APPROSSIMAZIONE ED ERRORI

Rappresentazione approssimata di un numero: cifre significative

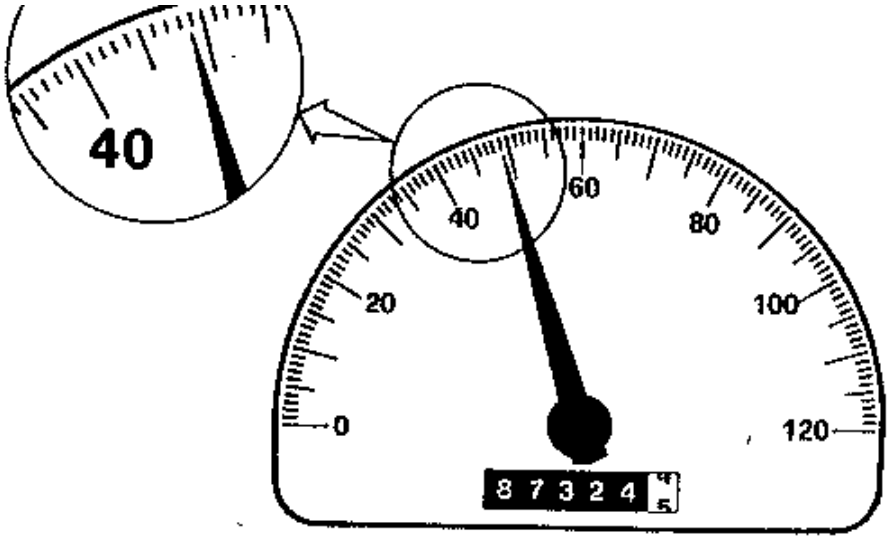


Figura 3.1 Tachimetro e contachilometri di automobile come applicazione del concetto di cifra significativa.

Il concetto di "cifre significative" è strettamente correlato a quello della rappresentazione approssimata di un numero. Nella figura viene mostrato il tachimetro ed il contachilometri di un'automobile.
Il tachimetro (strumento analogico) indica che la velocità dell'automobile è compresa tra 48 e 49 Km/h. Osservando che la lancetta ha superato il punto di mezzo delle due tacche è possibile dire che la velocità è di circa 49 km/h. Diventa invece azzardato dare un'indicazione sulla prima cifra decimale: la velocità potrebbe essere di 48.7 come di 48.8 km/h. In altre parole solo le prime due cifre sono significative, mentre lo strumento non permette di valutare chiaramente la terza.
Nel caso del contachilometri (strumento digitale) si leggono chiaramente le prime sei cifre significative: l'auto ha percorso poco meno di 87324,5 km; è la settima cifra che non può essere individuata.

Il concetto di cifra significativa è pertanto legato all'attendibilità di un valore numerico per rappresentare un'assegnata grandezza. Rappresenta il numero di cifre "certe" più un'ulteriore cifra stimata la cui precisazione abbia ancora significato.

Si noti che gli zeri iniziali non sono significativi in quanto vengono usati solo per posizionare il punto decimale.

Il concetto di cifra significativa ha due importanti implicazioni nello studio dei metodi numerici

1. I metodi numerici danno risultati approssimati. Bisogna, pertanto, definire dei criteri che ci permettano di specificare quale attendibilità vada accreditata ai nostri risultati approssimati. Col criterio delle cifre significative, potremmo dire, per esempio, che l'approssimazione è accettabile se sono corrette le prime 4 cifre significative.

2. Sebbene alcuni numeri come π oppure $\sqrt{7}$ hanno un significato ben preciso, non possono essere esattamente rappresentati con un numero finito di cifre significative. L'omissione di alcune cifre significative è detta errore di arrotondamento

Esempio

Rappresentiamo il risultato della divisione tra i numeri 20 e 3:

- per una rappresentazione **esatta** occorrerebbero infinite cifre significative: 6.666666… a meno di ricorrere ad una notazione convenzionale 6.$\underline{6}$ ovvero 6.(6) (la cifra 6 si ripete all'infinito), o di lasciarlo in modo involuto (ad es. nella forma frazionale 20/3).

- Un'alternativa è quella di utilizzare una rappresentazione **approssimata** conservando solo alcune cifre: ad esempio con 3 cifre il risultato della divisione è 6.67 oppure 6.66 a seconda se si decide di effettuare l'approssimazione per **arrotondamento** oppure per **taglio** (ovvero tenendo conto o meno del valore delle cifre dopo la terza)

  N.B. in alternativa a taglio in letteratura si usa spesso impropriamente il termine troncamento che va invece utilizzato, come vedremo per definire una diversa sorgente di errore.

Rappresentazione approssimata dei numeri sul calcolatore

Virgola fissa e virgola mobile

I primi calcolatori elettronici operavano in virgola fissa, vale a dire che nei calcoli si lavora con un numero fisso di cifre decimali.
I calcolatori moderni operano invece tutti in virgola mobile, cioè con un numero fisso di cifre significative.

Lavorando con tre decimali in virgola fissa e utilizzando tre cifre significative in virgola mobile, supponiamo ad esempio di voler eseguire il prodotto:

3.002*0.001432 = 0.00429886400000 RISULTATO ESATTO

In virgola fissa si ottiene:

3.002*0.001 = 0.00300200000000 VIRGOLA FISSA

In virgola mobile si ottiene:

3.00*0.00143 =0.00429 VIRGOLA MOBILE

Come si vede, il risultato in virgola mobile è più vicino al risultato esatto.
Va subito sottolineato che l'uso della virgola mobile non risolve tutti i problemi di calcolo: **va bene con le operazioni di addizione, moltiplicazione e divisione**, ma crea **problemi** con l'operazione di **sottrazione**, come si vedrà in seguito.

I NUMERI NEL COMPUTER

Per migliorare l'efficienza nella rappresentazione dei numeri sul calcolatore **viene utilizzata** la base **binaria** (oppure quella esadecimale) anziché quella decimale.

Il computer, con 7 cifre significative, indica il numero reale x=-645.0234

come: -0.6450234E3

dove	0.6450234	è la **mantissa $0 \leq m \leq 1$**
e	3	è l'**esponente**

Il numero viene però archiviato e trattato nei calcoli nella forma:

$$x = - m * 2^{E}$$

dove	la **mantissa *m***	**$0 \leq m \leq 1$**
e	l'**esponente E**	

sono rappresentati in forma binaria.

Per rappresentare il segno si usa un ulteriore bit.

Il numero di cifre decimali significative restituite dal calcolatore dipende:

1. dal tipo del numero da rappresentare (Integer, Real, Double)
2. dal linguaggio di programmazione (Basic, Fortran, C)
3. dal calcolatore (PC, workstation, Mini, Supercomputer)

RAPPR. APPROSSIMATA DI UNA GRANDEZZA FISICA: PRECISIONE ED ACCURATEZZA

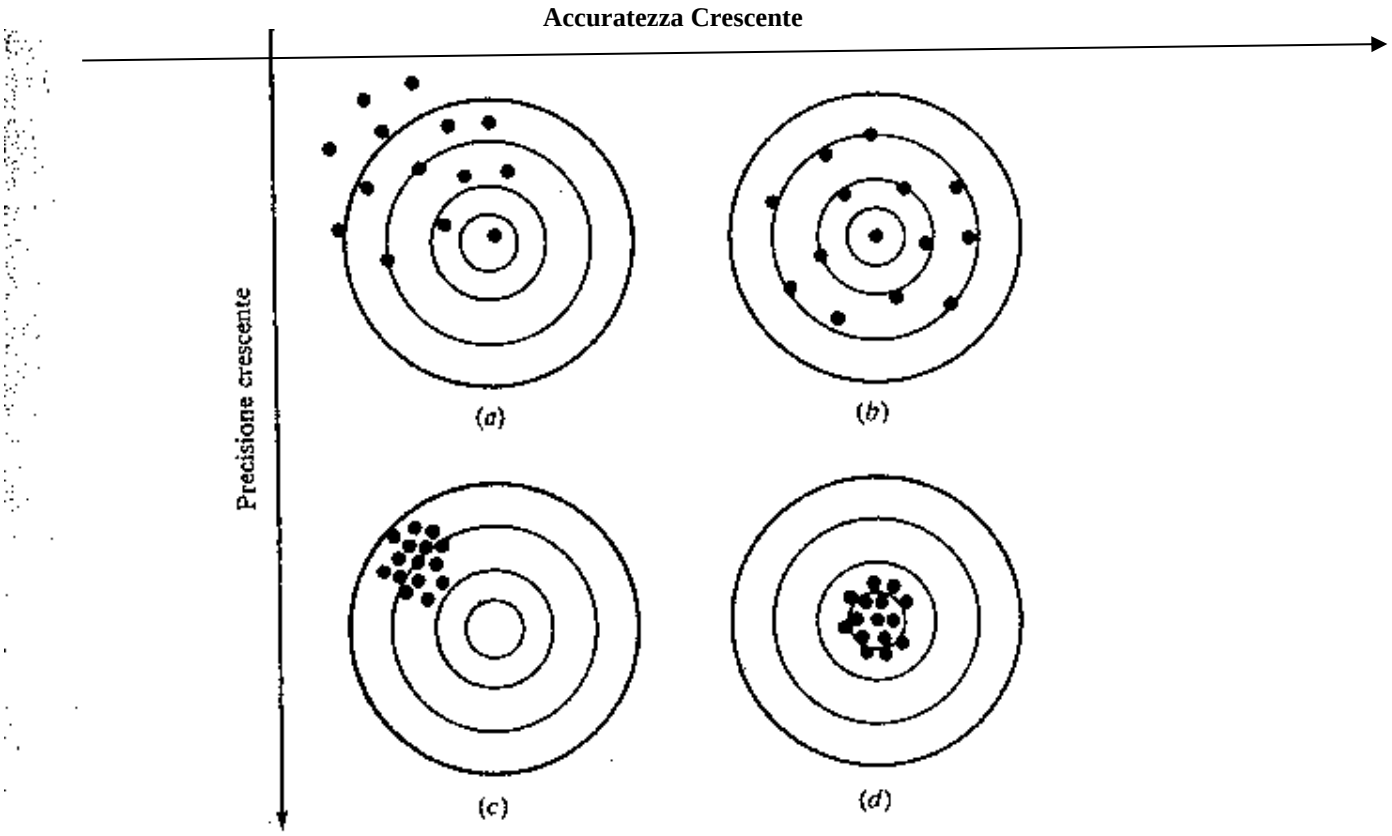


Figura 3.2 Esemplificazione dei concetti di accuratezza e precisione per mezzo di rose di colpi su bersagli: (*a*) inaccurata e imprecisa; (*b*) accurata ma imprecisa; (*c*) inaccurata ma precisa; **(d) accurata e precisa**

La **precisione** è riferita al numero di cifre significative con le quali è rappresentata una data quantità o alla dispersione dei valori letti su uno strumento durante la misura di una grandezza fisica.
(È un concetto legato alla ripetibilità dei risultati: in uno strumento più o meno preciso diverse misure effettuate sulla stessa grandezza forniscono **lo stesso valore con diverse cifre significative**)
Da un punto di vista statistico ad un insieme di misure precise della stessa grandezza è associata una **varianza** piccola.
L'accuratezza si riferisce alla maggiore o minore corrispondenza del valore misurato con quello reale. Da un punto di vista statistico il **valore medio** associato ad un insieme di misure accurate di una grandezza si **discosta poco dal valore vero**.

APPROSSIMAZIONE ED ERRORI

Errore assoluto. Errore relativo percentuale vero

Intimamente collegato al concetto di approssimazione (ovvero di rappresentazione approssimata di un numero o di una grandezza fisica) è il concetto di errore. In effetti qualunque approssimazione introduce un errore la cui entità dipende dal modo in cui l'approssimazione viene eseguita.
Va sottolineato immediatamente che di errore non si parla solo nel campo della matematica astratta. Qualunque misura effettuata sul mondo reale è viziata dall'imprecisione dello strumento, dalle condizioni ambientali, dal giudizio soggettivo dell'operatore, etc.: in una parola è affetta da errore.

In tutti i casi risulta:

valore vero = **approssimazione** + **errore assoluto vero**

ovvero

Errore assoluto vero E_t = valore vero - approssimazione **(1)**

Dove $\mathbf{E_t}$ indica il valore esatto dell'errore (il pedice t simboleggia la parola TRUE)

La presente definizione non tiene conto dell'ordine di grandezza della quantità in esame: un errore di 1 cm sulla lunghezza di un chiodo è molto più pesante dell'errore di 1 cm nella determinazione della lunghezza di un ponte.

Per tener conto dell'ordine di grandezza si può normalizzare l'errore rispetto al valore vero della grandezza e moltiplicarlo per 100 in modo da esprimerlo come termine percentuale:

Errore relativo percentuale vero = $\varepsilon = \dfrac{\text{Errore Assoluto Vero}}{\text{valore vero}} 100\,\%$ **(2)**

Esempio:

Misurando la lunghezza di un ponte e di un chiodo si ottengano rispettivamente: 99.99 m e 9 cm. Se i valori veri sono 100 m e 10 cm quali sono gli errori assoluto e relativo percentuale vero nei due casi?

Per il ponte:

$E_t = 10000 - 9999 = 1;$ $\varepsilon = 1/10000 * 100\ \% = 0.01\ \%$

Per il chiodo:

$E_t = 10 - 9 = 1;$ $\varepsilon = 1/10 * 100\ \% = 10\ \%$

Errore assoluto, errore relativo percentuale approssimato

Nella maggior parte delle situazioni il valore reale di una grandezza non è noto e, dunque, si può, al più, disporre di una semplice **stima dell'errore** ottenuta attraverso il confronto di due o più approssimazioni della quantità reale.

Errore approssimato:

E_a = approssimazione attuale - approssimazione precedente (3)

Errore relativo approssimato percentuale:

Errore rel. Appross. percentuale= $\varepsilon_a = \frac{\text{Errore Assoluto appr.}}{\text{appr. attuale}} 100\ \%$ (4)

Si può dimostrare che, se la soluzione ha *n* cifre significative, risulta:

$$\varepsilon_a < 0.5 * 10^{2-n}\ \%$$

SORGENTI DI ERRORE

Esistono varie sorgenti di errore per i metodi numerici:

1. Errori di arrotondamento

2. Errori di troncamento

3. Errori di formulazione (Per semplificazioni introdotte nel modello; 1Nel caso limite questi errori portano a problemi mal condizionati)

4. Errore nell'algoritmo (Problema ben condizionato, algoritmo instabile)

5. Errori dovuti a sviste

6. Incertezze nei dati

SORGENTI DI ERRORE

Errori di arrotondamento nelle operazioni elementari

Ci proponiamo di vedere come l'errore di rappresentazione sugli operandi si rifletta sul risultato per le varie operazioni aritmetiche.

1. Occorre premettere che, poiché alcuni metodi comportano l'esecuzione di un numero molto elevato di operazioni elementari, anche se il singolo errore di arrotondamento può essere piccolo, l'effetto cumulativo di una serie di operazioni può risultare significativo
2. L'effetto dell'arrotondamento può essere amplificato quando si eseguono manipolazioni algebriche utilizzando contemporaneamente numeri molto grandi e numeri molto piccoli.

Esempio:
Calcolare la differenza tra due grandi numeri:

```
 32 981 108. 1234 -
 32 981 107.9989 =
-----------------
          0.1245
```

Ripetiamo il calcolo aumentando il minuendo dello 0.001% così da simulare l'effetto di un errore di arrotondamento sulla 6 cifra significativa

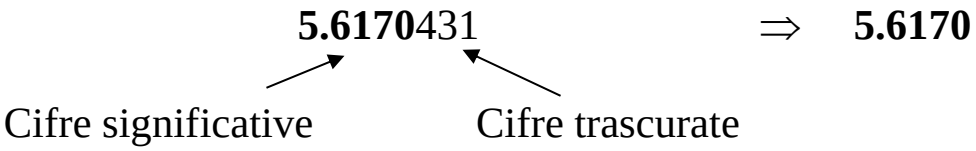
```
 32 981 437.9345 -
 32 981 107.9989 =
-----------------
        329.9356
```

Come si vede l'errore sul risultato è enorme.

Regole per l'arrotondamento

Arrotondamento di un numero

1) L'arrotondamento consiste nel conservare le cifre significative, mentre quelle non significative vengono scartate.



- L'ultima cifra conservata viene aumentata di 1 se la prima cifra scartata è maggiore di 5;
- se questa ultima cifra è minore di 5, l'ultima cifra conservata rimane invariata;
- Se la prima cifra scartata è 5, oppure 5 seguito da zeri, l'ultima cifra conservata deve essere aumentata di 1 solo se è dispari.

Esempi:

Numero esatto	Numero arrotondato	Cifre significative
5.6723	5.67	3
10.406	10.41	4
7.3500	7.4	2
88.21650	88.216	5
1.25001	1.3	2

Regole per l'arrotondamento

Addizione e sottrazione

Nell'addizione e sottrazione, se gli operandi sono noti con precisione diversa, l'arrotondamento va eseguito in modo che l'ultima cifra conservata nel risultato corrisponda all'ultima cifra più significativa dei numeri sommati o sottratti. Va tenuto presente che una cifra nella colonna dei centesimi è più significativa di una nella colonna dei millesimi, etc.

Esempi: (L'ultima cifra significativa è indicata in neretto)

Primo add.	Secondo add.	Terzo add	Ris.esatto	Ris. arrot.
2.**2**	-1.**7**68		0.432	0**.**4
4.68E-7	8.3E-4	-228E-6		
0.**0**0468E-4	8.**3**E-4	-2.**2**8E-4	6.02468E-4	6.0E-4

N.b. Si suppone che la precisione con la quale il calcolatore esegue i calcoli sia infinita e così la rappresentazione dei numeri nei registri della CPU. L'arrotondamento viene effettuato perché i dati sono noti con differente precisione.

Regole per l'arrotondamento

Moltiplicazione e divisione

Nella moltiplicazione e divisione, se gli operandi sono noti con precisione diversa, l'arrotondamento va eseguito in modo che il numero delle cifre significative nel risultato sia uguale al più piccolo numero di cifre significative degli operandi.

Esempi: (L'ultima cifra significativa è indicata in neretto)

Prodotto

Primo fattore	Secondo fattore	Ris.esatto	Ris. arrot.
0.0642	4.**8**	0.30816	0.3**1**

Divisione

Dividendo	Divisore	Ris.esatto	Ris. arrot.
94**5**	0.3185	2967.032967	29**7**0

Regole per l'arrotondamento

Combinazione di operazioni

Per la combinazione di operazioni aritmetiche, esistono due casi generali, il primo è la somma o sottrazione dei risultati di moltiplicazioni o divisioni:

(moltiplicazioni o divisioni)	±	(moltiplicazioni o divisioni)

L'altro caso riguarda la moltiplicazione o divisione dei risultati di somme o sottrazioni

(addizioni o sottrazioni)	* /	(addizioni o sottrazioni)

In entrambi i casi, bisogna arrotondare i risultati dei calcoli racchiusi tra parentesi prima di procedere con le altre operazioni, anziché arrotondare solo i risultati finali.

Esempi:

Calcolare

[15.2 * 2.**8**e-4]+[8.456E-4 / 0.17**7**]

Si eseguono per prima cosa le divisioni e moltiplicazioni:

15.2 * 2.**8**e-4 = 4.256e-3

8.456E-4 / 0.17**7**= 4.777401e-3

Poi si esegue l'arrotondamento:

4.256e-3 → 4.3e-3

4.777401e-3→ 4.78e-3

Infine si somma e si arrotonda il risultato:

4.**3**e-3+4.78e-3=9.08e-3=9.1e-3

SORGENTI DI ERRORE

Errori di troncamento

Sono dovuti al fatto che di ogni routine iterativa teoricamente infinita, l'algoritmo deve comunque limitarsi a eseguire un numero finito di cicli.
Se non intervengono fattori d'instabilità e quindi l'algoritmo è convergente questi errori sono normalmente i meno pericolosi.

Alcuni esempi classici:

Esempio 1

Il calcolo approssimato di un numero irrazionale, per il quale si deve troncare la serie infinita che lo approssima.

Per esempio, la base dei logaritmi naturali, (e= 2.71828182845905) si può ottenere attraverso il seguente sviluppo in serie:

$$e = 1 + 1/1! + 1/(2!) + \ldots + 1/(n!) + \ldots$$

In questo caso la serie converge rapidamente e, quindi, già con pochi termini si ottiene una rappresentazione accurata del numero. Per una rappresentazione con 5 cifre significative si vede dalla tabella che basta prendere i primi 7 termini della serie:

n	1	2	3	4	5	6	7	8
E	2.0000	2.5000	2.6667	2.7084	2.7167	2.7181	2.7183	2.7183

Esempio 2
Il calcolo approssimato di una funzione trascendente può essere ottenuto ugualmente attraverso una serie di potenze.

Per esempio la funzione tg(x) vale 1.0000 per $x=\pi/4$.
Sviluppando in serie di potenze e troncando al terzo termine, risulta:

$$tg(x)= x + x^3/3 + x^5/5 + \ldots = 1.0066586$$

Esempio 3

<u>Uso della serie di Taylor per la stima degli errori di troncamento</u>

Sviluppiamo in serie di Taylor attorno al punto x_i la funzione f .
Arrestando lo sviluppo al secondo ordine, risulta:

$$f(x_{i+1}) \cong f(x_i) + f'(x_i)\,(x_{i+1} - x_i) + f''(x_i)/2!\,(x_{i+1} - x_i)^2$$

Aggiungendo tutti i termini si arriva allo sviluppo completo in serie di Taylor:

$$f(x_{i+1}) = f(x_i) + f'(x_i)\,(x_{i+1} - x_i) + f''(x_i)/2!\,(x_{i+1} - x_i)^2 + \ldots + f^n(x_i)/n!\,(x_{i+1} - x_i)^n + R_n$$

Il termine resto R_n tiene conto di tutti i termini che vanno da n+1 all'infinito.
Per un teorema di analisi matematica è possibile trovare un valore ξ di x con $x_i \leq \xi \leq x_{i+1}$, tale per cui risulti:

$$R_n = f^{n+1}(\xi)/(n+1)!\,(x_{i+1} - x_i)^{n+1}$$

Per comodità di notazione poniamo $h = x_{i+1} - x_i$
Utilizzando questo risultato si vede che troncando l'espressione al termine n^{mo} dell'espansione si commette un errore dell'ordine di h^{n+1}, ovvero $\Theta(h^{n+1})$.

Allora, ritornando allo sviluppo del secondo ordine si può sostituire al circa uguale il segno di uguale a patto di introdurre il termine di errore:

$$f(x_{i+1}) = f(x_i) + f'(x_i)\,h + f''(x_i)/2!\,h^2 + \Theta(h^3).$$

Derivata numerica (Equazione alle differenze finite divise)

Utilizzando lo sviluppo in serie di Taylor è possibile pervenire a delle espressioni che permettono di ricavare attraverso semplici operazioni algebriche le derivate di ordine primo o superiore di una funzione.
Per la derivata di un dato ordine esistono varie espressioni ciascuna caratterizzata da un certo grado di accuratezza.

Approssimata della derivata prima tramite la prima differenza in avanti
Sviluppiamo in serie di Taylor attorno al punto x_i la funzione f ed arrestando lo sviluppo al secondo ordine, risulta:

$$f(x_{i+1}) = f(x_i) + f'(x_i)\, h + f''(x_i)/2!\, h^2 + \Theta(h^3) \qquad (1)$$

Da cui si ottiene: $f'(x_i) = \dfrac{f(x_{i+1}) - f(x_i)}{h} - f''(x_i)/2!\ \ h - \Theta(h^2)$

In altre parole, se si approssima la derivata col solo primo addendo si commette un errore dell'ordine di h.

Appross. della derivata prima tramite la prima differenza all'indietro
Alternativamente si può porre:

$$f(x_{i-1}) = f(x_i) - f'(x_i)\, h + f''(x_i)/2!\, h^2 + \Theta(h^3) \qquad (2)$$

Da cui si ottiene: $f'(x_i) = \dfrac{f(x_i) - f(x_{i-1})}{h} + f''(x_i)/2!\ \ h + \Theta(h^2)$

A questa espressione è associato ancora un errore di ordine h

Approssimazione della derivata prima tramite le differenze centrali

Sottraendo la (2) dalla (1) si perviene all'espressione

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h} + \Theta(h^2)$$

cui è associato un errore di ordine h^2 (migliore approssimazione).

Errore numerico globale

L'errore numerico globale è la somma dell'errore di arrotondamento e dell'errore di troncamento.

