$ .
2. Osserviamo che il metodo di Newton è stato sostanzialmente ottenuto sviluppando  $f(x)$  in serie di Taylor nell'intorno del punto  $x = x_n$

$$f(\xi) = f(x_n) + (\xi - x_n)f'(x_n) + \frac{1}{2}(\xi - x_n)^2 f''(\eta_n) = 0$$

e trascurando il termine di ordine  $(\xi - x_n)^2$ . Estendiamo questa idea. Consideriamo lo sviluppo

$$f(\xi) = f(x_n) + (\xi - x_n)f'(x_n) + \frac{1}{2}(\xi - x_n)^2 f''(x_n) + \frac{1}{6}(\xi - x_n)^3 f'''(\delta_n) = 0$$

e trascuriamo il termine di ordine 3; otteniamo

$$\xi \simeq x_{n+1} = x_n - \frac{2f(x_n)}{f'(x_n) + \text{sgn}[f'(x_n)]\sqrt{[f'(x_n)]^2 - 2f(x_n)f''(x_n)}}$$

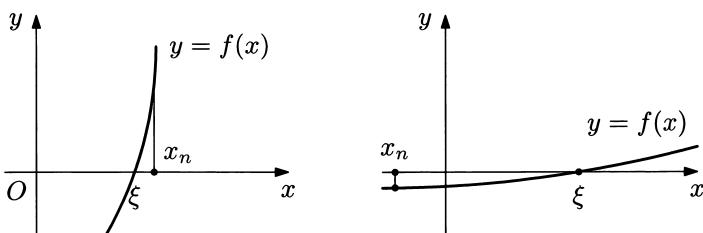
L'ordine di convergenza di questa formula è  $p = 3$ . Quest'ultimo metodo, nonostante coinvolga  $f''(x_n)$ , è certamente preferibile a quello di Newton in quei casi, non rari, in cui la funzione  $f(x)$  è soluzione di una equazione differenziale del secondo ordine esplicita nella  $f''(x)$ .

### 6.2.3 Test di convergenza

Quale criterio adottiamo per decidere se una data approssimazione  $x_n$  è sufficientemente accurata? Potremmo imporre la condizione

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

Le seguenti due situazioni pongono in evidenza gli inconvenienti che questo test può determinare:



**Figura 6.6**

Una seconda possibilità consiste nel richiedere

$$|x_n - x_{n+1}| \leq x_{\text{toll}} \quad \text{oppure} \quad |x_n - x_{n+1}| \leq x_{\text{toll}}|x_{n+1}|$$

Mentre questo test risulta valido per successioni convergenti con ordine  $p > 1$ , può invece provocare degli inconvenienti quando la convergenza è solamente lineare.

Quando non si ha la certezza della bontà di uno dei due test conviene includerli entrambi. Occorre inoltre precisare sempre un numero massimo di iterazioni e arrestare il processo di calcolo quanto tale limite viene superato.

---

**Algoritmo 13:** Secant( $x_0, x_1, f, x_{\text{toll}}, f_{\text{toll}}, n_{\text{max}}, \text{ier}$ )

---

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

*Parametri.* **Input:**  $x_0, x_1, f, x_{\text{toll}}, f_{\text{toll}}, n_{\text{max}}$

**Output:**  $n_{\text{max}}, x, \text{ier}$

---

```

1: se  $f(x_0)f(x_1) > 0$  allora ier  $\leftarrow -1$ ; esci
2: ciclo 1:  $n = 1, \dots, n_{\max}$ 
3:    $x_{n+1} = x_n - f(x_n)(x_n - x_{n-1})/(f(x_n) - f(x_{n-1}))$ 
4:   se  $|x_n - x_{n+1}| > x_{\text{toll}}|x_{n+1}|^{(\dagger)}$  allora ier  $\leftarrow 1$ ; vai al punto 10
5:   se  $|f(x_{n+1})| > f_{\text{toll}}$  allora ier  $\leftarrow 2$ ; vai al punto 10
6:   ier  $\leftarrow 0$ 
7:    $x \leftarrow x_{n+1}$ 
8:    $n_{\max} \leftarrow n$ 
9:   esci
10: fine ciclo 1
11:  $x \leftarrow x_{n+1}$ 
12: esci

```

---

#### 6.2.4 Metodi iterativi in generale

Supponiamo di riscrivere l'equazione in esame  $f(x) = 0$  nella forma

$$(6.6) \quad x = g(x)$$

con  $g(x)$  derivabile<sup>(†)</sup> in un intorno  $I$  della radice incognita  $\xi$ , e tale che  $\xi = g(\xi)$  se e solo se  $f(\xi) = 0$ . Nota un'approssimazione iniziale  $x_0$  della radice  $\xi$ , utilizziamo il procedimento iterativo

$$(6.7) \quad x_{n+1} = g(x_n), \quad n = 0, 1, 2, \dots$$

per costruire la successione  $\{x_n\}$ . Se quest'ultima converge a un punto  $\alpha$ , allora  $\xi \equiv \alpha$ , cioè  $f(\alpha) = 0$

Esaminiamo in figura 6.7 il significato geometrico della formula ricorsiva (6.6) in quattro situazioni diverse.

L'equazione  $f(x) = 0$  può sempre essere riscritta nella forma  $x = g(x)$ ; anzi, ciò può esser fatto in un numero infinito di modi. Come mostrano i grafici suddetti la funzione  $g(x)$ , e in particolare il valore di  $g'(x)$ , svolge un ruolo fondamentale; quest'ultimo è infatti il diretto responsabile della convergenza o meno della successione  $\{x_n\}$ .

Consideriamo l'equazione  $x^4 - 4 = 0$ ; essa potrebbe, per esempio, venire riscritta nei due modi seguenti:

- (i)  $x = x^4 + x - 4$
- (ii)  $x = (4 + 11x - x^4)/11$

Partendo dall'approssimazione iniziale  $x_0 = 1$  otteniamo quanto riportato in tabella 6.4.

---

<sup>(†)</sup> Oppure  $|x_n - x_{n+1}| > x_{\text{toll}}$ .

<sup>(††)</sup> È tuttavia sufficiente che la funzione  $g(x)$  risulti lipschitziana in un intorno  $I$  della radice  $\xi$ , ossia  $|g(x_1) - g(x_2)| \leq m|x_1 - x_2|$  per ogni coppia  $x_1, x_2 \in I$ .

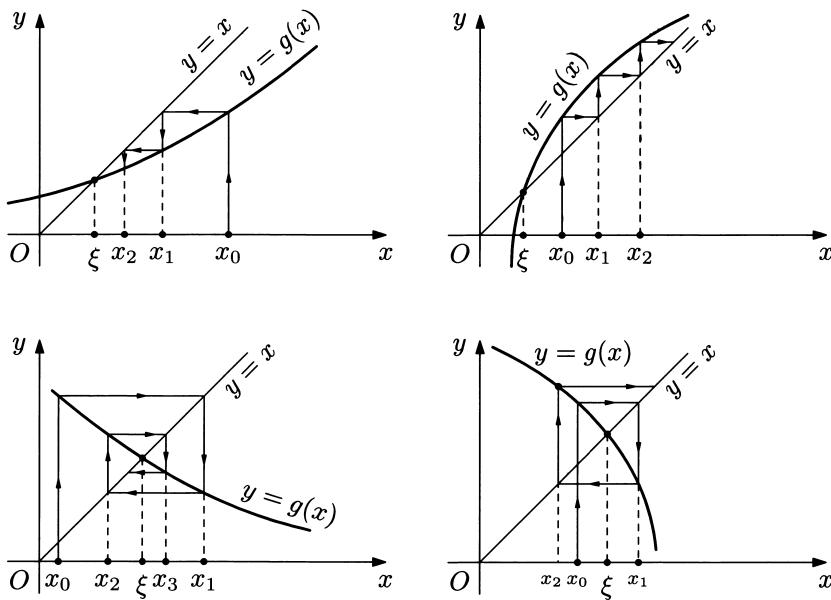


Figura 6.7

	(i)	(ii)
$x_1$	-2	1.272727
$x_2$	10	1.397831
$x_3$	$10^4$	1.414390
$x_4$	$10^{16}$	1.414209
$x_5$	$10^{64}$	1.4142138

Tabella 6.4

Il valore esatto della radice incognita è  $\xi = \sqrt{2} = 1.414213562\dots$ . Come vediamo, solo la (ii) converge. La (i) addirittura diverge. Che cosa succede? Come mai la forma di  $g(x)$  condiziona così pesantemente il comportamento della successione  $\{x_n\}$ ? Per dare una risposta a queste domande esaminiamo analiticamente la successione degli errori  $e_n = \xi - x_n$ . Concatenando le relazioni

$$\xi - x_1 = g(\xi) - g(x_0) = (\xi - x_0)g'(\xi_0)$$

$$\xi - x_2 = g(\xi) - g(x_1) = (\xi - x_1)g'(\xi_1)$$

$$\dots$$

$$\xi - x_n = g(\xi) - g(x_{n-1}) = (\xi - x_{n-1})g'(\xi_{n-1})$$

dove  $\xi$  è un punto (non noto) dell'intervallo con estremi in  $\xi$  e  $x_i$ , otteniamo

$$\begin{aligned}\xi - x_n &= (\xi - x_{n-1})g'(\xi_{n-1}) = (\xi - x_{n-2})g'(\xi_{n-2})g'(\xi_{n-1}) = \dots \\ &= (\xi - x_0)g'(\xi_0) \dots g'(\xi_{n-2})g'(\xi_{n-1})\end{aligned}$$

Se poi

$$|g'(\xi_i)| \leq m, \quad i = 0, 1, 2, \dots$$

allora possiamo scrivere

$$|e_n| \leq m^n |e_0|$$

e quindi concludere (vedi anche [13, pag. 152]):

**Teorema 6.1.** *La successione  $\{x_n\}$  risulta certamente convergente quando  $m < 1$ , cioè quando per la funzione  $g(x)$  scelta si ha  $|g'(x)| \leq m < 1$  in tutto un intervallo  $I$  contenente l'approssimazione iniziale  $x_0$  e tale che  $g(x) \in I$  per ogni  $x \in I$ <sup>(†)</sup>. In tale situazione  $\xi$  è l'unica radice dell'equazione  $f(x) = 0$  presente nel suddetto intervallo. Quando invece in tutto un intorno di  $\xi$  risulta  $|g'(x)| > 1$  la successione non può convergere.*

Osserviamo che quando  $-m \leq g'(x) < 0$  nell'intervallo  $I$  predetto, le  $\{x_n\}$  convergono alla radice  $\xi$  approssimandola alternativamente per difetto e per eccesso, mentre nel caso  $0 < g'(x) \leq m$  le successive approssimazioni sono tutte per difetto (se  $x_0 < \xi$ ) oppure tutte per eccesso (se  $x_0 > \xi$ ). Inoltre, più piccola è la costante  $m$  più rapida è la convergenza.

I risultati che abbiamo ottenuto ci consentono ora di spiegare i diversi comportamenti delle due successioni (i) e (ii) dell'esempio iniziale  $x^4 - 4 = 0$ . Infatti, nel caso dell'equazione (i) abbiamo  $g'(x) > 1$  per ogni  $x > 0$ , mentre nella (ii), considerato per esempio l'intervallo  $I = [1, 1.7]$  che ha immagine  $g(I) \subset I$ , in  $I$  abbiamo  $-0.79 < g'(x) < 0.64$ .

Supposto di aver individuato un intervallo  $I$  contenente la radice  $\xi$  dell'equazione  $f(x) = 0$ , scegliamo

$$(6.8) \quad g(x) = x - \frac{1}{k}f(x)$$

con la costante  $k$  tale da avere, se possibile,  $g(x) \in I$  per ogni  $x \in I$  e

$$\left| 1 - \frac{1}{k}f'(x) \right| \leq m < 1, \quad x \in I$$

Poiché la convergenza del metodo è tanto più rapida quanto più piccola è  $m$ , voler mantenere  $k$  costante, anche quando l'approssimazione  $x_n$  si avvicina alla radice  $\xi$ , appare poco efficiente; una scelta  $k = k_n$  che permetta di ridurre progressivamente il fattore  $m$  (per  $x_n \rightarrow \xi$ ), quando ciò è consentito, è senza dubbio migliore. I metodi delle secanti e delle tangenti visti nei paragrafi precedenti possono essere interpretati proprio come processi iterativi di tipo (6.5) originati da una (6.8) con  $k = k_n$ .

(†) La condizione  $g(x) \in I$  per ogni  $x \in I$  garantisce l'esistenza in  $I$  di almeno una soluzione, di solito chiamata *punto fisso* della trasformazione  $y = g(x)$ , dell'equazione  $x = g(x)$ .

Supponiamo di aver scelto una particolare espressione per la funzione  $g(x)$ , e di aver costruito la successione convergente  $x_{n+1} = g(x_n)$ . Qual è l'ordine di convergenza di tale successione? Per rispondere a questa domanda supponiamo  $g(x) \in C^{l+1}(I)$ ,  $x_n \in I$ , e scriviamo

$$e_{n+1} = g(\xi) - g(x_n) = g'(\xi)e_n - \frac{g''(\xi)}{2!}e_n^2 + \cdots + \frac{(-1)^{l-1}g^{(l)}(\xi)}{l!}e_n^l + \frac{(-1)^lg^{(l+1)}(\eta_n)}{(l+1)!}e_n^{l+1}$$

Quando  $g'(\xi) \neq 0$  otteniamo

$$\lim_{n \rightarrow \infty} \frac{e_{n+1}}{e_n} = g'(\xi)$$

e il processo (6.6) ha ordine 1. Se invece  $g'(\xi) = g''(\xi) = \cdots = g^{(l)}(\xi) = 0$  e  $g^{(l+1)}(\xi) \neq 0$  l'ordine del metodo (6.6) sale a  $l+1$ . Infatti abbiamo

$$\lim_{n \rightarrow \infty} \frac{e_{n+1}}{e_n^{l+1}} = \frac{(-1)^l}{(l+1)!}g^{(l+1)}(\xi)$$

### 6.2.5 Metodo di accelerazione $\Delta^2$ di Aitken

Supponiamo che la successione  $\{x_n\}$  prodotta dal processo iterativo  $x_{n+1} = g(x_n)$ ,  $n = 0, 1, 2, \dots$ , abbia ordine di convergenza  $p = 1$ , ossia

$$\lim_{n \rightarrow \infty} \frac{\xi - x_{n+1}}{\xi - x_n} = g'(\xi) \neq 0$$

possiamo allora scrivere

$$(6.9) \quad \xi - x_{n+1} = (\xi - x_n)[g'(\xi) + \alpha_n]$$

con

$$\lim_{n \rightarrow \infty} \alpha_n = 0$$

In questa situazione è possibile dedurre dalla successione  $\{x_n\}$  una seconda successione  $\{x'_n\}$  convergente a  $\xi$  più rapidamente della prima, nel senso che

$$(6.10) \quad \lim_{n \rightarrow \infty} \frac{\xi - x'_n}{\xi - x_n} = 0$$

Per ottenere una  $\{x'_n\}$  con la caratteristica (6.10) osserviamo preliminarmente che la (6.9) legittima la bontà della relazione approssimata

$$\frac{\xi - x_{n+1}}{\xi - x_n} \simeq g'(\xi)$$

almeno per  $n$  sufficientemente grande, dove il segno  $\simeq$  converge all'uguaglianza quando  $n \rightarrow \infty$ . Pertanto

$$\frac{\xi - x_{n+2}}{\xi - x_{n+1}} \simeq \frac{\xi - x_{n+1}}{\xi - x_n}$$

ovvero,

$$(\xi - x_{n+1})^2 - (\xi - x_{n+2})(\xi - x_n) \simeq 0$$

Sostituendo l'uguaglianza approssimata con quella esatta e risolvendo l'ultima equazione nella variabile  $\xi$  non troveremo la vera radice  $\xi$ , bensì una sua nuova approssimazione che denotiamo con  $x'_{n+2}$ :

$$(6.11) \quad x'_{n+2} = x_{n+2} - \frac{(x_{n+2} - x_{n+1})^2}{x_{n+2} - 2x_{n+1} + x_n} = x_n - \frac{(x_{n+1} - x_n)^2}{x_{n+2} - 2x_{n+1} + x_n}, \quad n = 0, 1, 2, \dots$$

Questa è la *formula di accelerazione* (della convergenza) di Aitken. Infatti è possibile dimostrare ([6, v. 1, pag. 248]) che quando  $|g'(\xi)| < 1$  la nuova successione  $\{x'_{n+2}\}$  ha la proprietà (6.10). Tuttavia, se l'ordine di convergenza della successione  $\{x_n\}$  non è esattamente 1, la nuova successione  $\{x'_n\}$ , generata dalla (6.11), non gode più della proprietà (6.10); anzi, la  $\{x'_n\}$  potrebbe addirittura non convergere.

La formula (6.11) è stata utilizzata per accelerare la successione prodotta dalla regula falsi, che ha ordine di convergenza  $p = 1$ , nell'esempio relativo alla tabella 6.2; abbiamo ottenuto i seguenti risultati:

$n$	regula falsi	Aitken
0	0.5000000	
1	2.0000000	
2	0.9816479	
3	0.9431759	0.9416654
4	0.9346181	0.9321700
5	0.9326298	0.9320281
6	0.9321634	0.9320205
7	0.9329537	0.9320200
8	0.9329279	
9	0.9329219	
10	0.9320205	
11	0.9320201	

**Tabella 6.5**

Nelle applicazioni si preferisce tuttavia procedere con l'algoritmo seguente, noto con il nome di *metodo di Steffensen*:

- 1: **ciclo 1:**  $n = 1, \dots, n_{\max}$
- 2:  $y_n \leftarrow g(x_n)$
- 3:  $z_n \leftarrow g(y_n)$
- 4:  $x_{n+1} \leftarrow x_n - \frac{(y_n - x_n)^2}{z_n - 2y_n + x_n}$
- 5: se la tolleranza richiesta è raggiunta allora esci
- 6: **fine ciclo 1**

Ritornando all'equazione iniziale  $f(x) = 0$ , osserviamo che il metodo di Steffensen può essere riformulato come segue:

$$x_{n+1} = x_n - \frac{f(x_n)}{h(x_n)}, \quad h(x_n) = \frac{f(x_n + f(x_n)) - f(x_n)}{f(x_n)}, \quad n = 0, 1, 2, \dots$$

È possibile dimostrare che l'ordine di convergenza di questo metodo è  $p = 2$ . Esso richiede 2 valutazioni di funzione ad ogni passo.

Concludiamo questo paragrafo rilevando che il metodo di Aitken può essere utilizzato per accelerare la convergenza di una qualsiasi successione numerica che abbia ordine di convergenza  $p = 1$ . Per esempio potrebbe essere utilizzato per accelerare il metodo delle potenze per il calcolo dell'autovalore di modulo massimo; infatti, quando supponiamo, per esempio,  $|\lambda_1| > |\lambda_2| > |\lambda_3| \geq \dots$  abbiamo

$$\lim_{m \rightarrow \infty} \frac{\lambda_1 - \lambda_1^{(m+1)}}{\lambda_1 - \lambda_1^{(m)}} = \frac{\lambda_2}{\lambda_1}$$

## 6.3 Sistemi di equazioni non lineari

In questo paragrafo esaminiamo la costruzione di metodi numerici per la risoluzione di sistemi di equazioni non lineari

$$(6.12) \quad \begin{cases} f_1(x_1, x_2, \dots, x_n) = 0 \\ f_2(x_1, x_2, \dots, x_n) = 0 \\ \dots \\ f_n(x_1, x_2, \dots, x_n) = 0 \end{cases}$$

che possiamo anche scrivere nella forma più compatta  $f(x) = 0$ , definendo in quest'ultima  $x = (x_1, x_2, \dots, x_n)^T$  e

$$f(x) = \begin{pmatrix} f_1(x_1, x_2, \dots, x_n) \\ f_2(x_1, x_2, \dots, x_n) \\ \dots \\ f_n(x_1, x_2, \dots, x_n) \end{pmatrix}$$

Alcuni dei metodi proposti nel caso di una singola equazione possono venire generalizzati con successo a sistemi di equazioni. È a questi che essenzialmente ci limiteremo.

### 6.3.1 Metodo di Newton e sue varianti

Il metodo di Newton-Raphson per le equazioni non lineari descritto nel paragrafo 6.2.2 può essere facilmente generalizzato al caso di sistemi di equazioni non lineari. A tale fine consideriamo un'approssimazione  $x^{(i)} = (x_1^{(i)}, x_2^{(i)}, \dots, x_n^{(i)})^T$  della radice  $\xi = (\xi_1, \xi_2, \dots, \xi_n)^T$ . Supponiamo che ogni singola funzione  $f_j$  in (6.12) sia derivabile due volte, con derivate (parziali) seconde continue, in un intorno di  $\xi$  contenente  $x^{(i)}$ , in modo che risultino validi i seguenti sviluppi in serie di Taylor:

$$\left\{ \begin{array}{l} f_1(x^{(i)}) + (\xi_1 - x_1^{(i)}) \left( \frac{\partial f_1}{\partial x_1} \right)_{x=x^{(i)}} + \dots + (\xi_n - x_n^{(i)}) \left( \frac{\partial f_1}{\partial x_n} \right)_{x=x^{(i)}} + \text{termini ord. 2} = 0 \\ f_2(x^{(i)}) + (\xi_1 - x_1^{(i)}) \left( \frac{\partial f_2}{\partial x_1} \right)_{x=x^{(i)}} + \dots + (\xi_n - x_n^{(i)}) \left( \frac{\partial f_2}{\partial x_n} \right)_{x=x^{(i)}} + \text{termini ord. 2} = 0 \\ \dots \\ f_n(x^{(i)}) + (\xi_1 - x_1^{(i)}) \left( \frac{\partial f_n}{\partial x_1} \right)_{x=x^{(i)}} + \dots + (\xi_n - x_n^{(i)}) \left( \frac{\partial f_n}{\partial x_n} \right)_{x=x^{(i)}} + \text{termini ord. 2} = 0 \end{array} \right.$$

Trascurando i termini di ordine 2 otteniamo un sistema lineare la cui soluzione, se unica, non sarà  $\xi = (\xi_1, \xi_2, \dots, \xi_n)^T$  bensì  $x^{(i+1)} = (x_1^{(i+1)}, x_2^{(i+1)}, \dots, x_n^{(i+1)})^T$ :

$$(6.13) \quad \begin{aligned} J^{(i)} h^{(i)} &= -f(x^{(i)}) \\ x^{(i+1)} &= x^{(i)} + h^{(i)} \end{aligned} \quad i = 0, 1, 2, \dots$$

La matrice

$$J^{(i)} = J(x^{(i)}) = \begin{pmatrix} \left( \frac{\partial f_1}{\partial x_1} \right)_{x=x^{(i)}} & \left( \frac{\partial f_1}{\partial x_2} \right)_{x=x^{(i)}} & \dots & \left( \frac{\partial f_1}{\partial x_n} \right)_{x=x^{(i)}} \\ \left( \frac{\partial f_2}{\partial x_1} \right)_{x=x^{(i)}} & \left( \frac{\partial f_2}{\partial x_2} \right)_{x=x^{(i)}} & \dots & \left( \frac{\partial f_2}{\partial x_n} \right)_{x=x^{(i)}} \\ \dots & \dots & \dots & \dots \\ \left( \frac{\partial f_n}{\partial x_1} \right)_{x=x^{(i)}} & \left( \frac{\partial f_n}{\partial x_2} \right)_{x=x^{(i)}} & \dots & \left( \frac{\partial f_n}{\partial x_n} \right)_{x=x^{(i)}} \end{pmatrix}$$

è lo *Jacobiano* del sistema (6.12).

La (6.13) definisce il metodo di Newton in  $n$  variabili. Nota un'approssimazione iniziale  $x^{(0)}$  “sufficientemente” buona, il processo iterativo (6.13) determina una successione di approssimazioni  $\{x^{(i)}\}$  convergente alla radice incognita  $\xi$ . Come nel caso monodimensionale, l'ordine di convergenza è ancora  $p = 2$ <sup>(†)</sup>; tuttavia, il problema della convergenza

---

<sup>(†)</sup> Nel caso di una successione di vettori la relazione che definisce l'ordine di convergenza assume la forma seguente:

$$\lim_{n \rightarrow \infty} \frac{\|\xi - x^{(i+1)}\|}{\|\xi - x^{(i)}\|^p} = c$$

(connesso con la possibilità di avere o di determinare un'approssimazione iniziale  $x^{(0)}$  sufficientemente buona) nel caso di un sistema è molto più delicato. Osserviamo inoltre che quando lo Jacobiano pur essendo non singolare risulta “quasi” singolare, il sistema (6.13) si rivela malcondizionato. Con il metodo di Newton (6.13) ad ogni iterazione dobbiamo

1. valutare  $J^{(i)}$  e  $f(x^{(i)})$
2. risolvere il sistema  $J^{(i)}h^{(i)} = -f(x^{(i)})$
3. porre  $x^{(i+1)} = x^{(i)} + h^{(i)}$

I punti 1 e 2 risultano estremamente dispendiosi, tranne in quei casi in cui lo Jacobiano è sparso (cioè quando ogni funzione  $f_j$  dipende da poche variabili). Un altro inconveniente di questo metodo è la necessità di conoscere o poter valutare le  $n^2$  derivate parziali presenti in  $J^{(i)}$ . Al fine di ridurre il “costo” di ogni iterazione del metodo di Newton, sono state proposte diverse alternative, tutte rivolte alla ricerca di approssimazioni “efficienti”  $B^{(i)}$  dello Jacobiano  $J^{(i)}$ , con convergenza *superlineare* ( $1 < p < 2$ ).

L'approccio forse più ovvio per costruire un'approssimazione  $B^{(i)}$  dello Jacobiano consiste nell'approssimare le derivate parziali presenti in  $J^{(i)}$  con dei rapporti incrementali:

$$\left( \frac{\partial f_j}{\partial x_k} \right)_{x=x^{(i)}} \simeq (B^{(i)})_{jk} = \frac{f_j(x^{(i)} + e_k h_{kj}) - f_j(x^{(i)})}{h_{ki}}$$

dove  $(e_k)_j = \delta_{kj}$  e  $h_{ki} \in \mathbb{R}$ . Il metodo che così otteniamo è l'analogo in  $n$  dimensioni del metodo delle secanti visto nel paragrafo 6.2.2. Osserviamo che la scelta degli  $h_{ki}$  “opportuni” può risultare difficile: se sono troppo grandi la corrispondente approssimazione dello Jacobiano può risultare insufficiente, se sono troppo piccoli può invece insorgere il fenomeno della cancellazione numerica; inoltre la matrice  $B^{(i)}$  può risultare singolare o quasi singolare anche quando  $J^{(i)}$  non lo è.

Quando il calcolo dello Jacobiano  $J^{(i)}$  (o di una sua approssimazione  $B^{(i)}$ ) ad ogni iterazione si rivela eccessivamente oneroso, possiamo provare a usare lo stesso Jacobiano (o la sua approssimazione) valutato nel punto  $x = x^{(i)}$ , e quindi la sua decomposizione di Gauss, per tutte le successive  $m$  iterazioni, cioè porre

$$J^{(i+l)} = J^{(i)}, \quad l = 1, \dots, m$$

Giunti a  $x^{(i+m+1)}$  valutiamo lo Jacobiano in quest'ultimo punto e utilizziamo la nuova matrice e la sua decomposizione di Gauss per le altre  $m$  iterazioni; e così via. Questo modo di procedere diminuisce la velocità di convergenza del metodo; tuttavia esso comporta anche una riduzione notevole di calcolo.

Un'alternativa al metodo di Newton, più significativa della precedente, è senz'altro quella suggerita da C. G. Broyden (vedere [6.9, pag. 268]), che, in una forma lievemente

semplificata, si presenta come segue:

- 1: Scegliere una matrice iniziale  $B^{(0)}$ ; per esempio una matrice non singolare ottenuta dallo Jacobiano del sistema (nel punto iniziale  $x^{(0)}$ ) approssimando le derivate parziali con rapporti incrementali; oppure prendere  $B^{(0)} = I$ .
- (6.14) 2: **ciclo 1:**  $i = 1, \dots, i_{\max}$
- 3:  $B^{(i)} h^{(i)} = -f(x^{(i)}) \Rightarrow h^{(i)}$
- 4:  $x^{(i+1)} \leftarrow x^{(i)} + h^{(i)}$
- 5: **se**  $\|h^{(i)}\| \leq \text{toll} \|x^{(i+1)}\|$  **allora esci**
- 6:  $B^{(i+1)} \leftarrow B^{(i)} + f(x^{(i+1)}) h^{(i)\top} / (h^{(i)\top} h^{(i)})$
- 7: **fine ciclo 1**

Osserviamo che il rango della “correzione”  $B^{(i+1)} - B^{(i)}$ , del tipo  $ab^\top$  con  $a, b \in \mathbb{R}^n$ , è  $r = 1$ . Ciò semplifica notevolmente la risoluzione dei sistemi  $B^{(i)} h^{(i)} = -f(x^{(i)})$ ; infatti, nota la fattorizzazione di Gauss  $G^{(i)} B^{(i)} = U^{(i)}$ , le matrici  $G^{(i+1)}$ ,  $U^{(i+1)}$  della nuova decomposizione  $G^{(i+1)} B^{(i+1)} = U^{(i+1)}$  possono essere dedotte dalle precedenti  $G^{(i)}$ ,  $U^{(i)}$  mediante l’uso di tecniche ad hoc che rendono assai economica l’operazione. Ricordiamo infine che è stato dimostrato (vedere la bibliografia citata in [6.9, pag. 269]) che quando lo Jacobiano  $J(x)$  del sistema esiste ed è continuo in tutto un intorno  $I$  della soluzione  $\xi$ , è non singolare in  $\xi$  e lipschitziano in  $I$  (ovvero  $\|J(x_1) - J(x_2)\| \leq L \|x_1 - x_2\|$ ,  $x_1, x_2 \in I$ ), ed inoltre le approssimazioni iniziali  $x^{(0)}$  e  $B^{(0)}$  sono sufficientemente buone, tutte le matrici  $B^{(i)}$ ,  $i \geq 1$ , sono non singolari e l’algoritmo (6.14) produce una successione  $\{x^{(i)}\}$  convergente superlinearmente alla soluzione  $\xi$ .

Per una presentazione più completa e dettagliata dei metodi proposti per la risoluzione di sistemi non lineari consigliamo le letture [6.5], [6.8] e [6.11].

### 6.3.2 Metodi iterativi in generale

Seguendo l’idea già sviluppata nel paragrafo 6.2.4 nel caso di una singola equazione, riscriviamo il sistema (6.12) nella forma

$$(6.15) \quad x = \varphi(x)$$

con  $x = (x_1, x_2, \dots, x_n)^\top$  e  $\varphi(x) = (\varphi_1(x), \varphi_2(x), \dots, \varphi_n(x))^\top$ . Nota un’approssimazione  $x^{(0)}$  di una soluzione della (6.15), inneschiamo il seguente processo iterativo:

$$(6.16) \quad x^{(i+1)} = \varphi(x^{(i)}), \quad i = 0, 1, 2, \dots$$

Sotto quali ipotesi il metodo (6.16) genera una successione di approssimazioni  $x^{(1)}, x^{(2)}, \dots$  convergente alla soluzione  $\xi$ ? Le condizioni di convergenza sono del tutto simili a quelle già esposte nel caso di una singola equazione.

Il teorema seguente, valido ovviamente anche per le equazioni in una sola variabile, presenta una lieve generalizzazione rispetto a quello omologo presentato nel paragrafo 6.2.4.

**Teorema 6.2.** (vedi ad esempio, [6.9], pag. 251). *Data un'approssimazione iniziale  $x^{(0)}$  definiamo la successione  $x^{(i+1)} = \varphi(x^{(i)})$ ,  $i = 0, 1, 2, \dots$ . Supponiamo che esista un intorno  $S_r(x^{(0)}) = \{x \in \mathbb{R}^n : \|x - x^{(0)}\| < r\}$  e una costante  $0 < L < 1$  tali che*

- (i)  $\|\varphi(x) - \varphi(y)\| \leq L\|x - y\|$  per tutte le coppie  $x, y \in \bar{S}_r(x^{(0)}) = \{x \in \mathbb{R}^n : \|x - x^{(0)}\| \leq r\}$ ;
- (ii)  $\|\varphi(x^{(0)}) - x^{(0)}\| \leq (1 - L)r$ .

Allora

- (a) tutte le approssimazioni  $\{x^{(i)}\}$  appartengono a  $S_r(x^{(0)})$ ;
- (b) il processo iterativo converge alla radice  $\xi$ ,  $\xi = \varphi(\xi)$ , e  $\xi$  è l'unica radice presente in  $\bar{S}_r(x^{(0)})$ ;
- (c)  $\|\xi - x^{(i)}\| \leq \frac{L^i}{1-L} \|x^{(1)} - x^{(0)}\|$ .

Quando lo Jacobiano della funzione  $\varphi(x)$ ,  $J(x)$ , è definito ed è continuo in  $\bar{S}_r(x^{(0)})$ , e in tale intorno  $\|J(x)\| \leq L < 1$ , la condizione (i) del teorema è automaticamente soddisfatta. Ricordiamo inoltre che in alcune importanti applicazioni il sistema da risolvere si presenta nella forma seguente:

$$x = a + h\varphi(x)$$

dove  $a \in \mathbb{R}^n$  è un vettore noto e  $h$  è un parametro  $0 < h \ll 1$ . In questa situazione per soddisfare la condizione (i) suddetta è sufficiente che  $\|J(x)\| \leq L$  in  $\bar{S}_r(x^{(0)})$  e  $h$  sia scelto in modo che  $hL < 1$ .

## 6.4 Equazioni algebriche (a coefficienti reali)

La letteratura relativa ai metodi numerici per la determinazione delle radici di polinomi è assai vasta. In questo paragrafo noi ci limiteremo ad utilizzare metodi presentati in 6.2 per determinare singoli zeri reali, o al più coppie complesse coniugate di zeri, di polinomi a coefficienti reali.

Prima di descrivere l'implementazione dei metodi proposti, ricordiamo che esistono diversi teoremi che permettono di dedurre dalla conoscenza dei coefficienti  $\{a_i\}$  del polinomio

$$p_N(x) \equiv x^N + a_1 x^{N-1} + \cdots + a_{N-1} x + a_N$$

utili informazioni sulla localizzazione delle radici (vedere ad esempio [6.2]). In particolare ricordiamo che, posto  $r = \max_{1 \leq i \leq N} |a_i|$ , tutte le radici (reali e complesse) dell'equazione  $p_N(x) = 0$  sono contenute nel cerchio  $C = \{z \in \mathbb{C} : |z| \leq 1 + r\}$ .

### 6.4.1 Radici reali

I metodi iterativi presentati nel paragrafo 6.2 richiedono la valutazione del polinomio, ed eventualmente della sua derivata prima e seconda, nelle successive approssimazioni  $x_n$ .

Pertanto, descriviamo preliminarmente un algoritmo che consente di effettuare queste valutazioni in modo efficiente.

Il calcolo diretto di  $p_N(x) = x^N + a_1x^{N-1} + \dots + a_{N-1}x + a_N$ , utilizzando la relazione  $x^{k+1} = x \cdot x^k$ , richiede  $N$  addizioni e  $2N - 2$  moltiplicazioni. L'*algoritmo di Horner*

$$(6.17) \quad \begin{cases} p_0 = 1 \\ p_k = p_{k-1}x + a_k, & k = 1, 2, \dots, N \end{cases}$$

dove  $p_N \equiv p_N(x)$ , permette invece di ottenere lo stesso valore con sole  $N$  addizioni e  $N - 1$  moltiplicazioni. Tale algoritmo scaturisce dalla seguente riformulazione dell'espressione di  $p_N(x)$ :

$$a_N + x(a_{N-1} + x(a_{N-2} + \dots + x(a_1 + x) \dots))$$

Prima di procedere alla valutazione delle derivate di  $p_N(x)$ , dato un generico punto  $x_i$  consideriamo il rapporto

$$q_{N-1}(x) = \frac{p_N(x) - p_N(x_i)}{x - x_i}$$

e determiniamo i coefficienti  $\{b_k\}$  del *polinomio ridotto*

$$q_{N-1}(x) = x^{N-1} + b_1x^{N-2} + \dots + b_{N-2}x + b_{N-1}$$

L'identità

$$x^N + a_1x^{N-1} + \dots + a_{N-1}x + a_N = (x - x_i)(x^{N-1} + b_1x^{N-2}x + \dots + b_{N-1}) + p_N(x_i)$$

fornisce immediatamente per i  $\{b_k\}$  la seguente relazione ricorsiva:

$$(6.18) \quad \begin{cases} b_0 = 1 \\ b_k = b_{k-1}x_i + a_k, & k = 1, 2, \dots, N - 1 \end{cases}$$

del tutto simile alla (6.17). Anzi, per  $k = N$  abbiamo proprio  $b_N \equiv p_N(x_i)$ . Pertanto l'algoritmo (6.18), con  $k = 1, 2, \dots, N$ , non solo fornisce i coefficienti  $\{b_k\}$  del polinomio quoziante in

$$\frac{p_N(x)}{x - x_i} = q_{N-1}(x) + \frac{b_N}{x - x_i}$$

ma anche il resto della divisione  $b_N \equiv p_N(x_i)$ .

Sviluppiamo ora  $p_N(x)$  in serie di Taylor nell'intorno del punto  $x_i$

$$p_N(x) = p_N(x_i) + (x - x_i)p'_N(x_i) + \frac{(x - x_i)^2}{2!}p''_N(x_i) + \dots + \frac{(x - x_i)^N}{N!}p_N^{(N)}(x_i)$$

e scriviamo

$$q_{N-1}(x) = \frac{p_N(x) - p_N(x_i)}{x - x_i} = p'_N(x_i) + \frac{x - x_i}{2!}p''_N(x_i) + \dots + \frac{(x - x_i)^{N-1}}{N!}p_N^{(N)}(x_i)$$

Quest'ultima espressione valutata nel punto  $x = x_i$  ci dà

$$q_{N-1}(x_i) = \frac{p_N(x) - p_N(x_i)}{x - x_i} \Big|_{x=x_i} = p'_N(x_i)$$

Ricordando la (6.17) possiamo senz'altro concludere che

$$p'_N(x_i) = c_{N-1}$$

con

$$(6.19) \quad \begin{cases} c_0 = 1 \\ c_k = c_{k-1}x_i + b_k, & k = 1, 2, \dots, N-1 \end{cases}$$

In modo analogo possiamo dedurre una formula ricorsiva per il calcolo della derivata seconda  $p''_N(x_i)$ . Infatti

$$q_{N-1}(x) = p'_N(x_i) + \frac{x - x_i}{2!} p''_N(x_i) + \dots + \frac{(x - x_i)^{N-1}}{N!} p^{(N)}_N(x_i)$$

e

$$\frac{q_{N-1}(x) - q_{N-1}(x_i)}{x - x_i} = \frac{1}{2!} p''_N(x_i) + \dots + \frac{(x - x_i)^{N-2}}{N!} p^{(N)}_N(x_i)$$

quindi

$$p''_N(x_i) = 2 \frac{q_{N-1}(x) - q_{N-1}(x_i)}{x - x_i} \Big|_{x=x_i}$$

ossia

$$\begin{cases} d_0 = 1 \\ d_k = d_{k-1}x_i + c_k, & k = 1, 2, \dots, N-2 \\ p''_N(x_i) = 2d_{N-2} \end{cases}$$

Così proseguendo possiamo ottenere formule ricorsive analoghe per il calcolo di tutte le derivate successive  $p_N^{(l)}(x_i)$ ,  $l > 2$ .

Supponiamo di avere delle approssimazioni  $x_i^{(0)}$  delle radici  $x_i$  del polinomio  $p_N(x)$ . Scelto uno dei metodi presentati nel paragrafo 6.2, quello delle tangenti per esempio, come procediamo per il calcolo delle radici  $x_i$  con la precisione desiderata? Valutiamo dapprima la radice  $x_1$  con il metodo prescelto, utilizzando le formule ricorsive (6.18) e (6.19) per il calcolo di  $p_N(x_1^{(n)})$  e  $p'_N(x_1^{(n)})$ . Riapplichiamo poi il metodo scelto al polinomio ridotto

$$q_{N-1}(x) = \frac{p_N(x) - p_N(x_1)}{x - x_1}, \quad p_N(x_1) = 0$$

e calcoliamo la seconda radice  $x_2$ . Successivamente determiniamo il nuovo polinomio ridotto

$$q_{N-2}(x) = \frac{q_{N-1}(x) - q_{N-1}(x_2)}{x - x_2}, \quad q_{N-1}(x_2) = 0$$

e proseguiamo finché tutte le radici richieste non sono state calcolate.

Purtroppo le radici  $x_i$  calcolate non sono esatte, ma determinate con una certa tolleranza; inoltre non possiamo ignorare gli errori di arrotondamento commessi nel calcolo dei coefficienti dei successivi polinomi ridotti  $q_{N-1}(x)$ ,  $q_{N-2}(x)$ , ..., Pertanto, esiste il rischio (reale) che questi errori allontanino sempre di più gli zeri dei successivi polinomi ridotti da quelli del polinomio iniziale  $p_N(x)$ . L'analisi della propagazione degli errori nei polinomi ridotti suggerisce di seguire, quando possibile, la strategia seguente:

- (i) determinare le radici in ordine crescente (in modulo):  $|x_1| \leq |x_2| \leq |x_3| \leq \dots$ ;
- (ii) calcolare ogni radice con la massima precisione raggiungibile.

In ogni caso, dopo aver calcolato le approssimazioni delle radici  $x_i$  con la tecnica dei successivi polinomi ridotti, conviene prendere i valori così trovati come approssimazioni iniziali e procedere al loro raffinamento applicando il metodo scelto al polinomio iniziale  $p_N(x)$ ; di solito una sola iterazione del metodo è sufficiente.

#### 6.4.2 Metodo di Bairstow

I metodi delle secanti e di Newton-Raphson potrebbero essere usati anche per calcolare le radici complesse di un'equazione  $f(x) = 0$ . Ovviamente in tale situazione l'approssimazione iniziale  $x^{(0)}$  (e  $x^{(1)}$  nel caso delle secanti) deve essere complessa, altrimenti le formule predette produrrebbero valori  $x^{(1)}, x^{(2)}, \dots, x^{(n)}, \dots$  tutti reali, cioè approssimazioni al più di una eventuale radice reale. Prendere una  $x^{(0)}$  complessa significa però “fare l'ingresso” nel campo dei complessi  $\mathbb{C}$ .

Quando  $f(x)$  è un polinomio a coefficienti reali è possibile determinare le sue radici, reali e complesse, rimanendo sempre all'interno del campo dei reali  $\mathbb{R}$ . È sufficiente infatti osservare che tutte le radici complesse compaiono in coppie coniugate  $x_j \pm iy_j$  e che ad ogni coppia corrisponde un fattore quadratico  $x^2 + p_jx + q_j$  con coefficienti  $p_j$  e  $q_j$  reali. Possiamo allora pensare di decomporre il polinomio iniziale in fattori di questo tipo; la ricerca degli zeri di un trinomio di secondo grado è cosa che sappiamo fare molto bene.

Sia

$$p_N(x) = x^N + a_1x^{N-1} + \dots + a_{N-1}x + a_N$$

Dati due reali  $p$  e  $q$ , dividiamo  $p_N(x)$  per  $x^2 + px + q$ ; otteniamo

$$(6.20) \quad p_N(x) = (x^2 + px + q)(x^{N-2} + b_1x^{N-3} + \dots + b_{N-3}x + b_{N-2}) + Rx + S$$

dove  $b_k = b_k(p, q)$ ,  $R = R(p, q)$  e  $S = S(p, q)$ . Dobbiamo determinare quei valori (reali)  $p_0$  e  $q_0$  che rendono  $x^2 + p_0x + q_0$  divisore esatto di  $p_N(x)$ , ovvero risolvere il sistema non lineare

$$(6.21) \quad \begin{cases} R(p, q) = 0 \\ S(p, q) = 0 \end{cases}$$

Scelta un'approssimazione iniziale  $p^{(0)}, q^{(0)}$ , per esempio  $p^{(0)} = q^{(0)} = 0$ , applichiamo a tale sistema il metodo di Newton. Otteniamo

$$(6.22) \quad \begin{pmatrix} \frac{\partial R(p^{(n)}, q^{(n)})}{\partial p} & \frac{\partial R(p^{(n)}, q^{(n)})}{\partial q} \\ \frac{\partial S(p^{(n)}, q^{(n)})}{\partial p} & \frac{\partial S(p^{(n)}, q^{(n)})}{\partial q} \end{pmatrix} \begin{pmatrix} \Delta p^{(n)} \\ \Delta q^{(n)} \end{pmatrix} = - \begin{pmatrix} R(p^{(n)}, q^{(n)}) \\ S(p^{(n)}, q^{(n)}) \end{pmatrix}, \quad n = 0, 1, \dots$$

con  $p^{(n+1)} = p^{(n)} + \Delta p^{(n)}$  e  $q^{(n+1)} = q^{(n)} + \Delta q^{(n)}$ .

Per poter risolvere il suddetto sistema lineare occorre conoscere il termine noto e gli elementi dello Jacobiano. A tal fine, ricorrendo al principio di identità dei polinomi, dalla (6.20) deduciamo le relazioni

$$\begin{cases} a_1 = b_1 + p \\ a_2 = b_2 + pb_1 + q \\ a_k = b_k + pb_{k-1} + qb_{k-2}, \quad k = 3, \dots, N-2 \\ a_{N-1} = R + pb_{N-2} + qb_{N-3} \\ a_N = S + qb_{N-2} \end{cases}$$

che ci consentono di determinare ricorsivamente i coefficienti  $b_1, b_2, \dots, b_{N-2}$ ,  $R$  e  $S$ . Infatti, introducendo le quantità ausiliarie  $b_{-1} = 0$ ,  $b_0 = 1$  e  $b_{N-1}$ ,  $b_N$  abbiamo

$$(6.23) \quad \begin{cases} b_{-1} = 0 \\ b_0 = 1 \\ b_k = a_k - pb_{k-1} - qb_{k-2}, \quad k = 1, 2, \dots, N \\ R = b_{N-1} \\ S = b_N + pb_{N-1} \end{cases}$$

Successivamente, dalle ultime due relazioni in (6.23) otteniamo

$$\begin{aligned} \frac{\partial R}{\partial p} &= \frac{\partial b_{N-1}}{\partial p} & \frac{\partial R}{\partial q} &= \frac{\partial b_{N-1}}{\partial q} \\ \frac{\partial S}{\partial p} &= \frac{\partial b_N}{\partial p} + b_{N-1} + p \frac{\partial b_{N-1}}{\partial p} & \frac{\partial S}{\partial q} &= \frac{\partial b_N}{\partial q} + p \frac{\partial b_{N-1}}{\partial q} \end{aligned}$$

dove i termini  $\partial b_k / \partial p$  e  $\partial b_k / \partial q$ ,  $k = N-1, N$ , possono essere determinati utilizzando il

seguente processo ricorsivo

$$(6.24) \quad \begin{cases} \frac{\partial b_0}{\partial p} = \frac{\partial b_{-1}}{\partial p} = 0 \\ \frac{\partial b_k}{\partial p} = -b_{k-1} - p \frac{\partial b_{k-1}}{\partial p} - q \frac{\partial b_{k-2}}{\partial p}, \quad k = 1, 2, \dots, N \\ \frac{\partial b_0}{\partial q} = \frac{\partial b_{-1}}{\partial q} = 0 \\ \frac{\partial b_k}{\partial q} = -b_{k-2} - p \frac{\partial b_{k-1}}{\partial q} - q \frac{\partial b_{k-2}}{\partial q}, \quad k = 1, 2, \dots, N \end{cases}$$

conseguito derivando le (6.23).

Per semplificare le notazioni poniamo

$$c_{k-1} = -\frac{\partial b_k}{\partial p}, \quad k = 1, 2, \dots, N$$

e osserviamo (per induzione matematica) che

$$\frac{\partial b_k}{\partial q} = \frac{\partial b_{k-1}}{\partial p} = -c_{k-2}$$

Le (6.24) possono allora venire sintetizzate con la formula

$$(6.25) \quad \begin{cases} c_{-1} = 0 \\ c_0 = 1 \\ c_k = b_k - pc_{k-1} - qc_{k-2}, \quad k = 1, 2, \dots, N-1 \end{cases}$$

Pertanto, al sistema (6.22) possiamo dare la forma

$$(6.26) \quad \begin{pmatrix} c_{N-2} & c_{N-3} \\ c_{N-1} - b_{N-1} + pc_{N-2} & c_{N-2} + pc_{N-3} \end{pmatrix} \begin{pmatrix} \Delta p^{(n)} \\ \Delta q^{(n)} \end{pmatrix} = \begin{pmatrix} b_{N-1} \\ b_N + pb_{N-1} \end{pmatrix}$$

dove i coefficienti  $b_{N-1}$  e  $b_N$  vengono calcolati con la (6.23), mentre  $c_{N-3}$ ,  $c_{N-2}$  e  $c_{N-1}$  sono determinati dalla (6.25). Infine, osservando che la prima equazione del sistema (6.26) è

$$c_{N-2}\Delta p^{(n)} + c_{N-3}\Delta q^{(n)} = b_{N-1}$$

perveniamo ad un'ulteriore semplificazione:

$$(6.27) \quad \begin{aligned} \begin{pmatrix} c_{N-2} & c_{N-3} \\ c_{N-1} - b_{N-1} & c_{N-2} \end{pmatrix} \begin{pmatrix} \Delta p^{(n)} \\ \Delta q^{(n)} \end{pmatrix} &= \begin{pmatrix} b_{N-1} \\ b_N \end{pmatrix} \quad n = 0, 1, \dots \\ p^{(n+1)} &= p^{(n)} + \Delta p^{(n)} \\ q^{(n+1)} &= q^{(n)} + \Delta q^{(n)} \end{aligned}$$

Determinati  $p_0$  e  $q_0$  (o meglio, delle loro approssimazioni), consideriamo il polinomio ridotto  $q_{N-2}(x) = x^{N-2} + b_1 x^{N-3} + \dots + b_{N-3} x + b_{N-2}$  e procediamo nella ricerca di un suo fattore  $x^2 + p_1 x + q_1$ , prendendo, per esempio,  $p_1^{(0)} = p_0$  e  $q_1^{(0)} = q_0$ ; e così via. Alla fine otteniamo una decomposizione (a meno degli errori di arrotondamento e di troncamento) del polinomio iniziale del tipo

$$p_N(x) = (x^2 + p_0 x + q_0)(x^2 + p_1 x + q_1) \dots$$

## 6.5 Ottimizzazione

Un problema di grande interesse nelle applicazioni è quello dell'*ottimizzazione*. In questo paragrafo ci limitiamo a descrivere molto brevemente alcune situazioni di base, che possono essere riformulate come sistemi di equazioni non lineari e quindi risolte con i metodi visti nel paragrafo 6.3.

Sia data una funzione  $f : \mathbb{R}^n \rightarrow \mathbb{R}$ . Supposto  $f \in C^1(\mathbb{R}^n)$ , è noto dall'Analisi Matematica che i punti di stazionarietà locale (massimi, minimi, punti di sella) sono soluzione del seguente sistema (in generale non lineare):

$$(6.28) \quad \nabla f(x) = 0$$

ove  $\nabla f(x)$  denota il gradiente della funzione  $f$ :

$$\nabla f(x) = \left( \frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \right)^T$$

Pertanto, la determinazione di un punto di stazionarietà  $x^*$  può venire effettuata risolvendo, con uno dei metodi descritti nel paragrafo 6.3, il sistema (6.28). Per verificare poi se tale punto è un massimo, un minimo, oppure un punto di sella, occorrerà in generale esaminare la matrice hessiana  $H(x^*)$

$$(H(x^*))_{ij} = \frac{\partial^2 f(x^*)}{\partial x_i \partial x_j}, \quad i, j = 1, \dots, n$$

In questo modo possiamo, per esempio, risolvere il seguente problema di *ottimizzazione* (o di *minimizzazione*) *non vincolata*

$$(6.29) \quad \min_{x \in \mathbb{R}^n} f(x)$$

Anche un problema di *ottimizzazione vincolata* del tipo

$$(6.30) \quad \begin{cases} \min_{x \in \mathbb{R}^n} f(x) \\ g_1(x) = 0 \\ \vdots \\ g_m(x) = 0 \end{cases}$$

con  $f, g_i \in C^1(\mathbb{R}^n)$ , può essere ricondotto ad un sistema non lineare. A tale fine ricordiamo preliminarmente che in un punto  $\bar{x} \in \mathbb{R}^n$  i vincoli  $g_i(\bar{x}) = 0$ ,  $i = 1, \dots, m$ , sono detti *regolari* quando i vettori  $\nabla g_i(\bar{x})$ ,  $i = 1, \dots, m$ , sono linearmente indipendenti. Sussiste inoltre il seguente risultato fondamentale: se  $x^*$  è un punto di ottimalità del problema (6.30), e se i vincoli  $g_i(x)$ ,  $i = 1, \dots, m$ , sono regolari nel punto  $x^*$ , allora esistono  $m$  coefficienti  $\lambda_i^*$ ,  $i = 1, \dots, m$ , chiamati moltiplicatori di Lagrange, tali che

$$(6.31) \quad \nabla f(x^*) = \sum_{i=1}^m \lambda_i^* \nabla g_i(x^*)$$

Da questo risultato segue che, definita la funzione, detta *lagrangiana*,

$$(6.32) \quad L(x, \lambda) = f(x) - \sum_{i=1}^m \lambda_i g_i(x), \quad \lambda_i \in \mathbb{R}$$

si ha

$$\nabla_x L(x^*, \lambda^*) = 0, \quad \text{ovvero (6.31)}$$

e

$$\nabla_\lambda L(x^*, \lambda^*) = 0, \quad \text{ovvero } g_i(x^*) = 0, \quad i = 1, \dots, m$$

Con il simbolo  $\nabla_x (\nabla_\lambda)$  denotiamo il gradiente rispetto alla variabile  $x(\lambda)$ .

La relazione (6.31) ci ha così consentito di trasformare il problema con vincoli di uguaglianza (6.30) in un problema equivalente senza vincoli, associato alla nuova funzione (6.32), e quindi di ridurre quest'ultimo ad un sistema non lineare di  $n + m$  equazioni (le (6.31) e i vincoli in (6.30)) in  $n + m$  incognite ( $x^* e \{\lambda_i^*\}$ ).

Osserviamo infine che nella nostra descrizione non è stato limitativo considerare solo problemi di minimo in quanto un problema di massimo può sempre essere trasformato in uno equivalente di minimo semplicemente cambiando di segno la funzione  $f(x)$ . Anzi, ciò che abbiamo finora detto con riferimento ad un problema di minimizzazione vale anche nel caso più generale della ricerca di punti stazionari di  $f(x)$ .

Come è stato illustrato, un problema di minimo, con o senza vincoli di uguaglianza, può sempre essere ricondotto ad un sistema (quadrato) di equazioni non lineari. Vale però anche il viceversa: un sistema non lineare

$$(6.33) \quad \begin{cases} f_1(x) = 0 \\ \vdots \\ f_n(x) = 0 \end{cases}$$

può essere trasformato in un problema di minimo assoluto senza vincoli. Infatti, il problema di ottimizzazione

$$\min_{x \in \mathbb{R}^n} \sum_{i=1}^n [f_i(x)]^2$$

ha come punti di minimo assoluto proprio le soluzioni del sistema (6.33).

Ritorniamo, per semplicità, al problema (6.29) e risolviamolo applicando, per esempio, il metodo di Newton al sistema (6.28); otteniamo il seguente processo iterativo:

$$(6.34) \quad H(x^{(k)})(x^{(k+1)} - x^{(k)}) = -\nabla f(x^{(k)}), \quad k = 0, 1, \dots,$$

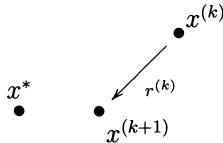
dove  $H(x^{(k)})$  denota la matrice hessiana calcolata nel punto  $x^{(k)}$ . La (6.34) può essere scritta nella forma

$$(6.35) \quad x^{(k+1)} = x^{(k)} + \lambda_k r^{(k)}, \quad k = 0, 1, \dots$$

con  $\lambda_k = 1$  e

$$(6.36) \quad r^{(k)} = -[H(x^{(k)})]^{-1} \nabla f(x^{(k)})$$

Nella (6.35) il termine  $r^{(k)}$  rappresenta una direzione di discesa; partendo da  $x^{(k)}$  ci muoviamo seguendo la direzione definita dal vettore  $r^{(k)}$  sino a raggiungere la nuova approssimazione  $x^{(k+1)}$  (che però nel caso specifico del metodo di Newton preferiamo determinare risolvendo la (6.34)).



In realtà l'espressione (6.35) definisce tutta una classe di metodi, ciascuno dei quali è caratterizzato dalla scelta della direzione di discesa ( $r^{(k)}$ ) verso il punto di minimo  $x^*$ . Il metodo di Newton è uno di tali metodi. Inoltre, fissata la direzione  $r^{(k)}$ , potremmo scegliere il parametro  $\lambda_k$  in modo che  $f(x^{(k)} + \lambda_k r^{(k)})$  sia minimo (risolvendo in questo caso un problema di minimo in una sola variabile).

In letteratura sono state proposte strategie alternative, e meno onerose dal punto di vista del costo computazionale, a quella di Newton (6.36). A tal proposito ricordiamo i metodi Quasi-Newton, ottenibili approssimando nella (6.36) l'inversa  $[H(x^{(k)})]^{-1}$  con una matrice la cui determinazione risulti meno onerosa. Il metodo di Cauchy, detto anche di *massima pendenza*, corrisponde alla scelta

$$r^{(k)} = -\nabla f(x^{(k)})$$

Esso può anche essere interpretato come un metodo Quasi-Newton ottenuto approssimando l'hessiana  $H(x^{(k)})$  con la matrice identità  $I$ . Ricordiamo che tale metodo era già stato menzionato nel paragrafo 3.3.4. La sua convergenza è generalmente lenta; per questo motivo esso viene utilizzato al più per ottenere una stima iniziale sufficientemente buona, tale da assicurare poi la convergenza di un metodo più rapido, per esempio di tipo Quasi-Newton.

Ricordiamo infine che il metodo del gradiente coniugato descritto nel paragrafo 3.3.4 rappresenta un esempio di scelta alternativa, e più efficiente, delle direzioni di discesa.

Per una descrizione più esaurente dei problemi di ottimizzazione e dei relativi metodi di risoluzione suggeriamo le letture [6.10], [6.11] e [6.12].

## Bibliografia

- [6.1] J. H. Wilkinson, *Rounding errors in algebraic processes*, Prentice-Hall, Englewood Cliffs, New Jersey, 1963.
- [6.2] M. Marden, *Geometry of polynomials*, Amer. Math. Soc., Providence, R. I., 1966.
- [6.3] J. F. Traub, *Iterative methods for the solution of equations*, Prentice-Hall, Englewood Cliffs, New Jersey, 1964.
- [6.4] A. S. Householder, *The numerical treatment of a single non-linear equation*, McGraw-Hill, New York, 1970.
- [6.5] J. M. Ortega, W. C. Rheinboldt, *Iterative solution of non-linear equations in several variables*, Academic Press, New York, 1970.
- [6.6] A. M. Ostrowski, *Solution of equations in euclidean and Banach spaces*, Academic Press, New York, 1973.
- [6.7] P. Henrici, *Applied and computational complex analysis*, Vol. I, John Wiley & Sons, New York, 1974.
- [6.8] W. C. Rheinboldt, *Methods for solving systems of nonlinear equations*, SIAM, Philadelphia, 1974.
- [6.9] J. Stoer, R. Bulirsch, *Introduction to numerical analysis*, Springer-Verlag, New York, 1980.
- [6.10] R. Fletcher, *Practical methods in optimization*, vol. 1 e 2, John Wiley & Sons, New York, 1980.
- [6.11] J. E. Dennis, R. B. Schnabel, *Numerical methods for unconstrained optimization and nonlinear equations*, Prentice-Hall, Englewood Cliff, New Jersey, 1983.
- [6.12] P. Boggs, R. Byrd, R. Schnabel, eds., *Numerical optimization 1984*, SIAM, Philadelphia, 1985.
- [6.13] P. Deuflhard, *Newton methods for nonlinear problems*, Springer-Verlag, Berlin, 2006.

## Esercizi proposti

**6.1.** Localizzare graficamente gli zeri delle funzioni seguenti:

$$f(x) = 4 \sin x + 1 - x, \quad f(x) = 3x^2 + \tan x, \quad f(x) = (x+1)e^{x-1} - 1$$

**6.2.** Utilizzare dapprima il metodo di bisezione per approssimare con errore assoluto  $\leq 0.02$  le radici delle equazioni seguenti:

$$\cos x = \ln x, \quad e^{-2x-1} = 1 - x,$$

Migliorare successivamente con il metodo delle secanti le approssimazioni conseguite, sino ad ottenere 5 cifre significative.

**6.3.** Supponendo di poter utilizzare solo le operazioni somma e prodotto, costruire un algoritmo che permetta di calcolare, con precisione di macchina, il rapporto  $a/b$  di due numeri assegnati.

**6.4.** Avendo a disposizione solo le quattro operazioni aritmetiche, utilizzare il metodo di Newton-Raphson per calcolare, con precisione di macchina, la radice quadrata di un numero positivo.

**6.5.** Determinare la radice  $\xi \cong 0.5$  dell'equazione  $x + \ln x = 0$  utilizzando le seguenti formule iterative:

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

Quale di queste tre formule produce una successione convergente? Quale delle tre è da preferirsi? Costruirne una quarta migliore di quelle date.

**6.6.** Proporre un processo iterativo convergente, del tipo  $x_{n+1} = g(x_n)$ , per calcolare la radice positiva dell'equazione

$$e^{-x} - 3x^2 = 0$$

**6.7.** Applicare il metodo di Newton-Raphson alla funzione

$$f(x) = \begin{cases} \sqrt{x} & \text{per } x \geq 0 \\ -\sqrt{-x} & \text{per } x < 0 \end{cases}$$

per il calcolo della radice  $\xi = 0$ . Esaminare il comportamento della successione  $\{x_n\}$ . Che cosa succede, e perché, nel caso della funzione

$$f(x) = \begin{cases} \sqrt[3]{x^3} & \text{per } x \geq 0 \\ -\sqrt[3]{-x^3} & \text{per } x < 0 \end{cases}$$

**6.8.** Calcolare con il metodo di Newton-Raphson lo zero del polinomio

$$p_4(z) = z^4 - z^3 - z^2 + 2z - 2$$

più vicino al punto  $z_0 = 0.4 + i0.8$ .

**6.9.** Supponendo che la radice incognita  $\xi$  dell'equazione  $f(x) = 0$  sia semplice, cioè  $f(\xi) \neq 0$ , dimostrare che il metodo di Steffensen ha ordine di convergenza  $p = 2$ .

**6.10.** Dimostrare il teorema 6.2.

**6.11.** Determinare i valori dell'incognita  $c_f$  nell'equazione

$$\sqrt{\frac{1}{c_f}} = -0.4 + 1.74 \ln(R\sqrt{c_f})$$

corrispondenti ai valori  $R = 10^4, 10^5, 10^6$ .

**6.12.** Includere nell'algoritmo delle potenze per il calcolo dell'autovalore di modulo massimo di una matrice reale (vedi paragrafo 4.2) la formula di accelerazione di Aitken. Sperimentare il nuovo algoritmo su alcuni problemi test.

**6.13.** Costruire un algoritmo efficiente che permetta di determinare, con 1a sola aritmetica reale, le tre radici del generico polinomio di grado 3

$$x^3 + a_1x^2 + a_2x + a_3$$

**6.14.** Ricordando che il polinomio ortogonale di Chebyshev di prima specie  $y = T_n(x)$  (vedi 228) è soluzione della seguente equazione differenziale

$$(1 - x^2)y'' - xy' + n^2y = 0$$

utilizzare il metodo di ordine 3 presentato a pagina 194 per determinare lo zero  $x_1$  di  $T_{10}(x)$  più vicino a  $x = 1$ . Confrontare il risultato ottenuto con il valore esatto  $x_1 = \cos(\pi/20)$ .

**6.15.** Ricavare il metodo di Muller descritto a pagina 194.

**6.16.** Utilizzare il metodo Newton per determinare la soluzione  $(1, -1, -1)$  del sistema

$$\begin{aligned} x_1^2 + 2x_1x_2 + x_3 &= 0 \\ x_2^3 + x_3^2 &= 0 \\ x_1x_3 &= 1 \end{aligned}$$

Sperimentare, prendendo approssimazioni iniziali diverse, il comportamento del metodo.

**6.17.** Dimostrare che con la seguente formula di Newton modificata

$$x_{n+1} = x_n - \lambda \frac{f(x_n)}{f'(x_n)}, \quad n = 0, 1, \dots$$

dove supponiamo  $f(x_n) \neq 0$  e  $f'(x_n) \neq 0$ , e  $\lambda$  è un parametro positivo sufficientemente piccolo, abbiamo sempre

$$|f(x_{n+1})| < |f(x_n)|$$

**6.18.** Implementare il metodo di Bairstow. Trovare le radici del polinomio

$$3x^6 + 3x^5 + 5x^4 + 2x^3 + x^2 - x - 1$$

**6.19.** Trovare il minimo della seguente funzione

$$f(x) = x_1^4 + x_1x_2 + (1 + x_2)^2$$

**6.20.** Risolvere il seguente problema di minimo vincolato:

$$\begin{cases} \min(x_1^2 + x_2^2) \\ x_1^2 + 2x_1x_2 + 3x_2^2 - 1 = 0 \end{cases}$$



# Capitolo 7

## Calcolo di integrali

### 7.1 Preliminari. Formule di quadratura interpolatorie

Tema principale di questo capitolo è la valutazione numerica di integrali definiti

$$I(f) = \int_a^b f(x) dx$$

Spesso è impossibile determinare  $I(f)$  per via analitica; ma anche quando tale via risulta percorribile, l'espressione finale è sovente così complessa rispetto alla funzione integranda da suggerire l'uso di approcci più semplici. Inoltre l'eventuale soluzione analitica potrebbe coinvolgere funzioni elementari e non, che devono poi venire valutate (e quindi approssimate). Se invece la funzione  $f(x)$  è nota solo per punti, oppure valutabile per ogni valore dell'argomento  $x$  mediante una routine, l'approccio analitico non può neppure essere preso in considerazione. Pertanto, supponendo di conoscere o di poter valutare la funzione integranda  $f(x)$  in punti  $\{x_i\}$ , prefissati oppure da noi scelti, esaminiamo la costruzione di *formule*, che denomineremo *di quadratura*, del tipo

$$(7.1) \quad \int_a^b f(x) dx \approx \sum_{i=1}^n w_i f(x_i)$$

I numeri (reali)  $\{x_i\}$  e  $\{w_i\}$  vengono chiamati rispettivamente *nodi* e *pesi* della formula di quadratura.