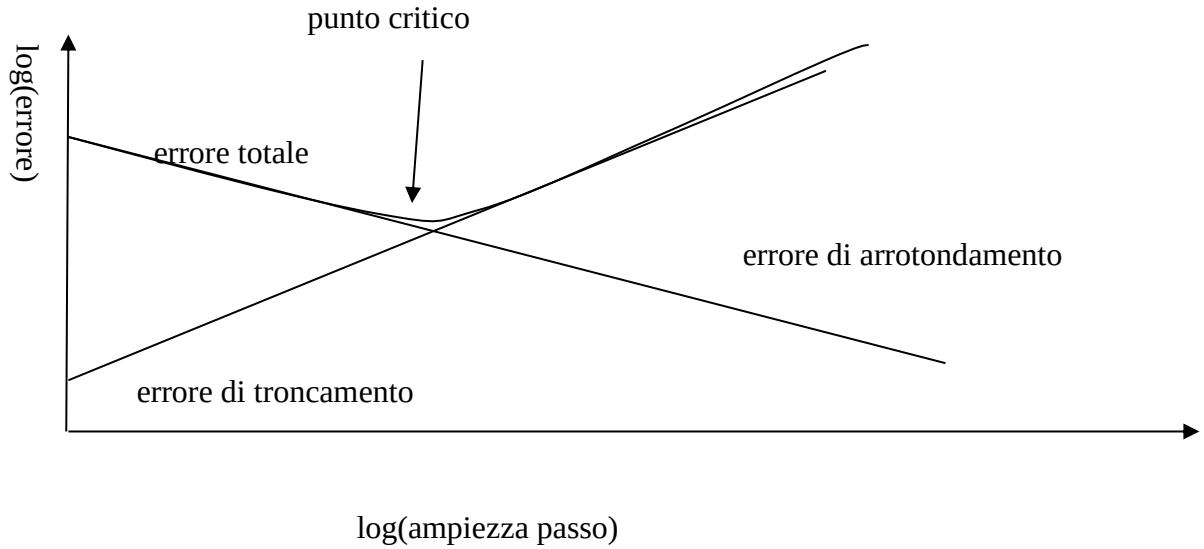
Per diminuire l'errore di arrotondamento è necessario aumentare il numero di cifre significative nella rappresentazione dei numeri sul calcolatore.

Inoltre gli errori di arrotondamento crescono all'aumentare del numero di operazioni richieste da un'elaborazione

D'altra parte l'errore di troncamento decresce al crescere del numero di operazioni (ad esempio diminuisce se in uno sviluppo in serie aumenta il numero dei termini della serie utilizzati nel calcolo, oppure se nel risolvere un'equazione differenziale si diminuisce il passo di integrazione in modo da migliorare la stima della derivata)

In conclusione si perviene al seguente dilemma: se si vuole diminuire un errore bisogna necessariamente accettare che l'altro cresca.

Esiste un punto critico nel quale gli arrotondamenti cominciano ad annullare i benefici della diminuzione dell'errore di troncamento.
Nei casi comuni l'errore di troncamento prevale sull'errore di arrotondamento



SORGENTI DI ERRORE

Algoritmo instabile

Dove si manifesta?
Si verifica nei programmi matematici che utilizzano algoritmi iterativi e conduce a dei risultati inattendibili.

Quando?
I fenomeni di instabilità si presentano quando un errore dovuto ad una delle cause precedenti (p.e. un errore di arrotondamento o di troncamento) si esalta nel meccanismo stesso dell'algoritmo di calcolo, portando più o meno rapidamente ad una situazione di pericolosa di instabilità dei risultati.

DEFINIZIONE
Un algoritmo si dice instabile se un errore relativo si accumula e cresce al crescere del numero dei cicli dell'algoritmo stesso ed è stabile in caso contrario.

N.B. L'instabilità può presentarsi solo per particolari valori iniziali delle variabili. Tra i compiti dell'analisi numerica vi è quello di stabilire dei criteri per individuare "a priori" il range dei valori delle variabili all'interno dei quali l'algoritmo è stabile.

Algoritmi iterativi

Supponiamo di voler trovare lo zero di una funzione reale f(x) a valori reali, $\Re \rightarrow \Re$
Tra i vari metodi vi è quello **delle sostituzioni successive** che consiste nella ricerca del cosiddetto **punto fisso*** α di una trasformazione g(x) definita opportunamente.

Ad esempio si può porre $g(x) = x + k*f(x)$, $k \in \Re$,

di modo che nel punto fisso $x=\alpha$ risulti $g(\alpha)=\alpha$ (e $f(\alpha)=0$).

Per la ricerca del punto fisso si può utilizzare la tecnica iterativa:

$x^{k+1}=g(x^k), k=0,1,2\ldots$
$x^0=x_0$, dove $x_0 \in \Re$ è una stima iniziale della soluzione.

In generale si ha convergenza se risulta $|g(x)'| < 1$ (ovvero se g(x) è una **contrazione**).

La velocità di convergenza verso la soluzione aumenta al diminuire di $|g(\alpha)'|$.

*Si dice **punto fisso** di un'applicazione g(x) quel valore di x per il quale risulta: x=g(x)

Esempio

L'equazione $f(x)=x^2-c=0$, con $c>0$

può essere messa nella forma $x=g(x)$ nei due modi seguenti:

$$x=\frac{1}{2}(x+\frac{c}{x}); \quad \text{con} \quad g'(\alpha)=\frac{1}{2}(1-\frac{c}{x^2})\Big|_{x=\sqrt{c}}=0$$

$$x=\frac{c}{x}; \quad \text{con} \quad g'(\alpha)=-\frac{c}{x^2}\Big|_{x=\sqrt{c}}=-1$$

Nel primo caso si ha convergenza, mentre nel secondo la successione oscilla alternativamente tra x_0 e c/x_0.

Ad esempio per $c=2$, $x_0=1.5$ lavorando in doppia precisione si ottiene la seguente tabella:

k	x^k	$\sqrt{2}$
0	1.5	
1	1.416666666666667	1.414213562373095
2	1.414215686274510	1.414213562373095
3	1.414213562374690	1.414213562373095
4	1.414213562373095	1.414213562373095

La convergenza è di tipo quadratico, cioè se x^k ha *t* cifre significative esatte x^{k+1} ne ha almeno $2t-1$.

Convergenza

Quando si utilizza un metodo iterativo occorre porsi il problema della convergenza della successione delle soluzioni stimate x_k alla soluzione vera x*:

$$\lim_{k\to\infty} x_k = x^* \qquad (1)$$

E, in caso affermativo, quello della rapidità di convergenza.
Per quantificare questo parametro, si introduca l'errore al passo k come:

$$e_k = x_k - x^* \qquad (2)$$

Se esiste un intero p>0 ed una costante C≠0 tali che:

$$\lim_{k\to\infty} \frac{|e_k|}{|e_{k-1}|^p} = C \qquad (3)$$

allora p si chiama *ordine di convergenza* della successione, e C si chiama *costante asintotica di errore*.
Intuitivamente, l'ordine di convergenza stabilisce quanto rapidamente l'errore diminuisce tra un'iterazione e l'altra.

OSSERVAZIONI

Anche la costante asintotica influenza la rapidità di convergenza, ossia il numero di iterazioni necessarie ad ottenere un errore al di sotto di una prefissata soglia. Pertanto un metodo a p=1 potrebbe convergere più rapidamente di un metodo a p=2 se la sua costante asintotica C è minore.

La convergenza della successione x_k a x* va intesa al limite come da (1).

Non essendo nota la x*, non si può arrestare la successione quando la (2) scende sotto una tolleranza prefissata. Il problema del criterio di arresto di un algoritmo iterativo non è banale e verrà affrontato operativamente nel seguito.

SORGENTI DI ERRORE

Semplificazioni introdotte nel modello

Sono gli errori dovuti al fatto che, ad esempio il modello è supposto di tipo lineare, oppure si suppongono "trascurabili" alcune grandezze fisiche. In questo caso si parla di **adeguatezza** del modello

Nel caso limite, le semplificazioni del modello conducono a

Problemi mal condizionati

In alcuni casi è il modello matematico stesso associato ad un problema che può dare origine a fenomeni di instabilità e non l'algoritmo che lo risolve.

Questo avviene quando il problema numerico è tale che anche piccoli errori, dovuti alle approssimazioni nella rappresentazione dei dati che qualsiasi algoritmo numerico introduce, comportano elevate variazioni nei risultati finali.
(Definizione di problema mal condizionato secondo **Hadamard**).

N.B. Se il problema è mal condizionato, non esiste algoritmo matematico che lo possa migliorare. Vuol dire che è il problema stesso che è stato mal concepito oppure che il suo modello matematico è sbagliato perché sono state adoperate semplificazioni azzardate.

Esempio:

È ben noto che condizione necessaria e sufficiente perché un sistema di equazioni lineari sia **improprio** (**impossibile** oppure **indeterminato**) che il determinante caratteristico sia nullo (ovvero quando una riga o una colonna della matrice associata al sistema è combinazione lineare delle altre).

Per un sistema di due equazioni lineari

$$\begin{cases} ax + by = c \\ a'x + b'y = c' \end{cases}$$

ciò si verifica quando risulta:

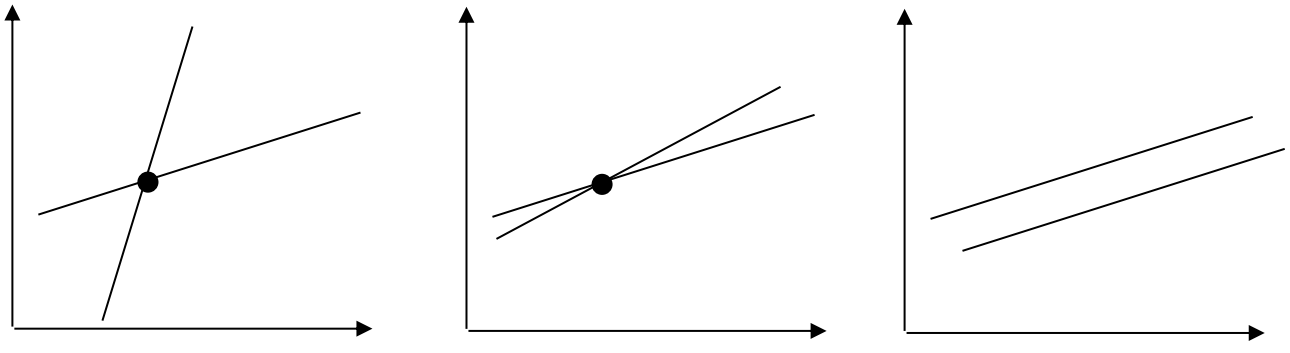
$$\Delta = ab' - a'b = 0$$

Da un punto di vista grafico ciascuna equazione rappresenta una retta e la soluzione è data dal punto di intersezione delle rette.

La condizione di nullità del determinante può essere posta nella forma alternativa che stabilisce l'uguaglianza tra i coefficienti angolari delle due rette:

$$-a/b = -a'/b'$$

Quando, pur **non** essendo **esattamente verificata** la condizione di **nullità del determinante**, la differenza è dello stesso ordine di grandezza della precisione con cui si esegue il calcolo, oppure delle approssimazioni dovute agli arrotondamenti, la soluzione perde di attendibilità perché il minimo errore può modificare notevolmente il risultato.



Ben condizionato | Mal condizionato | Impossibile

ULTERIORI SORGENTI DI ERRORE

Errori dovuti a sviste o a banali errori di calcolo

Dipendono unicamente dall'attenzione dell'uomo addetto al calcolo ovvero all'implementazione dell'algoritmo sul calcolatore.

Per evitare gli errori più banali è consigliabile introdurre nel codice dei blocchi di controllo sulla congruenza dei dati (segno, range di valori atteso), ovvero sulle dimensioni di un vettore oppure di una matrice.

Un altro accorgimento è quello di obbligare l'utente, dopo aver scritto una serie di numeri, a rileggere quanto scritto confermandone la correttezza oppure correggendoli.

Errori di precisione dei dati

Sono dovuti all'inevitabile margine di precisione delle misure o dei dati di partenza.

Questo tipo di errore deve essere ben presente all'utente in modo da evitare di introdurre dati e di stampare risultati con un numero di cifre superiori a quelle effettivamente utili. Valgono per questo errore gli stessi problemi e le medesime considerazioni fatte nel caso dell'errore di arrotondamento.

SOLUZ. DI SISTEMI DI EQ. ALGEBRICHE NON LINEARI

La soluzione dei sistemi di equazioni algebriche non lineari riveste un ruolo fondamentale nella modellistica numerica.
Ad esempio, i sistemi a parametri concentrati non lineari in condizioni stazionarie sono governati da equazioni algebriche non lineari.

Più in generale, la soluzione numerica di equazioni differenziali ed integrali non lineari si riduce alla soluzione di sistemi di equazioni algebriche non lineari.

Formalizziamo il problema che intendiamo considerare:

Data una funzione **f** = **f** (**x**) definita in $D \supseteq \Re^n$ e avente codominio $C \supseteq \Re^n$,

$$\mathbf{f}(\mathbf{x}) \equiv |\, f_1(x_1, x_2, ..., x_n),\ \ldots,\ f_n(x_1, x_2, ..., x_n)\,|^{T} \in C \supseteq \Re^n \quad (1.1)$$

bisogna cercare un vettore $\mathbf{u} = |\, u_1, u_2, ..., u_n \,|^{T} \in D \supseteq \Re^n$, tale che

$$\mathbf{f(u)} = 0 \Leftrightarrow \begin{cases} f_1(u_1, u_2, ..., u_n) = 0 \\ f_2(u_1, u_2, ..., u_n) = 0 \\ \\ f_n(u_1, u_2, ..., u_n) = 0 \end{cases} \quad (2.1)$$

Il problema (2.1) è un sistema di *n* equazioni algebriche non lineari in *n* incognite. Le soluzioni **u** sono dette anche "**zeri**" o "**radici**" della funzione **f**.

Un'equazione del tipo (2.1) può avere una sola soluzione, più di una soluzione o addirittura nessuna soluzione. Si intuisce che la soluzione dell'equazione (2.1) esiste ed è unica se:

- il codominio di **f** contiene lo zero;
- la funzione **f** definisce un omeomorfismo del dominio *D* nel codominio *C*

RADICI DI EQUAZIONI: CASO SCALARE

Definizione:
Le **radici** di un'equazione sono quei valori di x che rendono la funzione f(x)=0.
Le **radici** vengono anche chiamate **zeri**.
Per definizione una funzione y=f(x) viene detta algebrica se può essere espressa nella forma

$$y = f(x) = \sum_{i=0}^{n} g_i x^i = 0 \qquad (2.2a)$$

Una funzione è detta **trascendente** se non è algebrica.

$$y = f(x) = 0 \qquad (2.2)$$

Le radici di un'equazione possono essere sia **reali** che complesse.
Per le equazioni algebriche a coefficienti reali le radici complesse si presentano sempre in **coppie coniugate**.

Per le applicazioni ingegneristiche si richiede nella maggior parte dei casi di determinare:

1. Le radici reali delle equazioni algebriche e trascendenti.

2. Tutte le radici (reali e complesse) delle equazioni polinomiali.

RADICI DELLE EQUAZIONI E PROBLEMI DI INGEGNERIA

I problemi che richiedono la determinazione delle radici di un'equazione sono tipici della pratica ingegneristica. Esistono una serie di principi fondamentali che vengono utilizzati per derivare i modelli matematici associati ai problemi reali.

Principio fondamentale	Variabile dipendente	Variabile indipendente	Parametri
Conservazione del calore	Temperatura	Tempo e posizione	Proprietà termiche materiale, geometria
Conservazione della massa	Quantità di massa	Tempo e posizione	Struttura materiale, geometria
Bilancio di forze	Intensità e direzione forze	Tempo e posizione	Resistenza materiale, conf. del sistema
Conservazione dell'energia	Variazione en. cinetica e potenziale	Tempo e posizione	Proprietà termiche, massa, geometria
Leggi del moto di Newton	Accelerazione, velocità, posizione	Tempo e posizione	Massa, geometria, parametri dissipativi
Leggi di Kirchhoff	Correnti e tensioni	Tempo	Proprietà elettriche

Le equazioni correlano l'andamento delle variabili dipendenti a quello delle variabili indipendenti.
Le **variabili indipendenti** tengono conto delle sorgenti esterne e sono imposte.
Le **variabili dipendenti** riflettono lo stato ovvero l'evoluzione del sistema.
I **parametri** rappresentano le proprietà o la composizione del sistema.

Metodi per la ricerca delle radici

GRAFICO

Si disegna il grafico della funzione e si identificano approssimativamente i punti di intersezione con l'asse delle x.

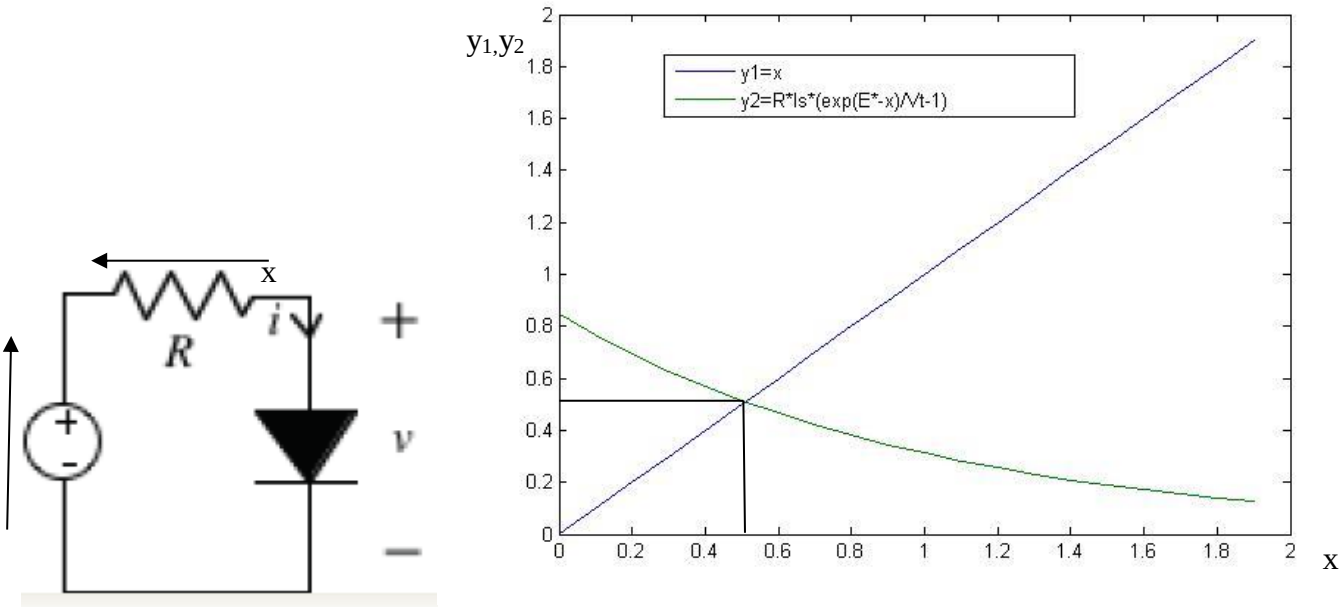
Vantaggi:

- Applicabile ad una classe ampia di funzioni

Svantaggi

- Ricerca della soluzione laboriosa
- Soluzione poco accurata e poco precisa
- Non adeguato per uno studio parametrico

Esempio 2.1:



(a) (b)

Fig.2.1: a) Circuito; b) Ricerca del punto di lavoro per via grafica

Problema Assumendo R=100 Ω, e*(t)=10 V e che il diodo abbia una caratteristica del tipo $i_d = q(v_d) = I_s\left(e^{vd/Vt} - 1\right)$ dove I_s è la corrente di saturazione (assunta pari a $10*10^{-9}$ A) e V_t è l'equivalente in volt della temperatura (dell'ordine di 26mV), si determini la tensione ai capi del resistore (il valore vero della tensione alla 4 cifra significativa è x=0.5092 V).

Lo zero della funzione $f(x)=x-R*I_s*(e^{(E*-x)/Vt}-1)$ non è ricavabile analiticamente.

È possibile ricavare in modo approssimato la radice come intersezione delle due curve $y_1=x$ e $y_2= R*I_s*(e^{(E*-x)/Vt}-1)$

ANALITICO

Vantaggi:

- Soluzione esatta (Estrema accuratezza e precisione).
- La soluzione, calcolata in forma simbolica, si presta ad uno studio parametrico

Svantaggi:

- Applicabile ad una classe ristretta di funzioni

Esempio:

$f(x)=ax^2+bx+c=0$ (2.3) → $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ *(2.4)*

Per a=1, b=5,c=6 → x1=-2; x2=-3

NUMERICO

Vantaggi:

- Applicabile ad una vasta classe di funzioni
- Ricerca della soluzione semplice
- Soluzione accurata e precisa
- Possibilità di uno studio parametrico

Classificazione dei metodi numerici per la ricerca delle radici

Metodi chiusi

Presentano una coppia di valori di tentativo che delimitano il dominio in cui le radici vengono ricercate. L'ampiezza del campo di ricerca viene via via ristretto fino a determinare la soluzione. Descriveremo il Metodo **di bisezione** e quello **della falsa posizione**

Metodi aperti

In questi metodi il dominio in cui vengono ricercate le radici coincide col dominio di definizione del problema. Descriveremo i metodi **delle sostituzioni successive**, **di Newton-Raphson** e **delle secanti.**

Esempio 2.2

Si consideri un circuito composto da un generatore ideale di tensione *E*, un resistore lineare di resistenza *R* e un resistore non lineare *N* controllato in tensione (Figura 2.2)
L'equazione caratteristica di *N* può essere così rappresentata

$$i = g\,(v), \qquad (2.5)$$

dove *i* e *v* sono rispettivamente l'intensità di corrente e la tensione del bipolo (con i versi di riferimento concordi con la convenzione dell'utilizzatore).
Il resistore *N* potrebbe essere, ad esempio, un diodo a giunzione *pn* o un diodo tunnel, Figura 2.3. Per il diodo a giunzione *pn* una buona approssimazione della funzione *g* è data da

$$g\,(v) = I_S\,(e^{v/V_T} - 1) \qquad (2.6)$$

per tensioni molto più grandi della tensione di breakdown; I_S è l'intensità di corrente di saturazione inversa e V_T è la tensione termica. Valori tipici di I_S e V_T sono, rispettivamente, 10 nA e 26 mV.
Per il diodo tunnel una buona approssimazione della funzione *g* è data da

$$g\,(v) = a_1 v + a_2 v^2 + a_3 v^3 \qquad (2.7)$$

per $v \geq 0$, dove $a_1 = 6.0\ \Omega^{-1}$, $a_2 = -4.5\ (\mathrm{V}\,\Omega)^{-1}$ e $a_3 = 1.0\ (\mathrm{V}^2\,\Omega)^{-1}$

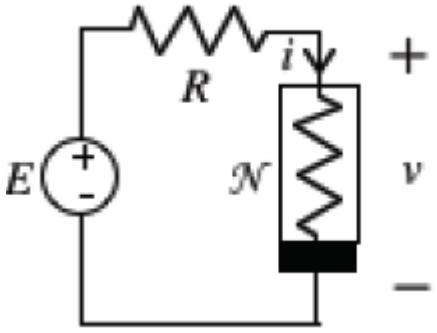


Figura 2.2 Un circuito non lineare

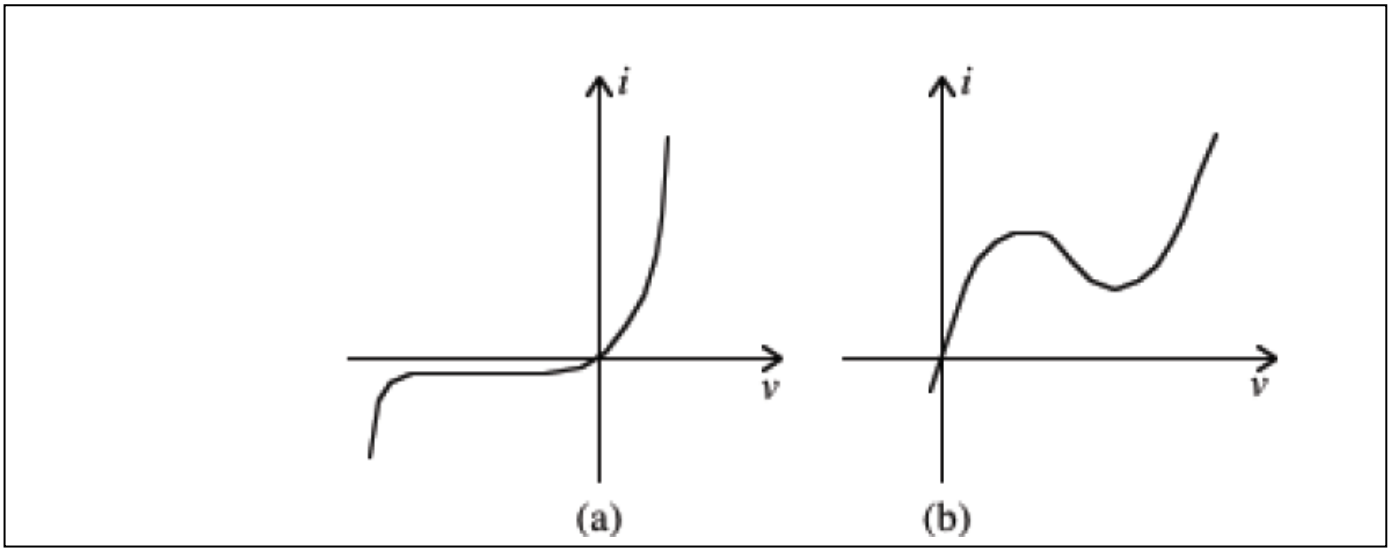


Figura 2.3 Curve caratteristiche del diodo a giunzione pn (a) e diodo tunnel (b)

L'equazione caratteristica del bipolo composto dalla serie generatore di tensione – resistore è

$$i = \frac{E - v}{R} \qquad (2.8)$$

Combinando le equazioni (2.5) e (2.8) si ottiene l'equazione per la tensione del resistore non lineare

$$g(v) + \frac{v}{R} - \frac{E}{R} = 0 \qquad (2.9)$$

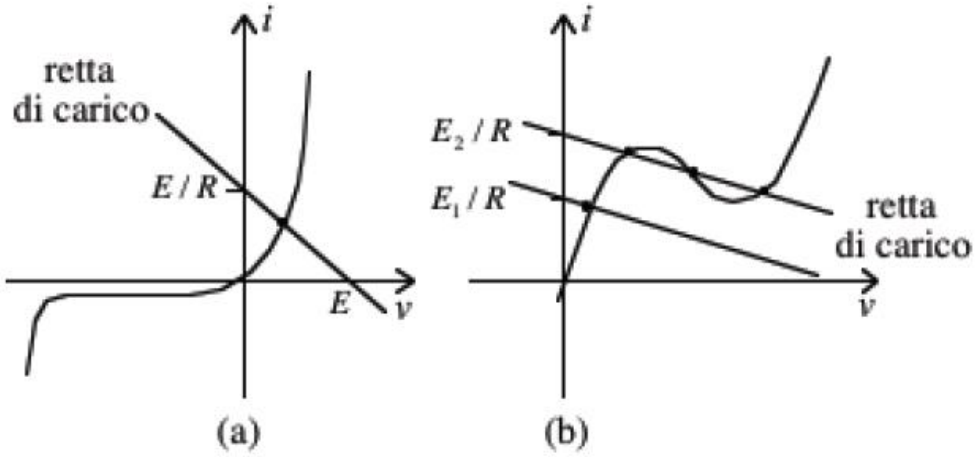


Figura 2.4 La retta di carico definita dalla (2.8) interseca sempre in un solo punto la curva caratteristica del diodo a giunzione pn (a); la retta di carico può intersecare in tre punti distinti la curva caratteristica del diodo tunnel (b).

Ecco un esempio d'equazione algebrica non lineare scalare. La soluzione V dipende dai parametri R ed E, $V = V(R,E)$.

Se il resistore non lineare è un diodo a giunzione *p-n* l'equazione (2.9) ha una ed una sola soluzione per qualsiasi valore dei parametri; invece, nel caso di un diodo tunnel l'equazione (2.9) può avere una sola soluzione o tre soluzioni distinte, a seconda dei valori dei parametri. Ciò può essere facilmente verificato attraverso il metodo grafico illustrato in Figura 2.4.

Assumiamo, per ora, che l'equazione (2.2) abbia una e una sola soluzione nell'intervallo [*a*, *b*] di $\Re$. A partire da un valore iniziale x_0 opportunamente scelto, un metodo iterativo per risolvere la (2.2) genera, secondo una legge opportuna

$$x_k = G\,(x_{k-1}\,)\,, \quad (2.10)$$

una successione di valori

$$x_1, x_2\,,\ldots,\, x_k\,,\ldots \quad (2.11)$$

che, in generale, converge alla soluzione x_T per $k \to \infty$. Non essendo possibile iterare all'infinito la (2.10) attraverso una macchina di calcolo, occorre definire un criterio d'arresto per stabilire a quale iterazione *N* si ha una stima accettabile della soluzione,

$$x_T \cong x_N\,. \quad (2.12)$$

Il problema del criterio d'arresto sarà trattato ampiamente più avanti per ognuno dei metodi iterativi presentati.
Esistono diversi metodi per risolvere un'equazione algebrica non lineare. In queste Lezioni illustreremo le caratteristiche principali dei seguenti metodi:

metodi chiusi
- il metodo di bisezione;
- il metodo della falsa posizione;

metodi aperti
- il metodo di Picard (o delle sostituzioni successive);
- il metodo di Newton-Raphson
- il metodo delle secanti

I metodi aperti si estendono senza alcuna difficoltà alle equazioni vettoriali, mentre l'estensione del metodo della bisezione e di quello della falsa posizione presenta non poche difficoltà.

METODI CHIUSI

Sono così chiamati perché basati sull'osservazione che *tipicamente* una funzione cambia segno nell'intorno di una radice.
L'idea è quella di cercare la radice all'interno del dominio $]x_l, x_u[$, dove x_l e x_u sono due valori iniziali tali che $f(x_l)$ e $f(x_u)$ assumono segno opposto. L'algoritmo di ricerca, di tipo iterativo, mira a restringere l'ampiezza del dominio. I metodi proposti differiscono sulla strategia adottata per restringere il dominio.

Possibili comportamenti delle radici di f(x) in $]x_l,x_u[$

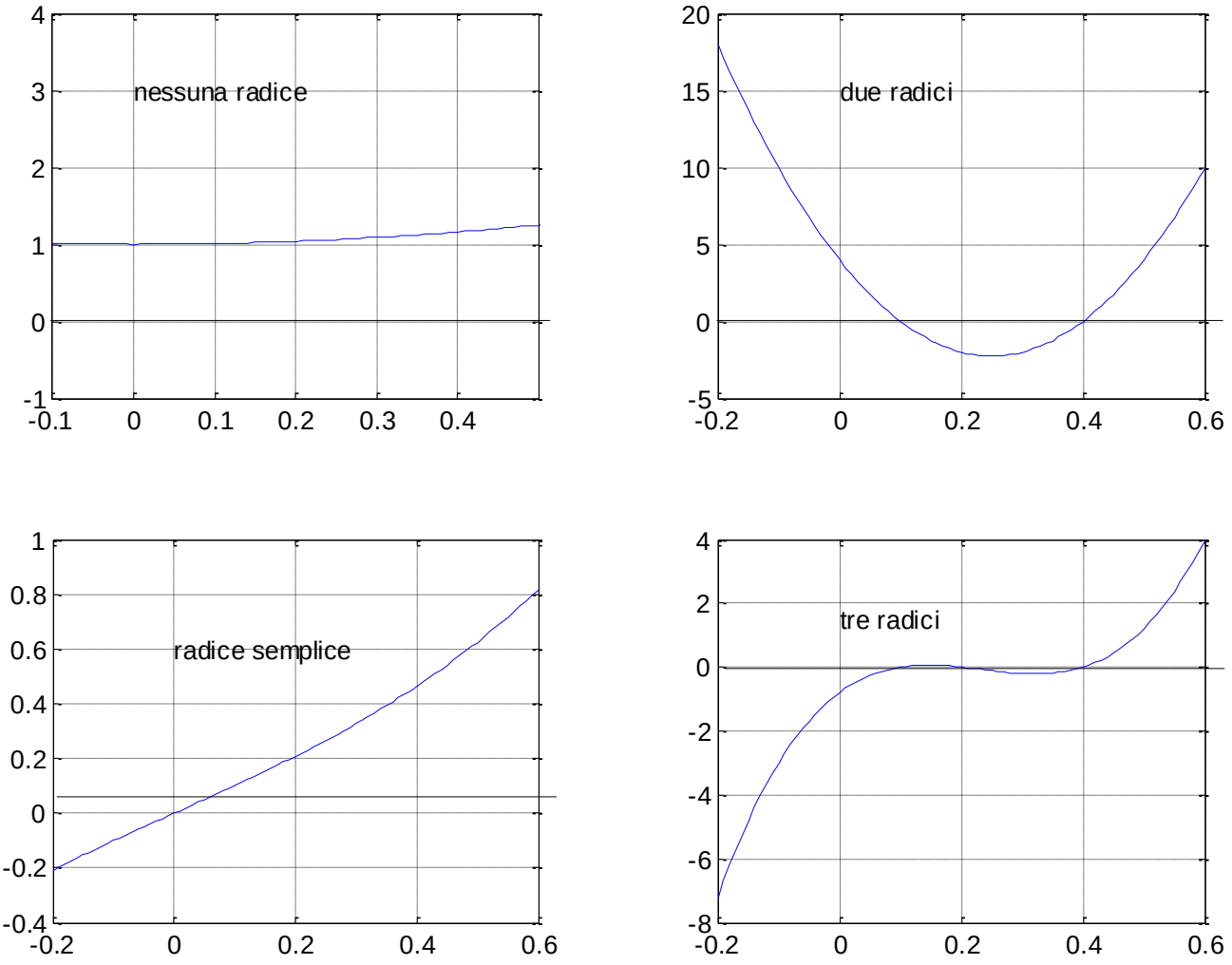


Fig.2.5a $f(x_l)$ e $f(x_u)$ hanno segno concorde nelle figure in alto (nessuna o numero pari di radici), discorde in quelle in basso (numero dispari di radici)

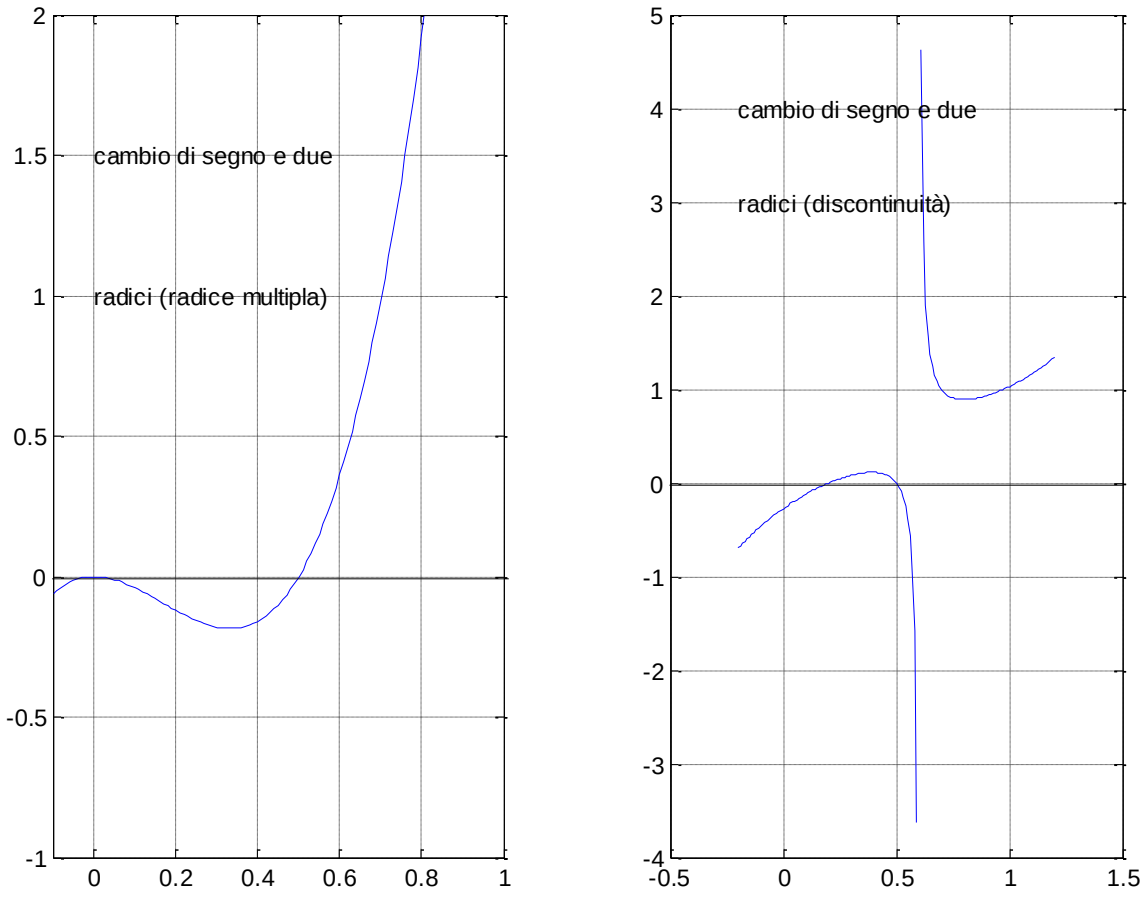


Fig.2.5b Eccezioni : f(x_l) e f(x_u) hanno segno discorde e un numero pari di radici. In questi casi la determinazione delle radici richiede delle strategie particolari.

Come si vede l'uso della **grafica al calcolatore** permette di localizzare in maniera sistematica le radici di una funzione.

Metodo di bisezione

Si è notato che se f(x) è reale e continua, cambia di segno in un intorno di una radice singola, cioè:

se f(x) è reale e continua, $x_l < x_u$ e $f(x_l)*f(x_u)<0$, allora esiste almeno una radice reale x_r, con $x_l < x_r < x_u$

I *metodi incrementali di ricerca*, concentrano la ricerca in un intervallo dove la funzione cambia di segno.

La posizione del *punto di transizione* viene identificata procedendo per iterazioni successive nelle quali l'intervallo viene diviso in sotto-intervalli sempre più ristretti nei quali viene verificata la presenza del punto di transizione.

Nel **metodo di bisezione**, l'intervallo viene sempre diviso a metà.

L'algoritmo del metodo è riportato di seguito.

Algoritmo

1. Si sceglie una stima inferiore, x_l,ed una stima superiore, x_u, della radice, tali per cui $f(x_l)*f(x_u)<0$
2. Si assume che in prima approssimazione la radice si trovi esattamente al centro dell' intervallo $x_r=(x_l+x_u)/2$
3. Si calcola il valore della funzione in x_r e lo si confronta con un indice di tolleranza ε_{tol}
    a) se $|f(x_r)|\leq\varepsilon_{tol}$, si assume come radice x_r e il programma termina;
    b) se $f(\mathbf{x_l})*f(\mathbf{x_r})<0$, la radice si trova in $]x_l,x_r[$; $x_u=x_r$
    c) se $f(\mathbf{x_l})*f(\mathbf{x_r})>0$, la radice si trova in $]x_r,x_u[$; $x_l=x_r$
4. Si calcola una nuova stima della radice ponendo $x'_r=(x_l+x_u)/2$.
5. Si controlla se l'accuratezza della stima è soddisfacente, confrontando l'ampiezza dell'intervallo $\Delta x=|(x_u-x_l)|$ con un indice preassegnato $\varepsilon_{amp,des}$; equivalentemente si può verificare che risulti l'err. appr. perc. rel. $\varepsilon_a=|(x'_r-x_r)/x'_r)|*100\% <\varepsilon_{a,des}$
    a) se $\Delta x\leq\varepsilon_{amp,des}$ ($\varepsilon_a\leq\varepsilon_{a,des}$), o il numero di iterazioni *nit* > *nitmax* si assume come radice $\mathbf{x_r}$ e il programma termina.
    b) se $\Delta x\geq\varepsilon_{amp,des}$ ($\varepsilon_a\geq\varepsilon_{a,des}$) si pone $x_r=x'_r$ $nit=nit+1$ si riprende dal passo 3

Pro

- Garanzia di convergenza scegliendo opportunamente l'intervallo di partenza
- Necessità di calcolare solo il valore della funzione in determinati punti.

Contro

- Necessità di individuare l'intervallo di partenza
- Applicabilità solo a casi monodimensionali.

Osservazioni

- La convergenza ad una soluzione è garantita solo se sono rispettate le condizioni indicate in precedenza
  (continuità della funzione e scelta corretta dell'intervallo iniziale $]x_l, x_r[$)
- Spesso la scelta dell'intervallo $]x_l, x_r[$ può essere fatta sulla base di considerazioni fisiche relative al problema sotto indagine.
- Il test di arresto è triplice:
  1. Si verifica se risulta $|f(\mathbf{x_r})| \leq \varepsilon_{tol}$;
  2. si verifica se risulta $\Delta x = |(x_u - x_l)| \leq \varepsilon_{amp,des}$
     (o equivalentemente che $\varepsilon_a \leq \varepsilon_{a,des}$)
  3. si controlla che non si sia superato il massimo numero di iterazioni *nitmax*

A parte il test 3) di banale comprensione, il test 1) permette di stabilire una accuratezza desiderata in termini di valore della funzione ottenuto in corrispondenza della soluzione numerica, mentre il test 2) nelle due forme ci permette di stabilire l'accuratezza della soluzione, dato che, nelle ipotesi prima citate, lo zero della funzione risulta compreso tra x_u e x_l, e quindi la quantità $\Delta x = |(x_u - x_l)|$ è una stima per eccesso dell'errore.

Rappresentazione grafica dell'algoritmo

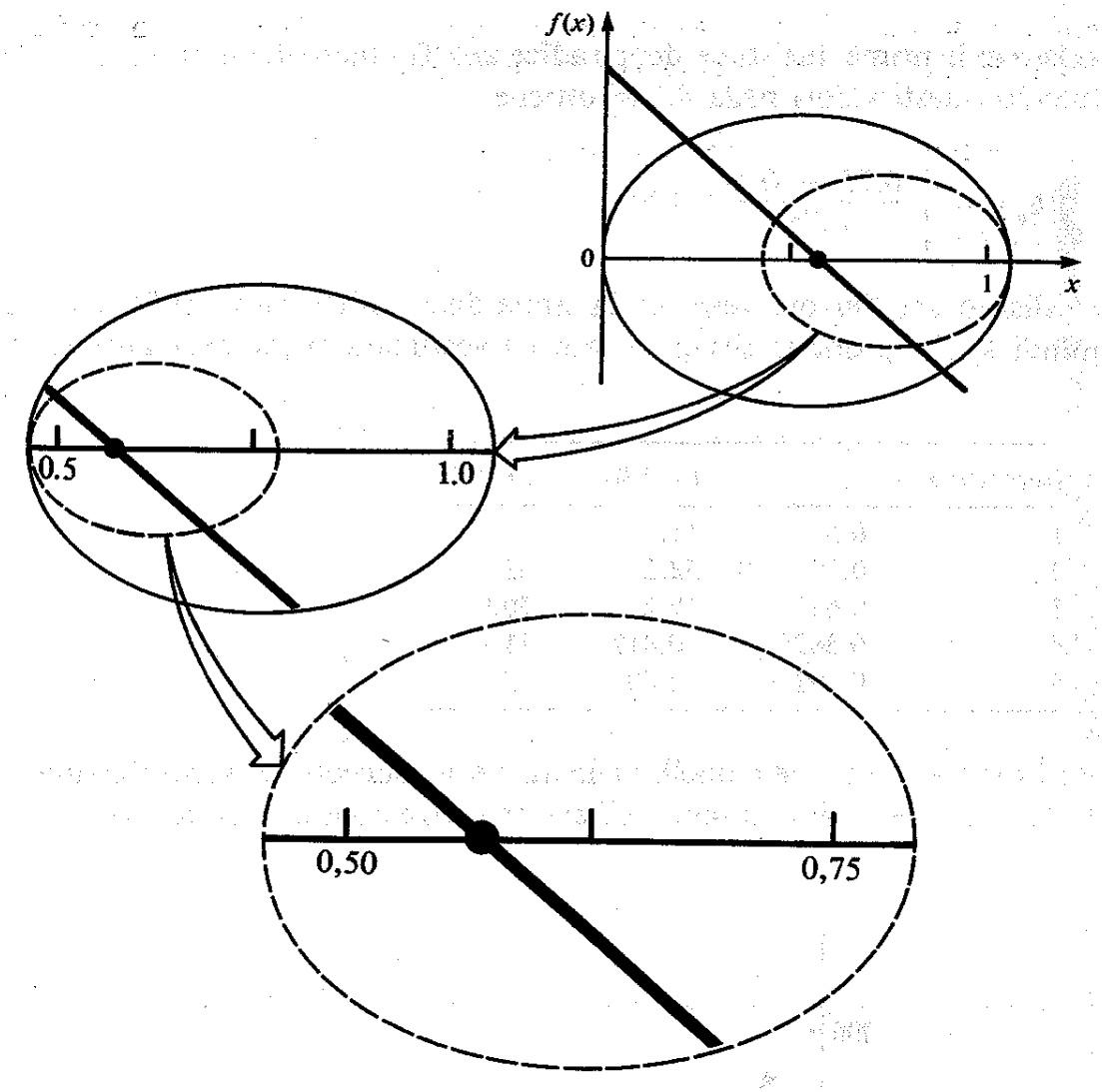


Fig.2.6 Rappresentazione grafica del metodo di bisezione. Le prime tre iterazioni dell'esempio

Esempio 2.3: usare il metodo di bisezione per determinare la radice di $f(x)=e^{-x}-x$ (valore esatto 0.56714329)

Dal grafico notiamo che la radice si trova tra $x_l=0$ e $x_u=1$ → $x_r=(0+1)/2=0.5$

It	xl	f(xl)	xu	f(xu)	xr	f(xr)	Err rel per
1	0	1	1	-0.6321	0.5	0.10653	11.8%
2	0.5	0.10653	1	-0.6321	0.75	-0.2776	32.3%
3	0.5	0.10653	0.75	-0.2776	0.625	-0.0897	10.2%
4	0.5	0.10653	0.625	-0.0897	0.5625	0.0073	0.819%

Errore relativo percentuale stimato e massimo errore assoluto

$|\in_a|=|(x_{r,att}-x_{r,pre})/x_{r,att}|\ 100\%$ $\quad -(x_u-x_l)/2 \le E_a \le (x_u-x_l)/2$

It	xr	Err rel per. vero	Err rel perc. stim.	Err.ass. Max
1	0.5	11.8%		±0.5
2	0.75	32.3%	33.3%	±0.25
3	0.625	10.2%	20.0%	±0.125
4	0.5625	0.819%	11.1%	±0.0625
5	0.59375	4.69%	5.3%	±0.03125

Si noti che nel metodo di bisezione l'errore relativo percentuale stimato è sempre maggiore rispetto all'errore percentuale relativo vero. L'errore assoluto massimo si ha quando la radice coincide con un estremo dell'intervallo di ricerca $]x_l,x_r[$

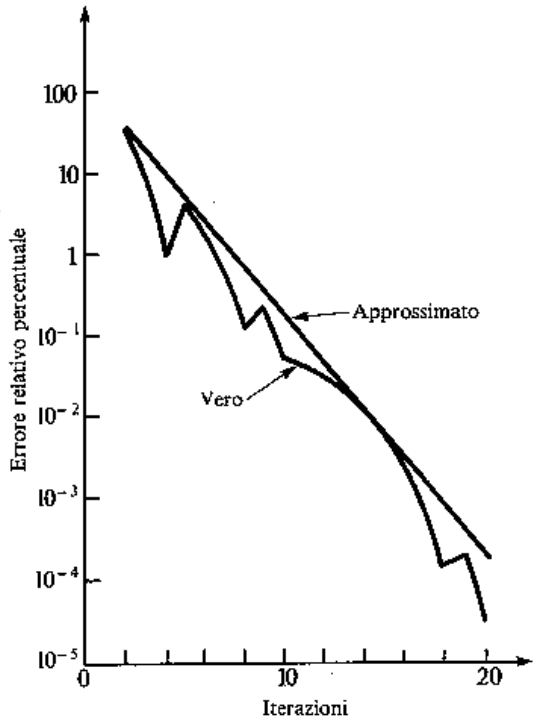


Figura 2.7 Errori nel metodo di bisezione: sono riportati in grafico l'errore vero e l'errore approssimato in funzione del numero di iterazioni

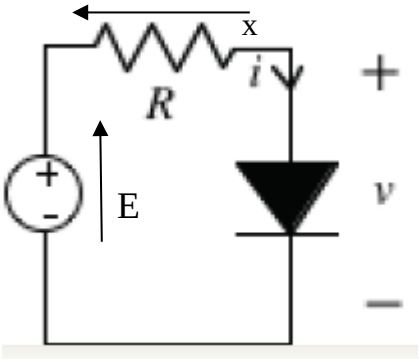
Dall'analisi dell'errore si può dimostrare che il metodo ha una convergenza del primo ordine. Infatti, alla generica iterazione k sia Δ_k l'ampiezza dell'intervallo in cui si cerca la radice. Allora l'errore assoluto massimo che si può commettere è $\Delta_k/2$. Pertanto, osservando che $\Delta_k=\Delta_{k-1}/2$ si ha: $\frac{|e_k|}{|e_{k-1}|} \le \frac{\Delta_k/2}{\Delta_{k-1}/2} = \frac{1}{2}$

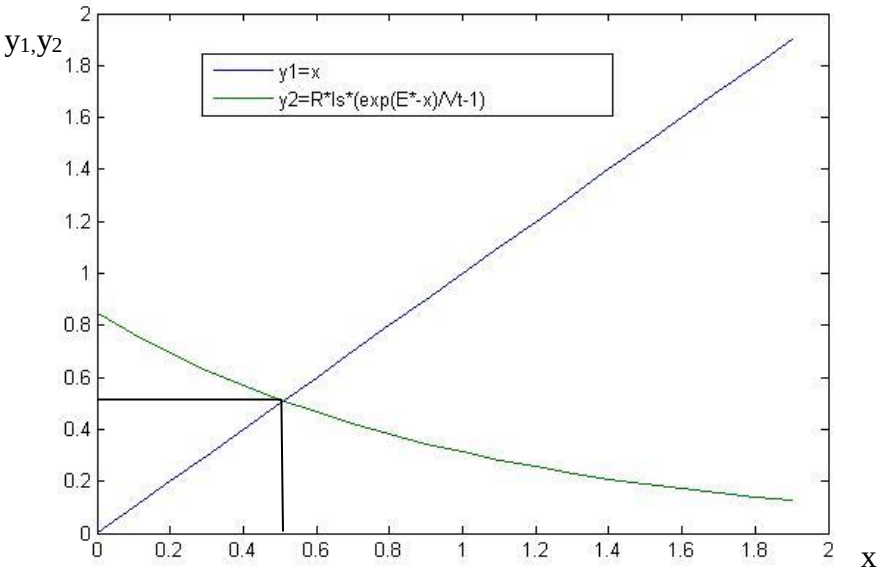
PROGRAMMA FORTRAN PER IL METODO DI BISEZIONE

```
EXTERNAL f
REAL f
      READ(5,1)xl,xu,es,im                ! xl,xu = valori di tentativo
1     FORMAT(3F10.0,I5)                   ! es = errore percentuale accettabile
      aa=f(xl)*f(xu)                      ! im = massimo numero di iterazioni
      IF(aa.GE.0)GOTO 301
      xr=(xl+xu)/2
      DO 240 ni=2,im
            aa=f(xl)*f(xr)                ! controlla se la radice È compresa
            IF(aa.EQ.0)GOTO 300           ! fra xl e xu
            IF(aa.LT.0)xu=xr
            IF(aa.GT.0)xl=xr
            xn=(xl+xu)/2                  ! xn = nuova stima della radice
            IF(xn.EQ.0) GOTO 230
            ea=ABS((xn-xr)/xn)*100        ! ea = stima dell'errore percentuale
            IF(ea.LE.es)goto 280          ! valutazione dell'errore
230         xr=xn
240   CONTINUE
      WRITE(6,2)
2     FORMAT(' ','Root not reached')
      WRITE(6,3)xr,ea
3     FORMAT(' ',2F10.3)
      GOTO 300
280   WRITE(6,4)xn,ea,ni
4     FORMAT(' ',2F10.3,I5)
      GOTO 300
300   WRITE(6,5)xr
5     FORMAT(' ','Exact Root = ',F10.3)
      STOP
301 write(6,*)'Error, xl and xr of equal sign'
      END

REAL FUNCTION f(x)
! f(x)=funzione della quale si ricerca la radice (va passata in external)
      f(x)=exp(-x)-x
RETURN
END
```

Esempio2.4:





Per il circuito dell'esempio 2.1, assumendo R=100 Ω, e*(t)=10 V e che il diodo abbia una caratteristica del tipo $i_d = q(v_d) = I_s\left(e^{vd/Vt} - 1\right)$ dove I_s è la corrente di saturazione (assunta pari a $10*10^{-9}$ A) e V_t equivalente in volt della temperatura (dell'ordine di 26mV), si determini attraverso il metodo di bisezione la tensione ai capi del resistore (il valore vero della tensione alla 4 cifra significativa è x=0.5092 V).