Assegnati i nodi  $\{x_i\}$ , *distinti* e preferibilmente nell'intervallo  $[a, b]$  che per ora supponiamo limitato, l'idea più semplice ed immediata che ci consente di costruire un'espressione di forma (7.1) è l'approssimazione della funzione integranda  $f(x)$  con il polinomio di grado  $n - 1$ ,  $L_{n-1}(f; x)$ , unico, che interpola quest'ultima nei nodi  $\{x_i\}$ :

$$\int_a^b f(x) dx = \int_a^b [L_{n-1}(f; x) + E_n(f; x)] dx = \int_a^b L_{n-1}(f; x) dx + \int_a^b E_n(f; x) dx$$

con

$$L_{n-1}(f; x_i) = f(x_i), \quad i = 1, \dots, n$$

Infatti, rappresentando  $L_{n-1}(f; x)$  nella forma di Lagrange

$$L_{n-1}(f; x) = \sum_{i=1}^n l_i(x) f(x_i)$$

otteniamo

$$(7.2) \quad \int_a^b f(x) dx = \sum_{i=1}^n w_i f(x_i) + R_n(f)$$

dove

$$(7.3) \quad w_i = \int_a^b l_i(x) dx$$

Il termine

$$R_n(f) = \int_a^b E_n(f; x) dx$$

rappresenta l'errore della formula di quadratura.

Le formule costruite in questo modo vengono chiamate *interpolatorie*. Esse sono “esatte”, cioè l'errore  $R_n(f)$  è nullo, ogniqualvolta  $f(x)$  è un polinomio di grado  $\leq n-1$ ; infatti quando  $f \in \mathbb{P}_{n-1}$  abbiamo  $E_n(f; x) \equiv 0$ . In particolare,  $R_n(f) = 0$  quando  $f(x) = 1, x, \dots, x^{n-1}$ , ossia

$$(7.4) \quad \left\{ \begin{array}{l} w_1 + w_2 + \cdots + w_n = m_0 \\ w_1 x_1 + w_2 x_2 + \cdots + w_n x_n = m_1 \\ \dots \\ w_1 x_1^{n-1} + w_2 x_2^{n-1} + \cdots + w_n x_n^{n-1} = m_{n-1} \end{array} \right.$$

dove  $m_k = \int_a^b x^k dx$ .

Pertanto i pesi  $\{w_i\}$  della (7.2) sono univocamente definiti dal suddetto sistema, non singolare perché di tipo Vandermonde. L'unicità della soluzione  $\{w_i\}$  di (7.4) ci consente di affermare che, fissati i nodi  $x_i$ ,  $i = 1, \dots, n$ , distinti, la *corrispondente formula interpolatoria* (7.2) è *unica*; cioè non esistono in (7.2) altre scelte dei pesi  $\{w_i\}$ , diverse dalla (7.3), che producono altre formule di tipo (7.1) esatte ogniqualvolta la funzione integranda  $f(x)$  è un polinomio di grado  $\leq n-1$ .

Come già è stato osservato nel capitolo 5, la soluzione del sistema (7.4), almeno per valori di  $n$  non piccolissimi, risulta mal condizionata. Anche la rappresentazione (7.3) non è sempre la più efficiente per il calcolo di  $w_i$ . Per scelte particolari dei nodi  $\{x_i\}$ , soprattutto quando questi ultimi coincidono con gli zeri di un polinomio ortogonale (vedi paragrafo 7.3), esiste la possibilità di riformulare le espressioni dei pesi  $\{w_i\}$  in modo da giungere alla loro determinazione con algoritmi efficienti e numericamente stabili.

Un concetto utile per confrontare il grado di accuratezza (“di tipo polinomiale”) delle diverse formule di quadratura, interpolatorie e non, è il seguente:

**Definizione 7.1.** Una formula di quadratura ha grado di precisione  $d$  se è esatta quando la funzione integranda  $f(x)$  è un polinomio qualsiasi di grado  $\leq d$  ed inoltre esiste almeno un polinomio di grado  $d+1$  per cui l’errore  $R_n(f)$  risulta non nullo<sup>(†)</sup>.

Allora ogni formula (7.2) di tipo interpolatorio ha grado di precisione almeno  $n-1$ .

Prima di proseguire nella costruzione di altre formule di quadratura osserviamo che quando l’intervallo di integrazione  $(a, b)$  è limitato, con un semplice cambiamento (lineare) di variabile possiamo sempre trasformare il corrispondente integrale in un altro integrale avente lo stesso valore ma definito in un intervallo base di riferimento;  $(-1, 1)$  per esempio. Ciò ci consente di standardizzare la costruzione delle formule per il calcolo dei predetti integrali associandole tutte all’intervallo  $(-1, 1)$ :

$$\int_{-1}^1 f(x) dx \approx \sum_{i=1}^n w_i f(x_i)$$

Risulta poi immediato ottenere da quest’ultima la corrispondente formula per l’intervallo (limitato)  $(a, b)$ :

$$\int_a^b f(t) dt \approx \frac{b-a}{2} \sum_{i=1}^n w_i f\left(\frac{b-a}{2}x_i + \frac{b+a}{2}\right)$$

Supponiamo ora di voler valutare l’integrale  $I(f)$  con una formula di quadratura costruita su  $n$  nodi (distinti); supponiamo inoltre che la funzione integranda  $f(x)$  o una delle sue prime derivate presenti delle singolarità oppure punti di discontinuità nell’intervallo  $[a, b]$ . Supponiamo infine che  $f(x)$  sia fattorizzabile nella forma  $f(x) = w(x)g(x)$ , dove  $w(x)$  è una funzione di forma semplice contenente le singolarità di  $f(x)$ , mentre  $g(x)$  è la parte più regolare di  $f(x)$ ; per esempio,

$$\begin{aligned} f(x) &= \frac{x^2 e^{-x}}{\sqrt{x-a}} = \frac{1}{\sqrt{x-a}} x^2 e^{-x} \\ w(x) &= \frac{1}{\sqrt{x-a}} \\ g(x) &= x^2 e^{-x} \end{aligned}$$

Generalizzando il procedimento seguito per ottenere le (7.2), (7.3) possiamo, almeno in teoria, costruire formule di tipo interpolatorio per integrali posti nella forma

$$\int_a^b w(x)g(x) dx$$

---

(†) Se il grado di precisione della formula è  $d$ , certamente  $R_n(x^{d+1}) \neq 0$ .

dove l'intervallo  $(a, b)$  può anche essere illimitato. È infatti sufficiente utilizzare la formula di interpolazione di Lagrange

$$g(x) = \sum_{i=1}^n l_i(x)g(x_i) + E_n(g; x)$$

per pervenire alla quadratura

$$(7.5) \quad \int_a^b w(x)g(x) dx = \sum_{i=1}^n w_i g(x_i) + R_n(g)$$

con

$$w_i = \int_a^b w(x)l_i(x) dx \quad \text{e} \quad R_n(g) = \int_a^b w(x)E_n(g; x) dx$$

Ovviamente la funzione  $w(x)$  deve essere tale da garantire l'esistenza degli integrali coinvolti e permettere la costruzione dei pesi  $w_i$ . La formula (7.5) ha grado di precisione almeno  $n - 1$ , nel senso che in essa  $R_n(g) = 0$  ogniqualvolta  $g(x)$  è un polinomio di grado  $\leq n - 1$ . Osserviamo che la (7.2) altro non è che una formula di tipo (7.5) con  $w(x) \equiv 1$ . Pertanto d'ora in avanti faremo riferimento a *formule interpolatorie pesate*

$$\int_a^b w(x)f(x) dx = \sum_{i=1}^n w_i f(x_i) + R_n(f)$$

supporremo inoltre i nodi  $\{x_i\}$  ordinati nel modo seguente:

$$x_1 < x_2 < \cdots < x_{n-1} < x_n$$

Di solito la *funzione peso*  $w(x)$  ha segno costante in tutto  $(a, b)$ .

Osserviamo infine che non è difficile verificare che quando i nodi  $\{x_i\}$  della (7.1) sono simmetrici rispetto al punto medio  $c = (a + b)/2$ , cioè  $\frac{1}{2}(x_i + x_{n+1-i}) = c$ ,  $i = 1, 2, \dots, n$ , e  $w(c + x) = w(c - x)$ , i pesi  $\{w_i\}$  risultano anch'essi simmetrici, ossia  $w_i = w_{n+1-i}$ ,  $i = 1, 2, \dots, n$ , e la quadratura viene definita simmetrica.

Affinché la formula di quadratura (7.1) definisca una “buona” *discretizzazione* dell'integrale è necessario che

$$\lim_{n \rightarrow \infty} \sum_{i=1}^n w_i f(x_i) = \int_a^b w(x)f(x) dx$$

in questo caso diciamo che la formula è *convergente*. Esistono condizioni che assicurano la convergenza, per intervalli sia finiti che illimitati; per esempio quando  $(a, b)$  è limitato e  $f \in C[a, b]$  la convergenza è garantita se

$$\sum_{i=1}^n |w_i| \leq K$$

dove  $K$  è una costante (indipendente da  $n$ ). La dimostrazione di quest'ultimo risultato è un'applicazione assai semplice del teorema di approssimazione di Weierstrass. Anzi, utilizzando il teorema di Jackson [7.6] è possibile dimostrare che quando  $f \in C^k[a, b]$

$$|R_n(f)| \leq \frac{A}{n^k}$$

Pertanto, più regolare è la funzione  $f(x)$  più rapida è la convergenza della quadratura al valore esatto dell'integrale; di qui l'importanza di poter<sup>(†)</sup> individuare l'eventuale parte non regolare  $w(x)$  della funzione integranda.

Finora l'unica condizione imposta ai nodi  $\{x_i\}$  è che essi siano distinti. Nei paragrafi che seguono esamineremo con maggiori dettagli formule associate alle seguenti due scelte di nodi:

- (i) nodi equidistanti (formule di Newton-Cotes)
- (ii) nodi coincidenti con gli zeri di polinomi ortogonali (formule Gaussiane)

## 7.2 Formule di Newton-Cotes

Nell'intervallo  $[a, b]$ , che supponiamo limitato, prendiamo  $n$  nodi equidistanti

$$x_i = a + (i - 1)h, \quad i = 1, 2, \dots, n, \quad h = \frac{b - a}{n - 1}$$

e costruiamo la corrispondente formula di tipo interpolatorio, che chiameremo di Newton-Cotes,

$$(7.6) \quad \int_a^b f(x) dx = \sum_{i=1}^n w_i f(a + (i - 1)h) + R_n(f)$$

Le costanti

$$c_i = \frac{w_i}{b - a}, \quad i = 1, \dots, n$$

vengono chiamate *numeri di Cotes*; esse coincidono con i pesi della formula (7.6) associata all'intervallo  $[0, 1]$ , e godono delle seguenti proprietà:

$$c_i = c_{n+1-i}, \quad i = 1, 2, \dots, n \quad \text{e} \quad \sum_{i=1}^n c_i = 1$$

---

(†) Ribadiamo che tale fattorizzazione ha senso solo quando la forma di  $w(x)$  consente la determinazione effettiva dei pesi  $w_i$ .

Un esame più attento, tutt’altro che banale, dell’errore  $R_n(f)$  permette di dedurre (vedere [2, pag. 163]) le seguenti rappresentazioni:

$$\begin{aligned} R_n(f) &= K_n h^{n+1} \frac{f^{(n)}(\xi_1)}{n!} && \text{quando } f \in C^n[a, b] \quad \text{e } n \text{ è pari} \\ R_n(f) &= K_{n+1} h^{n+2} \frac{f^{(n+1)}(\xi_2)}{(n+1)!} && \text{quando } f \in C^{n+1}[a, b] \quad \text{e } n \text{ è dispari} \end{aligned}$$

con

$$K_n = \int_0^{n-1} t(t-1)\dots(t-n+1) dt \quad \text{e} \quad K_{n+1} = \int_0^{n-1} t^2(t-1)\dots(t-n+1) dt$$

Osserviamo che quando  $n$  è pari il grado di precisione della formula è  $n - 1$ , mentre quando  $n$  è dispari il grado di precisione è  $n$ . In generale quindi le formule con un numero dispari di nodi sono da preferirsi.

Le formule che abbiamo costruito sono di *tipo chiuso*, in quanto includono tra i loro nodi anche gli estremi di integrazione. In modo perfettamente analogo possiamo definire formule di tipo aperto prendendo come nodi i punti

$$x_i = a + ih, \quad i = 1, 2, \dots, n, \quad h = \frac{b-a}{n+1}$$

I pesi  $c_i$  delle formule di Newton-Cotes, sia di tipo chiuso che di tipo aperto, per i primi valori di  $n$  sono reperibili in [7.2].

Le formule di Newton-Cotes di regola non vengono applicate direttamente all’integrale in questione. Esse sono generalmente scelte con pochi nodi e utilizzate per la costruzione di schemi di calcolo di tipo composto, quali quelli descritti nel paragrafo 7.7. Tuttavia esse appaiono oramai superate. Per gli scopi predetti le formule di Newton-Cotes più note sono certamente quelle di tipo chiuso con 2 e 3 nodi:

(i) *formula del trapezio*

$$\int_a^b f(x) dx = \frac{h}{2} [f(a) + f(b)] - \frac{h^3}{12} f''(\xi_1), \quad a < \xi_1 < b, \quad h = b - a$$

(ii) *formula di Simpson*

$$\int_a^b f(x) dx = \frac{h}{3} [f(a) + 4f(a+h) + f(b)] - \frac{h^5}{90} f^{(4)}(\xi_2), \quad a < \xi_2 < b, \quad h = \frac{b-a}{2}$$

Formule di quadratura interpolatorie con nodi equidistanti, non necessariamente tutti interni all’intervallo di integrazione, verranno da noi utilizzate (vedere il paragrafo 8.3.2) nella costruzione di metodi numerici per la risoluzione di equazioni differenziali ordinarie.

### 7.3 Polinomi ortogonali

Sia data una funzione peso  $w(x)$  non negativa nell'intervallo finito o infinito  $(a, b)$  e non identicamente nulla, e supponiamo che tutti i momenti

$$m_k = \int_a^b w(x)x^k dx, \quad k = 0, 1, 2, \dots$$

esistano. Un sistema di polinomi  $\{P_0(x), P_1(x), \dots, P_n(x), \dots\}$ , con  $P_n(x) = k_{n,0}x^n + k_{n,1}x^{n-1} + \dots + k_{n,n}$  e  $k_{n,0} \neq 0$ , è detto *ortogonale* in  $(a, b)$  rispetto alla funzione peso  $w(x)$  se

$$\begin{aligned} \int_a^b w(x)P_n(x)P_m(x) dx &= 0 \quad \text{per } n \neq m \\ &\neq 0 \quad \text{per } n = m \end{aligned}$$

In questo caso i polinomi  $P_n(x)$  vengono denominati ortogonali. Ovviamente

$$h_n = \int_a^b w(x)P_n^2(x) dx > 0, \quad n = 0, 1, 2, \dots$$

Il sistema è detto ortonormale se  $h_n = 1$ ,  $n = 0, 1, 2, \dots$

L'intervallo  $(a, b)$  e la funzione  $w(x)$  definiscono univocamente i polinomi  $P_n(x)$ , a meno di fattori costanti non nulli; infatti la successione  $\{c_0P_0(x), c_1P_1(x), \dots, c_nP_n(x), \dots\}$ , con  $c_0, c_1, \dots, c_n, \dots$  costanti arbitrarie non nulle, rappresenta lo "stesso" sistema di polinomi ortogonali. Quando il coefficiente  $k_{n,0}$  di  $x^n$  in  $P_n(x)$  vale 1 il polinomio viene denominato *monico*.

Enunciamo ora alcuni teoremi fondamentali riguardanti tali polinomi; le dimostrazioni di questi risultati possono essere reperite, per esempio, nei testi [7.6] e [7.7].

**Teorema 7.1.** *Ogni sistema di polinomi ortogonali  $\{P_n(x)\}$  soddisfa una relazione di ricorrenza a tre termini del tipo:*

$$(7.7) \quad P_{n+1}(x) = (A_n x + B_n)P_n(x) - C_n P_{n-1}(x), \quad n = 1, 2, \dots$$

con  $C_n > 0$ . Inoltre, risulta

$$A_n = \frac{k_{n+1,0}}{k_{n,0}}, \quad B_n = A_n \left( \frac{k_{n+1,1}}{k_{n+1,0}} - \frac{k_{n,1}}{k_{n,0}} \right), \quad C_n = \frac{A_n}{A_{n-1}} \frac{h_n}{h_{n-1}}$$

Questo risultato ci consente di affermare che il sistema  $\{P_n(x)\}$  definito dalla coppia  $\{(a, b), w(x)\}$ , e dalla normalizzazione scelta, può essere univocamente individuato dall'insieme dei coefficienti  $\{(A_n, B_n, C_n)\}$  della relazione (7.7). Di solito si preferisce definire implicitamente i polinomi ortogonali proprio mediante i coefficienti predetti: i due polinomi iniziali  $P_0(x)$ ,  $P_1(x)$  e le terne  $\{(A_n, B_n, C_n), n = 1, \dots, N-1\}$  individuano i polinomi  $P_2(x), P_3(x), \dots, P_N(x)$ .

**Teorema 7.2.** Per ogni intero  $n \geq 1$  il polinomio ortogonale  $P_n(x)$  possiede  $n$  zeri reali, distinti e tutti contenuti in  $(a, b)$ . Inoltre, gli zeri di  $P_n(x)$  si alternano con quelli di  $P_{n-1}(x)$ ; ossia tra due zeri consecutivi di  $P_n(x)$  esiste un sol zero di  $P_{n-1}(x)$ .

Prima di presentare il prossimo teorema ricordiamo che, scelto un sistema di polinomi ortogonali, il generico polinomio  $Q_m(x)$  di grado  $m$  può venire rappresentato (univocamente) nella forma seguente:

$$(7.8) \quad Q_m(x) = \sum_{k=0}^m d_k P_k(x)$$

con

$$d_k = \frac{\int_a^b w(x) Q_m(x) P_k(x) dx}{\int_a^b w(x) P_k^2(x) dx}, \quad k = 0, 1, \dots, m$$

L'espressione (7.8) e la definizione stessa di ortogonalità conducono al risultato che segue.

**Teorema 7.3.** Per ogni polinomio  $q(x)$  di grado  $\leq n - 1$  abbiamo

$$\int_a^b w(x) P_n(x) q(x) dx = 0$$

In particolare

$$\int_a^b w(x) P_n(x) x^k dx = 0, \quad k = 0, 1, \dots, n - 1$$

quest'ultima relazione definisce univocamente, a meno di una costante moltiplicativa, il polinomio ortogonale  $P_n(x)$ .

Nota la rappresentazione (7.8) di un polinomio  $Q_m(x)$ , ossia noti i coefficienti  $\{d_k\}$ , e assegnato un punto  $\bar{x}$ , per valutare  $Q_m(\bar{x})$  non è necessario determinare i singoli valori  $P_k(\bar{x})$ ,  $k = 0, 1, \dots, m$ . Un procedimento di calcolo più efficiente e stabile è l'*algoritmo di Clenshaw*, che utilizza proprio la relazione ricorsiva (7.7) per definire implicitamente i  $P_k(x)$  e giungere al risultato finale  $Q_m(\bar{x})$  senza effettuare la valutazione esplicita dei  $P_k(\bar{x})$  stessi:

- 1:  $y_{m+2} \leftarrow 0$
- 2:  $y_{m+1} \leftarrow 0$
- 3:  $y_k \leftarrow (A_k \bar{x} + B_k) y_{k+1} - C_{k+1} y_{k+2} + d_k, \quad k = m, m-1, \dots, 1$
- 4:  $Q_m(\bar{x}) \equiv y_0 \leftarrow P_1(\bar{x}) y_1 + P_0(\bar{x})(-C_1 y_2 + d_0)$

▷ **Osservazione.** Quasi sempre i polinomi  $Q_m(x)$  vengono da noi espressi come combinazione lineare delle potenze di  $x$ :

$$Q_m(x) = a_0 x^m + a_1 x^{m-1} + \cdots + a_{m-1} x + a_m$$

Questa rappresentazione, pur essendo comoda per alcune ragioni, può non esserlo per altre; per esempio essa può rendere mal condizionato il problema del calcolo dei suoi zeri. Per ovviare a inconvenienti di questo tipo, e anche perché in molte applicazioni la manipolazione analitica delle espressioni e la stessa determinazione dei coefficienti incogniti nella rappresentazione di  $Q_m(x)$  risultano semplificate dalle eventuali proprietà di ortogonalità della base scelta, spesso conviene rappresentare  $Q_m(x)$  come combinazione lineare di polinomi ortogonali appartenenti ad un sistema  $\{P_n(x)\}$  ritenuto conveniente per il problema in esame.  $\triangleleft$

I teoremi presentati in questo paragrafo definiscono alcune proprietà essenziali dei polinomi ortogonali. Le caratteristiche di tali polinomi sono tuttavia numerosissime (vedere ad esempio [7.7]) ed il ruolo che essi assumono in molte applicazioni è fondamentale. Noi li abbiamo introdotti in due sole applicazioni: quella dei minimi quadrati (paragrafo 5.10) e quella delle quadrature Gaussiane (paragrafo 7.4). Al lettore interessato all'argomento suggeriamo le letture [7.7] e [7.10].

I polinomi ortogonali più noti e più utilizzati sono quelli denominati *classici*:

(i) Polinomi di *Legendre*  $P_n(x)$ :  $w(x) = 1$ ,  $(a, b) = (-1, 1)$

$$\begin{cases} P_0(x) = 1, & P_1(x) = x \\ (n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x), & n = 1, 2, \dots \end{cases}$$

$$k_{n,0} = \frac{(2n)!}{2^n(n!)^2}, \quad h_n = \frac{2}{2n+1}$$

(ii) Polinomi di *Jacobi*  $P_n^{(\alpha,\beta)}(x)$ :  $w(x) = (1-x)^\alpha(1+x)^\beta$ ,  $\alpha, \beta > -1$ ,  $(a, b) = (-1, 1)$

$$\begin{cases} P_0(x) = 1, & P_1(x) = [1 + \frac{1}{2}(\alpha + \beta)]x + \frac{1}{2}(\alpha - \beta) \\ 2(n+1)(n+\alpha+\beta+1)(2n+\alpha+\beta)P_{n+1}^{(\alpha,\beta)}(x) = \\ (2n+\alpha+\beta+1)[(\alpha^2 - \beta^2) + (2n+\alpha+\beta+2)(2n+\alpha+\beta)x]P_n^{(\alpha,\beta)}(x) - \\ 2(n+\alpha)(n+\beta)(2n+\alpha+\beta+2)P_{n-1}^{(\alpha,\beta)}(x), & n = 1, 2, \dots \end{cases}$$

$$k_{n,0} = 2^{-n} \left[ \binom{n+\alpha}{n} + \binom{n+\beta}{n} \right]$$

$$h_n = \frac{2^{\alpha+\beta+1}}{2n+\alpha+\beta+1} \cdot \frac{\Gamma(n+\alpha+1)\Gamma(n+\beta+1)}{n!\Gamma(n+\alpha+\beta+1)}$$

Casi particolari:

- $\alpha = \beta$ : polinomi *ultrasferici*  $P_n^{(\lambda)}(x)$ ,  $\lambda = \alpha + \frac{1}{2} > -\frac{1}{2}$ ;

$$P_n^{(\lambda)}(x) = \frac{\Gamma(\alpha+1)\Gamma(n+2\alpha+1)}{\Gamma(2\alpha+1)\Gamma(n+\alpha+1)} P_n^{(\alpha,\alpha)}(x)$$

- $\alpha = \beta = -\frac{1}{2}$ : polinomi di *Chebyshev di 1<sup>a</sup> specie*  $T_n(x) = \frac{n}{2} \lim_{\lambda \rightarrow 0} \lambda^{-1} P_n^\lambda(x)$

- $\alpha = \beta = \frac{1}{2}$ : polinomi di *Chebyshev di 2<sup>a</sup> specie*  $U_n(x) = P_n^{(1)}(x)$

(iii) Polinomi di *Laguerre*  $L_n(x)$ :  $w(x) = e^{-x}$ ,  $(a, b) = (0, \infty)$

$$\begin{cases} L_0(x) = 1, \\ L_1(x) = 1 - x \\ (n+1)L_{n+1}(x) = (2n+1-x)L_n(x) - nL_{n-1}(x), n = 1, 2, \dots \end{cases}$$

$$k_{n,0} = \frac{(-1)^n}{n!}, \quad h_n = 1$$

(iv) Polinomi di *Hermite*  $H_n(x)$ :  $w(x) = e^{-x^2}$ ,  $(a, b) = (-\infty, \infty)$

$$\begin{aligned} H_0(x) &= 1 \\ H_1(x) &= 2x \\ H_{n+1}(x) &= 2xH_n(x) - 2nH_{n-1}(x), \quad n = 1, 2, \dots \\ k_{n,0} &= 2^n, \quad h_n = \sqrt{\pi}2^n n! \end{aligned}$$

I polinomi classici sono inoltre soluzione di equazioni differenziali lineari omogenee del secondo ordine.

Prima di terminare il paragrafo osserviamo che ai polinomi di Chebyshev di 1<sup>a</sup> e 2<sup>a</sup> specie nell'intervallo di ortogonalità  $(-1, 1)$  possiamo dare una forma particolarmente semplice; infatti operando il cambiamento di variabile  $x = \cos \theta$ ,  $0 \leq \theta \leq \pi$ , non è difficile verificare la validità delle espressioni

$$\begin{aligned} T_n(\cos \theta) &= \cos(n\theta) \\ U_n(\cos \theta) &= \frac{\sin(n+1)\theta}{\sin \theta} \end{aligned}$$

Anche le formule di ricorrenza per questi polinomi assumono una forma assai semplice:

$$\begin{cases} T_0(x) = 1 \\ T_1(x) = x \\ T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \quad n = 1, 2, \dots \end{cases}$$

$$\begin{cases} U_0(x) = 1 \\ U_1(x) = 2x \\ U_{n+1}(x) = 2xU_n(x) - U_{n-1}(x), \quad n = 1, 2, \dots \end{cases}$$

## 7.4 Formule di quadratura Gaussiane

Scelti  $n$  nodi distinti  $\{x_i\}$ , nel paragrafo 7.1 abbiamo dimostrato che è sempre possibile costruire una, ed una sola, formula di quadratura del tipo

$$(7.9) \quad \int_a^b w(x)f(x) dx = \sum_{i=1}^n w_i f(x_i) + R_n(f)$$

con grado di precisione almeno  $n - 1$ , cioè tale che  $R_n(f) = 0$  ognualvolta  $f(x)$  è un polinomio di grado  $\leq n - 1$ . Potendo però scegliere i nodi  $\{x_i\}$ , qual è la distribuzione più conveniente? Ovvero, esistono formule (7.9) di tipo interpolatorio con grado di precisione maggiore di  $n - 1$ , anzi, il più elevato possibile? Per dare una risposta a questa domanda supporremo che la funzione peso  $w(x)$  soddisfi le seguenti ipotesi:

- (i)  $w(x) \not\equiv 0$  e  $w(x) \geq 0$  in  $(a, b)$
- (ii) esistano tutti i momenti  $m_k = \int_a^b w(x)x^k dx$ ,  $k = 0, 1, \dots$

Osserviamo preliminarmente che  $2n - 1$  è il massimo grado di precisione raggiungibile da una formula con  $n$  nodi reali; ossia non esistono formule con  $n$  nodi reali e grado di precisione  $2n$ . Infatti, se una tale formula esistesse, il resto  $R_n(f)$  dovrebbe risultare nullo quando in (7.9) poniamo

$$\begin{aligned} f(x) &= \prod_{i=1}^n (x - x_i)^2 \in \mathbb{P}_{2n} \\ \int_a^b w(x) \prod_{i=1}^n (x - x_i)^2 dx &= \sum_{i=1}^n w_i \cdot 0 + R_n(f) = 0 + 0 = 0 \end{aligned}$$

ma ciò è impossibile perché l'integrale a primo membro è sempre positivo.

Il teorema che segue ci garantisce, per ogni  $n = 1, 2, \dots$ , l'esistenza di una ed una sola formula di tipo (7.9) con grado di precisione  $2n - 1$ . Tali formule vengono definite *Gaussiane* (o di Gauss).

**Teorema 7.4.** *Condizione necessaria e sufficiente affinché la formula*

$$(7.10) \quad \int_a^b w(x)f(x) dx = \sum_{i=1}^n w_i f(x_i) + R_n(f)$$

*sia Gaussiana, cioè abbia grado di precisione  $2n - 1$ , è che essa sia di tipo interpolatorio e che i nodi  $\{x_i\}$  coincidano con gli  $n$  zeri del polinomio  $P_n(x)$ , di grado  $n$ , ortogonale in  $(a, b)$  rispetto alla funzione peso  $w(x)$ .*

*Dimostrazione.* Supponiamo dapprima che la (7.10) sia Gaussiana, cioè abbia grado di precisione  $2n - 1$ . Essa è allora di tipo interpolatorio<sup>(†)</sup>, e inoltre, preso in (7.10)  $f(x) =$

---

<sup>(†)</sup> Ricordiamo che una formula quale la (7.10) con grado di precisione almeno  $n - 1$  è necessariamente di tipo interpolatorio. Vedere inoltre l'esercizio 7.2.

$P_n(x)x^k$ ,  $k = 0, 1, \dots, n-1$ , con  $P_n(x) = \prod_{i=1}^n (x - x_i)$ , otteniamo

$$\int_a^b w(x) P_n(x) x^k dx = 0, \quad k = 0, 1, \dots, n-1$$

ovvero, ricordando il teorema 7.3,  $P_n(x)$  è ortogonale in  $(a, b)$  rispetto alla funzione peso  $w(x)$ .

Se invece assumiamo che la (7.10) sia interpolatoria e che  $P_n(x) = \prod_{i=1}^n (x - x_i)$  risulti ortogonale in  $(a, b)$  rispetto a  $w(x)$ , preso un generico polinomio  $\pi_d(x)$ , di grado  $d$  ( $n \leq d \leq 2n-1$ ), dividiamo quest'ultimo per  $P_n(x)$ ; otteniamo la relazione

$$\pi_d(x) = P_n(x) q_{d-n}(x) + r_{n-1}(x)$$

dove  $q_{d-n}(x)$  è il polinomio quoziente, di grado  $d-n \leq n-1$ , e  $r_{n-1}(x)$  il polinomio resto, di grado al più  $n-1$ . Poiché l'errore  $R_n(f)$  della (7.10) è un funzionale lineare, ovvero

$$R_n(\alpha f + \beta g) = \alpha R_n(f) + \beta R_n(g), \quad \alpha, \beta \in \mathbb{R}$$

abbiamo

$$R_n(\pi_d) = R_n(P_n q_{d-n}) + R_n(r_{n-1})$$

Ricordando che  $R_n(r_{n-1}) = 0$  perché per ipotesi la (7.10) ha almeno grado di precisione  $n-1$ , e che  $R_n(P_n q_{d-n}) = 0$  perché  $P_n(x)$  è ortogonale e  $q_{d-n}(x)$  ha grado al più  $n-1$ , possiamo concludere affermando che  $R_n(\pi_d) = 0$ ; la formula ha pertanto grado di precisione  $2n-1$ .  $\square$

Volendo trovare una rappresentazione per i pesi  $w_i$ , nella (7.10) prendiamo

$$f(x) = \frac{P_n(x)}{x - x_k} = \prod_{\substack{i=1 \\ i \neq k}}^n (x - x_i)$$

otteniamo

$$\int_a^b w(x) \frac{P_n(x)}{x - x_k} dx = w_k P'_n(x_k)$$

e quindi

$$w_k = \frac{1}{P'_n(x_k)} \int_a^b w(x) \frac{P_n(x)}{x - x_k} dx, \quad k = 1, 2, \dots, n$$

Tuttavia, questa espressione non pone in evidenza proprietà particolari, anche perché non utilizza la maggior precisione delle formule Gaussiane. Scegliendo invece

$$f(x) = \left[ \frac{P_n(x)}{x - x_k} \right]^2 \in \mathbb{P}_{2n-2}$$

dalla (7.10) deduciamo

$$w_k = \frac{1}{[P'_n(x_k)]^2} \int_a^b w(x) \left[ \frac{P_n(x)}{x - x_k} \right]^2 dx, \quad k = 1, 2, \dots, n$$

Quest'ultima rappresentazione dimostra che i pesi  $\{w_i\}$  delle formule Gaussiane sono tutti positivi. Quando l'intervallo  $(a, b)$  è limitato, la positività di tutti i  $w_i$  ci garantisce la convergenza delle formule di quadratura per tutte le  $f(x) \in C[a, b]$ ; infatti, prendendo nella (7.10)  $f(x) \equiv 1$  otteniamo

$$\sum_{i=1}^n |w_i| = \sum_{i=1}^n w_i = \int_a^b w(x) dx < \infty$$

Osserviamo infine che la formula Gaussiana (7.10) può essere conseguita anche integrando la formula di interpolazione di Hermite associata agli zeri  $\{x_i\}$  del polinomio  $P_n(x)$  ortogonale in  $(a, b)$  rispetto alla funzione peso  $w(x)$ :

$$(7.11) \quad f(x) = \sum_{i=1}^n h_{0,i}(x)f(x_i) + \sum_{i=1}^n h_{1,i}(x)f'(x_i) + \frac{P_n^2(x)}{(2n)!}f^{(2n)}(\eta_x)$$

dove, ricordiamo,

$$h_{0,i}(x) = [1 - 2l'_i(x_i)(x - x_i)]l_i^2(x), \quad l_i(x) = \frac{P_n(x)}{(x - x_i)P'_n(x_i)}$$

$$h_{1,i}(x) = (x - x_i)l_i^2(x)$$

e la rappresentazione del resto è valida se  $f(x) \in C^{2n}[a, b]$ . Nella relazione

$$(7.12) \quad \begin{aligned} \int_a^b w(x)f(x) dx &= \sum_{i=1}^n \left[ \int_a^b w(x)h_{0,i}(x) dx \right] f(x_i) + \\ &+ \sum_{i=1}^n \left[ \int_a^b w(x)h_{1,i}(x) dx \right] f'(x_i) + \int_a^b w(x) \frac{P_n^2(x)}{(2n)!} f^{(2n)}(\eta_x) dx \end{aligned}$$

abbiamo

$$\int_a^b w(x)h_{1,i}(x) dx = \frac{1}{[P'_n(x_i)]^2} \int_a^b w(x)P_n(x) \frac{P_n(x)}{x - x_i} dx = 0$$

e

$$w_i = \int_a^b w(x)h_{0,i}(x) dx$$

Inoltre, applicando il teorema della media pesata per gli integrali al resto presente in (7.12)

$$R_n(f) = \int_a^b w(x) \frac{P_n^2(x)}{(2n)!} f^{(2n)}(\eta_x) dx = \frac{f^{(2n)}(\xi)}{(2n)!} \int_a^b w(x)P_n^2(x) dx$$

otteniamo anche una rappresentazione dell'errore in (7.10).

Le formule Gaussiane classiche sono quelle associate ai polinomi ortogonali classici, e precisamente:

(i) *formula di Gauss-Legendre*

$$\int_{-1}^1 f(x) dx \approx \sum_{i=1}^n w_i f(x_i)$$

dove  $k_{n,0} \prod_{i=1}^n (x - x_i) = P_n(x)$  (polinomio di Legendre);

(ii) *formula di Gauss-Jacobi*

$$\int_{-1}^1 (1-x)^\alpha (1+x)^\beta f(x) dx \approx \sum_{i=1}^n w_i f(x_i), \quad \alpha, \beta > -1$$

dove  $k_{n,0} \prod_{i=1}^n (x - x_i) = P_n^{(\alpha, \beta)}(x)$ ;

(iii) *formula di Gauss-Laguerre*

$$\int_0^\infty e^{-x} f(x) dx \approx \sum_{i=1}^n w_i f(x_i)$$

dove  $k_{n,0} \prod_{i=1}^n (x - x_i) = L_n(x)$ ;

(iv) *formula di Gauss-Hermite*

$$\int_{-\infty}^\infty e^{-x^2} f(x) dx \approx \sum_{i=1}^n w_i f(x_i)$$

dove  $k_{n,0} \prod_{i=1}^n (x - x_i) = H_n(x)$ .

Particolarmente interessanti risultano le formule di Gauss-Chebyshev, ossia quelle di Gauss-Jacobi con  $\alpha = \beta = -\frac{1}{2}$ :

$$\int_{-1}^1 \frac{1}{\sqrt{1-x^2}} f(x) dx \approx \frac{\pi}{n} \sum_{i=1}^n f\left(\cos \frac{2i-1}{2n}\pi\right)$$

In alcune applicazioni è necessario utilizzare formule Gaussiane chiuse o semichiuse, del tipo

$$(7.13) \quad \int_a^b w(x) f(x) dx \approx w_0 f(a) + \sum_{i=1}^n w_i f(x_i) + w_{n+1} f(b)$$

$$(7.14) \quad \int_a^b w(x) f(x) dx \approx w_0 f(a) + \sum_{i=1}^n w_i f(x_i)$$

$$(7.15) \quad \int_a^b w(x) f(x) dx \approx \sum_{i=1}^n w_i f(x_i) + w_{n+1} f(b)$$

con grado di precisione  $2n+1$  la prima, e  $2n$  le restanti due. La (7.13) è nota con il nome di formula del tipo di *Lobatto*, mentre le (7.14) e (7.15) sono dette di *Radau*. Anche queste formule sono ovviamente interpolatorie, e i nodi  $\{x_i\}$  coincidono con gli zeri del polinomio  $P_n(x)$  ortogonale in  $(a, b)$  rispetto alla funzione peso  $w(x)(x - a)(b - x)$  la prima,  $w(x)(x - a)$  la seconda, e  $w(x)(b - x)$  la terza.

Nella tabella 7.1 riportiamo i risultati ottenuti applicando la formula di Gauss-Legendre (a  $n$  nodi) ai due integrali

$$I_1 = \int_0^1 e^{-x^2} dx = 0.7468241328\dots$$

$$I_2 = \int_0^1 \ln(x) \cos^2(x) dx = -0.9013532442\dots$$

Osserviamo che la funzione integranda è molto regolare in  $I_1$ , ed ha una singolarità nell'estremo  $x = 0$  in  $I_2$ .

$n$	$I_1$	$n$	$I_2$
2	0.7465947	2	-0.8019467
4	0.7468245	4	-0.8699360
8	0.7468241	8	-0.8925932
		16	-0.8990325
		32	-0.9007553
		64	-0.9012015
		128	-0.9013150
		256	-0.9013436

Tabella 7.1

## 7.5 Costruzione delle formule Gaussiane

Nel paragrafo precedente abbiamo caratterizzato le formule Gaussiane; tuttavia, per la loro effettiva costruzione non vengono utilizzate le espressioni là presentate. Anzi, il metodo più efficiente per la loro determinazione non si avvale affatto della rappresentazione esplicita del polinomio ortogonale  $P_n(x)$  (i cui zeri forniscono i nodi delle formule).

Riteniamo interessante illustrare, seppure brevemente, la riformulazione del problema che permette di pervenire al nuovo procedimento di costruzione. A tale scopo è indispensabile normalizzare il sistema di polinomi ortogonali  $\{P_m(x)\}$  in questione, in modo da avere

$$h_m = \int_a^b w(x) P_m^2(x) dx = 1$$

Per il sistema normalizzato  $\{P_m^*(x) = h_m^{-1/2} P_m(x) = k_{m,0}^* x^m + k_{m,1}^* x^{m-1} + \dots + k_{m,m}^*\}$  la relazione di ricorrenza (7.7) assume la nuova espressione

$$(7.16) \quad P_{m+1}^*(x) = (A_m^* x + B_m^*) P_m^*(x) - C_m^* P_{m-1}(x), \quad m = 1, 2, \dots$$

con

$$A_m^* = \frac{k_{m+1,0}^*}{k_{m,0}^*}, \quad B_m^* = A_m^* \left( \frac{k_{m+1,1}^*}{k_{m+1,0}^*} - \frac{k_{m,1}^*}{k_{m,0}^*} \right), \quad C_m^* = \frac{A_m^*}{A_{m-1}^*}$$

Riscrivendo la (7.16) come segue

$$x P_m^*(x) = \frac{1}{A_m^*} P_{m+1}^*(x) - \frac{B_m^*}{A_m^*} P_m^*(x) + \frac{C_m^*}{A_m^*} P_{m-1}^*(x), \quad m = 1, 2, \dots$$

e ricordando l'espressione di  $C_m^*$ , possiamo dare a quest'ultima una forma nuova e ben più significativa:

$$(7.17) \quad x P_m^*(x) = \alpha_{m-1} P_{m-1}^*(x) + \beta_m P_m^*(x) + \alpha_m P_{m+1}^*(x), \quad m = 0, 1, 2, \dots$$

con coefficienti

$$\alpha_{-1} = 0, \quad \alpha_m = \frac{1}{A_m^*} \quad \text{e} \quad \beta_m = -\frac{B_m^*}{A_m^*}$$

L'interesse per la (7.17) nasce dalla constatazione che essa, con  $m = 0, 1, \dots, n-1$ , può essere interpretata nel modo seguente:

$$x \begin{pmatrix} P_0^*(x) \\ P_1^*(x) \\ P_2^*(x) \\ P_3^*(x) \\ \vdots \\ P_{n-2}^*(x) \\ P_{n-1}^*(x) \end{pmatrix} = \begin{pmatrix} \beta_0 & \alpha_0 & 0 & 0 & \dots & 0 & 0 \\ \alpha_0 & \beta_1 & \alpha_1 & 0 & \dots & 0 & 0 \\ 0 & \alpha_1 & \beta_2 & \alpha_2 & \dots & 0 & 0 \\ 0 & 0 & \alpha_2 & \beta_3 & \dots & 0 & 0 \\ \ddots & & & & & & \\ 0 & 0 & 0 & 0 & \dots & \beta_{n-2} & \alpha_{n-2} \\ 0 & 0 & 0 & 0 & \dots & \alpha_{n-2} & \beta_{n-1} \end{pmatrix} \begin{pmatrix} P_0^*(x) \\ P_1^*(x) \\ P_2^*(x) \\ P_3^*(x) \\ \vdots \\ P_{n-2}^*(x) \\ P_{n-1}^*(x) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ \alpha_{n-1} P_n^*(x) \end{pmatrix}$$

ossia

$$(7.18) \quad x p(x) = T p(x) + \alpha_{n-1} P_n^*(x) e_n$$

dove

$$p(x) = (P_0^*(x), P_1^*(x), \dots, P_{n-1}^*(x))^T \quad \text{e} \quad e_n = (0, 0, \dots, 0, 1)^T$$

Infatti, dalla (7.18) deduciamo che  $P_n^*(x_i) = 0$  se e solo se

$$T p(x_i) = x_i p(x_i), \quad i = 1, 2, \dots, n$$

ovvero, gli zeri  $x_i$  del polinomio  $P_n^*(x)$  coincidono con gli autovalori della matrice tri-diagonale simmetrica  $T$ , e i  $p(x_i)$  sono i corrispondenti autovettori. La rappresentazione

(reperibile in [7.13])

$$(7.19) \quad w_i = \frac{1}{\sum_{j=0}^{n-1} [P_j^*(x_i)]^2}, \quad i = 1, 2, \dots, n$$

ci consente di esprimere i pesi della formula Gaussiana mediante i suddetti autovettori.

L'espressione (7.19) può essere semplificata osservando che dall'identità

$$1 = w_i \sum_{j=0}^{n-1} [P_j^*(x_i)]^2 = w_i p(x_i)^T p(x_i) = \|\sqrt{w_i} p(x_i)\|_2^2$$

segue che il vettore

$$p^*(x_i) = \sqrt{w_i} p(x_i)$$

rappresenta l'autovettore normalizzato di  $T$  corrispondente all'autovalore  $x_i$ . In particolare, considerando la prima componente di  $p^*(x_i) = (p_{0,i}^*, p_{1,i}^*, \dots, p_{n-1,i}^*)^T$  e osservando che

$$P_0^*(x_i) = \left[ \int_a^b w(x) dx \right]^{-\frac{1}{2}}$$

otteniamo

$$(7.20) \quad w_i = (p_{0,i}^*)^2 \int_a^b w(x) dx, \quad i = 1, 2, \dots, n$$

Pertanto per determinare i pesi  $w_i$  è sufficiente calcolare la prima componente degli autovettori normalizzati della matrice  $T$ .

La costruzione delle formule Gaussiane è quindi ricondotta ad un problema (ben condizionato) autovalori-autovettori di una matrice tridiagonale simmetrica, che possiamo risolvere efficientemente con il metodo  $QR$ . Ovviamente questo metodo presuppone la conoscenza o il calcolo dei coefficienti della formula di ricorrenza (7.7) oppure (7.16).

## 7.6 Stima dell'errore $R_n(f)$

Supponiamo di aver scelto una formula di quadratura

$$Q_n(f) = \sum_{i=1}^n w_i f(x_i)$$

per approssimare l'integrale

$$I(f) = \int_{-1}^1 w(x) f(x) dx$$

Affinché la stima  $Q_n(f)$  sia credibile occorre poter associare ad essa una indicazione della sua precisione.

Per le formule di tipo interpolatorio abbiamo dato una rappresentazione integrale dell'errore

$$R_n(f) = I(f) - Q_n(f)$$

altre sono reperibili nella letteratura specializzata. Tuttavia queste espressioni hanno un'importanza soprattutto teorica. Di solito la stima cercata di  $R_n(f)$  viene prodotta nel modo seguente: si prende una seconda formula  $Q_m(f)$ , dello stesso tipo di  $Q_n(f)$  ma più “precisa” ovvero con un numero maggiore di nodi ( $m > n$ ), e si pone

$$(7.21) \quad |R_n(f)| \approx |Q_n(f) - Q_m(f)|$$

Generalmente si sceglie  $m = n + 1$  quando la funzione  $f(x)$  è ritenuta sufficientemente “regolare”, e  $m \approx 2n$  quando non lo è. La coppia  $(Q_n(f), Q_m(f))$  in generale “costa”  $n + m$  valutazioni di funzione  $\{f(x_i)\}$ ; a meno che le due formule  $Q_n(f)$  e  $Q_m(f)$  non abbiano nodi in comune, nel qual caso il costo complessivo risulta inferiore. In particolare, nel caso delle formule Gaussiane

$$G_n(f) = \sum_{i=1}^n w_i f(x_i)$$

assumendo

$$|R_n(f)| \approx |G_n(f) - G_{n+1}(f)|$$

imponiamo il calcolo di  $2n + 1$  valori  $f(x_i)$ ; infatti,  $G_n(f)$  e  $G_{n+1}(f)$  non hanno nodi in comune. Inoltre, il grado di precisione della formula più precisa ( $G_{n+1}(f)$ ) è  $2n + 1$ . Ma a parità di “costo” è possibile ottenere molto di più. Come seconda, e più precisa, formula prendiamo

$$(7.22) \quad K_{2n+1}(f) = \sum_{i=1}^n w_i^{(1)} f(x_i) + \sum_{j=1}^{n+1} w_j^{(2)} f(y_j)$$

dove i nodi  $\{x_i\}$  sono gli stessi di  $G_n(f)$ , mentre i punti  $\{y_j\}$  e i pesi  $\{w_i^{(1)}\}$  e  $\{w_j^{(2)}\}$  vengono scelti in modo da raggiungere il massimo grado di precisione ( $3n + 1$ , salvo casi eccezionali nei quali risulta addirittura superiore).

Quando in  $I(f)$  scegliamo  $w(x) \equiv 1$ , le corrispondenti formule  $K_{2n+1}(f)$ , attribuite a A. S. Kronrod, esistono per ogni intero  $n$  e sono simmetriche; inoltre

$$-1 < y_1 < x_1 < y_2 < x_2 < \dots < x_{n-1} < y_n < x_n < y_{n+1} < 1$$

e i pesi  $w_i^{(1)}$  e  $w_j^{(2)}$  sono tutti positivi<sup>(†)</sup>. In questo modo, senza aumentare il costo complessivo, abbiamo una seconda formula ( $K_{2n+1}(f)$ ) molto più precisa di  $G_{n+1}(f)$ . Inoltre, il numero di nodi di quest'ultima è doppio rispetto a quello di  $G_n(f)$ , per cui

---

(†) Questi risultati valgono anche nel caso della funzione peso di Jacobi con  $\alpha = \beta$  e  $-\frac{1}{2} \leq \alpha \leq \frac{3}{2}$ .

$n = 7$		
$y_j/x_i$	$w_i$	$w_j^{(2)}/w_i^{(1)}$
$\pm 0.9914553711208126$ $\pm 0.9491079123427586$	0.1294849661688697	$0.2293532201052920 \cdot 10^{-1}$ $0.6309209262997860 \cdot 10^{-1}$
$\pm 0.8648644233597690$ $\pm 0.7415311855993944$	0.2797053914892767	$0.1047900103222502$ $0.1406532597155260$
$\pm 0.5860872354676912$ $\pm 0.4058451513773972$	0.3818300505051189	$0.1690047266392680$ $0.1903505780647854$
$\pm 0.2077849550078984$ 0.0	0.4179591836734694	$0.2044329400752988$ $0.2094821410847278$

$n = 10$		
$y_j/x_i$	$w_i$	$w_j^{(2)}/w_i^{(1)}$
$\pm 0.9956571630258081$ $\pm 0.9739065285171717$	$0.6667134430868814 \cdot 10^{-1}$	$0.1169463886737187 \cdot 10^{-1}$ $0.3255816230796473 \cdot 10^{-1}$
$\pm 0.9301574913557082$ $\pm 0.8650633666889845$	0.1494513491505806	$0.5475589657435200 \cdot 10^{-1}$ $0.7503967481091995 \cdot 10^{-1}$
$\pm 0.7808177265864169$ $\pm 0.6794095682990244$	0.2190863625159820	$0.9312545458369761 \cdot 10^{-1}$ $0.1093871588022976$
$\pm 0.5627571346686047$ $\pm 0.4333953941292472$	0.2692667193099964	$0.1234919762620659$ $0.1347092173114733$
$\pm 0.2943928627014602$ $\pm 0.1488743389816312$	0.2955242247147529	$0.1427759385770601$ $0.1477391049013385$
0.0		0.1494455540029169

Tabella 7.2

la stima  $|G_n(f) - K_{2n+1}(f)|$  risponde ad entrambe le esigenze sollevate in (7.21): costo complessivo non superiore a  $2n + 1$  quando  $f(x)$  è sufficientemente regolare e  $m \approx 2n$  quando non lo è. Nella tabella 7.2 riportiamo nodi e pesi della coppia di formule  $(G_n(f), K_{2n+1}(f))$ ,  $n = 7, 10$ , per l'approssimazione dell'integrale  $\int_{-1}^1 f(x) dx$ :

$$\left\{ \begin{array}{l} G_n(f) = \sum_{i=1}^n w_i f(x_i) \\ K_{2n+1}(f) = \sum_{i=1}^n w_i^{(1)} f(x_i) + \sum_{j=1}^{n+1} w_j^{(2)} f(y_j) \end{array} \right.$$

## 7.7 Formule composte

Nella costruzione di formule di quadratura di tipo interpolatorio la funzione integranda viene approssimata sull'intero intervallo di integrazione con un unico polinomio (quello interpolante la funzione integranda nei nodi scelti). Quando la formula scelta risulta convergente (per  $n \rightarrow \infty$ ), la precisione desiderata può essere conseguita prendendo un numero di nodi  $\{x_i\}$  sufficientemente elevato. Per ogni intero  $n$  scelto occorre però costruire la corrispondente formula.

Una strategia alternativa che consente di evitare di dover costruire una nuova formula interpolatoria per ogni  $n$ , e che talvolta<sup>(†)</sup> produce risultati apprezzabili, è quella delle *formule composte*. Scelta una formula base  $Q_n(f) \equiv Q_n(a, b; f)$ , l'intervallo di integrazione  $(a, b)$ , che qui supponiamo limitato, viene suddiviso in  $m$  parti, per semplicità uguali di ampiezza  $h = (b - a)/m$ :

$$\int_a^b f(x) dx = \sum_{i=0}^{m-1} \int_{a_i}^{a_{i+1}} f(x) dx, \quad a_i = a + ih$$

e la quadratura  $Q_n(f)$  viene applicata ad ogni sottointervallo  $(a_i, a_{i+1})$ :

$$(7.23) \quad \int_a^b f(x) dx \approx \sum_{i=0}^{m-1} Q_n(a_i, a_{i+1}; f)$$

Il grado di precisione della formula composta (7.23) coincide con quello della formula base scelta.

Le formule composte più usate sono quelle dei trapezi e di Simpson:

(i) formula composta dei trapezi

(7.24)

$$\int_a^b f(x) dx = \frac{h}{2} \left[ f(a) + 2 \sum_{i=1}^{m-1} f(a + ih) + f(b) \right] + R_m(f)$$

$$h = \frac{b - a}{m}, \quad R_m(f) = -\frac{b - a}{12} h^2 f^{(2)}(\xi_m), \quad a < \xi_m < b, \quad f \in C^2[a, b]$$

(ii) formula composta di Simpson

(7.25)

$$\int_a^b f(x) dx = \frac{h}{3} \left[ f(a) + 2 \sum_{i=1}^{m-1} f(a + 2ih) + 4 \sum_{i=1}^m f(a + (2i-1)h) + f(b) \right] + R_m(f)$$

$$h = \frac{b - a}{2m}, \quad R_m(f) = -\frac{b - a}{180} h^4 f^{(4)}(\eta_m), \quad a < \eta_m < b, \quad f \in C^4[a, b]$$

---

(†) Soprattutto in connessione con *processi di estrappolazione* (vedi [7.12]).

$I_1$			$I_2$		
$n$	trapezi	Gauss-Legendre	$n$	trapezi	Gauss-Legendre
2	6.283186	0.4041909	2	0.8623652	1.395492
4	3.559457	3.659233	4	1.383995	1.429683
8	3.977393	4.322892	8	1.421485	1.429777
16	3.977463	3.975158	16	1.427976	
32		3.977463	32	1.429356	
			64	1.429675	
			128	1.429752	
			256	1.429771	
			512	1.429775	
			1024	1.429777	

Tabella 7.3

La formula composta dei trapezi, nonostante il suo basso grado di precisione polinomiale, fornisce risultati molto accurati quando la funzione integranda  $f(x)$  è periodica e l'ampiezza  $b - a$  dell'intervallo di integrazione coincide con un periodo o con un suo multiplo intero. Infatti, *in questo caso, se  $f(x) \in C^{2k+1}[a, b]$  e  $f^{(2j-1)}(a) = f^{(2j-1)}(b)$ ,  $j = 1, 2, \dots, k$ , il comportamento dell'errore  $R_m(f)$  è  $O(h^{2k+1})$  anziché  $O(h^2)$ .*

Per verificare su un esempio concreto la suddetta proprietà, abbiamo applicato la formula dei trapezi composta (su  $n - 1$  intervalli ovvero con  $n$  nodi complessivi) ai due integrali

$$I_1 = \int_0^{2\pi} \cos^2(x)e^{\sin(2x)} dx = 3.977463260\dots$$

$$I_2 = \int_0^1 \cos^2(x)e^{\sin(2x)} dx = 1.429777221\dots$$

I risultati ottenuti vengono confrontati con quelli forniti dalla corrispondente formula di Gauss-Legendre (tabella 7.3). Il diverso comportamento della formula dei trapezi nei due casi è dovuto al fatto che mentre in  $I_1$  l'intervallo di integrazione coincide con un multiplo intero (2) del periodo della funzione integranda, in  $I_2$  ciò non si verifica.

Le formule composte con suddivisione uniforme dell'intervallo di integrazione sono ormai superate, tranne in casi particolarissimi quale ad esempio quello delle funzioni periodiche, da quelle associate a suddivisioni di tipo adattativo che presenteremo nel prossimo paragrafo.

## 7.8 Routine automatiche

Supponiamo di dover approssimare, con tolleranza relativa `toll`, l'integrale

$$I(f) = \int_a^b f(x) \, dx, \quad (a, b) \text{ limitato}$$

Quando la funzione integranda è ritenuta sufficientemente regolare, per esempio quando  $f(x) \in C^k[a, b]$ ,  $k \gg 1$  e  $f(z)$  non presenta singolarità in  $\mathbb{C}$  “troppo” vicine all'intervallo  $(a, b)$ <sup>(†)</sup> conviene utilizzare formule con grado di precisione polinomiale massimo. Infatti nell'ipotesi suddetta tali formule generalmente permettono di raggiungere la precisione richiesta con un numero di nodi  $x_i$  inferiore a quello necessario con formule di altro tipo. In questa situazione T. N. L. Patterson estendendo l'idea di Kronrod presentata nel paragrafo 7.6 ha prodotto una successione “ottimale” di formule encapsulate  $\{Q_3(f) \equiv G_3(f), Q_7(f), Q_{15}(f), \dots, Q_{255}(f), Q_{511}(f)\}$ , nel senso che

$$\begin{aligned} Q_{2n+1}(f) &= \sum_{i=1}^n w_i^{(1)} f(x_i) + \sum_{j=1}^{n+1} w_j^{(2)} f(y_j) \\ Q_n(f) &= \sum_{i=1}^n w_i f(x_i) \end{aligned}$$

dove

$$-1 < y_1 < x_1 < y_2 < x_2 < \dots < x_{n-1} < y_n < x_n < y_{n+1} < 1$$

I coefficienti  $\{w_i^{(1)}\}$  e  $\{w_j^{(2)}\}$  sono tutti positivi, e il grado di precisione del generico elemento  $Q_m(f)$ ,  $m = 7, 15, \dots, 255, 511$ , è  $3[m/2] + 2$ . Le formule  $Q_m(f)$  sono tutte simmetriche, cioè hanno nodi e pesi simmetrici. La famiglia di formule  $\{Q_m(f)\}$  di Patterson è stata utilizzata per costruire routine automatiche del tipo seguente

- 1: determina  $Q_3(f)$
- 2: **ciclo 1:**  $m = 2, \dots, m_{\max}$
- 3:  $n \leftarrow 2^m - 1$
- 4: determina  $Q_{2n+1}(f)$
- 5:  $\text{err} \leftarrow |Q_n(f) - Q_{2n+1}(f)|$
- 6: se  $\text{err} \leq \text{toll}|Q_{2n+1}(f)|$  allora  $\text{ier} \leftarrow 0$ ,  $\text{est} \leftarrow Q_{2n+1}(f)$ ; **esci**
- 7: **fine ciclo 1**
- 8:  $\text{ier} \leftarrow 1$
- 9:  $\text{est} \leftarrow Q_{2n+1}(f)$
- 10: **esci**

---

(†) Le funzioni  $f(x) = e^x$ ,  $\ln(x+2)$ ,  $\cos(x)$  sono “regolari” in  $[-1, 1]$  mentre  $f(x) = |x|$ ,  $x^{-1/2}$ ,  $x^{3/2}$  non lo sono. La funzione  $f(x) = 1/(x^2 + 10^{-4})$  ha due poli complessi ( $\pm i10^{-2}$ ) troppo vicini a  $[-1, 1]$ .

dove **est** denota la stima dell'integrale  $I(f)$  fornita dalla routine.

Quando invece la funzione integranda  $f(x)$  presenta delle irregolarità conviene procedere con una strategia di tipo adattativo, che suddivide l'intervallo di integrazione  $(a, b)$  in sottointervalli di ampiezza diversa e applichi a questi ultimi una formula base (con non molti nodi), Gaussiana per esempio. La strategia scelta deve addensare i punti là dove la  $f(x)$  presenta irregolarità, e collocare pochi nodi là dove la funzione è regolare.

Con la suddivisione uniforme delle formule composte del paragrafo 7.7, le irregolarità della  $f(x)$  richiedendo un addensamento di nodi nelle loro vicinanze impongono contemporaneamente lo stesso addensamento in tutto l'intervallo  $(a, b)$ ; anche in quelle zone dove è del tutto superfluo.

Per poter attuare una strategia di suddivisione adattativa è indispensabile poter disporre di una buona stima dell'errore associato alla formula base scelta. Le coppie di formule  $(G_n, K_{2n+1})$  introdotte da Kronrod si rivelano estremamente utili proprio a questo scopo.

Lo schema seguente descrive una possibile strategia di integrazione adattativa:

- 1: **sia**  $I = [a, b]$ ; determina  $G_n(I; f)$  e  $K_{2n+1}(I; f)$  e quindi **poni**  $\text{est}(I) \leftarrow K_{2n+1}(I; f)$  e  $\text{err}(I) = |G_n(I; f) - K_{2n+1}(I; f)|$ ;
- 2: **se**  $\text{err}(I) \leq \text{tol}$   $|K_{2n+1}(I; f)|$  **assumi**  $\text{est}(I)$  come stima accettabile; **esci**
- 3: **altrimenti** suddividi  $I$  in due parti uguali  $I_1$  e  $I_2$ , e determina  $(G_n(I_1; f), K_{2n+1}(I_1; f))$  e  $(G_n(I_2; f), K_{2n+1}(I_2; f))$ ; **poni**  
 $\text{est}(I_1) \leftarrow K_{2n+1}(I_1; f)$   
 $\text{err}(I_1) \leftarrow |G_n(I_1; f) - K_{2n+1}(I_1; f)|$   
 $\text{est}(I_2) \leftarrow K_{2n+1}(I_2; f)$   
 $\text{err}(I_2) \leftarrow |G_n(I_2; f) - K_{2n+1}(I_2; f)|$ .
- 4: dopo il passo  $m$ -esimo l'intervallo  $[a, b]$  è suddiviso in  $m$  sottointervalli  $I_i$ ,  $i = 1, 2, \dots, m$ , per ognuno dei quali sono state determinate le stime  $\text{est}(I_i)$  e  $\text{err}(I_i)$ .  
**se**  $\sum_{i=1}^m \text{err}(I_i) \leq \text{tol} |\sum_{i=1}^m \text{est}(I_i)|$  **allora** come stima dell'integrale prendi  $\sum_{i=1}^m \text{est}(I_i)$ ; **esci**
- 5: **se** l'ampiezza di uno dei sottointervalli  $I_i$  è inferiore al minimo stabilito, **esci** segnalando l'inconveniente; **altrimenti** considera l'intervallo  $I_k$  dove  $\text{err}(I_k) = \max_{1 \leq i \leq m} \text{err}(I_i)$ , suddividi  $I_k$  in due parti uguali (eliminando quindi  $I_k$ ,  $\text{est}(I_k)$  e  $\text{err}(I_k)$ ), applica ad entrambe la coppia  $(G_n, K_{2n+1})$  e **ritorna** al punto 4.

Una eccellente raccolta di routine Fortran per la valutazione di integrali monodimensionali è stata pubblicata in [7.12].

## 7.9 Integrazione su intervalli infiniti

Quando l'intervallo  $(a, b)$  è illimitato, per esempio  $(-\infty, \infty)$  oppure  $(0, \infty)$ , per determinare un'approssimazione dell'integrale

$$(7.26) \quad \int_a^b f(x) \, dx$$

possiamo procedere, oltre che con formule Gaussiane quali quelle di Hermite e di Laguerre, ricorrendo alle formule per intervalli finiti: si sostituisce l'intervallo infinito con uno limitato sufficientemente grande in modo che il contributo relativo alla parte troncata risulti trascurabile. A volte, esaminando il comportamento della funzione integranda<sup>(†)</sup> è possibile stabilire a priori da quale punto in poi il contributo di quest'ultima è trascurabile. Per facilitare l'individuazione di tali punti di troncamento talvolta può rivelarsi utile un cambiamento (non lineare) di variabile, che, pur lasciando infinito l'intervallo di integrazione, aumenti la rapidità di decadimento a zero della funzione integranda (per  $x \rightarrow \infty$ ).

Quando la  $f(x)$  in (7.26) va a zero, per  $x \rightarrow \infty$ , molto rapidamente, si possono ottenere buoni risultati semplicemente applicando la formula dei trapezi composta

$$\begin{aligned} \int_a^\infty f(x) \, dx &\approx h \left[ \frac{1}{2}f(a) + \sum_{i=1}^{\infty} f(a + ih) \right] \approx h \left[ \frac{1}{2}f(a) + \sum_{i=1}^n f(a + ih) \right] \\ \int_{-\infty}^\infty f(x) \, dx &\approx h \sum_{i=-\infty}^{\infty} f(ih) \approx h \sum_{i=-N}^N f(ih) \end{aligned}$$

con il passo  $h$  sufficientemente piccolo e l'intero  $N$  sufficientemente grande.

Il problema dell'integrazione su intervalli infiniti può anche essere affrontato effettuando dapprima un cambiamento (non lineare) di variabile che trasformi l'intervallo infinito in uno finito, e utilizzando poi i metodi di integrazione per intervalli finiti. Tuttavia occorre rilevare che non sempre la forma dell'integrale trasformato risulta più conveniente; per esempio la funzione integranda, nella nuova variabile, potrebbe risultare eccessivamente oscillante nell'intervallo di integrazione oppure contenere delle singolarità difficili da trattare e quindi richiedere l'uso di formule di quadratura con un numero elevato di nodi.

Le idee che abbiamo esposto in questo paragrafo sono illustrate, con maggiori dettagli ed esempi, nel testo [7.13].

## 7.10 Alcune applicazioni delle formule di quadratura

1. Le formule di quadratura sviluppate per l'approssimazione di integrali su intervalli dell'asse reale possono essere utilizzate anche per il calcolo di integrali su curve  $\Gamma$  del

(†) Ricordiamo che condizione necessaria perché l'integrale, su  $(0, \infty)$  per esempio, esista è che la funzione integranda  $f(x)$  vada a zero quando  $x \rightarrow \infty$ .

piano  $(x, y)$

$$(7.27) \quad \int_{\Gamma} f(x, y) dx, \quad \int_{\Gamma} f(x, y) dy, \quad \int_{\Gamma} f(x, y) ds$$

La variabile  $s$  presente nell'ultimo integrale denota l'ascissa curvilinea su  $\Gamma$ . Infatti, se la curva  $\Gamma$  è data in forma parametrica

$$\Gamma : \begin{cases} x = x(t) \\ y = y(t) \end{cases} \quad t_0 \leq t \leq t_1$$

con  $x(t), y(t) \in C^1[t_0, t_1]$ , gli integrali (7.27) possono essere facilmente ricondotti ad integrali sull'intervallo  $(t_0, t_1)$ :

$$\begin{aligned} \int_{\Gamma} f(x, y) dx &= \int_{t_0}^{t_1} f(x(t), y(t)) x'(t) dt \\ \int_{\Gamma} f(x, y) dy &= \int_{t_0}^{t_1} f(x(t), y(t)) y'(t) dt \\ \int_{\Gamma} f(x, y) ds &= \int_{t_0}^{t_1} f(x(t), y(t)) \sqrt{[x'(t)]^2 + [y'(t)]^2} dt \end{aligned}$$

Esaminiamo il caso in cui  $\Gamma$  è, per esempio, l'ellisse

$$\Gamma : \begin{cases} x = a \cos \theta \\ y = b \sin \theta \end{cases} \quad 0 \leq \theta \leq 2\pi$$

abbiamo

$$\int_{\Gamma} f(x, y) ds = \int_0^{2\pi} f(a \cos \theta, b \sin \theta) (a^2 \sin^2 \theta + b^2 \cos^2 \theta)^{\frac{1}{2}} d\theta$$

La funzione integranda a secondo membro è periodica in  $\theta$ , con periodo  $2\pi$ ; se la funzione  $f(x, y)$  è sufficientemente regolare nelle due variabili  $x$  e  $y$  possiamo approssimare l'integrale in  $\theta$  con la formula composta dei trapezi.

**2.** Supponiamo di dover risolvere numericamente un'*equazione integrale di Fredholm di seconda specie*

$$(7.28) \quad u(x) = h(x) + \int_a^b K(x, y) u(y) dy, \quad a \leq x \leq b$$

nell'incognita  $u(x)$ . Scelta una formula di quadratura “idonea”

$$(7.29) \quad \int_a^b f(t) dt \approx \sum_{i=1}^n w_i f(t_i)$$

possiamo discretizzare l'integrale in (7.28) e quindi collocare l'equazione risultante

$$(7.30) \quad u_n(x) = h(x) + \sum_{i=1}^n w_i K(x, t_i) u_n(t_i), \quad a \leq x \leq b$$

dove  $u_n(x)$  denota l'approssimazione di  $u(x)$  "provocata" dalla (7.29), nei nodi  $x = t_j$ ,  $j = 1, 2, \dots, n$ ; otteniamo il seguente sistema lineare

$$\sum_{i=1}^n [\delta_{ji} - w_i K(t_j, t_i)] u_i = h(t_j), \quad j = 1, 2, \dots, n$$

nell'incognita  $(u_1, u_2, \dots, u_n)^T$ , con  $u_i = u_n(t_i) \approx u(t_i)$ . Determinati i valori  $\{u_n(t_i)\}$  possiamo poi utilizzare l'espressione (7.30) per valutare  $u_n(x)$  in punti  $x \neq t_i$ ,  $i = 1, \dots, n$ .

**3.** Spesso ci troviamo di fronte al problema della valutazione numerica di integrali multipli

$$\iint_D f(x, y, \dots) dx dy \dots, \quad D \subseteq \mathbb{R}^k$$

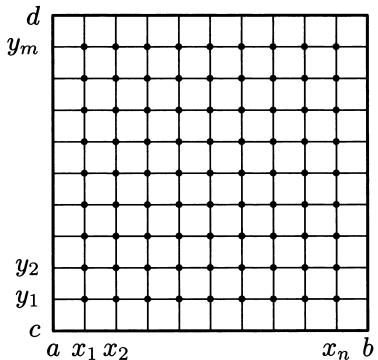
I risultati noti per gli integrali monodimensionali contribuiscono solo parzialmente alla risoluzione del problema. Mentre in  $\mathbb{R}$  le regioni di integrazione sono degli intervalli, finiti oppure infiniti, in  $\mathbb{R}^k$  esse possono essere molteplici: iper-rettangoli, simplessi, sfere, superfici sferiche, ecc. Per una visione panoramica della situazione consigliamo le letture [7.5] e [7.11].

Le formule costruite per gli integrali in una dimensione possono essere utilizzate per la costruzione di *formule prodotto* per integrali multipli su iper-rettangoli, per esempio. Esaminiamo il caso bidimensionale

$$I(f) = \int_a^b \int_c^d w_1(x) w_2(y) f(x, y) dx dy$$

Imitando la costruzione delle formule monodimensionali di tipo interpolatorio, scelti  $n$  punti  $\{x_i\}$ ,  $a \leq x_1 < x_2 < \dots < x_n \leq b$ , nell'intervallo  $[a, b]$ , e  $m$  punti  $\{y_j\}$ ,  $c \leq y_1 < y_2 < \dots < y_m \leq d$ , in  $[c, d]$ , consideriamo il polinomio  $P_{n,m}(x, y)$ , di grado  $(n-1)$  in  $x$  ed  $(m-1)$  in  $y$ , che interpola la funzione integranda  $f(x, y)$  negli  $n \times m$  nodi  $(x_i, y_j)$ ,  $i = 1, 2, \dots, n$ ,  $j = 1, 2, \dots, m$ :

$$P_{n,m}(x, y) = \sum_{i=1}^n \sum_{j=1}^m l_i(x) \bar{l}_j(y) f(x_i, y_j)$$



Integrando tale polinomio otteniamo

$$(7.31) \quad I(f) \approx \sum_{i=1}^n \sum_{j=1}^m w_{1,i} w_{2,j} f(x_i, y_j)$$

con

$$w_{1,i} = \int_a^b w_1(x) l_i(x) dx \quad \text{e} \quad w_{2,j} = \int_c^d w_2(y) \bar{l}_j(y) dy$$

Osserviamo che la formula (7.31) può anche essere facilmente dedotta dal “prodotto” dei nodi e pesi delle due formule *monodimensionali* di tipo interpolatorio

$$(7.32) \quad \begin{aligned} \int_a^b w_1(x) g(x) dx &\approx \sum_{i=1}^n w_{1,i} g(x_i) \\ \int_c^d w_2(y) g(y) dy &\approx \sum_{j=1}^m w_{2,j} g(y_j) \end{aligned}$$

Infatti, applicando la prima formula all’integrale  $I(f)$  posto nella forma

$$I(f) = \int_a^b w_1(x) F(x) dx$$

con

$$F(x) = \int_c^d w_2(y) f(x, y) dy$$

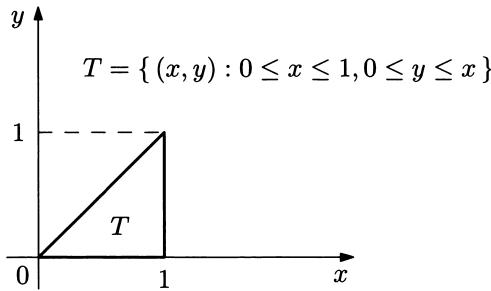
abbiamo

$$I(f) \approx \sum_{i=1}^n w_{1,i} F(x_i)$$

successivamente, approssimando ogni singolo  $F(x_i)$  mediante la seconda formula di quadratura otteniamo la (7.31). Pertanto, se per esempio come formule (7.32) prendiamo le corrispondenti di Gauss-Legendre a  $n$  e  $m$  nodi, la formula prodotto (7.31) avrà grado di precisione  $2n - 1$  nella variabile  $x$  (supponendo  $y$  costante) e  $2m - 1$  in  $y$ .

La limitazione più grave delle formule prodotto è la rapida crescita del numero complessivo di punti, su cui valutare la funzione integranda, all’aumentare della dimensione  $k$  del dominio di integrazione. Per esempio, nel caso di un iper-rettangolo in  $\mathbb{R}^k$  la scelta di  $n$  nodi per ogni variabile di integrazione conduce ad una formula con  $n^k$  punti.

Osserviamo infine che molte regioni, del piano  $(x, y)$  per esempio, possono essere trasformate, con semplici cambiamenti di variabile, in rettangoli. Per esempio, nel caso del triangolo



introducendo una nuova variabile  $z$ , e ponendo  $y = xz$ ,abbiamo

$$\iint_T f(x, y) dx dy = \int_0^1 \int_0^x f(x, y) dy dx = \int_0^1 \int_0^1 f(x, xz) x dz dx$$

## Bibliografia

- [7.1] V. I. Krylov, *Approximate calculation of integrals*, McMillan, New York, 1962.
- [7.2] M. Abramowitz, I. A. Stegun, *Handbook of mathematical functions*, National Bureau of Standards, Applied Mathematics Series 55, Washington D. C., 1964.
- [7.3] A. S. Kronrod, *Nodes and weights of quadrature formulas*, Consultants Bureau, New York, 1965.
- [7.4] A. H. Stroud, D. Secrest, *Gaussian quadrature formulas*, Prentice-Hall, Englewood Cliffs, New Jersey, 1966.
- [7.5] A. H. Stroud, *Approximate calculation of multiple integrals*, Prentice-Hall, Englewood Cliffs, New Jersey, 1971.
- [7.6] L. Gatteschi, *Funzioni speciali*, UTET, Torino, 1973.
- [7.7] G. Szegö, *Orthogonal polynomials*, Amer. Math. Soc. Colloquium Publications, v. 23, Providence, R. I., 1975.
- [7.8] K. E. Atkinson, *Survey of numerical methods for the solution of Fredholm integral equations of the second kind*, SIAM, Philadelphia, 1976.
- [7.9] C. T. H. Baker, *The numerical treatment of integral equations*, Clarendon Press, Oxford, 1977.
- [7.10] T. S. Chihara, *An introduction to orthogonal polynomials*, Gordon and Breach, New York, 1978.
- [7.11] H. Engels, *Numerical quadrature and cubature*, Academic Press, Londra, 1980.
- [7.12] R. Piessens, E. de Doncker, C. W. Überhuber, D. K. Kahaner, *Quadpack, a subroutine package for automatic integration*, Springer-Verlag, Heidelberg, 1983.
- [7.13] P. J. Davis, P. Rabinowitz, *Methods of numerical integration*, Academic Press, New York, 1984.

## Esercizi proposti

**7.1.** Supponiamo di voler costruire la seguente formula di quadratura di tipo interpolatorio

$$\int_0^1 f(x) dx \approx w_1 f(0) + w_2 f'(0) + w_3 f(1).$$

Determinare i pesi  $w_1$ ,  $w_2$ ,  $w_3$  e dire qual è il grado di precisione della formula.

**7.2.** Verificare che una formula di tipo interpolatorio

$$\int_{-1}^1 w(x) f(x) dx \approx \sum_{i=1}^n w_i f(x_i)$$

con  $w(x) = w(-x)$  e  $x_{n+1-i} = -x_i$  è simmetrica, ovvero  $w_{n+1-i} = w_i$ .

**7.3.** Partendo dalla definizione di sistema di polinomi ortogonali, dedurre la relazione di ricorrenza (7.7).

**7.4.** Approssimare l'integrale

$$\int_{-4}^4 \frac{dx}{1+x^2} = 2 \arctan 4$$

con la formula di Gauss-Legendre con  $n = 2, 4, 6, 8$ . Confrontare la bontà dei risultati ottenuti con le stime fornite dalle corrispondenti formule di Newton-Cotes con  $n = 2, 4, 6, 8$ . I nodi e i pesi delle formule suddette possono essere reperiti in [7.2].

**7.5.** Utilizzare le formule di Gauss-Legendre con  $n$  crescente (finché la precisione desiderata non sia stata raggiunta, oppure il numero massimo di nodi consentito non sia stato superato) per approssimare gli integrali seguenti:

$$\begin{aligned} \int_0^1 e^x dx &= e - 1 & \int_0^1 \cos x dx &= \sin(1) \\ \int_{0.01}^{1.1} \frac{1}{x^4} dx &= \frac{1}{3}[10^6 - (1.1)^{-3}] & \int_0^1 \sqrt{x} dx &= \frac{2}{3} \\ \int_0^1 \sin(100\pi x) dx &= 0 \end{aligned}$$

Commentare i risultati.

**7.6.** Proporre un'unica formula di quadratura (ottimale) per il calcolo “esatto” di tutti i coefficienti  $\{d_k\}$  dello sviluppo (7.8).

**7.7.** Dimostrare che l'algoritmo di Clenshaw esposto a pagina 226 effettivamente produce il valore  $Q_m(\bar{x})$ .

**7.8.** Proporre una formula di quadratura per il calcolo di integrali di tipo

$$\int_0^\infty e^{-x^2} f(x^2) dx$$

**7.9.** Calcolare l'integrale

$$\int_1^\infty e^{-x} x^{3/2} dx$$

**7.10.** Verificare che nella formula di Gauss-Lobatto (7.14) i nodi  $\{x_i\}$  coincidono con gli zeri del polinomio  $P_n(x)$  ortogonale in  $(a, b)$  rispetto alla funzione peso  $w(x)(x-a)(b-x)$ .

**7.11.** Qual è la formula più efficace per calcolare l'integrale

$$\int_{-1}^1 \frac{1}{\sqrt{1-x^2}} P_9(x) dx$$

dove  $P_9(x)$  è un polinomio di grado 9?

**7.12.** Calcolare l'integrale

$$\int_0^1 x \log x f(x) dx$$

dove  $f(x)$  è supposta regolare. Introdurre il cambiamento di variabile  $x = t^k$ . Osservare il comportamento della nuova funzione integranda e quindi applicare la formula di Gauss-Legendre.

**7.13.** Costruire una formula di quadratura per l'integrale

$$\int_a^b f(x) dx$$

approssimando la funzione integranda con la spline cubica naturale che intercala la  $f(x)$  nei nodi  $a = x_1 < x_2 < \dots < x_{n-1} < x_n = b$ . Esaminare il comportamento dell'errore.

**7.14.** Dimostrare che:

- (i) la formula composta dei trapezi, con  $h = 2\pi/(n+1)$ , è esatta per tutti i polinomi trigonometrici di grado  $\leq n$  e periodo  $2\pi$  quando l'intervallo di integrazione è  $b-a = 2\pi$ ;
- (ii) fissato un intero  $n$ , se la funzione integranda  $f(x)$  può essere approssimata con un polinomio trigonometrico  $T_n(x)$  di grado  $n$  tale che  $|f(x)-T_n(x)| < \varepsilon$  per  $x \in [0, 2\pi]$ , il valore assoluto dell'errore commesso nell'approssimazione dell'integrale

$$\frac{1}{2\pi} \int_0^{2\pi} f(x) dx$$

con la formula composta dei trapezi, con  $h = 2\pi/(n+1)$ , è inferiore a  $2\varepsilon$ .

**7.15.** Qual è la formula più efficiente per calcolare un integrale del tipo

$$\int_{-\pi/2}^{\pi/2} f(\cos 2x, \sin 2x) dx$$

dove  $f(x)$  è regolare?

**7.16.** Sia data la formula di Gauss-Legendre

$$\int_{-1}^1 f(x) dx \approx \sum_{i=1}^d w_i f(x_i)$$

Costruire la versione composta per il calcolo dell'integrale

$$\int_a^b f(x) dx$$

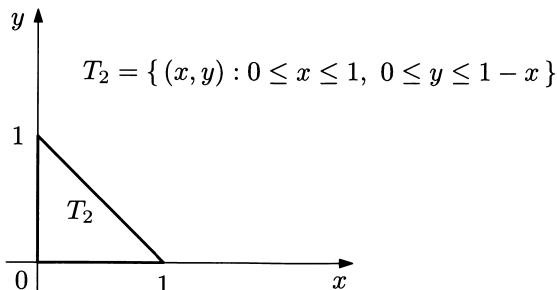
Qual è il grado di precisione della formula composta?

**7.17.** Costruire una routine automatica di tipo adattativo che implementi la strategia descritta a pagina 241.

**7.18.** Proporre una formula prodotto per l'approssimazione dell'integrale

$$\iint_{T_2} f(x, y) dx dy$$

dove



Successivamente, ricordando che ogni triangolo è *affine* a  $T_2$ , scrivere la corrispondente formula prodotto per l'approssimazione dell'integrale di  $f(x, y)$  su un generico triangolo di vertici  $(x_a, y_a)$ ,  $(x_b, y_b)$ ,  $(x_c, y_c)$ .

**7.19.** Proporre una formula prodotto per il calcolo dell'integrale

$$\iint_S f(x, y) dx dy$$

dove  $S$  è un cerchio di raggio  $R$  e centro  $(x_0, y_0)$ . Utilizzare le coordinate polari.

**7.20.** Calcolare l'integrale (vedi pagina 246)

$$\int_0^1 \int_0^x \frac{f(x, y)}{\sqrt{x^2 + y^2}} dx dy$$

dove  $f(x, y)$  è supposta regolare. Introdurre il cambiamento di variabile  $y = xz$ . Osservare il comportamento della nuova funzione e quindi proporre una formula prodotto.

# Capitolo 8

## Equazioni differenziali ordinarie

### 8.1 Preliminari

Molti fenomeni fisici sono descrivibili con modelli matematici che, direttamente o indirettamente, richiedono la soluzione di una o più equazioni differenziali ordinarie del tipo

$$(8.1) \quad \begin{cases} y'_1(x) = f_1(x, y_1(x), \dots, y_m(x)) \\ \vdots \\ y'_m(x) = f_m(x, y_1(x), \dots, y_m(x)) \end{cases}$$

con condizioni (iniziali) in un unico punto  $x = a$

$$(8.2) \quad \begin{cases} y_1(a) = y_{1,0} \\ \vdots \\ y_m(a) = y_{m,0} \end{cases}$$

In questi casi l'obiettivo matematico diventa la determinazione di funzioni (con derivate prime continue)  $y_1(x), \dots, y_m(x)$  soluzione del sistema (8.1) e passanti per i punti definiti dalla (8.2).

Ricordiamo che anche le equazioni differenziali di ordine  $m$

$$y^{(m)}(x) = f(x, y(x), y'(x), \dots, y^{(m-1)}(x))$$

con condizioni iniziali

$$\begin{cases} y(a) = y_{1,0} \\ y'(a) = y_{2,0} \\ \vdots \\ y^{(m-1)}(a) = y_{m,0} \end{cases}$$

possono essere facilmente ricondotte a sistemi di tipo (8.1), (8.2). È infatti sufficiente introdurre i seguenti cambiamenti di nome di funzione

$$\begin{aligned} z_1(x) &= y(x) \\ z_2(x) &= y'(x) \\ &\vdots \\ z_m(x) &= y^{(m-1)}(x) \end{aligned}$$

per ottenere il sistema

$$\begin{cases} z'_1(x) = z_2(x) & z_1(a) = y_{1,0} \\ \vdots \\ z'_{m-1}(x) = z_m(x) & z_{m-1}(a) = y_{m-1,0} \\ z'_m(x) = f(x, z_1(x), z_2(x), \dots, z_m(x)) & z_m(a) = y_{m,0} \end{cases}$$

Altre volte le condizioni associate al sistema (8.1) sono su due punti distinti  $x = a$  e  $x = b$ , cioè del tipo

$$(8.3) \quad \begin{cases} g_1(y_1(a), \dots, y_m(a); y_1(b), \dots, y_m(b)) = 0 \\ \vdots \\ g_m(y_1(a), \dots, y_m(a); y_1(b), \dots, y_m(b)) = 0 \end{cases}$$

In questo caso si parla di *problema ai limiti*<sup>(†)</sup>.

Per alleggerire il formalismo nelle pagine che seguono, introduciamo le notazioni vettoriali

$$\begin{aligned} y(x) &= \begin{pmatrix} y_1(x) \\ \vdots \\ y_m(x) \end{pmatrix} & f(x, y(x)) &= \begin{pmatrix} f_1(x, y_1(x), \dots, y_m(x)) \\ \vdots \\ f_m(x, y_1(x), \dots, y_m(x)) \end{pmatrix} \\ y_0 &= \begin{pmatrix} y_{1,0} \\ \vdots \\ y_{m,0} \end{pmatrix} & g(y(a), y(b)) &= \begin{pmatrix} g_1(y_1(a), \dots, y_m(a); y_1(b), \dots, y_m(b)) \\ \vdots \\ g_m(y_1(a), \dots, y_m(a); y_1(b), \dots, y_m(b)) \end{pmatrix} \end{aligned}$$

---

(†) Di solito siamo interessati ai valori che la soluzione del problema assume nell'intervallo  $[a, b]$ .

così che i sistemi precedenti possano essere riformulati nelle forme più compatte

$$(8.4) \quad \begin{cases} y'(x) = f(x, y(x)) \\ y(a) = y_0 \end{cases}$$

$$(8.5) \quad \begin{cases} y'(x) = f(x, y(x)) \\ g(y(a), y(b)) = 0 \end{cases}$$

e trattati, almeno formalmente, come se fossero costituiti da un'unica equazione.

Nei paragrafi che seguiranno ci occuperemo della costruzione di approssimazioni numeriche delle soluzioni dei problemi a valori iniziali (ossia con condizioni iniziali) di forma (8.4). Tuttavia, prima di procedere allo studio di metodi numerici ricordiamo alcuni importanti risultati di carattere teorico riguardanti l'esistenza e il comportamento delle soluzioni di problemi di tipo (8.4).

**Teorema 8.1.** (vedi ad es. [8.2, pagina 113]). *Sia  $f(x, y)$  definita e continua nella striscia infinita*

$$S = \{(x, y) : -\infty < \alpha \leq x \leq \beta < \infty, y \in \mathbb{R}^m\}$$

*Supponiamo inoltre che la funzione  $f(x, y)$  sia, in  $S$ , lipschitziana nella variabile  $y$ , uniformemente rispetto a  $x$ , ovvero esista una costante  $L > 0$  tale che*

$$\|f(x, y_1) - f(x, y_2)\| \leq L \|y_1 - y_2\| \quad (\text{condizione di Lipschitz})$$

*per ogni  $x \in [\alpha, \beta]$  e per ogni coppia  $y_1, y_2 \in \mathbb{R}^m$ . Allora per ogni  $a \in [\alpha, \beta]$  e per ogni  $y_0 \in \mathbb{R}^m$  esiste esattamente una funzione  $y(x)$  tale che*

- $$(8.6) \quad \begin{aligned} \text{(i)} \quad & y(x) \in C^1[\alpha, \beta] \\ \text{(ii)} \quad & y'(x) = f(x, y(x)) \quad \text{per ogni } x \in [\alpha, \beta] \\ \text{(iii)} \quad & y(a) = y_0 \end{aligned}$$

Osserviamo che la condizione di lipschitzianità della  $f(x, y)$  è certamente soddisfatta quando le derivate parziali  $\partial f_i / \partial y_j$ ,  $i, j = 1, \dots, m$ , esistono in  $S$  e sono ivi continue e limitate.

Il teorema precedente ci assicura che il problema (8.4) ammette, nell'intervallo  $[\alpha, \beta]$ , una ed una sola soluzione quando la funzione  $f(x, y)$  soddisfa le predette condizioni nella striscia  $S$ . Quando invece tali condizioni non sono tutte verificate, il problema potrebbe non ammettere soluzione, oppure averne infinite e richiedere quindi ulteriori condizioni per individuarne una particolare.

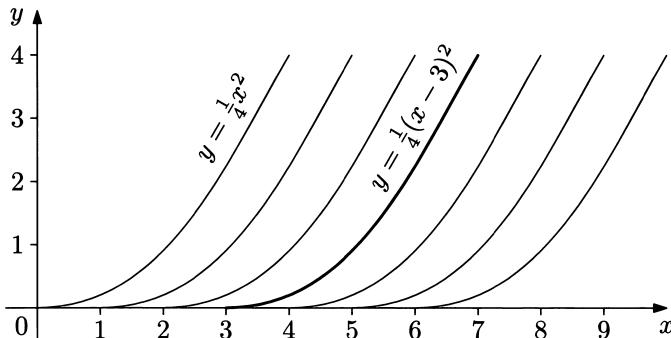
Per esempio, nel problema

$$\begin{cases} y'(x) = \lambda y(x) \\ y(a) = y_0 \end{cases}$$

le condizioni richieste dal teorema 8.1 sono soddisfatte:  $f(x, y) = \lambda y$  e  $\partial f / \partial y = \lambda$ . Il problema ammette pertanto una ed una sola soluzione,  $y(x) = y_0 e^{\lambda(x-a)}$ , in qualunque intervallo  $[\alpha, \beta]$ ,  $a \in [\alpha, \beta]$ . Nel caso seguente invece

$$\begin{cases} y'(x) = \sqrt{|y(x)|} \\ y(0) = 0 \end{cases}$$

essendo  $f(x, y) = \sqrt{|y|}$ , non abbiamo la lipschitzianità della  $f(x, y)$  su  $y = 0$ . Infatti, risulta  $\partial f / \partial y = y^{-1/2}/2$  per  $y > 0$  e  $\partial f / \partial y = -(-y)^{-1/2}/2$  per  $y < 0$



**Figura 8.1**

Il problema in questione ammette infinite soluzioni:  $y = 0$  e  $y = x^2/4$  sono due di esse. Ma non sono le sole: ogni curva composta da una porzione della retta  $y = 0$  e da una della parabola  $y = (x - c)^2/4$ , con  $c$  scelto in modo che tra le due vi sia continuità (vedi, per esempio, la curva disegnata in neretto in figura 8.1), è soluzione del problema. Tuttavia, se richiediamo che  $y'(x)$ ,  $x > 0$ , sia (continua e) positiva, allora definiamo un'unica soluzione:  $y = x^2/4$ .

Spesso nelle applicazioni  $f \in C^1(S)$ , ma la sua derivata parziale (Jacobiano nel caso di sistemi)  $\partial f / \partial y$  non risulta limitata quando  $\|y\| \rightarrow \infty$ . In questo caso il problema (8.4) ammette ancora una soluzione di tipo (8.6), ma l'esistenza e unicità di quest'ultima è garantita solo in un intorno del punto iniziale  $x = a$ , e non necessariamente in tutto un intervallo  $[\alpha, \beta]$ ,  $a \in [\alpha, \beta]$ , fissato a priori (vedi [8.11, cap. 1]). Per esempio, la soluzione  $y(x) = 1/(1-x)$  (vedi figura 8.2) del problema

$$\begin{cases} y'(x) = [y(x)]^2 \\ y(0) = 1 \end{cases}$$

è di classe  $C^1$  solo per  $x < 1$ .

Il teorema che segue, la cui dimostrazione può essere reperita in [6, v. 2, pag. 103], ci garantisce che il problema (8.4), nelle ipotesi del teorema 8.1, è *ben posto*; ossia ci assicura

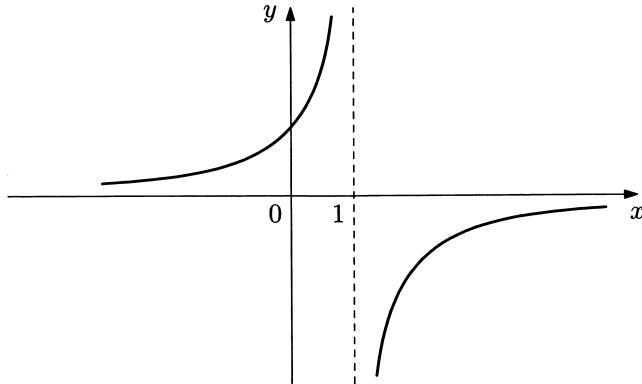


Figura 8.2

che il problema non solo ammette un'unica soluzione, ma anche che quest'ultima è una funzione continua del dato iniziale  $y_0$ .

**Teorema 8.2.** *Sia  $f : S \rightarrow \mathbb{R}^m$  continua in  $S$ ; supponiamo inoltre che la condizione di Lipschitz*

$$\|f(x, y_1) - f(x, y_2)\| \leq L\|y_1 - y_2\|$$

*sia soddisfatta per tutte le coppie di punti  $(x, y_1), (x, y_2)$  di  $S$ . Allora, per la soluzione (unica)  $y(x) \equiv y(x; y_0)$  del problema*

$$(8.7) \quad \begin{cases} y'(x) = f(x, y(x)), & \alpha \leq x \leq \beta \\ y(a; y_0) = y_0, & a \in [\alpha, \beta] \end{cases}$$

*vale la seguente diseguaglianza:*

$$\|y(x; y_1) - y(x; y_0)\| \leq e^{L|x-a|} \|y_1 - y_0\|, \quad \alpha \leq x \leq \beta$$

Se  $y(x; y_0)$  rappresenta la soluzione di (8.7), e  $y(x; y_0 + \varepsilon)$  quella dello stesso problema ma con condizione iniziale perturbata di  $\varepsilon$  ( $y(a; y_0 + \varepsilon) = y_0 + \varepsilon$ ), il teorema precedente ci assicura che, fissata una tolleranza  $\delta > 0$ , è sempre possibile avere

$$\|y(x; y_0 + \varepsilon) - y(x; y_0)\| \leq \delta$$

in tutto  $[\alpha, \beta]$ , purché la perturbazione iniziale  $\varepsilon = \varepsilon(\delta)$  sia sufficientemente piccola (figura 8.3).

Tuttavia, poiché generalmente non conosceremo esattamente l'ordinata  $y_0$ , ma avremo solo una sua approssimazione, per esempio  $y_0 + \varepsilon = \text{fl}(y_0)$ , e la perturbazione  $\varepsilon$  per piccola che sia non potrà tendere a zero, ma rimarrà fissa, è importante sapere di quanto la nuova curva  $y(x; y_0 + \varepsilon)$  si discosta da quella,  $y(x; y_0)$ , che doveva essere il nostro

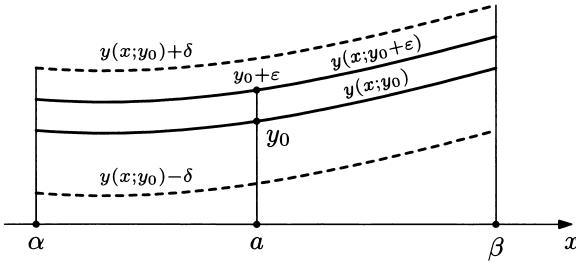


Figura 8.3

obiettivo. Questo studio, che definisce il condizionamento del problema (8.7) in relazione al dato iniziale  $y_0$ , può essere fatto con esattezza solo per problemi lineari del tipo

$$(8.8) \quad \begin{cases} y'(x) = Ay(x) + g(x) \\ y(a) = y_0 \end{cases}$$

dove  $A \in \mathbb{R}^{m \times m}$ <sup>(†)</sup> è diagonalizzabile, cioè ammette un sistema di  $m$  autovettori  $\{\xi_l\}$  linearmente indipendenti, cosicché, posto  $H = (\xi_1, \dots, \xi_m)$ , risulta

$$(8.9) \quad H^{-1}AH = \Lambda = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_m \end{pmatrix}$$

con  $A\xi_l = \lambda_l \xi_l$ ,  $l = 1, \dots, m$ . In generale dovremo invece accontentarci di linearizzare localmente il problema, approssimarlo con uno di tipo (8.8) ed esaminare quest'ultimo.

Pertanto, consideriamo dapprima il caso (8.8), cui associamo il problema perturbato

$$(8.10) \quad \begin{cases} z'(x) = Az(x) + g(x) \\ z(a) = y_0 + \varepsilon \end{cases}$$

essendo  $z(x) = y(x; y_0 + \varepsilon)$ . Posto  $\delta(x) = z(x) - y(x)$ ,

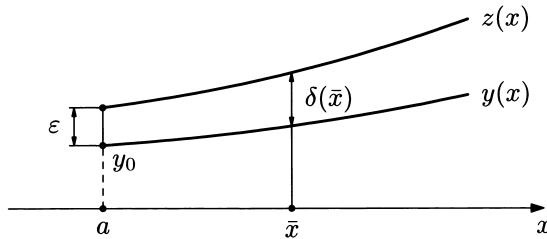


Figura 8.4

(†) Ovviamente, nel caso di una singola equazione avremo  $A \in \mathbb{R}$ .

sottraendo la (8.8) dalla (8.10) otteniamo

$$(8.11) \quad \begin{cases} \delta'(x) = A\delta(x) \\ \delta(a) = \varepsilon \end{cases}$$

Utilizzando la trasformazione (8.9) possiamo disaccoppiare il sistema (8.11); infatti abbiamo

$$\begin{cases} H^{-1}\delta'(x) = H^{-1}AHH^{-1}\delta(x) \\ H^{-1}\delta(a) = H^{-1}\varepsilon \end{cases}$$

e, ponendo  $d(x) = H^{-1}\delta(x)$ ,

$$\begin{cases} d'(x) = \Lambda d(x) \\ d(a) = \eta, \quad \eta = H^{-1}\varepsilon \end{cases}$$

donde

$$(8.12) \quad \begin{cases} d'_l(x) = \lambda_l d_l(x) \\ d_l(a) = \eta_l \end{cases} \quad l = 1, \dots, m$$

Ricordando che

$$d_l(x) = \eta_l e^{\lambda_l(x-a)}$$

possiamo dare a  $\delta(x)$  la rappresentazione seguente

$$\delta(x) = Hd(x) = \sum_{l=1}^m \eta_l e^{\lambda_l(x-a)} \xi_l$$

Gli autovalori  $\{\lambda_l\}$  della matrice  $A$  caratterizzano la risposta del sistema (8.8) all'introduzione di perturbazioni nel dato iniziale  $y_0$ ; in particolare, supponendo che le curve dell'equazione differenziale in (8.8) siano definite in  $[a, \infty)$ , abbiamo  $\|\delta(x)\| \rightarrow \infty$ , per  $x \rightarrow \infty$ , quando  $\text{Re}(\lambda_l) > 0$  per almeno un indice  $l$ , e  $\|\delta(x)\| \rightarrow 0$  quando  $\text{Re}(\lambda_l) < 0$ ,  $l = 1, \dots, m$ . In quest'ultimo caso definiamo la soluzione del problema (8.8) *asintoticamente stabile*. La curva  $y(x)$  viene detta *semplicemente stabile* se  $\|\delta(x)\|$  si mantiene limitata in  $[a, \infty)$ .

Nel caso invece di un generico sistema non lineare

$$(8.13) \quad \begin{cases} y'(x) = f(x, y(x)) \\ y(a) = y_0 \end{cases}$$

occorre sviluppare la funzione  $f(x, z(x)) = f(x, y(x) + \delta(x))$  del sistema perturbato

$$\begin{cases} z'(x) = f(x, z(x)) \\ z(a) = y_0 + \varepsilon \end{cases}$$

in serie di Taylor<sup>(†)</sup> nell'intorno di  $(x, y(x))$ :

$$(8.14) \quad f(x, z(x)) = f(x, y(x)) + f_y(x, y(x))\delta(x) + O(\|\delta\|^2)$$

e, supponendo che il termine  $O(\|\delta\|^2)$  sia trascurabile, considerare la sola parte lineare del problema:

$$\begin{cases} y'(x) + \delta'(x) \cong f(x, y(x)) + f_y(x, y(x))\delta(x) \\ y(a) + \delta(a) = y_0 + \varepsilon \end{cases}$$

da cui segue

$$\begin{cases} \delta'(x) \cong f_y(x, y(x))\delta(x) \\ \delta(a) = \varepsilon \end{cases}$$

Assumendo infine che lo Jacobiano  $f_y(x, y(x))$  sia pressoché costante, cioè  $f_y(x, y(x)) \cong f_y(a, y_0)$ , e sostituendo le uguaglianze approssimate con delle uguaglianze vere, possiamo affermare che, in prima approssimazione, la propagazione dell'errore iniziale  $\varepsilon$  è definita dal sistema

$$\begin{cases} \delta'(x) = f_y(a, y_0)\delta(x) \\ \delta(a) = \varepsilon \end{cases}$$

di tipo (8.11). Nel caso in cui il problema (8.13) sia definito in tutto  $[a, \infty)$  diremo che la soluzione  $y(x)$  è asintoticamente stabile in prima approssimazione quando tutti gli autovalori della matrice  $f_y(a, y_0)$  hanno parte reale negativa.

Concludiamo questo esame preliminare ribadendo che lo studio della propagazione di una perturbazione iniziale  $\varepsilon$  è stato possibile solo supponendo trascurabile il termine  $O(\|\delta\|^2)$  in (8.14) e costante lo Jacobiano  $f_y(x, y(x))$ . In realtà queste ipotesi spesso non sono verificate e il comportamento di  $\delta(x)$  può non essere validamente rappresentato dagli autovalori di  $f_y(a, y_0)$ .

Con i metodi numerici che presenteremo nei prossimi paragrafi dovremo accontentarci di determinare, in un numero finito di punti  $\{x_n\}$  dell'intervallo di interesse  $[\alpha, \beta]$ , delle approssimazioni  $\{y_n\}$  dei valori che la soluzione  $y(x)$  assume nei nodi  $\{x_n\}$ . Ovviamente, note le  $\{y_n\}$ , mediante l'uso delle tecniche di interpolazione descritte nel capitolo 5 sarà sempre possibile dedurre approssimazioni della  $y(x)$  in punti  $x$  diversi dagli  $\{x_n\}$ . I predetti metodi verranno da noi raggruppati in due classi principali:

1. *Metodi one-step*, o ad un solo passo. Il valore  $y_{n+1}$  viene calcolato utilizzando unicamente l'approssimazione precedente  $y_n$ ; ossia il metodo si presenta in una delle due forme seguenti:

$$y_{n+1} = y_n + h\Phi(x_n, y_n; h) \quad \text{metodo one-step } esplicito$$

$$y_{n+1} = y_n + h\Phi(x_n, y_n, y_{n+1}; h) \quad \text{metodo one-step } implicito$$

con  $x_{n+1} = x_n + h$ . Generalmente  $h = h(n)$ .

---

(†) Supponendo  $f \in C^2(S)$ .

2. *Metodi multistep*, o a più passi. Il valore  $y_{n+1}$  viene definito utilizzando più approssimazioni precedenti  $y_n, y_{n-1}, \dots, y_{n-k+1}$  relative ai punti  $x_n, x_{n-1}, \dots, x_{n-k+1}$ ,  $x_i = a + ih$ , con formule del tipo

$$y_{n+1} = \Psi(x_n, y_n, y_{n-1}, \dots, y_{n-k+1}; h) \quad (\text{metodo multistep esplicito})$$

oppure

$$y_{n+1} = \Psi(x_n, y_{n+1}, y_n, \dots, y_{n-k+1}; h) \quad (\text{metodo multistep implicito})$$

l'intero  $k$  va supposto fisso per tutti gli  $n$ . Il metodo viene denominato *multistep* a  $k$  passi.

Per esempio

$$y_{n+1} = y_{n-1} + 2hf(x_n, y_n), \quad n = 1, 2, \dots$$

è un metodo multistep esplicito, mentre la formula

$$y_{n+1} = y_n + \frac{h}{2}[f(x_n, y_n) + f(x_{n+1}, y_{n+1})]$$

rappresenta un metodo one-step implicito.

## 8.2 Metodi one-step esplicativi. Metodi Runge-Kutta

Consideriamo il seguente problema a valori iniziali

$$(8.15) \quad \begin{cases} y'(x) = f(x, y(x)), & a \leq x \leq b \\ y(a) = y_0 \end{cases}$$

Il generico metodo one-step esplicito è definito da una relazione del tipo

$$(8.16) \quad \begin{cases} y_{n+1} = y_n + h_n \Phi(x_n, y_n; h_n) \\ x_0 = a \\ x_{n+1} = x_n + h_n \end{cases} \quad n = 0, 1, \dots$$

La funzione  $\Phi(x_n, y_n; h_n) \equiv \Phi(x_n, y_n; h_n; f)$ , che supporremo continua nelle variabili  $x_n, y_n, h_n$ , definisce il metodo. Raramente il passo  $h_n$  viene mantenuto costante per tutti gli  $n$ .

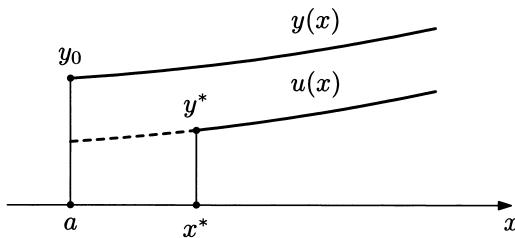
Come vedremo più avanti, il grado di accuratezza delle approssimazioni fornite da un metodo one-step è strettamente legato al *comportamento locale* del metodo stesso, ossia all'errore che il metodo introduce quando opera un singolo passo di integrazione. Pertanto, prima di costruire esplicitamente dei metodi numerici, introduciamo alcune definizioni che ci consentano poi di valutare e confrontare tra di loro i comportamenti locali dei diversi metodi.

### 8.2.1 Comportamento locale dei metodi one-step

Dato un generico punto  $(x^*, y^*) \in S$ , consideriamo la soluzione del problema

$$\begin{cases} u'(x) = f(x, u(x)), & x^* \leq x \leq b \\ u(x^*) = y^* \end{cases}$$

ossia la curva soluzione dell'equazione differenziale  $v'(x) = f(x, v(x))$  che esce dal punto  $(x^*, y^*)$ . Ricordiamo che con  $y(x)$  abbiamo denotato la curva (unica), della stessa equazione differenziale  $v'(x) = f(x, v(x))$ , che passa per il punto  $(a, y_0)$ .

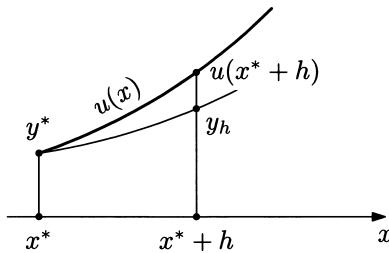


**Figura 8.5**

Il metodo one-step scelto, partendo dal punto  $(x^*, y^*)$ , avrà come obiettivo la curva  $u(x)$ , che è la soluzione del problema cui esso viene applicato in quell'“istante”; cioè “cercherà”, in generale senza riuscirvi, di seguire la  $u(x)$ . In realtà il metodo, considerato come funzione continua del passo  $h$ , definirà una seconda curva

$$(8.17) \quad y_h = y^* + h\Phi(x^*, y^*; h)$$

che esce dal punto  $(x^*, y^*)$  ma che in generale non coinciderà con la  $u(x)$ :



**Figura 8.6**

**Definizione 8.1.** Per ogni  $(x^*, y^*) \in S$  e  $h > 0$ , definiamo errore locale unitario di troncamento del metodo  $\Phi$ , nel punto  $x^* + h$ , la funzione

$$(8.18) \quad t(x^*, y^*; h) = \frac{1}{h} [u(x^* + h) - u(x^*) - h\Phi(x^*, u(x^*); h)]^{\dagger}$$

La quantità  $t(x^*, y^*; h)$  ci dà una indicazione della bontà dello schema discreto (8.17), il quale, scritto nella forma

$$(8.19) \quad \frac{y_h - y^*}{h} = \Phi(x^*, y^*; h)$$

rappresenta un'approssimazione dell'identità  $u'(x^*) = f(x^*, y^*)$ . Infatti, tale errore coincide con il residuo prodotto dalla (8.19) quando in quest'ultima introduciamo la soluzione  $u(x)$ , ovvero sostituiamoci  $y_h$ , con  $u(x^* + h)$ . Per un metodo di forma (8.17),  $t(x^*, y^*; h)$  corrisponde anche alla quantità  $[u(x^* + h) - y_h]/h$  (vedi figura 8.6).

Prima di proseguire, denotiamo con  $F_N(S)$  la classe delle funzioni  $f(x, y)$  le cui derivate parziali di ordine  $\leq N$  esistono in  $S$  e sono ivi continue e limitate.

**Definizione 8.2.** Il metodo  $\Phi$  è detto consistente se per ogni  $(x^*, y^*) \in S$  e qualunque  $f \in F_1(S)$  risulta

$$\lim_{h \rightarrow 0} t(x^*, y^*; h) = 0$$

La relazione (8.18) ci consente di dare una formulazione alternativa a quest'ultima definizione:

**Definizione 8.3.** Il metodo  $\Phi$  è consistente se per ogni  $(x^*, y^*) \in S$  e qualunque  $f \in F_1(S)$  risulta

$$\lim_{h \rightarrow 0} \Phi(x^*, y^*; h) = f(x^*, y^*)$$

Infatti,  $\lim_{h \rightarrow 0} t(x^*, y^*; h) = 0$  se e solo se

$$\lim_{h \rightarrow 0} \Phi(x^*, y^*; h) = \lim_{h \rightarrow 0} \frac{1}{h} [u(x^* + h) - u(x^*)] = u'(x^*) = f(x^*, y^*)$$

**Definizione 8.4.** Il metodo  $\Phi$  ha ordine (di consistenza)  $p$ , intero positivo, se per tutti i punti  $(x^*, y^*) \in S$  e qualunque  $f \in F_p(S)$  risulta

$$(8.20) \quad t(x^*, y^*; h) = O(h^p), \quad h \rightarrow 0$$

e  $p$  è l'intero più grande per cui vale la (8.20).

Osserviamo che  $p > 0$  implica la consistenza del metodo.

---

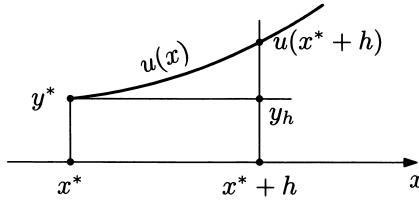
(†) Nel caso di metodi impliciti  $y_{n+1} = y_n + h\Phi(x_n, y_n, y_{n+1}; h)$  l'errore locale unitario di troncamento viene definito dall'analogia relazione

$$t(x^*, y^*; h) = \frac{1}{h} [u(x^* + h) - u(x^*) - h\Phi(x^*, u(x^*), u(x^* + h); h)]$$

### 8.2.2 Esempi di metodi one-step esplicativi

1. *Metodo di Eulero:*

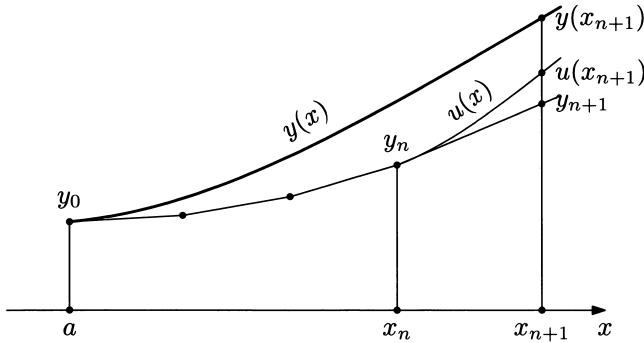
$$\Phi(x^*, y^*; h) = f(x^*, y^*), \quad y_h = y^* + h f(x^*, y^*) \quad (\dagger)$$



**Figura 8.7**

ovvero

$$y_{n+1} = y_n + h f(x_n, y_n), \quad n = 0, 1, \dots$$



**Figura 8.8**

Il metodo è, ovviamente, consistente; inoltre, per quanto riguarda l'errore locale unitario di troncamento abbiamo

$$t(x^*, y^*; h) = \frac{1}{h} [u(x^* + h) - u(x^*)] - f(x^*, y^*)$$

---

(†) Il numero  $f(x^*, y^*)$  rappresenta il coefficiente angolare della retta tangente alla curva di riferimento  $u(x)$  nel punto  $(x^*, y^*)$ .

Poiché  $f(x^*, y^*) = f(x^*, u(x^*)) = u'(x^*)$ , richiamando la formula di Taylor possiamo scrivere

$$\begin{aligned} t(x^*, y^*; h) &= \frac{1}{h}[u(x^*) + hu'(x^*) + \frac{1}{2}h^2u''(\eta) - u(x^*)] - u'(x^*) = \\ &= \frac{1}{2}hu''(\eta), \quad x^* < \eta < x^* + h \end{aligned}$$

Nel caso di un sistema di equazioni differenziali abbiamo una  $\eta_i$ ,  $x^* < \eta_i < x^* + h$ , diversa per ogni componente  $u''_i(x)$  del vettore  $u''(x)$ .

Supponendo  $f \in F_1(S)$  otteniamo

$$u''(x) = f_x(x, u(x)) + f_u(x, u(x))f(x, u(x))$$

quindi

$$\|u''(x)\| \leq M_2 \quad \text{in } S$$

Pertanto

$$t(x^*, y^*; h) = O(h), \quad h \rightarrow 0$$

ovvero il metodo di Eulero ha ordine di consistenza  $p = 1$ .

## 2. Metodi Runge-Kutta.

Ripercorrendo il precedente studio dell'errore locale di troncamento nel metodo di Eulero, scopriamo che per ottenere metodi one-step espliciti di ordine  $p \geq 2$  è sufficiente prendere come funzione  $\Phi(x^*, y^*; h)$  la ridotta di ordine  $p$  dello sviluppo in serie di Taylor del rapporto incrementale  $[u(x^* + h) - u(x^*)]/h$ . Tuttavia tali metodi risultano poco competitivi in quanto coinvolgono le derivate parziali, di ordine  $\leq p$ , della  $f(x, u)$  e richiedono la valutazione di queste ultime ad ogni singolo passo.

Metodi più efficienti di ordine  $p \geq 2$  possono essere costruiti scegliendo come  $\Phi$  una combinazione lineare di valori della  $f$ ; per esempio, nel caso  $p = 2$

$$(8.21) \quad \left\{ \begin{array}{l} \Phi(x^*, y^*; h) = a_1k_1 + a_2k_2 \\ k_1 = f(x^*, y^*) \\ k_2 = f(x^* + b_2h, y^* + b_2hk_1) \end{array} \right.$$

con coefficienti  $a_1$ ,  $a_2$  e  $b_2$  scelti in modo che lo sviluppo in serie di Taylor di  $t(x^*, y^*; h)$  nell'intorno di  $(x^*, y^*; 0)$  inizi con la potenza di  $h$  di ordine il più elevato possibile.

Sviluppando  $\Phi(x^*, y^*; h)$  in serie di Taylor nell'intorno del punto  $(x^*, y^*; 0)$  otteniamo

$$\begin{aligned} \Phi(x^*, y^*; h) &= a_1f(x^*, y^*) + a_2[f(x^*, y^*) + b_2hf_x(x^*, y^*) + b_2hb_2[f_x(x^*, y^*)f(x^*, y^*)] + O(h^2)] + O(h^2) \\ &= (a_1 + a_2)f(x^*, y^*) + a_2hb_2[f_x(x^*, y^*) + f_u(x^*, y^*)f(x^*, y^*)] + O(h^2) \end{aligned}$$

Il metodo  $\Phi$  avrà ordine  $p = 2$  se e solo se risulteranno nulli i coefficienti delle potenze  $h^0$  e  $h^1$  dello sviluppo in serie dell'errore locale di troncamento  $t(x^*, y^*; h)$ , cioè se e solo

se i coefficienti  $a_1, a_2$  e  $b_2$  soddisfano le condizioni

$$(8.22) \quad \begin{cases} a_1 + a_2 = 1 \\ a_2 b_2 = \frac{1}{2} \end{cases}$$

Una soluzione di tale sistema è

$$a_1 = a_2 = \frac{1}{2}, \quad b_2 = 1$$

e il corrispondente metodo è attribuito a *Heun*:

$$(8.23) \quad \begin{cases} y_{n+1} = y_n + \frac{h}{2}(k_1 + k_2) \\ k_1 = f(x_n, y_n) \\ k_2 = f(x_n + h, y_n + hk_1) \end{cases}$$

Un'altra soluzione è

$$a_1 = 0, \quad a_2 = 1, \quad b_2 = \frac{1}{2}$$

e il metodo cui dà origine è quello di *Eulero modificato*:

$$(8.24) \quad \begin{cases} y_{n+1} = y_n + hk_2 \\ k_1 = f(x_n, y_n) \\ k_2 = f\left(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1\right) \end{cases}$$

Entrambi i metodi richiedono ad ogni passo 2 valutazioni della funzione  $f$ .

È possibile verificare che quando le (8.22) sono soddisfatte, il coefficiente di  $h^2$  nello sviluppo di  $t(x^*, y^*; h)$  in serie di potenze di  $h$  è diverso da zero, per cui l'ordine massimo raggiungibile con un metodo di tipo (8.21) è  $p = 2$ . Osserviamo inoltre che di metodi (8.21) con coefficienti  $a_1, a_2$  e  $b_2$  soluzione del sistema (8.22), ovvero con ordine  $p = 2$ , ne esistono infiniti.

Per costruire metodi di ordine superiore è sufficiente generalizzare quanto fatto nella (8.21) e definire il metodo  $\Phi$  nel modo seguente:

$$\begin{cases} \Phi(x^*, y^*; h) = \sum_{i=1}^r a_i k_i \\ k_1 = f(x^*, y^*) \\ k_i = f\left(x^* + b_i h, y^* + h \sum_{j=1}^{i-1} c_{ij} k_j\right), \quad i = 2, \dots, r \end{cases}$$

onde

$$(8.25) \quad \begin{cases} y_{n+1} = y_n + h \sum_{i=1}^r a_i k_i \\ k_1 = f(x_n, y_n) \\ k_i = f\left(x_n + b_i h, y_n + h \sum_{j=1}^{i-1} c_{ij} k_j\right), \quad i = 2, \dots, r \end{cases}$$

Quest'ultimo è certamente consistente se

$$\sum_{i=1}^r a_i = 1$$

Di solito viene anche imposta la condizione

$$\sum_{j=1}^{i-1} c_{ij} = b_i, \quad i = 2, \dots, r$$

La formula (8.25) rappresenta il generico *metodo Runge-Kutta esplicito a r stadi*, che per convenzione identificheremo con la tabella

	0				
$b_2$	$c_{21}$				
$b_3$	$c_{31}$	$c_{32}$			
$b_4$	$c_{41}$	$c_{42}$	$c_{43}$		
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	
$b_r$	$c_{r1}$	$c_{r2}$	$c_{r3}$	$\dots$	$c_{r,r-1}$
	$a_1$	$a_2$	$a_3$	$\dots$	$a_{r-1} \quad a_r$

Fissato il numero di stadi  $r \geq 1$ , i parametri  $b_i$ ,  $c_{ij}$  e  $a_i$  devono essere determinati in modo che lo sviluppo di  $t(x^*, y^*; h)$  in serie di potenze di  $h$  inizi con il termine di ordine il più elevato possibile. Denotiamo con  $p = p^*(r)$  tale ordine massimo; è stato dimostrato (vedi [8.21, pag. 184]) che

$$\begin{aligned} p^*(r) &= r, & r &= 1, 2, 3, 4 \\ p^*(r) &= r - 1, & r &= 5, 6, 7 \\ p^*(r) &= r - 2, & r &= 8, 9 \\ p^*(r) &\leq r - 3, & r &\geq 10 \end{aligned}$$

La costruzione di metodi Runge-Kutta di ordine superiore a 3, soprattutto nel caso di sistemi di equazioni differenziali<sup>(†)</sup>, è assai laboriosa e richiede la risoluzione numerica

---

(†) Vedi ad esempio [8.21].

di sistemi di equazioni non lineari con un numero di incognite superiore al numero di equazioni, per cui avremo famiglie di metodi dipendenti da uno o più parametri. Il più noto di questi è il seguente metodo del 4° ordine:

$$(8.26) \quad \begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{1}{2} & 0 & \frac{1}{2} & & \\ 1 & 0 & 0 & 1 & \\ \hline \frac{1}{6} & \frac{2}{6} & \frac{2}{6} & \frac{1}{6} & \end{array} \quad \left\{ \begin{array}{l} y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ k_1 = f(x_n, y_n) \\ k_2 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_1) \\ k_3 = f(x_n + \frac{h}{2}, y_n + \frac{h}{2}k_2) \\ k_4 = f(x_n + h, y_n + hk_3) \end{array} \right.$$

Nei due esempi che seguono utilizziamo i metodi (8.23) e (8.26) con passo  $h = 2^{-k}$ ,  $k = 1, 2, \dots, 9$ , e riportiamo i corrispondenti errori  $|y(x_N) - y_N|$  prodotti nel punto  $x_N = 1$ .

► **Esempio 8.1.**

$$\begin{cases} y'(x) = y(x) \\ y(0) = 1 \end{cases} \quad y(x) = e^x$$

$k$	(8.23)	(8.26)
1	$7.77 \cdot 10^{-2}$	$9.36 \cdot 10^{-4}$
2	$2.34 \cdot 10^{-2}$	$7.19 \cdot 10^{-5}$
3	$6.44 \cdot 10^{-3}$	$4.98 \cdot 10^{-6}$
4	$1.69 \cdot 10^{-3}$	$3.28 \cdot 10^{-7}$
5	$4.32 \cdot 10^{-4}$	$2.10 \cdot 10^{-9}$
6	$1.09 \cdot 10^{-4}$	$1.33 \cdot 10^{-9}$
7	$2.75 \cdot 10^{-5}$	$8.38 \cdot 10^{-11}$
8	$6.89 \cdot 10^{-6}$	$5.26 \cdot 10^{-12}$
9	$1.73 \cdot 10^{-6}$	$3.29 \cdot 10^{-13}$

Tabella 8.1

► **Esempio 8.2.**

$$\begin{cases} y'(x) = -(1-x)^{5/2}y(x) \\ y(0) = 1 \end{cases} \quad y(x) = e^{\frac{2}{7}[(1-x)^{7/2}-1]}$$

$k$	(8.23)	(8.26)
1	$5.57 \cdot 10^{-2}$	$1.18 \cdot 10^{-4}$
2	$1.13 \cdot 10^{-2}$	$1.70 \cdot 10^{-6}$
3	$2.51 \cdot 10^{-3}$	$6.66 \cdot 10^{-7}$
4	$5.89 \cdot 10^{-4}$	$7.53 \cdot 10^{-8}$
5	$1.43 \cdot 10^{-4}$	$7.15 \cdot 10^{-9}$
6	$3.52 \cdot 10^{-5}$	$6.48 \cdot 10^{-10}$
7	$8.72 \cdot 10^{-6}$	$5.78 \cdot 10^{-11}$
8	$2.17 \cdot 10^{-6}$	$5.12 \cdot 10^{-12}$
9	$5.42 \cdot 10^{-7}$	$4.53 \cdot 10^{-13}$

Tabella 8.2



### 8.2.3 Convergenza dei metodi one-step esplicativi

Finora ci siamo limitati ad esaminare il comportamento locale dei metodi one-step esplicativi, ovvero l'errore introdotto dal metodo durante un singolo passo di integrazione, supponendo che nel valore “iniziale”  $y^*$  non siano presenti perturbazioni. Tuttavia, ciò che a noi interessa è il comportamento globale del metodo, ossia l'errore globale (di troncamento)  $y(x_n) - y_n$  prodotto dal metodo<sup>(†)</sup> dopo  $n$  passi successivi di integrazione. Tale errore è generato dagli  $n$  errori locali di troncamento introdotti negli  $n$  passi effettuati per avanzare da  $x_0$  a  $x_n$ .

**Definizione 8.5.** Il metodo (8.16) è detto convergente in  $[a, b]$  se, qualunque sia il problema (8.15), con  $f \in F_1(S)$ , per ogni  $x \in [a, b]$  e  $h_n = h = (x - a)/N$  risulta

$$\lim_{N \rightarrow \infty} y_N = y(x)$$

Ovviamente, perché un metodo possa essere preso in considerazione deve risultare convergente, in modo che per  $h$  sufficientemente piccolo la quantità  $\|y(x) - y_N\|$  possa essere resa, in aritmetica con precisione infinita, inferiore a qualsiasi tolleranza prestabilita.

Il teorema che segue, la cui dimostrazione può essere reperita in [8.2, pag. 124], caratterizza le funzioni  $\Phi$  che danno origine a metodi (8.16) convergenti.

**Teorema 8.3.** Sia  $\Phi(x^*, y^*; h)$  continua nel dominio  $D = S \times [0, h_0]$ ,  $h_0 > 0$ , e (uniformemente) lipschitziana nella variabile  $y^*$  in  $D$ . Allora la consistenza è condizione necessaria e sufficiente per la convergenza del corrispondente metodo (8.16). Inoltre, se l'ordine del metodo è  $p$  e in (8.15)  $f \in F_p(S)$ ,

$$\|y(x_n) - y_n\| \leq K h^p, h \leq h_0$$

(†) Supponendo per semplicità di operare con precisione di calcolo infinita.

dove  $K$  è una costante indipendente da  $n$  e  $h$ .

**Corollario 8.1.** Tutti i metodi one-step presentati nel paragrafo 8.2.2 sono convergenti.

Anzi, è possibile dimostrare (vedi [8.16, pag. 163]) che essi risultano convergenti anche quando il passo  $h_n$  non è mantenuto costante:  $a = x_0 < x_1 < \dots < x_{N-1} < x_N = x$ ,  $h = \max_n(x_{n+1} - x_n)$ . In questo caso abbiamo

$$\lim_{h \rightarrow 0} y_N = y(x)$$

#### 8.2.4 Stima dell'errore locale di troncamento e scelta del passo di integrazione

Consideriamo un metodo Runge-Kutta  $\Phi$  di ordine  $p \geq 1$ ; definiremo *accettabile* una stima  $r(x^*, y^*; h)$  dell'errore locale di troncamento  $t(x^*, y^*; h)$  se

$$r(x^*, y^*; h) = t(x^*, y^*; h) + O(h^{p+1}), \quad h \rightarrow 0$$

Le tecniche proposte per determinare stime accettabili sono sostanzialmente due. Una, denominata *estrapolazione di Richardson* (vedi ad esempio [8.19]), utilizza l'espressione

$$t(x^*, y^*; h) = Ch^p + O(h^{p+1})$$

valida quando  $f \in F_{p+1}(S)$ , per costruire la stima

$$r(x^*, y^*; h) = \frac{1}{h} \frac{y_h^{(2)} - y_h}{2^p - 1}$$

dove  $y_h$  e  $y_h^{(2)}$  rappresentano le approssimazioni di  $u(x^* + h)$  ottenute applicando il metodo  $\Phi$  rispettivamente una volta con passo  $h$  e due volte successive con passo  $h/2$ :



Figura 8.9

Con l'altra tecnica invece il valore  $y_h$  ottenuto con il metodo  $\Phi$  viene confrontato con l'approssimazione  $y_h^{(1)} = y^* + h\Phi^{(1)}(x^*, y^*; h)$  fornita da un secondo metodo  $\Phi^{(1)}$  di ordine  $p^{(1)} = p + 1$ :

$$r(x^*, y^*; h) = \frac{y_h^{(1)} - y_h}{h} = \Phi^{(1)}(x^*, y^*; h) - \Phi(x^*, y^*; h)$$

I metodi  $\Phi$  (di ordine  $p$  e a  $r$  stadi) e  $\Phi^{(1)}$  (di ordine  $p+1$  e a  $r^{(1)}$  stadi) vengono costruiti imponendo al metodo  $\Phi^{(1)}$  di riutilizzare tutti i valori  $k_i$ ,  $i = 1, \dots, r$ , presenti in  $\Phi$ , e

al tempo stesso cercando di ridurre al minimo il numero totale di stadi distinti  $r^{(1)}$ . La coppia di metodi  $\Phi$  e  $\Phi^{(1)}$  assumerà pertanto la forma

$$\begin{cases} y_h = y^* + h \sum_{i=1}^r a_i k_i \\ k_1 = f(x^*, y^*) \\ k_i = f\left(x^* + b_i h, y^* + h \sum_{j=1}^{i-1} c_{ij} k_j\right), \quad i = 2, \dots, r \end{cases}$$

$$\begin{cases} y_h^{(1)} = y^* + h \sum_{i=1}^{r^{(1)}} a_i^{(1)} k_i, \quad r^{(1)} > r \\ k_1 = f(x^*, y^*) \\ k_i = f\left(x^* + b_i h, y^* + h \sum_{j=1}^{i-1} c_{ij} k_j\right), \quad i = 2, \dots, r^{(1)} \end{cases}$$

Coppie di questo tipo sono state costruite, per diversi valori di  $p$ , da vari autori (vedere ad esempio, [8.21, pag. 37] e [8.16, §II.4]). Qui di seguito ne riportiamo una, attribuita a Fehlberg, relativa al caso  $p = 4$  ( $r = 5, r^{(1)} = 6$ ):

0					
$\frac{1}{4}$	$\frac{1}{4}$				
$\frac{3}{8}$	$\frac{3}{32}$	$\frac{9}{32}$			
$\frac{12}{13}$	$\frac{1932}{2197}$	$-\frac{7200}{2197}$	$\frac{7296}{2197}$		
1	$\frac{439}{216}$	-8	$\frac{3680}{513}$	$-\frac{845}{4104}$	
$\frac{1}{2}$	$-\frac{8}{27}$	2	$-\frac{3544}{2565}$	$\frac{1859}{4104}$	$-\frac{11}{40}$
$a_i$	$\frac{25}{216}$	0	$\frac{1408}{2565}$	$\frac{2197}{4104}$	$-\frac{1}{5}$
$a_i^{(1)}$	$\frac{16}{135}$	0	$\frac{6656}{12825}$	$\frac{28561}{56430}$	$-\frac{9}{50}$
					$\frac{2}{55}$

Tabella 8.3

Nella risoluzione approssimata di un problema a valori iniziali siamo ovviamente interessati al comportamento dell'errore globale, e cioè alla differenza tra la soluzione esatta del problema  $\{y(x_n)\}$  e quella approssimata  $\{y_n\}$  fornita dal metodo scelto. Inoltre, non solo siamo interessati a stimare l'errore globale  $y(x_n) - y_n$ , ma sarebbe nostro desiderio poter controllare quest'ultimo, ossia poter scegliere i successivi passi  $h = h_n$  in modo da mantenere in ogni punto  $x_n$  l'errore globale al di sotto di una tolleranza prestabilita.

Una semplice riflessione sul significato di errore globale ci farà capire come in genere non sia possibile “controllare” tale errore. Scriviamo

$$(8.27) \quad y(x_{n+1}) - y_{n+1} = [y(x_{n+1}) - u(x_{n+1})] + [u(x_{n+1}) - y_{n+1}]$$

dove  $u(x)$  denota la soluzione dell’equazione differenziale  $v'(x) = f(x, v(x))$  uscente dal punto  $x_n, y_n$ . La seconda parentesi quadra in (8.27) rappresenta l’errore locale di troncamento relativo al passo  $x_n \rightarrow x_{n+1}$ ; il termine  $y(x_{n+1}) - u(x_{n+1})$  invece è legato al condizionamento del problema (8.15).

Come abbiamo visto a pagina 258, responsabile dell’amplificazione in  $x_{n+1}$  dell’errore globale al passo precedente,  $y(x_n) - y_n$ , è, in prima approssimazione, lo Jacobiano  $f_y(x_n, y_n)$ . Se il problema è mal condizionato, per rendere accettabile l’amplificazione (8.27) occorre prendere  $h_n$  molto piccolo, con il rischio che il processo di integrazione non avanzi più. Oppure, volendo ridurre gli errori globali nei nodi precedenti, si ricomincia da capo il processo di integrazione richiedendo una tolleranza più stringente; ma quale, perché il processo possa giungere a conclusione senza arrestarsi nuovamente prima di raggiungere il punto finale  $b$ ?

Per le difficoltà suddette, nelle routine automatiche viene controllato il solo errore locale di troncamento (al più potremmo pensare di stimare l’errore globale nei punti  $\{x_n\}$  scelti dal meccanismo di controllo dell’errore locale). Vediamo come ciò sia realizzabile, per esempio nel caso in cui la stima  $r(x_n, y_n; h_n)$  dell’errore locale  $t(x_n, y_n; h_n)$  sia ottenuta dal confronto di due metodi  $\Phi(x_n, y_n; h_n)$  e  $\Phi^{(1)}(x_n, y_n; h_n)$ , di ordine rispettivamente  $p$  e  $p+1$ .