Dal grafico notiamo che la radice si trova tra x_l=0 e x_u=1 → x_r=(0+1)/2=0.5

It	xl	f(xl)	Xu	f(xu)	xr	f(xr)	Err rel per
1	0	-.8472	1	.6883	0.5	-.1383	33.3%
2	0.5	-.1383E-1	1	.6883	0.75	.3498	20.0%
3	0.5	-.1383E-1	0.75	.3498	0.625	.1715	11.1%
4	0.5	-.1383E-1	0.625	.1715	0.5625	.7979E-1	5.88%
5	0.5	-.1383E-1	0.5625	.7980E-1	0.53125	.3322E-1	3.03%
6	0.5	-.1383E-1	0.53125	.3322E-1	0.515625	.9756E-2	1.54%

PROGRAMMA MATLAB PER IL METODO DI BISEZIONE (ES.2.4)

```
% Metodo di bisezione per la ricerca degli zeri
% Circuito costituito da un gen. reale e da un diodo.
% Incognita tensione ai capi del diodo
% 1) determinare gli estremi dell'intervallo partire da x=0 e con dx=0.1 e valutare F
% fino a che la funzione non cambia di segno
clear all
Vt=0.026;
Res=100;
eg=10;
Is=10*10^(-9);
x(1)=0;
y(1)=x(1)-Res*Is*(exp(eg-x(1))/Vt-1);
dx=eg/10;
for i=2:10
 x(i)=x(i-1)+dx
 y(i)=x(i)-Res*Is*(exp(eg-x(i))/Vt-1);
 if y(i)*y(i-1) <= 0
 break
 end
end
%implementazione del metodo di bisezione
% im=massimo numero di iterazioni
% xl,xu=valori di tentativo es=errore percentuale accettabile
im=100
es=1e-3
xl(1)=x(i-1);
xu(1)=x(i);
yl(1)=y(i-1);
yu(1)=y(i);
prody=yl*yu;
if(prody<=0)
 xr(1)=(xl+xu)/2
 for ni=2:im
% yl(ni-1)=xl(ni-1)-Res*Is*(exp(eg-xl(ni-1))/Vt-1)
 yr(ni-1)=xr(ni-1)-Res*Is*(exp(eg-xr(ni-1))/Vt-1)
 prody=yl(ni-1)*yr(ni-1);
 if(prody==0)
 break
 end
 if prody < 0
 xu(ni)=xr(ni-1)
 yu(ni)=yr(ni-1)
 xl(ni)=xl(ni-1)
 yl(ni)=yl(ni-1)
 end
 if prody > 0
 xl(ni)=xr(ni-1)
 yl(ni)=yr(ni-1)
 xu(ni)=xu(ni-1)
 yu(ni)=yu(ni-1)
 end
 xn=(xl(ni)+xu(ni))/2
 if(xn~=0)
 ea(ni)=abs((xn-xr(ni-1))/xn)*100
 if(ea(ni)<=es)
 break
 end
 end
 xr(ni)=xn
 end
end
```

Un ulteriore svantaggio del metodo di bisezione

Nel metodo di Bisezione l'intervallo di ricerca della radice viene diviso esattamente a metà ad ogni iterazione.

Non si tiene cioè conto dell'informazione fornita dai valori di $f(x_l)$ e $f(x_u)$. Per esempio se $|f(x_l)|<<|f(x_u)|$ è probabile che la radice sia molto più vicina a x_l che a x_u.

Per sfruttare questa informazione si può ricorrere al

METODO DELLA FALSA POSIZIONE

Eseguiamo la seguente analisi grafica :

- si congiungano con una linea retta i punti $(x_l, f(x_l))$ e $(x_u, f(x_u))$
- come stima della radice si assuma il punto di intersezione, x_r, della retta con l'asse delle ascisse
- il valore ottenuto va poi a sostituire quell'estremo di partenza dell'intervallo, x_l o x_u, per il quale la funzione assume lo stesso segno che in $f(x_r)$.

Questa stima è presumibilmente migliore rispetto a $(x_u+x_l)/2$ ottenuta col metodo di bisezione (Vedi figura)

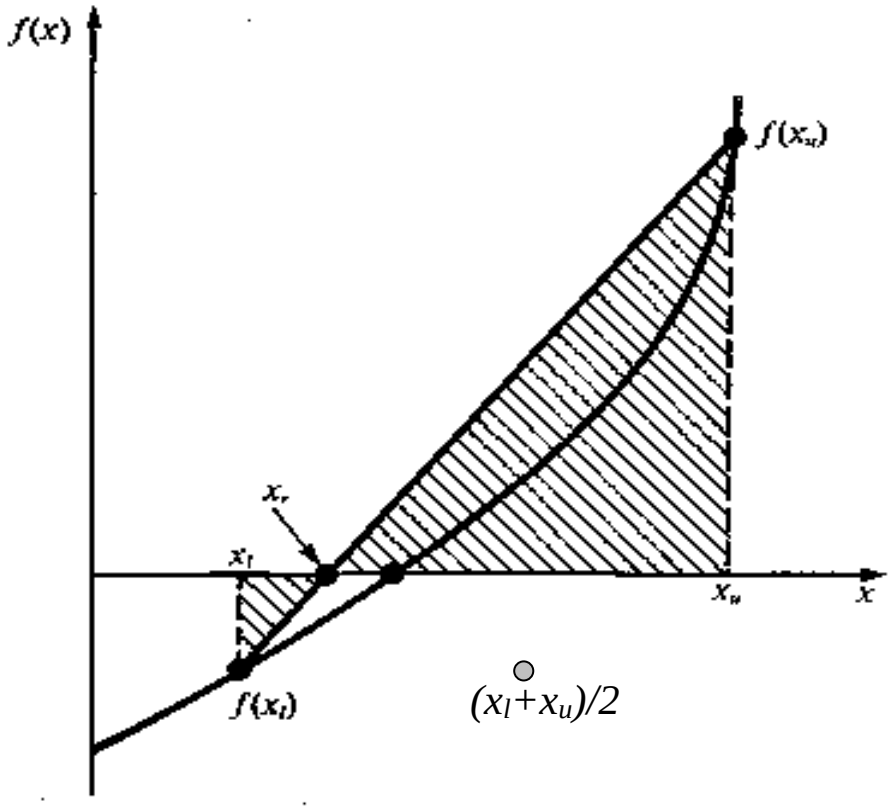


Fig.2.8 Rappresentazione grafica del metodo della falsa posizione

METODO DELLA FALSA POSIZIONE

Il nome del metodo (in letteratura sovente viene anche detto "regula falsi") deriva dal fatto che la sostituzione della curva di f(·) con una retta dà luogo ad una falsa posizione della radice.

Dai teoremi sui triangoli simili si può dedurre l'espressione analitica della radice:

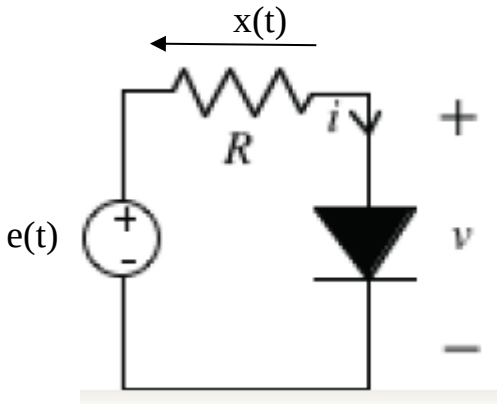
$$x_r = x_u - \frac{f(x_u)(x_l - x_u)}{f(x_l) - f(x_u)}$$

L'algoritmo è con piccole modifiche identico a quello di bisezione.

Algoritmo

1. Si sceglie una stima inferiore, x_l, ed una stima superiore, x_u, della radice, tali per cui $f(\mathbf{x_l})*f(\mathbf{x_u})<0$
2. Come prima stima della radice si assume $x_r = x_u - \frac{f(x_u)(x_l - x_u)}{f(x_l) - f(x_u)}$
3. Si calcola il valore della funzione in $\mathbf{x_r}$ e lo si confronta con un indice di tolleranza ε_{tol}
4. se $|f(\mathbf{x_r})|\leq\varepsilon_{tol}$, si assume come radice $\mathbf{x_r}$ e il programma termina;
5. se $f(\mathbf{x_l})*f(\mathbf{x_r})<0$, la radice si trova in $]x_l, x_r[$; $x_u=x_r$
6. se $f(\mathbf{x_l})*f(\mathbf{x_r})>0$, la radice si trova in $]x_r, x_u[$; $x_l=x_r$
7. Si calcola una nuova stima della radice ponendo $x'_r = x_u - \frac{f(x_u)(x_l - x_u)}{f(x_l) - f(x_u)}$
8. Si controlla se l'accuratezza della stima è soddisfacente, confrontando l'ampiezza dell'intervallo $\Delta x=|(x_u-x_l)|$ con un indice preassegnato $\varepsilon_{amp,des}$; equivalentemente si può verificare che risulti l'errore approssimato percentuale relativo $\varepsilon_a=|(x'_r-x_r)/x'_r)|*100\% <\varepsilon_{a,des}$
9. se $\Delta x\leq\varepsilon_{amp,des}$ ($\varepsilon_a\leq\varepsilon_{a,des}$), o il numero di iterazioni *nit* > *nitmax* si assume come radice $\mathbf{x_r}$ e il programma termina.
10. se $\Delta x\geq\varepsilon_{amp,des}$ ($\varepsilon_a\geq\varepsilon_{a,des}$) si pone $x_r=x'_r$ $nit=nit+1$ si riprende dal passo 3

Esempio 2.5:



Assumendo R=100 Ω, e*(t)=10 V e che il diodo abbia una caratteristica del tipo $i_d = q(v_d) = I_s\left(e^{vd/Vt} - 1\right)$ dove I_s è la corrente di saturazione (assunta pari a $10*10^{-9}$ A) e V_t equivalente in volt della temperatura (dell'ordine di 26mV), si determini la tensione ai capi del resistore col metodo della falsa posizione (il valore vero della tensione alla 4 cifra significativa è x=0.5092 V).

Soluzione Essendo il resistore e il diodo componenti strettamente passivi, dal teorema di Non amplificazione delle tensioni sappiamo che la tensione ai capi del diodo deve essere minore rispetto a quella imposta ai capi del generatore:

$0V \le v_d \le 10V, \longrightarrow x_l = 0\ V, \quad x_u = 10\ V$

L'equazione risolvente si ottiene facilmente dalla LKT e dalle relazioni caratteristiche dei componenti:

$$x(t) = R\, i_d(t) = R\, q(v_d(t)) = R\, q(e^*(t) - x(t))$$

$$\rightarrow \qquad f(t) = x(t) - R\, I_S\left(e^{(e^*(t)-x(t)/Vt} - 1\right) = 0$$

Sostituendo si ottiene: $f = x - 100*10*10^{-9}\left(e^{(10-x)/0.026} - 1\right) = 0$

$$x_r = x_u - \frac{f(x_u)(x_l - x_u)}{f(x_l) - f(x_u)} = 10 - \frac{-0.(0\text{-}10.00)}{-0.8472 - (10.00)} = 0.7810$$

It	xl	f(xl)	xu	f(xu)	xr	f(xr)	Err rel.v.	Err.r.ap	E.ass.a
1	0	-.8472	10	10	.7810	.3930	53.394%		
2	0	-.8472	0.7810	.3930	.5335	.3658E-1	4.780%	46.396%	.248e-1
3	0.	-.8472	.5335	.3658E-1	.5114	.3403E-2	0.443%	4.318%	.221e-2
4	0.	-.8472	.5114	.3403E-2	.5094	.3165E-3	0.041%	0.402%	.205e-3
5	0.	-.8472	.5094	.3165E-3	.5092	.2943E-4	0.0038%	0.037%	.190e-4

Esempio 2.6. Si ripeta l'esercizio, assumendo che il generatore che alimenta il circuito abbia un andamento sinusoidale ($e(t)=E_m \sin(\omega t)$).

PROG. MATLAB PER IL MET. DELLA FALSA POSIZIONE (ES.2.5)

```
%metodo della falsa posizione si parte dal valore per cui la funzione è
% cambiata di segno
clear all
Vt=0.026; Res=100; eg=10; Is=10*10^(-9);
xl=0; xu=eg;
xlsave=xl; xusave=xu;
yl=xl-Res*Is*(exp(eg-xl)/Vt-1);
yu=xu-Res*Is*(exp(eg-xu)/Vt-1);
niter=100;
epstol=1e-15;
for i=1:niter
 xr(i)=xu-yu*(xl-xu)/(yl-yu);
 yr(i)=xr(i)-Res*Is*(exp(eg-xr(i))/Vt-1);
 if abs(yr(i))<epstol
 xsol=xr(i);
 ysol=yr(i);
 break
 elseif yl*yr(i)>0
 yl=yr(i);
 xl=xr(i);
 else
 yu=yr(i);
 xu=xr(i);
 end
 if i==niter
 xsol=xr(i);
 ysol=yr(i);
 end
end
xr=[]
yr=[]
xsoltrue=xsol; xl=xlsave; xu=xusave;
yl=xl-Res*Is*(exp(eg-xl)/Vt-1); yu=xu-Res*Is*(exp(eg-xu)/Vt-1);
niter=20; epstol=1e-5
for i=1:niter
 xr(i)=xu-yu*(xl-xu)/(yl-yu);
 yr(i)=xr(i)-Res*Is*(exp(eg-xr(i))/Vt-1);
 errtruea(i)=abs(xr(i)-xsoltrue);
 errtruer(i)=100*errtruea(i)/xsoltrue;
 if i>1
 errappa(i)=abs(xr(i)-xr(i-1));
 errappr(i)=100*errappa(i)/abs(xr(i));
 end
 if abs(yr(i))<epstol
 xsol=xr(i);
 ysol=yr(i);
 break
 elseif yl*yr(i)>0
 yl=yr(i);
 xl=xr(i);
 else
 yu=yr(i);
 xu=xr(i);
 end
 if i==niter
 xsol=xr(i);
 ysol=yr(i);
 end
end
```

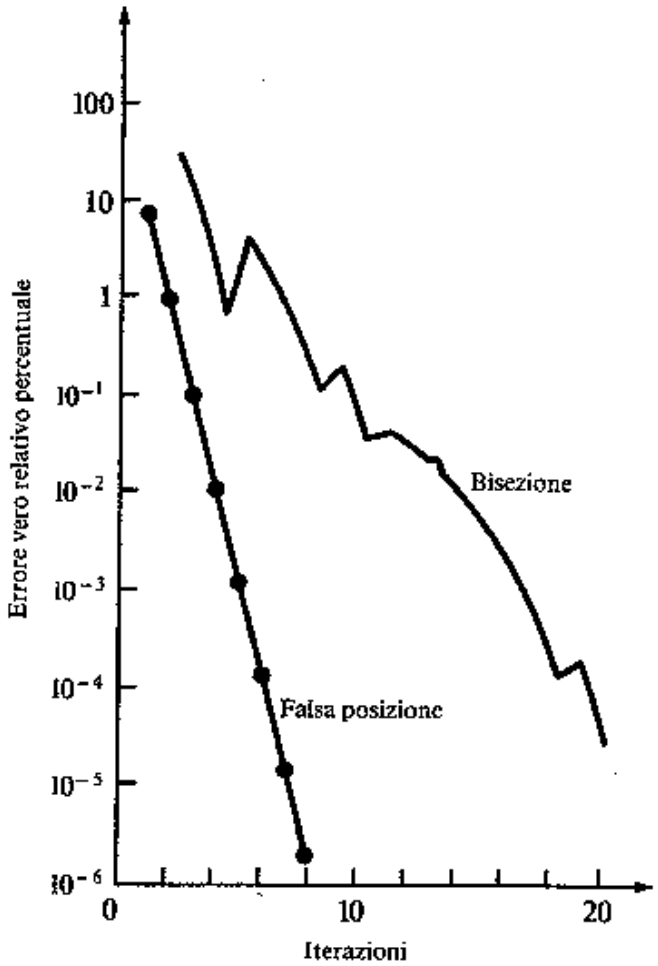


Fig. 2.9 Confronto tra gli errori relativi veri al crescere delle iterazioni per i metodi di bisezione e della falsa posizione per $f(x)=e^{-x}-x$. Si noti come in questo caso il metodo della falsa posizione è più efficiente

Programma Fortran per il metodo della falsa posizione

```
EXTERNAL f
REAL f
      READ(5,1)xl,xu,es,im                    ! xl,xu = valori di tentativo
1     FORMAT(3F10.0,I5)                       ! es = errore percentuale accettabile
      aa=f(xl)*f(xu)                          ! im = massimo numero di iterazioni
      IF(aa.GE.0)GOTO 310
      xr=xu-f(xu)*(xl-xu)/(f(xl)-f(xu))
      DO 240 ni=2,im
            aa=f(xl)*f(xu)                    ! controlla se la radice È compresa
            IF(aa.EQ.0)GOTO 300               ! fra xl e xu
            IF(aa.LT.0)xu=xr
            IF(aa.GT.0)xl=xr
            xn=xu-f(xu)*(xl-xu)/(f(xl)-f(xu)) ! xn = nuova stima della radice
            IF(xn.EQ.0) GOTO 230
            ea=ABS((xn-xr)/xn)*100            ! ea = stima dell'errore percentuale
            IF(ea.LE.es)goto 280              ! valutazione dell'errore
230         xr=xn
240   CONTINUE
      WRITE(6,2)
2     FORMAT(' ','Root not reached')
      WRITE(6,3)xr,ea
3     FORMAT(' ',2F10.3)
      GOTO 310
280   WRITE(6,4)xn,ea,ni
4     FORMAT(' ',2F10.3,I5)
      GOTO 310
300   WRITE(6,5)xr
5     FORMAT(' ','Exact Root = ',F10.3)
      STOP
310 WRITE(6,*)'Error: f(xl)*f(xu)>0'
      STOP
      END

REAL FUNCTION f(x)
! f(x)=funzione della quale si ricerca la radice (va passata in external)
      f(x)=exp(-x)-x
RETURN
END
```

Confronto tra i due metodi

Vantaggi del metodo della falsa posizione.

Nel caso della **falsa posizione** l'errore decresce molto più velocemente a causa della maggiore efficienza della strategia di ricerca delle radici

Svantaggi del metodo della falsa posizione

Nel caso della **bisezione** il margine di incertezza sulla soluzione (ovvero l'intervallo di ricerca delle radici) si riduceva. Per questo l'ampiezza dell'intervallo $\Delta x/2=|(x_u-x_l)|/2$ dava una misura dell'errore. Questo non è garantito nel caso del metodo della **falsa posizione** in quanto uno degli estremi può rimanere invariato durante tutta la ricerca, mentre l'altro converge sulla radice. In questi casi l'ampiezza dell'intervallo di ricerca tende a un valore costante.

N.B.

Si noti che, **nel caso di rapida convergenza verso la soluzione**, l'errore assoluto (e, conseguentemente, quello relativo) stimato costituisce un criterio di stima dell'errore molto conservativo (sempre più pessimistico) dell'errore assoluto vero. Infatti se la soluzione stimata alla k^{ma} iterazione, x_r^k, converge rapidamente verso la soluzione analitica, x_t, risulta:

$$E_a^{k+1}=|x_r^{k+1}-x_r^k| >> |x_r^{k+1}-x_t| =E_t^{k+1}$$

Si può dimostrare che questo metodo presenta una **convergenza** di carattere **super-lineare**, ma la stima del suo esatto ordine è piuttosto complicata.

Esempio in cui il metodo di bisezione è migliore rispetto a quello della falsa posizione

Determinare la radice di $f(x)=x^{10}-1$ con $x \in]0,1.3[$

Ea
8.7E-2
8.1e-2
7.5e-2
7.0e-2

Soluzione: con bisezione si ottengono i seguenti risultati

Ea	x_l	x_u	x_r	$\lvert \epsilon_t \rvert$ %	$\lvert \epsilon_a \rvert$ %
	0	1.3	0.65	35	
3.3e-1	0.65	1.3	0.975	2.5	33.3
1.6e-1	0.975	1.3	1.1375	13.8	14.3
8.1e-2	0.975	1.1375	1.05625	5.6	7.7
4.1e-2	0.975	1.05625	1.015625	1.6	4.0

Dopo cinque iterazioni, quindi, l'errore vero si è ridotto a meno del 2%. Con il metodo della falsa posizione, l'efficienza ottenibile è parecchio inferiore:

Iterazione	x_l	x_u	x_r	$\lvert \epsilon_t \rvert$ %	$\lvert \epsilon_a \rvert$ %
1	0	1.3	0.09430	90.6	
2	0.09430	1.3	0.18176	81.8	48.1
3	0.18176	1.3	0.26287	73.7	30.9
4	0.26287	1.3	0.33811	66.2	22.3
5	0.33811	1.3	0.40788	59.2	17.1

Fig. 2.10 Confronto tra i metodi di bisezione e della falsa posizione. In alcuni casi l'efficienza del metodo della falsa posizione crolla, mentre il metodo di bisezione continua a funzionare.

Nel presente esempio si vede che dopo 5 iterazioni l'errore vero col metodo della falsa posizione è ancora del 59%, mentre nel caso della bisezione è dell'1.6%

Inoltre risulta $\varepsilon_a \leq \varepsilon_t$, cioè l'errore approssimato è una stima ottimistica dell'errore vero (comportamento non gradito). Il motivo è che in questo caso la soluzione converge lentamente. Per capire se la soluzione converge rapidamente o meno, si osservi l'andamento dell'errore ass. stimato.