Giunti al punto  $x = x_n$ , il nostro obiettivo diventa la scelta di un nuovo passo  $h_n, x_n + h_n = x_{n+1}$ , il più ampio possibile ma tale da rendere soddisfatta la disuguaglianza

$$(8.28) \quad h_n \|r(x_n, y_n; h_n)\| \leq \varepsilon$$

dove  $\varepsilon$  denota la precisione locale (non unitaria) richiesta. A tale fine occorre supporre<sup>(†)</sup> che per il metodo  $\Phi$  scelto valga la rappresentazione

$$t(x^*, y^*; h) = \tau(x^*, y^*)h^p + O(h^{p+1}), h \rightarrow 0$$

Questa relazione ci permette di separare (in prima approssimazione) le variabili  $x^*, y^*$  dal passo  $h$ ; ciò significa che la conoscenza del “coefficiente”  $\tau(x_n, y_n)$  ci consente di valutare l’errore locale di troncamento nel generico punto  $x_n + h$  (purché  $h$  non sia troppo grande altrimenti il termine  $O(h^{p+1})$  può non risultare trascurabile).

Pertanto, dopo aver applicato una prima volta la coppia  $(\Phi, \Phi^{(1)})$  con un passo di tentativo  $h$ , per esempio  $h = h_{n-1}$ , e avere quindi determinato la stima

$$r(x_n, y_n; h) = \Phi^{(1)}(x_n, y_n; h) - \Phi(x_n, y_n; h)$$

dalla relazione

$$\|r(x_n, y_n; h)\| \cong \|\tau(x_n, y_n)\| h^p$$

---

<sup>(†)</sup> Questo è senz’altro vero per i metodi Runge-Kutta di ordine  $p \geq 1$ , purché  $f \in F_{p+1}(S)$ .

deduciamo

$$\|\tau(x_n, y_n)\| \cong h^{-p} \|r(x_n, y_n; h)\|$$

A questo punto è sufficiente imporre la condizione (8.28), ovvero

$$h_n \|r(x_n, y_n; h_n)\| = h_n^{p+1} \|\tau(x_n, y_n)\| = \left(\frac{h_n}{h}\right)^{p+1} h \|r(x_n, y_n; h)\| = \theta \varepsilon$$

dove  $0 < \theta < 1$  è un fattore di sicurezza, per ottenere

$$h_n = h^{p+1} \sqrt{\frac{\theta \varepsilon}{h \|r(x_n, y_n; h)\|}}$$

Ovviamente, causa i termini trascurati, non è detto che con il passo  $h_n$  così scelto si ottenga la disegualanza (8.28). Conviene applicare nuovamente la coppia  $(\Phi, \Phi^{(1)})$ , questa volta, con  $h = h_n$ , e controllare se la precisione richiesta è stata “effettivamente” raggiunta.

Le relazioni precedenti sono solo approssimate e valgono nell’ipotesi di una sufficiente regolarità del problema  $f(x, y)$  e per  $h \rightarrow 0$ . In assenza dei requisiti richiesti si possono ottenere indicazioni false sull’attendibilità dei risultati. Di solito viene prefissato un intervallo  $(h_{\min}, h_{\max})$  entro il quale è consentita la scelta del passo  $h_n$ ; ovvero, se  $h_n < h_{\min}$  o  $h_n > h_{\max}$  si pone rispettivamente  $h_n = h_{\min}$  e  $h_n = h_{\max}$ .

La costruzione di una routine efficiente ed affidabile, con scelta automatica del passo e controllo dell’errore locale di troncamento, è ben più complessa di quanto possa apparire dalla breve descrizione che abbiamo inteso fornire in questo paragrafo.

### 8.3 Metodi multistep lineari

Sia dato il problema a valori iniziali

$$(8.29) \quad \begin{cases} y'(x) = f(x, y(x)), & a \leq x \leq b \\ y(a) = y_0 \end{cases}$$

Il generico metodo multistep lineare a  $k$  passi ha la forma

$$(8.30) \quad \alpha_0 y_{n+1} + \alpha_1 y_n + \cdots + \alpha_k y_{n+1-k} = h[\beta_0 f_{n+1} + \beta_1 f_n + \cdots + \beta_k f_{n+1-k}]$$

$$n = k - 1, k, \dots$$

dove  $x_m = a + mh$ ,  $f_m = f(x_m, y_m)$  ed in cui supponiamo di aver già calcolato (sempre con la (8.30)) i valori precedenti  $y_n, y_{n-1}, \dots, y_{n+1-k}$  e di voler determinare l’approssimazione  $y_{n+1}$  di  $y(x_{n+1})$ . Ovviamente, per “innescare” la relazione ricorsiva (8.30) occorre conoscere i  $k$  valori iniziali  $y_0, y_1, \dots, y_{k-1}$ . Poiché il problema (8.29) ci fornisce solamente  $y_0$ , i restanti  $k - 1$  valori devono essere determinati con metodi numerici, per esempio

con metodi Runge-Kutta oppure con formule di tipo (8.30) ma che utilizzano solamente i valori  $y_i$  che di volta in volta si rendono disponibili (vedere a pag. 279).

Contrariamente a quanto succedeva nel caso dei metodi Runge-Kutta, il costo della (8.30) è indipendente dal numero di passi  $k$  ed è sempre pari ad una valutazione della  $f(x, y)$  per passo. Osserviamo inoltre che il metodo risulta *implicito* se  $\beta_0 \neq 0$ , in quanto l'incognita  $y_{n+1}$  è presente anche nel termine  $f_{n+1} = f(x_{n+1}, y_{n+1})$ , ed *esplicito* se  $\beta_0 = 0$ .

Nei paragrafi che seguono esamineremo la risoluzione numerica del problema (8.29) mediante l'uso di metodi di tipo (8.30). Ricordiamo che la (8.30) è un'*equazione alle differenze di ordine k*, non lineare se tale è  $f(x, y)$ . Dal punto di vista teorico queste equazioni non sono certamente più facili da studiare<sup>(†)</sup> delle corrispondenti equazioni differenziali; tuttavia esse permettono di determinare numericamente le approssimazioni  $\{y_i\}$ , noti i valori iniziali  $y_0, y_1, \dots, y_{k-1}$  necessari per poter innescare la relazione di ricorrenza (8.30).

### 8.3.1 Comportamento locale dei metodi multistep lineari

Denotiamo con  $u(x)$  la soluzione dell'equazione differenziale  $v'(x) = f(x, v(x))$  che passa per il punto  $(x^*, y^*)$ , ovvero

$$\begin{cases} u'(x) = f(x, u(x)) \\ u(x^*) = y^* \end{cases}$$

Come errore locale unitario di troncamento del metodo multistep (8.30) nel punto  $x^* + h$  definiamo la quantità

$$t(x^*, y^*; h) = \frac{1}{h} \sum_{i=0}^k \alpha_i u(x^* + (1-i)h) - \sum_{i=0}^k \beta_i u'(x^* + (1-i)h)$$

onde

$$\sum_{i=0}^k \alpha_i u(x^* + (1-i)h) - h \sum_{i=0}^k \beta_i u'(x^* + (1-i)h) = ht(x^*, y^*; h)$$

La quantità  $ht(x^*, y^*; h)$  rappresenta il residuo generato dal metodo (8.30) quando in quest'ultimo introduciamo la funzione  $u(x)$ , ossia sostituiamo  $y_{n+1-i}$  con  $u(x^* + (1-i)h)$ .

**Definizione 8.6.** Il metodo multistep (8.30) è consistente se per ogni  $(x^*, y^*) \in S$  e qualunque  $f \in F_1(S)$  risulta

$$\lim_{h \rightarrow 0} t(x^*, y^*; h) = 0$$

**Definizione 8.7.** Il metodo multistep (8.30) ha ordine (di consistenza)  $p$ , intero positivo, se per tutti i punti  $(x^*, y^*) \in S$  e qualunque  $f \in F_p(S)$  risulta

$$(8.31) \quad t(x^*, y^*; h) = O(h^p), \quad h \rightarrow 0$$

e  $p$  è l'intero più grande per cui vale la (8.31).

---

<sup>(†)</sup> Vedi [8.2, pag. 210], [8.16].

### 8.3.2 Metodi multistep di Adams

Una classe, assai vasta ed importante, di metodi multistep espliciti ed impliciti può essere costruita partendo dall'identità

$$(8.32) \quad u(x^* + h) = u(x^* - lh) + \int_{x^* - lh}^{x^* + h} u'(x) dx, \quad l \text{ (intero)} \geq 0$$

e approssimando l'integrale a secondo membro mediante formule di quadratura di tipo interpolatorio con nodi  $\{x_{-i}\}$ ,  $x_{-i} = x^* + (1-i)h$ ,  $i \geq 0$ , consecutivi e non necessariamente tutti appartenenti all'intervallo di integrazione  $[x^* - lh, x^* + h]$ . In questa classe, particolare rilievo assumono le formule di Adams, ottenibile dalla (8.32) con il procedimento predetto, dopo aver scelto  $l = 0$ .

Pertanto, riscritta la (8.32) nella forma

$$(8.33) \quad u(x^* + h) = u(x^*) + \int_{x^*}^{x^* + h} u'(x) dx$$

interpoliamo dapprima la funzione integranda  $u'(x)$  (utilizzando la formula di Lagrange (5.12)) nei nodi  $x_{-1} \equiv x^*, x_{-2}, \dots, x_{-k}$ :

$$(8.34) \quad u'(x) = \sum_{i=1}^k l_i(x) u'(x_{-i}) + E_k(x)$$

con

$$l_i(x) = \prod_{\substack{j=1 \\ j \neq i}}^k \frac{x - x_{-j}}{x_{-i} - x_{-j}}, \quad i = 1, \dots, k$$

e, se  $u(x) \in C^{k+1}[a, b]$ ,

$$E_k(x) = \prod_{j=1}^k (x - x_{-j}) \frac{u^{(k+1)}(\xi_x)}{k!}, \quad x_{-k} < \xi_x < x_{-1}$$

Introducendo una nuova variabile  $s$ ,  $s = (x_0 - x)/h$ , ovvero ponendo  $x = x_0 - sh$  e  $x_{-j} = x_0 - jh$ , abbiamo

$$x - x_{-j} = (x_0 - sh) - (x_0 - jh) = (j - s)h$$

donde

$$\int_{x^*}^{x^* + h} l_i(x) dx = h \int_0^1 \prod_{\substack{j=1 \\ j \neq i}}^k \frac{j - s}{j - i} ds$$

Infine, dopo aver introdotto le quantità

$$\beta_i = \int_0^1 \prod_{\substack{j=1 \\ j \neq i}}^k \frac{j - s}{j - i} ds, \quad i = 1, 2, \dots, k$$

sostituendo la (8.34) nella (8.33) otteniamo l'espressione

$$(8.35) \quad u(x^* + h) = y^* + h \sum_{i=1}^k \beta_i u'(x^* + (1-i)h) + ht(x^*, y^*; h), \quad y^* = u(x^*)$$

con

$$t(x^*, y^*; h) = \frac{1}{h} \int_{x^*}^{x^*+h} E_k(x) dx = \frac{h^k}{k!} \int_0^1 \prod_{j=1}^k (j-s) u^{(k+1)}(\eta_s) ds = O(h^k)$$

Il metodo multistep che deduciamo dalla (8.35)

$$(8.36) \quad y_{n+1} = y_n + h \sum_{i=1}^k \beta_i f(x_{n+1-i}, y_{n+1-i})$$

è quello esplicito di *Adams-Basforth*; esso è a  $k$  passi e di ordine  $k$ . Nella tabella che segue riportiamo i coefficienti  $\{\beta_i\}$  delle formule (8.36) corrispondenti ai valori  $k = 1, 2, 3, 4, 5$ .

$k$	$\beta_1$	$\beta_2$	$\beta_3$	$\beta_4$	$\beta_5$
1	1				
2	$\frac{3}{2}$	$-\frac{1}{2}$			
3	$\frac{23}{12}$	$-\frac{16}{12}$	$\frac{5}{12}$		
4	$\frac{55}{24}$	$-\frac{59}{24}$	$\frac{37}{24}$	$-\frac{9}{24}$	
5	$\frac{1901}{720}$	$-\frac{2774}{720}$	$\frac{2616}{720}$	$-\frac{1274}{720}$	$\frac{251}{720}$

Tabella 8.4

Dal punto di vista dell'implementazione del metodo, soprattutto se si desidera stimare l'errore locale di troncamento ed avere la possibilità di scegliere l'ordine  $k$  più conveniente, conviene interpolare  $u'(x)$  in (8.33) con il polinomio di Newton alle differenze regressive; in questo caso il metodo di Adams-Basforth (8.36) assume la forma

$$(8.37) \quad y_{n+1} = y_n + h \sum_{i=0}^{k-1} \gamma_i \nabla^i f(x_n, y_n)$$

dove i coefficienti

$$\gamma_i = \int_0^1 \binom{s+i-1}{i} ds, \quad i = 0, 1, \dots$$

$k$	$\beta_0^*$	$\beta_1^*$	$\beta_2^*$	$\beta_3^*$	$\beta_4^*$
1	$\frac{1}{2}$	$\frac{1}{2}$			
2	$\frac{5}{12}$	$\frac{8}{12}$	$-\frac{1}{12}$		
3	$\frac{9}{24}$	$\frac{19}{24}$	$-\frac{5}{24}$	$\frac{1}{24}$	
4	$\frac{251}{720}$	$\frac{646}{720}$	$-\frac{264}{720}$	$\frac{106}{720}$	$-\frac{19}{720}$

Tabella 8.5

soddisfano la relazione di ricorrenza

$$\begin{cases} \gamma_0 = 1 \\ \gamma_i = 1 - \sum_{j=1}^i \frac{1}{j+1} \gamma_{i-j}, \quad i = 1, 2, \dots \end{cases}$$

Inoltre, quando  $u(x) \in C^{k+2}[a, b]$  abbiamo

$$(8.38) \quad t(x_n, y_n; h) = \gamma_k h^k u^{(k+1)}(x_n) + O(h^{k+1})$$

Osserviamo che la formula di Adams-Bashfort di ordine  $k+1$  può essere ottenuta dalla (8.37) semplicemente estendendo la sommatoria sino a  $k$ , ovvero aggiungendo il termine

$$\begin{aligned} \gamma_k \nabla^k f(x_n, y_n) &= \gamma_k \nabla^k f(x_n, u(x_n)) = \gamma_k \nabla^k u'(x_n) \\ &= \gamma_k h^k u^{(k+1)}(\xi), \quad x_{n+1-k} < \xi < x_n \equiv x^* \end{aligned}$$

Ma quest'ultimo può essere riscritto nella forma

$$\gamma_k \nabla^k f(x_n, y_n) = \gamma_k h^k u^{(k+1)}(x_n) + O(h^{k+1}) = t(x_n, y_n; h) + O(h^{k+1})$$

quindi, non solo esso ci consente di passare dalla formula di ordine  $k$  a quella di ordine  $k+1$ , ma rappresenta anche una stima accettabile dell'errore locale (unitario) di troncamento nel punto  $x = x_{n+1}$ .

Se nella (8.33) interpoliamo la funzione integranda  $u'(x)$  con il polinomio di Lagrange, o di Newton alle differenze finite regressive, costruito sui nodi  $x_0, x_{-1}, \dots, x_{-k}$  otteniamo la formula implicita di *Adams-Moulton* a  $k$  passi di ordine  $k+1$ , che possiamo esprimere in una delle due forme seguenti:

$$(8.39) \quad y_{n+1} = y_n + h \sum_{i=0}^k \beta_i^* f(x_{n+1-i}, y_{n+1-i})$$

$$(8.40) \quad y_{n+1} = y_n + h \sum_{i=0}^k \gamma_i^* \nabla^i f(x_{n+1}, y_{n+1})$$

Nella tabella 8.5 riportiamo i coefficienti  $\{\beta_i^*\}$  relativi alle formule (8.39) con  $k = 1, 2, 3, 4$ . I coefficienti  $\{\gamma_i^*\}$  della (8.40) invece soddisfano la seguente relazione:

$$\begin{cases} \gamma_0^* = 1 \\ \gamma_i^* = -\sum_{j=1}^i \frac{1}{j+1} \gamma_{i-j}^*, \quad i = 1, 2, \dots \end{cases}$$

Per quanto riguarda l'errore locale unitario di troncamento nel punto  $x = x_{n+1}$  abbiamo le rappresentazioni

$$(8.41) \quad \begin{aligned} t(x_n, y_n; h) &= \gamma_{k+1}^* h^{k+1} u^{(k+2)}(x_n) + O(h^{k+2}) \\ &= \gamma_{k+1}^* \nabla^{k+1} f(x_{n+1}, y_{n+1}) + O(h^{k+2}) \end{aligned}$$

valide quando  $u(x) \in C^{k+3}[a, b]$ .

Osserviamo infine che la formula di Adams-Bashforth con  $k = 1$  coincide con il classico metodo di Eulero descritto a pagina 262, la formula di Adams-Moulton con  $k = 0$

$$y_{n+1} = y_n + h f(x_{n+1}, y_{n+1})$$

rappresenta il *metodo di Eulero implicito*, mentre la formula di Adams-Moulton con  $k = 1$

$$y_{n+1} = y_n + \frac{h}{2} [f(x_{n+1}, y_{n+1}) + f(x_n, y_n)]$$

è denominata *metodo dei trapezi*.

Queste tre formule in realtà rappresentano tutte dei metodi one-step, di tipo esplicito la prima e implicito le altre due.

Se in (8.32) prendiamo  $l = 1$  e interpoliamo  $u'(x)$  solamente nel punto  $x = x_n$ , otteniamo la *formula del punto medio*

$$y_{n+1} = y_{n-1} + 2h f(x_n, y_n)$$

### 8.3.3 Convergenza dei metodi multistep

Consideriamo il generico metodo multistep lineare a  $k$  passi

$$\sum_{i=0}^k \alpha_i y_{n+1-i} = h \sum_{i=0}^k \beta_i f(x_{n+1-i}, y_{n+1-i})$$

Esso potrà essere preso in considerazione per “risolvere” il problema (8.29) solo se risulterà *convergente*, ovvero se, qualunque sia la funzione  $f \in F_1(S)$ ,

$$\lim_{\substack{h \rightarrow 0 \\ a+Nh=x}} y_N = y(x)$$

per ogni  $x \in [a, b]$  e qualunque siano i valori iniziali  $y_0, y_1, \dots, y_{k-1}$ , purché  $y_i \rightarrow y(a)$  per  $h \rightarrow 0$ ,  $i = 0, 1, \dots, k-1$ .

Il teorema successivo riduce lo studio della convergenza di un metodo multistep lineare consistente all'esame delle radici della seguente *equazione caratteristica*:

$$(8.42) \quad \alpha(t) \equiv \sum_{i=0}^k \alpha_i t^{k-i} = 0$$

**Teorema 8.4.** (vedi [8.21, §8.20]) *Un metodo multistep lineare è convergente se e solo se:*

- (i) è consistente;
- (ii) le radici dell'equazione caratteristica (8.42) hanno tutte modulo  $\leq 1$ , e quelle di modulo unitario sono semplici<sup>(†)</sup>.

Inoltre, se il metodo ha ordine di consistenza  $p$  e i valori iniziali  $y_0, y_1, \dots, y_{k-1}$  sono affetti da un errore  $y(x_i) - y_i = O(h^p)$ ,  $h \rightarrow 0$ , abbiamo

$$y(x_N) - y_N = O(h^p), \quad h \rightarrow 0$$

Questa proprietà ci garantisce che, prendendo  $h$  sufficientemente piccolo e supponendo di operare con precisione di calcolo infinita, è possibile approssimare la soluzione del problema (8.29) in un punto  $x \in [a, b]$  con la precisione desiderata.

Osserviamo che per i metodi di Adams a  $k$  passi presentati nel precedente paragrafo abbiamo  $\alpha(t) = t^{k-1}(t-1)$ ; pertanto essi risultano certamente convergenti, e il comportamento dell'errore globale è  $O(h^k)$  per la formula di Adams-Bashforth e  $O(h^{k+1})$  per quella di Adams-Moulton.

### 8.3.4 Metodi previsore-correttore

L'uso di metodi di tipo esplicito è, ovviamente, meno oneroso dal punto di vista computazionale. Tuttavia, a parità di numero di passi,  $k$  per esempio, il metodo di Adams implicito ha ordine  $k+1$  mentre quello esplicito  $k$ ; inoltre, a parità di ordine ( $k$ ), il primo ha una costante d'errore  $\gamma_k^*$  in (8.41) più piccola della corrispondente  $\gamma_k$  in (8.38). L'interesse per i metodi impliciti è però destato soprattutto dalle loro migliori proprietà di stabilità assoluta (vedi paragrafo 8.4) che li fanno preferire a quelli espliciti nella risoluzione di problemi definiti *stiff* (vedi paragrafo 8.5).

Assunto pertanto che i metodi impliciti sono talvolta da preferirsi a quelli espliciti, vediamo come i primi possano essere implementati. In particolare consideriamo il metodo di Adams-Moulton di ordine  $k$ , che per semplicità scriviamo nella forma

$$(8.43) \quad y_{n+1} = h\beta_0^* f(x_{n+1}, y_{n+1}) + g_n$$

dove

$$g_n = y_n + h \sum_{i=1}^{k-1} \beta_i^* f(x_{n+1-i}, y_{n+1-i})$$

---

(†) Questa condizione è generalmente nota con il nome di zero-stabilità del metodo.

La (8.43) è un’equazione<sup>(†)</sup> non lineare nell’incognita  $y_{n+1}$ . Per determinare quest’ultima possiamo utilizzare i metodi del capitolo 6. Tra questi il metodo più semplice è quello iterativo descritto nel paragrafo 6.3.2:

$$(8.44) \quad y_{n+1}^{(j+1)} = h\beta_0^* f(x_{n+1}, y_{n+1}^{(j)}) + g_n, \quad j = 0, 1, \dots$$

Se  $h$  è sufficientemente piccolo, e comunque tale che

$$(8.45) \quad h|\beta_0^*|L < 1^{(\dagger\dagger)}$$

qualunque sia l’approssimazione iniziale  $y_{n+1}^{(0)}$  abbiamo (teorema 6.2)

$$\lim_{j \rightarrow \infty} y_{n+1}^{(j)} = y_{n+1}$$

Tuttavia, quando la funzione  $f(x, y)$  ha una costante  $L$  molto grande, per esempio dell’ordine di  $10^4 \div 10^{10}$ , per soddisfare la condizione (8.45) occorre prendere un passo  $h$  eccessivamente piccolo. In questi casi conviene utilizzare il metodo di Newton.

Per limitare il numero di iterazioni richiesto dal processo (8.44) o da quello di Newton per “conseguire” la convergenza (in pratica per raggiungere la situazione in cui due iterate successive differiscono di una quantità sufficientemente piccola), conviene determinare un’approssimazione iniziale  $y_{n+1}^{(0)}$  molto accurata; per esempio utilizzando il metodo di Adams-Bashforth a  $k$  passi (di ordine  $k$ ).

Una strategia alternativa per migliorare l’approssimazione  $y_{n+1}^{(0)}$  fornita dal metodo esplicito, che in questo frangente denominiamo *previsore*, consiste nell’iterare la (8.44) un numero prefissato di volte (di solito non più di 2), costante per tutti i passi. In questa situazione la formula (8.44) prende il nome di *correttore*. La coppia di formule previsore-correttore, utilizzata nel modo predetto, costituisce un nuovo metodo multistep esplicito, ma non più lineare. Ovviamente a questi metodi non possiamo applicare le definizioni e i teoremi dei paragrafi precedenti, propri dei metodi multistep lineari; esistono tuttavia dei risultati analoghi (vedi [8.19, pag. 103] e [8.21], [8.8]).

### 8.3.5 Metodi multistep a passo variabile

Nei metodi multistep che abbiamo esaminato il passo  $h$  è supposto costante, ovvero si assume che il processo di integrazione avanzi sempre con lo stesso passo  $h$ , indipendentemente dal comportamento locale della soluzione  $y(x)$ . La scelta di  $h$  costante ha come conseguenza la indipendenza dei coefficienti  $\alpha_i$  e  $\beta_i$  in (8.30) da  $n$  e  $h$ ; in particolare, nei metodi di Adams (8.36) e (8.39) i coefficienti  $\beta_i$  e  $\beta_i^*$  sono delle costanti reali proprie dei singoli nodi, indipendenti dai valori che gli stessi assumono. Inoltre, il teorema di convergenza 8.4 è applicabile solo nell’ipotesi di  $h$  costante. Tuttavia, come già è stato rilevato nel caso dei metodi Runge-Kutta, avanzare sempre con lo stesso passo  $h$  risulta

<sup>(†)</sup> O un sistema di equazioni.

<sup>(††)</sup> Con  $L$  denotiamo la costante di Lipschitz della funzione  $f(x, y)$  (vedi teorema 8.1).

poco efficiente, soprattutto quando il comportamento della soluzione è molto variabile nell’intervallo di interesse  $[a, b]$ . Nella costruzione di una routine automatica è importante invece avere la possibilità di variare il passo  $h$ .

Le tecniche finora più usate per consentire l’avanzamento con passo variabile sono sostanzialmente due. Con la prima, supposto di voler procedere con un metodo multistep, di tipo Adams a  $k$  passi, da  $x_n$  a  $x_{n+1} = x_n + h^*$ , avendo  $x_{n-i} = x_n - ih$ ,  $i = 1, \dots, k$ , e  $h \neq h^*$ , ci procuriamo i valori  $f(x_{n-i}^*, y_{n-i}^*)$ , dove  $x_{n-i}^* = x_n - ih^*$ ,  $i = 1, \dots, k-1$ , necessari per poter applicare la nostra formula con passo  $h^*$ , ricorrendo ad una interpolazione polinomiale sui dati noti  $f(x_{n-i}, y_{n-i})$ ,  $i = 0, 1, \dots, k$ . In questo caso, denotato con  $P_k(x)$  il polinomio di interpolazione di grado  $k$  individuato dalle condizioni  $P_k(x_{n-i}) = f(x_{n-i}, y_{n-i})$ ,  $i = 0, 1, \dots, k$  definiamo

$$f(x_{n-i}^*, y_{n-i}^*) = P_k(x_{n-i}^*), \quad i = 1, \dots, k-1$$

La seconda strategia consiste nell’utilizzare metodi multistep con nodi non equidistanti. In questo caso occorre però costruire una nuova formula, sempre a  $k$  passi, ad ogni “istante”, in quanto i coefficienti  $\beta_i$  risultano dipendenti dalla posizione dei nodi  $x_{n+1-i}$ ,  $i = 0, 1, \dots, k$ .

Nonostante il teorema di convergenza 8.4 non sia più applicabile, in entrambi i casi sono stati dimostrati teoremi analoghi; vedere ad esempio [8.8], pag. 100], [8.16], [8.19].

Segnaliamo infine la rappresentazione di Nordsieck dei metodi multistep (vedi ad esempio [8.19], [8.5], [15]), che consente la variazione del passo  $h$  in modo assai semplice.

I valori iniziali necessari per “innescare” il metodo multistep ( $a$   $k$  passi) scelto, con  $h$  costante o meno, potrebbero venire determinati con un metodo Runge-Kutta dello stesso ordine; di solito essi vengono però calcolati con formule multistep della stessa famiglia ma di ordine inferiore, ovvero tali da coinvolgere solamente le approssimazioni che sino a quel punto sono state determinate. Ovviamente in questa fase iniziale la precisione locale desiderata viene raggiunta prendendo  $h$  sufficientemente piccolo.

Le moderne routine automatiche non solo prevedono la variazione del passo  $h$ , ma anche quella dell’ordine; ad ogni avanzamento esse determinano l’accoppiata passo-ordine in un certo senso ottimale.

## 8.4 Stabilità dei metodi numerici

I teoremi di convergenza presentati nei paragrafi 8.2.3 e 8.3.3 ci assicurano che l’errore globale in un generico punto  $x \in [a, b]$ ,  $x \equiv x_N = a + Nh$ , tende a zero quando  $h \rightarrow 0$  (e  $N \rightarrow \infty$ ). Infatti, se consideriamo il problema

$$(8.46) \quad \begin{cases} y'(x) = \lambda y(x) \\ y(0) = 1 \end{cases}$$

con  $\lambda = 1$ , e utilizziamo, per esempio, il metodo di Heun (8.23) per determinare un’approssimazione di  $y(0.5) = e^{0.5}$ , otteniamo i seguenti risultati:

$h$	$y_N$	$ y(0.5) - y_N $
$0.1 \cdot 10^0$	$0.1647447 \cdot 10^1$	$1.27 \cdot 10^{-3}$
$0.5 \cdot 10^{-1}$	$0.1648390 \cdot 10^1$	$3.31 \cdot 10^{-4}$
$0.1 \cdot 10^{-1}$	$0.1648708 \cdot 10^1$	$1.36 \cdot 10^{-5}$
$0.1 \cdot 10^{-2}$	$0.1648721 \cdot 10^1$	$1.37 \cdot 10^{-7}$

Tabella 8.6

Tuttavia, non è escluso che in alcuni problemi per ottenere approssimazioni ragionevoli occorra scegliere un passo di integrazione  $h$  molto piccolo. Per esempio, se in (8.46) prendiamo  $\lambda = -10^4$  ( $y(0.5) = e^{-5000} \cong 0$ ) e ripetiamo l'esperimento, abbiamo

$h$	$y_N$
$0.1 \cdot 10^0$	$-0.1270216 \cdot 10^{16}$
$0.1 \cdot 10^{-1}$	$0.7783126 \cdot 10^{100}$
$0.1 \cdot 10^{-2}$	$\gg 10^{308}$
$0.2 \cdot 10^{-3}$	$0.1648639 \cdot 10^1$
$0.1 \cdot 10^{-3}$	0.0

Tabella 8.7

In questo secondo caso osserviamo con sorpresa che, almeno inizialmente, riducendo il passo  $h$  non si ha alcuna diminuzione dell'errore globale; anzi assistiamo addirittura ad un'esplosione di quest'ultimo. Per ottenere una riduzione dell'errore dobbiamo scendere al di sotto di una soglia  $h_0$ . Questo fenomeno non si manifesta invece quando in (8.46)  $\lambda = 1$ , o più in generale  $\lambda$  è positivo<sup>(†)</sup>.

Un esame più attento dei due casi  $\lambda = 1$  e  $\lambda = -10^4$  ci permette di rilevare che le due corrispondenti soluzioni  $y = e^x$  e  $y = e^{-10^4 x}$  hanno un comportamento diametralmente opposto: mentre la prima diverge quando  $x \rightarrow \infty$ , la seconda converge rapidamente a zero. Ma per ciascun valore di  $h$ , partendo dal punto iniziale  $x_0 = 0$  e avanzando con il metodo numerico scelto, il comportamento della successione  $\{y_n\}$  è simile a quello di  $y(x)$ ?

Ripetiamo questa volta l'esperimento mettendo a confronto i metodi di Heun ( $H$ ), di Adams-Bashforth di ordine 2 ( $AB$ ) e dei trapezi ( $T$ ) (tabelle 8.8 e 8.9).

Nel caso  $\lambda = -10^4$  affinché anche la successione fornita dalla formula  $AB$  assuma un comportamento corretto è sufficiente prendere un passo  $h$  più piccolo di quelli considerati nella tabella 8.9; per esempio  $h = 0.5 \cdot 10^{-4}$ .

Come era facilmente prevedibile, quando  $\lambda = 1$  le successioni  $\{y_n\}$  riproducono correttamente il comportamento di  $y(x) = e^x$  per ogni scelta di  $h$ , mentre con  $\lambda = -10^4$  ciò avviene solo per la formula dei trapezi. Per le formule di Heun e di Adams-Bashforth

(†) O negativo, ma piccolo in valore assoluto.

$\lambda = 1$							
		$ y(x_N) - y_N $					
		$h = 0.1$			$h = 0.1 \cdot 10^{-1}$		
$x_N$	$y(x_N)$	$H$	$AB$	$T$	$H$	$AB$	$T$
0.1	$1.1051710$	$1.71 \cdot 10^{-4}$	$4.13 \cdot 10^{-4}$	$9.22 \cdot 10^{-5}$	$1.83 \cdot 10^{-6}$	$4.57 \cdot 10^{-6}$	$9.21 \cdot 10^{-7}$
0.2	$1.2214028$	$3.78 \cdot 10^{-4}$	$9.31 \cdot 10^{-4}$	$2.04 \cdot 10^{-4}$	$4.04 \cdot 10^{-6}$	$1.01 \cdot 10^{-5}$	$2.04 \cdot 10^{-6}$
0.3	$1.3409588$	$6.26 \cdot 10^{-4}$	$1.55 \cdot 10^{-3}$	$3.38 \cdot 10^{-4}$	$6.70 \cdot 10^{-6}$	$1.68 \cdot 10^{-5}$	$3.37 \cdot 10^{-6}$
0.4	$1.4918248$	$9.23 \cdot 10^{-4}$	$2.30 \cdot 10^{-3}$	$4.98 \cdot 10^{-4}$	$9.87 \cdot 10^{-6}$	$2.47 \cdot 10^{-5}$	$4.97 \cdot 10^{-6}$
0.5	$1.6487213$	$1.27 \cdot 10^{-3}$	$3.18 \cdot 10^{-3}$	$6.88 \cdot 10^{-4}$	$1.36 \cdot 10^{-5}$	$3.41 \cdot 10^{-5}$	$6.87 \cdot 10^{-6}$
0.6	$1.8221188$	$1.69 \cdot 10^{-3}$	$4.22 \cdot 10^{-3}$	$9.13 \cdot 10^{-4}$	$1.81 \cdot 10^{-5}$	$4.53 \cdot 10^{-5}$	$9.11 \cdot 10^{-6}$
0.7	$2.0137527$	$2.18 \cdot 10^{-3}$	$5.45 \cdot 10^{-3}$	$1.18 \cdot 10^{-3}$	$2.33 \cdot 10^{-5}$	$5.84 \cdot 10^{-5}$	$1.17 \cdot 10^{-5}$
0.8	$2.2255409$	$2.75 \cdot 10^{-3}$	$6.89 \cdot 10^{-3}$	$1.49 \cdot 10^{-3}$	$2.95 \cdot 10^{-5}$	$7.37 \cdot 10^{-5}$	$1.48 \cdot 10^{-5}$
0.9	$2.4596031$	$3.42 \cdot 10^{-3}$	$8.57 \cdot 10^{-3}$	$1.85 \cdot 10^{-3}$	$3.66 \cdot 10^{-5}$	$9.17 \cdot 10^{-5}$	$1.84 \cdot 10^{-5}$
1.0	$2.7182820$	$4.20 \cdot 10^{-3}$	$1.05 \cdot 10^{-2}$	$2.27 \cdot 10^{-3}$	$4.50 \cdot 10^{-5}$	$1.13 \cdot 10^{-4}$	$2.27 \cdot 10^{-5}$

Tabella 8.8

$\lambda = -10^4$						
	$ y_N $					
	$h = 0.5 \cdot 10^{-2}$			$h = 0.1 \cdot 10^{-3}$		
$x_N$	$H$	$AB$	$T$	$H$	$AB$	$T$
0.1	$6.70 \cdot 10^{33}$	$2.64 \cdot 10^{37}$	$2.02 \cdot 10^{-1}$	0.0	$2.39 \cdot 10^{-1}$	0.0
0.2	$4.49 \cdot 10^{67}$	$7.02 \cdot 10^{74}$	$4.07 \cdot 10^{-2}$	0.0	$2.39 \cdot 10^{-1}$	0.0
0.3	$3.01 \cdot 10^{101}$	$1.86 \cdot 10^{112}$	$8.21 \cdot 10^{-3}$	0.0	$2.39 \cdot 10^{-1}$	0.0
0.4	$2.01 \cdot 10^{135}$	$4.94 \cdot 10^{149}$	$1.66 \cdot 10^{-3}$	0.0	$2.39 \cdot 10^{-1}$	0.0
0.5	$1.35 \cdot 10^{169}$	$1.31 \cdot 10^{187}$	$3.34 \cdot 10^{-4}$	0.0	$2.39 \cdot 10^{-1}$	0.0
0.6	$9.04 \cdot 10^{202}$	$3.48 \cdot 10^{224}$	$6.74 \cdot 10^{-5}$	0.0	$2.39 \cdot 10^{-1}$	0.0
0.7	$6.06 \cdot 10^{236}$	$9.25 \cdot 10^{261}$	$1.36 \cdot 10^{-5}$	0.0	$2.39 \cdot 10^{-1}$	0.0
0.8	$4.06 \cdot 10^{270}$	$2.46 \cdot 10^{299}$	$2.74 \cdot 10^{-6}$	0.0	$2.39 \cdot 10^{-1}$	0.0

Tabella 8.9

le corrispondenti successioni  $\{y_n\}$  decadono a zero, per  $n \rightarrow \infty$ , solamente quando  $h$  è sufficientemente piccolo, ovvero è al di sotto di un valore  $h_0$  che sembra essere proprio del metodo scelto.

Nel caso del problema (8.46), l'esame del comportamento della successione  $\{y_n\}$ , generata da un metodo Runge-Kutta oppure multistep, non è difficile. Se, per esempio,

consideriamo il generico metodo Runge-Kutta a  $r$  stadi (8.25), otteniamo

$$\begin{cases} y_{n+1} = y_n + h \sum_{i=1}^r a_i k_i \\ k_1 = \lambda y_n \\ k_i = \lambda \left( y_n + h \sum_{j=1}^{i-1} c_{ij} k_j \right), \quad i = 2, \dots, r \end{cases}$$

ovvero, come possiamo facilmente verificare,

$$y_{n+1} = p_r(h\lambda) y_n$$

dove  $p_r(\bar{h})$  è un polinomio di grado  $r$  nella variabile  $\bar{h} = h\lambda$ . Pertanto la successione  $\{y_n\}$  divergerà se  $|p_r(\bar{h})| > 1$ , mentre convergerà a zero(<sup>†</sup>) se  $|p_r(\bar{h})| < 1$ .

Quando invece utilizziamo un metodo multistep lineare a  $k$  passi (8.30), la successione  $\{y_n\}$  viene generata dalla relazione

$$\sum_{i=0}^k (\alpha_i - h\lambda\beta_i) y_{n+1-i} = 0$$

Quest'ultima è un'equazione alle differenze lineare, omogenea e a coefficienti costanti, di ordine  $k$ . La rappresentazione delle soluzioni di tali equazioni è nota ed è del tutto simile a quella delle corrispondenti equazioni differenziali (lineari a coefficienti costanti); vedi ad esempio i testi [8.2] e [8.16]. In particolare ricordiamo che la successione  $\{y_n\}$  convergerà a zero, per  $n \rightarrow \infty$ , se e solo se tutte le radici dell'equazione caratteristica

$$\sum_{i=0}^k (\alpha_i - h\lambda\beta_i) t^{k-i} = 0$$

hanno modulo minore di 1. Se invece sono presenti radici di modulo maggiore di 1, oppure di modulo unitario ma multiple, allora  $\|y_n\| \rightarrow \infty$  quando  $n \rightarrow \infty$ . La successione  $\{y_n\}$  non convergerà a zero ma rimarrà tuttavia limitata quando tutte le radici hanno modulo  $\leq 1$  e quelle di modulo unitario sono semplici.

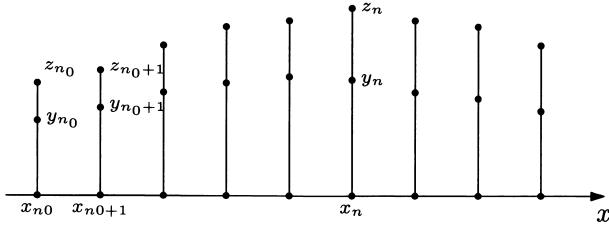
Dopo queste precisazioni, non dovrebbe risultare difficile al lettore dare una spiegazione ai comportamenti delle successioni  $\{|y(x_N) - y_N|\}$  riportate nelle tabelle 8.8 e 8.9.

Più in generale, di fronte ad un generico problema di forma (8.4), per studiare la stabilità del metodo numerico scelto, ovvero della successione  $\{y_n\}$  prodotta dal metodo quando  $h$  è fissato e  $n \rightarrow \infty$ , supporremo di introdurre delle perturbazioni(<sup>††</sup>) nei valori  $\{y_n\}$  necessari per far avanzare il metodo ed esamineremo la propagazione di tali errori nei punti successivi  $x_n$ ,  $n \rightarrow \infty$ .

(<sup>†</sup>) Al vettore nullo nel caso di un sistema di equazioni

(<sup>††</sup>) Per esempio gli errori introdotti nell'approssimazione di tali valori.

Consideriamo per esempio un metodo multistep lineare a  $k$  passi di tipo (8.30), con  $h$  costante, e denotiamo con  $\{y_n\}$  e  $\{z_n\}$  le soluzioni numeriche “uscenti” rispettivamente dai dati  $y_{n_0}, \dots, y_{n_0+k-1}$  e  $z_{n_0} = y_{n_0} + \delta_{n_0}, \dots, z_{n_0+k-1} = y_{n_0+k-1} + \delta_{n_0+k-1}$ .



**Figura 8.10**

Per esaminare il comportamento della perturbazione  $\delta_n = z_n - y_n$  quando  $n \rightarrow \infty$ <sup>(†)</sup>, dalle formule

$$\begin{aligned}\sum_{i=0}^k \alpha_i y_{n+1-i} &= h \sum_{i=0}^k \beta_i f(x_{n+1-i}, y_{n+1-i}) \\ \sum_{i=0}^k \alpha_i z_{n+1-i} &= h \sum_{i=0}^k \beta_i f(x_{n+1-i}, z_{n+1-i})\end{aligned}$$

deduciamo preliminarmente la relazione

$$\sum_{i=0}^k \alpha_i (z_{n+1-i} - y_{n+1-i}) = h \sum_{i=0}^k \beta_i [f(x_{n+1-i}, z_{n+1-i}) - f(x_{n+1-i}, y_{n+1-i})]$$

Successivamente sviluppiamo  $f(x_{n+1-i}, z_{n+1-i}) = f(x_{n+1-i}, y_{n+1-i} + \delta_{n+1-i})$  nell'intorno di  $(x_{n+1-i}, y_{n+1-i})$  e trascuriamo i termini di ordine superiore al primo; otteniamo

$$(8.47) \quad \sum_{i=0}^k \alpha_i \delta_{n+1-i} \cong h \sum_{i=0}^k \beta_i f_y(x_{n+1-i}, y_{n+1-i}) \delta_{n+1-i}$$

Ricordando le ipotesi fatte per lo studio della propagazione di  $\delta(x)$  nel problema (8.13), supponiamo lo Jacobiano  $f_y(x, y) \equiv A$  costante e diagonalizzabile ( $H^{-1}AH = \Lambda$ ), così che, ponendo  $d_{n+1-i} = H^{-1}\delta_{n+1-i}$  nella (8.47), possiamo scrivere

$$(8.48) \quad \sum_{i=0}^k (\alpha_i - h\lambda_l \beta_i) d_{n+1-i}^{(l)} = 0, \quad l = 1, \dots, m$$

con  $d_{n+1-i}^{(l)}$  denotiamo la  $l$ -esima componente del vettore  $d_{n+1-i} \in \mathbb{R}^m$ .

---

(†) Supponendo, per semplicità, di operare con precisione di calcolo infinita.

Rileviamo subito che la (8.48) poteva essere conseguita semplicemente applicando il metodo multistep al sistema (8.12). Questa osservazione vale anche per gli altri metodi numerici che abbiamo introdotto. Per esempio, nel caso di un metodo Runge-Kutta a  $r$  stadi abbiamo

$$\begin{cases} d_{n+1}^{(l)} = d_n^{(l)} + h \sum_{i=1}^r a_i k_i^{(l)} \\ k_1^{(l)} = \lambda_l d_n^{(l)} \\ k_i^{(l)} = \lambda_l [d_n^{(l)} + h \sum_{j=1}^{i-1} c_{ij} k_j^{(l)}] \end{cases} \quad l = 1, \dots, m$$

ovvero, come possiamo facilmente verificare,

$$(8.49) \quad d_{n+1}^{(l)} = p_r(h\lambda_l) d_n^{(l)}$$

con  $p_r(\bar{h})$  polinomio di grado  $r$  nella variabile  $\bar{h} = h\lambda_l$ . Per questo motivo l'equazione

$$(8.50) \quad d'(x) = \lambda d(x)$$

ove  $\lambda$  rappresenta il generico autovalore della matrice Jacobiana  $A = f_y(x, y)$  del sistema (8.11), viene di solito denominata *equazione test* per lo studio della stabilità del problema.

Sia nel caso della (8.48) che in quello della (8.49) diremo che il metodo numerico è, per  $h$  e  $\lambda_l$  assegnati, *assolutamente stabile* se  $\|d_n\| \rightarrow 0$ , ovvero  $\|\delta_n\| \rightarrow 0$ ,  $n \rightarrow \infty$ <sup>(†)</sup>.

Nelle (8.48) e (8.49) i coefficienti dei rispettivi polinomi caratteristici

$$\begin{aligned} \sum_{i=0}^k (\alpha_i - h\lambda_l \beta_i) t^{k-i} &= 0 \\ t - p_r(h\lambda_l) &= 0 \end{aligned}$$

dipendono dal parametro  $\bar{h} = h\lambda_l$ ; ha quindi senso chiederci per quali valori, in generale complessi, di  $\bar{h}$  le radici dei suddetti polinomi hanno tutte modulo minore di 1.

**Definizione 8.8.** Un metodo numerico è detto *assolutamente stabile in una regione*  $H_a = \{\bar{h} = h\lambda\}$  del piano complesso  $\mathbb{C}$  se per ogni  $\bar{h} \in H_a$  le radici del corrispondente polinomio caratteristico hanno tutte modulo minore di 1. La regione  $H_a$  viene chiamata *regione di stabilità assoluta del metodo*.

La regione di stabilità assoluta  $H_a$  è una caratteristica propria di ogni metodo.

Nel caso di metodi molto semplici, quali Eulero e trapezi, essa può essere dedotta direttamente dalle espressioni analitiche delle radici del polinomio caratteristico. Infatti, nel caso del metodo di Eulero

$$y_{n+1} = y_n + hf(x_n, y_n)$$

---

(†) Dal punto di vista del calcolo numerico, quando  $\|y_n\| \rightarrow \infty$  sarebbe sufficiente richiedere che la perturbazione relativa  $\|\delta_n\|/\|y_n\|$  si mantenga limitata per  $n \rightarrow \infty$ .

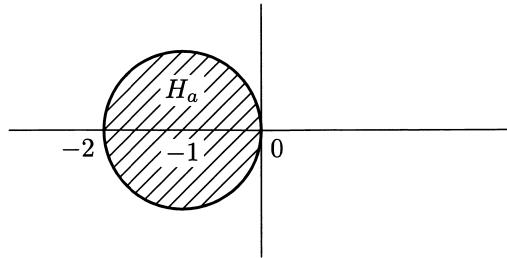
che applicato all'equazione test (8.50) assume la forma

$$d_{n+1} - (1 + h\lambda)d_n = 0$$

otteniamo un'*equazione caratteristica* di primo grado

$$t - (1 + \bar{h}) = 0$$

La regione  $H_a = \{\bar{h} \in \mathbb{C} : |1 + \bar{h}| < 1\}$  è rappresentata dal cerchio unitario di centro  $(-1, 0)$  (vedi figura 8.11). Per i trapezi



**Figura 8.11**

$$y_{n+1} = y_n + \frac{h}{2}[f(x_{n+1}, y_{n+1}) + f(x_n, y_n)]$$

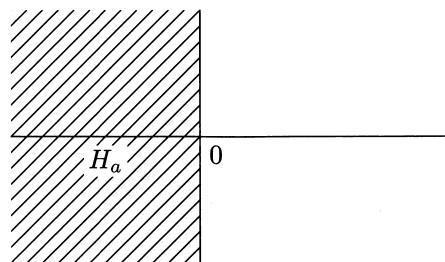
otteniamo invece

$$\left(1 - \frac{\bar{h}}{2}\right)d_{n+1} - \left(1 + \frac{\bar{h}}{2}\right)d_n = 0$$

donde

$$\left(1 - \frac{\bar{h}}{2}\right)t - \left(1 + \frac{\bar{h}}{2}\right) = 0$$

In questo caso  $H_a$  coincide con l'intero semipiano complesso  $\{\bar{h} \in \mathbb{C} : \operatorname{Re}(\bar{h}) < 0\}$ .



**Figura 8.12**

In situazioni più complesse occorre invece ricorrere a tecniche di tipo numerico (vedere [8.19]).

Nella tabella 8.10 riportiamo gli *intervalli di stabilità assoluta*<sup>(†)</sup>, di tipo  $(-\xi, 0)$ , dei metodi Adams-Bashforth (*AB*) e Adams-Moulton (*AM*) a  $k = 1, 2, 3, 4$  passi. L'intervallo di stabilità assoluta dei metodi impliciti è, soprattutto a parità di ordine,

$k$	$\xi$	
	<i>AB</i>	<i>AM</i>
1	2	$\infty$
2	1	6
3	$\frac{6}{11}$	3
4	$\frac{3}{10}$	$\frac{90}{49}$

Tabella 8.10

molto più ampio di quello dei corrispondenti metodi esplicativi.

Nella figura 8.13 sono infine disegnate le regioni di stabilità assoluta dei metodi Runge-Kutta a  $r$  stadi e ordine  $r$ ,  $r = 1, 2, 3, 4$ .

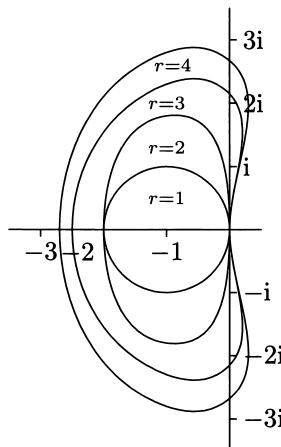


Figura 8.13

▷ **Osservazione.** Scegliere il passo  $h$  in modo che il metodo numerico risulti assolutamente stabile ha senso solo quando il problema, cui il metodo viene applicato, risulta asintoticamente stabile. Diversamente è sufficiente avere una propagazione stabile delle

(†) Con il termine *intervallo di stabilità assoluta* intendiamo l'intersezione di  $H_a$  con l'asse reale.

perturbazioni relative (vedi ad esempio il concetto di stabilità relativa presentato in [8.19, pag. 75]).  $\triangleleft$

Concludiamo la nostra analisi della stabilità dei metodi numerici con un ultimo esempio. Applichiamo i metodi di Heun e dei trapezi al problema

$$\begin{cases} y'(x) = -10^3(y(x) - \cos(x)) - \sin(x) \\ y(0) = 2 \end{cases}$$

la cui soluzione è  $y(x) = e^{-10^3 x} + \cos(x)$ . Nella tabella 8.11 riportiamo alcuni risultati prodotti dai due metodi.

$x_n$	$y(x_n)$	$ y(x_n) - y_n , h = 0.01$		$ y(x_n) - y_n , h = 0.001$	
		Heun	Trapezi	Heun	Trapezi
0.1	0.995004...	$1.34 \cdot 10^{16}$	$1.73 \cdot 10^{-2}$	$4.98 \cdot 10^{-7}$	$8.24 \cdot 10^{-12}$
0.2	0.980066...	$1.80 \cdot 10^{32}$	$3.01 \cdot 10^{-4}$	$4.90 \cdot 10^{-7}$	$1.65 \cdot 10^{-11}$
0.3	0.955336...	$2.42 \cdot 10^{48}$	$5.22 \cdot 10^{-6}$	$4.78 \cdot 10^{-7}$	$2.45 \cdot 10^{-11}$
0.4	0.921061...	$3.25 \cdot 10^{64}$	$9.37 \cdot 10^{-8}$	$4.61 \cdot 10^{-7}$	$3.24 \cdot 10^{-11}$
0.5	0.877582...	$4.36 \cdot 10^{80}$	$5.56 \cdot 10^{-9}$	$4.39 \cdot 10^{-7}$	$3.29 \cdot 10^{-11}$
0.6	0.825335...	$5.85 \cdot 10^{96}$	$4.73 \cdot 10^{-9}$	$4.13 \cdot 10^{-7}$	$4.70 \cdot 10^{-11}$
0.7	0.764842...	$7.85 \cdot 10^{112}$	$5.36 \cdot 10^{-9}$	$3.83 \cdot 10^{-7}$	$5.36 \cdot 10^{-11}$
0.8	0.696706...	$1.05 \cdot 10^{129}$	$5.97 \cdot 10^{-9}$	$3.49 \cdot 10^{-7}$	$5.97 \cdot 10^{-11}$
0.9	0.621610...	$1.41 \cdot 10^{145}$	$6.52 \cdot 10^{-9}$	$3.11 \cdot 10^{-7}$	$6.52 \cdot 10^{-11}$
1.0	0.540302...	$1.90 \cdot 10^{161}$	$7.01 \cdot 10^{-9}$	$2.71 \cdot 10^{-7}$	$7.01 \cdot 10^{-11}$

Tabella 8.11

Lasciamo al lettore il compito di spiegare il diverso comportamento dei due metodi considerati.

## 8.5 Sistemi stiff

Consideriamo il problema

$$(8.51) \quad \begin{cases} y'(x) = f(x, y(x)), & a \leq x \leq b \\ y(a) = y_0 \end{cases}$$

e denotiamo con  $\{\lambda_l \equiv \lambda_l(x, y)\}$  gli autovalori della matrice Jacobiana  $f_y(x, y) \in \mathbb{R}^{m \times m}$ .

**Definizione 8.9.** Il problema (8.51) è detto *stiff* nell'intervallo  $[a, b]$  se:

- (i) agli eventuali autovalori  $\lambda_i$  con parte reale positiva corrispondono quantità  $(b - a) \cdot \text{Re}(\lambda_i)$  non grandi;

(ii) esiste almeno un autovalore  $\lambda_j$  con parte reale negativa e  $(b-a)\operatorname{Re}(\lambda_j) \ll -1$ .

La quantità  $S_M = (b-a) \max_{\operatorname{Re}(\lambda_j) < 0} |\operatorname{Re}(\lambda_j)|$  rappresenta una misura del *grado di stiffness* del problema; non è raro incontrare sistemi stiff con  $S_M = 10^6 \div 10^{10}$ .

In un sistema stiff almeno un autovalore ha la parte reale negativa molto grande; ciò significa che anche la costante di Lipschitz  $L$  del sistema è molto grande:

$$L \geq \|f_y(x, y)\| \geq \rho(f_y) = \max_{1 \leq l \leq m} |\lambda_l|$$

Esaminiamo ora brevemente le principali difficoltà che insorgono quando affrontiamo un sistema stiff. Poiché  $\operatorname{Re}(\lambda_i) < 0$  per almeno un autovalore, affinché il comportamento (dal punto di vista della propagazione degli errori) del metodo scelto sia analogo a quello del problema (8.51) cui esso viene applicato, occorre scegliere il passo di integrazione  $h$  in modo che tutti i punti  $\bar{h}_j = h\lambda_j$ ,  $\operatorname{Re}(\lambda_j) < 0$ , appartengano alla regione di stabilità assoluta  $H_a$  del metodo; in caso contrario assisteremo ad una esplosione degli errori globali prodotti dal metodo (vedi ad esempio i risultati riportati nella tabella 8.11). Metodi con regioni  $H_a$  piccole, per esempio i metodi Runge-Kutta, impongono scelte di passi  $h$  che possono risultare eccessivamente piccoli rispetto all'ampiezza dell'intervallo di integrazione  $[a, b]$ . Per evitare queste limitazioni sulla scelta di  $h$ , occorrerebbe avere a disposizione metodi numerici con regioni di stabilità assoluta contenenti l'intero semipiano  $\operatorname{Re}(\bar{h}) < 0$ .

**Definizione 8.10.** Un metodo numerico è detto *A-stabile* se la sua regione di stabilità assoluta contiene l'intero semipiano  $\operatorname{Re}(\bar{h}) < 0$ .

Questa definizione è stata originariamente formulata in relazione ai soli metodi multistep lineari; tuttavia già per questa classe di metodi essa appare molto restrittiva. Infatti è stato dimostrato che

- (i) un metodo multistep lineare esplicito non può essere *A-stabile*;
- (ii) un metodo multistep lineare *A-stabile*, necessariamente implicito, ha ordine  $p \leq 2$ .

Osserviamo che la formula dei trapezi è *A-stabile*.

Definizioni di stabilità meno restrittive sono state successivamente introdotte; qui ne riportiamo solamente una.

**Definizione 8.11.** Un metodo numerico è detto  *$A(\alpha)$ -stabile*,  $\alpha \in (0, \pi/2)$ , se la sua regione di stabilità assoluta contiene la regione

$$W_\alpha = \{ \bar{h} : -\alpha < \pi - \arg \lambda < \alpha \}$$

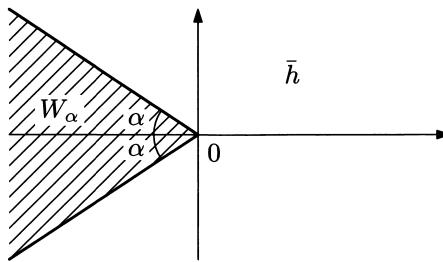


Figura 8.14

Un metodo è detto  $A(0)$ -stabile se è  $A(\alpha)$ -stabile per un  $\alpha \in (0, \pi/2)$  sufficientemente piccolo.

È stato dimostrato che nessun metodo multistep lineare esplicito può essere  $A(0)$ -stabile.

Per molte classi di metodi one-step e multistep la richiesta di  $A$ -stabilità, e anche di  $A(0)$ -stabilità, impone al metodo di essere implicito. Nel caso, per esempio, di un metodo multistep lineare questo vincolo comporta, per ogni singolo passo di integrazione, la risoluzione di un sistema di equazioni (in generale non lineare) del tipo

$$(8.52) \quad y_{n+1} = h\beta_0 f(x_{n+1}, y_{n+1}) + g_n$$

dove  $g_n$  è un vettore noto. Gli schemi previsore-correttore, con correttore iterato un numero prefissato di volte, non risultano idonei per i sistemi stiff; essi sono di tipo esplicito e la loro regione di stabilità assoluta non coincide affatto con quella del correttore ed è generalmente molto piccola. Neppure l'iterazione del correttore alla convergenza produce dei miglioramenti, in quanto essa impone la scelta di un passo  $h$  eccessivamente piccolo:

$$(8.53) \quad h < \frac{1}{\beta_0 L}$$

(non dimentichiamo che in un sistema stiff la costante di Lipschitz  $L$  è molto grande); anzi, tale condizione potrebbe risultare più restrittiva di quella richiesta dalla stabilità assoluta di un metodo esplicito.

La risoluzione del sistema (8.52) viene effettuata, generalmente con maggiore successo, utilizzando il metodo di Newton:

$$\begin{aligned} J^{(i)}(y_{n+1}^{(i+1)} - y_{n+1}^{(i)}) &= -y_{n+1}^{(i)} + h\beta_0 f(x_{n+1}, y_{n+1}^{(i)}) + g_n, \quad i = 0, 1, \dots \\ J^{(i)} &= I - h\beta_0 f_y(x_{n+1}, y_{n+1}^{(i)}) \end{aligned}$$

Ovviamente, dovremo scegliere il passo  $h$  sufficientemente piccolo in modo che la matrice  $J^{(i)}$  sia non singolare e il processo iterativo di Newton risulti convergente. In pratica questa condizione su  $h$  si rivela meno restrittiva della (8.53). Occorre inoltre possedere

una buona approssimazione iniziale  $y_{n+1}^{(0)}$ , che potremmo ottenere, per esempio, con una formula esplicita (previsore).

Poiché le formule implicite di Adams-Moulton di ordine  $p \geq 3$  hanno regioni di stabilità assoluta  $H_a$  molto limitate (vedi ad esempio la tabella 8.10), sono state costruite formule multistep alternative di ordine  $p \geq 3$  e regioni  $H_a$  illimitate. Tra queste, particolare successo hanno avuto i metodi BDF (*Backward Differentiation Formulas*), ottenibili interpolando, con un polinomio di grado  $k$  nei nodi  $x_0 = x^* + h, x_{-1} = x^*, \dots, x_{-k}$ , la soluzione del problema

$$\begin{cases} u'(x) = f(x, u(x)) \\ u(x^*) = y^* \end{cases}$$

e sostituendo, nell'identità  $u'(x^* + h) = f(x^* + h, u(x^* + h))$ ,  $u'(x^* + h)$  con la derivata della predetta formula di interpolazione (di Lagrange oppure di Newton alle differenze regressive). I metodi che così otteniamo, a  $k$  passi, hanno ordine di consistenza  $p = k$ ; tuttavia, solo per  $k \leq 6$  essi risultano anche convergenti. Essi possono essere scritti nella forma

$$\sum_{i=0}^k \alpha_i y_{n+1-i} = hf_{n+1}$$

con

$k$	$\alpha_0$	$\alpha_1$	$\alpha_2$	$\alpha_3$	$\alpha_4$	$\alpha_5$	$\alpha_6$
1	1	-1					
2	$\frac{3}{2}$	-2	$\frac{1}{2}$				
3	$\frac{11}{6}$	-3	$\frac{3}{2}$	$-\frac{1}{3}$			
4	$\frac{25}{12}$	-4	3	$-\frac{4}{3}$	$\frac{1}{4}$		
5	$\frac{137}{60}$	-5	5	$-\frac{10}{3}$	$\frac{5}{4}$	$-\frac{1}{5}$	
6	$\frac{147}{60}$	-6	$\frac{15}{2}$	$-\frac{20}{3}$	$\frac{15}{4}$	$-\frac{6}{5}$	$\frac{1}{6}$

Tabella 8.12

oppure

$$\sum_{j=1}^k \frac{1}{j} \nabla^j y_{n+1} = hf_{n+1}$$

e risultano tutti  $A(\alpha)$ -stabili (vedi, ad esempio, [8.4] e [8.21, pag. 311]).

La prima formula ( $k = 1$ )

$$y_{n+1} = y_n + hf(x_{n+1}, y_{n+1})$$

coincide con il metodo di Eulero implicito.

Per una visione più completa dei metodi proposti per la risoluzione di sistemi stiff consigliamo le letture [8.12], [8.13], [8.19], [8.20].

Concludiamo osservando che il grado di stiffness di un problema può variare in modo sensibile nell'intervallo di integrazione  $[a, b]$ . Infatti, gli autovalori  $\{\lambda_l\}$  del modello lineare sono in realtà funzioni della variabile  $x \in [a, b]$ ; inoltre, il comportamento locale del problema può non essere validamente rappresentato dallo spettro dello Jacobiano  $f_y(x, y)$ . A volte è difficile capire se un sistema è stiff oppure no.

## 8.6 Problemi con valori ai limiti

Finora abbiamo considerato equazioni differenziali con condizioni (iniziali) assegnate in un unico punto  $x = a$ . In questo ultimo paragrafo vogliamo affrontare la risoluzione numerica di equazioni differenziali con condizioni (ai limiti) assegnate in due punti distinti; per esempio,

$$\begin{cases} y''(x) = y(x), & 0 \leq x \leq 1 \\ y(0) = 0 \\ y(1) = 1 \end{cases}$$

oppure

$$\begin{cases} \frac{d^2}{dx^2}(EI(x)y''(x)) + ky(x) = q(x), & -L \leq x \leq L \\ y''(-L) = y'''(-L) = 0 \\ y''(L) = y'''(L) = 0 \end{cases}$$

dove  $E$ ,  $I(x)$ ,  $k$  e  $q(x)$  sono quantità note.

Problemi con condizioni su due punti sono tutt'altro che rari, e la loro risoluzione numerica può presentare notevoli difficoltà.

Le condizioni ai limiti possono non essere *separate* come nei due esempi sopra riportati. In generale il problema può essere descritto da un sistema di  $m$  equazioni differenziali non lineari del primo ordine, accoppiate a  $m$  condizioni non lineari sui valori di  $y(a)$  e  $y(b)$ :

$$(8.54) \quad \begin{cases} y'(x) = f(x, y(x)), & a \leq x \leq b \\ g(y(a), y(b)) = 0 \end{cases}$$

Nel caso lineare abbiamo

$$\begin{cases} y'(x) = A(x)y(x) + r(x), & a \leq x \leq b \\ B_a y(a) + B_b y(b) = \alpha \end{cases}$$

dove  $A(x)$ ,  $B_a$  e  $B_b$  sono matrici di ordine  $m$ .

La determinazione di condizioni che garantiscano l'esistenza e unicità di soluzioni di problemi con valori ai limiti risulta ben più complessa che nel caso di problemi a valori

iniziali. Anche quando le ipotesi del teorema 8.1, che assicurano l'esistenza e unicità della soluzione del corrispondente problema a valori iniziali, sono verificate, il sistema (8.54) può non avere soluzione oppure averne infinite. L'esempio che segue, nonostante sia di forma assai semplice, pone in evidenza le difficoltà che un problema di tipo (8.54) può presentare.

Consideriamo

$$(8.55) \quad \begin{cases} y''(x) + q^2 y(x) = 0, & 0 \leq x \leq 1, \\ y(0) = 0 \\ y(1) = \beta \end{cases} \quad q \neq 0$$

L'integrale generale dell'equazione differenziale, che assume il valore  $y(0) = 0$ , è  $y(x) = C \sin(qx)$ , dove  $C$  è una costante arbitraria. Se  $q = n\pi$ ,  $n \neq 0$  intero, il problema (8.55) non ha soluzione quando  $\beta \neq 0$ , mentre ne ha infinite se  $\beta = 0$ .

Osserviamo che il corrispondente problema a valori iniziali

$$\begin{cases} y''(x) + q^2 y(x) = 0, & 0 \leq x \leq 1, \\ y(0) = 0 \\ y'(0) = \gamma \end{cases} \quad q \neq 0$$

ammette invece una ed una sola soluzione:  $y(x) = \frac{\gamma}{q} \sin(qx)$ .

Per individuare le possibili soluzioni del sistema (8.54), associamo a quest'ultimo il seguente problema a valori iniziali:

$$(8.56) \quad \begin{cases} u'(x) = f(x, u(x)), & a \leq x \leq b \\ u(a) = s \end{cases}$$

dove  $s \in \mathbb{R}^m$  denota un parametro. Nell'ipotesi che il problema (8.56) ammetta un'unica soluzione per ogni valore del dato iniziale  $s$ , che denotiamo con  $u(x; s)$ , quest'ultima sarà anche soluzione di (8.54), ovvero  $y(x) \equiv u(x; s)$ , solo se risulterà verificata la condizione

$$(8.57) \quad \Phi(s) \equiv g(s, u(b; s)) = 0$$

Pertanto il problema (8.54) avrà tante soluzioni quante sono le radici distinte  $s^*$  dell'equazione (8.57). Dette soluzioni coincideranno quindi con quelle di (8.56) definite dai valori iniziali  $s = s^*$ .

Nel caso particolare di un problema lineare del secondo ordine, di forma

$$\begin{cases} y''(x) = p(x)y'(x) + q(x)y(x) + r(x), & a \leq x \leq b \\ a_0y(a) - a_1y'(a) = \alpha \\ b_0y(b) + b_1y'(b) = \beta \end{cases}$$

con  $p(x), q(x), r(x) \in C[a, b]$ , le condizioni

$$q(x) > 0, \quad a \leq x \leq b$$

$$\begin{aligned} a_0 a_1 &\geq 0, & b_0 b_1 &\geq 0 \\ |a_0| + |a_1| &\neq 0, & |b_0| + |b_1| &\neq 0, & |a_0| + |b_0| &\neq 0 \end{aligned}$$

garantiscono (vedi [8.3, pag.11]) l'esistenza e unicità della soluzione (classica) del problema. Esse sono sufficienti ma in generale non necessarie; per esempio, anche il problema

$$\begin{cases} y''(x) = y(x), & 0 \leq x \leq 1 \\ y'(0) = 0 \\ y'(1) = 1 \end{cases}$$

nel quale  $|a_0| + |b_0| = 0$ , ammette un'unica soluzione:  $y(x) = \frac{e}{e^2-1}(e^x + e^{-x})$ .

Una trattazione completa dell'argomento risulterebbe troppo vasta, e comunque non di pertinenza di un primo corso sui fondamenti del calcolo numerico. Al lettore interessato ad approfondire l'argomento consigliamo le letture [8.1], [8.3] e [8.18].

Anche nei paragrafi che seguono ci limiteremo ad esporre le idee fondamentali che sono alla base della costruzione di alcuni dei metodi numerici più comuni. Per una trattazione più completa suggeriamo i testi [8.3] e [8.18].

### 8.6.1 Metodo delle differenze finite

Supponiamo per esempio di voler risolvere il problema

$$(8.58) \quad \begin{cases} y''(x) = f(x, y(x), y'(x)), & a \leq x \leq b \\ y(a) = \alpha \\ y(b) = \beta \end{cases}$$

assumendo che esso abbia soluzione  $y(x) \in C^4[a, b]$ . Dividiamo l'intervallo di interesse  $[a, b]$  in  $N$  parti uguali di ampiezza  $h$ :

$$a \equiv x_0 \quad x_1 \quad x_2 \quad \cdots \quad x_n \quad \cdots \quad x_N \equiv b \quad x_n = x_0 + nh$$

Consideriamo l'equazione differenziale nei soli nodi interni  $x_n$ ,  $n = 1, \dots, N - 1$ , e approssimiamo le derivate  $y'(x_n)$  e  $y''(x_n)$  con le formule alle differenze di pagina 174

$$\begin{aligned} y'(x_n) &= \frac{1}{2h}[y(x_{n+1}) - y(x_{n-1})] + O(h^2) \\ y''(x_n) &= \frac{1}{h^2}[y(x_{n+1}) - 2y(x_n) + y(x_{n-1})] + O(h^2) \end{aligned}$$

Otteniamo il problema approssimato

$$(8.59) \quad \begin{cases} y_{n+1} - 2y_n + y_{n-1} = h^2 f\left(x_n, y_n, \frac{y_{n+1} - y_{n-1}}{2h}\right), & n = 1, 2, \dots, N - 1 \\ y_0 = \alpha \\ y_N = \beta \end{cases}$$

sistema, in generale non lineare, che possiamo risolvere, per esempio, con il metodo di Newton. Osserviamo che ogni equazione coinvolge 3 sole incognite  $y_{n-1}, y_n, y_{n+1}$ , per cui lo Jacobiano del sistema risulterà tridiagonale.

Quando  $f(x, y, y')$  è lineare, di forma

$$(8.60) \quad \begin{cases} y''(x) - p(x)y'(x) - q(x)y(x) = r(x) \\ y(a) = \alpha \\ y(b) = \beta \end{cases}$$

il sistema (8.59) è lineare tridiagonale:

$$(8.61) \quad \begin{cases} \left(1 + p(x_n)\frac{h}{2}\right)y_{n-1} - (2 + q(x_n)h^2)y_n + \left(1 - p(x_n)\frac{h}{2}\right)y_{n+1} = h^2r(x_n) & n = 1, 2, \dots, N-1 \\ y_0 = \alpha \\ y_N = \beta \end{cases}$$

Se inoltre  $p(x), q(x), r(x) \in C[a, b]$ ,  $q(x) > 0$  in  $[a, b]$ , e scegliamo  $h \leq 2/\|p(x)\|_\infty$ , il sistema (8.61) risulta anche a diagonale dominante e quindi non singolare. Poiché abbiamo supposto  $y(x) \in C^4[a, b]$ , è possibile dimostrare (vedi [8.18, §5.1.1]) che lo *schema alle differenze finite* (8.61) è convergente per  $h \rightarrow 0$ :

$$\max_{1 \leq n \leq N-1} |y(x_n) - y_n| = O(h^2)$$

► **Esempio.** Applichiamo lo schema (8.61) al problema

$$\begin{cases} y''(x) - y(x) = x, & 0 \leq x \leq 1 \\ y(0) = 0 \\ y(1) = 0 \end{cases}$$

la cui soluzione è  $y(x) = (e^x - e^{-x})/(e - e^{-1}) - x$ . Nella tabella 8.13 riportiamo alcuni errori relativi  $\varepsilon_n = |y(x_n) - y_n| / |y(x_n)|$ .

$x_n$	$\varepsilon_n$			
	$N = 5$	$N = 10$	$N = 20$	$N = 40$
0.2	$2.97 \cdot 10^{-3}$	$7.45 \cdot 10^{-4}$	$1.89 \cdot 10^{-4}$	$3.14 \cdot 10^{-5}$
0.4	$2.99 \cdot 10^{-3}$	$7.50 \cdot 10^{-4}$	$1.90 \cdot 10^{-4}$	$3.18 \cdot 10^{-5}$
0.6	$3.02 \cdot 10^{-3}$	$7.57 \cdot 10^{-4}$	$1.92 \cdot 10^{-4}$	$3.28 \cdot 10^{-5}$
0.8	$3.06 \cdot 10^{-3}$	$7.68 \cdot 10^{-4}$	$1.94 \cdot 10^{-4}$	$3.55 \cdot 10^{-5}$

Tabella 8.13

Se le condizioni ai limiti sono del tipo

$$\begin{aligned}y'(a) + \gamma y(a) &= 0 \\y'(b) + \delta y(b) &= 0\end{aligned}$$

per mantenere l'ordine  $O(h^2)$  nelle approssimazioni delle singole derivate possiamo approssimare  $y'(a)$  e  $y'(b)$  con le formule

$$\begin{aligned}y'(a) &= \frac{1}{2h}[y(x_1) - y(x_{-1})] + O(h^2) \\y'(b) &= \frac{1}{2h}[y(x_{N+1}) - y(x_{N-1})] + O(h^2)\end{aligned}$$

Occorre però introdurre i due nodi esterni  $x_{-1}$  e  $x_{N+1}$  e approssimare l'equazione differenziale anche in  $x_0$  e  $x_N$ . Così facendo otteniamo il sistema

$$\begin{cases} y_{n-1} - 2y_n + y_{n+1} = h^2 f\left(x_n, y_n, \frac{y_{n+1} - y_{n-1}}{2h}\right), & n = 0, 1, \dots, N \\ y_{-1} = y_1 + 2h\gamma y_0 \\ y_{N+1} = y_{N-1} - 2h\delta y_N \end{cases}$$

La scelta del passo  $h$  costante non è sempre efficiente, soprattutto quando il comportamento della soluzione  $y(x)$  non è uniforme su tutto  $[a, b]$ . In quest'ultimo caso è tuttavia possibile approssimare le derivate  $y'(x_n)$  e  $y''(x_n)$  utilizzando formule alle differenze con nodi non equidistanti (vedi ad esempio [8.18, §5.6.1]).

Il procedimento che abbiamo sin qui illustrato viene di solito proposto per la risoluzione di una singola equazione differenziale del secondo ordine. Nel caso di un generico sistema di equazioni del primo ordine si preferisce agire diversamente, ricorrendo a formule di discretizzazione di tipo one-step implicito, proprie dei metodi numerici per problemi a valori iniziali.

Consideriamo il sistema lineare di ordine  $m$

$$(8.62) \quad \begin{cases} y'(x) = A(x)y(x) + r(x), & a \leq x \leq b \\ B_a y(a) + B_b y(b) = \alpha \end{cases}$$

e suddividiamo l'intervallo  $[a, b]$  in  $N$  parti  $a = x_0 < x_1 < x_2 < \dots < x_{N-1} < x_N = b$  non necessariamente uguali. Un primo metodo può essere costruito richiamando la formula dei trapezi:

$$(8.63) \quad \begin{cases} y_{n+1} = y_n + \frac{h_n}{2}[A(x_{n+1})y_{n+1} + A(x_n)y_n + r(x_{n+1}) + r(x_n)], & n = 0, 1, \dots, N-1 \\ B_a y_0 + B_b y_N = \alpha \end{cases}$$

con  $h_n = x_{n+1} - x_n$ .

Lo schema di calcolo può essere posto nella forma

$$\begin{pmatrix} S_1 & R_1 & & \\ & S_2 & R_2 & \\ & \ddots & \ddots & \\ & & S_N & R_N \\ B_a & & & B_b \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ y_{N-1} \\ y_N \end{pmatrix} = \begin{pmatrix} q_0 \\ q_1 \\ \vdots \\ q_{N-1} \\ \alpha \end{pmatrix}$$

dove  $S_{n+1} = -(1/h_n)I - (1/2)A(x_n)$ ,  $R_{n+1} = (1/h_n)I - (1/2)A(x_{n+1})$  e  $q_n = [r(x_{n+1}) + r(x_n)]/2$ . L'ordine di convergenza del metodo è 2, ovvero

$$\max_{0 \leq n \leq N} \|y(x_n) - y_n\| = O(h^2), \quad h = \max_n h_n$$

Quando invece la formula dei trapezi viene applicata ad un problema non lineare (8.54), perveniamo ad un sistema (non lineare) di forma

$$(8.64) \quad \begin{cases} y_{n+1} = y_n + \frac{h_n}{2}[f(x_{n+1}, y_{n+1}) + f(x_n, y_n)], & n = 0, \dots, N-1 \\ g(y_0, y_N) = 0 \end{cases}$$

Metodi con ordine di convergenza superiore a 2 sono costruibili ricorrendo a formule Runge-Kutta di tipo implicito oppure a tecniche di accelerazione applicate a schemi quale quello dei trapezi (vedere [8.18, cap. 5]).

### 8.6.2 Metodo shooting (o di puntamento)

Questo metodo è una diretta conseguenza di quanto esposto nella parte introduttiva 8.6: al sistema (8.54) associamo il corrispondente problema a valori iniziali (8.56) che, per ogni valore di  $s$ , risolviamo con un metodo one-step o multistep. Ricordando la condizione (8.57) e la successiva affermazione, determiniamo quindi, con uno dei metodi numerici (iterativi) del capitolo 6, una successione di approssimazioni  $s_n$ ,  $n = 0, 1, 2, \dots$ , convergente alla radice  $s^*$  dell'equazione (8.57). Ad ogni iterazione del metodo occorrerà risolvere un problema a valori iniziali (8.56) con  $s = s_n$ :

$$\begin{cases} u'(x) = f(x, u(x)), & a \leq x \leq b \\ u(a) = s_n \end{cases}$$

La soluzione del problema ai limiti coinciderà con quella del corrispondente problema (8.56) quando il valore iniziale di quest'ultimo coinciderà con la radice  $s^*$ . Il metodo nel suo complesso consente pertanto di determinare, quando converge, un'approssimazione della soluzione cercata mediante la risoluzione di una successione finita di problemi a valori iniziali.

Come esempio consideriamo il sistema

$$(8.65) \quad \begin{cases} y''(x) = f(x, y(x), y'(x)), & a \leq x \leq b \\ y(a) = \alpha \\ y(b) = \beta \end{cases}$$

ovvero

$$(8.66) \quad \begin{cases} y'(x) = z(x) \\ z'(x) = f(x, y(x), z(x)), & a \leq x \leq b \\ y(a) = \alpha \\ y(b) = \beta \end{cases}$$

cui associamo il seguente problema a valori iniziali

$$(8.67) \quad \begin{cases} u'(x) = v(x) \\ v'(x) = f(x, u(x), v(x)), & a \leq x \leq b \\ u(a) = \alpha \\ v(a) = s \end{cases}$$

Supponiamo per semplicità che quest'ultimo ammetta per ogni valore del parametro  $s$  una ed una sola soluzione definita in tutto  $[a, b]$ , che indichiamo con  $u(x; s)$ . La funzione  $u(x; s)$  è anche soluzione del problema (8.65) solo se risulta

$$(8.68) \quad u(b; s) \equiv \Phi(s) = \beta$$

La (8.68) rappresenta un'equazione, in generale non lineare, nell'incognita  $s = s^*$ , che possiamo risolvere, per esempio, con il metodo di Newton:

$$s_{n+1} = s_n - \frac{\Phi(s_n) - \beta}{\Phi'(s_n)}, \quad n = 0, 1, \dots$$

tuttavia per determinare  $\Phi'(s_n)$  dovremmo risolvere un ulteriore problema a valori iniziali. Questo aggravio di calcolo non si presenta se invece applichiamo alla (8.68) il metodo delle secanti.

Il procedimento iterativo per determinare  $y(x) \equiv u(x; s^*)$  può essere riassunto dal seguente schema di calcolo, nel quale si suppone di aver scelto i valori di  $s_0$  e  $s_1$  richiesti dal metodo delle secanti:

1: Risolvi il problema a valori iniziali

$$\begin{cases} u'(x) = v(x) \\ v'(x) = f(x, u(x), v(x)) \\ u(a) = \alpha \\ v(a) = s_0 \end{cases}$$

da  $x = a$  sino a  $x = b$

$$\Phi(s_0) \leftarrow u(b; s_0).$$

2: **ciclo 1:**  $n = 1, 2, \dots, n_{\max}$

3: Risolvi il problema a valori iniziali

$$\begin{cases} u'(x) = v(x) \\ v'(x) = f(x, u(x), v(x)) \\ u(a) = \alpha \\ v(a) = s_n \end{cases}$$

da  $x = a$  sino a  $x = b$

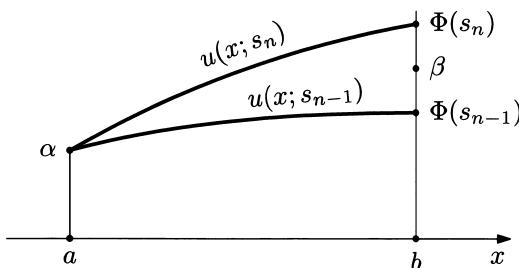
$$\Phi(s_n) \leftarrow u(b; s_n)$$

4: Determina la nuova approssimazione

$$s_{n+1} \leftarrow s_n + (s_n - s_{n-1}) \frac{\beta - \Phi(s_n)}{\Phi(s_n) - \Phi(s_{n-1})}$$

5: se  $|\Phi(s_n) - \beta| < \text{tol}$  allora stop

6: fine ciclo 1



**Figura 8.15**

Quando  $f(x, y, y')$  è lineare nelle variabili  $y$  e  $y'$ , ovvero  $f(x, y, y') = c_1(x)y + c_2(x)y' + c_3(x)$ , una sola applicazione del metodo delle secanti individua la radice dell'equazione (8.68); infatti in questo caso la funzione  $\Phi(s)$  è lineare nella variabile  $s$ , e  $s_2$  rappresenta

la radice cercata (a meno degli errori di troncamento commessi nelle determinazioni di  $\Phi(s_0)$  e  $\Phi(s_1)$ ).

▷ **Osservazione.** Se  $f(x, y, y')$  è lineare, la combinazione lineare

$$z(x) = \gamma u(x; s_0) + (1 - \gamma)u(x; s_1), \quad a \leq x \leq b$$

con  $\gamma$  costante arbitraria e  $s_0 \neq s_1$ , rappresenta ancora una soluzione dell'equazione differenziale in (8.65); inoltre  $z(a) = \alpha$  qualunque sia il valore di  $\gamma$ . Supponendo  $u(b; s_0) \neq u(b; s_1)$  possiamo determinare  $\gamma$  in modo che anche la seconda condizione  $z(b) = \beta$  sia soddisfatta, e quindi  $z(x) \equiv y(x)$ . Infatti, dalla relazione

$$z(b) = \gamma u(b; s_0) + (1 - \gamma)u(b; s_1)$$

deduciamo

$$\gamma = \frac{\beta - \Phi(s_1)}{\Phi(s_0) - \Phi(s_1)}$$

□

Il metodo shooting che abbiamo descritto non è sempre utilizzabile. Può succedere che nonostante il problema ai limiti (8.54) sia ben condizionato, il corrispondente ai valori iniziali (8.56) non lo sia. Inoltre, partendo da un errato valore iniziale di tentativo  $s = s_0$ , la soluzione  $u(x; s_0)$  potrebbe non essere definita su tutto  $[a, b]$ . Per attenuare tali inconvenienti viene proposto il metodo di shooting multiplo ([8.18]).

### 8.6.3 Metodo di collocazione

Il metodo di collocazione in generale può essere applicato a sistemi di equazioni differenziali non necessariamente del primo ordine. Noi lo illustreremo considerando per semplicità una singola equazione del secondo ordine

$$(8.69) \quad \begin{cases} y''(x) = f(x, y(x), y'(x)), & a \leq x \leq b \\ g_1(y(a), y(b)) = 0 \\ g_2(y(a), y(b)) = 0 \end{cases}$$

Sia  $\{\varphi_i(x), i = 0, 1, \dots, N\}$  un sistema di funzioni linearmente indipendenti, per esempio polinomi algebrici (possibilmente ortogonali) o trigonometrici, oppure funzioni spline, base di uno spazio lineare normato<sup>(†)</sup>  $\mathbb{F}_N \subset C^2[a, b]$ . Proponiamoci di determinare un'approssimazione

$$(8.70) \quad y_N(x) = \sum_{i=0}^N c_i \varphi_i(x) \in \mathbb{F}_N$$

della soluzione  $y(x) \in C^2[a, b]$  del problema (8.69).

---

(†) Ossia dotato di norma.

Per definire la (8.70) sostituiamo la stessa in (8.69) al posto di  $y(x)$  e consideriamo il residuo

$$R_N(x) = y''_N(x) - f(x, y_N(x), y'_N(x)), \quad a \leq x \leq b$$

Fissati poi  $N - 1$  punti distinti  $\{x_n\}$  in  $[a, b]$ , imponiamo al suddetto residuo di annullarsi nei punti  $\{x_n\}$ :

$$(8.71) \quad y''_N(x_n) - f(x_n, y_N(x_n), y'_N(x_n)) = 0, \quad n = 1, 2, \dots, N - 1$$

aggiungiamo inoltre le due condizioni ai limiti

$$(8.72) \quad \begin{aligned} g_1(y_N(a), y_N(b)) &= 0 \\ g_2(y_N(a), y_N(b)) &= 0 \end{aligned}$$

Le (8.71) insieme alle (8.72) costituiscono un sistema di  $N + 1$  equazioni nelle  $N + 1$  incognite  $c_0, c_1, \dots, c_N$ .

Ai fini della definizione della soluzione  $\{c_i\}$  e del condizionamento del sistema, è di particolare importanza la scelta della base  $\{\varphi_i(x)\}$  e dei nodi di collocazione  $\{x_n\}$ . Inoltre, quando lo spazio  $\mathbb{F}_N$  e i punti  $\{x_n\}$  sono scelti “correttamente”, la precisione dell’approssimazione  $y_N(x)$  aumenta al crescere di  $N$ , nel senso che

$$\lim_{N \rightarrow \infty} \|y(x) - y_N(x)\| = 0$$

Concludiamo descrivendo tre particolari esempi di applicazione del metodo di collocazione.

► **Esempio 8.3.** Sia dato il problema

$$\begin{cases} y''(x) - y(x) = x, & 0 \leq x \leq 1 \\ y(0) = 0 \\ y(1) = 0 \end{cases}$$

Come approssimante consideriamo la funzione

$$y_N(x) = \sum_{i=0}^N c_i P_i(x)$$

dove con  $P_i(x)$  denotiamo il polinomio di Legendre di grado  $i$  ortogonale in  $(0, 1)$  (vedi paragrafo 7.3).

Fissati  $N - 1$  punti distinti  $\{x_n\}$  in  $(0, 1)$ , per esempio gli  $N - 1$  zeri di  $P_{N-1}(x)$ , imponendo le condizioni

$$\begin{cases} y_N(0) = 0 \\ y''_N(x_n) - y_N(x_n) = x_n, & n = 1, 2, \dots, N - 1 \\ y_N(1) = 0 \end{cases}$$

perveniamo al sistema

$$\begin{pmatrix} P_0(0) & P_1(0) & P_2(0) & \dots \\ -P_0(x_1) & -P_1(x_1) & [P''_2(x_1) - P_2(x_1)] & \dots \\ -P_0(x_2) & -P_1(x_2) & [P''_2(x_2) - P_2(x_2)] & \dots \\ \dots & \dots & \dots & \dots \\ -P_0(x_{N-1}) & -P_1(x_{N-1}) & [P''_2(x_{N-1}) - P_2(x_{N-1})] & \dots \\ P_0(1) & P_1(1) & P_2(1) & \dots \\ \dots & P_{N-1}(0) & P_N(0) & \\ \dots & [P''_{N-1}(x_1) - P_{N-1}(x_1)] & [P''_N(x_1) - P_N(x_1)] & \\ \dots & [P''_{N-1}(x_2) - P_{N-1}(x_2)] & [P''_N(x_2) - P_N(x_2)] & \\ \dots & \dots & \dots & \dots \\ \dots & P''_{N-1}(x_{N-1}) & [P''_N(x_{N-1}) - P_N(x_{N-1})] & \\ \dots & P_{N-1}(1) & P_N(1) & \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{N-1} \\ c_N \end{pmatrix} = \begin{pmatrix} 0 \\ x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ 0 \end{pmatrix}$$

che, se non singolare, ci consentirà di determinare i coefficienti  $c_0, c_1, c_2, \dots, c_{N-1}, c_N$ . Per il calcolo dei valori  $\{P_k(x_n)\}$  e  $\{P''_k(x_n)\}$  ricordiamo la relazione di ricorrenza di pagina 227. ◀

► **Esempio 8.4.** Dato il problema

$$\begin{cases} y''(x) + y(x) = x^2, & 0 \leq x \leq 1 \\ y(0) = 0 \\ y(1) = 1 \end{cases}$$

scegliamo

$$y_N(x) = \sin\left(\frac{\pi}{2}x\right) + \sum_{i=1}^N c_i \sin(i\pi x)$$

In questo caso le condizioni  $y_N(0) = 0$  e  $y_N(1) = 1$  sono soddisfatte a priori. Scelti pertanto  $N$  nodi distinti  $\{x_n\}$  in  $(0, 1)$ , per esempio

$$x_n = \frac{2n-1}{2N}, \quad n = 1, \dots, N$$

determiniamo i coefficienti  $\{c_i\}$  risolvendo il sistema

$$y''_N(x_n) + y_N(x_n) = x_n^2, \quad n = 1, \dots, N$$



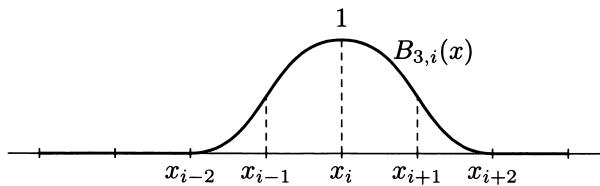
► **Esempio 8.5.** Consideriamo il problema (8.60)

$$\begin{cases} y''(x) - p(x)y'(x) - q(x) = r(x), & a \leq x \leq b \\ y(a) = \alpha \\ y(b) = \beta \end{cases}$$

e come  $y_N(x)$  prendiamo una spline cubica definita dalla partizione  $a \equiv x_0 < x_1 < \dots < x_N \equiv b$ ,  $x_n = a + nh$ ,  $n = 0, 1, \dots, N$ , dell'intervallo di integrazione.

Ricordando l'esercizio 5.16, esprimiamo la nostra  $y_N(x)$  come combinazione lineare delle  $B$ -splines cubiche associate ai nodi  $\{x_n\}$ :

$$y_N(x) = \sum_{i=-1}^{N+1} c_i B_{3,i}(x)$$



**Figura 8.16**

Successivamente imponiamo le condizioni

$$(8.73) \quad \begin{cases} y_N(x_0) = \alpha \\ y''_N(x_n) - p(x_n)y'_N(x_n) - q(x_n)y_N(x_n) = r(x_n), & n = 0, 1, \dots, N \\ y_N(x_N) = \beta \end{cases}$$

Poiché ogni singola  $B$ -spline è diversa da zero in tre soli nodi, la (8.73) si riduce ad un sistema tridiagonale nelle incognite  $\{c_i\}$ . Supponendo  $p(x)$ ,  $q(x)$ ,  $r(x) \in C[a, b]$ ,  $q(x)$  positivo in  $[a, b]$  e  $y(x) \in C^4[a, b]$ , è possibile dimostrare (vedi ad esempio [8.6, §8.5]) che per  $h$  sufficientemente piccolo il sistema risulta non singolare e  $\|y(x) - y_N(x)\|_\infty = O(h^2)$ .

◀

## Bibliografia

- [8.1] E. A. Coddington, N. Levinson N., *Theory of ordinary differential equations*, McGraw-Hill, New York, 1955.
- [8.2] P. Henrici, *Discrete variable methods in ordinary differential equations*, John Wiley & Sons, New York, 1962.
- [8.3] H. B. Keller, *Numerical methods for two-point boundary value problems*, Ginn-Blaisdell Walthman, Mass., 1968.
- [8.4] W. C. Gear, *Numerical initial value problems in ordinary differential equations*, Prentice-Hall, Englewood Cliffs, N. J., 1971.
- [8.5] L. Shampine, M. Gordon, *Computer solution of ordinary differential equations*, Freeman, San Francisco, 1974.
- [8.6] P. M. Prenter, *Splines and variational methods*, John Wiley & Sons, New York, 1975.

- [8.7] H. B. Keller, *Numerical solution of two-point boundary value problems*, SIAM regional conference series in applied mathematics, n. 24, Philadelphia, 1976.
- [8.8] G. Hall, J. M. Watt, Eds., *Modern numerical methods for ordinary differential equations*, Clarendon Press, Oxford, 1976.
- [8.9] L. Lapidus, W. E. Schiesser, Eds., *Numerical methods for differential systems*, Academie Press, New York, 1976.
- [8.10] M. Bozzini, M. Macconi, A. Pasquali, *Risoluzione numerica di equazioni differenziali ordinarie*, Quaderni U.M.I., n. 3, Pitagora Editrice, Bologna, 1977.
- [8.11] L. C. Piccinini, G. Stampacchia, G. Vidossich, *Equazioni differenziali ordinarie in  $\mathbb{R}^n$  (problemi e metodi)*, Liguori Editore, Napoli, 1978.
- [8.12] I. Gladwell, D. K. Sayers, Eds., *Computational techniques for ordinary differential equations*, Academic Press, London, 1980.
- [8.13] W. L. Miranker, *Numerical methods for stiff differential equations and singular perturbation problems*, D. Reidel Publ. Co., Dordrecht, 1981.
- [8.14] R. C. Aiken, Ed., *Stiff computation*, Oxford University Press, Oxford, 1985.
- [8.15] S. K. Godunov, V. S. Ryabenkii, *Difference schemes*, North-Holland, Amsterdam, 1987.
- [8.16] E. Hairer, S. P. Norsett, G. Wanner, *Solving ordinary differential equations I*, Springer-Verlag, Heidelberg, 1987.
- [8.17] V. Lakshmikantham, D. Trigiante, *Theory of difference equations – numerical methods and applications*, Academic Press, New York, 1988.
- [8.18] U. M. Ascher, R. M. M. Mattheij, R. D. Russell, *Numerical solution of boundary value problems for ordinary differential equations*, Prentice-Hall, Englewood Cliffs, New Jersey, 1988.
- [8.19] J. D. Lambert, *Numerical methods for ordinary differential systems*, John Wiley & Sons, New York, 1991.
- [8.20] E. Hairer, G. Wanner, *Solving ordinary differential equations II*, Springer-Verlag, Berlino, 1991.
- [8.21] J. C. Butcher, *Numerical methods for ordinary differential equations*, John Wiley & Sons, Chichester, 2008.

## Esercizi proposti

**8.1.** Trasformare i problemi seguenti in sistemi di equazioni differenziali del primo ordine:

- (a)  $y'' - 3y' + 2y = 0, \quad y(0) = 1, \quad y'(0) = 1$
- (b)  $y'' - 0.1(1 - y^2)y' + y = 0, \quad y(0) = 1, \quad y'(0) = 0$
- (c) 
$$\begin{cases} x''(t) = -\frac{x(t)}{r^3}, & x(0) = 0.5, \quad x'(0) = 0 \\ y''(t) = -\frac{y(t)}{r^3}, & y(0) = 0, \quad y'(0) = 1 \end{cases}$$

dove  $r = \sqrt{[x(t)]^2 + [y(t)]^2}$ .

**8.2.** Trasformare l'equazione di Sturm-Liouville

$$\frac{d}{dx}[p(x)y'(x)] + q(x)y(x) = 0, \quad p(x) \neq 0$$

in un sistema di equazioni del primo ordine che non coinvolgano la derivata di  $p(x)$ .

**8.3.** Disegnare nel piano  $(x, y)$  la curva

$$\Gamma = \{ P(t) \equiv (x(t), y(t)), \quad 0 \leq t \leq 2\pi \}$$

definita dal sistema

$$\begin{cases} x'(t) = -y(t), & x(0) = 1 \\ y'(t) = x(t), & y(0) = 1 \end{cases}$$

**8.4.** Applicare il metodo di Eulero (esplicito) al problema

$$\begin{cases} y' = y^{1/3}, & x \geq 0 \\ y(0) = 0 \end{cases}$$

e motivare il risultato ottenuto.

**8.5.** Verificare che un metodo Runge-Kutta a due stadi non può avere ordine  $p = 3$ .

**8.6.** Risolvere il problema

$$\begin{aligned} y'' &= 0.1(1 - y^2)y' - y \\ y(0) &= 1 \\ y'(0) &= 0 \end{aligned}$$

utilizzando il metodo di Eulero.

**8.7.** Risolvere il problema dell'esercizio precedente con il metodo di Heun.

**8.8.** Applicare la formula di Heun (8.23) ad un sistema di  $m$  equazioni differenziali del primo ordine. Come deve essere interpretata la (8.23)?

**8.9.** Costruire una routine automatica per integrare problemi di tipo (8.15), utilizzando le formule di Fehlberg e scegliendo il passo ottimale  $h_n$  in modo che l'errore locale unitario di troncamento stimato sia inferiore ad una tolleranza prestabilita.

**8.10.** Costruire le formule di Adams-Bashforth e di Adams-Moulton a due passi, determinando esplicitamente i coefficienti  $\{\beta_i\}$  e  $\{\beta_i^*\}$ .

**8.11.** Verificare che i metodi di Adams-Bashforth e di Adams-Moulton a  $k$  passi hanno *grado di precisione* rispettivamente  $p = k$  e  $p = k + 1$ , nel senso che risultano esatti ognqualvolta la soluzione  $y(x)$  è un polinomio di grado  $\leq p$ .

**8.12.** Data la formula dei trapezi per il calcolo di integrali, ricavare la nota formula dei trapezi per la risoluzione di equazioni differenziali ordinarie.

**8.13.** Dimostrare che il metodo

$$y_{n+2} = y_n + \frac{h}{3}[f(x_n, y_n) + 4f(x_{n+1}, y_{n+1}) + f(x_{n+2}, y_{n+2})]$$

ottenibile applicando la formula di Simpson all'integrale

$$\int_{x_n}^{x_{n+2}} y'(x) dx = y(x_{n+2}) - y(x_n)$$

risulta convergente.

**8.14.** Volendo applicare il metodo di Eulero (esplicito) all'equazione

$$y' = -10^3 y$$

come dobbiamo scegliere il passo di integrazione  $h$  affinché il metodo risulti assolutamente stabile?

**8.15.** Trovare la regione di stabilità assoluta del metodo previsore-correttore formato dalla coppia Eulero esplicito-trapezi.

**8.16.** Determinare esplicitamente i coefficienti della formula BDF a due passi definita a pagina 290

**8.17.** Integrare il seguente sistema non lineare

$$\begin{cases} y' = 0.01 - (0.01 + y + z)[1 + (y + 1000)](y + 1) \\ z' = 0.01 - (0.1 + y + z)(1 + z^2) \\ y(0) = z(0) = 0 \end{cases}$$

nell'intervallo  $[0, 100]$ . Gli autovalori della matrice jacobiana nel punto  $x = 0$  sono  $\lambda_1 = -1012$  e  $\lambda_2 = -0.01$ , mentre in  $x = 100$  abbiamo  $\lambda_1 = -21.7$  e  $\lambda_2 = -0.089$ .

**8.18.** Integrare il sistema

$$\begin{cases} y'' + y = x, & 0 \leq x \leq \frac{\pi}{2} \\ y(0) = 1 \\ y\left(\frac{\pi}{2}\right) = \frac{\pi}{2} - 1 \end{cases}$$

con il metodo delle differenze finite, suddividendo dapprima l'intervallo  $[0, \pi/2]$  in  $N = 5$  parti uguali e successivamente in  $N = 10$  parti.

**8.19.** Utilizzare il metodo di collocazione per determinare una spline cubica che approssimi la soluzione del seguente problema:

$$\begin{cases} y'' - (e^x + 1)y = \cos(2\pi x), & 0 \leq x \leq 1 \\ y(0) = y(1) = 1 \end{cases}$$

**8.20.** Risolvere i due problemi precedenti con il metodo shooting.

# Capitolo 9

## Equazioni alle derivate parziali

### 9.1 Preliminari

Un'equazione alle derivate parziali è una relazione del tipo

$$(9.1) \quad F(x, y, \dots, u, u_x, u_y, u_{xx}, u_{xy}, u_{yy}, \dots) = 0$$

tra le variabili indipendenti  $x, y, \dots$ , la funzione incognita  $u(x, y, \dots)$  e un numero finito di sue derivate parziali  $u_x = \partial u(x, y, \dots)/\partial x$ ,  $u_y = \partial u(x, y, \dots)/\partial y$ ,  $\dots$ ,  $u_{xx} = \partial^2 u(x, y, \dots)/\partial x^2$ ,  $u_{xy} = \partial^2 u(x, y, \dots)/(\partial x \partial y)$ ,  $\dots$ . Quando la derivata (parziale) di ordine più elevato presente in (9.1) ha ordine  $m$  diciamo che l'equazione è di ordine  $m$ .

Una funzione  $u(x, y, \dots)$  è detta soluzione *classica* della (9.1) (di ordine  $m$ ) in una regione aperta  $\mathcal{R}$  dello spazio delle variabili indipendenti  $x, y, \dots$  se risulta ivi continua con tutte le sue derivate parziali (di ordine  $\leq m$ ) presenti nell'equazione e soddisfa la stessa in tutti i punti di  $\mathcal{R}$ . Tuttavia talvolta le derivate, e quindi l'equazione, non sono definite puntualmente; esse vanno intese nel senso delle distribuzioni e la corrispondente soluzione viene denominata *soluzione debole*. Quest'ultima viene quindi caratterizzata da una formulazione integrale associata alla (9.1), che coinvolge derivate di ordine inferiore a  $m$  definite nel senso delle distribuzioni (vedi [9.25] e il paragrafo 9.5). In molti problemi la formulazione più corretta è proprio quest'ultima, che viene denominata *debole*. In questo testo, tuttavia, per semplicità considereremo solo problemi le cui formulazioni (puntuali o *forti*) di tipo (9.1) ammettono soluzioni classiche. La formulazione debole, che in questo caso risulta del tutto equivalente a quella forte, verrà introdotta nel paragrafo 9.5 solo per giustificare la costruzione del metodo degli elementi finiti.

La (9.1) (di ordine  $m$ ) è detta *lineare* se la funzione  $F$  è lineare nella incognita  $u$  e nelle sue derivate parziali, con coefficienti dipendenti unicamente dalle variabili indipendenti  $x, y, \dots$ ; altrimenti l'equazione è definita *non lineare*. In quest'ultimo caso, essa viene a sua volta denominata *quasi-lineare* se risulta lineare nelle derivate di ordine  $m$ , con coefficienti dipendenti al più da  $x, y, \dots$  e da  $u$  e le sue derivate di ordine  $\leq m - 1$ . Per

esempio, l'equazione

$$u_y = du_{xx} - vu_x + au + f$$

dove, oltre all'incognita  $u$ , anche i coefficienti (noti)  $d, v$  e  $a$ , nonché il termine noto  $f$ , sono funzioni solo delle variabili  $x$  e  $y$ , è lineare del secondo ordine, mentre

$$u_y u_{xx} - u_x^2 - u_y^2 + u = 1$$

è quasi-lineare del secondo ordine.

I problemi fisici descritti da equazioni alle derivate parziali si collocano in modo del tutto naturale in due classi distinte (<sup>†</sup>):

- (i) Problemi di propagazione (o non stazionari).
- (ii) Problemi di “equilibrio” (o stazionari) e problemi agli autovalori.

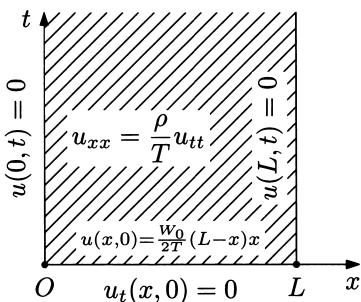
Esaminiamo brevemente alcune loro peculiarità.

I problemi di propagazione sono *problemI a valori iniziali*, detti anche *di Cauchy*, che rappresentano un fenomeno (non stazionario) in evoluzione. Assegnati i dati iniziali (all'istante  $t = 0$ ), si desidera determinare il comportamento del fenomeno in esame negli istanti successivi ( $t > 0$ ). Il modello matematico in questione è costituito da: una o più equazioni differenziali definite in un dominio spaziale (aperto)  $D$  per ogni  $t > 0$ , dalle equazioni che descrivono lo stato iniziale, e da eventuali *condizioni al bordo* assegnate sul contorno  $\Gamma$  di  $D$ . La soluzione  $u = u(t, x, \dots)$  dipende dalla variabile “tempo” e da una o più variabili spaziali.

Tipici esempi di problemI a valori iniziali sono: propagazione delle onde di pressione in un fluido, propagazione di tensioni e spostamenti in sistemi elastici, propagazione del calore in un mezzo.

Consideriamo una corda elastica di lunghezza (a riposo)  $L$ , densità  $\rho$ , tesa con tensione  $T$  e fissata ai suoi estremi 0 e  $L$ . All'istante  $t = 0$  perturbiamo la configurazione di equilibrio, per esempio facendo assumere alla corda la posizione  $\frac{W_0}{2T}(L-x)x$ . Con  $u(x, t)$  denotiamo lo spostamento verticale all'istante  $t$  del punto  $x$  della corda dalla posizione orizzontale. La configurazione  $u(x, t)$  agli istanti  $t > 0$  risulta, in prima approssimazione, soluzione del seguente problema:

$$(9.2) \quad \begin{cases} \frac{\partial^2 u}{\partial x^2} - \frac{\rho}{T} \frac{\partial^2 u}{\partial t^2} = 0, & 0 < x < L, \quad t > 0 \\ u(x, 0) = \frac{W_0}{2T}(L-x)x \\ u_t(x, 0) = 0 \\ u(0, t) = u(L, t) = 0, & t > 0 \end{cases} \quad 0 \leq x \leq L$$

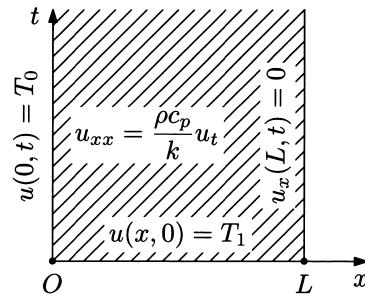


(<sup>†</sup>) Tale suddivisione vale, ovviamente, anche per i problemI rappresentati da equazioni differenziali ordinarie.

Consideriamo infine il problema della conduzione di calore in un filo metallico omogeneo di lunghezza  $L$ , densità  $\rho$ , calore specifico  $c_p$ , conduttività termica  $k$ , termicamente isolato lungo la sua lunghezza e all'estremo  $L$ . Supponiamo che inizialmente tutto il filo si trovi a temperatura  $T_1$ . Successivamente l'estremo 0 viene immerso in un mezzo a temperatura  $T_0 \ll T_1$ . La temperatura  $u(x, t)$  nel generico punto  $x$  del filo, all'istante  $t > 0$ , è descritta dal seguente modello matematico:

(9.3)

$$\begin{cases} \frac{\partial^2 u}{\partial x^2} - \frac{\rho c_p}{k} \frac{\partial u}{\partial t} = 0, & 0 < x < L, \quad t > 0 \\ u(x, 0) = T_1, & 0 \leq x \leq L \\ u(0, t) = T_0, & t > 0 \\ u_x(L, t) = 0, & t > 0 \end{cases}$$



I problemi di equilibrio sono problemi stazionari (ovvero indipendenti dal tempo) nei quali la configurazione di equilibrio  $u = u(x, y, \dots)$  nel dominio di interesse  $D$  viene descritta da una o più equazioni differenziali, definite in  $D$ , e da condizioni (su  $u$ ) assegnate sul bordo di  $D$ . Essi vengono generalmente denominati *problem con valori al contorno*. Spesso tali problemi si presentano nello studio della configurazione a regime di fenomeni di evoluzione (dipendenti quindi dal tempo).

Tra i fenomeni fisici che danno origine a problemi di questo tipo ricordiamo: il flusso viscoso stazionario, la distribuzione stazionaria di temperature in un mezzo, l'equilibrio di tensioni in strutture elastiche.

Consideriamo un supporto rigido chiuso  $F$  nel piano (orizzontale)  $(x, y)$ . Supponiamo che tale supporto costituisca il contorno di una membrana elastica ideale, di densità uniforme, sottoposta a tensione uniforme  $T$  lungo il contorno e ad una pressione verticale uniforme  $P$ . Con  $(x, y, u(x, y))$  denotiamo le coordinate del generico punto della membrana nella sua configurazione di equilibrio. Facendo delle opportune ipotesi è possibile dimostrare che l'ordinata  $u(x, y)$  è, nel dominio  $D$  del piano  $(x, y)$  delimitato dalla curva  $\Gamma$ , soluzione dell'*equazione di Poisson*

$$-\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = \frac{P}{T}$$

La configurazione di equilibrio  $u(x, y)$  risulta pertanto definita dal seguente problema al contorno:

$$(9.4) \quad \begin{cases} -\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = \frac{P}{T} & \text{in } D \\ u = 0 & \text{su } \Gamma \end{cases}$$

L'intuizione ci "assicura" che i problemi (9.2), (9.3) e (9.4) hanno un'unica soluzione; è tuttavia possibile dimostrare che essi sono addirittura *ben posti*, ossia che (i) la loro soluzione  $u$  esiste ed è unica, e (ii) dipende con continuità dai dati iniziali e al contorno.

I problemi agli autovalori possono essere pensati come estensioni di quelli di equilibrio, laddove valori critici di un parametro (autovalori) devono essere determinati, insieme alle corrispondenti configurazioni stazionarie. Problemi agli autovalori si presentano nello studio di deformazioni e stabilità di strutture, di fenomeni di risonanza in circuiti elettrici e acustici, nella ricerca di frequenze naturali in problemi di vibrazioni.

Il problema agli autovalori più semplice è quello della membrana vibrante (vedi ad esempio [9.3]), descritto dalle equazioni

$$(9.5) \quad \begin{cases} -\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = \lambda u & \text{in } D \\ u = 0 & \text{su } \Gamma \end{cases}$$

Esso consiste nel determinare valori numerici  $\lambda$  ai quali corrispondono autosoluzioni (quindi non identicamente nulle)  $u = u(x, y)$  del sistema (9.5), nonché queste ultime.

Come per le equazioni differenziali ordinarie, anche nel caso di quelle alle derivate parziali possiamo cercare una rappresentazione analitica dell'integrale generale. Purtroppo tale soluzione può essere individuata in casi molto rari; e anche quando ciò avviene, spesso non è di alcun aiuto per determinare l'integrale particolare che soddisfa le condizioni (iniziali e al contorno) imposte. Infatti l'integrale generale coinvolge funzioni arbitrarie, e non più costanti arbitrarie come nel caso delle equazioni differenziali ordinarie. Consideriamo per esempio l'*equazione di Laplace*

$$\Delta u = 0, \quad \text{ovvero} \quad \nabla^2 u = 0, \quad \text{ovvero} \quad \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

È noto che la parte reale di una qualsiasi funzione analitica

$$f(z) = u(x, y) + i v(x, y), \quad z = x + iy$$

è soluzione dell'equazione di Laplace; e viceversa, ogni soluzione dell'equazione di Laplace è parte reale di una funzione analitica<sup>(†)</sup>.

Le equazioni alle derivate parziali costituiscono uno dei capitoli più interessanti e vasti della Matematica sia pura che applicata. In questo corso introduttivo al Calcolo Numerico noi ci limiteremo ai problemi lineari di ordine al più due ed a una breve presentazione di alcuni degli approcci numerici più noti. La letteratura sull'argomento è vastissima. Nella bibliografia di fine capitolo abbiamo riportato un elenco di alcuni dei testi più significativi, la cui lettura consigliamo a coloro che desiderano approfondire e ampliare quanto da noi esposto in questo capitolo.

---

<sup>(†)</sup> Ricordiamo che le soluzioni dell'equazione di Laplace vengono chiamate *funzioni armoniche*.

## 9.2 Caratteristiche. Classificazione delle equazioni quasi-lineari di ordine 2

Consideriamo dapprima la generica equazione quasi-lineare del 1° ordine

$$(9.6) \quad au_x + bu_y = c$$

nell'incognita  $u(x, y)$ . Supponiamo di conoscere la soluzione  $u(x, y)$  su una curva prefissata  $\gamma$  del dominio di definizione della (9.6), di equazioni parametriche  $x = \varphi(\tau)$  e  $y = \psi(\tau)$ ,  $\tau \in I$ , entrambe di classe  $C^1(I)$ :  $u(\varphi(\tau), \psi(\tau)) = f(\tau)$ . Su tale curva sono allora noti i coefficienti  $a \equiv a(\varphi(\tau), \psi(\tau), f(\tau))$ ,  $b \equiv b(\varphi(\tau), \psi(\tau), f(\tau))$ ,  $c \equiv c(\varphi(\tau), \psi(\tau), f(\tau))$  e la derivata

$$\frac{d}{d\tau} u(x, y) \equiv u_x \frac{dx}{d\tau} + u_y \frac{dy}{d\tau} = f'(\tau)$$

Poniamoci ora la seguente domanda: la conoscenza di  $u$ , e quindi di  $du/d\tau$ , sulla curva  $\gamma$  è sufficiente per definire (sempre su  $\gamma$ ) in modo univoco le derivate parziali  $u_x = u_x(\varphi(\tau), \psi(\tau))$  e  $u_y = u_y(\varphi(\tau), \psi(\tau))$ ? La questione è di fondamentale importanza in quanto la definizione stessa di soluzione (classica) richiede l'esistenza (e quindi l'unicità) delle derivate prime sulla curva  $\gamma$ . In altri termini, quand'è che le seguenti due condizioni

$$\begin{cases} au_x + bu_y = c \\ \frac{dx}{d\tau} u_x + \frac{dy}{d\tau} u_y = f'(\tau) \end{cases}$$

sono sufficienti per garantire l'esistenza e unicità di  $u_x$  e  $u_y$  su  $\gamma$ ? Per rispondere a tale quesito è sufficiente esaminare il determinante

$$\left| \begin{array}{cc} a & b \\ \frac{dx}{d\tau} & \frac{dy}{d\tau} \end{array} \right|$$

La soluzione  $u_x$ ,  $u_y$  del precedente sistema esiste ed è unica se e solo se

$$\frac{dy}{dx} \neq \frac{b}{a}$$

cioè quando il coefficiente angolare della tangente alla curva  $\gamma$  nel generico punto  $(\varphi(\tau), \psi(\tau))$  è diverso da  $b/a$ . Poiché la *direzione caratteristica*  $dy/dx$  che rende singolare il precedente sistema è unica e reale, l'equazione (del 1° ordine) (9.6) viene definita *iperbolica*.

Le curve  $\gamma$  del piano  $(x, y)$  che hanno in ogni loro punto il coefficiente angolare della retta tangente uguale a  $b/a$  vengono denominate *curve caratteristiche* dell'equazione (9.6). Si ricorda (vedi il Teorema 8.1 del capitolo precedente) che quando  $b/a$  è una funzione continua dei suoi argomenti, l'equazione differenziale  $dy/dx = b/a$  ammette sempre soluzione. In particolare, quando le condizioni (sufficienti) richieste dal teorema

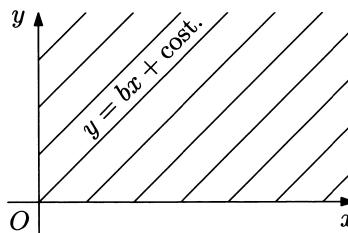
8.1 sono soddisfatte, per ogni punto del dominio di definizione della (9.6) passerà un'unica curva caratteristica.

► **Esempio.** Consideriamo il seguente problema

$$\begin{cases} u_x + bu_y = -ku, & x > 0, \quad y > 0 \\ u(x, 0) = u_0, & x > 0 \\ u(0, y) = 0, & y \geq 0 \end{cases}$$

dove  $b, k, u_0$  sono costanti assegnate. Esso ha una famiglia di infinite curve caratteristiche di equazione

$$y = bx + \text{costante}$$



**Figura 9.1**

L'esame del precedente problema di Cauchy associato alla (9.6) può venire facilmente esteso ad un sistema quasi-lineare di due equazioni del 1° ordine

$$(9.7) \quad \begin{cases} a_1u_x + b_1u_y + c_1v_x + d_1v_y = f_1 \\ a_2u_x + b_2u_y + c_2v_x + d_2v_y = f_2 \end{cases}$$

nelle incognite  $u = u(x, y)$  e  $v = v(x, y)$ .

Supponendo di conoscere su  $\gamma$  le funzioni  $u(x, y)$  e  $v(x, y)$ , e quindi le corrispondenti derivate

$$\begin{aligned} \frac{d}{d\tau}u(x, y) &= u_x \frac{dx}{d\tau} + u_y \frac{dy}{d\tau} \\ \frac{d}{d\tau}v(x, y) &= v_x \frac{dx}{d\tau} + v_y \frac{dy}{d\tau} \end{aligned}$$

perveniamo al seguente sistema:

$$\begin{pmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ \frac{dx}{d\tau} & \frac{dy}{d\tau} & 0 & 0 \\ 0 & 0 & \frac{dx}{d\tau} & \frac{dy}{d\tau} \end{pmatrix} \begin{pmatrix} u_x \\ u_y \\ v_x \\ v_y \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \frac{du}{d\tau} \\ \frac{dv}{d\tau} \end{pmatrix}$$

la cui soluzione esiste ed è unica solo quando

$$(a_1c_2 - a_2c_1) \left( \frac{dy}{d\tau} \right)^2 - (a_1d_2 - a_2d_1 + b_1c_2 - b_2c_1) \frac{dy}{d\tau} \frac{dx}{d\tau} + (b_1d_2 - b_2d_1) \left( \frac{dx}{d\tau} \right)^2 \neq 0$$

Volendo cercare le *direzioni caratteristiche*  $dy/dx$  che rendono singolare il sistema, è sufficiente studiare le radici dell'*equazione caratteristica*

$$(9.8) \quad (a_1c_2 - a_2c_1) \left( \frac{dy}{dx} \right)^2 - (a_1d_2 - a_2d_1 + b_1c_2 - b_2c_1) \frac{dy}{dx} + (b_1d_2 - b_2d_1) = 0$$

In particolare occorre esaminare il segno del discriminante

$$\mathcal{D} = (a_1d_2 - a_2d_1 + b_1c_2 - b_2c_1)^2 - 4(a_1c_2 - a_2c_1)(b_1d_2 - b_2d_1)$$

Per ogni punto della regione del piano  $(x, y)$  in cui  $\mathcal{D}$  è positivo abbiamo due direzioni caratteristiche reali e distinte, e il sistema è ivi denominato *iperbolico*; quando  $\mathcal{D} = 0$  le due direzioni caratteristiche risultano reali e coincidenti, e il sistema è detto *parabolico*; infine, quando  $\mathcal{D} < 0$  le direzioni caratteristiche sono complesse coniugate e il sistema è denominato *ellittico*.

Una curva  $\gamma$  del piano  $(x, y)$  è detta *caratteristica* del sistema quando la tangente in ogni suo punto soddisfa l'equazione (9.8). Per individuare le curve caratteristiche di un sistema è sufficiente determinare le radici  $dy/dx$  dell'equazione (9.8) e integrare le corrispondenti equazioni differenziali (del 1° ordine).

► **Esempio.** Consideriamo il seguente sistema

$$\begin{cases} u_x - v_y = 0 \\ v_x - u_y = 0 \end{cases}$$

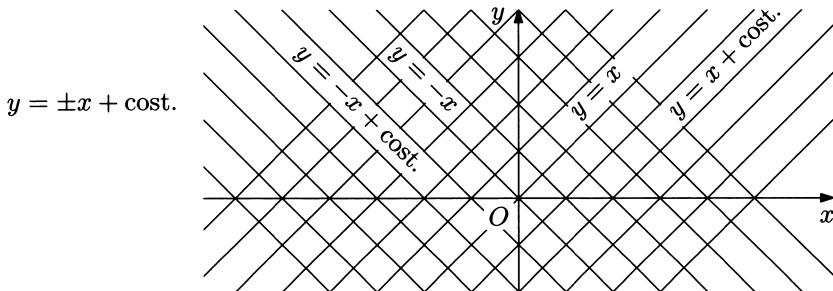


Figura 9.2

L'equazione caratteristica associata risulta

$$\left(\frac{dy}{dx}\right)^2 - 1 = 0$$

e le direzioni caratteristiche sono  $dy/dx = \pm 1$ . Il sistema è pertanto di tipo iperbolico ed ammette, nel suo dominio di definizione, due famiglie di curve caratteristiche reali e distinte (figura 9.2). In particolare, per ogni punto del dominio passano due curve (rette in questo caso) caratteristiche reali e distinte.

Consideriamo infine un'equazione quasi-lineare del 2° ordine

$$(9.9) \quad au_{xx} + bu_{xy} + cu_{yy} = f$$

Ci chiediamo sotto quali condizioni la conoscenza della  $u$  lungo una curva  $\gamma = \{(\varphi(\tau), \psi(\tau)) : \tau \in I\}$ ,  $\varphi, \psi \in C^2(I)$ , della regione di definizione  $\mathcal{R}$  della (9.9), e della derivata parziale di  $u$  secondo la normale alla curva  $\partial u / \partial n_\gamma$  (vedi pag. 320) permetta di definire univocamente, sempre su  $\gamma$ , le derivate seconde  $u_{xx}$ ,  $u_{xy}$  e  $u_{yy}$ . Si noti che dalla conoscenza (su  $\gamma$ ) di  $u$  e  $\partial u / \partial n_\gamma$  si deducono entrambe le derivate parziali prime  $\partial u / \partial x$  e  $\partial u / \partial y$  lungo la curva; quindi possiamo supporre note le derivate tangenziali  $du_x / d\tau$  e  $du_y / d\tau$ . Pertanto la (9.9), insieme con i dati iniziali  $u$  e  $\partial u / \partial n_\gamma$ , definirà univocamente le derivate seconde  $u_{xx}$ ,  $u_{xy}$  e  $u_{yy}$  se e solo se il sistema

$$\begin{pmatrix} a & b & c \\ \frac{dx}{d\tau} & \frac{dy}{d\tau} & 0 \\ 0 & \frac{dx}{d\tau} & \frac{dy}{d\tau} \end{pmatrix} \begin{pmatrix} u_{xx} \\ u_{xy} \\ u_{yy} \end{pmatrix} = \begin{pmatrix} f \\ \frac{du_x}{d\tau} \\ \frac{du_y}{d\tau} \end{pmatrix}$$

risulta non singolare, ossia

$$(9.10) \quad a \left( \frac{dy}{dx} \right)^2 - b \frac{dy}{dx} + c \neq 0$$

Quando in ogni punto di  $\gamma$  risulta  $b^2 - 4ac > 0$ , abbiamo due direzioni caratteristiche reali e distinte (equazione iperbolica); quando  $b^2 - 4ac = 0$  abbiamo due direzioni caratteristiche reali e coincidenti (equazione parabolica); infine, quando  $b^2 - 4ac < 0$  le due direzioni sono complesse coniugate (equazione ellittica).

In base alle definizioni che abbiamo appena dato, l'equazione della corda vibrante (9.2) è di tipo iperbolico, quella del calore (9.3) è di tipo parabolico, mentre l'equazione della membrana elastica (9.4) risulta ellittica. Osserviamo tuttavia che un'equazione potrebbe non essere dello stesso tipo su tutto il dominio di interesse. Per esempio, l'equazione

$$yu_{xx} + u_{yy} = 0$$

è iperbolica in  $y < 0$ , parabolica in  $y = 0$ , ed ellittica in  $y > 0$ .

L'equazione (9.9) può essere posta nella forma di sistema di due equazioni del primo ordine. Per esempio, ponendo in (9.9)  $w = u_x$  e  $v = u_y$  otteniamo

$$(9.11) \quad \begin{cases} aw_x + bw_y + cv_y = f \\ w_y - v_x = 0 \end{cases}$$

mentre con  $w = u_x$  e  $v = u_x + u_y$ abbiamo

$$(9.12) \quad \begin{cases} aw_x + (b - c)w_y + cv_y = f \\ w_y - v_x - w_x = 0 \end{cases}$$

Alcune forme possono risultare più convenienti di altre, soprattutto dal punto di vista dell'approccio numerico. Ovviamente, i problemi (9.9), (9.11) e (9.12) sono tutti dello stesso tipo.

▷ **Osservazione 1.** Le curve caratteristiche di un'equazione sono determinabili a priori, eventualmente ricorrendo a metodi numerici, solo quando l'equazione è lineare. Nel caso non lineare la costruzione di tali curve deve procedere in parallelo con la risoluzione del problema alle derivate parziali in esame. ◁

▷ **Osservazione 2.** Il concetto di caratteristica e la conseguente classificazione possono essere estese a equazioni e sistemi di ordine superiore a 2, in due o più variabili indipendenti. ◁

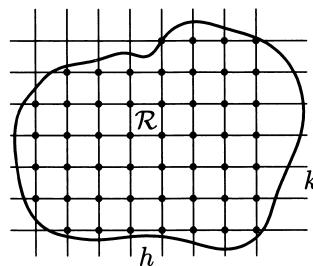
Le linee caratteristiche assumono un ruolo fondamentale sia nella teoria delle equazioni alle derivate parziali che nella costruzione di metodi numerici per la risoluzione di queste ultime.

### 9.3 Metodi alle differenze finite

#### 9.3.1 Generalità

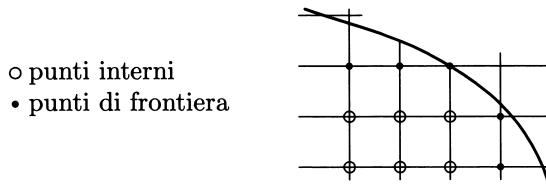
Nei paragrafi successivi ci proponiamo di approssimare i modelli continui, costituiti da equazioni differenziali, con modelli discreti che coinvolgano solamente approssimazioni della soluzione del modello continuo in un numero finito di punti della regione di interesse  $\mathcal{R}$  e possano essere risolti in modo “efficiente” con l’ausilio del calcolatore.

L’idea base consiste nel sostituire la regione  $\mathcal{R}$  con un *reticolo* (rettangolare) di punti di  $\mathcal{R}$ , “collocare” il sistema differenziale sui nodi del reticolo, e approssimare (nei nodi) le derivate parziali con formule (alle differenze finite) di derivazione numerica del tutto simili a quelle introdotte a pagina 174.



**Figura 9.3**

Un punto del reticolo è detto *interno* se i suoi 4 nodi “vicini” sono nella regione  $\mathcal{R}$ , contorno incluso. Gli altri punti appartenenti a  $\mathcal{R}$ , ma non interni, vengono chiamati punti di *frontiera* del reticolo.



**Figura 9.4**

Richiamiamo brevemente alcune formule di derivazione numerica, deducibili direttamente dalle corrispondenti per le derivate ordinarie presentate a pagina 174, i cui errori hanno i comportamenti descritti solo quando la funzione  $u$  è sufficientemente regolare<sup>(†)</sup> (figura 9.5).

(†) Occorre osservare l’ordine della derivata presente nella rappresentazione dell’errore.

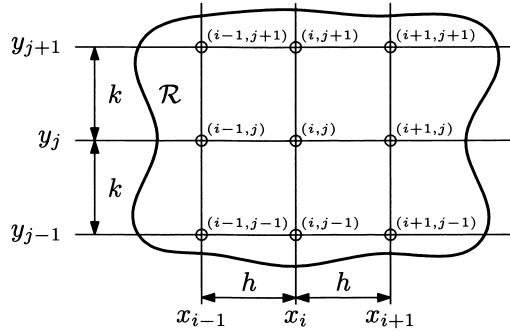
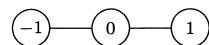
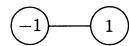
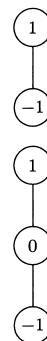


Figura 9.5

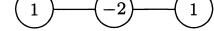
$$\begin{aligned}
 \frac{\partial u}{\partial x}(x_i, y_j) &= \frac{u(x_{i+1}, y_j) - u(x_i, y_j)}{h} + O(h) \\
 &= \frac{u(x_i, y_j) - u(x_{i-1}, y_j)}{h} + O(h) \\
 &= \frac{u(x_{i+1}, y_j) - u(x_{i-1}, y_j)}{2h} + O(h^2)
 \end{aligned}$$



$$\begin{aligned}
 \frac{\partial u}{\partial y}(x_i, y_j) &= \frac{u(x_i, y_{j+1}) - u(x_i, y_j)}{k} + O(k) \\
 &= \frac{u(x_i, y_j) - u(x_i, y_{j-1})}{k} + O(k) \\
 &= \frac{u(x_i, y_{j+1}) - u(x_i, y_{j-1})}{2k} + O(k^2)
 \end{aligned}$$



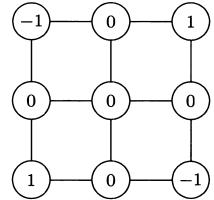
$$\frac{\partial^2 u}{\partial x^2}(x_i, y_j) = \frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j)}{h^2} + O(h^2)$$



$$\frac{\partial^2 u}{\partial y^2}(x_i, y_j) = \frac{u(x_i, y_{j+1}) - 2u(x_i, y_j) + u(x_i, y_{j-1})}{k^2} + O(k^2)$$



$$\begin{aligned}\frac{\partial^2 u(x_i, y_j)}{\partial x \partial y} &= \frac{u(x_{i+1}, y_{j+1}) - u(x_{i-1}, y_{j+1}) - u(x_{i+1}, y_{j-1})}{4hk} \\ &\quad + \frac{u(x_{i-1}, y_{j-1})}{4hk} + O((h+k)^2)\end{aligned}$$



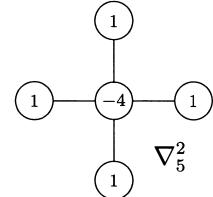
Supponiamo, per esempio,  $h = k$  e approssimiamo gli operatori

$$\nabla = \left( \frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right)^T \quad (\text{gradiente})$$

$$\Delta = \nabla^T \nabla = \nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \quad (\text{laplaciano})$$

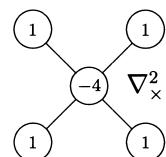
ricorrendo alle sudette formule di derivazione numerica; otteniamo

$$\begin{aligned}\nabla^2(u(x_i, y_j)) &\approx \frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j)}{h^2} \\ &\quad + \frac{u(x_i, y_{j+1}) - 2u(x_i, y_j) + u(x_i, y_{j-1})}{h^2} \\ &= \frac{u(x_{i+1}, y_j) + u(x_{i-1}, y_j) + u(x_i, y_{j+1}) + u(x_i, y_{j-1})}{h^2} \\ &\quad - 4 \frac{u(x_i, y_j)}{h^2} \\ &\equiv \nabla_5^2 u(x_i, y_j)\end{aligned}$$



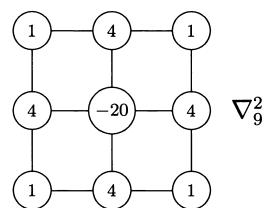
Un'altra formula per  $\nabla^2 u(x_i, y_j)$  potrebbe essere la seguente:

$$\begin{aligned}\nabla^2 u(x_i, y_j) &\approx \frac{u(x_{i+1}, y_{j+1}) + u(x_{i-1}, y_{j-1})}{2h^2} \\ &\quad + \frac{-4u(x_i, y_j) + u(x_{i+1}, y_{j-1}) + u(x_{i-1}, y_{j+1})}{2h^2}\end{aligned}$$



L'errore di troncamento (locale) nelle precedenti approssimazioni è  $O(h^2)$ <sup>(†)</sup>. Volendo un'approssimazione di ordine più elevato, potremmo approssimare  $\nabla^2 u(x_i, y_j)$  con

$$\nabla_9^2 u(x_i, y_j) = \frac{2}{3} \nabla_5^2 u(x_i, y_j) + \frac{1}{3} \nabla_x^2 u(x_i, y_j)$$



(†) Quando  $u \in C^4(\mathcal{R})$ , con derivate quarte limitate.

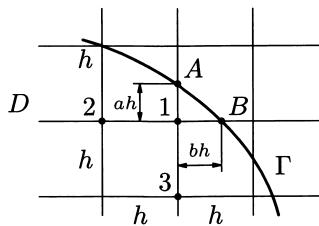


Figura 9.6

In questo caso l'errore è  $O(h^4)$ .

Quando il dominio spaziale  $D$  del problema non ha una geometria tale da consentire la costruzione di reticolati con tutti i nodi di frontiera appartenenti al suo contorno  $\Gamma$  possiamo procedere adottando, per esempio (vedi figura 9.6), una delle due strategie seguenti (vedere tuttavia [9.17, pag. 52]):

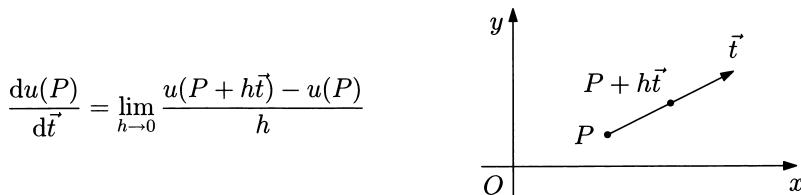
- (i) assegnamo a  $u_1$  il valore che la  $u(x, y)$  assume in un qualsiasi punto di  $\Gamma$  che disti dal nodo 1 meno di  $h$ ; per esempio  $u_1 = u_B$ ;
- (ii) approssimiamo le derivate relative al punto 1 con formule costruite sui nodi (non equidistanti) 2, 1,  $B$ , e 3, 1,  $A$ . In questo caso, per esempio, il laplaciano nel punto 1 verrebbe approssimato dalla formula seguente:

$$\left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)_1 = \frac{2}{h^2} \left[ \frac{u_2}{b+1} + \frac{u_3}{a+1} + \frac{u_A}{a(a+1)} + \frac{u_B}{b(b+1)} - \frac{a+b}{ab} u_1 \right] + O(h)$$

Va tuttavia rilevato che in tali situazioni si riduce l'efficienza e la competitività del metodo alle differenze finite; in questi casi è preferibile ricorrere ad un metodo agli elementi finiti (vedi paragrafo 9.5).