RICERCHE INCREMENTALI E DETERMINAZIONE DEI VALORI INIZIALI

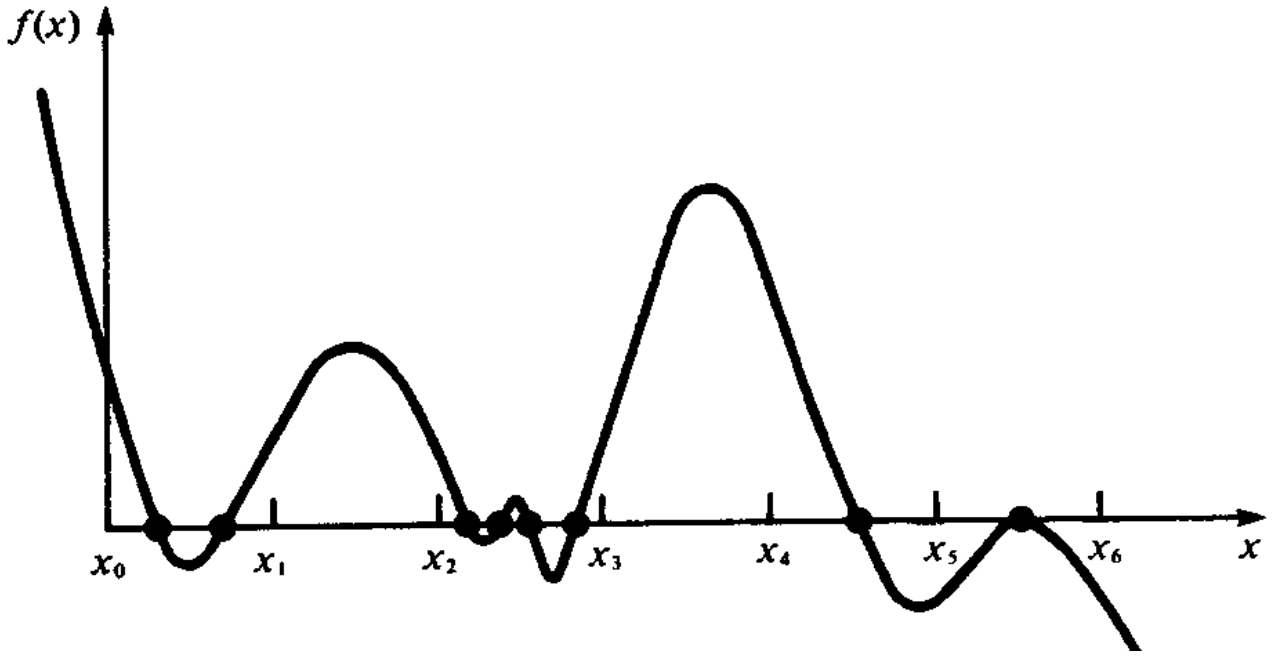


Fig. 2.11 Un'equazione non-lineare con molti zeri.

In alcuni problemi viene richiesto di determinare **tutte le radici di un'equazione f(x)**. In questi casi il grafico della funzione può essere un utile strumento per indirizzare la ricerca.

Alternativamente si può effettuare una ricerca incrementale all'inizio del programma: per trovare gli zeri di f(x) nell'intervallo $[x_{in},x_{fin}]$

1. si fissa l'incremento Δx,
2. si determina $N=(x_{fin}, x_{in})/\Delta x$;
3. viene valutata $f(x_j)$, per $x_j=x_{in} + j*\Delta x$, $j = 1,N$
4. quando si scopre un cambiamento di segno(assunta la funzione continua e senza radici multiple), si deduce che all'interno dell'ultimo incremento si trova una radice
5. si pone $x_l=x_j$; $x_u=x_{j+1}$; e si utilizza uno dei metodi chiusi esaminati

Problema

Se Δx è troppo grande si possono tralasciare delle radici

Se Δx è troppo piccolo la ricerca può richiedere un tempo eccessivo

ESERCIZI

Esercizio 2.1

Risolvere con il metodo della bisezione il circuito di Figura 2.1 assumendo che il resistore non lineare sia un diodo a giunzione *p-n*, $R = 1\Omega$ e $E = 1V$. Ripetere il calcolo variando i parametri del circuito.
Con il metodo della bisezione è possibile determinare tutte le soluzioni reali di un'equazione algebrica scalare scegliendo in modo opportuno l'intervallo iniziale I_0 .

Esercizio 2.2

Risolvere con il metodo della bisezione il circuito di Figura 2.1 assumendo che il resistore non lineare sia un diodo tunnel e considerando due insiemi di valori per i parametri: E = 12V , R = 6Ω; E = 15V , R = 6Ω . Nel primo caso il circuito ha una sola soluzione, mentre nell'altro ne ha tre. Ciò può essere facilmente verificato attraverso il metodo grafico.

Problema 2.1

Il circuito di Figura 2.1 può essere considerato come un sistema ingresso-uscita, assumendo, ad esempio, come ingresso la tensione del generatore e come uscita l'intensità di corrente. Determinare la caratteristica ingresso-uscita al variare della resistenza R sia nel caso in cui N è un diodo a giunzione pn, sia nel caso in cui è un diodo tunnel.

Problema 1.2

Studiare con il metodo della bisezione il circuito raddrizzatore rappresentato in Figura 2.12 al variare dei parametri E_m, ω, R . In particolare, determinare l'andamento del valore medio dell'intensità di corrente e delle ampiezze delle prime tre armoniche al variare dei parametri E_m, ω, R.

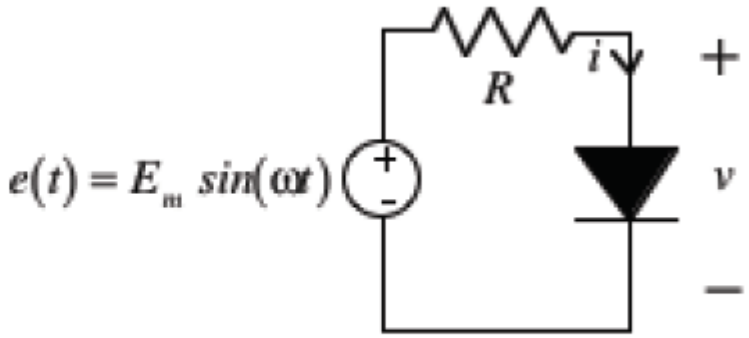


Fig.2.12 *Circuito raddrizzatore*

METODI APERTI

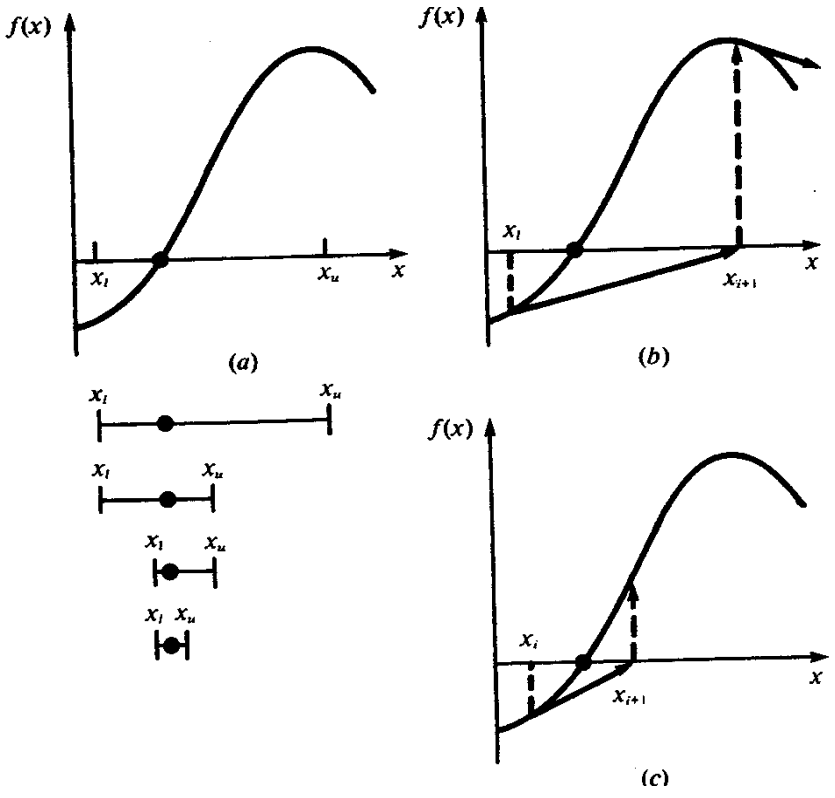
Nei metodi chiusi

1. Occorre fissare a priori l'intervallo di ricerca della radice;
2. La convergenza verso la soluzione è assicurata in quanto l'intervallo di ricerca diminuisce sempre col numero di iterazioni (perciò questi metodi sono detti anche *convergenti*).

Nei metodi aperti

1. Si utilizzano formule che richiedono *un* solo valore di x oppure *due* ***che non racchiudono necessariamente la radice***.

2. La convergenza non è assicurata, cioè questi metodi possono talvolta divergere. Tuttavia se convergono, lo fanno molto più rapidamente dei metodi chiusi.



a) Metodo di bisezione: sempre convergente

b) metodo aperto: divergente (punto iniziale errato)

c) metodo aperto: convergente (punto iniziale corretto)

Metodo di Picard o delle sostituzioni successive

È già stato introdotto quando si è illustrato il concetto di convergenza.

Definiamo **punto fisso** di un'applicazione g(x) quel valore di x per il quale risulta: x=g(x)
Il metodo delle sostituzioni successive consiste nella ricerca del cosiddetto **punto fisso*** α di una trasformazione g(x) definita opportunamente.

Ad esempio si può porre $g(x) = x + k*f(x)$, $k \in \Re$,

di modo che nel punto fisso $x=\alpha$ risulti $g(\alpha)=\alpha$ (e $f(\alpha)=0$).

Per la ricerca del punto fisso si può utilizzare la tecnica iterativa:

$x^{k+1}=g(x^k), k=0,1,2\ldots$
$x^0=x_0$, dove $x_0 \in \Re$ è una stima iniziale della soluzione.

Esempio 2.6
Determinare la radice di $f(x)=e^{-x}-x$ col metodo di Picard (valore esatto 0.56714329). Si assuma come valore iniziale $x_0=0$. Si può porre:

$$x^{k+1}=\exp(-x^k), \text{ con } x^0=0$$

Iterazione	x_i	$\lvert \epsilon_t \rvert$ %	$\lvert \epsilon_a \rvert$ %
0	0	100	
1	1.000000	76.3	100.0
2	0.367879	35.1	171.8
3	0.692201	22.1	46.9
4	0.500473	11.8	38.3
5	0.606244	6.89	17.4
6	0.545396	3.83	11.2
7	0.579612	2.20	5.90
8	0.560115	1.24	3.48
9	0.571143	0.705	1.93
10	0.564879	0.399	1.11

Come si vede il metodo è convergente e l'errore relativo approssimato dà una stima conservativa dell'errore relativo vero (valore vero x=0.567143).

Convergenza del metodo delle sostituzioni successive

Si è affermato in precedenza che il metodo converge se risulta $|g(x)'| < 1$ (ovvero se g(x) è una **contrazione**).

Si noti che nell'esempio precedente, l'errore relativo esatto per ogni iterazione è proporzionale con un fattore (0.5:0.6) all'errore commesso nell'iterazione precedente. **Cioè la convergenza del metodo è lineare**.

Queste affermazioni possono essere dimostrate nel modo seguente:

La soluzione esatta è un punto fisso, si ha cioè $x_t = g(x_t)$ (1)

L'equazione approssimata risulta essere alla k^{ma} iterazione $x^{k+1} = g(x^k)$ (2)

Sottraendo membro a membro, risulta: $x_t - x^{k+1} = g(x_t) - g(x^k)$ (3)

Per il teorema del valore medio se *g(x)* è continua con la sua derivata prima nell'intervallo $x^k \leq x \leq x_t$, allora esiste $\xi \in [x^k, x_t]$, tale che $g'(\xi) = \dfrac{g(x_t) - g(x^k)}{x_t - x^k}$

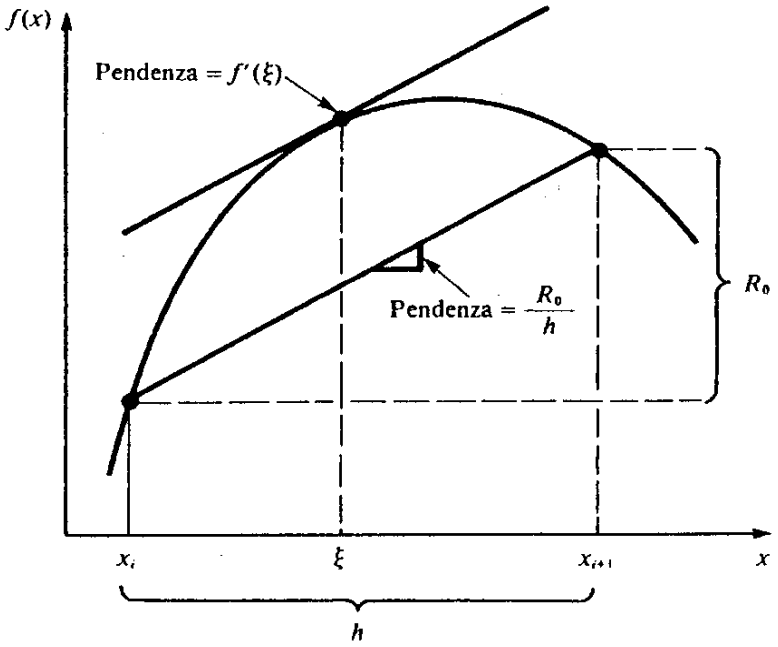


Figura 3.5 Rappresentazione grafica del teorema del valor medio.

Il membro di destra è la pendenza della linea che congiunge i punti (a,g(a)) e (b,g(b)). Pertanto questo teorema stabilisce che esiste almeno un punto in [a,b], nel quale la derivata alla funzione g(.) risulta parallela alla linea congiungente i punti estremi (a, g(a)) e (b, g(b)).

Dall'espressione della derivata risulta $(x_t - x^k) g'(\xi) = g(x_t) - g(x^k)$

Sostituendo nella (2): $E^{k+1} = (x_t - x^{k+1}) = (x_t - x^k)\, g'(\xi) = E^k\, g'(\xi)$ **(convergenza lineare)**

1) se $|g'(\xi)|<1$ (risp. >1) l'errore diminuisce (risp. cresce).
2) se $|g'(\xi)|<1$ la rapidità della convergenza aumenta al diminuire di $|g'(\xi)|$

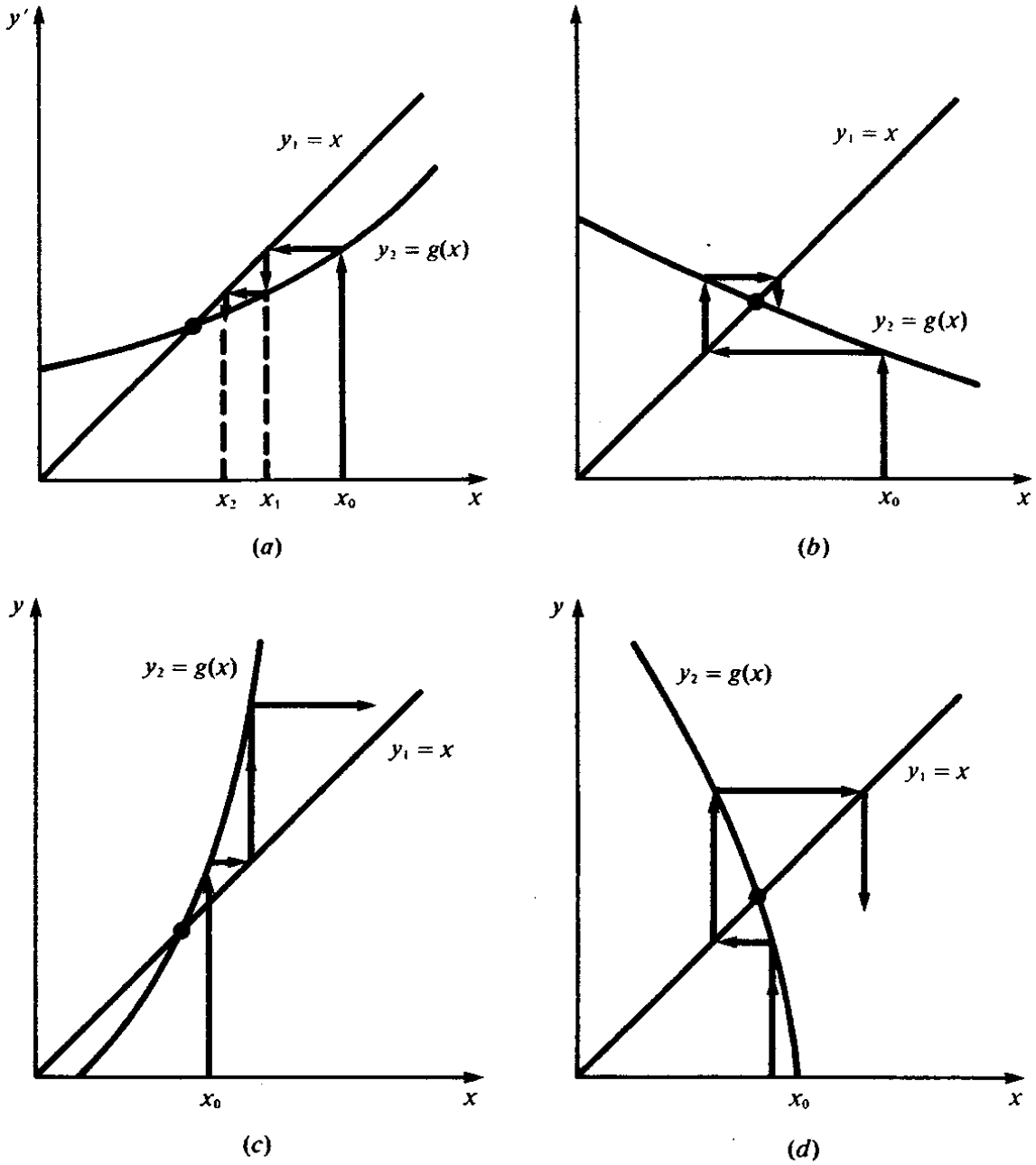


Figura 5.3 Rappresentazione grafica della (*a* e *b*) convergenza e (*c* e *d*) divergenza delle sostituzioni successive. Gli andamenti (*a*) e (*c*) vengono detti monotoni mentre in (*b*) e (*d*) l'andamento è oscillante o spiraliforme. Osservate che la convergenza ha luogo quando $|g'(x)| < 1$.

PROGRAMMA IN FORTRAN PER LE SOSTITUZIONI SUCCESSIVE

```
EXTERNAL f
REAL f
! xr = valore di tentativo
! es = errore percentuale accettabile
! im = massimo numero di iterazioni
      READ(5,1)xr, es, im
1     FORMAT(2F10.0,I5)
      DO 180 ni=1,im
! xn = stima della radice
            xn=f(xr)
            IF(xn.EQ.0) GOTO 170
! ea = stima dell'errore percentuale
            ea=ABS((xn-xr)/xn)*100
            IF(ea.LE.es)goto 210
170         xr=xn
180   CONTINUE
      WRITE(6,2)
2     FORMAT(' ','Root not reached')
      ni=ni-1
210 WRITE(6,3)xn, ea, ni
3     FORMAT(' ',2F10.3,I5)
      STOP
      END

REAL FUNCTION f(x)
! f(x)=funzione della quale si ricerca la radice (va passata in external)
      f(x)=exp(-x)
RETURN
END
```

Programma Matlab per le sostituzioni successive (Picard)

```
function [zero, niter]=picard1D(f,x0,toll,nmax)
% Metodo di Picard
x=x0;
fx=eval(f); niter=0; diff=toll+1;
while diff>=toll&niter<=nmax
 niter=niter+1;diff=fx; x=x+fx;
 diff=abs(diff); fx=eval(f);
end;
zero=x;
return
```

METODO DI NEWTON-RAPHSON

La formula più usata per il calcolo delle radici è quella di Newton-Raphson.

L'idea è quella di sviluppare la f(x) in serie di Taylor nell'intorno di una stima iniziale della radice x_i arrestando lo sviluppo al secondo termine ed imponendo lo sviluppo uguale a zero:

$$0 = f(x_{i+1}) = f(x_i) + \left.\frac{\partial f(x)}{\partial x}\right|_{x=xi} (x_{i+1} - x_i) \rightarrow \quad x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

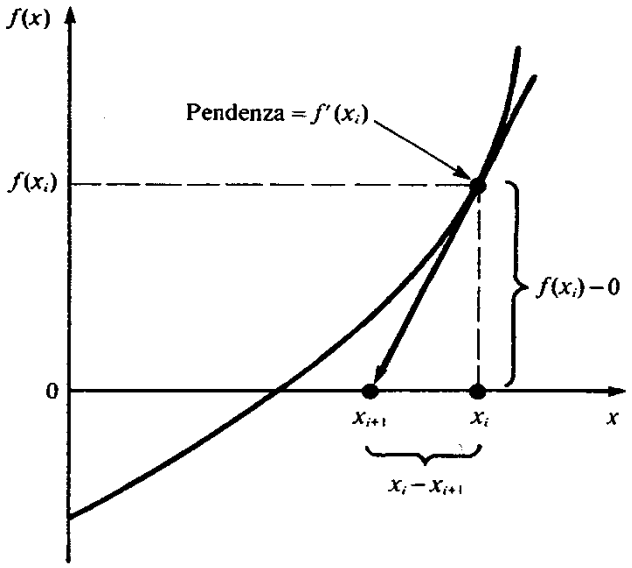


Figura 5.5 Rappresentazione grafica del metodo di Newton-Raphson. La tangente alla funzione in x_i (cioè con pendenza $f'(x_i)$) viene fatta intersecare con l'asse x per ottenere una stima della radice in x_{i+1}.

Esempio: usare la formula di Newton-Raphson per determinare la radice di $f(x)=e^{-x}-x$ (valore esatto 0.567143290409784). Si assuma $x_0=0$

It	x_i	$f(x_i)$	$f'(x_i)$	ε_t	ε_a
1	0	1	-2		
2	0.500000000	1.0653e-1	-1.6065	11.8%	100%
3	0.566311003	1.3045e-3	-1.5676	1.47e-1 %	11.71%
4	0.567143165	1.9648e-7	-1.5671	2.20e-5 %	0.147%
5	0.567143290	4.5519e-15	-1.5671	5.1e-13 %	2.21e-5%

CRITERI DI ARRESTO E STIMA DELL'ERRORE

Dimostriamo che il metodo di Newton-Raphson presenta una convergenza del secondo ordine.
Ricordiamo che lo sviluppo di Taylor può essere scritto come:

$$f(x_{i+1})= f(x_i)+ f'(x_i)(x_{i+1}-x_i)+ f''(\xi)/ 2! (x_{i+1}-x_i)^2 \text{ , con } \xi\in[x_i,x_{i+1}] \quad (1)$$

Si è visto come troncando al secondo termine si ottiene la formula di Newton-Raphson:

$f(x_{i+1})\cong f(x_i)+ f'(x_i)(x_{i+1}-x_i)$ (2) $\longrightarrow$ $x_{i+1} = x_i - \dfrac{f(x_i)}{f'(x_i)}$

Per stimare l'errore, si deve tenere conto che il valore esatto della radice può essere ottenuto usando la (1) e che nella radice $x_{i+1}= x_r$:

$$0 = f(x_i)+f'(x_i)(x_r-x_i) + f''(\xi)/ 2! (x_r-x_i)^2 \quad (3)$$

Sottraendo la (2) dalla (3), e ricordando che $E_{i+1}= (x_r-x_{i+1})$ e che $E_i= (x_r-x_i)$ si ottiene:

$$0 = f'(x_i) (x_r-x_{i+1}) + f''(\xi)/ 2! (x_r-x_i)^2 =$$
$$= f'(x_i) E_{i+1} + f''(\xi)/ 2! E_i^2$$

Ed infine: $E_{i+1} =- E_i^2 f''(\xi)/(f'(x_i) 2!)$ (4)

Cioè ammettendo la convergenza del metodo, da una certa iterazione in poi (N.B. **la convergenza è asintotica**) il numero di cifre decimali esatte raddoppia (almeno) ad ogni iterazione.