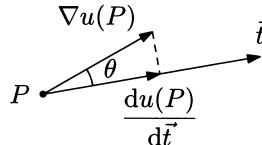
A volte nelle condizioni al contorno del problema sono presenti delle derivate direzionali, e in particolare quelle normali. Ricordiamo dapprima il significato di derivata direzionale di una funzione  $u(x, y) \equiv u(P)$  in un punto  $P$ .

Sia  $\vec{t}$  un vettore unitario. Per definizione



Alternativamente, possiamo affermare che  $du(P)/d\vec{t}$  è la componente del gradiente

$\nabla u(P)$  lungo la direzione del vettore unitario  $\vec{t}$ , cioè

$$\frac{du(P)}{d\vec{t}} = \nabla u(P) \cdot \vec{t} = \|\nabla u(P)\|_2 \cos \theta$$


Esaminiamo ora il problema di approssimare la derivata normale (ad una curva) in un punto  $(R)$ .

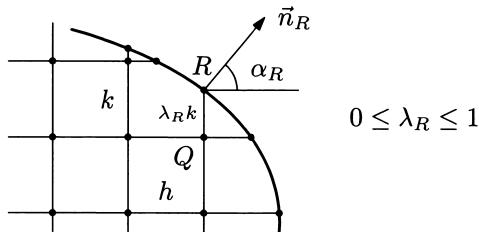


Figura 9.7

Sviluppiamo la funzione  $u(x, y)$  in serie di Taylor nell'intorno di  $Q = (x_Q, y_Q)$

$$\begin{aligned} u(x, y) &= u(Q) + (x - x_Q) \frac{\partial u(Q)}{\partial x} + (y - y_Q) \frac{\partial u(Q)}{\partial y} + \frac{(x - x_Q)^2}{2} \frac{\partial^2 u(Q)}{\partial x^2} \\ &\quad + \frac{(y - y_Q)^2}{2} \frac{\partial^2 u(Q)}{\partial y^2} + (x - x_Q)(y - y_Q) \frac{\partial^2 u(Q)}{\partial x \partial y} + \dots \end{aligned}$$

Derivando (rispetto a  $x$  prima, e a  $y$  poi) l'espressione precedente, otteniamo

$$\begin{aligned} u_x(x, y) &= u_x(Q) + (x - x_Q) u_{xx}(Q) + (y - y_Q) u_{xy}(Q) + \dots \\ u_y(x, y) &= u_y(Q) + (y - y_Q) u_{yy}(Q) + (x - x_Q) u_{xy}(Q) + \dots \end{aligned}$$

Richiamando poi la definizione di derivata direzionale possiamo infine concludere che

$$\begin{aligned} \frac{du(R)}{d\vec{n}} &= \nabla u_R \cdot \vec{n}_R = u_x(R) \cos \alpha_R + u_y(R) \sin \alpha_R = \\ &= [u_x(Q) + \lambda_R k u_{xy}(Q)] \cos \alpha_R + [u_y(Q) + \lambda_R k u_{yy}(Q)] \sin \alpha_R + O(h^2) + O(k^2) \end{aligned}$$

Il metodo delle differenze finite, con passi di discretizzazione  $h, k, \dots$  costanti in tutto  $\mathcal{R}$ , risulta particolarmente semplice ed efficiente quando la regione  $\mathcal{R}$  è regolare, nel senso che in essa è costruibile un reticolo regolare con passi  $h, k, \dots$  costanti e piccoli quanto si voglia, e la soluzione  $u$  del problema è sufficientemente regolare ed ha un comportamento uniforme in tutto  $\mathcal{R}$ . Il reticolo è detto regolare se, qualunque siano i passi  $h, k, \dots$  scelti, non ha punti di frontiera (vedere la definizione data a pag. 316) che

non appartengono anche al contorno di  $\mathcal{R}$ . Con il termine “sufficientemente regolare”, riferito alla soluzione  $u$ , intendiamo l’uniforme limitatezza delle derivate presenti nelle stime di errore delle formule alle differenze finite. Nel seguito considereremo solo problemi con queste caratteristiche.

Ricordiamo infine che la discretizzazione delle derivate non è l’unico procedimento per costruire schemi di calcolo alle differenze finite (vedere [9.30], [9.31]). Schemi analoghi vengono per esempio costruiti applicando il *metodo dei volumi finiti* a formulazioni integrali che rappresentano *leggi di conservazione* (di massa, di quantità di moto, di energia).

### 9.3.2 Equazioni di tipo iperbolico

Quale primo esempio di applicazione delle formule alle differenze finite consideriamo il seguente problema (iperbolico) del prim’ordine, lineare e a coefficienti costanti:

$$(9.13) \quad \begin{cases} u_t + au_x = 0, & 0 < x < 1, \quad t > 0 \\ u(x, 0) = u_0(x), & 0 \leq x \leq 1 \\ u(0, t) = f(t), & t > 0 \end{cases}$$

dove  $a$  è una costante positiva,  $u_0 \in C^1[0, 1]$  e  $f \in C^1(0, \infty)$ . Poiché le irregolarità presenti nei dati del problema si propagano all’interno della regione di definizione di quest’ultimo, affiché la soluzione risulti classica devono essere soddisfatte le seguenti condizioni di raccordo del dato al bordo con quello iniziale:

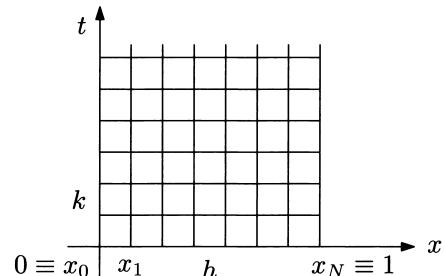
$$f(0^+) = u_0(0), \quad f'(0^+) + au'_0(0) = 0$$

Le formule alle differenze finite che seguono hanno tuttavia il comportamento descritto (in termini di  $O(\cdot)$ ) solo se si ha  $u \in C^2$ .

Tale modello è in realtà assai elementare, ma ci consente di introdurre due semplici schemi di calcolo e di far vedere come può essere condotto lo studio della loro *stabilità*.

Dopo aver costruito il reticolato rettangolare sottostante, approssimiamo le derivate presenti in (9.13) con le formule seguenti:

$$\begin{aligned} u_t(x_i, t_j) &= \frac{u(x_i, t_{j+1}) - u(x_i, t_j)}{k} + O(k) \\ u_x(x_i, t_j) &= \frac{u(x_i, t_j) - u(x_{i-1}, t_j)}{h} + O(h) \end{aligned}$$



Successivamente trascuriamo i termini  $O(k)$  e  $O(h)$ , i quali definiscono l’*errore locale* (unitario) di troncamento nel punto  $(x_i, t_j)$ . Denotando con  $u_{i,j}$  la conseguente appross-

simazione di  $u(x_i, t_j)$ , otteniamo il seguente schema esplicito di calcolo:

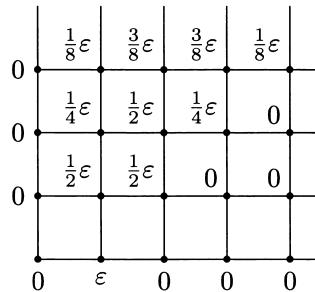
$$(9.14) \quad \begin{cases} u_{0,j} = f(t_j) & j = 1, 2, \dots \\ u_{i,0} = u_0(x_i) & i = 0, 1, \dots, N \\ u_{i,j+1} = (1 - \alpha)u_{i,j} + \alpha u_{i-1,j} & j = 0, 1, \dots; i = 1, \dots, N \end{cases}$$

dove  $\alpha = ak/h$ .

L'errore locale di troncamento rappresenta una misura della *consistenza* del metodo numerico, ovvero della “distanza”, nel punto  $(x_i, t_j)$ , del modello discreto dall'equazione differenziale. La proprietà di stabilità è fondamentale perché, come vedremo nel paragrafo 9.3.4, essa ci garantirà che la consistenza del metodo comporta anche la convergenza dell'approssimazione discreta prodotta da quest'ultimo alla soluzione del problema differenziale, quando  $h, k \rightarrow 0$ .

Supponiamo ora di introdurre delle perturbazioni nei dati del problema e di esaminare la loro propagazione all'interno della regione  $\mathcal{R} = \{0 < x < 1, t > 0\}$ . Diremo che *lo schema di calcolo è stabile se, fissato un generico istante  $t = \tau > 0$ , per tutti gli  $h = 1/N$  e  $k = \tau/M$  sufficientemente piccoli, ovvero con  $N$  e  $M$  interi positivi arbitrariamente grandi, le perturbazioni rimangono limitate in  $(0, \tau]$ , uniformemente rispetto ad  $h$  e  $k$ ; instabile altrimenti*. In particolare, senza nulla perdere in generalità, per illustrare con un semplice esempio il fenomeno della stabilità/instabilità perturbiamo il solo dato iniziale in un singolo punto, considerando due diverse scelte del parametro  $\alpha$ :

$$(i) \quad \alpha = \frac{1}{2}$$



**Figura 9.8**

La propagazione della perturbazione presente nel punto  $(x_1, 0)$  è stabile; anzi, essa converge addirittura a zero quando  $i, j \rightarrow \infty$ , uniformemente rispetto ad  $h$  e  $k$ .

$$(ii) \quad \alpha = 2$$

In questo secondo caso, come si evince dalla figura 9.9 che segue, il comportamento dello schema risulta instabile:

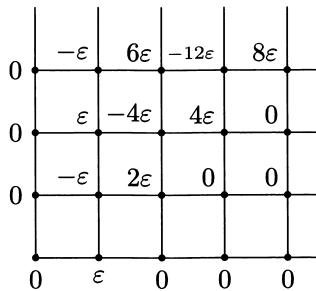


Figura 9.9

Appare quindi logico chiederci per quali valori del parametro  $\alpha$  la propagazione è stabile. Consideriamo solo i nodi  $(x_i, t_j)$ ,  $i = 1, \dots, N$ ,  $j \geq 0$ , e supponiamo, per semplicità, di perturbare solamente i dati iniziali  $u_{i,0}$  (non quelli al bordo  $x = 0$ ). Introduciamo il vettore soluzione

$$U_j = (u_{1,j}, u_{2,j}, \dots, u_{N,j})^T$$

e la corrispondente perturbazione  $E_j = (\varepsilon_{1,j}, \varepsilon_{2,j}, \dots, \varepsilon_{N,j})^T$ . Il metodo (9.14) può allora venire riformulato nel modo seguente:

$$U_{j+1} = AU_j + \alpha v_j, \quad U_0 = (u_0(x_1), u_0(x_2), \dots, u_0(x_N))^T$$

dove

$$A = \begin{pmatrix} 1-\alpha & & & 0 \\ \alpha & \diagdown & & \\ 0 & & \alpha & 1-\alpha \end{pmatrix}, \quad v_j = \begin{pmatrix} u_{0,j} \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

L'errore  $E_j$  soddisfa la relazione

$$E_{j+1} = AE_j, \quad j = 0, 1, \dots$$

quindi

$$E_{j+1} = A^{j+1}E_0$$

Per quali valori di  $\alpha$  la matrice  $A^{j+1}$  è limitata in norma, uniformemente rispetto sia a  $j$  che a  $N$ ?

Consideriamo la norma  $p$  di vettore,  $1 \leq p \leq \infty$ , e la corrispondente norma naturale di matrice. Ricordiamo che la definizione di norma  $p$  in  $\mathbb{R}^n$  può essere estesa al caso di vettori, e quindi di matrici, con infinite componenti  $v = (\dots, v_{-1}, v_0, v_1, \dots)^T$  nel modo

seguinte:

$$\|v\|_p = \left( \sum_{i=-\infty}^{\infty} |v_i|^p \right)^{1/p}, \quad 1 \leq p < \infty$$

$$\|v\|_\infty = \sup_{-\infty < i < \infty} |v_i|$$

Inoltre quando i valori  $v_i$  rappresentano ordinate corrispondenti ad ascisse  $x_i$ , con  $h = x_{i+1} - x_i$ , e il numero di componenti è  $N = N(h) \rightarrow \infty$  per  $h \rightarrow 0$  oppure è infinito, nel caso  $p \neq \infty$  può risultare indispensabile considerare le seguenti versioni “pesate”:

$$\|v\|_{p,h} = \left( h \sum_{i=1}^N |v_i|^p \right)^{1/p} \quad 1 \leq p < \infty$$

$$\|v\|_{p,h} = \left( h \sum_{i=-\infty}^{\infty} |v_i|^p \right)^{1/p}$$

Quando  $p = \infty$ , per convenzione poniamo  $\|v\|_{\infty,h} = \|v\|_\infty$ .

Se  $v_i$  è a sua volta un vettore, per esempio in  $\mathbb{R}^l$ , nelle definizioni di norma precedenti occorre sostituire  $|v_i|$  con  $\|v_i\|_p$ , dove quest’ultima è la classica norma  $p$  in  $\mathbb{R}^l$ .

Nel caso di problemi di due o più variabili spaziali le predette definizioni vengono facilmente generalizzate (vedi [9.25]). Per esempio, nel caso in cui si hanno valori  $v_{i,j}$  corrispondenti ad ascisse  $(x_i, y_j)$ , con  $h_1 = x_{i+1} - x_i$  e  $h_2 = y_{j+1} - y_j$ , definiamo

$$\|v\|_{p,h} = \left( h_1 h_2 \sum_{j=1}^M \sum_{i=1}^N |v_{ij}|^p \right)^{1/p}$$

Una condizione sufficiente perché  $\|E_{j+1}\|_p \leq C$ , ovvero  $\|E_{j+1}\|_{p,h} \leq C$ , con  $C$  indipendente da  $j$  e da  $N$ , è la seguente

$$\|A\|_1 \leq 1$$

ossia

$$|\alpha| + |1 - \alpha| \leq 1$$

Infatti, poiché nel nostro caso  $\|A\|_1 = \|A\|_\infty$  abbiamo<sup>(†)</sup>  $\|A\|_p \leq \|A\|_1$ ,  $1 \leq p \leq \infty$ , e quindi

$$\|E_{j+1}\|_p \leq \|A\|_1^{j+1} \|E_0\|_p$$

Possiamo pertanto concludere che quando  $0 < \alpha \leq 1$  abbiamo

$$\|E_{j+1}\|_p \leq \|E_0\|_p, \quad \text{ovvero} \quad \|E_{j+1}\|_{p,h} \leq \|E_0\|_{p,h}$$

per ogni  $1 \leq p \leq \infty$ , e quindi affermare che lo schema (9.14) è, nella norma  $p$ , oppure  $p, h$ , stabile quando la scelta dei passi  $h$  e  $k$  soddisfa la condizione

$$a \frac{k}{h} \leq 1$$

---

(†) Per un noto teorema di Riesz-Thorin sull’interpolazione di operatori lineari.

In questo caso diciamo che il metodo (9.14) è *condizionatamente stabile*, in quanto  $h$  e  $k$  debbono obbedire al vincolo predetto.

Poiché nello studio della stabilità la dimensione  $N$  dei vettori e delle matrici in questione non è supposta costante, ma arbitrariamente grande, l'uniforme limitatezza di  $\|E_{j+1}\|$ , sia rispetto a  $j$  che a  $N$ , in una certa norma non implica necessariamente l'uniforme limitatezza in un'altra norma. Ciò risulterebbe vero solo se la dimensione  $N$  fosse fissa, e non variabile. Pertanto, in generale la stabilità di uno schema di calcolo in una certa norma non significa che lo schema continua ad essere stabile se cambiamo la norma. Nel caso dello schema (9.14) abbiamo invece dimostrato la sua stabilità (condizionata) qualunque sia la norma  $p$  (oppure  $p, h$ ).

Osserviamo infine che con l'analisi da noi fatta abbiamo stabilito le condizioni ( $\alpha \leq 1$ ) che assicurano solo la stabilità della propagazione degli errori presenti nei dati iniziali (all'istante  $t = 0$ ); lasciamo al lettore l'esame della propagazione di quelli presenti sul bordo  $x = 0$ .

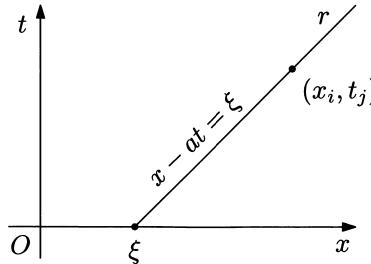
La soluzione del problema

$$\begin{cases} u_t + au_x = 0 \\ u(x, 0) = u_0(x) \end{cases} \quad -\infty < x < \infty, \quad t > 0$$

è nota analiticamente:

$$u(x, t) = u_0(x - at)$$

Osserviamo che il valore  $u(x_i, t_j)$  dipende unicamente dal dato iniziale  $u_0(\xi)$ ,  $\xi = x_i - at_j$ ; anzi,  $u(x, t) = u_0(\xi)$  per tutti i punti  $(x, t)$  della retta (caratteristica) di equazione  $x - at = \xi$ :

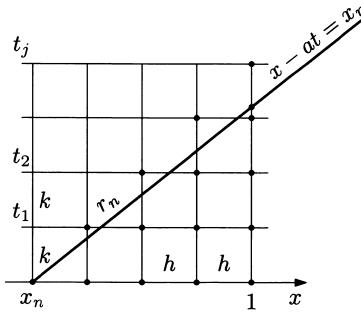


**Figura 9.10**

L'ascissa  $x = \xi$  è definita *dominio di dipendenza* del punto  $(x_i, t_j)$ , mentre la caratteristica  $x - at = \xi$  rappresenta il *dominio di influenza* del dato iniziale  $u_0(\xi)$ , ovvero del punto  $x = \xi$ .

Nel caso del problema (9.13) i dati iniziali assegnati sul segmento  $0 \leq x \leq 1$  definiscono univocamente la soluzione  $u(x, t)$  solamente nel triangolo delimitato dalle rette  $t = x/a$ ,  $t = 0$  e  $x = 1$ .

Il dato  $u(0, t) = f(t)$  sul bordo  $x = 0$  è indispensabile per poter definire la soluzione in tutta la striscia  $0 \leq x \leq 1, t > 0$ . Se il coefficiente  $a$  in (9.13) fosse stato negativo, per poter definire  $u(x, t)$  in tutta la regione  $\mathcal{R}$  avremmo dovuto assegnare un dato sul bordo  $x = 1$ , e non su  $x = 0$ .



**Figura 9.11**

I dati assegnati sul segmento  $[x_n, 1]$  definiscono la soluzione  $u(x, t)$  solo nel triangolo delimitato superiormente dalla caratteristica  $r_n$  (vedi figura 9.11). La condizione  $\alpha \leq 1$ , da noi precedentemente individuata per garantire la stabilità del metodo di risoluzione, appare indispensabile anche da un altro punto di vista. Infatti se fosse  $\alpha > 1$ , lo schema (9.14) fornirebbe, utilizzando solamente dati relativi a punti dell'intervallo  $[x_n, 1]$  approssimazioni della soluzione in punti situati fuori dal predetto triangolo<sup>(†)</sup>.

Un secondo possibile schema di calcolo, apparentemente di tipo implicito, ma di fatto esplicito, è il seguente:

$$(9.15) \quad \frac{u_{i,j+1} - u_{i,j}}{k} + a \frac{u_{i,j+1} - u_{i-1,j+1}}{h} = 0$$

Anche in questo caso l'errore locale di troncamento, nel punto  $(x_i, t_{j+1})$ , è  $O(h) + O(k)$ . Ripercorrendo lo studio della stabilità effettuato per lo schema precedente otteniamo che l'errore  $E_j$  associato alla (9.15) soddisfa la relazione

$$AE_{j+1} = E_j, \quad j = 0, 1, \dots$$

---

<sup>(†)</sup> Questa osservazione rappresenta, sostanzialmente, il noto criterio di Courant-Friedrichs-Lowy (vedi ad esempio [9.15]).

con

$$A = \begin{pmatrix} 1 + \alpha & & & & \\ -\alpha & 1 & & & \\ & -\alpha & 1 + \alpha & & \\ 0 & & -\alpha & 1 & \\ & & & -\alpha & 1 + \alpha \end{pmatrix} = (1 + \alpha) \begin{pmatrix} 1 & & & & \\ -\beta & 1 & & & \\ 0 & -\beta & 1 & & \\ & & -\beta & 1 & \\ & & & -\beta & 1 \end{pmatrix}, \quad \beta = \frac{\alpha}{1 + \alpha}$$

Possiamo pertanto scrivere

$$E_{j+1} = A^{-1} E_j$$

e dedurre per  $A^{-1}$  la seguente espressione

$$A^{-1} = \frac{1}{1 + \alpha} \begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ \beta & 1 & 0 & 0 & \dots & 0 \\ \beta^2 & \beta & 1 & 0 & \dots & 0 \\ \beta^3 & \beta^2 & \beta & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \beta^{N-1} & \beta^{N-2} & \beta^{N-3} & \beta^{N-4} & \dots & 1 \end{pmatrix}$$

Il calcolo diretto della norma  $\|A^{-1}\|_1$  ci consente di affermare che per tutti gli interi  $1 \leq N \leq \infty$  abbiamo

$$\|A^{-1}\|_1 \leq \frac{1}{1 + \alpha} \sum_{k=0}^{\infty} \beta^k = 1$$

Lo schema (9.15) risulta pertanto stabile (nella norma  $p$ ,  $1 \leq p \leq \infty$ , e quindi anche nella norma  $p, h$ ) qualunque sia la scelta dei passi  $h$  e  $k$ . In questo caso diciamo che lo schema è *incondizionatamente stabile*.

Per una descrizione più completa del concetto di stabilità di uno schema alle differenze finite e dei vari criteri proposti per determinare le condizioni di stabilità, suggeriamo le letture [9.3] e [9.4].

Poiché d'ora in avanti lo studio della stabilità verrà effettuato con riferimento ad una generica norma  $p$ ,  $1 \leq p \leq \infty$ , eventualmente con peso  $h$ , tralascieremo di specificare che le condizioni di stabilità ottenute valgono qualunque sia la norma  $p$  scelta.

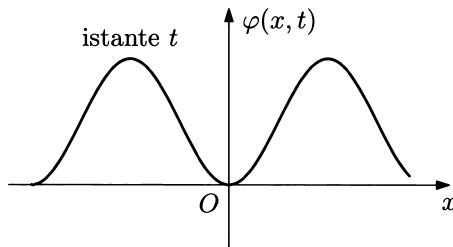
Come già abbiamo visto nel paragrafo 9.2 e nel problema precedente, nelle equazioni (o sistemi) di tipo iperbolico le linee caratteristiche sono tutte reali e distinte. Queste ultime possono svolgere un ruolo fondamentale nello studio della soluzione del problema; infatti, le eventuali discontinuità presenti nei dati si propagano proprio lungo le caratteristiche. La caratteristica su cui una “singolarità”<sup>(†)</sup> si propaga separa due “lembi di soluzione” più regolari. In situazioni di questo tipo ha avuto un certo successo, almeno nel caso di problemi in una sola variabile spaziale, il noto *metodo delle caratteristiche*, che consiste nell'integrare il sistema avanzando proprio lungo le curve caratteristiche (vedi ad esempio [9.1]).

<sup>(†)</sup> Per esempio una discontinuità sulla derivata prima.

Nel caso di problemi con soluzioni sufficientemente regolari, l'approccio delle differenze finite costruite su un sistema di coordinate ortogonali classiche non solo è generalmente più efficiente ma è anche facilmente generalizzabile al caso di problemi in 2 e 3 dimensioni (spaziali).

Nelle pagine che seguono prendiamo in esame la classica equazione delle onde in una dimensione (spaziale)

$$(9.16) \quad \frac{\partial^2 \varphi}{\partial t^2} - c^2 \frac{\partial^2 \varphi}{\partial x^2} = 0$$



dove  $c > 0$  è una costante che rappresenta la velocità di propagazione, di cui è noto l'integrale generale:

$$\varphi(x, t) = f_1(x + ct) + f_2(x - ct)$$

essendo  $f_1$  e  $f_2$  due funzioni arbitrarie ( $\in C^2(\mathbb{R})$  per esempio, se vogliamo che  $\varphi$  sia classica). Se alla (9.16) associamo le due condizioni iniziali

$$(9.17) \quad \begin{cases} \varphi(x, 0) = f(x) \\ \frac{\partial \varphi}{\partial t}(x, 0) = g(x) \end{cases}$$

per la soluzione (unica) del sistema (9.16), (9.17) abbiamo (vedi ad esempio [9.15]) la rappresentazione

$$\varphi(x, t) = \frac{1}{2}[f(x + ct) + f(x - ct)] + \frac{1}{2c} \int_{x-ct}^{x+ct} g(\xi) d\xi$$

Quest'ultima espressione ci consente di affermare che quando  $f \in C^2(\mathbb{R})$  e  $g \in C^1(\mathbb{R})$  la soluzione  $\varphi$  è classica. Inoltre, il valore della soluzione  $\varphi(x, t)$  nel punto  $(x_0, t_0)$  dipende solo dai dati iniziali sul segmento  $[x_0 - ct_0, x_0 + ct_0]$ .

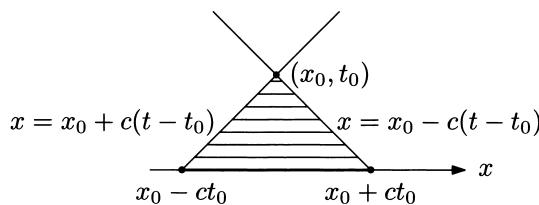


Figura 9.12

Il segmento  $[x_0 - ct_0, x_0 + ct_0]$  è il *dominio di dipendenza del punto*  $(x_0, t_0)$ . Le rette  $x = x_0 + c(t - t_0)$  e  $x = x_0 - c(t - t_0)$  sono le due caratteristiche del sistema passanti per il punto  $(x_0, t_0)$ . La conoscenza della soluzione in  $[x_0 - ct_0, x_0 + ct_0]$  definisce (univocamente) la soluzione solo nel triangolo di vertici  $(x_0 - ct_0, 0)$ ,  $(x_0 + ct_0, 0)$ ,  $(x_0, t_0)$ .

Consideriamo il dato iniziale nel punto  $(x_0, 0)$ . Quali sono i punti del piano  $(x, t)$  i cui valori  $\varphi(x, t)$  risultano influenzati da tale dato? Il valore della soluzione nel punto  $(x_0, 0)$  influenza la soluzione in tutti i punti della regione infinita delimitata dalle caratteristiche  $x = x_0 + ct$  e  $x = x_0 - ct$  (vedi figura 9.13). Questa regione rappresenta il *dominio di influenza* del punto  $(x_0, 0)$ .

Osserviamo infine che per  $t \rightarrow \infty$  la soluzione  $\varphi(x, t)$  non decade, cioè in generale non converge ad uno stato stazionario.

Dopo queste considerazioni preliminari di carattere teorico, necessarie per porre in evidenza l'importanza che le caratteristiche assumono nello studio dei sistemi iperbolici, ritorniamo agli aspetti numerici, e più precisamente alla costruzione di uno schema di calcolo alle differenze finite per un problema definito in un dominio spaziale limitato, con valori iniziali e al bordo.

Pertanto, all'equazione (9.16) e alle condizioni iniziali (9.17), che ora consideriamo definite rispettivamente nelle regioni  $\mathcal{R} = \{0 < x < 1, t > 0\}$  e  $0 \leq x \leq 1$ , aggiungiamo per esempio le condizioni “al contorno”  $\varphi(0, t) = \alpha(t)$ ,  $\varphi(1, t) = \beta(t)$ ,  $t > 0$ . Queste ultime ci consentono di definire univocamente la soluzione  $\varphi(x, t)$  in tutto  $\mathcal{R}$  (vedi, ad esempio, [9.15, pag. 43]).

Al fine di ottenere uno schema di calcolo alle differenze finite con il massimo ordine di convergenza (2), tenuto conto della rappresentazione degli errori di troncamento delle formule alle differenze finite per l'approssimazione delle derivate seconde, supponiamo che le condizioni iniziali e al bordo soddisfino le condizioni di raccordo che garantiscono addirittura l'esistenza di una soluzione  $\varphi \in C^4$  in  $0 \leq x \leq 1$ ,  $t \geq 0$ . Per semplificare la descrizione che segue, effettuiamo in (9.16) il cambiamento di variabile  $y = ct$  e poniamo  $u(x, y) = \varphi(x, t)$ ; otteniamo

$$(9.18) \quad \begin{cases} \frac{\partial^2 u}{\partial x^2} = \frac{\partial^2 u}{\partial y^2} \\ u(x, 0) = f(x) \\ \frac{\partial u}{\partial y}(x, 0) = \frac{1}{c} g(x) \equiv g_1(x) \\ u(0, y) = \alpha\left(\frac{y}{c}\right) \equiv \alpha_1(y) \\ u(1, y) = \beta\left(\frac{y}{c}\right) \equiv \beta_1(y) \end{cases}$$

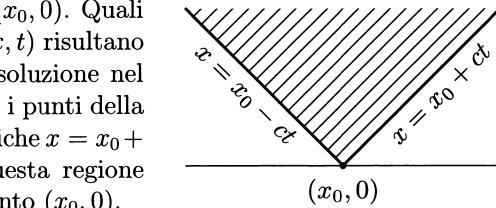
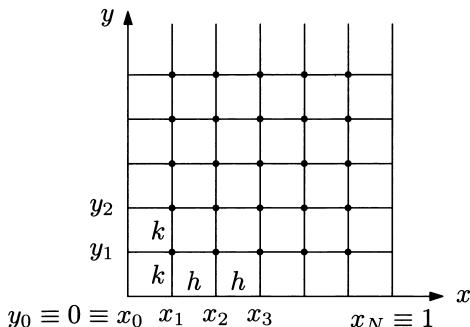


Figura 9.13



Sostituiamo il dominio  $D = \{(x, y), 0 < x < 1, 0 < y < \infty\}$  con il reticolo rettangolare sopra disegnato, e il modello (9.18) con

$$(9.19) \quad \begin{cases} \frac{\partial^2 u}{\partial x^2}(x_i, y_j) = \frac{\partial^2 u}{\partial y^2}(x_i, y_j), & i = 1, 2, \dots, N-1, \quad j = 1, 2, 3, \dots \\ u(x_i, 0) = f(x_i), & i = 1, 2, \dots, N-1 \\ \frac{\partial u}{\partial y}(x_i, 0) = g_1(x_i), & i = 1, 2, \dots, N-1 \\ u(0, y_j) = \alpha_1(y_j), & j = 0, 1, \dots \\ u(1, y_j) = \beta_1(y_j), & j = 0, 1, \dots \end{cases}$$

Approssimiamo le derivate nei nodi del reticolo con le formule alle differenze:

$$\begin{aligned} \frac{\partial^2 u}{\partial x^2}(x_i, y_j) &= \frac{u(x_{i+1}, y_j) - 2u(x_i, y_j) + u(x_{i-1}, y_j)}{h^2} + O(h^2) \\ \frac{\partial^2 u}{\partial y^2}(x_i, y_j) &= \frac{u(x_i, y_{j+1}) - 2u(x_i, y_j) + u(x_i, y_{j-1})}{k^2} + O(k^2) \\ \frac{\partial u}{\partial y}(x_i, 0) &= \frac{u(x_i, y_1) - u(x_i, y_{-1})}{2k} + O(k^2) \end{aligned}$$

i cui errori sono del secondo ordine in virtù dell'ipotesi di regolarità fatta su  $u(x, y)$ . Trascurando i termini  $O(\cdot)$ , e denotando con  $u_{i,j}$  l'approssimazione di  $u(x_i, y_j)$  che lo schema risultante fornirà, otteniamo il seguente modello discreto:

$$(9.20) \quad \begin{cases} \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{k^2} \\ u_{i,0} = f(x_i) & i = 1, 2, \dots, N-1 \\ \frac{u_{i,1} - u_{i,-1}}{2k} = g_1(x_i) & j = 0, 1, 2, \dots \\ u_{0,j} = \alpha_1(y_j) \\ u_{N,j} = \beta_1(y_j) \end{cases}$$

L'errore locale di troncamento nel punto  $(x_i, t_j)$  è  $O(h^2) + O(k^2)$ , mentre la seconda condizione iniziale in (9.18) è stata discretizzata con errore  $O(k^2)$ .

Per poter costruire questo schema di calcolo abbiamo dovuto introdurre un livello “fittizio”  $j = -1$ , e quindi considerare l'equazione differenziale (discretizzata) anche al livello  $j = 0$ . Il livello  $j = -1$  è di fatto fittizio in quanto viene poi eliminato. Infatti dalla terza equazione in (9.20) ricaviamo la quantità ausiliaria

$$u_{i,-1} = u_{i,1} - 2kg_1(x_i)$$

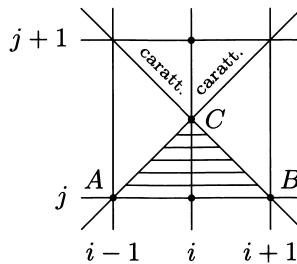
che inseriamo nella prima equazione quando  $j = 0$ . Introducendo poi il parametro  $\lambda = k/h$ , possiamo riscrivere il sistema (9.20) nella forma

$$\left\{ \begin{array}{l} u_{i,j+1} = \lambda^2(u_{i-1,j} + u_{i+1,j}) + 2(1 - \lambda^2)u_{i,j} - u_{i,j-1} \\ u_{i,0} = f(x_i) \\ u_{i,-1} = u_{i,1} - 2kg_1(x_i) \\ u_{0,j} = \alpha_1(y_j) \\ u_{N,j} = \beta_1(y_j) \end{array} \right.$$

e quindi ottenere il seguente schema di calcolo (esplicito):

$$(9.21) \quad \left. \begin{array}{l} u_{i,0} = f(x_i) \\ u_{i,1} = \frac{1}{2}\lambda^2[f(x_{i-1}) + f(x_{i+1})] \\ \quad + (1 - \lambda^2)f(x_i) + kg_1(x_i) \\ u_{0,j} = \alpha_1(y_j) \\ u_{i,j+1} = \lambda^2(u_{i-1,j} + u_{i+1,j}) + 2(1 - \lambda^2)u_{i,j} - u_{i,j-1} \\ u_{N,j} = \beta_1(y_j) \end{array} \right\} \quad j = 1, 2, \dots$$

Osserviamo infine che  $u_{i,j+1}$  dipende unicamente dalle approssimazioni  $u_{i-1,j}$ ,  $u_{i,j}$  e  $u_{i+1,j}$ , tutte relative allo stadio precedente  $j$ , e da  $u_{i,j-1}$ .



**Figura 9.14**

Quando scegliamo  $\lambda > 1$  non dobbiamo attenderci da questo schema una buona approssimazione. Infatti, la conoscenza della soluzione sull'intervallo  $(A, B)$  determina univocamente la soluzione del problema solo nel triangolo  $ABC$ ; la soluzione numerica  $u_{i,j+1}$  si riferisce invece ad un punto che si trova fuori da detto triangolo.

Denotiamo con  $U_j$  il vettore, di dimensione finita  $N - 1$ , contenente le approssimazioni  $u_{1,j}, u_{2,j}, \dots, u_{N-1,j}$  nei nodi scelti all'“istante”  $j$ . Lo schema (9.21) può allora venire

riscritto in forma vettoriale:

$$(9.22) \quad \begin{cases} U_0 = [f(x_i)] \\ U_1 = \left[ \frac{1}{2} \lambda^2 (f(x_{i-1}) + f(x_{i+1})) + (1 - \lambda^2) f(x_i) + k g_1(x_i) \right] \\ U_{j+1} = AU_j - U_{j-1} + \lambda^2 b_j, \quad j = 1, 2, \dots \end{cases}$$

$$A = \begin{pmatrix} 2(1 - \lambda^2) & \lambda^2 & & & 0 \\ \lambda^2 & - & - & - & \\ & - & - & - & \lambda^2 \\ & & - & - & \\ 0 & & & \lambda^2 & 2(1 - \lambda^2) \end{pmatrix} \in \mathbb{R}^{(N-1) \times (N-1)}, \quad b_j = \begin{pmatrix} \alpha_1(y_j) \\ 0 \\ \vdots \\ 0 \\ \beta_1(y_j) \end{pmatrix}$$

La relazione ricorsiva sulle  $U_j$  può anche essere riformulata nel modo seguente

$$U_{j+1} = (2I - \lambda^2 T)U_j - U_{j-1} + \lambda^2 b_j$$

dove

$$T = \begin{pmatrix} 2 & -1 & & & 0 \\ -1 & - & - & - & \\ & - & - & - & -1 \\ & & - & - & \\ 0 & & & -1 & 2 \end{pmatrix} \in \mathbb{R}^{(N-1) \times (N-1)}$$

Possiamo ora ricondurre la (9.22) alla forma

$$V_{j+1} = BV_j + d_j$$

in modo da poter studiare<sup>(†)</sup> la stabilità dello schema esaminando gli autovalori della matrice  $B$ . A tale scopo introduciamo il nuovo vettore

$$V_j = \begin{pmatrix} U_j \\ U_{j-1} \end{pmatrix}$$

e la matrice

$$B = \begin{pmatrix} A & -I \\ I & O \end{pmatrix}$$

così che

$$\begin{pmatrix} U_{j+1} \\ U_j \end{pmatrix} = \begin{pmatrix} A & -I \\ I & O \end{pmatrix} \begin{pmatrix} U_j \\ U_{j-1} \end{pmatrix} + \lambda^2 \begin{pmatrix} b_j \\ o \end{pmatrix} = \begin{pmatrix} AU_j - U_{j-1} + \lambda^2 b_j \\ U_j \end{pmatrix}$$

---

(†) Supponendo per semplicità di introdurre delle perturbazioni nei soli dati iniziali.

È sufficiente ora determinare gli autovalori  $\mu_i$  della matrice  $B$ , ossia risolvere l'equazione

$$(9.23) \quad \begin{pmatrix} A & -I \\ I & O \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \mu \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$$

Dalla (9.23) otteniamo

$$\begin{cases} Au_1 - u_2 = \mu u_1 \\ u_1 = \mu u_2 \end{cases}$$

e quindi

$$(9.24) \quad (\mu^2 I - \mu A + I)u_2 = o$$

Osserviamo che quest'ultima relazione può essere dedotta direttamente dalla (9.22) con la sostituzione  $U_j = \mu^j u^{(\dagger)}$ .

Infine, richiamando la forma data alla matrice  $A$  per ottenere la (9.3.2), possiamo riscrivere la (9.24) come segue:

$$[(\mu^2 - 2\mu + 1)I + \mu\lambda^2 T]u_2 = o$$

cioè

$$(T - \alpha I)u_2 = o, \quad \alpha = -\frac{(\mu - 1)^2}{\mu\lambda^2}$$

La quantità  $\alpha$  rappresenta il generico autovalore di  $T$ . Ma gli autovalori della matrice  $T$  sono noti:

$$\alpha_i = 4 \sin^2 \frac{i\pi}{2N}, \quad i = 1, \dots, N-1$$

e per gli autovalori  $\mu_i$  di  $B$  abbiamo l'espressione

$$\mu_i = \left(1 - \frac{\lambda^2 \alpha_i}{2}\right) \pm \sqrt{\left(1 - \frac{\lambda^2 \alpha_i}{2}\right)^2 - 1}$$

Osserviamo preliminarmente che  $\mu_i$  non deve essere reale, altrimenti una delle due radici risulta sicuramente  $> 1$  (il loro prodotto vale 1). Se  $\mu_i$  è complesso allora necessariamente  $|\mu_i| = 1$ . Tuttavia, poiché gli autovalori sono tutti distinti, la condizione  $|\mu_i| = 1$  garantisce la limitatezza, uniforme rispetto a  $j$  e  $N$ , dei moduli di tutte le componenti del vettore perturbazione  $E_{j+1} = B^j E_1$  presente in  $V_{j+1}$  e dovuto alla propagazione dell'errore iniziale  $E_1$  in  $V_1$ , e quindi la stabilità dello schema. Infatti, nel caso in questione gli autovalori  $\mu_i$  sono tutti distinti e la matrice  $B$  risulta diagonalizzabile, cioè

$$B = HDH^{-1}$$

---

(<sup>†</sup>) In modo perfettamente analogo a quanto viene fatto nello studio della zero-stabilità delle equazioni differenziali ordinarie; vedi pagina 277.

dove  $H$  è la matrice che ha come colonne gli autovettori di  $B$  e  $D$  è la matrice diagonale degli autovalori; quindi

$$B^j = HD^jH^{-1}$$

Inoltre,

$$E_{j+1} = HD^jH^{-1}E_1$$

e, posto  $\bar{E}_{j+1} = H^{-1}E_{j+1}$ ,

$$\bar{E}_{j+1} = D^j\bar{E}_1$$

Da quest'ultima relazione deduciamo

$$(\bar{E}_{j+1})_i = \mu_i^j (\bar{E}_1)_i, \quad i = 1, \dots, N-1$$

onde

$$|(\bar{E}_{j+1})_i| = |(\bar{E}_1)_i|, \quad i = 1, \dots, N-1$$

Gli autovalori  $\mu_i$  sono complessi solo se  $0 < \lambda^2 \alpha_i / 2 < 2$ , cioè solo se

$$\lambda^2 < \frac{1}{\sin^2 \frac{(N-1)\pi}{2N}}$$

Ma  $\sin^2(N-1)\pi/(2N) < 1$  e  $\lim_{N \rightarrow \infty} \sin^2(N-1)\pi/(2N) = 1$ , per cui possiamo senz'altro concludere che lo schema è stabile quando  $\lambda \leq 1$ .

### 9.3.3 Equazioni di tipo parabolico

Tipico esempio di tali equazioni è l'equazione del calore (omogenea):

$$(9.25) \quad \frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = 0$$

Le curve caratteristiche di tale equazione sono le rette  $t = \text{costante}$ ; dunque non è possibile prescrivere arbitrariamente sull'asse delle  $x$  il valore di  $u$  e di  $u_t$  (problema di Cauchy per la (9.25)). Risulta invece ben posto il problema dato dalla (9.25) e dalla condizione iniziale

$$u(x, 0) = f(x), \quad -\infty < x < \infty$$

In tal caso la soluzione esatta è esprimibile nella forma (vedi [9.15, pag. 209])

$$u(x, t) = \frac{1}{\sqrt{4\pi t}} \int_{-\infty}^{\infty} e^{-(x-s)^2/4t} f(s) ds$$

Il valore che la  $u(x, t)$  assume nel generico punto  $(x_0, t_0)$  dipende dal dato iniziale sull'intero asse  $x$ , ossia il dominio di dipendenza del punto  $(x_0, t_0)$  è l'asse  $x$ . Inoltre, per  $t \rightarrow \infty$  la soluzione decade esponenzialmente verso uno stato stazionario.

Osserviamo infine che l'equazione (9.25) ha un effetto regolarizzante, nel senso che pur avendo, per esempio, il dato iniziale  $f(s)$  solo limitato e continuo a tratti sull'asse  $x$ , la soluzione  $u(x, t)$  risulta infinitamente derivabile per ogni  $t > 0$ . Inoltre

$$\lim_{\substack{x \rightarrow \xi \\ t \rightarrow 0}} u(x, t) = f(\xi)$$

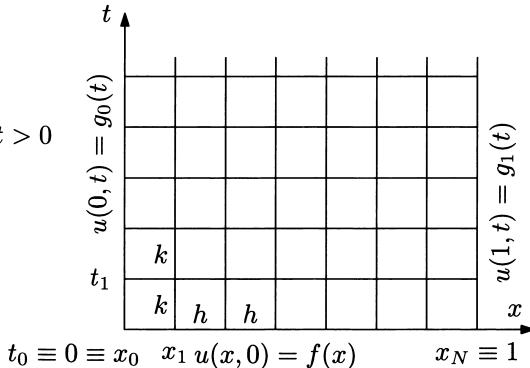
Questo comportamento è proprio delle equazioni paraboliche.

Consideriamo un filo metallico (di lunghezza unitaria) termicamente isolato, con distribuzione iniziale della temperatura nota. Supponiamo inoltre che gli estremi del filo siano mantenuti a temperature note, in ogni istante  $t > 0$ . Vogliamo conoscere la distribuzione della temperatura  $u(x, t)$  negli istanti successivi a quello iniziale.

Il suddetto problema è descritto, in termini adimensionali, dal seguente sistema di equazioni:

(9.26)

$$\begin{cases} \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, & 0 < x < 1, \quad t > 0 \\ u(x, 0) = f(x), & 0 \leq x \leq 1 \\ u(0, t) = g_0(t), & t > 0 \\ u(1, t) = g_1(t), & t > 0 \end{cases}$$



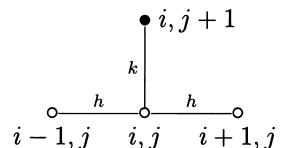
Supponiamo che i dati  $f, g_0, g_1$  soddisfino le condizioni di regolarità e di raccordo che rendono la soluzione  $u$ , che è  $C^\infty$  all'interno del dominio di definizione, anche continua nella chiusura di tale regione.

Discretizziamo dapprima il problema utilizzando le formule

$$\begin{aligned} \frac{\partial u}{\partial t}(x_i, t_j) &= \frac{u(x_i, t_{j+1}) - u(x_i, t_j)}{k} + O(k) \\ \frac{\partial^2 u}{\partial x^2}(x_i, t_j) &= \frac{u(x_{i+1}, t_j) - 2u(x_i, t_j) + u(x_{i-1}, t_j)}{h^2} + O(h^2) \end{aligned}$$

Posto  $\lambda = k/h^2$ , dalla prima equazione del sistema (9.26) otteniamo

$$u_{i,j+1} = \lambda u_{i-1,j} + (1 - 2\lambda)u_{i,j} + \lambda u_{i+1,j}, \quad i = 1, 2, \dots, N-1,$$



L'errore locale di troncamento (in  $(x_i, t_j)$ ) è  $O(k) + O(h^2)$ .

Supponendo note tutte le  $u_{i,j}$ ,  $i = 0, 1, \dots, N$ , al tempo  $t_j$ , la precedente relazione di ricorrenza ci dà esplicitamente le  $\{u_{i,j}\}$  al tempo  $t_{j+1}$ .

Lo schema di calcolo (esplicito) è dunque il seguente:

$$(9.27) \quad \begin{aligned} u_{i,0} &= f(x_i) & i = 0, 1, \dots, N \\ j = 0, 1, \dots : \left\{ \begin{array}{ll} u_{0,j+1} &= g_0(t_{j+1}) \\ u_{i,j+1} &= \lambda u_{i-1,j} + (1 - 2\lambda)u_{i,j} + \lambda u_{i+1,j} \\ u_{N,j+1} &= g_1(t_{j+1}) \end{array} \right. & i = 1, \dots, N-1 \end{aligned}$$

ovvero

$$U_{j+1} = AU_j + \lambda v_j, \quad j = 0, 1, 2, \dots$$

dove

$$U_j = \begin{pmatrix} u_{1,j} \\ u_{2,j} \\ \vdots \\ u_{N-1,j} \end{pmatrix}, \quad A = \begin{pmatrix} 1-2\lambda & \lambda & & 0 \\ \lambda & - & & \\ & - & & \lambda \\ 0 & & \lambda & 1-2\lambda \end{pmatrix} \quad \text{e} \quad v_j = \begin{pmatrix} u_{0,j} \\ 0 \\ \vdots \\ 0 \\ u_{N,j} \end{pmatrix}$$

Supponiamo di introdurre delle perturbazioni  $E_0$  su dati  $U_0$  (non sui valori assegnati ai bordi  $x = 0$  e  $x = 1$ ). Le corrispondenti perturbazioni  $E_{j+1}$  presenti in  $U_{j+1}$  soddisfano la relazione  $E_{j+1} = AE_j$ ,  $j = 0, 1, \dots$ . Lo schema è certamente stabile quando  $\|A\| \leq 1$ ; inoltre, se gli autovalori di  $A$  sono distinti lo schema risulta stabile se e solo se  $\rho(A) \leq 1$ .

Al fine di determinare gli autovalori di  $A$ , decomponiamo la matrice nella forma

$$A = I - \lambda T$$

dove  $T$  denota la stessa matrice di pagina 332. Sappiamo che gli autovalori di  $T$  sono

$$\alpha_i = 4 \sin^2 \frac{i\pi}{2N}, \quad i = 1, \dots, N-1$$

quindi, per gli autovalori  $\mu_i$  di  $A$  abbiamo

$$\mu_i = 1 - 4\lambda \sin^2 \frac{i\pi}{2N}, \quad i = 1, \dots, N-1$$

e per il raggio spettrale

$$\rho(A) \leq \max(1, 4\lambda - 1)$$

Lo schema risulta pertanto stabile solo quando  $\lambda \leq 1/2$ , ovvero  $k \leq h^2/2$ .

Poiché la condizione precedente potrebbe risultare eccessivamente restrittiva, costruiamo ora uno schema di calcolo alternativo che, come vedremo, risulterà incondizionatamente stabile. A tale fine collichiamo l'equazione differenziale nel punto  $(x_i, t_{j+1/2})$ , ove

$t_{j+1/2} = t_j + k/2$ . Poiché i nodi del reticolo sono invece i punti  $(x_i, t_j)$ , per non coinvolgere esplicitamente l'istante  $t_{j+1/2}$  osserviamo che

$$\frac{\partial^2 u}{\partial x^2}(x_i, t_{j+1/2}) = \frac{1}{2}[u_{xx}(x_i, t_{j+1}) + u_{xx}(x_i, t_j)] + O(k^2)$$

Successivamente, approssimando  $u_{xx}(x_i, t_{j+1})$  e  $u_{xx}(x_i, t_j)$  con le note formule, e ricordando che

$$\frac{\partial u}{\partial t}(x_i, t_{j+1/2}) = \frac{u(x_i, t_{j+1}) - u(x_i, t_j)}{k} + O(k^2)$$

otteniamo il seguente schema implicito a due livelli, di ordine 2 sia in  $h$  che in  $k$ :

$$(9.28) \quad -\lambda u_{i-1,j+1} + 2(1 + \lambda)u_{i,j+1} - \lambda u_{i+1,j+1} = \lambda u_{i-1,j} + 2(1 - \lambda)u_{i,j} + \lambda u_{i+1,j}$$

ovvero

$$AU_{j+1} = BU_j + \lambda(v_{j+1} + v_j), \quad j = 0, 1, 2, \dots$$

con

$$A = \begin{pmatrix} 2(1 + \lambda) & -\lambda & & & \\ -\lambda & \ddots & & & \\ & & \ddots & & 0 \\ & & & -\lambda & \\ 0 & & & & 2(1 + \lambda) \end{pmatrix}, \quad B = \begin{pmatrix} 2(1 - \lambda) & \lambda & & & \\ \lambda & \ddots & & & 0 \\ & & \ddots & & \\ & & & \lambda & \\ 0 & & & & 2(1 - \lambda) \end{pmatrix}$$

Gli autovalori  $\mu_i$  della matrice  $A^{-1}B$  sono facilmente deducibili da quelli di  $T$ :

$$\mu_i = \frac{2 - 4\lambda \sin^2 \frac{i\pi}{2N}}{2 + 4\lambda \sin^2 \frac{i\pi}{2N}} < 1, \quad i = 1, 2, \dots, N-1$$

Lo schema (9.28) risulta pertanto incondizionatamente stabile. Esso è solitamente indicato con il nome di *Crank-Nicolson*.

Quest'ultimo metodo è隐式的, ma ha il vantaggio di risultare stabile per ogni scelta dei passi  $h$  e  $k$ . Nello schema precedente invece, per avere la stabilità  $h$  e  $k$  devono essere scelti in modo che  $k \leq h^2/2$ , e ciò può comportare un avanzamento nel tempo eccessivamente piccolo.

▷ **Osservazione.** Se le condizioni iniziali e quelle al contorno non coincidono nei vertici  $(0, 0)$  e  $(1, 0)$  del dominio di interesse, la soluzione  $u(x, t)$  risulta discontinua in quei punti. Ricordiamo tuttavia che nei problemi parabolici le discontinuità non si propagano. Nella circostanza suddetta possiamo porre

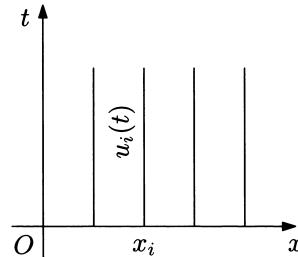
$$u_{0,0} = \frac{1}{2}[\lim_{x \rightarrow 0} f(x) + \lim_{t \rightarrow 0} g_0(t)]$$

analogamente per  $u_{N,0}$ . Oppure può essere più semplice ignorare la discontinuità e considerare solo uno dei due valori.  $\triangleleft$

Invece di discretizzare entrambe le variabili in (9.26), riducendo il problema ad un sistema di equazioni algebriche, possiamo procedere alla discretizzazione di tutte le variabili tranne una, pervenendo così ad un sistema di equazioni differenziali ordinarie. Tale metodo è noto con il nome di *metodo delle linee* (verticali oppure orizzontali).

Discretizziamo la sola variabile  $x$ , e poniamo  $u_i(t) = u(x_i, t)$ . Otteniamo

$$\frac{\partial}{\partial t} u(x_i, t) \equiv \frac{du_i(t)}{dt} \simeq \frac{1}{h^2} [u_{i-1}(t) - 2u_i(t) + u_{i+1}(t)], \quad i = 1, 2, \dots, N-1$$



da cui deduciamo il seguente sistema di equazioni differenziali ordinarie a valori iniziali

$$(9.29) \quad \begin{cases} \bar{u}'_1(t) = \frac{1}{h^2} [\bar{u}_0(t) - 2\bar{u}_1(t) + \bar{u}_2(t)], & \bar{u}_1(0) = f(h) \\ \bar{u}'_2(t) = \frac{1}{h^2} [\bar{u}_1(t) - 2\bar{u}_2(t) + \bar{u}_3(t)], & \bar{u}_2(0) = f(2h) \\ \dots \\ \bar{u}'_{N-1}(t) = \frac{1}{h^2} [\bar{u}_{N-2}(t) - 2\bar{u}_{N-1}(t) + \bar{u}_N(t)], & \bar{u}_{N-1}(0) = f((N-1)h) \\ \bar{u}_0(t) = g_0(t) \\ \bar{u}_N(t) = g_1(t) \end{cases}$$

nelle incognite  $\bar{u}_1(t), \bar{u}_2(t), \dots, \bar{u}_{N-1}(t)$ , con  $\bar{u}_i(t) \simeq u_i(t)$ . Questo sistema può anche venire scritto nella forma vettoriale

$$U' = -\frac{1}{h^2} (TU - v)$$

$$U = \begin{pmatrix} \bar{u}_1(t) \\ \bar{u}_2(t) \\ \vdots \\ \bar{u}_{N-1}(t) \end{pmatrix}, \quad v = \begin{pmatrix} g_0(t) \\ 0 \\ \vdots \\ 0 \\ g_1(t) \end{pmatrix}, \quad T = \begin{pmatrix} 2 & -1 & & & 0 \\ -1 & 2 & -1 & & \\ & -1 & 2 & -1 & \\ & & & -1 & 2 \end{pmatrix}$$

A questo punto risolviamo (9.29) con uno dei metodi visti nel capitolo 8. Per esempio, applicando al sistema (9.29) i metodi di Eulero e dei trapezi otteniamo proprio gli schemi (9.27) e (9.28).

Gli autovalori di  $-(1/h^2)T$  sono noti:

$$\alpha_i = -\frac{4}{h^2} \sin^2 \frac{i\pi}{2N}, \quad i = 1, \dots, N-1$$

Essi sono reali e negativi; inoltre, il più piccolo (in modulo) è  $\alpha_1 \simeq -\pi^2/(h^2 N^2)$ , e il più grande è  $\alpha_{N-1} \simeq -4/h^2$ . Quando intendiamo procedere nell'integrazione di (9.29) sino al superamento del transitorio, il rapporto  $\alpha_{N-1}/\alpha_1 \simeq 4N^2/\pi^2$  rappresenta l'indice di stiffness del sistema. Il sistema risulta pertanto tanto più stiff quanto più grande è  $N$ . Da questo punto di vista i risultati di stabilità relativi ai due metodi (9.27) e (9.28) non devono sorprendere.

Consideriamo infine l'equazione del calore in due dimensioni (spaziali)

$$(9.30) \quad \frac{\partial u}{\partial t} = \gamma \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)$$

dove  $\gamma$  è una costante positiva, definita nel dominio spaziale

$$D = \{(x, y), 0 < x < 1, 0 < y < 1\}$$

per ogni istante  $t > 0$ . A tale equazione associamo le seguenti condizioni iniziale e al bordo:

$$\begin{cases} u(x, y, 0) = 1 & 0 \leq x \leq 1 \\ u(0, y, t) = u(1, y, t) = 0 & 0 \leq y \leq 1 \\ u(x, 0, t) = u(x, 1, t) = 0 & t > 0 \end{cases}$$

con le quali viene definita un'unica soluzione (classica).

Per la determinazione di un'approssimazione discreta dell'incognita  $u \equiv u(x, y, t)$  utilizziamo ancora il metodo delle linee. Generalizzando quindi il procedimento seguito nel caso precedente, discretizziamo le sole variabili spaziali  $x, y$  (vedi figura 9.15) e poniamo  $u(x_i, y_j, t) \equiv u_{i,j}(t)$ . Otteniamo

$$(9.31) \quad \begin{cases} \frac{d\bar{u}_{i,j}(t)}{dt} = \gamma \left( \frac{\bar{u}_{i+1,j}(t) - 2\bar{u}_{i,j}(t) + \bar{u}_{i-1,j}(t)}{h^2} + \frac{\bar{u}_{i,j+1}(t) - 2\bar{u}_{i,j}(t) + \bar{u}_{i,j-1}(t)}{k^2} \right) \\ \bar{u}_{i,j}(0) = 1 & i = 1, \dots, N-1 \\ \bar{u}_{0,j}(t) = \bar{u}_{N,j}(t) = 0 & j = 1, \dots, M-1 \\ \bar{u}_{i,0}(t) = \bar{u}_{i,M}(t) = 0 & t > 0 \end{cases}$$

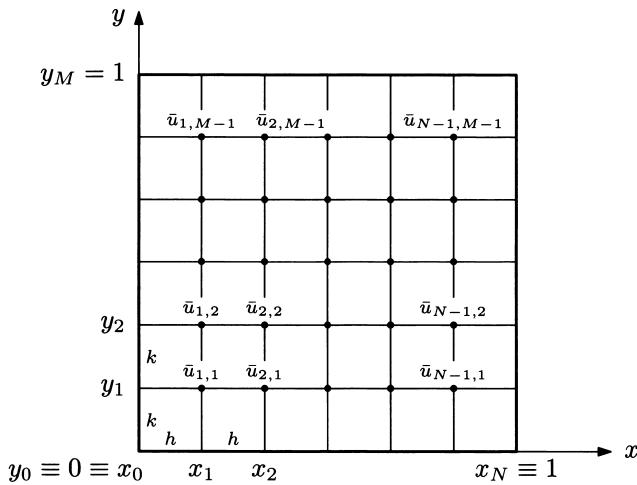


Figura 9.15

dove con  $\bar{u}_{i,j}(t)$  ( $\simeq u_{i,j}(t)$ ) denotiamo le nuove (funzioni) incognite generate dall'approssimazione delle derivate con le corrispondenti formule alle differenze finite.

Introducendo il nuovo vettore

$$\begin{aligned} \mathbf{u}(t) = & (\bar{u}_{1,1}(t), \bar{u}_{1,2}(t), \dots, \bar{u}_{1,M-1}(t), \bar{u}_{2,1}(t), \bar{u}_{2,2}(t), \dots, \bar{u}_{2,M-1}(t), \\ & \dots, \bar{u}_{N-1,1}(t), \bar{u}_{N-1,2}(t), \dots, \bar{u}_{N-1,M-1}(t))^T \end{aligned}$$

e la matrice

$$A = \left( \begin{array}{ccccccccc} c & k^{-2} & \overbrace{0 & \dots & 0}^{M-3} & h^{-2} & 0 & \dots & & & \\ k^{-2} & & 0 & & & & & & \\ 0 & & c & k^{-2} & 0 & \dots & 0 & h^{-2} & 0 & \dots \\ \vdots & & k^{-2} & & & & & & & \\ 0 & & & & & & & & & \\ h^{-2} & & 0 & & & & & & & \\ & & & & & & & & & \\ & & & & & & & & & \end{array} \right) \quad c = -2(h^{-2} + k^{-2})$$

simmetrica, di ordine  $(M - 1)(N - 1)$ , con sole 5 diagonali di elementi non nulli, possiamo esprimere il sistema (9.31) nella forma più compatta

$$\begin{cases} \frac{du(t)}{dt} = \gamma Au(t) + v(t), & t > 0 \\ u(0) = 1 \end{cases}$$

dove il vettore  $v(t)$  contiene termini noti (i dati assegnati al contorno ad ogni istante  $t > 0$ ).

Questo sistema (lineare a coefficienti costanti) di equazioni differenziali ordinarie a valori iniziali ha pressoché lo stesso grado di stiffness del corrispondente problema unidimensionale. È pertanto indispensabile integrare tale sistema con metodi di tipo implicito.

L'applicazione diretta di un metodo隐式的, per esempio la formula dei trapezi, richiede tuttavia, ad ogni passo nella direzione del tempo, la risoluzione di un sistema lineare di ordine  $(M - 1) \cdot (N - 1)$ , con sole 5 diagonali di elementi non nulli ma con due diagonali “distanti” dalle 3 centrali. Tale distanza rende “costosa” la risoluzione numerica del sistema, effettuata per esempio con il metodo di Gauss.

Il *metodo delle direzioni alternate* (ADI) che ora descriveremo ci consente di ridurre notevolmente il costo di risoluzione, mantenendo tuttavia le caratteristiche di stabilità degli schemi impliciti.

Utilizziamo dapprima il metodo di Eulero (implicito) all'indietro nella direzione  $x$ , e quello di Eulero esplicito nella direzione  $y$ , per passare dall'istante  $t = t_n$ , all'istante  $t = t_{n+1/2} = t_n + k/2$ :

$$(9.32) \quad u_{i,j}^{n+1/2} = u_{i,j}^n + \gamma \frac{k}{2} \left( \frac{u_{i+1,j}^{n+1/2} - 2u_{i,j}^{n+1/2} + u_{i-1,j}^{n+1/2}}{h^2} + \frac{u_{i,j+1}^n - 2u_{i,j}^n + u_{i,j-1}^n}{k^2} \right)$$

$$i = 1, \dots, N - 1$$

$$j = 1, \dots, M - 1$$

Successivamente avanziamo da  $t = t_{n+1/2}$  a  $t = t_{n+1} = t_n + k$  utilizzando il metodo di Eulero all'indietro nella direzione  $y$  e quello di Eulero esplicito nella direzione  $x$ :

$$(9.33) \quad u_{i,j}^{n+1} = u_{i,j}^{n+1/2} + \gamma \frac{k}{2} \left( \frac{u_{i+1,j}^{n+1/2} - 2u_{i,j}^{n+1/2} + u_{i-1,j}^{n+1/2}}{h^2} + \frac{u_{i,j+1}^{n+1} - 2u_{i,j}^{n+1} + u_{i,j-1}^{n+1}}{k^2} \right)$$

$$i = 1, \dots, N - 1$$

$$j = 1, \dots, M - 1$$

È possibile dimostrare che la combinazione delle formule (9.32) e (9.33) rappresenta un metodo di ordine  $2(\dagger)$ , ossia  $u_{i,j}^{n+1}$  è un'approssimazione del 2° ordine, incondizionatamente stabile.

(†) E ciò nonostante che i metodi utilizzati (di Eulero) siano di ordine 1.

Per quanto riguarda il costo complessivo di ogni singolo passo  $k$  nella direzione  $t$ , osserviamo dapprima che il sistema (9.32) può essere interpretato come insieme di  $M - 1$  sistemi di ordine  $N - 1$

$$(9.34) \quad Au_j^{n+1/2} = v_j^n, \quad j = 1, 2, \dots, M - 1$$

nelle incognite

$$u_j^{n+1/2} = (u_{1j}^{n+1/2}, u_{2j}^{n+1/2}, \dots, u_{N-1,j}^{n+1/2})^T$$

La matrice

$$A = I + \gamma \frac{k}{2h^2} T, \quad T = \begin{pmatrix} 2 & -1 & & & & 0 \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & 0 & -1 & 2 & \\ & & & & -1 & \end{pmatrix}$$

è simmetrica, tridiagonale, a diagonale dominante. Ciascun sistema (9.34) può essere risolto con il metodo di Gauss senza pivoting né scaling con sole  $3(N - 2)$  moltiplicazioni e addizioni e  $2N - 3$  divisioni. Analogamente per i sistemi (9.33).

Pertanto con il metodo ADI il costo di ogni singolo passo nella direzione del tempo richiede sostanzialmente  $10MN$  operazioni.

Il metodo ADI appartiene a quella classe di metodi denominati “di tipo splitting”, che trovano applicazione anche nella risoluzione di problemi di tipo iperbolico e di tipo ellittico (vedere [9.4]).

### 9.3.4 Consistenza, stabilità e convergenza degli schemi alle differenze finite per problemi a valori iniziali

Consideriamo nella regione  $\mathcal{R} = D \times (0 < t < \infty)$  un’equazione differenziale rappresentata simbolicamente da

$$(9.35) \quad L u = f$$

Indichiamo con  $\mathcal{R}_\Delta$  il reticolo di punti costruito in  $\mathcal{R}$ . Con  $L_\Delta$  denotiamo la discretizzazione scelta dell’operatore  $L$ . Per esempio, nel caso dell’equazione del calore abbiamo

$$L u = \frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2}$$

e, se utilizziamo lo schema (9.27),

$$L_\Delta u_{i,j} = \frac{u_{i,j+1} - u_{i,j}}{k} - \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h^2}$$

Per ogni funzione  $v$  “sufficientemente regolare” ( $\dagger$ ) in  $\mathcal{R}$  definiamo l’*errore locale (unitario) di troncamento* nel punto  $(P_i, t_j) \in \mathcal{R}_\Delta$ :

$$(9.36) \quad t v(P_i, t_j) = L v(P_i, t_j) - L_\Delta v(P_i, t_j)$$

Nel caso del metodo (9.27), con la notazione  $L_\Delta v(P_i, t_j)$  indichiamo l’espressione

$$\frac{v(x_i, t_j + k) - v(x_i, t_j)}{k} - \frac{v(x_{i-1}, t_j) - 2v(x_i, t_j) + v(x_{i+1}, t_j)}{h^2}$$

Nel paragrafo 9.3.2 abbiamo definito l’errore locale di troncamento e introdotto il concetto di consistenza di uno schema numerico in un generico punto della griglia  $\mathcal{R}_\Delta$ . Qui preferiamo invece richiedere qualcosa in più, e cioè considerare il vettore degli errori di troncamento in tutti i nodi di  $\mathcal{R}_\Delta$  situati sul generico livello temporale  $\tau = t_M$ , ed imporre che la norma di tale vettore converga a zero quando i parametri di discretizzazione tendono a zero.