Dalla (4), ponendo $\xi=x_i=x_t$ =0.567143290409784 si determina l'errore:

$f''(x_t)= 0.56714329$; $f'(x_t)= -1.56714329$; $E_{i+1} =- 0.18095 E_i^2$

It	x_i	$f(x_i)$	$f'(x_i)$	**Estimato (4)**	**Et**
1	0	1	-2	0.56714329	0.56714329
2	0.500000000	1.0653e-1	-1.6065	5.82E-2	6.714329E-2
3	0.566311003	1.3045e-3	-1.5676	8.158E-4	8.323E-4
4	0.567143165	1.9648e-7	-1.5671	1.25E-7	1.25E-7
5	0.567143290	4.5519e-15	-1.5671	2.83E-15	2.83E-15

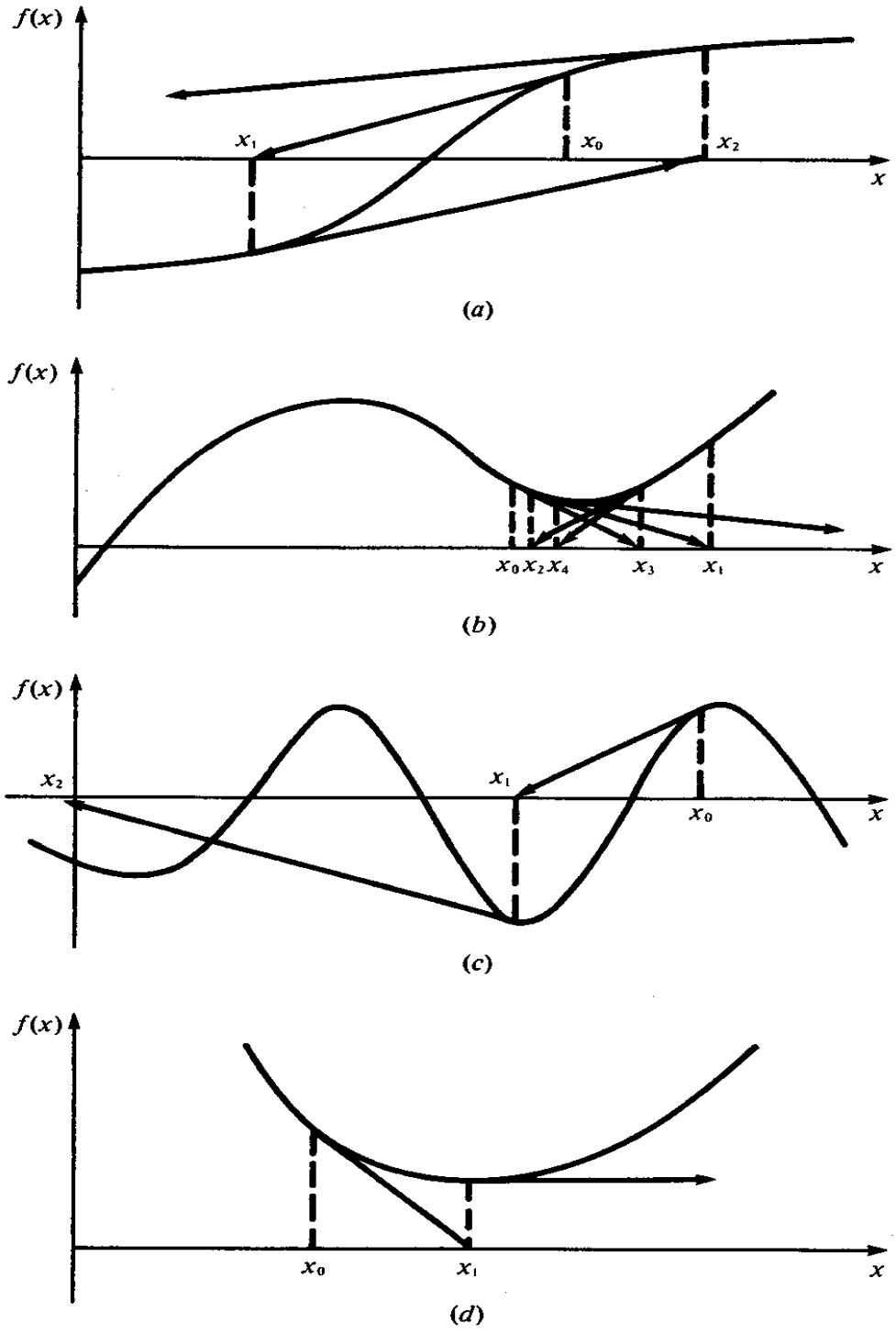
Il miglioramento della stima è dovuto al fatto che al crescere di i, $\xi \rightarrow x_t$

Programma Matlab per il metodo di Newton-Raphson

```
function [zero,niter]=newton1D(f,df,x0,toll,nmax)
% Metodo di Newton-Raphson scalare
x=x0;
fx=eval(f);dfx=eval(df);
niter=0; diff=toll+1;
while diff>=toll&niter<=nmax
niter=niter+1;diff=-fx/dfx;
x=x+diff;
diff=abs(diff);
fx=eval(f); dfx=eval(df);
end; zero=x;
return
```

Punti deboli del metodo di Newton

Il metodo di Newton, pur essendo molto efficiente, può talvolta fornire dei risultati molto scadenti, ad esempio nel caso di radici multiple.
Anche nel caso di radici singole il metodo può fallire. Alcuni esempi:



Punto di flesso vicino radice
$f''(x_f)=0$, $x_f \sim x_t$

Tendenza ad oscillare attorno a un estremo locale.
$f'(x_m)=0$, $x_m \sim x_i$

Nelle zone a pendenza ridotta ci si allontana dalla radice $f'(x_i) \sim 0$

Fallisce se una stima della radice è un estremo locale $f'(x_i)=0$

Figura 5.6 Quattro casi in cui il metodo di Newton-Raphson non converge

Metodo delle secanti

Il **metodo di Newton** presenta l'inconveniente di richiedere la **conoscenza della derivata analitica** della funzione.
Nel caso in cui la derivata non sia disponibile (ad esempio è complicata da calcolare) può essere approssimata attraverso il rapporto incrementale:

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

Sostituendo questa espressione nella formula di Newton si perviene alla formula del *metodo delle secanti* :

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

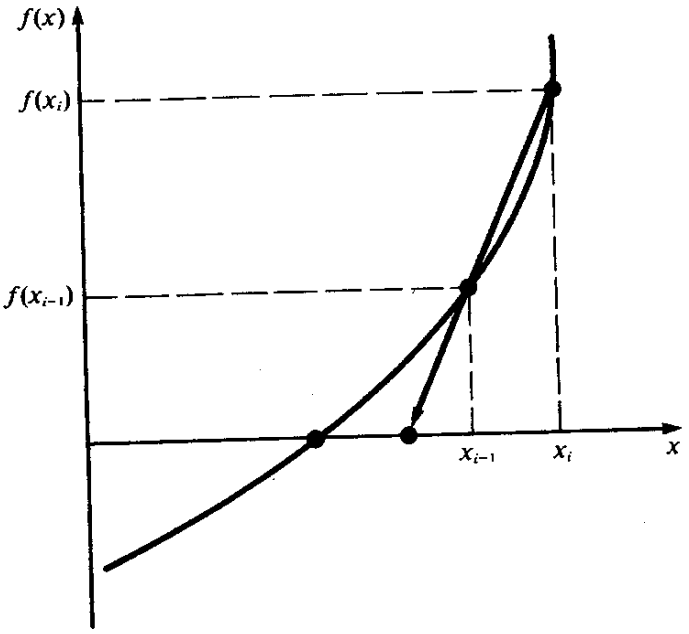


Figura 5.7 Rappresentazione grafica del metodo delle secanti. Questa tecnica è simile al metodo di Newton-Raphson (Figura 5.5) in quanto si ottiene una stima della radice come intersezione di una tangente con l'asse x. Il metodo delle secanti, però, utilizza un rapporto incrementale al posto della derivata per stimare la pendenza della tangente.

In questo approccio vengono richieste due stime iniziali della radice, ma la radice non deve essere necessariamente compresa tra le stime (ovvero non si richiede un cambiamento di segno). Il metodo è pertanto **aperto**

Esempio: usare il metodo delle secanti per determinare la radice *di* $f(x)=e^{-x}-x$ (valore esatto 0.567143290409784). Si assuma $x_{-1}=0$ *e* $x_0=1$

It	x_i	$f(x_i)$	ε_t	ε_a
-1	0	1.000000		
0	1	-6.3212E-1		
1	0.61270	-7.081E-2	8.0%	63.2%
2	0.56384	5.18E-3	5.8E-1%	8.67%
3	0.56717	-4.19e-005	4.8E-3%	5.9e-1%

Programma Matlab per il metodo delle Secanti

```
function [zero,niter]=secanti(f,x0,xm1,toll,nmax)
% Metodo di Newton-Raphson scalare
x=xm1; fxp=eval(f);
x=x0; fx=eval(f);
dfx=(fx-fxp)/(x0-xm1);
niter=0; diff=toll+1;
while diff>=toll&niter<=nmax
niter=niter+1;diff=-fx/dfx;
fxp=fx; xm1=x;
x=x+diff;
diff=abs(diff);
fx=eval(f); dfx=(fx-fxp)/(x-xm1);
end;
zero=x;
return
```

DIFFERENZA TRA IL METODO DELLE SECANTI E QUELLO DELLA FALSA POSIZIONE

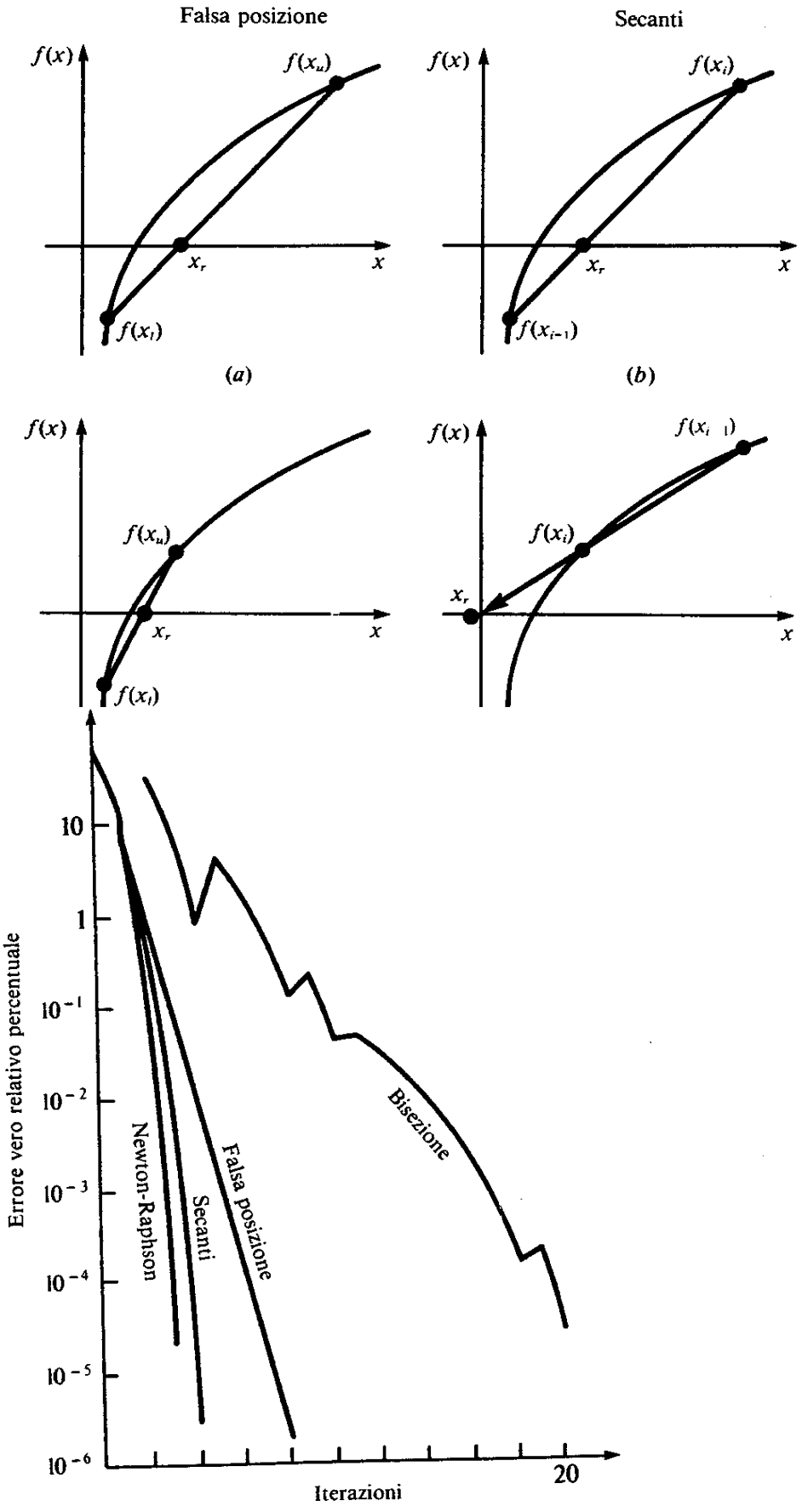
I metodi delle secanti e della falsa posizione si somigliano molto:

Entrambi infatti utilizzano due stime di partenza per calcolare un'approssimazione della pendenza della funzione che viene utilizzata per la nuova stima della radice.
Esiste però una differenza fondamentale nella strategia con cui una delle stime iniziali viene sostituita con una nuova stima:

Nel metodo della falsa posizione la nuova stima prende il posto della vecchia che dava luogo ad un valore della funzione di eguale segno.
Per questo le due stime continuano a racchiudere la radice ed il metodo converge sempre.

Nel metodo delle secanti, le stime vengono aggiornate in modo strettamente sequenziale: $x_i=x_{i+1}$; $x_{i-1}=x_i$. Pertanto le due stime possono trovarsi entrambe a destra o a sinistra della radice. Questa circostanza può provocare talvolta la divergenza del metodo.

Va tuttavia aggiunto che se il metodo delle secanti converge, la sua velocità di convergenza è del secondo ordine, superiore a quella del metodo della falsa posizione.



rima ite-
d), però,
gere, co-

Figura 5.9 Confronto tra gli errori veri relativi percentuali ϵ_t dei vari metodi nella determinazione della radice di $f(x) = e^{-x} - x$.

Il problema delle radici multiple

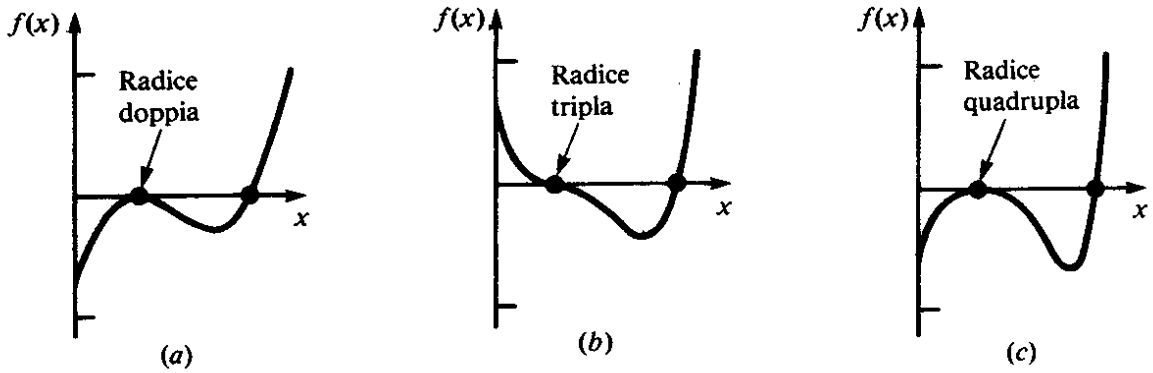


Figura 5.10 Esempi di radici multiple (funzione tangente all'asse *x*). Osservate che la funzione non attraversa l'asse in corrispondenza di radici con molteplicità pari (*a*) e (*c*), mentre lo attraversa se la molteplicità della radice è dispari (*b*).

Una **radice multipla** corrisponde ad un punto in cui oltre che la funzione si annullano un certo numero di derivate. Dal punto di vista geometrico si ha la tangenza della funzione con l'asse delle ascisse.
Nel caso di **radici con grado di molteplicità dispari** l'asse delle ascisse viene attraversato, mentre se la molteplicità è pari la funzione non cambia di segno. In questo ultimo caso **non è possibile utilizzare metodi chiusi** per determinare il valore della radice.

Nel metodo di Newton come in quello delle secanti si richiede la determinazione della derivata prima f'(x) per il calcolo della successiva stima di f(x). Nel caso di radici multiple, l'annullarsi della derivata prima implica una divisione per un numero piccolo (al limite zero) quando la stima tende al valore vero della radice.

Tuttavia (Ralston e Rabinowitz, 1978) sfruttando il fatto che la *f(x)* tende a zero più rapidamente della *f'(x)*, con un controllo sul valore assunto da *f(x)* è possibile arrestare il calcolo prima che *f'(x)* raggiunga lo zero.

Si noti che nel caso di radici multiple tanto il metodo delle secanti quanto quello di Newton presentano una velocità di convergenza lineare.

Tuttavia se si usa la variante $x_{i+1} = x_i - m\dfrac{f(x_i)}{f'(x_i)}$ (1)

dove *m* è la molteplicità della radice, la velocità di convergenza torna quadratica.

In alternativa definendo: $u(x) = \dfrac{f(x)}{f'(x)}$ si ottiene una funzione che ha le stesse radici di f(x). Sostituendo nella formula di Newton si ottiene:

$$x_{i+1} = x_i - \frac{u(x_i)}{u'(x_i)} \qquad \text{con} \qquad u'(x) = \frac{f'(x)f'(x) - f(x)f''(x)}{\left[f'(x)\right]^2}$$

e sostituendo: $$x_{i+1} = x_i - \frac{f(x_i)f'(x_i)}{\left[f'(x_i)\right]^2 - f(x_i)f''(x_i)} \qquad (2)$$

Esempio: Trovare le radici multiple dell'equazione $f(x) = x^3-5x^2+7x-3 = (x-1)^2(x-3)$

Usando la formula di Newton standard $$x_{i+1} = x_i - \frac{x_i^3 - 5x_i^2 + 7x_i - 3}{3x_i^2 - 10x_i + 7}$$

si ha una convergenza lineare verso la soluzione (x_0=0). Invece utilizzando la formula modificata (2) la convergenza è, come previsto, quadratica. Come si vede dalla tabella, nella ricerca della radice singola il metodo standard risulta superiore.

I	x_i	$\lvert\varepsilon_t\rvert$%	$\lvert\varepsilon_a\rvert$%
0	0	100	
1	0.428571429	57	100
2	0.685714286	31	37.5
3	0.832865400	17	17.67
4	0.913328983	8.7	8.81
5	0.955783293	4.4	4.44
6	0.977655101	2.2	2.24

Formula standard (radice doppia)

I	x_i	$\lvert\varepsilon_t\rvert$%	$\lvert\varepsilon_a\rvert$%
0	0	100	
1	0.428571429	57	100
2	0.685714286	31	37.5
3	0.832865400	17	17.67
4	0.913328983	8.7	8.81
5	0.955783293	4.4	4.44

Formula modif. (radice doppia)

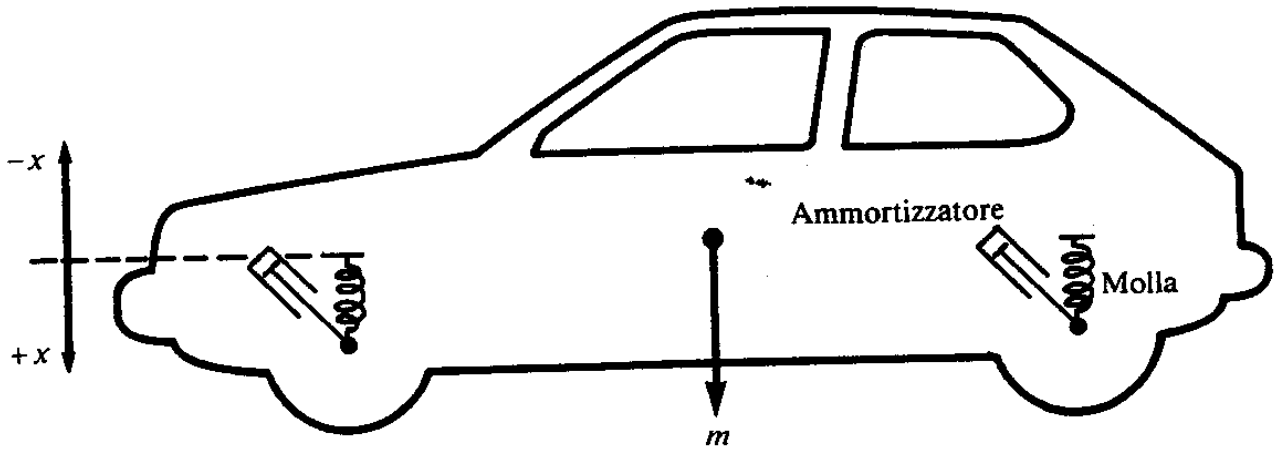
I	x_i	$\lvert\varepsilon_t\rvert$%
0	4	33
1	3.4	13
2	3.1	3.3
3	3.008695652	2.9e-1
4	3.00074641	2.5E-3
5	3.000000006	2E-7

Formula stand. (radice semplice)

I	x_i	$\lvert\varepsilon_t\rvert$%
0	4	33
1	2.636363637	12
2	2.8229224729	6.0
3	2.961728211	1.3
4	2.998478719	5.1e-2
5	2.999997682	7.7e-5

Formula modif. (radice semplice)

Applicazione pratica: Analisi delle vibrazioni



a 6.9 Automobile di massa m.

Un'automobile di massa m è sostenuta da molle di costante elastica k. Vengono utilizzati degli ammortizzatori di coefficiente di smorzamento c per smorzare i moti verticali del veicolo, causati dalla presenza di buche.

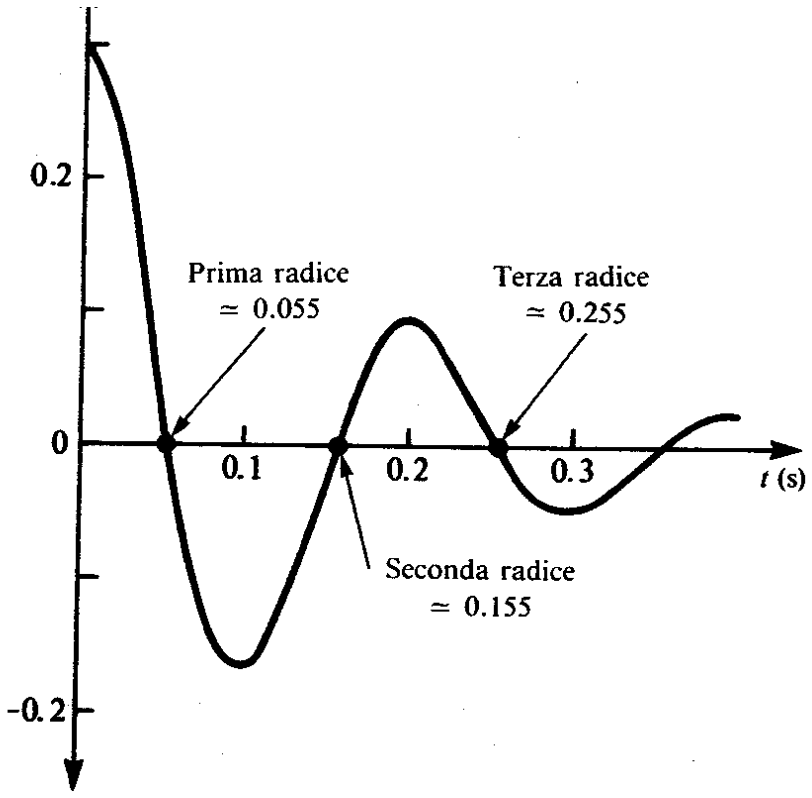


Figura 6.10 Grafico della posizione di un ammortizzatore in funzione del tempo dopo che la ruota ha incontrato una buca.