**Definizione 9.1.** *Lo schema alle differenze*

$$(9.37) \quad L_\Delta u_{i,j} = f(P_i, t_j)$$

è *consistente*, nella norma scelta, con l’equazione (9.35) se, fissato un generico istante  $\tau > 0$ , e considerato un reticolo  $\mathcal{R}_\Delta$  con passo temporale  $k = \tau/M$ ,  $M$  intero positivo, cosicché  $\tau = t_M$  e  $(P_i, \tau) \in \mathcal{R}_\Delta$ , per il vettore  $t_\tau v = \{t v(P_i, t_M)\}$  definito in (9.36) si ha

$$\|t_\tau v\| \rightarrow 0$$

quando i parametri di discretizzazione  $(h, \dots, k)$  tendono a zero ( $\ddagger$ ) in modo completamente arbitrario ovvero legati tra di loro da un ben preciso rapporto. In questo secondo caso lo schema è definito condizionatamente consistente. Se inoltre il predetto errore ha un comportamento del tipo  $O(h^p) + \dots + O(k^q)$  allora diciamo che l’ordine di consistenza è  $(p, \dots, q)$ .

Se come funzione  $v$  prendiamo una soluzione  $u$  dell’equazione  $L u = f$ , allora osserviamo che l’errore locale (unitario) di troncamento altro non è che il residuo generato dallo schema numerico, scritto nella forma (9.37), quando in quest’ultimo inseriamo una soluzione dell’equazione differenziale. Tale definizione è del tutto analoga a quella associata ai metodi numerici per le equazioni differenziali ordinarie.

Riprendiamo l’equazione del calore; in questo esempio, supponendo  $\tau = t_M$  fisso, abbiamo

$$\begin{aligned} t v(x_i, \tau) &= \frac{\partial v(x_i, \tau)}{\partial t} - \frac{\partial^2 v(x_i, \tau)}{\partial x^2} - \\ &\quad \left( \frac{v(x_i, \tau + k) - v(x_i, \tau)}{k} - \frac{v(x_{i-1}, \tau) - 2v(x_i, \tau) + v(x_{i+1}, \tau)}{h^2} \right) \\ &= O(k) + O(h^2) \end{aligned}$$

( $\dagger$ ) In modo da poter effettuare gli sviluppi in serie di Taylor necessari.

( $\ddagger$ ) In particolare  $k \rightarrow 0$  quando  $M \rightarrow \infty$ .

dove le costanti presenti nei due termini  $O(\cdot)$  sono indipendenti da  $h, k$  e dall'indice  $i$ . Infatti sviluppando  $v(x_i, \tau + k)$ ,  $v(x_{i-1}, \tau)$  e  $v(x_{i+1}, \tau)$  nell'intorno del punto  $(x_i, \tau)$  otteniamo

$$t v(x_i, \tau) = -\frac{k}{2} \frac{\partial^2 v(x_i, \zeta_i)}{\partial t^2} + \frac{h^2}{24} \left[ \frac{\partial^4 v(\xi_i, \tau)}{\partial x^4} + \frac{\partial^4 v(\eta_i, \tau)}{\partial x^4} \right]$$

dove  $\tau < \zeta_i < \tau + k$ ,  $x_i - h < \xi_i < x_i$  e  $x_i < \eta_i < x_i + h$ , da cui segue, supponendo che le predette derivate esistano e siano (uniformemente) limitate, la maggiorazione

$$\| t_\tau v \| \leq c_1 k + c_2 h^2$$

con costanti  $c_1$  e  $c_2$  dipendenti unicamente da  $v$  e da  $\tau$ . Lo schema (9.27) risulta pertanto incondizionatamente consistente.

Se le condizioni iniziali o al bordo associate alla (9.35) contengono derivate che devono essere discretizzate, allora occorre determinare anche l'errore di troncamento che lo schema numerico produce in quei punti della frontiera in cui esso viene modificato a causa della discretizzazione che le suddette derivate subiscono. Per esempio, nel caso della precedente equazione del calore, se la condizione al bordo  $u(0, t) = g_0(t)$  nel problema (9.26) viene sostituita da  $u_x(0, t) = g_0(t)$ , possiamo discretizzare quest'ultima utilizzando una delle due formule seguenti:

$$\begin{aligned} u_x(0, t_j) &= \frac{u(x_1, t_j) - u(x_0, t_j)}{h} + O(h) \\ u_x(0, t_j) &= \frac{u(x_1, t_j) - u(x_{-1}, t_j)}{2h} + O(h^2) \end{aligned}$$

Utilizzando la prima otteniamo

$$u(x_1, t_j) = u(x_0, t_j) + h g_0(t_j) + O(h^2)$$

da cui segue

$$(9.38) \quad u_{1,j} = u_{0,j} + h g_0(t_j)$$

mentre dalla seconda, che richiede l'introduzione del livello fittizio  $x_{-1} = x_0 - h$ , abbiamo

$$u(x_{-1}, t_j) = u(x_1, t_j) - 2h g_0(t_j) + O(h^3)$$

ovvero

$$(9.39) \quad u_{-1,j} = u_{1,j} - 2h g_0(t_j)$$

La (9.38) o la (9.39) viene infine inserita nello schema (9.27), quando  $i = 1$  nel caso della prima e quando poniamo  $i = 0$  nel caso della seconda.

Scegliendo la (9.39), la più precisa delle due, quando  $i = 0$  otteniamo

$$L_\Delta u_{0,j} = \frac{u_{0,j+1} - u_{0,j}}{k} - \frac{-2u_{0,j} + 2u_{1,j} - 2h g_0(t_j)}{h^2}$$

da cui seguono le stime

$$\mathbf{t} v(x_0, \tau) = O(h) + O(k)$$

e

$$\| \mathbf{t}_\tau v \| = O(h) + O(k)$$

La discretizzazione della condizione al bordo mediante la formula (9.39) provoca pertanto un aumento dell'errore di troncamento da  $O(k) + O(h^2)$  a  $O(k) + O(h)$ .

Se utilizziamo invece la (9.38) lo schema finale non risulta più consistente.

Quando il metodo numerico proposto è rappresentato, per esempio, da uno schema esplicito a due livelli associato ad un problema con valori iniziali e al bordo, definito da una relazione del tipo

$$u_{j+1} = Au_j + b_j$$

dove  $u_j$  è il vettore contenente tutte le incognite  $\{u_{i,j}\}$  al livello temporale  $t_j$ , l'errore locale di troncamento, inteso come il vettore  $\mathbf{t}_\tau u$  dei corrispondenti errori al tempo  $t = \tau \equiv t_M$ , viene definito come il residuo, diviso per il passo  $k$ , prodotto dal metodo quando in esso inseriamo la soluzione del problema differenziale, ovvero, posto  $u_\tau = \{u(P_i, \tau)\}$  e  $b_\tau = \{b(P_i, \tau)\}$ ,

$$k \mathbf{t}_\tau u = u_{\tau+1} - Au_\tau - b_\tau$$

In analogia con la corrispondente definizione data per i metodi numerici per la risoluzione di equazioni differenziali ordinarie,  $\mathbf{t}_\tau u$  rappresenta l'errore locale *unitario* di troncamento.

**Definizione 9.2.** Sia  $u(P, t)$  la soluzione del problema (9.35), e sia  $u_{P,t}$  la soluzione (discreta) del corrispondente schema (9.37). Il metodo (9.37) è definito convergente nella norma scelta se, fissato un generico istante  $t = \tau > 0$  e scelto  $k = \tau/M$ , per il vettore  $e_\tau = \{u(P_i, \tau) - u_{P_i, \tau}\}$  risulta

$$\|e_\tau\| \rightarrow 0$$

quando i parametri di discretizzazione  $h, \dots, k$  tendono a zero. Esso ha inoltre ordine  $(p, \dots, q)$ , se il predetto errore si comporta come  $O(h^p) + \dots + O(k^q)$ .

La convergenza può risultare condizionata oppure incondizionata. Osserviamo inoltre che uno schema di calcolo alle differenze potrebbe rivelarsi consistente ma non convergente. Infatti la consistenza di uno schema numerico con un problema differenziale assicura che il primo confluisce sul secondo quando i parametri di discretizzazione tendono a zero (eventualmente in modo condizionato). Ciò non significa però che anche la soluzione discreta prodotta dallo schema alle differenze finite converga alla soluzione del problema differenziale. Affinché questo si verifichi occorre che lo schema sia anche stabile. Sussiste infatti il seguente risultato fondamentale, noto con il nome di Teorema di Lax (talvolta di Lax-Richtmeyer), del tutto analogo a quello presentato nel teorema 8.4 per la risoluzione di equazioni differenziali ordinarie a valori iniziali con metodi multistep. Esso stabilisce un importante legame tra consistenza, stabilità e convergenza, supponendo ovviamente che la norma scelta sia sempre la stessa.

**Teorema 9.1.** ([9.3, cap. 3]). *Sia dato un problema lineare a valori iniziali<sup>(†)</sup> ben posto. Per ogni schema consistente la stabilità è condizione necessaria e sufficiente per la convergenza; inoltre, l'ordine di convergenza è pari a quello di consistenza.*

### 9.3.5 Equazioni di tipo ellittico

Un'equazione importante della Fisica Matematica è quella di Poisson

$$(9.40) \quad -\Delta u = f$$

alla quale vengono associate delle condizioni sul contorno  $\Gamma$  del dominio di interesse  $D$ :

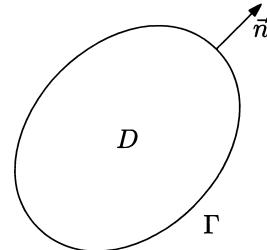
$$u = g \quad (\text{condizione di Dirichlet})$$

oppure

$$\frac{du}{d\vec{n}} = g \quad (\text{condizione di Neumann})$$

oppure

$$au + b\frac{du}{d\vec{n}} = g \quad (\text{condizione di tipo misto})$$



dove  $du/d\vec{n}$  denota la derivata normale al bordo  $\Gamma$ .

La funzione  $f$  è assegnata e  $u$  rappresenta l'incognita. Quando  $f \equiv 0$  la (9.40) prende il nome di equazione di Laplace.

Ricordiamo (vedere ad esempio [9.18, pag. 42], [9.20, §5.6]) che se il termine noto  $f$  in (9.40) è di classe  $C^m(D)$  (oppure analitico) allora tali risultano anche le soluzioni  $u$  in ogni punto del dominio (aperto)  $D$ . Questo comportamento è tipico delle equazioni ellittiche. Tuttavia, quando alla (9.40) aggiungiamo, per esempio, la condizione di Dirichlet, la soluzione  $u$  del sistema non manterrà in generale le stesse proprietà di regolarità nel chiuso  $D \cup \Gamma$ , a meno che la funzione  $g$  e la curva  $\Gamma$  non siano esse stesse sufficientemente regolari.

Consideriamo l'equazione  $-\Delta u = 1$  in un dominio (aperto)  $D$  di forma indicata in figura 9.16, con condizione al bordo  $u = 0$ . La soluzione di tale problema è analitica in ogni punto  $P \in D$ ; ma quando  $P \rightarrow O$  la funzione  $u$  si comporta come il termine

$$|P - O|^{2/3}$$

Nella costruzione di uno schema alle differenze finite ci limitiamo al problema di Dirichlet, ossia allo studio di

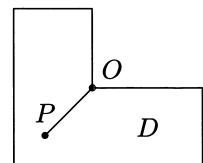


Figura 9.16

$$\begin{cases} -\Delta u(x, y) = f(x, y) & \text{in } D \\ u(x, y) = g(x, y) & \text{su } \Gamma \end{cases}$$

(†) Con eventuali condizioni (lineari) al contorno.

Consideriamo inoltre il dominio rettangolare con reticolo a maglie quadrate di figura 9.17 e approssimiamo l'operatore  $\Delta \equiv \nabla^2$  con l'operatore discreto  $\nabla_5^2$  introdotto a pagina 318.

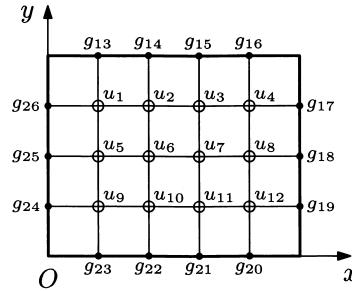


Figura 9.17

Per semplificare le notazioni numeriamo i nodi del reticolo, e quindi le incognite  $u(x, y)$  corrispondenti, come in figura; otteniamo il seguente sistema lineare

$$\left\{ \begin{array}{lll} 4u_1 - u_2 & - u_5 & = h^2 f_1 + g_{13} + g_{26} \\ -u_1 + 4u_2 - u_3 & - u_6 & = h^2 f_2 + g_{14} \\ -u_2 + 4u_3 - u_4 & - u_7 & = h^2 f_3 + g_{15} \\ -u_3 + 4u_4 & - u_8 & = h^2 f_4 + g_{16} + g_{17} \\ -u_1 & + 4u_5 - u_6 & - u_9 & = h^2 f_5 + g_{25} \\ -u_2 & - u_5 + 4u_6 - u_7 & - u_{10} & = h^2 f_6 \\ -u_3 & - u_6 + 4u_7 - u_8 & - u_{11} & = h^2 f_7 \\ -u_4 & - u_7 + 4u_8 & - u_{12} & = h^2 f_8 + g_{18} \\ -u_5 & & + 4u_9 - u_{10} & = h^2 f_9 + g_{23} + g_{24} \\ -u_6 & & - u_9 + 4u_{10} - u_{11} & = h^2 f_{10} + g_{22} \\ -u_7 & & - u_{10} + 4u_{11} - u_{12} & = h^2 f_{11} + g_{21} \\ -u_8 & & - u_{11} + 4u_{12} & = h^2 f_{12} + g_{19} + g_{20} \end{array} \right.$$

con matrice dei coefficienti

$$A = \left( \begin{array}{cccc|cccc|cccc} 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 4 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ \hline -1 & 0 & 0 & 0 & 4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & 0 & 0 & 0 & -1 \\ \hline 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 4 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & -1 & 4 \end{array} \right)$$

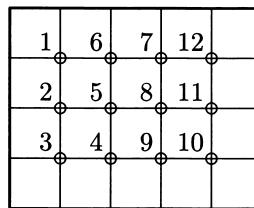


Figura 9.18

simmetrica, a diagonale dominante in senso debole<sup>(†)</sup>, non singolare. È inoltre possibile dimostrare che essa è anche definita positiva. Osserviamo infine che la matrice  $A$  ha la struttura tridiagonale a blocchi

$$\begin{pmatrix} C & B & O \\ B & C & B \\ O & B & C \end{pmatrix}$$

con

$$C = \begin{pmatrix} 4 & -1 & 0 & 0 \\ -1 & 4 & -1 & 0 \\ 0 & -1 & 4 & -1 \\ 0 & 0 & -1 & 4 \end{pmatrix} \quad \text{e} \quad B = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$$

La struttura di  $A$  è determinata dalla numerazione dei nodi del reticolo. Se numeriamo, per esempio, tali nodi come in figura 9.18 otteniamo una matrice

$$A = \left( \begin{array}{cccc|cccc|ccc|ccc} 4 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & -1 & 4 & -1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & -1 & 4 & -1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 4 & -1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 4 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & -1 & 4 \end{array} \right)$$

<sup>(†)</sup> Una matrice  $A$  è definita a diagonale dominante in senso debole quando

$$|a_{ii}| \geq \sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}|, \quad i = 1, \dots, n; \quad a_{ij} = (A)_{ij}$$

diversa dalla precedente, ma che ha ancora la struttura tridiagonale a blocchi

$$\begin{pmatrix} C & B & O & O \\ B & C & B & O \\ O & B & C & B \\ O & O & B & C \end{pmatrix}$$

con

$$C = \begin{pmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 4 \end{pmatrix} \quad \text{e} \quad B = \begin{pmatrix} 0 & 0 & -1 \\ 0 & -1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$$

Numerando invece a caso i nodi del reticolo, la corrispondente matrice  $A$  non presenta in generale le predette caratteristiche.

Quando il numero di nodi non è troppo grande il sistema lineare può essere risolto con il metodo di Gauss senza pivoting; altrimenti un processo iterativo come il metodo SOR è preferibile (consultare il paragrafo 3.3.3).

Le matrici  $A$  generate dalla discretizzazione di equazioni di tipo ellittico con il metodo delle differenze finite spesso esibiscono strutture a blocchi

$$A = \begin{pmatrix} A_{11} & \dots & A_{1N} \\ \vdots & \ddots & \vdots \\ A_{N1} & \dots & A_{NN} \end{pmatrix}, \quad A_{ij} \in \mathbb{R}^{n_i \times n_j}$$

dove le  $A_{ij}$  hanno una struttura molto semplice, diagonale o tridiagonale per esempio. Inoltre i blocchi diagonale  $A_{ii}$  sono quadrati e non singolari. In tali situazioni possiamo generalizzare i metodi iterativi presentati nel paragrafo 3.3.

Per esempio, il metodo di Jacobi può essere riformulato nella versione a blocchi:

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

dove  $b_i$ ,  $x_i^{(k)} \in \mathbb{R}^{n_i}$ . L'incognita  $x_i^{(k+1)}$  viene determinata utilizzando la fattorizzazione  $A_{ii} = L_iU_i$ . Il nuovo metodo risulta competitivo con quello standard solo quando  $A_{ii}$  ha una struttura molto semplice, tridiagonale per esempio.

Nel caso del metodo di Gauss-Seidel abbiamo invece

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

Anche per le equazioni di tipo ellittico possiamo introdurre i concetti di consistenza, stabilità e convergenza (vedi ad esempio [1, pag. 514]). In particolare, nel caso del problema (9.40) in un dominio rettangolare, risolto utilizzando lo schema di discretizzazione

$\nabla^5_2$ , è possibile dimostrare ([1, pag. 450]) che quando la soluzione  $u(x, y)$  ha le derivate quarte continue e limitate nel dominio  $D$

$$\max_{i,j} |u(x_i, y_j) - u_{i,j}| = O(h^2)$$

## 9.4 Metodi dei residui pesati

La tecnica delle differenze finite è stata utilizzata nei paragrafi precedenti per *discretizzare* gli operatori presenti nel modello matematico e pervenire quindi a delle equazioni alle differenze la cui soluzione viene presa come approssimazione discreta della funzione incognita. Con i metodi dei residui pesati che ora ci accingiamo a presentare, ci proponiamo invece di approssimare direttamente la soluzione del modello matematico con una combinazione lineare (finita) di funzioni base opportunamente scelte.

Consideriamo per esempio un'equazione differenziale di tipo ellittico

$$(9.41) \quad L u(x) = f(x), \quad x \in D \subset \mathbb{R}^d$$

dove  $L$  è un operatore lineare e  $f(x)$  una funzione assegnata. La soluzione  $u(x)$ , che qui e nel seguito supponiamo classica, è anche soggetta a delle condizioni sul contorno  $\Gamma$  del dominio  $D$ , che per semplicità in questo paragrafo assumiamo di tipo Dirichlet. Equazioni con condizioni al contorno di tipo diverso verranno considerate nel paragrafo successivo. Supponiamo inoltre che il problema sia ben posto, e denotiamo con  $X \subset L^2(D)$  uno spazio lineare (dotato di norma  $\|\cdot\|$ ) “opportuno” (vedi ad esempio [9.24], [9.28]) contenente la soluzione  $u(x)$

Sia  $\{X_M, M = 1, 2, \dots\}$  una successione di sottospazi lineari di  $X$  di dimensione finita  $M$ ; in generale  $u \notin X_M$ . Fissato  $M$ , ci proponiamo di scegliere in  $X_M$  un'approssimazione dell'incognita  $u(x)$ <sup>(†)</sup>. La successione  $\{X_M\}$  prescelta deve godere della seguente proprietà: prefissata una qualsiasi tolleranza  $\varepsilon > 0$ , esiste un sottospazio  $X_{M_0}$  ed un suo elemento  $u_{M_0}$  tale che  $\|u - u_{M_0}\| \leq \varepsilon$ . Negli esempi che considereremo in questo paragrafo e nel successivo, tutti gli spazi  $X_M$  scelti godranno di questa proprietà.

Ovviamente sarebbe auspicabile poter scegliere gli spazi  $X$  e  $X_M$  in modo tale che i loro elementi soddisfino a priori la condizione (di Dirichlet) imposta sul bordo  $\Gamma$ . A tale fine di solito si scelgono dei sottospazi lineari “traslati”, denominati *sottospazi affini*, che con abuso di notazione definiamo come  $W = u_0 + V \subset X$  e  $W_M = u_0 + V_M \subset X_M$ , dove  $u_0 \in X$  (o  $u_0 \in X_M$ ) è una funzione nota che su  $\Gamma$  assume i valori preassegnati, mentre  $V$  e  $V_M$  sono spazi lineari, di dimensione infinita il primo e finita il secondo, i cui elementi assumono valori nulli su  $\Gamma$ .

In questo caso, denotando con  $\varphi_1(x), \dots, \varphi_M(x)$  una base di  $V_M$ , l'approssimante assumerà quindi la forma

$$(9.42) \quad u_M(x) = u_0(x) + \sum_{i=1}^M c_i \varphi_i(x) \quad (\varphi_i(x) = 0, \quad x \in \Gamma)$$

---

(†) O eventualmente di  $\bar{u}(x) = u(x) - u_0(x)$ , con  $u_0(x) \in X$  funzione nota.

e soddisferà automaticamente le condizioni al bordo assegnate qualunque siano i valori dei coefficienti  $\{c_i\}$ . Questi ultimi dovranno quindi essere determinati in modo da approssimare “bene”, secondo il criterio scelto, la funzione incognita  $\bar{u} = u - u_0$ .

Per quanto riguarda la precisione fornita da  $u_M(x)$ , di fondamentale importanza è la scelta dei sottospazi  $X_M$  ( $V_M$ ) e delle corrispondenti funzioni base  $\{\varphi_i(x)\}$ . Se si hanno informazioni sul comportamento e regolarità della soluzione  $u(x)$ , è possibile individuare dei sottospazi e delle basi che consentano non solo di raggiungere la precisione desiderata con un valore di  $M$  più piccolo, ma anche di determinare i coefficienti  $\{c_i\}$  in modo efficiente e con un numero minore di operazioni aritmetiche.

Supponendo pertanto di aver scelto l'approssimante  $u_M(x)$  più conveniente, di forma (9.42), consideriamo il residuo

$$R_M(x) = L u_M(x) - f(x) \not\equiv 0$$

Sceglieremo poi una seconda successione di sottospazi lineari  $Y_M$  di dimensione  $M$ ,  $M = 1, 2, \dots$ , di uno spazio lineare  $Y$  non necessariamente coincidente con  $X$ , ed una base  $\psi_1(x), \psi_2(x), \dots, \psi_M(x)$  di  $Y_M$ . Definiamo il prodotto scalare  $\langle g, h \rangle = \int_D g(x)h(x) dx$ . Con il *metodo dei residui pesati* i coefficienti incogniti  $\{c_i\}$  vengono determinati imponendo la relazione di ortogonalità.

$$\langle R_M, \psi_j \rangle = 0, \quad j = 1, 2, \dots, M$$

e quindi risolvendo il sistema lineare

$$(9.43) \quad \begin{pmatrix} \langle L \varphi_1, \psi_1 \rangle & \langle L \varphi_2, \psi_1 \rangle & \dots & \langle L \varphi_M, \psi_1 \rangle \\ \langle L \varphi_1, \psi_2 \rangle & \langle L \varphi_2, \psi_2 \rangle & \dots & \langle L \varphi_M, \psi_2 \rangle \\ \dots & \dots & \dots & \dots \\ \langle L \varphi_1, \psi_M \rangle & \langle L \varphi_2, \psi_M \rangle & \dots & \langle L \varphi_M, \psi_M \rangle \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_M \end{pmatrix} = \begin{pmatrix} \langle f - L u_0, \psi_1 \rangle \\ \langle f - L u_0, \psi_2 \rangle \\ \vdots \\ \langle f - L u_0, \psi_M \rangle \end{pmatrix}$$

Con il criterio di *Galerkin* prendiamo

$$X \equiv Y, \quad Y_M \equiv X_M \quad \text{ovvero } \psi_j(x) = \varphi_j(x), \quad j = 1, \dots, M$$

Nel caso dei *minimi quadrati* invece

$$\psi_j(x) = L \varphi_j(x), \quad j = 1, \dots, M$$

Tale scelta conduce alla minimizzazione della quantità

$$I(c) = \langle R_M, R_M \rangle = \int_D R_M^2(x) dx, \quad c = (c_1, \dots, c_M)^T$$

infatti,

$$\frac{\partial I(c)}{\partial c_j} = 2 \int_D R_M(x) \frac{\partial R_M(x)}{\partial c_j} dx = 0, \quad j = 1, \dots, M$$

donde

$$\psi_j(x) = \frac{\partial R_M(x)}{\partial c_j} = L\varphi_j(x)$$

Un terzo criterio è rappresentato dal cosiddetto *metodo di collocazione*. Scelti  $M$  punti distinti  $x_j$  nel dominio  $D$ , imponiamo le condizioni

$$R_M(x_j) = 0, \quad j = 1, \dots, M$$

Anche quest'ultimo criterio può essere posto nella forma

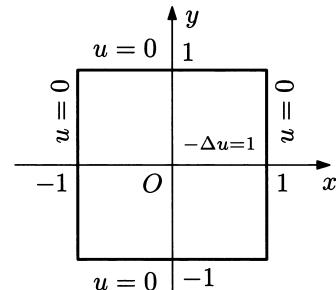
$$\int_D R_M(x)\psi_j(x) dx = 0$$

con  $\psi_j(x) = \delta(x - x_j)$ , dove  $\delta(s)$  è la nota funzione di Dirac.

Dei metodi descritti il più usato è certamente quello di Galerkin. Pertanto nelle pagine che seguono ci limiteremo a descrivere l'applicazione di quest'ultimo ad alcuni problemi.

► **Esempio 9.1.** Consideriamo il seguente problema di Dirichlet per l'equazione di Poisson

$$(9.44) \quad \begin{cases} -(u_{xx} + u_{yy}) = 1 \\ u = 0 \quad \text{per } x = \pm 1 \quad \text{e} \quad y = \pm 1 \end{cases}$$



Osserviamo preliminarmente che  $u(-x, y) = u(x, y)$  e  $u(x, -y) = u(x, y)$ . Poiché il dominio del problema è un quadrato, che può quindi essere interpretato come prodotto cartesiano di due segmenti, e la soluzione ha le derivate di ordine qualsiasi continue nel dominio di integrazione, come approssimante scegliamo la seguente funzione:

$$u_M(x, y) = \sum_{i=1}^M \sum_{j=1}^M c_{ij} \underbrace{\cos\left((2i-1)\frac{\pi}{2}x\right) \cos\left((2j-1)\frac{\pi}{2}y\right)}_{\varphi_{ij}(x, y)}$$

che, oltre ad avere le stesse simmetrie della soluzione  $u$ , soddisfa a priori le condizioni al bordo. In questo caso particolare la (9.42) viene espressa mediante una doppia sommatoria che, operando una separazione delle variabili (indipendenti), semplificherà notevolmente la determinazione dei coefficienti incogniti.

Con il criterio di Galerkin imponiamo le condizioni

$$(9.45) \quad \langle R_M, \varphi_{mn} \rangle = 0, \quad m = 1, \dots, M, \quad n = 1, \dots, M$$

L'ortogonalità della base  $\{\cos(k\pi x/2)\}$  nell'intervallo  $(-1, 1)$ , ovvero

$$\int_{-1}^1 \cos\left(k\frac{\pi}{2}x\right) \cos\left(j\frac{\pi}{2}x\right) dx = 0, \quad k \neq j$$

trasforma le (9.45) in un sistema diagonale la cui soluzione è

$$c_{ij} = c_{ji} = \left(\frac{8}{\pi^2}\right)^2 \frac{(-1)^{i+j}}{(2i-1)(2j-1)[(2i-1)^2 + (2j-1)^2]}$$

Una scelta alternativa, avente le stesse simmetrie della soluzione  $u(x, y)$ , potrebbe essere

$$u_M(x, y) = \sum_{i=1}^M \sum_{j=1}^M c_{ij} (1-x^2)^i (1-y^2)^j$$

ma in questo caso il corrispondente sistema (9.43) risulterebbe denso.

Se volessimo invece determinare un'approssimante  $u_M$  applicando il criterio di collocazione, potremmo scegliere  $(M+1)^2$  punti nel dominio del problema, per esempio  $(x_l, y_m)$ ,  $l, m = 0, \dots, M$ , dove  $\{x_l\}$  e  $\{y_m\}$  coincidono con gli  $M+1$  nodi della formula di quadratura di Gauss-Lobatto (7.13), porre

$$u_M(x, y) = \sum_{j=0}^M \sum_{i=0}^M c_{ij} l_i(x) l_j(y)$$

ovvero, tenendo conto della condizione al bordo,

$$u_M(x, y) = \sum_{j=1}^{M-1} \sum_{i=1}^{M-1} c_{ij} l_i(x) l_j(y)$$

dove  $\{l_i(x)\}$  e  $\{l_j(y)\}$  denotano i polinomi fondamentali di Lagrange associati ai predetti insiemi di nodi, collocare l'equazione nei nodi interni  $(x_l, y_m)$ ,  $l, m = 1, \dots, M-1$  e quindi risolvere il sistema

$$-\left(\frac{\partial^2 u_M(x_l, y_m)}{\partial x^2} + \frac{\partial^2 u_M(x_l, y_m)}{\partial y^2}\right) = 1, \quad l, m = 1, \dots, M-1$$

nelle incognite  $\{c_{ij}\}$ . ◀

► **Esempio 9.2.** Questo secondo esempio si riferisce ad un problema ai limiti per un'equazione differenziale ordinaria:

$$(9.46) \quad \begin{cases} u''(x) - u'(x) + u(x) = -\cos(x + \pi/4), & 0 < x < \pi \\ u(0) = \sqrt{2}/2 \\ u(\pi) = -\sqrt{2}/2 \end{cases}$$

la cui soluzione è  $u(x) = \sin(x + \pi/4)$ . Esso ci è utile per osservare come la tecnica di *integrazione per parti* possa essere utilizzata per consentire l'introduzione di approssimanti meno regolari (*formulazione debole del metodo di Galerkin*).

Come funzione approssimante prendiamo un'espressione del tipo

$$u_M(x) = u_0(x) + \sum_{i=1}^M c_i \varphi_i(x) \in C^2[0, \pi]$$

con  $u_0(0) = \sqrt{2}/2$ ,  $u_0(\pi) = -\sqrt{2}/2$  e  $\varphi_i(0) = \varphi_i(\pi) = 0$ <sup>(†)</sup>. Applicando il criterio di Galerkin otteniamo

$$\int_0^\pi [u''_M(x) - u'_M(x) + u_M(x) + \cos(x + \pi/4)] \varphi_j(x) dx = 0, \quad j = 1, \dots, M$$

Ricorrendo all'integrazione per parti, il termine

$$\int_0^\pi u''_M(x) \varphi_j(x) dx$$

può essere sostituito, poiché  $u'_M(x) \varphi_j(x) = 0$  per  $x = 0, \pi$ , da

$$-\int_0^\pi u'_M(x) \varphi'_j(x) dx$$

La nuova formulazione

$$(9.47) \quad \int_0^\pi \{-u'_M(x) \varphi'_j(x) - [u'_M(x) - u_M(x) - \cos(x + \pi/4)] \varphi_j(x)\} dx = 0 \\ j = 1, \dots, M$$

non richiede più la presenza di  $u''_M(x)$ . Se lo desideriamo, la (9.47) ci consente di scegliere una  $u_M(x)$  meno regolare; per esempio una funzione polinomiale a tratti (vedi paragrafo 9.5); è infatti sufficiente che  $u'_M \in L^2(0, \pi)$ <sup>(††)</sup>. ◀

Ulteriori esempi dell'applicazione del metodo di Galerkin verranno presentati nel prossimo paragrafo.

Finora abbiamo considerato problemi ellittici e espresso la funzione approssimante  $u_M$  come combinazione lineare, con coefficienti  $\{c_i\}$  costanti, di funzioni base dipendenti da tutte le variabili (indipendenti); i coefficienti  $\{c_i\}$  vengono poi determinati come soluzione di un sistema di equazioni lineari. In qualche problema può invece risultare conveniente procedere diversamente. Nel caso, per esempio, di un problema dipendente da tre variabili  $x, y, z$  potremmo scegliere le funzioni base dipendenti dalle sole  $x$  e  $y$ , e

(†) Per esempio  $u_0(x) = \sqrt{2}/2 \cos(x)$  e  $\varphi_i(x) = \sin(ix)$ .

(††) Con  $L^2(D)$  definiamo lo spazio delle funzioni *v a quadrato sommabile*, ovvero tali che  $\langle v, v \rangle$  sia un numero finito.

i coefficienti della combinazione lineare non più costanti, ma funzione della variabile  $z$ , cioè

$$u_M(x, y, z) = u_0(x, y, z) + \sum_{i=1}^M c_i(z) \varphi_i(x, y)$$

dove  $u_0(x, y, z)$  soddisfa la condizione di Dirichlet sul bordo  $\Gamma$  del dominio delle variabili  $x, y$  qualunque sia il valore di  $z$ , mentre le  $\varphi_i(x, y)$  sono nulle su  $\Gamma$ . Una rappresentazione di questo tipo può risultare conveniente solo quando il dominio delle  $(x, y)$  è costante rispetto alla terza variabile  $z$ . Per esempio ciò si verifica generalmente nei problemi dipendenti dal tempo ( $t \equiv z$ ). Ovviamente le derivate di  $u_M$  rispetto a  $z$  resteranno presenti nel sistema finale, che sarà pertanto formato da equazioni differenziali ordinarie nelle incognite  $\{c_i(z)\}$ . Questo approccio verrà illustrato nella parte finale del paragrafo che segue.

## 9.5 Formulazione debole e elementi finiti

La scelta di approssimanti di tipo polinomiale, algebrico o trigonometrico, risulta efficace solo quando il problema è definito in una regione regolare, per esempio un rettangolo, e la soluzione stessa è molto regolare e non ha variazioni repentine di gradiente. Quando invece il dominio di definizione delle variabili spaziali ha una geometria non regolare, e conseguentemente anche la corrispondente soluzione è poco regolare, oppure, pur essendo la predetta geometria regolare i dati del problema rendono la soluzione poco regolare o addirittura non classica, allora il metodo numerico da utilizzare è quello di Galerkin, associato ad una formulazione debole del problema e alla scelta di un'approssimante di tipo polinomiale a tratti definita da una partizione del dominio spaziale.

Pertanto in questo paragrafo, limitandoci ad equazioni differenziali di ordine non superiore a due, riformuleremo dapprima i problemi considerati in forma debole e poi applicheremo a quest'ultima il criterio di Galerkin, accoppiato alla scelta di spazi di funzioni approssimanti  $X_M$  ( $V_M$ ) di tipo polinomiale a tratti e di basi  $\{\varphi_i\}$  opportune, ottenendo in questo modo la formulazione di un *metodo agli elementi finiti*.

A tale fine osserviamo preliminarmente che se la matrice dei coefficienti in (9.43) è densa, la determinazione della soluzione del sistema può divenire problematica quando  $M$  assume valori molto grandi. Ciò è proprio quanto avviene quando un metodo numerico viene applicato a problemi con geometrie complesse e/o con soluzioni poco regolari. Tuttavia, se lo spazio  $X_M$  ( $V_M$ ) scelto ammette delle funzioni di base  $\varphi_i$  con *supporto locale*<sup>(†)</sup>, ossia nulle quasi ovunque, tranne in piccoli sottoinsiemi del dominio  $D$ , i prodotti scalari  $\langle L \varphi_i, \varphi_j \rangle$  risultano non nulli solo quando i supporti di  $\varphi_i$  e  $\varphi_j$  hanno intersezione di “misura” non nulla. Pertanto, scegliendo oculatamente lo spazio  $X_M$  ( $V_M$ ) e le funzioni  $\{\varphi_i\}$  possiamo ottenere sistemi (9.43) di tipo sparso, con conseguenti risparmi di occupazione di memoria e di operazioni aritmetiche, e quindi con la possibilità di trattare sistemi di ordine  $M$  molto grande.

---

(†) Il supporto di una funzione  $\varphi(x)$  definita in  $D$  è la chiusura dell'insieme  $\{x \in D, f(x) \neq 0\}$ .

Per meglio illustrare ciò che intendiamo per funzione base a supporto locale, consideriamo dapprima il caso monodimensionale  $D \equiv (a, b)$  e suddividiamo tale dominio in  $M$  parti  $a \equiv x_0 < x_1 < \dots < x_M \equiv b$ . Come spazio  $X_M$  prendiamo l'insieme di tutte le poligonali associate alla suddetta partizione

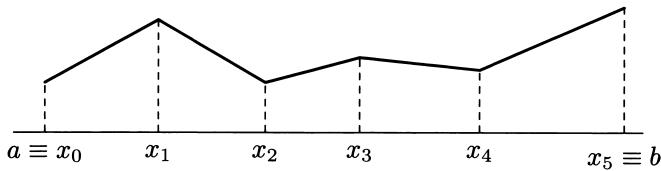


Figura 9.19

Tali funzioni sono continue in  $[a, b]$ ; le loro derivate prime risultano però discontinue nei nodi  $x_i$ ,  $i = 1, \dots, M - 1$ . Una base a supporto locale di  $X_M$  è data dalle  $M + 1$  funzioni  $\{N_i(x), i = 0, \dots, M\}$  definite in figura 9.20.

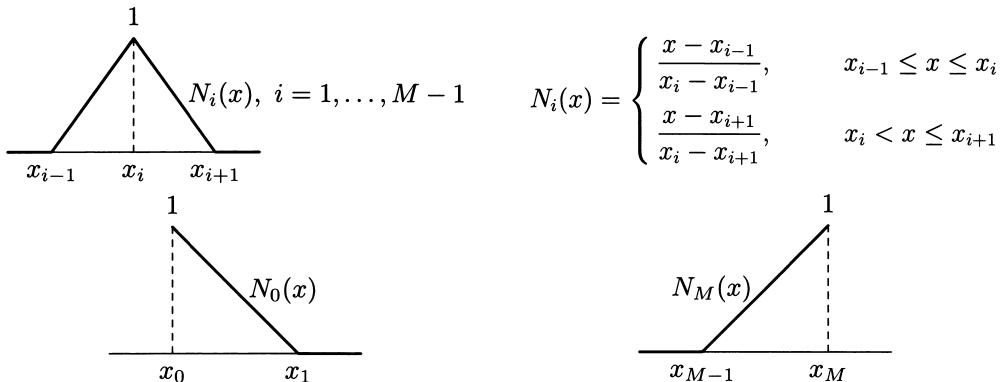
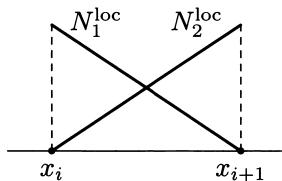


Figura 9.20

Per ottenere tale base è sufficiente considerare il generico *elemento*  $[x_i, x_{i+1}]$ , costruire i due polinomi fondamentali di Lagrange (vedi pag. 128)<sup>(†)</sup> associati ai due nodi  $x_i, x_{i+1}$



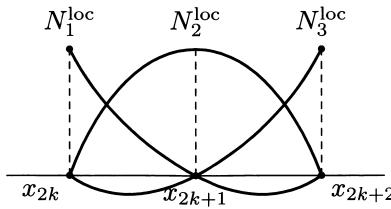
<sup>(†)</sup> Che costituiranno la cosiddetta *base lagrangiana locale*.

e unire segmenti attigui in modo da avere in  $[a, b]$  una funzione continua con supporto minimo. La generica  $N_i(x)$ ,  $i = 1, \dots, M - 1$ , che assume il valore 1 in  $x_i$ , viene ottenuta raccordando i due polinomi locali, uno in  $[x_{i-1}, x_i]$  l'altro in  $[x_i, x_{i+1}]$ , che assumono il valore 1 nel nodo  $x_i$ ; essa risulta nulla per  $x \leq x_{i-1}$  e  $x \geq x_{i+1}$ . Avremo pertanto

$$(9.48) \quad \int_{x_0}^{x_M} N_i(x)N_j(x) dx \neq 0$$

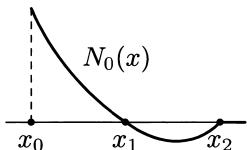
solo quando  $|i - j| = 0, 1$ .

Nel caso precedente, dopo aver suddiviso l'intervallo  $[a, b]$  in  $M$  parti (elementi)  $[x_j, x_{j+1}]$ ,  $j = 0, \dots, M - 1$ , potremmo scegliere un nodo intermedio  $x_j < x_{j+1/2} < x_{j+1}$  in ciascun elemento e prendere come  $X_M$  lo spazio delle funzioni polinomiali a tratti di grado (locale) 2 associate alla partizione iniziale. In questo caso, per costruire una base di  $X_M$  a supporto locale conviene rinumerare i  $2M + 1$  nodi:  $a = x_0 < x_1 < \dots < x_{2M-1} < x_{2M} = b$ , considerare le  $M$  terne di nodi consecutivi  $\{x_{2k}, x_{2k+1}, x_{2k+2}\}$ ,  $k = 0, \dots, M - 1$ , associare

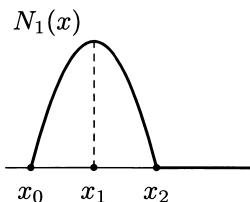


**Figura 9.21**

a ciascuna di esse i corrispondenti tre polinomi fondamentali di Lagrange (vedi figura 9.21) e definire la base  $\{N_i(x), i = 0, \dots, 2M\}$  di  $X_M$  come segue:

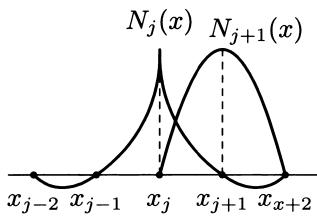


$$N_0(x) = \begin{cases} \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} & \text{in } [x_0, x_2] \\ 0 & \text{per } x \geq x_2 \end{cases}$$



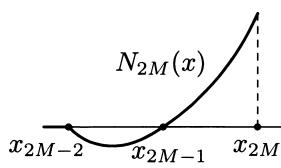
$$N_1(x) = \begin{cases} \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} & \text{in } [x_0, x_2] \\ 0 & \text{per } x \geq x_2 \end{cases}$$

$j = 2, 4, 6, \dots, 2M - 2 :$



$$N_j(x) = \begin{cases} \frac{(x - x_{j-2})(x - x_{j-1})}{(x_j - x_{j-2})(x_j - x_{j-1})} & \text{in } [x_{j-2}, x_j] \\ \frac{(x - x_{j+1})(x - x_{j+2})}{(x_{j+1} - x_j)(x_{j+1} - x_{j+2})} & \text{in } [x_j, x_{j+2}] \\ 0 & \text{per } x \leq x_{j-2} \text{ e } x \geq x_{j+2} \end{cases}$$

$$N_{j+1}(x) = \begin{cases} \frac{(x - x_j)(x - x_{j+2})}{(x_{j+1} - x_j)(x_{j+1} - x_{j+2})} & \text{in } [x_j, x_{j+2}] \\ 0 & \text{per } x \leq x_j \text{ e } x \geq x_{j+2} \end{cases}$$



$$N_{2M}(x) = \begin{cases} \frac{(x - x_{2M-2})(x - x_{2M-1})}{(x_{2M} - x_{2M-2})(x_{2M} - x_{2M-1})} & \text{in } [x_{2M-2}, x_{2M}] \\ 0 & \text{per } x \leq x_{2M-2} \end{cases}$$

La funzione  $N_j(x)$ ,  $j = 2, 4, 6, \dots, 2M - 2$ , è diversa da zero in due soli elementi della partizione iniziale dell'intervallo  $[a, b]$ ; le rimanenti  $N_j(x)$  sono invece non nulle in un solo elemento ciascuna.

► **Esempio.** Quando  $M = 3$  otteniamo quanto riportato in figura 9.22.

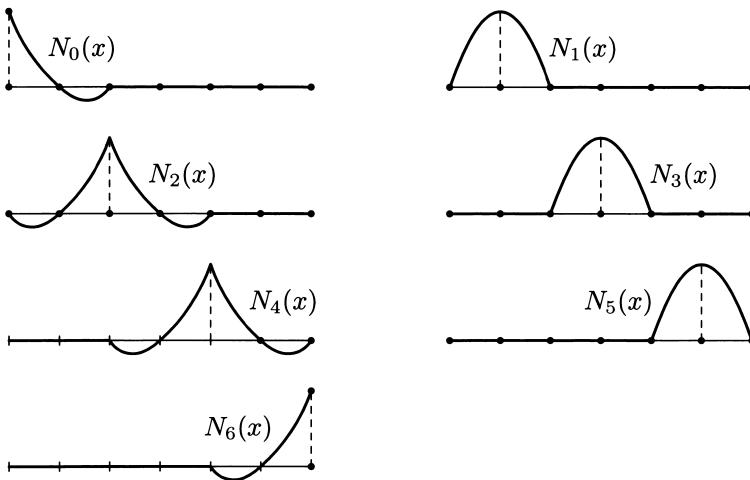


Figura 9.22

Procedendo in modo simile è possibile considerare spazi (lineari) di funzioni polinomiali a tratti di grado locale  $d \geq 3$ , associando a ciascuno di essi una base lagrangiana a supporto locale.

Nel caso di un dominio bidimensionale  $D \subset \mathbb{R}^2$  possiamo introdurre in  $D$  una decomposizione  $D_h$  fatta di rettangoli o triangoli, tale che due rettangoli o triangoli contigui abbiano in comune un vertice oppure tutto un lato (figura 9.23) e scegliere come  $X_M$  uno spazio di *funzioni localmente polinomiali* definite su  $D_h$ .

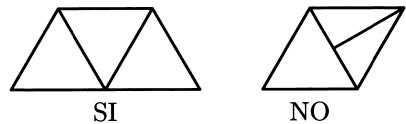


Figura 9.23

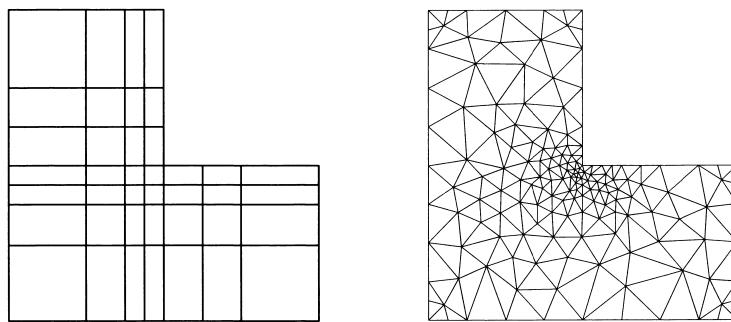


Figura 9.24

Per semplificare la costruzione di una base di  $X_M$  a supporto locale conviene far riferimento agli elementi fondamentali

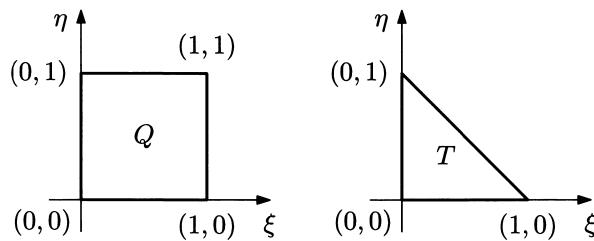


Figura 9.25

Esiste sempre una trasformazione affine invertibile, cioè del tipo

$$(9.49) \quad \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} \xi \\ \eta \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

con matrice non singolare, tra il quadrato unitario  $Q$  e un generico rettangolo, o tra  $T$  e un qualsiasi triangolo.

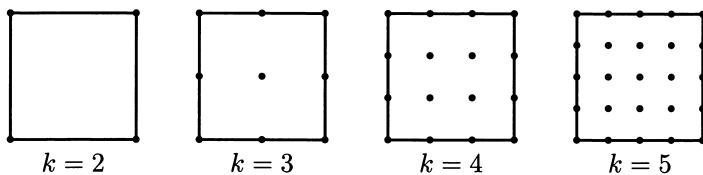
Esaminiamo dapprima il caso rettangolare. Scelti  $k \geq 2$  nodi distinti  $0 = \xi_1 < \xi_2 < \dots < \xi_k = 1$  sull'asse  $\xi$ , e  $k$  nodi distinti  $0 = \eta_1 < \eta_2 < \dots < \eta_k = 1$  sull'asse  $\eta$ , costruiamo la corrispondente *base lagrangiana locale* (vedi paragrafo 5.8) relativa all'elemento fondamentale  $Q$ :

$$(9.50) \quad \bar{N}_{ij}^{\text{loc}}(\xi, \eta) = l_i(\xi) \bar{l}_j(\eta) \quad \begin{array}{l} i = 1, \dots, k \\ j = 1, \dots, k \end{array}$$

che assume i valori

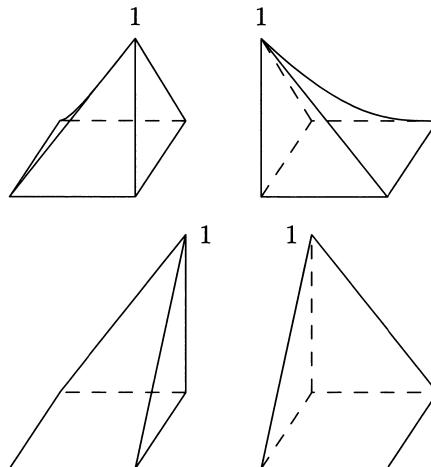
$$\bar{N}_{ij}^{\text{loc}}(\xi_m, \eta_n) = \begin{cases} 1 & \text{per } i = m \text{ e } j = n \\ 0 & \text{altrimenti} \end{cases}$$

Distribuzione nodi base lagrangiana:



**Figura 9.26**

Per esempio, quando  $k = 2$  otteniamo quanto riportato in figura 9.27.



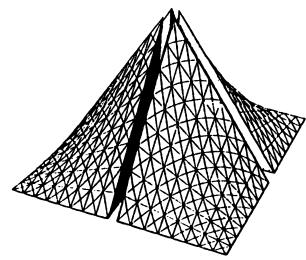
**Figura 9.27**

La generica funzione localmente polinomiale  $u_M(x, y)$ , associata alla decomposizione  $D_h$  e all'intero  $k$  scelto, può essere rappresentata nel quadrato unitario  $Q$  del piano  $(\xi, \eta)$  con la combinazione lineare

$$u_M(x(\xi, \eta), y(\xi, \eta)) = \bar{u}_M(\xi, \eta) = \sum_{i=1}^k \sum_{j=1}^k u_{ij} \bar{N}_{ij}^{\text{loc}}(\xi, \eta) = \sum_{i=1}^{k^2} u_i \bar{N}_i^{\text{loc}}(\xi, \eta) (\dagger)$$

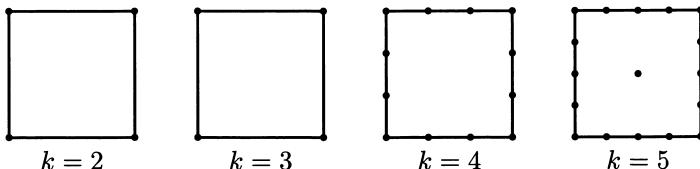
Da quest'ultima espressione potremo sempre dedurre, tramite la trasformazione affine inversa della (9.49), la rappresentazione di  $u_M(x, y)$ . Ad ogni funzione  $\bar{N}_i^{\text{loc}}(\xi, \eta)$  corrisponderà una  $N_i^{\text{loc}}(x, y)$  diversa per ogni elemento di  $D_h$ . La generica *funzione di base globale*  $N_n(x, y)$ , che assume il valore 1 nel nodo  $P_n \equiv (x_r, y_s)$  è ottenuta raccordando i lembi delle  $N_i^{\text{loc}}(x, y)$  relative ai rettangoli che hanno il nodo  $P_n$  in comune e che assumono in tale nodo il valore 1. Il supporto di  $N_n(x, y)$  sarà costituito da quattro rettangoli se  $P_n$  è un vertice, da due rettangoli se  $P_n$  giace su un lato ma non è vertice, da un solo rettangolo qualora  $P_n$  sia un nodo interno (figura 9.28).

Ricordiamo che nel sistema di equazioni che otteniamo applicando il metodo di Galerkin, vedi l'esempio monodimensionale di pagina 371, ad ogni nodo  $P_n \equiv (x_r, y_s)$  corrisponderà l'incognita  $u_n = u_M(x_r, y_s)$ . Pertanto, per cercare di ridurre il numero complessivo di tali incognite (e quindi l'ordine del sistema) conviene collocare i nodi  $(x_r, y_s)$  sui vertici e sui lati dei rettangoli. Al fine di ridurre, e se possibile eliminare, i nodi interni<sup>(††)</sup> presenti in ciascun elemento di  $D_h$  (necessari per poter costruire una base locale lagrangiana), sono state proposte particolari funzioni localmente polinomiali generate da una famiglia di nuove basi locali (non lagrangiane) nota con il nome di *serendipity* (vedi [9.7]).



**Figura 9.28**

Distribuzione nodi base serendipity:



**Figura 9.29**

Consideriamo ora gli elementi di forma triangolare, che si dimostrano più flessibili di quelli rettangolari e possono quindi risultare più convenienti, soprattutto quando il

(†) Nell'ultima espressione supponiamo di aver rinumerato localmente i nodi utilizzando un solo indice.  
 (††) Ma non solo per questo motivo; vedi ad esempio [9.7].

dominio da decomporre ha una geometria irregolare.

Per semplificare la costruzione delle funzioni di base locali conviene introdurre il sistema di coordinate (locali) baricentriche ([9.7], [9.28]) rispetto ai tre vertici, numerati in senso antiorario, dell'elemento considerato. Se con  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$  denotiamo le coordinate cartesiane dei tre vertici di un triangolo  $T_{123}$ , un punto  $P \equiv (x, y)$  di quest'ultimo può essere individuato dalla terna  $(L_1, L_2, L_3)$  definita dalle relazioni

$$\begin{aligned} x &= L_1 x_1 + L_2 x_2 + L_3 x_3 \\ y &= L_1 y_1 + L_2 y_2 + L_3 y_3 \\ 1 &= L_1 + L_2 + L_3 \end{aligned}$$

È inoltre noto che, posto

$$A = \frac{1}{2} \det \begin{pmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{pmatrix}$$

le coordinate  $L_1$ ,  $L_2$  e  $L_3$  sono date dalle espressioni

$$(9.51) \quad \begin{aligned} L_1 &= \frac{\text{area}(T_{23})}{\text{area}(T_{123})} = \frac{a_1 + b_1 x + c_1 y}{2A} \\ L_2 &= \frac{\text{area}(T_{13})}{\text{area}(T_{123})} = \frac{a_2 + b_2 x + c_2 y}{2A} \\ L_3 &= \frac{\text{area}(T_{12})}{\text{area}(T_{123})} = \frac{a_3 + b_3 x + c_3 y}{2A} \end{aligned}$$

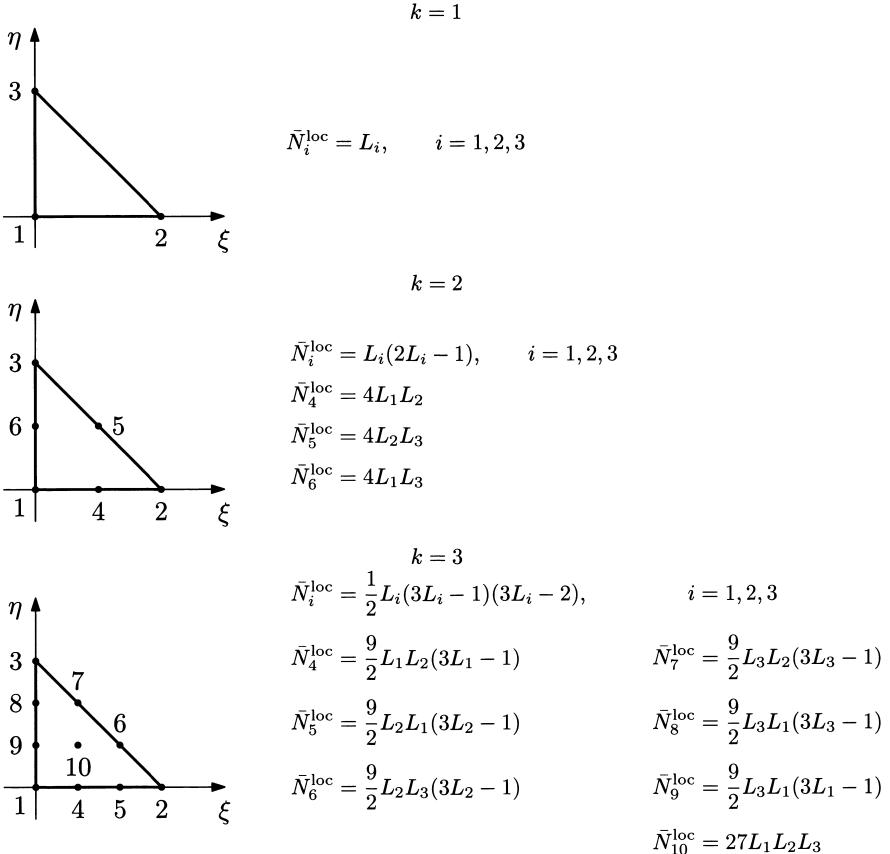
ove

$$\begin{aligned} a_1 &= x_2 y_3 - x_3 y_2 & a_2 &= x_3 y_1 - x_1 y_3 & a_3 &= x_1 y_2 - x_2 y_1 \\ b_1 &= y_2 - y_3 & b_2 &= y_3 - y_1 & b_3 &= y_1 - y_2 \\ c_1 &= x_3 - x_2 & c_2 &= x_1 - x_3 & c_3 &= x_2 - x_1 \end{aligned}$$

Nel caso del triangolo fondamentale  $T$  di figura 9.25 le precedenti rappresentazioni delle coordinate baricentriche assumono una forma assai semplice:

$$(9.52) \quad \begin{aligned} L_1 &= \xi \\ L_2 &= \eta \\ L_3 &= 1 - \xi - \eta \end{aligned}$$

Nella tabella 9.1 che segue:



**Tabella 9.1**

riportiamo una base locale  $\{\bar{N}_i^{\text{loc}}\}$  per i polinomi di grado  $k = 1, 2, 3$ , definiti nel triangolo fondamentale  $T(\dagger)$ .

La generica funzione di base globale  $N_n(x, y)$ , che assume il valore 1 nel nodo  $P_n$ , viene ottenuta raccordando i lembi delle  $N_i^{\text{loc}}(x, y)$  che assumono in  $P_n$  il valore 1. Nel caso  $k = 1$ , per esempio, abbiamo quanto riportato in figura 9.30.

---

( $\dagger$ ) Le funzioni  $\{\bar{N}_i^{\text{loc}}\}$  riportate costituiscono una base locale anche quando il triangolo è generico; in tal caso però le coordinate  $\{L_i\}$  sono definite dalle (9.51) e non dalle (9.52).

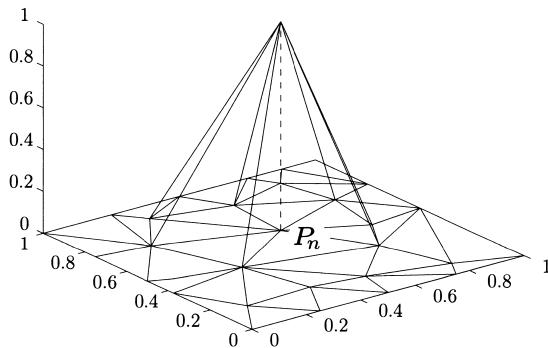
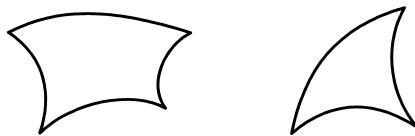


Figura 9.30

Quando il contorno  $\Gamma$  del dominio  $D$  è curvilineo, può convenire introdurre nella discretizzazione di  $D$  elementi quadrangolari o triangolari con uno o più lati curvilinei; più in generale di tipo

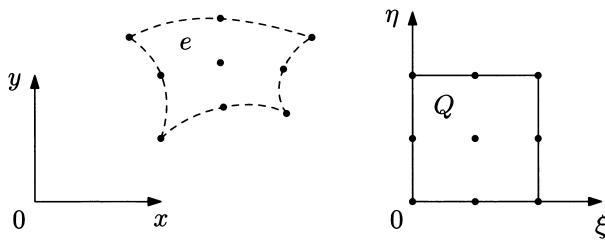


denominati *isoparametrici*<sup>(†)</sup>.

Questi elementi vengono costruiti ricorrendo alle stesse formule di interpolazione utilizzate per approssimare localmente (con riferimento all'elemento fondamentale) la soluzione  $u(x, y)$ . In particolare, se nell'elemento fondamentale l'approssimante  $\bar{u}_M$  è definita dall'espressione

$$\bar{u}_M(\xi, \eta) = \sum_{i=1}^K u_i \bar{N}_i^{\text{loc}}(\xi, \eta)$$

scelti  $K$  punti di coordinate  $(x_i, y_i)$ , che caratterizzano l'elemento curvilineo in costruzione, per esempio



(†) Vedere tuttavia [9.7], [9.23].

definiamo l'elemento  $e$  mediante la trasformazione

$$x = \sum_{i=1}^K x_i \bar{N}_i^{\text{loc}}(\xi, \eta)$$

$$y = \sum_{i=1}^K y_i \bar{N}_i^{\text{loc}}(\xi, \eta)$$

Ricordiamo infine che gli integrali presenti in (9.43) possono essere tutti approssimati con formule quali quelle presentate nel paragrafo 7.10 (vedi tuttavia [9.5]), riconducendoli, mediante la trasformazione (9.49), all'elemento fondamentale  $Q$  oppure a  $T$ .

L'aumento di precisione nell'approssimazione della soluzione  $u$  viene conseguito raffinando la decomposizione del dominio  $D$ , ovvero riducendo il *diametro*<sup>(†)</sup> massimo  $h$  degli elementi triangolari o rettangolari, e mantenendo costante il grado dei polinomi locali<sup>(††)</sup>.

Dopo aver descritto la costruzione di approssimanti di tipo polinomiale a tratti e delle relative basi a supporto locale, illustriamo ora con alcuni esempi il procedimento che ci consente di ottenere, partendo dalla formulazione classica del problema, la corrispondente formulazione debole. A quest'ultima applicheremo poi, scegliendo come approssimanti le predette funzioni, il criterio di Galerkin. Lo studio delle proprietà di convergenza delle approssimanti così ottenute non verrà effettuato, in quanto esso richiederebbe concetti e risultati matematici che sono propri di un corso avanzato sui metodi numerici per equazioni alle derivate parziali. Al lettore interessato ad approfondire tale questione suggeriamo i testi [9.33] e [9.34].

► **Esempio 9.3.** Consideriamo il seguente problema:

$$\begin{cases} -u''(x) + \sigma(x)u(x) = f(x), & a < x < b \\ u(a) = h_a \\ u(b) = h_b \end{cases}$$

dove  $[a, b]$  è un intervallo limitato,  $f, \sigma \in C[a, b]$  e  $\sigma(x) > 0$ ,  $a \leq x \leq b$ .

Le ipotesi fatte sui coefficienti  $\sigma$ ,  $f$  ci assicurano (vedi pagina 292) che esso ammette una ed una sola soluzione (classica)  $u(x) \in C^2[a, b]$ . Per ottenere la formulazione debole del problema assegnato consideriamo dapprima il seguente sottospazio lineare di  $L^2(a, b)$ :

$$H^1 \equiv H^1(a, b) = \{ v \in L^2(a, b) : v' \in L^2(a, b) \}$$

(†) Per diametro di un elemento intendiamo la massima distanza tra due punti qualsiasi dello stesso elemento.

(††) È tuttavia possibile, e spesso auspicabile, poter operare una scelta tra la riduzione del diametro di un elemento e l'aumento del grado del polinomio locale definito su tale elemento.

dove la derivata  $v'$  è definita nel senso delle distribuzioni, dotato della norma

$$\|v\| = \left( \int_a^b |v(x)|^2 dx + \int_a^b |v'(x)|^2 dx \right)^{\frac{1}{2}}$$

Esso è un particolare *spazio di Sobolev* (vedi ad esempio [9.33], [9.34]). Si ricorda che quando il dominio è un intervallo limitato, come nel nostro caso, risulta  $H^1 \subset C[a, b]$  e quindi, in particolare, gli elementi di tale spazio sono funzioni definite in ogni punto di  $[a, b]$  (<sup>†</sup>). Richiamando questa proprietà, successivamente definiamo il sottospazio (lineare) di  $H^1$

$$H_0^1 \equiv H_{0,\Gamma_D}^1(a, b) = \{v \in H^1 : v(a) = 0, v(b) = 0\}, \quad (\Gamma_D = \{a, b\})$$

Posto quindi  $V \equiv H_0^1$ , moltiplichiamo l'equazione differenziale per la generica funzione  $v \in V$ . Integrando su  $(a, b)$  otteniamo

$$-\int_a^b u''(x)v(x) dx + \int_a^b \sigma(x)u(x)v(x) dx = \int_a^b f(x)v(x) dx, \quad \forall v \in V$$

da cui segue, applicando la regola di integrazione per parti al primo addendo,

$$-[u'(b)v(b) - u'(a)v(a)] + \int_a^b u'(x)v'(x) dx + \int_a^b \sigma(x)u(x)v(x) dx = \int_a^b f(x)v(x) dx$$

Tenendo poi conto che  $u'$  è limitata e  $v(a) = v(b) = 0$  perveniamo infine alla seguente relazione:

$$(9.53) \quad \int_a^b u'(x)v'(x) dx + \int_a^b \sigma(x)u(x)v(x) dx = \int_a^b f(x)v(x) dx, \quad \forall v \in V$$

Poiché per la nota diseguaglianza di Cauchy-Schwarz il primo integrale in (9.53) esiste anche quando  $u(x)$  è un elemento di  $H^1$ , definiamo la seguente formulazione debole del problema iniziale: trovare

$$u \in W = u_0 + V = \{u = u_0 + \bar{u} : \bar{u} \in V\}$$

con  $u_0 \in H^1$  scelta in modo che

$$u_0(a) = h_a, \quad u_0(b) = h_b$$

per esempio  $u_0(x) = [h_a(b-x) + h_b(x-a)]/(b-a)$ , ovvero trovare  $\bar{u} \in V$  ( $u = u_0 + \bar{u}$ ) tale che

$$\begin{aligned} & \int_a^b \bar{u}'(x)v'(x) dx + \int_a^b \sigma(x)\bar{u}(x)v(x) dx = \\ & \int_a^b [f(x) - \sigma(x)u_0(x)]v(x) dx - \int_a^b u'_0(x)v'(x) dx, \quad \forall v \in V \end{aligned}$$

---

(<sup>†</sup>) In realtà, poiché  $H^1$  è un sottospazio di  $L^2$ , ad ogni elemento continuo di  $H^1$  è associata tutta una classe di equivalenza di funzioni (di  $H^1$ ) che differiscono da tale elemento solo in un insieme di punti di misura nulla.