Il modello matematico del sistema e la legge oraria dello spostamento sono

$$\frac{d^2x}{dt^2}+\frac{c}{m}\frac{dx}{dt}+\frac{k}{m}x=0 \qquad \rightarrow\ x(t)=\mathrm{x}_0 e^{-m}\left(\cos(pt)+\frac{n}{p}\sin(pt)\right)$$

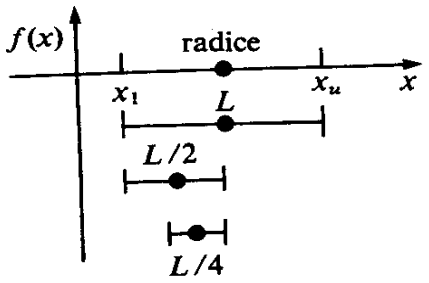
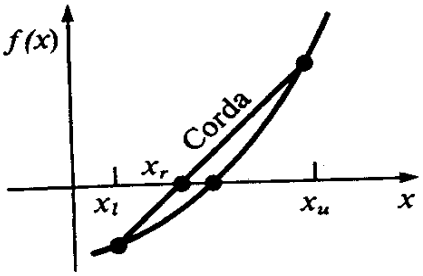
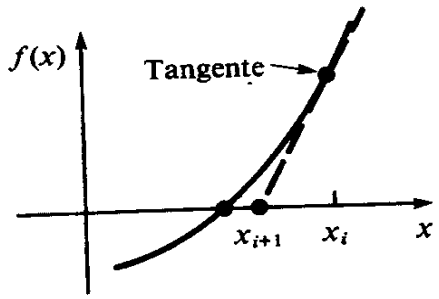
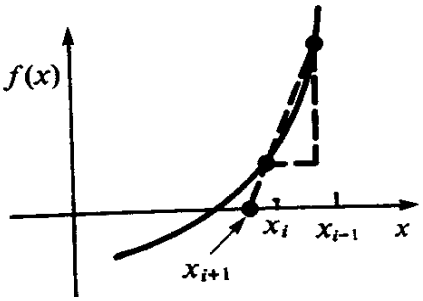
dove $n=c/(2m), p=\sqrt{k/m-c^2/(4m^2)}$. Si assumano i seguenti parametri: $k=1.25\cdot10^6 kg/s^2$; $c=1.4\cdot10^4\ kg/s$, $m=1.2\cdot10^3 kg$,

Risultati:

Dalla figura si evince che vi sono tre radici. Dai risultati si vede che i **metodi della falsa posizione e delle secanti** sono più efficienti rispetto al **metodo di bisezione.** Per tutti i metodi l'errore approssimato risulta maggiore rispetto all'errore vero. Come già fatto notare nei metodi a convergenza rapida l'errore stimato è molto conservativo rispetto all'errore vero. Se vi sono molte radici da determinare è importante utilizzare un metodo con alta velocità di convergenza.

Prospetto riassuntivo

Tabella II.4 Riepilogo delle nozioni di particolare importanza presentate nella Parte Seconda.

Metodo Formule	Interpretazione grafica	Errori e criteri di terminazione
Bisezione $x_r = \frac{x_l + x_u}{2}$ se $f(x_l)f(x_r) < 0$, $x_u = x_r$ se $f(x_l)f(x_r) > 0$, $x_l = x_r$	*Metodi chiusi:* 	Criterio di terminazione: $\left\|\frac{x_r^{att} - x_r^{prec}}{x_r^{att}}\right\| 100\% \le \epsilon_s$
Falsa posizione $x_r = x_u - \frac{f(x_u)(x_l - x_u)}{f(x_l) - f(x_u)}$ se $f(x_l)f(x_r) < 0$, $x_u = x_r$ $f(x_l)f(x_r) > 0$, $x_l = x_r$		Criterio di terminazione: $\left\|\frac{x_r^{att} - x_r^{prec}}{x_r^{att}}\right\| 100\% \le \epsilon_s$
	Metodi aperti:	
Newton-Raphson $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$		Criterio di terminazione: $\left\|\frac{x_{i+1} - x_i}{x_{i+1}}\right\| 100\% \le \epsilon_s$ Errore: $E_{i+1} = O(E_i^2)$
Secanti $x_{i+1} = x_i - \frac{f(x_i)(x_{i-1} - x_i)}{f(x_{i-1}) - f(x_i)}$		Criterio di terminazione: $\left\|\frac{x_{i+1} - x_i}{x_{i+1}}\right\| 100\% \le \epsilon_s$

I metodi per la ricerca delle radici di un'equazione si dividono in aperti e chiusi.

Nei **metodi chiusi** (Bisezione, Falsa posizione), la soluzione deve essere compresa tra due stime iniziali. La funzione valutata nelle stime iniziali deve assumere segno opposto.
I metodi chiusi sono sempre convergenti.

Nei **metodi aperti** (Newton-Raphson, Secanti) utilizzano una sola stima della radice (oppure due che non devono necessariamente racchiuderla).
La convergenza non è garantita, ma se c'è la velocità di convergenza (quadratica) risulta più elevata rispetto a quella dei metodi aperti.

Il metodo di **Newton-Raphson** ha una rapidità di convergenza superiore rispetto a quello delle **secanti** nelle prime iterazioni. Tuttavia presenta l'inconveniente di richiedere la valutazione analitica della derivata funzione.

Il metodo di **Newton-Raphson** modificato permette di ottenere una convergenza rapida nel caso di radici multiple, ma richiede la conoscenza analitica della derivata prima e seconda della funzione.

È possibile utilizzare i punti di forza di entrambi i metodi nella seguente maniera:

1. Si utilizza un metodo chiuso per avere una buona stima iniziale della radice.
2. Si passa questa stima ad un metodo aperto per avere rapidamente una soluzione precisa.

MATRICI

Si definisce matrice una tabella rettangolare di elementi rappresentati da un unico simbolo:

$\underline{\underline{\mathbf{A}}}$ è una matrice $\in \Re^M x \Re^N$ (M righe x N colonne)

$$\underline{\underline{\mathbf{A}}} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}$$

il simbolo a_{ij} individua l'elemento posto nella i^{-ma} riga e nella j^{-ma} colonna.

Una matrice con M=1 viene detto vettore colonna.

$$\underline{\mathbf{b}} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

Una matrice con N=1 viene detto vettore riga.

$$\underline{\mathbf{c}} = \begin{bmatrix} c_1 & c_2 & \cdots & c_n \end{bmatrix}$$

Le matrici con M=N vengono dette **quadrate**

Nelle matrici quadrate, la diagonale che contiene gli elementi a_{ii}, i=1…N viene detta **diagonale principale** della matrice.

CLASSIFICAZIONE DELLE MATRICI

Si riportano alcune definizioni adoperate nella classificazione delle matrici:

- Una matrice è detta **reale** se tutti i suoi elementi appartengono al dominio dei numeri reali.
- Una matrice **A** quadrata di ordine N ad elementi reali tale che $\mathbf{A}=\mathbf{A}^t$ è detta **simmetrica.**
- Una matrice **A** quadrata di ordine N ad elementi complessi tale che $\mathbf{A}=\mathbf{A}^*$ è detta **hermitiana.**
- Una matrice reale **A** di ordine N tale che per ogni $\mathbf{x}\varepsilon\mathbf{R}^n$ diverso dall'elemento nullo si abbia $\mathbf{x}^t\mathbf{A}\mathbf{x}>0$ viene detta **definita positiva.**
- Una matrice reale **A** quadrata di ordine N e di elementi $a_{i,j}$ tale che si abbia:

$$a_{ii} \geq \sum_{j=1,j\neq i}^{N} |a_{ij}|$$

$\forall i\varepsilon[1,N]$ è detta **a dominanza diagonale**

- Una matrice quadrata **D** è detta **diagonale** se tutti i suoi elementi che non si trovano sulla diagonale principale sono nulli.
- Una matrice quadrata **U** è detta **triangolare superiore** se tutti i suoi elementi che si trovano sotto la diagonale principale sono nulli.
- Una matrice quadrata **L** è detta **triangolare inferiore** se tutti i suoi elementi che giacciono sopra la diagonale principale sono nulli.
- Si dimostra che se **A** è una matrice quadrata simmetrica è sempre possibile trovare una matrice diagonale **D** e una matrice triangolare superiore **U** tale che $\mathbf{A}=\mathbf{U}^T\mathbf{D}\mathbf{U}$ (**Fattorizzazione** di **A**).
- Una matrice in cui prevalgono termini diversi da zero viene chiamata **piena**, mentre una in cui dominano i termini nulli si classifica come **sparsa**.
- Le matrici nelle quali tutti gli elementi sono nulli tranne quelli contenuti in una fascia centrata sulla diagonale principale vengono dette **a banda**
- Esistono diverse tecniche di immagazzinamento dei coefficienti delle matrici di tipo sparso, e la scelta della più adatta al problema in esame viene determinata dalle caratteristiche strutturali di sparsità della matrice.

I differenti approcci di immagazzinamento delle matrici vengono usualmente classificati in letteratura sotto le etichette di **metodi a bande**, **metodi d'inviluppo** e **metodi di sparsità generale**.

Operazioni con le matrici

Analogamente alle operazioni tra scalari vengono definite quelle tra matrici:

- Due matrici MxN **A** e **B** sono dette **uguali** se $a_{ij}=b_{ij} \quad \forall$ i,j
- L'addizione di due matrici **A** e **B** viene eseguita sommando i termini corrispondenti delle matrici. Gli elementi della matrice risultante **C** sono dati da: $c_{ij}=a_{ij}+b_{ij} \ \forall$ i=1,2,…m; j=1,2,…n.
- La differenza di due matrici **A** e **B** viene eseguita sottraendo i termini della seconda da quelli corrispondenti della prima. Gli elementi della matrice risultante **C** sono dati da:

$$c_{ij}=a_{ij}-b_{ij} \ \forall \ i=1,2,\ldots m; \ j=1,2,\ldots n.$$

L'addizione e la sottrazione

- possono essere eseguite solo su matrici delle stesse dimensioni.
- sono operazioni commutative: **A + B = B +A; A - B = - B +A;**
- sono operazioni associative: **A + (B + C) =(A + B)+ C;**

- La moltiplicazione di una matrice **A** per uno scalare *g* avviene moltiplicando ogni elemento di **A** per *g* : $b_{ij}= g\, a_{ij}$.
- Il prodotto di due matrici **A** (MxL), **B** (LxN) viene rappresentato da una matrice **C** =**A** · **B** di dimensioni (MxN) dove gli elementi di **C**

sono definiti come: $$c_{ij} = \sum_{k=1}^{L} a_{ik} b_{kj}; \qquad i = 1,..., M; \qquad j = 1,..., N;$$

- Se le dimensioni delle matrici sono compatibili, la moltiplicazione è **associativa A(B C) =(A B)C** e **distributiva A (B + C)=A B+A C; ma non è commutativa: A B ≠ B A**
- La divisione tra matrici non è un'operazione definita; tuttavia se **A** è quadrata, allora esiste una matrice $\mathbf{A^{-1}}$ detta **inversa** di **A** per la quale vale la relazione: $\mathbf{A^{-1}A} = \mathbf{A\ A^{-1}} = \mathbf{I}$ dove la matrice **I**, detta matrice unitaria è una matrice quadrata, diagonale, con coefficienti tutti pari ad 1.
- L'operazione di trasposizione consiste nel sostituire le righe con le colonne e viceversa: $(a_{ij})^T=(a_{ji})$. Tramite la trasposizione, un vettore riga **x** si trasforma in un vettore colonna $\mathbf{x^T}$ e viceversa.

Metodi di immagazzinamento delle matrici

Metodi di banda

I metodi di banda sono adeguati per il trattamento di matrici, caratterizzate dalla concentrazione dei termini non nulli entro uno stretto numero di diagonali adiacenti la diagonale principale. Il parametro caratteristico, m_A, associato alla matrice *bandata* **A** viene chiamato *ampiezza di banda*, ed è definito come il più piccolo intero tale che $a_{i,j}=0$ per $|i-j|>m_A$. Una importante proprietà delle matrici a bande è che la fattorizzazione $\mathbf{A}=\mathbf{U}^T\mathbf{D}\mathbf{U}$, non comporta allargamento della banda per la matrice triangolare superiore: per questo particolare processo di fattorizzazione risulta cioè $m_A=m_U$. Sistemi di matrici bandate provengono, ad esempio, da problemi agli elementi finiti monodimensionali, o anche bidimensionali ma definiti in domini geometricamente molto regolare.

Per tutti i tipi di matrici sparse sono state elaborate efficaci strategie per ridurre al minimo la quantità di memoria necessaria per l'immagazzinamento dei coefficienti. In particolare nel caso di matrici bandate simmetriche si usa immagazzinare le diagonali che contengono termini non nulli in un array bidimensionale. La diagonale principale riempie la prima riga dell'array, quella immediatamente sopra viene immagazzinata nella seconda riga, e così di seguito finché l'intera banda collocata nel triangolo superiore delle matrice di partenza non ha trovato posto nell'array. Non è necessario per la simmetria della matrice stipare anche i termini del triangolo inferiore. Un esempio è mostrato in figura [1].

$$\mathbf{A}=\begin{bmatrix} a_{11}\,a_{12}\,a_{13}\,a_{14}\,0\;0 \\ a_{12}\,a_{22}\,a_{23}\,a_{24}\,0\;0 \\ a_{13}\,a_{23}\,a_{33}\,a_{34}\,a_{35}\,a_{36} \\ a_{14}\,a_{24}\,a_{34}\,a_{44}\,a_{45}\,a_{46} \\ 0\;0\;\;a_{35}\,a_{45}\,a_{55}\,a_{56} \\ 0\;0\;\;a_{36}\,a_{46}\,a_{56}\,a_{66} \end{bmatrix} \text{(a)} \qquad \begin{bmatrix} a_{11}\;a_{22}\;a_{33}\;a_{44}\;a_{55}\;a_{66} \\ a_{12}\;a_{23}\;a_{34}\;a_{45}\;a_{56}\;* \\ a_{13}\;a_{24}\;a_{35}\;a_{46}\;*\;* \\ a_{14}\;0\;a_{36}\;*\;*\;* \end{bmatrix} \text{(b)}$$

Figura [1] (a) Matrice bandata **A** con $m_A=3$;

(b) Schema di immagazzinamento per A con un array bidimensionale.

(* = Elementi dell'array di immagazzinamento non adoperati)

Adoperando il criterio sopra illustrato lo spazio di immagazzinamento di una matrice bandata di ordine N e di ampiezza di banda m_A passa da N^2 a $Nx(1+m_A)$ locazioni di memoria che includono anche $m_A\ (1+m_A)/2$ posizioni dell'array che non vengono adoperate. Queste locazioni vengono tuttavia mantenute allo scopo di facilitare l'accesso all'intera struttura, e, in ogni caso non occupano porzioni di memoria

significative, essendo m_A<<N. Si noti infine che le righe del triangolo superiore di **A** coincidono con le colonne dell'array di immagazzinamento. Questo torna molto utile nei linguaggi come il Fortran che immagazzinano gli array per colonne, in quanto la corrispondenza dello schema di immagazzinamento al linguaggio di programmazione riduce al minimo la difficoltà di accedere alle righe di **A** con un complessivo aumento delle prestazioni dell'algoritmo in termini di velocità.

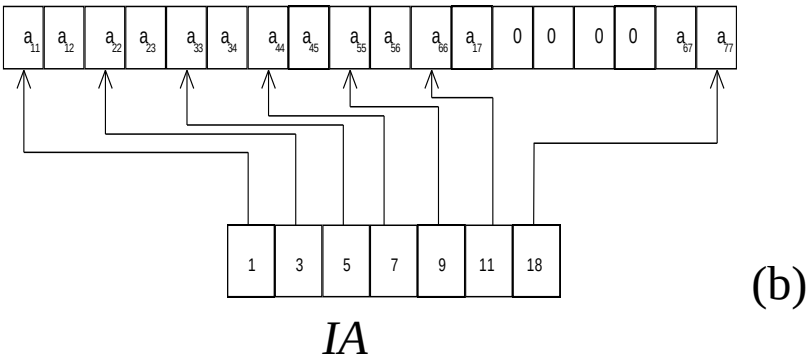
Metodi di inviluppo

Un problema FEM bidimensionale, con geometria del dominio di definizione abbastanza complessa, conduce ad un moderato grado di variazione locale dell'ampiezza di banda della matrice. Attraverso una opportuna renumerazione degli elementi geometrici (nodi o lati) cui sono associate le incognite del problema è possibile ottenere una matrice nella quale l'inviluppo degli elementi non nulli ha una forma lenticolare. In questo caso, sebbene l'ampiezza di banda della matrice resti uguale a m_A, fatta eccezione per la riga centrale N/2, gli elementi non nulli si trovano generalmente ben all'interno della banda di ampiezza m_A. Detta **A** la matrice, che supponiamo simmetrica, ed **U** la matrice triangolare superiore ottenuta dalla fattorizzazione di **A,** l'analisi del processo di fattorizzazione mostra che $u_{ij}=0$ se $i<f_j \equiv \min(\{k:a_{kj}\neq 0\})$; cioè, per ciascuna colonna gli elementi di **U** diversi da zero sono localizzati tra il primo elemento non nullo di **A** sulla stessa colonna e la diagonale principale. Gli elementi di **A** appartenenti all'insieme $Env(\mathbf{A})=\{ a_{ij}:f_j<i<j\}$ costituiscono *l'inviluppo* di **A.**

La struttura di immagazzinamento dell'inviluppo di A è più complessa, ma, in certi casi, più efficiente rispetto a quella esaminata precedentemente che conservava l'intera banda della matrice. Gli elementi vengono immagazzinati per colonne in un array monodimensionale che denoteremo con *A*. In questo schema, risulta necessario definire un ulteriore array *IA*, i cui elementi sono dei puntatori ai termini che formano la diagonale principale di A. Un esempio della struttura è mostrato in figura [2].

$$\mathbf{A}=\begin{bmatrix} a_{11} & a_{12} & 0 & 0 & 0 & 0 & a_{17} \\ a_{12} & a_{22} & a_{23} & 0 & 0 & 0 & 0 \\ 0 & a_{23} & a_{33} & a_{34} & 0 & 0 & 0 \\ 0 & 0 & a_{34} & a_{44} & a_{45} & 0 & 0 \\ 0 & 0 & 0 & a_{45} & a_{55} & a_{56} & 0 \\ 0 & 0 & 0 & 0 & a_{56} & a_{66} & a_{67} \\ a_{17} & 0 & 0 & 0 & 0 & a_{67} & a_{77} \end{bmatrix} \quad \text{(a)}$$

A



(b)

Figura [2]: (a) Matrice **A**
(b) Struttura di immagazzinamento di inviluppo per **A**

Con questo schema di immagazzinamento è semplice calcolare f_j come:

$$f_j = \begin{cases} 1 & \text{se } j = 1 \\ j + 1 - (IA(j) - IA(j-1)) & \text{se } j \neq 1 \end{cases}$$

e da qui accedere agli elementi di ogni colonna della matrice.

La struttura di inviluppo risulta più efficiente rispetto a quella di banda in vari casi. Un esempio è costituito dai problemi bidimensionali in cui si sia provveduto a numerare i nodi della mesh lungo le diagonali (Vedi Fig. [3]). In tal caso si può osservare come lo schema di immagazzinamento di inviluppo consenta un risparmio del 33% in termini di locazioni di memoria e del 50% nel tempo di fattorizzazione.

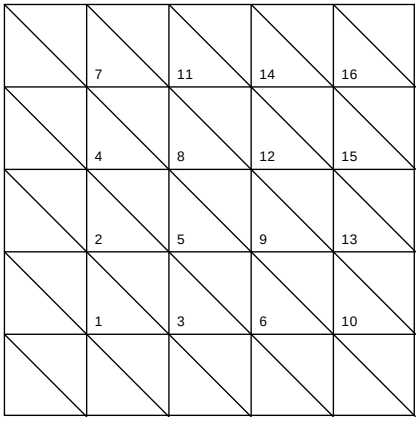


Figura [3]: Ordinamento diagonale dei nodi

Metodi di sparsità generale

I metodi a banda e d'inviluppo, mentre risultano vantaggiosi nel trattamento di matrici che presentano blocchi strutturali compatti (sia cioè possibile partizionare la matrice in zone in cui si trovano soltanto elementi nulli e in zone che presentano un'alta densità di elementi diversi da zero), si dimostrano totalmente insoddisfacenti in tutti i problemi che danno origine a matrici la cui sparsità è di tipo non strutturale: ciò è vero,

in particolare, per una vasta classe di problemi FEM bi- e tri-dimensionali. Si preferiscono utilizzare in questi casi dei metodi che a prezzo di una maggiore complessità della struttura di immagazzinamento dei coefficienti garantiscono una significativa riduzione della quantità di dati da conservare.
Sia **A** una matrice simmetrica di ordine N che presenta caratteristiche di sparsità non strutturali. Per la simmetria sarà possibile immagazzinare solo i coefficienti che si trovano sulla diagonale **D** oppure nel blocco triangolare inferiore **E** di **A** Per l'immagazzinamento dei coefficienti si utilizzano tre vettori monodimensionali *A*, *IA* e *JA.*
A è un vettore di reali che contiene, ordinati per righe, tutti gli elementi di **E** e **D** diversi da zero.
JA è un vettore di interi che contiene l'indice di colonna degli elementi corrispondenti immagazzinati in *A*
IA contiene, infine, i puntatori agli elementi diagonali immagazzinati in *A.*

Se il numero di elementi non nulli di **E** + **D** è *nsize*, allora la dimensione di *A* e *JA* sarà *nsize*, mentre quella di *IA* sarà *N*.
Gli elementi a_{ij} con j≤i dell' i-ma riga di **A** sono immagazzinati nelle posizioni di *A* che vanno dall'indice *IA(i-1)+1* all'indice *IA(i)*
In figura [4] è mostrato come possa essere immagazzinata attraverso questo schema la matrice di figura [.a]:

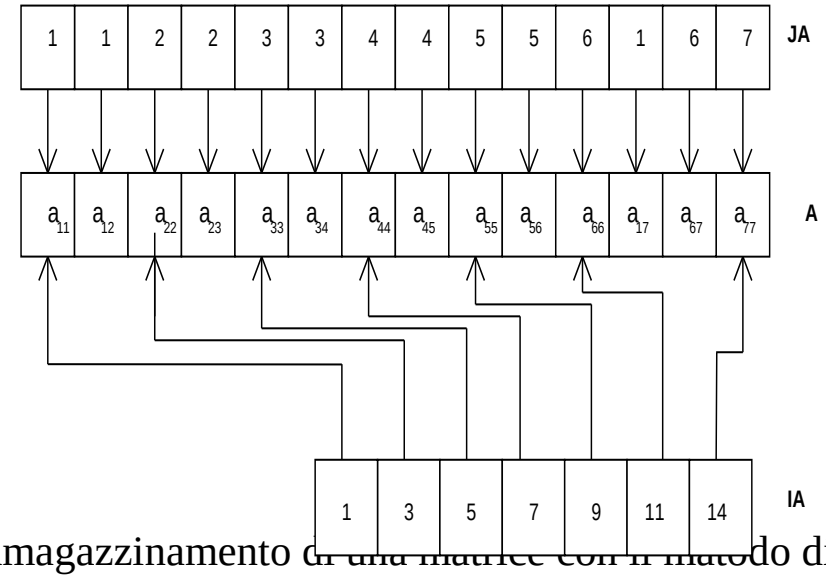


Fig.[4] Immagazzinamento di una matrice con il metodo di sparsità generale

Si nota immediatamente che questo schema non permette né un accesso random, né un accesso per colonne agli elementi di **A**.
Va infine notato che se -come spesso succede- bisogna risolvere diversi sistemi lineari ai quali sono associate matrici sparse della medesima struttura ed eventualmente con diversi coefficienti, allora le sole informazioni contenute in *A* debbono essere

aggiornate mentre i vettori *JA* e *IA* si mantengono costanti e possono pertanto essere calcolati preventivamente al processo solutivo. Ovviamente se lo schema risolutivo che si intende adottare prevede la fattorizzazione della matrice **A**, non si può prescindere nella generazione di *JA* e *IA* dal portare in conto anche gli elementi che pur nulli per **A** non lo saranno a causa del fill-in nella matrice **U** che viene generata dalla fattorizzazione: come si era precedentemente anticipato, in questi casi la fattorizzazione simbolica diventa indispensabile.

SISTEMI DI EQUAZIONI LINEARI ALGEBRICHE

Si è visto nelle lezioni precedenti come determinare il valore di x che soddisfa una singola equazione *f(x)=0*

Si vuole ora determinare una N^{pla} di valori x_j, j=1..N che soddisfi simultaneamente un sistema di M equazioni

$$f_i(x_1,\ldots,x_{j,\ \ldots,}x_n)=0; \qquad i=1\ldots M \qquad (1)$$

Particolarmente importante è il caso in cui le M equazioni sono lineari algebriche, cioè della forma:

$$\begin{aligned}
&a_{11}\,x_1 + a_{12}\,x_2 + \ldots a_{1j}\,x_j + \ldots a_{1N}\,x_N = c_1\\
&a_{21}\,x_1 + a_{22}\,x_2 + \ldots a_{2j}\,x_j + \ldots a_{2N}\,x_N = c_2\\
&\quad\vdots \qquad\vdots \qquad\qquad \vdots\\
&a_{i1}\,x_1 + a_{i2}\,x_2 + \ldots a_{ij}\,x_j + \ldots a_{iN}\,x_N = c_i \qquad (2)\\
&\quad\vdots \qquad\vdots \qquad\qquad \vdots\\
&a_{M1}\,x_1 + a_{M2}\,x_2 + \ldots a_{Mj}\,x_j + \ldots a_{MN}\,x_N = c_M
\end{aligned}$$

dove gli *MxN* coefficienti a_{ij} e gli M coefficienti c_i sono costanti note.

Dedicheremo le prossime lezioni a presentare diversi metodi per la soluzione per via numerica dei sistemi di equazioni lineari algebriche.