È possibile dimostrare (si veda [9.33]) che tale formulazione ammette una ed una sola soluzione, che nelle ipotesi di regolarità fatte per i dati del problema iniziale coincide necessariamente con la soluzione (classica) di quest'ultimo. Se invece il coefficiente  $\sigma(x)$  pur essendo una funzione limitata presentasse, per esempio, delle discontinuità, e  $f(x)$  fosse un generico elemento di  $L^2(a, b)$ , la soluzione del problema non risulterebbe classica e l'equazione differenziale dovrebbe essere interpretata nel senso distribuzionale. È tuttavia possibile dimostrare che anche in questo caso la predetta formulazione debole ha, nello spazio  $W$ , come unica soluzione quella (debole) del problema scritto in forma differenziale. Proprietà analoghe risultano valide anche per le formulazioni deboli dei problemi che verranno trattati negli esempi che seguono.

Come approssimante sceglieremo una funzione polinomiale a tratti  $u_M(x)$  di grado locale 1, associata alla partizione  $I_M : a \equiv x_0 < x_1 < \dots < x_{M-1} < x_M \equiv b$  del dominio di definizione del problema, che soddisfi a priori le condizioni (di Dirichlet) imposte agli estremi  $a, b$ . Definito quindi lo spazio

$$W_M = u_0 + V_M \subset W, \quad u_0(x) = h_a N_0(x) + h_b N_M(x)$$

dove  $V_M$  è lo spazio lineare di dimensione  $M - 1$ :

$$V_M = \{v_M \text{ è una poligonale definita da } I_M : v_M(a) = 0, v_M(b) = 0\} \subset V$$

e le funzioni  $\{N_i(x)\}$  sono definite a pagina 356, il metodo di Galerkin consiste nel determinare l'elemento  $u_M \in V_M$  che soddisfa la formulazione debole (9.53) in  $V_M$  anziché in  $V$ :

$$\int_a^b u'_M(x)v'_M(x) \, dx + \int_a^b \sigma(x)u_M(x)v_M(x) \, dx = \int_a^b f(x)v_M(x) \, dx, \quad \forall v_M \in V_M$$

Tenendo conto che il sottospazio  $V_M$  è lineare ed è generato dalle funzioni di base  $N_1(x), N_2(x), \dots, N_{M-1}(x)$ , la relazione che definisce il metodo di Galerkin risulta equivalente al sistema, di dimensione finita,

$$\int_a^b u'_M(x)N'_i(x) \, dx + \int_a^b \sigma(x)u_M(x)N_i(x) \, dx = \int_a^b f(x)N_i(x) \, dx, \quad i = 1, \dots, M-1$$

Quest'ultimo coincide con la forma debole del metodo di Galerkin introdotta a pagina 354. È proprio questa interpretazione che consente l'esame delle proprietà di convergenza di quel metodo.

Osservando infine che il generico elemento di  $W_M$  può essere rappresentato nella forma:

$$u_M(x) = h_a N_0(x) + h_b N_M(x) + \sum_{j=1}^{M-1} c_j N_j(x)$$

dove i coefficienti  $c_j$  assumono il seguente significato:  $c_j \equiv u_M(x_j)$ , sostituiamo l'espressione di  $u_M$  nelle equazioni che definiscono il metodo di Galerkin; otteniamo il seguente

sistema di  $M - 1$  equazioni lineari nelle incognite  $c_j$ ,  $j = 1, \dots, M - 1$ :

$$\begin{aligned} & \sum_{j=1}^{M-1} \left[ \int_a^b N'_j(x) N'_i(x) \, dx + \int_a^b \sigma(x) N_j(x) N_i(x) \, dx \right] c_j \\ &= \int_a^b f(x) N_i(x) \, dx - h_a \int_a^b N'_0(x) N'_i(x) \, dx - h_b \int_a^b N'_M(x) N'_i(x) \, dx \\ & - h_a \int_a^b \sigma(x) N_0(x) N_i(x) \, dx - h_b \int_a^b \sigma(x) N_M(x) N_i(x) \, dx, \quad i = 1, \dots, M - 1 \end{aligned}$$

Ponendo

$$b_{ij} = \int_a^b N'_j(x) N'_i(x) \, dx, \quad a_{ij}^\sigma = \int_a^b \sigma(x) N_j(x) N_i(x) \, dx$$

e osservando che  $b_{ij} = b_{ji}$  e  $a_{ij}^\sigma = a_{ji}^\sigma$  sono diversi da zero solo se  $j = i - 1, i, i + 1$ , abbiamo il sistema lineare tridiagonale di ordine  $M - 1$ :

$$\left\{ \begin{array}{l} (b_{11} + a_{11}^\sigma)c_1 + (b_{12} + a_{12}^\sigma)c_2 = \int_a^b f(x) N_1(x) \, dx - h_a b_{10} - h_a a_{10}^\sigma \\ \vdots \\ (b_{ii-1} + a_{ii-1}^\sigma)c_{i-1} + (b_{ii} + a_{ii}^\sigma)c_i + (b_{ii+1} + a_{ii+1}^\sigma)c_{i+1} = \int_a^b f(x) N_i(x) \, dx \\ \vdots \\ (b_{M-1M-2} + a_{M-1M-2}^\sigma)c_{M-2} + (b_{M-1M-1} + a_{M-1M-1}^\sigma)c_{M-1} \\ = \int_a^b f(x) N_{M-1}(x) \, dx - h_b b_{M-1M} - h_b a_{M-1M}^\sigma \end{array} \right.$$

la cui matrice (detta *di rigidezza* o *di stiffness*) è simmetrica definita positiva (vedi [9.33]) e quindi, in particolare, non singolare.

*Osservazione.* Sia in questo caso che in quelli che seguiranno, per semplificare la descrizione delle equazioni che vengono ottenute applicando il criterio di Galerkin e scegliendo come approssimanti delle funzioni polinomiali a tratti, in particolare delle poligonali, tutti gli integrali sono definiti con riferimento all'intero dominio spaziale. Si richiama tuttavia l'attenzione sulle caratteristiche che hanno le funzioni di base scelte per rappresentare tali approssimanti. Poiché il loro supporto è locale, la definizione degli integrali è limitata ai sottodomini nei quali le funzioni integrande non sono identicamente nulle.



► **Esempio 9.4.** Consideriamo nuovamente l'equazione dell'esempio precedente, soggetta alle stesse ipotesi, associandole però una condizione di tipo misto nell'estremo  $b$ :

$$\begin{cases} -u''(x) + \sigma(x)u(x) = f(x), & a < x < b \\ u(a) = h_a \\ u'(b) + \gamma u(b) = h_b, \quad \gamma \geq 0 \end{cases}$$

Anche in questo caso è possibile dimostrare che il problema assegnato ammette una ed una sola soluzione  $u(x) \in C^2[a, b]$ . Per ottenere la formulazione debole del problema assegnato moltiplichiamo l'equazione differenziale per la generica funzione

$$v \in V \equiv H_{0,\Gamma_D}^1 = \{v \in H^1 : v(a) = 0\}, \quad (\Gamma_D = \{a\})$$

Si osservi come lo spazio delle *funzioni test*  $V \subset H^1$  sia costituito da funzioni che si annullano solo nell'estremo dell'intervallo ( $x = a$ ) in cui è assegnata la condizione di Dirichlet. Se le condizioni ai due estremi  $a, b$  fossero di tipo Neumann o misto, alle funzioni test non imporremmo alcuna condizione in  $a, b$ , ovvero prenderemmo  $V \equiv H^1$ .

Integriamo ora su  $(a, b)$

$$-\int_a^b u''(x)v(x) \, dx + \int_a^b \sigma(x)u(x)v(x) \, dx = \int_a^b f(x)v(x) \, dx, \quad \forall v \in V$$

e quindi applichiamo la regola di integrazione per parti al primo addendo; otteniamo:

$$-[u'(b)v(b) - u'(a)v(a)] + \int_a^b u'(x)v'(x) \, dx + \int_a^b \sigma(x)u(x)v(x) \, dx = \int_a^b f(x)v(x) \, dx$$

Tenendo conto che  $u'(x)$  è limitata,  $v(a) = 0$  e  $u'(b) = h_b - \gamma u(b)$ , perveniamo alla seguente relazione:

$$\int_a^b u'(x)v'(x) \, dx + \int_a^b \sigma(x)u(x)v(x) \, dx + \gamma u(b)v(b) = \int_a^b f(x)v(x) \, dx + h_b v(b), \quad \forall v \in V$$

La formulazione debole diventa quindi: trovare

$$u \in W = u_0 + V$$

con  $u_0 \in H^1$  scelta in modo che

$$u_0(a) = h_a$$

ovvero trovare  $\bar{u} \in V$  che soddisfi la relazione

$$\begin{aligned} & \int_a^b \bar{u}'(x)v'(x) \, dx + \int_a^b \sigma(x)\bar{u}(x)v(x) \, dx + \gamma u(b)v(b) = \\ & \int_a^b f(x)v(x) \, dx - \int_a^b u'_0(x)v'(x) \, dx - \int_a^b \sigma(x)u_0(x)v(x) \, dx + h_b v(b), \quad \forall v \in V \end{aligned}$$

Anche in questo caso è possibile dimostrare che tale formulazione definisce univocamente la soluzione  $\bar{u} \in V$  e quindi  $u \in W$ . Questo significa che la condizione di tipo misto nell'estremo  $b$ , che abbiamo inserito direttamente nella formulazione (debole) finale, è automaticamente soddisfatta dalla soluzione di quest'ultimo, anche se, a differenza di quella (di Dirichlet) in  $a$ , tale condizione non è stata esplicitamente imposta alle funzioni dello spazio  $W$  scelto. Per questo motivo le condizioni di tipo Neumann o misto vengono definite *naturali*, mentre quelle di Dirichlet, che vengono imposte esplicitamente a tutti gli elementi dello spazio  $W$ , sono denominate *essenziali*.

Per semplicità, come approssimante sceglieremo nuovamente una funzione polinomiale a tratti  $u_M(x)$  di grado locale 1, associata alla partizione  $I_M : a \equiv x_0 < x_1 < \dots < x_{M-1} < x_M \equiv b$  dell'intervallo  $[a, b]$ , che soddisfi a priori la condizione (di Dirichlet) del problema imposta nell'estremo  $x = a$ . Posto quindi

$$W_M = u_0 + V_M \subset W, \quad u_0(x) = h_a N_0(x)$$

dove ora  $V_M$  è lo spazio lineare, di dimensione  $M$ :

$$V_M = \{v_M \text{ è una poligonale definita da } I_M : v_M(a) = 0\} \subset V$$

il metodo di Galerkin consiste nella determinazione della poligonale  $u_M \in V_M$ , che risulterà unica, definita mediante la relazione

$$\begin{aligned} \int_a^b u'_M(x)v'_M(x) \, dx + \int_a^b \sigma(x)u_M(x)v_M(x) \, dx + \gamma u_M(b)v_M(b) \\ = \int_a^b f(x)v_M(x) \, dx + h_b v_M(b), \quad \forall v_M \in V_M \end{aligned}$$

Tenendo conto che il predetto sottospazio (lineare)  $V_M$  è generato dalle funzioni base  $N_1(x), N_2(x), \dots, N_M(x)$ , l'equazione che definisce il metodo di Galerkin risulta pertanto equivalente al sistema:

$$\begin{aligned} \int_a^b u'_M(x)N'_i(x) \, dx + \int_a^b \sigma(x)u_M(x)N_i(x) \, dx + \gamma u_M(b)N_i(b) = \\ \int_a^b f(x)N_i(x) \, dx + h_b N_i(b), \quad i = 1 \dots, M \end{aligned}$$

Osservando infine che il generico elemento di  $W_M$  può essere rappresentato nella forma seguente:

$$u_M(x) = h_a N_0(x) + \sum_{j=1}^M c_j N_j(x)$$

dove, come nel caso precedente,  $c_j \equiv u_M(x_j)$ , sostituiamo l'espressione di  $u_M$  nelle equazioni che definiscono il metodo di Galerkin ed otteniamo il seguente sistema di  $M$  equa-

zioni lineari nelle incognite  $c_j$ ,  $j = 1, \dots, M$ :

$$\begin{aligned} & \sum_{j=1}^M \left[ \int_a^b N'_j(x) N'_i(x) \, dx + \int_a^b \sigma(x) N_j(x) N_i(x) \, dx \right] c_j + \gamma \left[ h_a N_0(b) + \sum_{j=1}^M c_j N_j(b) \right] N_i(b) \\ &= \int_a^b f(x) N_i(x) \, dx - h_a \int_a^b N'_0(x) N'_i(x) \, dx - h_a \int_a^b \sigma(x) N_0(x) N_i(x) \, dx + h_b N_i(b) \\ & \quad i = 1, \dots, M \end{aligned}$$

Ponendo

$$\begin{aligned} b_{ij} &= \int_a^b N'_j(x) N'_i(x) \, dx, \\ a_{ij}^\sigma &= \int_a^b \sigma(x) N_j(x) N_i(x) \, dx \end{aligned}$$

e osservando che  $b_{ij} = b_{ji}$  e  $a_{ij}^\sigma = a_{ji}^\sigma$  sono diversi da zero solo se  $j = i-1, i, i+1$  e che  $N_i(b) = 0$  per  $i = 1, \dots, M-1$  e  $N_M(b) = 1$ , otteniamo il sistema lineare tridiagonale

$$\left\{ \begin{array}{l} (b_{11} + a_{11}^\sigma) c_1 + (b_{12} + a_{12}^\sigma) c_2 = \int_a^b f(x) N_1(x) \, dx - h_a b_{10} - h_a a_{10}^\sigma \\ \vdots \\ (b_{ii-1} + a_{ii-1}^\sigma) c_{i-1} + (b_{ii} + a_{ii}^\sigma) c_i + (b_{ii+1} + a_{ii+1}^\sigma) c_{i+1} = \int_a^b f(x) N_i(x) \, dx \\ \vdots \\ (b_{MM-1} + a_{MM-1}^\sigma) c_{M-1} + (b_{MM} + a_{MM}^\sigma + \gamma) c_M = \int_a^b f(x) N_M(x) \, dx + h_b \end{array} \right.$$

la cui matrice (di rigidezza) è simmetrica definita positiva.

A differenza della condizione (essenziale) imposta nell'estremo  $a$ , quella (naturale) relativa all'estremo  $b$  in generale non è soddisfatta dall'approssimante  $u_M$ ; lo sarà invece solo per  $M \rightarrow \infty$ .

► **Esempio 9.5.** Proseguiamo la nostra descrizione considerando ora un problema dipendente dal tempo: l'equazione del calore

$$(9.54) \quad \frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} = f, \quad 0 < x < 1, \quad t > 0$$

soggetta alle condizioni

$$\begin{aligned} (9.55) \quad u(x, 0) &= g(x), & 0 \leq x \leq 1 \\ u(0, t) &= \alpha(t), & t > 0 \\ u(1, t) &= \beta(t), & t > 0 \end{aligned}$$

e con dati  $f, g, \alpha, \beta$  che soddisfano le condizioni di regolarità che garantiscono l'esistenza e unicità di una soluzione classica con derivate limitate.

Per ottenere la corrispondente formulazione debole, come nel primo esempio consideriamo lo spazio  $W = u_0 + V$ ,  $V = H_0^1$ , con  $u_0 \in H^1$  tale che  $u_0(0, t) = \alpha(t)$ ,  $u_0(1, t) = \beta(t)$ , e, qualunque sia l'istante  $t > 0$ , moltiplichiamo l'equazione del calore per la generica funzione  $v \in V$  e integriamo in  $(0, 1)$ :

$$\int_0^1 u_t v \, dx - \int_0^1 u_{xx} v \, dx = \int_0^1 f v \, dx, \quad \forall v \in V$$

Successivamente integriamo per parti il secondo termine; otteniamo

$$(9.56) \quad \int_0^1 u_t v \, dx + \int_0^1 u_x v' \, dx = \int_0^1 f v \, dx, \quad \forall v \in V$$

Il nuovo problema assume quindi la formulazione seguente: per ogni istante  $t > 0$  trovare  $u \in W$  tale che la relazione (9.56) sia soddisfatta. Anche in questo caso è possibile dimostrare ([9.33], p.136) che tale soluzione esiste ed è unica e coincide con quella del problema differenziale.

Analogamente ai casi precedenti, suddividiamo l'intervallo  $[0, 1]$  in  $M$  parti:  $0 = x_0 < x_1 < \dots < x_{M-1} < x_M = 1$ . Come approssimante prendiamo

$$u_M(x, t) = \alpha(t)N_0(x) + \beta(t)N_M(x) + \sum_{j=1}^{M-1} c_j(t)N_j(x)$$

Osserviamo subito il significato particolare che i coefficienti  $c_j(t)$  assumono:

$$c_j(t) = u_M(x_j, t) \quad j = 1, \dots, M-1$$

Applicando il metodo di Galerkin alla (9.56) otteniamo l'espressione

$$(9.57) \quad \int_0^1 \frac{\partial u_M}{\partial t} N_i \, dx + \int_0^1 \frac{\partial u_M}{\partial x} N'_i \, dx = \int_0^1 f N_i \, dx, \quad i = 1, \dots, M-1$$

ovvero

$$\begin{aligned} \sum_{j=1}^{M-1} c'_j(t) \int_0^1 N_j(x) N_i(x) \, dx + \sum_{j=1}^{M-1} c_j(t) \int_0^1 N'_j(x) N'_i(x) \, dx &= \int_0^1 f(x, t) N_i(x) \, dx \\ - \alpha'(t) \int_0^1 N_0(x) N_i(x) \, dx - \beta'(t) \int_0^1 N_M(x) N_i(x) \, dx - \alpha(t) \int_0^1 N'_0(x) N'_i(x) \, dx \\ &\quad - \beta(t) \int_0^1 N'_M(x) N'_i(x) \, dx, \quad i = 1, \dots, M-1 \end{aligned}$$

Ricordando infine la forma delle funzioni  $N_i(x)$  e la (9.48), otteniamo

$$(9.58) \quad \sum_{j=i-1}^{i+1} a_{ij} c'_j(t) + \sum_{j=i-1}^{i+1} b_{ij} c_j(t) = d_i(t), \quad i = 1, \dots, M-1$$

dove

$$a_{ij} = \int_0^1 N_i(x)N_j(x) dx = \begin{cases} \int_{x_{i-1}}^{x_i} N_i(x)N_{i-1}(x) dx, & j = i-1 \\ \int_{x_{i-1}}^{x_{i+1}} N_i(x)N_i(x) dx, & j = i \\ \int_{x_i}^{x_{i+1}} N_i(x)N_{i+1}(x) dx, & j = i+1 \end{cases}$$

$$b_{ij} = \int_0^1 N'_i(x)N'_j(x) dx = \begin{cases} -\frac{1}{x_i - x_{i-1}} & j = i-1 \\ \frac{1}{x_i - x_{i-1}} + \frac{1}{x_{i+1} - x_i} & j = i \\ -\frac{1}{x_{i+1} - x_i} & j = i+1 \end{cases}$$

$$d_1(t) = \int_0^{x_2} f(x, t)N_1(x) dx - \alpha'(t)a_{01} - \alpha(t)b_{01}$$

$$d_i(t) = \int_{x_{i-1}}^{x_{i+1}} f(x, t)N_i(x) dx, \quad i = 2, \dots, M-2 \quad (M \geq 4)$$

$$d_{M-1}(t) = \int_{x_{M-2}}^1 f(x, t)N_{M-1}(x) dx - \beta'(t)a_{M-1} - \beta(t)b_{M-1}$$

Richiamando la condizione iniziale in (9.55), alle equazioni (9.58) associamo i valori iniziali

$$(9.59) \quad c_j(0) = g(x_j), \quad j = 1, \dots, M-1$$

Il sistema (9.58), (9.59), di tipo

$$\begin{cases} Ac'(t) = -Bc(t) + d(t) \\ c(0) = \bar{g} \end{cases}$$

con  $A$  e  $B$  matrici tridiagonali di ordine  $M-1$ <sup>(†)</sup>, entrambe simmetriche definite positive, risulta generalmente stiff e quindi potrebbe essere risolto, per esempio, utilizzando il metodo dei trapezi. ◀

Finora sono stati affrontati problemi definiti in domini spaziali rappresentati da intervalli limitati. I concetti e i metodi illustrati negli esempi precedenti possono tuttavia essere facilmente estesi ad analoghi problemi definiti in domini limitati di  $\mathbb{R}^2$  e  $\mathbb{R}^3$  con contorni *Lipschitz-continui*. Nei due esempi che seguono, con procedimenti del tutto

---

(†) Dette *di massa* la prima, e *di rigidezza* la seconda.

simili a quelli finora seguiti dedurremo prima la formulazione debole e poi il corrispondente metodo di Galerkin agli elementi finiti per un problema stazionario e per uno di evoluzione, entrambi definiti in domini limitati di  $\mathbb{R}^2$ .

In questi due casi occorre considerare lo spazio lineare

$$H^1 = H^1(D) = \{u \in L^2(D) : \frac{\partial u}{\partial x}, \frac{\partial u}{\partial y} \in L^2(D)\}$$

dotato della norma

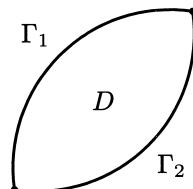
$$\|v\| = \left[ \sum_{k_1+k_2=0}^1 \int_D \left| \frac{\partial^{k_2}}{\partial y^{k_2}} \frac{\partial^{k_1}}{\partial x^{k_1}} v(x, y) \right|^2 dx dy \right]^{\frac{1}{2}}$$

Purtroppo, quando la dimensione del dominio  $D$  è superiore a 1, la generica funzione  $u \in H^1$  non è necessariamente definita nel generico punto di  $D \cup \Gamma$ , né tanto meno ivi continua. Ciò nonostante è possibile generalizzare la definizione di valore di  $u \in H^1$  su una curva (si veda il concetto di *traccia* di una funzione (vedi per esempio [9.33], [9.34])). Avvalendosi di questa generalizzazione, nella deduzione delle formulazioni deboli dei problemi che seguono potremo formalmente procedere come se il valore che una funzione di  $H^1$  assume sulla curva  $\Gamma$  fosse definito in senso classico (puntuale). Inoltre possiamo definire il seguente sottospazio (lineare) di  $H^1$ :

$$H_{0,\Gamma}^1(D) = \{u \in H^1 : u|_\Gamma = 0\}$$

► **Esempio 9.6.** Sia dato il problema

$$(9.60) \quad \begin{cases} \nabla^T(d \nabla u) - au + b = 0 \\ u = r \quad \text{su } \Gamma_1 \quad (\text{condizione essenziale}) \\ d \frac{\partial u}{\partial n} = -pu + q \quad \text{su } \Gamma_2 \\ \qquad \qquad \qquad (\text{condizione naturale}) \end{cases}$$



definito in una regione limitata  $D \subset \mathbb{R}^2$  con contorno  $\Gamma = \Gamma_1 \cup \Gamma_2$  Lipschitz-continuo. Le funzioni (note)  $d > 0$ ,  $a$ ,  $b$ ,  $r$ ,  $p$  e  $q$ , che in generale dipendono dalle variabili  $x, y$ , per semplicità sono supposte tali da garantire l'esistenza e unicità di una soluzione classica con derivate limitate. Ricordiamo inoltre che

$$\nabla^T(d \nabla u) = \operatorname{div}(d \operatorname{grad} u) = \frac{\partial}{\partial x} \left( d \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left( d \frac{\partial u}{\partial y} \right)$$

Scelta una funzione  $u_0 \in H^1$  tale che  $u_0 = r$  su  $\Gamma_1$  e posto  $u = u_0 + \bar{u}$ , con  $\bar{u} \in V = H_{0,\Gamma_1}^1$ , moltiplichiamo entrambi i membri dell'equazione differenziale per la generica funzione  $v \in V$  e integriamo poi in  $D$ . Otteniamo:

$$(9.61) \quad \int_D [\nabla^T(d \nabla u) - au + b] v \, dx \, dy = 0, \quad \forall v \in V$$

Quando il dominio  $D \subset \mathbb{R}^d$ ,  $d > 1$ , la formula di integrazione per parti viene sostituita dalla seguente identità di Green (vedi ad esempio [9.34, pag. 425])

$$\int_D [\nabla^T(d\nabla u)]v \, dD = - \int_D d(\nabla u)^T \nabla v \, dD + \int_{\Gamma} d \frac{\partial u}{\partial n} v \, d\Gamma$$

che, utilizzata in (9.61) richiamando la condizione naturale su  $\Gamma_2$ , ci consente di riscrivere la relazione (9.61) nella seguente forma debole:

$$(9.62) \quad \int_D [-d(\nabla u)^T \nabla v - auv + bv] \, dx \, dy + \int_{\Gamma_2} [-pu + q]v \, ds = 0 \quad \forall v \in V$$

che ammette in  $u_0 + V$  un'unica soluzione, coincidente nel nostro caso con quella (classica) del problema iniziale.

Supponiamo per semplicità che il contorno  $\Gamma$  sia una poligonale e, come negli esempi precedenti, di voler scegliere come approssimante una superficie continua e lineare a tratti, ovvero l'equivalente in  $\mathbb{R}^3$  delle poligonali nel piano. A tale fine partizioniamo il dominio (piano)  $D$  in triangoli e consideriamo le corrispondenti funzioni di base a supporto locale descritte nella prima parte del paragrafo. Sempre procedendo come negli esempi precedenti, definiamo i corrispondenti spazi  $V_M$  e  $W_M = u_0 + V_M$  e quindi l'approssimante

$$u_M(x, y) = u_0(x, y) + \sum_{j=1}^M c_j \varphi_j(x, y)$$

dove le funzioni base  $\varphi_j(x, y)$  si annullano su  $\Gamma_1$  e  $u_0(x, y) = r(x, y)$  su  $\Gamma_1$ , cosicché  $u_M(x, y) = r(x, y)$  su  $\Gamma_1$  qualunque siano i valori  $\{c_j\}$ .

Inserendo quest'ultima espressione in (9.62), e sostituendo  $V$  con  $V_M$ , otteniamo la corrispondente formulazione debole del metodo di Galerkin<sup>(†)</sup>:

$$(9.63) \quad \int_D [-d(\nabla u_M)^T (\nabla \varphi_i) - au_M \varphi_i + b\varphi_i] \, dx \, dy + \int_{\Gamma_2} [-pu_M + q]\varphi_i \, ds = 0 \\ i = 1, \dots, M$$

dalla quale ricaviamo il sistema lineare

$$Ac = f$$

nell'incognita  $c = (c_1, \dots, c_M)^T$ , con

$$(A)_{ij} = \int_D [d(\nabla \varphi_j)^T (\nabla \varphi_i) + a\varphi_j \varphi_i] \, dx \, dy + \int_{\Gamma_2} p\varphi_j \varphi_i \, ds$$

e

$$f_i = \int_D [b\varphi_i - d(\nabla u_0)^T (\nabla \varphi_i) - au_0 \varphi_i] \, dx \, dy + \int_{\Gamma_2} [q - pu_0]\varphi_i \, ds$$

---

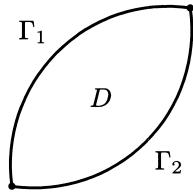
<sup>(†)</sup> Ricordiamo che  $\varphi_j = 0$  su  $\Gamma_1$ .

La matrice  $A$  è simmetrica. Se inoltre supponiamo  $a \geq 0$  e  $p \geq 0$ , e  $\Gamma_1$  non è vuota, allora è possibile dimostrare che  $A$  è anche definita positiva (vedi [9.27, pag. 192]) e quindi non singolare.  $\blacktriangleleft$

Se il contorno del dominio  $D$  fosse curvilineo, allora prima di procedere alla sua triangolazione occorre approssimare  $\Gamma$  con una poligonale  $\bar{\Gamma}$  che la interpoli in  $J_M$  punti scelti e sostituire  $D$  con il dominio  $\bar{D}$  delimitato da  $\bar{\Gamma}$ . Successivamente si procede come sopra. La convergenza dell'approssimante viene ottenuta facendo tendere a  $\infty$  simultaneamente sia  $J_M$  che  $M$ .

► **Esempio 9.7.** Consideriamo il problema (parabolico)

$$(9.64) \quad \begin{cases} \frac{\partial u}{\partial t} = \nabla^T(d \nabla u) - au + b & \text{in } \mathcal{R} = D \times (0, \infty) \\ u = r & \text{su } \Gamma_1 \quad \forall t > 0 \\ d \frac{\partial u}{\partial n} = -pu + q & \text{su } \Gamma_2 \quad \forall t > 0 \\ u(x, y, 0) = h(x, y) & \text{in } \bar{D} \end{cases}$$



con coefficienti (noti)  $d > 0, a, b$  indipendenti dal tempo. Anche in questo caso per semplicità supponiamo che i dati del problema siano tali da garantire l'esistenza e unicità di una soluzione classica con derivate limitate. Scelta quindi l'approssimante

$$u_M(x, y, t) = u_0(x, y, t) + \sum_{j=1}^M c_j(t) \varphi_j(x, y)$$

procedendo come negli esempi 9.5 e 9.6 perveniamo alla seguente formulazione debole del metodo di Galerkin:

$$\int_D \frac{\partial u_M}{\partial t} \varphi_i \, dx \, dy = \int_D [-d(\nabla u_M)^T (\nabla \varphi_i) - au_M \varphi_i + b \varphi_i] \, dx \, dy + \int_{\Gamma_2} [-pu_M + q] \varphi_i \, ds, \quad i = 1, \dots, M$$

da cui otteniamo un sistema di  $N$  equazioni differenziali ordinarie del tipo

$$(9.65) \quad Ac'(t) = -Bc(t) + f(t)$$

dove  $A, B$  sono due matrici di ordine  $M$  e  $c(t) = (c_1(t), \dots, c_M(t))^T$ . I valori iniziali da associare a quest'ultimo possono essere ottenuti imponendo le condizioni di ortogonalità

$$\int_D [u_M(x, y, 0) - h(x, y)] \varphi_i(x, y) \, dx \, dy = 0, \quad i = 1, \dots, M$$

oppure, scelti  $M$  punti  $(x_i, y_i)$ , imponendo le condizioni di interpolazione

$$u_M(x_i, y_i, 0) = h(x_i, y_i), \quad i = 1, \dots, M$$

Supponendo  $a \geq 0$  e  $p \geq 0$  è possibile verificare (vedi [9.27, pag. 228]) che le matrici (simmetriche)  $A$  e  $B$  sono rispettivamente definita positiva e semidefinita positiva. Inoltre, la matrice  $-A^{-1}B$  ha tutti gli autovalori reali e non positivi e il sistema (9.65) può risultare stiff.

Ribadiamo infine che per le formulazioni deboli costruite negli esempi 9.4–9.7 valgono osservazioni analoghe a quella fatta nella prima parte di pagina 367.

**Conclusione.** Prima di terminare quest'ultimo paragrafo, nel quale abbiamo presentato alcune delle idee che stanno alla base dei metodi agli elementi finiti, vogliamo menzionare alcuni importanti temi che non sono stati trattati, ma che il lettore interessato all'argomento potrà trovare illustrati nei testi riportati in bibliografia:

- lo studio della convergenza del metodo numerico quando il diametro massimo  $h$  degli elementi della partizione  $D_h$  del dominio  $D$  tende a zero; per esempio, in tutti i problemi considerati nel paragrafo precedente è possibile dimostrare che per le approssimanti scelte (localmente lineari) l'errore  $\|u - u_M\|$  si comporta come  $O(h^2)$  se la norma è quella dello spazio  $L^2$ , e come  $O(h)$  quando invece la norma scelta è quella del sottospazio  $H^1$ ;
- le strategie di decomposizione del dominio  $D$ , eventualmente di tipo adattativo in modo da addensare i nodi nelle vicinanze di possibili comportamenti irregolari del contorno del dominio e laddove la soluzione  $u$  presenta un comportamento “anomalo” (per esempio una variazione di gradiente repentina);
- il problema della *numerazione globale* dei nodi introdotti in  $D_h$ <sup>(†)</sup>, l’assemblaggio delle equazioni generate dal metodo di Galerkin e l’introduzione delle condizioni al contorno essenziali (vedi esempio pag. 374);
- la risoluzione dei sistemi lineari finali (sparsi e di grandi dimensioni);
- l’eventuale introduzione di *termini singolari* nella rappresentazione di  $u_M$ , al fine di riprodurre il corretto comportamento dell’incognita  $u$  quando quest’ultima presenta delle *singolarità*;
- l’utilizzazione (quando esiste) di una *formulazione variazionale* del problema fisico in esame (in luogo di quella differenziale), ovvero la caratterizzazione della soluzione  $u(x)$  quale punto di stazionarietà per un *funzionale* del tipo

$$I(u) = \int_D G(x, u) \, dx + \int_{\Gamma} g(s, u) \, d\Gamma$$

dove gli operatori  $G$  e  $g$  coinvolgono derivate della  $u$ , definito in un insieme di funzioni (distribuzioni)  $u$  ammissibili.

(†) Tale numerazione è responsabile della struttura della matrice dei coefficienti del sistema (9.43).

## Bibliografia

- [9.1] G. E. Forsythe, W. R. Wasow, *Finite-difference methods for partial differential equations*, John Wiley & Sons, New York, 1960.
- [9.2] R. Courant, D. Hilbert, *Methods of mathematical physics*, vol. II, Wiley-Interscience, New York, 1962.
- [9.3] R. Richtmeyer, K. Morton, *Difference methods for initial value problems*, John Wiley & Sons, New York, 1967.
- [9.4] A. R. Mitchell, *Computational methods in partial differential equations*, John Wiley & Sons, New York, 1967.
- [9.5] V. S. Vladimirov, *Equations of mathematical physics*, Marcel Dekker, New York, 1971.
- [9.6] O. C. Zienkiewicz, *The finite element method in engineering science*, McGraw-Hill, New York, 1971.
- [9.7] B. A. Finlayson, *The method of weighted residuals and variational principles*, Academic Press, New York, 1972.
- [9.8] G. Strang, G. J Fix, *An analysis of the finite element method*, Prentice-Hall, Englewood Cliffs, New Jersey, 1973.
- [9.9] W. F. Ames, *Numerical methods for partial differential equations*, Academic Press, London, 1977.
- [9.10] A. R. Mitchell, R. Wait, *The finite element method in partial differential equations*, Oxford University Press, 1979.
- [9.11] J. Gladwell, R. Wait, *A survey of numerical methods for partial differential equations*, Oxford University Press, 1979.
- [9.12] E. B. Becker, G. F. Carey, J. T. Olden, *Finite elements: an introduction*, Prentice-Hall, Englewood-Cliffs, New Jersey, 1981.
- [9.13] J. H. Ferziger, *Numerical methods for engineering application*, John Wiley & Sons, New York, 1981.
- [9.14] T. Meis, U. Marcowitz, *Numerical solution of partial differential equations*, Springer-Verlag, New York, 1981.
- [9.15] F. John, *Partial differential equations*, Springer-Verlag, New York, 1982.
- [9.16] L. Lapidus, G. Pinder, *Numerical methods of partial differential equations in science and engineering*, John Wiley & Sons, New York, 1982.
- [9.17] G. Birkhoff, R. E. Lynch, *Numerical solution of elliptic problems*, SIAM Publications, Philadelphia, 1983.
- [9.18] J. R. Rice, R. F. Boisvert, *Elliptic problem solving with ELLPACK*, Springer-Verlag, New York, 1983.
- [9.19] E. Zauderer, *Partial differential equations of applied mathematics*, John Wiley & Sons, New York, 1983.
- [9.20] O. Axelsson, V. A. Barker, *Finite element solution of boundary value problems. Theory and computation*, Academic Press, New York, 1984.
- [9.21] C. A. J. Fletcher, *Computational Galerkin methods*, Springer-Verlag series in computations physics, New York, 1984.

- [9.22] R. Garabedian, *Partial differential equations*, Chelsea Publ. Co., New York, 1986.
- [9.23] G. Strang, *Introduction to applied mathematics*, Wellesley-Cambridge Press, Cambridge, 1986.
- [9.24] H. Kardestuncer, D. H. Norrie, *Finite element handbook*, McGraw-Hill, New York, 1987.
- [9.25] C. Johnson, *Numerical solution of partial differential equations by the finite element method*, Cambridge University Press, Cambridge, 1987.
- [9.26] M.B Allen, I. Herrera, G. F. Pinder, *Numerical modeling in science and engineering*, John Wiley & Sons, New York, 1988.
- [9.27] G. Sewell, *The numerical solution of ordinary and partial differential equations*, Academic Press, London, 1988.
- [9.28] P. A. Raviart, J. M. Thomas, *Introduzione all'analisi numerica delle equazioni alle derivate parziali*, Masson, Milano, 1989.
- [9.29] J. Strikwerda, *Finite difference schemes and partial differential equations*, Wadsworth & Brooks-Cole, 1989.
- [9.30] A. Quarteroni, A. Valli, *Numerical approximation of partial differential equations*, Springer-Verlag, Berlin, 1994.
- [9.31] J. W. Thomas, *Numerical partial differential equations: finite difference methods*, Springer-Verlag, New York, 1995.
- [9.32] L. C. Evans, *Partial differential equations*, Graduate Studies in Mathematics, vol. 19, American Mathematical Society, Providence, Rhode Island, 1998.
- [9.33] A. Quarteroni, *Modellistica numerica per problemi differenziali*, Springer-Verlag Italia, Milano, 2003.
- [9.34] P. Šolín, *Partial differential equations and the finite element method*, John Wiley & Sons, Hoboken, New Jersey, 2006.
- [9.35] B. Gustafsson, *High order difference methods for time dependent PDE*, Springer-Verlag, Berlin, 2008.

## Esercizi proposti

**9.1.** Classificare le equazioni

$$\begin{aligned} u_t &= -(2x+1)e^{x+t} + \frac{1}{2}u + \frac{1}{2}u_{xx} \\ x^2u_{xx} - t^2u_{tt} - \frac{xu}{2}u_x - x^2t^2u^3 &= 0 \\ u_{xx} + (1+x^2)u_x + u_{yy} &= 6xye^{x+y} \end{aligned}$$

**9.2.** Determinare le linee caratteristiche dell'equazione

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial}{\partial x} \left( c^2(x) \frac{\partial u}{\partial x} \right)$$

**9.3.** Trovare le linee caratteristiche dell'equazione

$$u_{tt} + au_{xt} + \frac{1}{4}(a^2 - 4)u_{xx} = 0$$

dove  $a$  è una costante positiva, e classificare l'equazione stessa.

**9.4.** Studiare la stabilità del seguente schema alle differenze finite

$$u_{i,j+1} = \frac{1}{2}(u_{i+1,j} + u_{i-1,j}) - \frac{ak}{2h}(u_{i+1,j} - u_{i-1,j})$$

proposto da P. Lax per la risoluzione del problema (9.13).

**9.5.** Considerare il problema corrispondente a (9.13) quando  $a < 0$ . Proporre uno schema incondizionatamente stabile.

**9.6.** Studiare la consistenza e la stabilità dello *schemma di Du Fort-Frankel*

$$(1 + 2\lambda)u_{i,j+1} = (1 - 2\lambda)u_{i,j-1} + 2\lambda(u_{i+1,j} + u_{i-1,j}), \quad \lambda = \frac{k}{h^2}$$

proposto per la risoluzione dell'equazione del calore  $u_t = u_{xx}$ .

Esaminare anche l'influenza sulla stabilità delle possibili perturbazioni presenti sulle condizioni al contorno del problema.

**9.7.** Proporre uno schema numerico alle differenze finite per la risoluzione dell'*equazione di Burgers*

$$u_t + uu_x = \nu u_{xx}, \quad 0 < x < 1, \quad t > 0, \quad \nu \text{ costante}$$

con condizioni

$$\begin{aligned} u(x, 0) &= \sin \pi x & 0 \leq x \leq 1 \\ u(0, t) &= u(1, t) = 0 & t > 0 \end{aligned}$$

**9.8.** Integrare con un metodo alle differenze finite il problema

$$\begin{cases} \frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left( (1 + x^2) \frac{\partial u}{\partial x} \right) & -1 < x < 1, \quad t > 0 \\ u(x, 0) = 100(1 - |x|) & -1 \leq x \leq 1 \\ u(-1, t) = u(1, t) = 0 & t > 0 \end{cases}$$

sino all'istante  $t = 0.2$ .

**9.9.** Risolvere il problema

$$\begin{cases} u_t = u_{xx} + u_{yy} & 0 < x < 2, \quad 0 < y < 1, \quad t > 0 \\ u(x, y, 0) = \sin \pi x \sin \pi y & 0 < x < 2, \quad 0 < y < 1 \\ u(0, y, t) = u(2, y, t) = 0 & t > 0 \\ u(x, 0, t) = u(x, 1, t) = 0 & \end{cases}$$

utilizzando il metodo delle direzioni alternate (ADI). Confrontare le approssimazioni trovate con la soluzione esatta del problema:

$$u(x, y, t) = e^{-2\pi^2 t} \sin \pi x \sin \pi y$$

**9.10.** Risolvere il seguente problema

$$\begin{cases} u_t = u_{xx}, & 0 < x < 1, \quad t > 0 \\ u(x, 0) = f(x) \\ u(0, t) = g_0(t) \\ u_x(1, t) = g_1(t) \end{cases}$$

con il metodo di Crank-Nicolson.

**9.11.** Risolvere il problema precedente con un metodo delle linee.

**9.12.** Risolvere il problema

$$\begin{cases} u_{xx} + u_{yy} = (x^2 + y^2)u \\ u(x, 0) = u(0, y) = 1 \\ u(x, 1) = e^{-x} \\ u(1, y) = e^{-y} \end{cases} \quad 0 \leq x \leq 1, \quad 0 \leq y \leq 1$$

con uno schema alle differenze finite e confrontare le approssimazioni ottenute con la soluzione esatta  $u(x, y) = e^{-xy}$ .

**9.13.** Risolvere, con uno schema alle differenze finite, il problema

$$\begin{cases} u_{xx} + u_{yy} - 32u = 0, & -1 < x < 1, \quad -1 < y < 1 \\ u = 0, & y = 1, \quad -1 \leq x \leq 1 \\ u = 1, & y = -1, \quad -1 \leq x \leq 1 \\ u_x = -u/2, & x = 1, \quad -1 < y < 1 \\ u_x = u/2, & x = -1, \quad -1 < y < 1 \end{cases}$$

**9.14.** Risolvere, con un metodo alle differenze finite, il problema

$$\begin{cases} \frac{\partial^2 T}{\partial r^2} + \frac{1}{r} \frac{\partial T}{\partial r} = (1 - r^2) \frac{\partial T}{\partial z}, & 0 < r < 1, \quad 0 < z < \infty \\ T = 0, & z = 0, \quad 0 \leq r \leq 1 \\ T = 1, & r = 1, \quad z > 0 \\ \frac{\partial T}{\partial r} = 0, & r = 0, \quad z > 0 \end{cases}$$

**9.15.** Scrivere l'*equazione biarmonica*

$$\frac{\partial^4 u}{\partial x^4} + 2 \frac{\partial^4 u}{\partial x^2 \partial y^2} + \frac{\partial^4 u}{\partial y^4} = f$$

come sistema di due equazioni di ordine 2 e proporre uno schema di discretizzazione alle differenze finite.

**9.16.** Determinare un'approssimazione dell'autovalore di modulo minimo, e un'approssimazione della corrispondente autosoluzione, del problema

$$\begin{cases} -\Delta u = \lambda u & \text{in } D \\ u = 0 & \text{su } \Gamma \end{cases}$$

dove  $D$  è il triangolo di vertici i punti  $(0, 0)$ ,  $(0, 1)$ ,  $(1, 0)$ .

**9.17.** Risolvere con il metodo delle differenze finite il seguente problema:

$$\begin{cases} -\Delta u = x^2 + y^2 & \text{in } D \\ u = 1 & \text{su } \Gamma \end{cases}$$

La simmetria di  $u(x, y)$  consente di risolvere il problema considerando solo la porzione di  $D$  situata nel  $I$  quadrante.

**9.18.** Risolvere il seguente problema

$$\begin{cases} y''(x) + 3y'(x) - 2(x+1)y(x) = 0, & 0 < x < 1 \\ y(0) - y'(0) = 1 \\ y(1) + 2y'(1) = 2 \end{cases}$$

utilizzando la formulazione debole del metodo di Galerkin. Prendere come approssimante una poligonale definita su una partizione ottenuta suddividendo l'intervallo  $(0, 1)$  in 5 parti uguali.

**9.19.** Sia dato il problema:

$$\begin{cases} T''(x) - \sigma T(x) = 0, & 0 < x < 1 \\ T(0) = 10 \\ T'(1) = 0 \end{cases}$$

Ricavare dapprima la formulazione debole del problema; successivamente, determinare con il metodo di Galerkin un'approssimazione dell'incognita  $T(x)$  di tipo lineare a tratti (poligonale), attribuendo, per esempio, al coefficiente  $\sigma$  i valori seguenti:  $0.5, 10^4$ .

Riscrivere la formulazione debole del problema nel caso in cui la condizione nell'estremo  $x = 1$  sia data da:  $T(1) + T'(1) = 0$ .

**9.20.** Consideriamo l'equazione

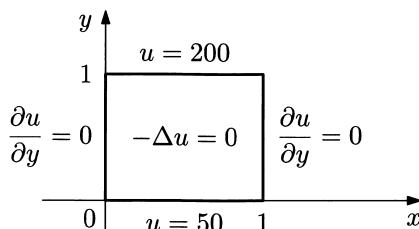
$$u_t + u_x - \nu u_{xx} = 0$$

nella regione  $-1 < x < 1, t > 0$ , con condizioni

$$\begin{aligned} u(x, 0) &= \begin{cases} 1, & -1 \leq x \leq 0 \\ 0, & 0 < x \leq 1 \end{cases} \\ u(-1, t) &= 1 \\ u(1, t) &= 0 \end{aligned}$$

con  $\nu = 1$  prima e  $\nu = 10^{-2}$  poi. Ricavare la formulazione debole del problema e quindi applicare il metodo di Galerkin agli elementi finiti.

**9.21.** Sia dato il problema ellittico



Discretizzare il dominio con elementi quadrati prima, e triangolari poi. Utilizzando basi lagrangiane locali di grado 1, determinare con il metodo di Galerkin un'approssimazione localmente polinomiale della soluzione  $u$ .

Osservare come la struttura del sistema (lineare) finale dipenda dalle numerazione globale dei nodi.

**9.22.** Risolvere il problema proposto in 9.9 con un metodo agli elementi finiti.

## Bibliografia Generale

- [1] E. Isaacson, H. B. Keller H.B., *Analysis of numerical methods*, John Wiley & Sons, New York, 1966.
- [2] L. Gatteschi, *Lezioni di analisi numerica*, Levrotto & Bella, Torino, 1970.
- [3] R. W. Hamming, *Numerical methods for scientists and engineers*, McGraw-Hill, New York, 1973.
- [4] G. Dahlquist, A. Björck, *Numerical methods*, Prentice-Hall, Englewood Cliffs, New Jersey, 1974.
- [5] F. B. Hildebrand, *Introduction to numerical analysis*, McGraw-Hill, New York, 1974.
- [6] J. Stoer, R. Bulirsch, *Introduzione all'analisi numerica*, Vol. 1, 2, Zanichelli, Bologna, 1975.
- [7] G. E. Forsythe, M. A. Malcolm, C. B. Moler, *Computer methods for mathematical computations*, Prentice-Hall, Englewood Cliffs, New Jersey, 1977.
- [8] K. E. Atkinson, *An introduction to numerical analysis*, John Wiley & Sons, New York, 1978.
- [9] A. Ralston, P. Rabinowitz, *A first course in numerical analysis*, McGraw-Hill, New York, 1978.
- [10] J. S. Vandergraft, *Introduction to numerical computation*, Academic Press, New York, 1978.
- [11] S. D. Conte, C. De Boor, *Elementary numerical analysis; an algorithmic approach*, McGraw-Hill, New York, 1980.
- [12] F. Fontanella, A. Pasquali, *Calcolo numerico; metodi e algoritmi*, Vol. 1, 2, Pitagora Editrice, Bologna, 1982.
- [13] L. W. Johnson, R. D. Riess, *Numerical analysis*, Addison-Wesley, Reading Massachusetts, 1982.
- [14] J. R. Rice, *Numerical methods, software, and analysis*, McGraw-Hill, New York, 1983.

- [15] D. Kahaner, C. Moler, S. Nash, *Numerical methods and software*, Prentice-Hall, Englewood Cliffs, New Jersey, 1988.
- [16] V. Comincioli, *Analisi numerica: metodi, modelli, applicazioni*, McGraw-Hill, Milano, 1990.
- [17] R. Bevilacqua, D. Bini, M. Capovani, O. Menchi, *Metodi numerici*, Zanichelli, Bologna, 1992.
- [18] G. Gambolati, *Lezioni di metodi numerici per ingegneria e scienze applicate*, Libreria Cortina, Padova, 1994.
- [19] W. Gautschi, *Numerical analysis: an introduction*, Birkhäuser, Boston, 1997.
- [20] G. Monegato, *Fondamenti di calcolo numerico*, C.L.U.T. Editrice, Torino, 1998.

# Indice analitico

Accelerazione  
formula di — di Aitken, 200

Adams-Bashfort  
metodo di —, 274

Adams-Moulton  
metodo di —, 266

Adattativa  
strategia di integrazione —, 241

ADI  
metodo —, 341

Affine  
trasformazione —, 249, 359

Aitken  
metodo di accelerazione di —, 199

Algoritmo  
stabilità di un —, 14

Amplificazione  
coefficienti di —, 15  
fattore di —, 40, 182

Aritmetica, 1

Arrotondamento  
errori di —, 6  
tecniche di —, 7

Asintoticamente stabile, 257  
— in prima approssimazione, 258

Assoluta  
intervallo di stabilità —, 286  
regione di stabilità —, 284

Assolutamente stabile  
metodo —, 284

Assoluto  
errore —, 8

$A$ -stabile, 288

$A(\alpha)$ -stabile, 288

$A(0)$ -stabile, 289

Automatiche  
routine —, 240

Autovalore  
calcolo di un — particolare, 98

Autovalori  
— di una matrice, 30  
calcolo degli — di una matrice tridiagonale simmetrica, 112  
problemi agli —, 310

Bairstow  
metodo di —, 208

Banda  
matrice a —, 27

Base, 124  
— lagrangiana locale, 356, 360

- del sistema di numerazione, 1
- funzioni —, 165
- ortogonale, 171
- ortonormale, 171
- BDF
  - metodi —, 290
- Ben condizionato
  - problema —, 14
- Ben posto
  - problema —, 254, 310
- Binaria
  - cifra —, 2
- Binario
  - sistema —, 2
- Binet
  - formule di —, 28
- Bisezione
  - metodo di —, 183
- Bit, 2
- Blocchi
  - matrice a —, 26
  - matrice tridiagonale a —, 75
- Broyden, 203
- B-spline* cubiche, 178
- Calore
  - equazione del —, 315, 334, 339
- Cancellazione numerica, 10, 192
- Caratteristica
  - di un numero, 4
  - curva —, 313
  - equazione —, 30, 277, 282
- Caratteristiche, 311
  - curve —, 311
  - direzioni —, 313
  - metodo delle —, 327
- Caratteristico
  - polinomio —, 91
- Cardinali
  - spline cubiche —, 163
- Cauchy
  - problema di —, 308
- Cauchy-Schwarz
  - disuguaglianza di —, 32
- Chebyshev
  - polinomi di — di prima specie, 228
  - polinomi di — di seconda specie, 228
- Choleski
  - metodo di —, 60, 168
- Cifre significative
  - perdita di —, 10
- Classica
  - soluzione —, 307
- Classificazione
  - delle equazioni quasi-lineari di ordine 2, 311
- Clenshaw
  - algoritmo di —, 226
- Collocazione
  - metodo di —, 299, 352
- Complessi
  - sistemi —, 62
- Composte
  - formule —, 238
- Condizionamento
  - del problema, 14, 181
  - numero di —, 15, 40
- Condizione
  - di Dirichlet, 346
  - di Neumann, 346
  - essenziale, 374
  - naturale, 374
- Condizioni
  - ai limiti, 291
  - al bordo, 308
  - al contorno, 319
  - iniziali, 251, 328
- Consistente, 261, 272, 343
  - condizionatamente —, 343
  - incondizionatamente —, 344
- Consistenza, 342
  - di un metodo, 261
  - ordine di —, 261, 272
- Convergente
  - formula —, 222

- metodo —, 267, 276, 345
- Convergenza**
- degli schemi alle differenze finite per problemi ai valori iniziali, 342
  - dei metodi multistep, 276
  - di un metodo iterativo, 67
  - di un metodo one-step, 267
  - di un processo di interpolazione, 132
  - di una formula di quadratura, 222, 231
  - ordine di —, 186
  - test di —, 195
- Convergenza uniforme**, 125
- Coordinate baricentriche**, 362
- Cotes**
- numeri di —, 223
- Crank-Nicolson**
- metodo di —, 337
- Debole**
- soluzione —, 307
- Decomposizione**
- $QR$ , 115
  - ai valori singolari, 110
  - di Gauss, 48
- Definita positiva**
- matrice simmetrica —, 27
- Derivata normale**, 320
- Derivazione numerica**, 173
- formula di —, 260
- Determinante**, 28
- calcolo del — di una matrice, 52
- DFT**, 148
- Diagonale**
- matrice —, 27
- Diagonale dominante**
- matrice a —, 28
  - matrice a — in senso debole, 348
  - matrice a — per colonne, 28
  - matrice a — per righe, 28
- Diagonalizzabile**
- matrice —, 31
- Diametro**
- di un elemento, 365
- Differenze divise**, 135, 139
- formula di Newton alle —, 135
- Differenze finite**, 142
- progressive, 142
  - regressive, 142
  - formula di Newton alle —, 142
  - metodo delle —, 293
- Diretti**
- metodi —, 38, 41
- Direzioni alternate**
- metodo delle —, 341
- Dirichlet**
- condizione di —, 346
  - problema di —, 346
- Discretizzazione**, 222
- errore di —, 16
  - parametri di —, 343, 345
- Dominio**
- di dipendenza, 325, 329
  - di influenza, 325, 329
- Elementi**
- fondamentali, 359
  - isoparametrici, 364
  - rettangolari, 360
  - triangolari, 361
- Elementi finiti**, 355
- metodi agli elementi finiti, 355
- Eliminazione di Gauss**
- metodo di —, 41
- Ellittico**
- equazioni di tipo —, 346
- Equazione**
- alle differenze, 272
  - caratteristica, 30, 277, 313
  - del calore, 315, 334, 339
  - della corda vibrante, 315
  - delle onde, 328
  - di Laplace, 310, 346
  - di Poisson, 309, 346
  - ellittica, 315
  - integrale di Fredholm, 243

- iperbolica, 315
- parabolica, 315
- test, 284
- Equazioni**
  - algebriche, 205
  - alle derivate parziali, 307
  - di tipo ellittico, 346
  - di tipo iperbolico, 321
  - di tipo parabolico, 334
  - differenziali ordinarie, 251
  - non lineari, 181
  - normali, 167
  - classificazione delle — quasi-lineari di ordine 2, 311
  - sistemi di — lineari, 37
  - sistemi di — non lineari, 201
- Equazioni normali**
  - sistema delle —, 167, 171
- Equilibratura, 47**
- Errore**
  - assoluto, 8
  - di discretizzazione, 16
  - di interpolazione polinomiale, 129
  - di troncamento, 343
  - di una formula di quadratura, 220, 236
  - globale, 269
  - locale di troncamento, 318
  - locale unitario di troncamento, 261, 272
  - relativo, 8
- Esponente di un numero, 4**
- Esponenziali**
  - somme —, 123
- Estrapolazione**
  - di Richardson, 268
  - processi di —, 238
- Euclidea**
  - norma —, 32
- Eulero**
  - metodo di —, 262
  - metodo di — implicito, 290
- metodo di — modificato, 264**
- Fattorizzazione**
  - $LL^T$ , 59, 168
  - $LU$ , 48
  - $QR$ , 103
- Fehlberg**
  - metodo di —, 269
- FFT**
  - algoritmo —, 148
- Fibonacci**
  - numeri di —, 189
- Floating-point**
  - rappresentazione —, 4
- Formula di quadratura, 219**
- Formulazione**
  - variazionale, 377
- Formule prodotto**
  - per integrali multipli, 244
- Fourier**
  - analisi discreta di —, 148
  - coefficienti generalizzati di —, 172
  - sintesi discreta di —, 148
  - trasformata di — discreta, 148
- Fredholm**
  - equazione integrale di —, 243
- Frobenius**
  - norma di —, 33
- Frontiera**
  - punti di —, 316
- Funzionale lineare, 230**
- Funzione peso, 222, 225, 229**
- Funzioni**
  - base, 351
  - localmente polinomiali, 359
  - polinomiali a tratti, 153
  - razionali, 122
  - spline, 153
- Galerkin**
  - criterio di —, 351
  - formulazione debole di —, 354
- Gauss**

- decomposizione di —, 48  
formule di —, 229  
metodo delle eliminazioni di —, 41
- Gauss-Chebyshev  
formule di —, 232
- Gauss-Siedel  
metodo di —, 69  
metodo di — a blocchi, 349
- Gaussiane  
costruzione delle formule —, 233  
formule —, 223  
formule di quadratura —, 229
- Gershgorin  
teorema di —, 88
- Gradiente, 318
- Gradiente coniugato  
metodo del —, 76
- Grado di precisione  
— di una formula di quadratura, 221
- Gram-Schmidt  
processo di ortogonalizzazione di —, 170
- Hermite  
formula di interpolazione di —, 134  
polinomi di —, 228
- Hessenberg  
matrice di —, 100
- Heun  
metodo di —, 264
- Hilbert  
matrice di —, 40
- Horner  
algoritmo di —, 206
- Householder  
trasformazioni —, 99  
trasformazioni di —, 100, 103
- Identità  
matrice —, 26
- Influenza  
dominio di —, 325, 329
- Iniziali
- condizioni —, 251, 328  
problemi ai valori —, 308
- Instabile  
algoritmo —, 14
- Integrali  
— multipli, 244  
calcolo di —, 219
- Interpolatorie  
formule — pesate, 222  
formule di quadratura —, 220
- Interpolazione  
— con funzioni di più variabili, 163  
— con funzioni polinomiali a tratti, 153  
— con funzioni spline cubiche, 165  
— polinomiale, 126  
— trigonometrica, 145  
formula di — di Hermite, 134  
formula di — di Lagrange, 126  
formula di — di Newton, 138, 144, 187  
polinomio di —, 127, 128, 131  
polinomio di — di Newton, 140
- Intervalli infiniti  
integrazione su —, 242
- Inversa  
— generalizzata di Moore-Penrose, 168  
matrice —, 29
- Irriducibile  
matrice —, 71
- Isoparametrici  
elementi —, 364
- Iterativi  
metodi —, 38, 66, 196, 204
- Jacobi  
metodo di —, 68  
polinomi di —, 227
- Jordan  
metodo di —, 45
- Kronrod A. S., 236
- Lagrange  
formula di interpolazione di —, 126

- multiplicatori di —, 212
- polinomi fondamentali di —, 128
- Laguerre
  - polinomi di —, 228
- Laplace
  - equazione di —, 310
- Laplaciano, 318
- Legendre
  - polinomi di —, 227
- Linee
  - metodo delle —, 338
- Lipschitz
  - condizione di —, 253
- Lipschitziana
  - funzione —, 253
- Lobatto
  - formule del tipo —, 233
- LU*
  - fattorizzazione —, 48
- Macchina
  - numeri di —, 3
  - operazioni di —, 9
- Mantissa, 4
- Matrice, 27
  - Hessenberg, 27
  - a banda, 27
  - a blocchi, 26
  - a diagonale dominante, 28
  - a diagonale dominante in senso debole, 348
  - antihermitiana, 27
  - antisimmetrica, 27
  - coniugata, 26
  - definita positiva, 27
  - di Hilbert, 40
  - di Vandermonde, 41
  - diagonale, 27
  - diagonalizzabile, 31
  - hermitiana, 27
  - inversa, 29
  - involutiva, 30
  - irriducibile, 71
  - non singolare, 29
  - nulla, 25
  - ortogonale, 30
  - pseudo inversa, 168
  - quasi-triangolare, 115
  - semidefinita positiva, 27
  - simmetrica, 27
  - singolare, 29
  - trasposta, 26
  - triangolare, 27
  - triangolare superiore a blocchi, 117
  - tridiagonale, 27
  - tridiagonale a blocchi, 75, 348
  - unità o identità, 26
  - unitaria, 30
- matrice
  - di rigidezza, 368
- Matrici
  - dense, 38
  - simili, 99
  - sparse, 38
  - operazioni tra —, 24
- Minimax
  - criterio —, 123
- Minimi quadrati, 123, 351
  - metodo dei —, 165, 170
- Momenti, 225
- Multistep
  - comportamento locale dei metodi —, 272
  - convergenza dei metodi —, 276
  - metodi —, 271
  - metodi — a passo variabile, 278
- Neumann
  - condizione di —, 346
- Newton
  - formula di interpolazione di —, 138, 144, 187
  - polinomio di interpolazione di —, 140
- Newton-Cotes
  - formule di —, 223
- Newton-Raphson

- metodo di —, 187, 190
- Nodi
  - di interpolazione, 132
  - di un reticolo, 316, 330, 337
  - di una formula di quadratura, 219
- Nordsieck, 279
- Norma, 170
  - $p$ , 32
  - compatibile, 33
  - di Frobenius, 33
  - di matrice, 31
  - di vettore, 31
  - euclidea, 32
  - indotta, 33
  - infinito, 32, 125
  - naturale, 33
  - spettrale, 33
- Numerazione
  - sistemi di —, 1
- Numeri
  - di macchina, 3
  - interi, 4
  - reali, 4
- Numero di condizionamento, 15, 40
- One-step
  - metodi —, 258, 259
- Operatore, 318, 350
- Operazioni
  - di macchina, 9
  - tra matrici, 24
- Ordine
  - di consistenza, 261, 272
  - di convergenza, 186
  - di un metodo, 187, 261
  - di una matrice, 25
- Ortogonale
  - matrice —, 30
  - sistema —, 170
- Ortogonalni
  - polinomi —, 225
  - vettori —, 29
- Ortogonalizzazione
- processo di — di Gram-Schmidt, 170
- Ortonormale
  - sistema —, 170
- Ottimizzazione, 211
- Overflow, 10
- Passo
  - scelta del — di integrazione, 268
- Patterson T. N. L., 240
- Pesi
  - di una formula di quadratura, 219
- Peso
  - funzione —, 222, 225, 229
- Pivot
  - elementi —, 43
- Pivoting, 45
  - completo, 46
  - parziale, 46
- Poisson
  - equazione di —, 309
- Polinomi ortogonali, 225
- Polinomio
  - caratteristico, 91
- Potenze
  - metodo delle —, 91
- Potenze inverse
  - metodo delle —, 98
- Precisione
  - di macchina, 8
  - doppia, 5
  - semplice —, 5
- Precondizionamento, 80
- Precondizionatore, 80
- Previsore-correttore
  - metodi —, 277
- Problema
  - (con valori) ai limiti, 252, 291
  - a valori iniziali, 253
  - ai valori iniziali, 308
  - ben condizionato, 14
  - ben posto, 254
  - di Cauchy, 308
  - di Dirichlet, 346

- mal condizionato, 14
- Problemi**
  - agli autovalori, 310
- Procedimento iterativo, 66, 196
- Prodotto scalare, 29, 351
- Puntamento
  - metodo di —, 296
- QR***
  - fattorizzazione — di una matrice, 103
  - metodo —, 114
- Quadratura, 222
  - formula di —, 219
  - formule di —, 242
- Quasi-lineare
  - equazione —, 307
- Quoziente di Rayleigh, 94
- Radau
  - formule di —, 233
- Radici
  - reali di equazioni non lineari, 183
- Radici reali
  - di equazioni algebriche, 205
- Raffinamento iterativo, 60, 66
- raggio spettrale, 31
- Rango di una matrice, 29
- Rappresentazione
  - binaria, 5
  - dei numeri in un calcolatore, 3
  - normalizzata, 4
- Rayleigh
  - quoziente di —, 94
- Regione
  - di stabilità assoluta, 284
- Regula falsi, 187, 190
- Residui pesati
  - metodi dei —, 350
- Richardson
  - estrapolazione di —, 268
- Ricorrenza
  - relazione di — a tre termini, 225
- Ridotto
- polinomio —, 206
- Rilassamento
  - metodo di —, 74
- Runge-Kutta
  - metodi —, 259
- Schemi alle differenze finite, 294, 342
- Secanti
  - metodo delle —, 186
- Serendipity, 361
- Shooting
  - metodo —, 296
- Similitudine
  - trasformazione di —, 99
- Simpson
  - formula composta di —, 238
  - formula di —, 224
- Sistema**
  - binario, 2
  - decimale, 1
  - delle equazioni normali, 167, 171
  - ortogonale, 225
  - ortonormale, 29, 170
  - triangolare, 41
  - tridiagonale simmetrico, 158
- Sistema decimale, 1
- Sistema di numerazione, 1
- Sistemi**
  - di equazioni lineari, 37
  - di equazioni non lineari, 201
- SOR**
  - metodo —, 73
- Spline**
  - bicubiche, 165
  - cubica, 155
  - funzioni —, 153
- Stabile**
  - curva —, 257
  - metodo condizionatamente —, 325
  - metodo incondizionatamente —, 327, 337
- Stabilità, 279, 321, 342
  - assoluta, 277

- di un algoritmo, 14
- relativa, 287
- Stiff**
  - sistemi —, 287
- Supporto**
  - di una funzione, 355
  - locale, 355
- Tangenti**
  - metodo delle —, 186
- Trapezi**
  - formula composta dei —, 238, 242
  - metodo dei —, 276
- Trapezio**
  - formula del —, 224
- Trasformata di Fourier discreta**, 148
- Trasformazioni**
  - di similitudine, 99
- Ultrasferici**
  - polinomi —, 227
- Underflow**, 10
- Valori singolari**
  - decomposizione ai —, 110
- Variazionale**
  - formulazione —, 377
- Velocità di convergenza**, 71, 78, 80, 185, 187
- Vettore**
  - colonna, 23
  - riga, 23
- Zero**
  - stabilità, 277