Infine, tratteremo brevemente anche il problema della soluzione di sistemi di equazioni non-lineari del tipo (1)

INTRODUZIONE AL FORMALISMO MATRICIALE

Il sistema di equazioni (2) può anche essere rappresentato in forma compatta utilizzando la notazione matriciale: $\underline{\underline{\mathbf{A}}}\,\underline{\mathbf{x}}^{\mathbf{T}}=\underline{\mathbf{c}}^{\mathbf{T}}$; dove

$\underline{\underline{\mathbf{A}}}$ è una matrice quadrata $\in \Re^N x \Re^N$ (N righe x N colonne) che contiene i coefficienti del sistema di equazioni

il vettore riga $\underline{\mathbf{c}} \in \Re^N$ è il vettore dei termini noti

il vettore riga $\underline{\mathbf{x}} \in \Re^N$ è il vettore delle incognite

$$\underline{\underline{\mathbf{A}}} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}; \qquad \underline{\mathbf{x}}^{\mathrm{T}} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \qquad \underline{\mathbf{c}}^{\mathrm{T}} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}$$

Le matrici quadrate hanno particolare importanza nella risoluzione dei sistemi lineari: sono, infatti, associate a sistemi lineari nei quali il numero delle incognite coincide col numero di equazioni. Questa situazione corrisponde ad una condizione necessaria (ma non sufficiente) affinchè il sistema ammetta soluzione unica.

$$\underline{\underline{\mathbf{A}}}\,\underline{\mathbf{x}}^{\mathbf{T}}=\underline{\mathbf{c}}^{\mathbf{T}} \rightarrow \underline{\underline{\mathbf{A}}}^{-1}\underline{\underline{\mathbf{A}}}\,\mathbf{x}^{\mathbf{T}}=\underline{\underline{\mathbf{A}}}^{-1}\underline{\mathbf{c}}^{\mathbf{T}} \rightarrow \underline{\mathbf{I}}\,\mathbf{x}^{\mathbf{T}}=\underline{\underline{\mathbf{A}}}^{-1}\underline{\mathbf{c}}^{\mathbf{T}} \rightarrow \mathbf{x}^{\mathbf{T}}=\underline{\underline{\mathbf{A}}}^{-1}\underline{\mathbf{c}}^{\mathbf{T}}$$

L'equazione è quindi risolta esplicitamente rispetto al vettore delle incognite **x**.

Si vede come la matrice inversa gioca nell'algebra delle matrici un ruolo simile a quello della divisione.

Combinando $\underline{\underline{\mathbf{A}}}$ e **c**, si ottiene la cosiddetta **matrice aggiunta**. La rappresentazione è utile perché in diversi algoritmi si eseguono le stesse operazioni su ogni riga di coefficienti di $\underline{\underline{\mathbf{A}}}$ e sul termine noto. Per N=3:

$$\left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & c_1 \\ a_{21} & a_{22} & a_{23} & c_2 \\ a_{31} & a_{32} & a_{33} & c_3 \end{array}\right]$$

I sistemi di equazioni lineari algebriche nella pratica ingegneristica

La risoluzione dei sistemi lineare riveste un ruolo basilare nella matematica applicata. Mentre, infatti, diversi modelli matematici significativi nascono direttamente lineari, attraverso i metodi numerici, anche problemi di dimensione infinita (ad esempio quelli modellati attraverso equazioni integro-differenziali non lineari), siano riconducibili a sistemi non lineari di dimensione finita che è spesso possibile linearizzare (ovvero risolvere per via iterativa attraverso problemi lineari).

Alla varietà di contesti da cui un sistema lineare può scaturire, corrisponde una uguale differenziazione nella struttura della matrice associata al sistema. Questa, infatti, può essere di volta in volta piena, bandata o sparsa, simmetrica oppure definita positiva.

È importante, dunque, avere a disposizione una varietà di algoritmi efficienti tra i quali scegliere il più adeguato per stabilità, occupazione di memoria, velocità di esecuzione a risolvere su un determinato calcolatore un problema particolare.

In questo parte del corso seguendo l'usuale ripartizione in metodi diretti ed iterativi, si presenteranno alcuni dei principali algoritmi utilizzati nella risoluzione di sistemi lineari *NxN* non-singolari.

Un elemento che bisogna portare in conto nella scelta dello schema risolutivo è **l'onere computazionale** che esso impone. Questo parametro viene comunemente valutato attraverso il numero di moltiplicazioni e divisioni -le operazioni elementari più costose per la macchina calcolatrice- che l'algoritmo richiede per la risoluzione di un sistema di ordine N.

SOLUZIONE DI SISTEMI DI PICCOLE DIMENSIONI (N≤3)

Nella soluzione di sistemi di equazioni lineari di piccole dimensioni (N≤3) ci si può avvalere di diversi metodi. Esamineremo i **metodi grafici**, la **regola di Cramer** e l'**eliminazione delle incognite.**

Metodi grafici

Consideriamo il sistema di due equazioni in due incognite e risolviamo le equazioni in funzione di x_2:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 &= c_1 \\ a_{21}x_1 + a_{22}x_2 &= c_2 \end{aligned} \longrightarrow \begin{aligned} x_2 &= -\left(\frac{a_{11}}{a_{12}}\right)x_1 + \frac{c_1}{a_{12}} \\ x_2 &= -\left(\frac{a_{21}}{a_{22}}\right)x_1 + \frac{c_2}{a_{22}} \end{aligned}$$

come si vede le equazioni hanno la forma canonica per la rappresentazione attraverso due rette:

$$x_2 = (\text{pendenza})^* x_1 + (\text{intercetta}).$$

Disegnando le equazioni sullo stesso grafico, il punto di intersezione fornisce la soluzione del sistema.

Il metodo può essere esteso al caso di un sistema di 3 equazioni in 3 incognite. In questo caso, dal punto di vista geometrico, ciascuna equazione rappresenta un piano cartesiano.

Il metodo non è ovviamente applicabile nel caso in cui N>3.

Tra i vantaggi offerti dal metodo grafico evidenziamo quello di fornire un'immediata interpretazione delle situazioni degeneri. Facendo riferimento al caso di N=2, se le equazioni sono linearmente dipendenti, le rette associate risultano coincidenti e esistono infinite soluzioni. Se i coefficienti angolari sono uguali mentre le intercette differiscono, le rette sono parallele e non esiste intersezione (ovvero soluzione del sistema).
Nel caso in cui le pendenze sono poco differenti, è difficile individuare correttamente il punto di intersezione e il sistema si dice *mal condizionato*. In questo ultimo caso il sistema richiede un trattamento particolare per essere risolto numericamente.

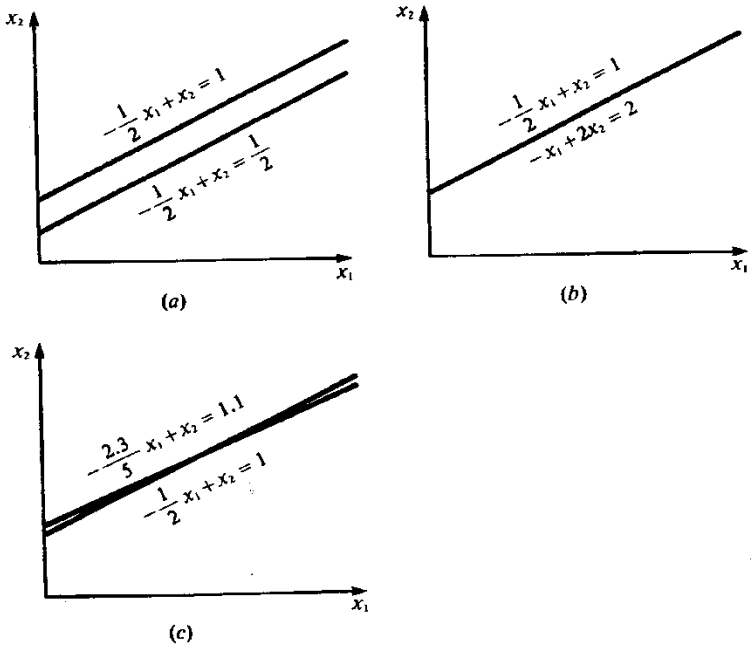


Figura 7.2 Rappresentazione grafica dei sistemi male condizionati: (*a*) nessuna soluzione, (*b*) infinite soluzioni, (*c*) sistema male condizionato in cui le pendenze sono così simili che il punto d'intersezione non può essere individuato visivamente con facilità.

Esempio

$$x_2 = -\left(\frac{3}{2}\right)x_1 + 9$$

$$x_2 = \left(\frac{1}{2}\right)x_1 + 1$$

3 x_1 + 2 x_2 = 18

- x_1 + 2 x_2 = 2

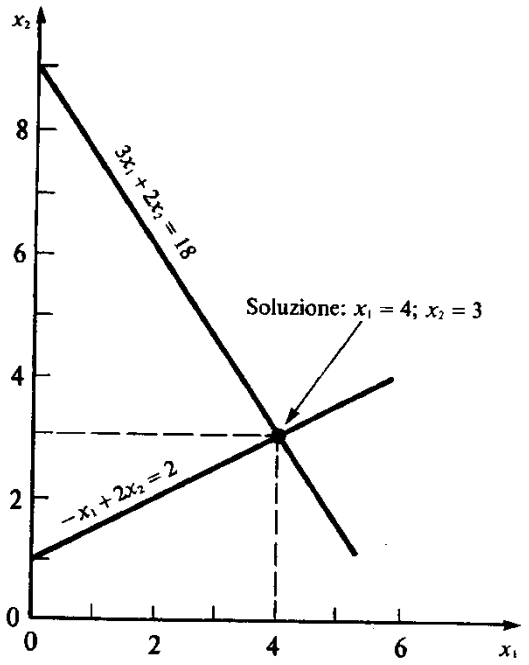


Figura 7.1 Soluzione grafica di un sistema di due equazioni lineari algebriche in due incognite. L'intersezione delle rette corrisponde alla soluzione.

La soluzione è costituita dall'intersezione delle due rette e vale (x_1=4, x_2=3).
Si può verificare la correttezza del risultato sostituendo i valori trovati nelle equazioni originarie.

SOLUZIONE DI SISTEMI DI PICCOLE DIMENSIONI (N≤3)

Concetto di Determinante

Si consideri il sistema di tre equazioni in tre incognite

$a_{11} x_1 + a_{12} x_2 + a_{13} x_3 = c_1$
$a_{21} x_1 + a_{22} x_2 + a_{23} x_3 = c_2$ ⟶ $\underline{\underline{A}}\ \underline{x}=\underline{C}$;
$a_{31} x_1 + a_{32} x_2 + a_{33} x_3 = c_3$

$\underline{\underline{A}}$ = Matrice dei coefficienti
$\underline{x}$ = vettore incognite
$\underline{C}$ = vettore termini noti

Il determinante D($\underline{\underline{A}}$) viene definito a partire dai coefficienti di $\underline{\underline{A}}$ attraverso la notazione simbolica:

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

dove $\begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} = a_{22}\, a_{33} - a_{23}\, a_{32}$ è il determinante associato a una sottomatrice di $\underline{\underline{A}}$ di dimensioni 2x2 (**minore**).

Regola di Cramer

La regola di Cramer stabilisce che il valore di ciascuna incognita del sistema può essere espresso come rapporto di due determinanti:

- al denominatore va il determinante associato alla matrice $\underline{\underline{A}}$.
- al numeratore va il determinante associato alla matrice ottenuta sostituendo la colonna di $\underline{\underline{A}}$ corrispondente all'incognita da calcolare col vettore dei termini noti.

Ad esempio:

$$x_1 = \frac{\begin{vmatrix} c_1 & a_{12} & a_{13} \\ c_2 & a_{22} & a_{23} \\ c_3 & a_{32} & a_{33} \end{vmatrix}}{D(\underline{\underline{A}})}$$

Esempio: Utilizzando la regola di Cramer risolvere il sistema:

$$0.3\ x_1 + 0.52\ x_2 + x_3 = -0.01$$
$$0.5\ x_1 + x_2 + 1.9\ x_3 = 0.67$$
$$0.1\ x_1 + 0.3\ x_2 + 0.5\ x_3 = -0.44$$

Il determinante può essere scritto come

$$D(\underline{\underline{\mathbf{A}}}) = \begin{vmatrix} 0.3 & 0.52 & 1 \\ 0.5 & 1 & 1.9 \\ 0.1 & 0.3 & 0.5 \end{vmatrix} = -0.0022$$

$$x_1 = \frac{\begin{vmatrix} -0.01 & 0.52 & 1 \\ 0.67 & 1 & 1.9 \\ -0.44 & 0.3 & 0.5 \end{vmatrix}}{D(\underline{\underline{\mathbf{A}}})} = \frac{0.03278}{-0.0022} = -14.9; \quad x_2 = \frac{\begin{vmatrix} 0.3 & -0.01 & 1 \\ 0.5 & 0.67 & 1.9 \\ 0.1 & -0.44 & 0.5 \end{vmatrix}}{D(\underline{\underline{\mathbf{A}}})} = \frac{0.0649}{-0.0022} = -29.5;$$

$$x_3 = \frac{\begin{vmatrix} 0.3 & 0.52 & -0.01 \\ 0.5 & 1 & 0.67 \\ 0.1 & 0.3 & -0.44 \end{vmatrix}}{D(\underline{\underline{\mathbf{A}}})} = \frac{-0.04356}{-0.0022} = 19.8;$$

Si noti che mentre dal punto di vista teorico la regola di Cramer può essere utilizzata con sistemi di equazioni lineari con N>3, dal punto di vista operativo non conviene perché la quantità di calcoli necessari cresce in modo estremamente rapido.

SOLUZIONE DI SISTEMI DI PICCOLE DIMENSIONI (N≤3)

Eliminazione delle incognite

Per illustrare questo approccio, si consideri il semplice caso di un sistema con N=2. Dopo aver moltiplicato la prima equazione per a_{21} e la seconda per a_{11} si esegua la differenza in modo da eliminare l'incognita x_1:

$$\begin{matrix} a_{11}\,x_1 + a_{12}\,x_2 = c_1 \\ a_{21}\,x_1 + a_{22}\,x_2 = c_2 \end{matrix} \longrightarrow \begin{matrix} a_{21}\,a_{11}\,x_1 + a_{21}\,a_{12}\,x_2 = a_{21}\,c_1 \\ a_{11}\,a_{21}\,x_1 + a_{11}\,a_{22}\,x_2 = a_{11}\,c_2 \end{matrix} \longrightarrow x_2 = \frac{a_{11}c_2 - a_{21}c_1}{a_{11}a_{22} - a_{12}a_{21}}$$

Sostituendo x_2 nella prima equazione si determina anche x_1 :

$$x_1 = \frac{c_1 - a_{12}x_2}{a_{11}} = \frac{c_1(a_{11}a_{22} - a_{12}a_{21}) - a_{12}(a_{11}c_2 - a_{21}c_1)}{a_{11}(a_{11}a_{22} - a_{12}a_{21})} = \frac{a_{22}c_1 - a_{12}c_2}{a_{11}a_{22} - a_{12}a_{21}}$$

Si noti che a queste stesse espressioni si può pervenire anche utilizzando la regola di Cramer.

Questa tecnica può essere estesa a sistemi con più di tre equazioni. Tuttavia, se il metodo deve essere applicato in modo manuale, diventa rapidamente ingestibile per il gran numero di calcoli. D'altra parte questo approccio si presta ad essere posto in forma algoritmica e dunque implementato su calcolatore.

Esempio: Utilizzando la regola di eliminazione delle incognite si risolva il sistema:

$$\begin{matrix} 3\,x_1 + 2\,x_2 = 18 \\ -\,x_1 + 2\,x_2 = 2 \end{matrix}$$

Utilizzando le espressioni precedenti risulta:

$$x_1 = \frac{a_{22}c_1 - a_{12}c_2}{a_{11}a_{22} - a_{12}a_{21}} = \frac{2*18 - (2)*2}{3*2 - 2*(-1)} = 4; \qquad x_2 = \frac{a_{11}c_2 - a_{21}c_1}{a_{11}a_{22} - a_{12}a_{21}} = \frac{3*2 - (-1)*18}{3*2 - 2*(-1)} = 3;$$

Metodi diretti: eliminazione Gaussiana semplificata

Il metodo dell'eliminazione delle incognite fa parte della famiglia degli approcci basati sull'eliminazione Gaussiana. Alcune di queste tecniche permettono di risolvere in modo automatico sistemi di equazioni lineari di grande dimensione.

$$
\begin{aligned}
&a_{11}\,x_1 + a_{12}\,x_2 + \ldots a_{1j}\,x_j + \ldots a_{1N}\,x_N = c_1\\
&a_{21}\,x_1 + a_{22}\,x_2 + \ldots a_{2j}\,x_j + \ldots a_{2N}\,x_N = c_2\\
&\quad\vdots \qquad\quad \vdots \qquad\qquad \vdots \qquad\qquad \longrightarrow \qquad \underline{\underline{A}}\;\underline{x} = \underline{C};\\
&a_{i1}\,x_1 + a_{i2}\,x_2 + \ldots a_{ij}\,x_j + \ldots a_{iN}\,x_N = c_i\\
&\quad\vdots \qquad\quad \vdots \qquad\qquad \vdots\\
&a_{N1}\,x_1 + a_{N2}\,x_2 + \ldots a_{Nj}\,x_j + \ldots a_{NN}\,x_N = c_N
\end{aligned}
$$

In pratica:

- si manipola un'equazione del sistema in modo che una delle incognite venga espressa esplicitamente in funzione delle altre;
- si sostituisce l'espressione ricavata in tutte le equazioni rimanenti, riducendo, in tal modo di uno l'ordine del sistema.
- si itera il procedimento fino a quando il problema originario viene ricondotto ad un sistema di una equazione in una incognita, banale da invertire.
- le altre componenti della soluzione possono successivamente essere calcolate attraverso una procedura di sostituzione a ritroso.

Per migliorare l'affidabilità dell'algoritmo è necessario evitare che il calcolatore tenti di eseguire una divisione per zero. Il metodo descritto di seguito non gestisce questa situazione ed è pertanto noto col nome di *Eliminazione Gaussiana semplificata.*

Come nel caso di due equazioni, la tecnica da applicare al sistema di N equazioni prevede due fasi: **l'eliminazione (in avanti) delle incognite** e la loro determinazione attraverso **la sostituzione all'indietro**.

Eliminazione Gaussiana semplificata: Eliminazione delle incognite

In questa sezione si vuole trasformare il sistema originario (cui è associato la generica matrice dei coefficienti **A** di dimensione *NxN*) in un sistema di equazioni equivalente (vale a dire che ammette la stessa soluzione) al quale sia associato una matrice **A**' di dimensioni NxN del tipo **triangolare superiore**.

L'algoritmo da utilizzare è il seguente:

1. Si divide la prima equazione del sistema per a_{11} in modo da rendere uguale a 1 il primo coefficiente della prima equazione (**Normalizzazione**):

$$x_1 + \frac{a_{12}}{a_{11}} x_2 + \ldots + \frac{a_{1n}}{a_{11}} x_n = \frac{c_1}{a_{11}} \qquad (1)$$

2. Si moltiplica l'equazione normalizzata per a_{21} (primo coeff. della seconda equazione):

$$a_{21}x_1 + \frac{a_{21}\,a_{12}}{a_{11}} x_2 + \ldots + \frac{a_{21}\,a_{1n}}{a_{11}} x_n = \frac{a_{21}\,c_1}{a_{11}} \qquad (2)$$

3. Poiché il primo coefficiente della prima e della seconda equazione sono ora identici, possiamo eliminare la prima incognita sottraendo la (2) dalla seconda equazione. Si ottiene pertanto (l'apice indica che i coefficienti sono cambiati)

$$\left(a_{22} - \frac{a_{21}\,a_{12}}{a_{11}}\right)x_2 + \ldots + \left(a_{2n} - \frac{a_{21}\,a_{1n}}{a_{11}}\right)x_n = c_2 - \frac{a_{21}\,c_1}{a_{11}} \quad \rightarrow \quad a'_{22}x_2 + \ldots + a'_{2n}x_n = c'_n$$

4. La procedura del punto 3 viene ripetuta per eliminare la prima incognita dalle equazioni rimanenti. La prima equazione (a) viene detta *equazione pi*vot, mentre il coefficiente a_{11} si chiama *coefficiente pivot* o semplicemente *pivot* .

$$\begin{aligned}
&a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots a_{1N}x_N = c_1 && \text{(a)} \\
&a'_{22}x_2 + a'_{23}x_3 + \ldots a'_{2N}x_N = c'_2 && \text{(b)} \\
&a'_{32}x_2 + a'_{33}x_3 + \ldots a'_{3N}x_N = c'_3 && \text{(c)} \qquad (3) \\
&\vdots \quad \vdots \quad \vdots && \\
&a'_{N2}x_2 + \ldots a'_{Nj}x_j + \ldots + a'_{NN}x_N = c'_N && \text{(d)}
\end{aligned}$$

5. Si ripetono i punti da 1 a 4 in modo da eliminare la seconda incognita dalle equazioni che vanno dalla terza alla n^{ma} usando la seconda equazione (b) come equazione pivot.

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1N}x_N &= c_1 \\ a'_{22}x_2 + a'_{23}x_3 + \ldots + a'_{2N}x_N &= c'_2 \\ a''_{33}x_3 + \ldots + a''_{3N}x_N &= c''_3 \\ \vdots \quad & \quad \vdots \quad \vdots \\ a''_{N3}x_j + \ldots + a''_{NN}x_N &= c''_N \end{aligned} \qquad (4)$$

6. Questa procedura viene ripetuta usando via via le equazioni successive come pivot. Alla fine si perviene ad un sistema equivalente a quello iniziale, ma associato ad una matrice di tipo triangolare superiore.

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1N}x_N &= c_1 \qquad (a) \\ a'_{22}x_2 + a'_{23}x_3 + \ldots + a'_{2N}x_N &= c'_2 \qquad (b) \\ a''_{33}x_3 + \ldots \quad + a''_{3N}x_N &= c''_3 \qquad (c) \\ \vdots \quad & \quad \vdots \quad \vdots \\ a^{(n-1)}_{NN}x_N &= c^{(n-1)}_N \qquad (d) \end{aligned} \qquad (5)$$

L'equazione (5d) si può ora risolvere facilmente: $x_N = \frac{c_N^{n-1}}{a_{NN}^{n-1}}$.

Il valore di x_N può essere sostituito nella penultima equazione in modo da ricavare x_{N-1}. Iterando la procedura di sostituzione, si determinano, una per volta, le restanti incognite:

$$x_i = \frac{c_i^{i-1} - \sum_{j=i+1}^{n} a_{ij}^{i-1} x_j}{a_{ii}^{i-1}}; \qquad i = n-1, n-2, \ldots, 1$$

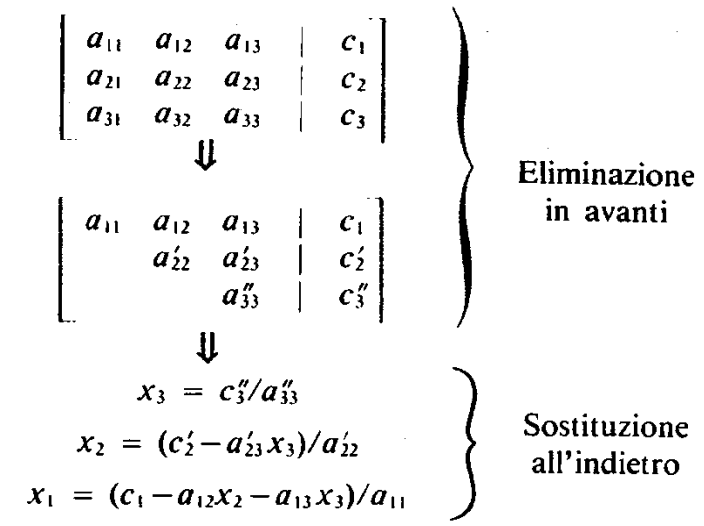


Figura 7.3 Rappresentazione grafica delle due fasi dell'eliminazione gaussiana. L'eliminazione in avanti riduce la matrice dei coefficienti in forma triangolare superiore. In seguito, la sostituzione all'indietro consente di ricavare il valore delle incognite.

Esempio: Utilizzando l'eliminazione gaussiana semplificata risolvere il sistema con una precisione di almeno 6 cifre significative.

$$3\,x_1 - 0.1\,x_2 - 0.2\,x_3 = 7.85 \qquad (6.a)$$
$$0.1\,x_1 + 7\,x_2 - 0.3\,x_3 = -19.3 \qquad (6.b)$$
$$0.3\,x_1 - 0.2\,x_2 + 10\,x_3 = 71.4 \qquad (6.c)$$

Soluzione: Eliminazione in avanti

1. *Normalizziamo l'equazione (6.a) dividendola per il coefficiente pivot:*
   $$x_1 - 0.0333333\,x_2 - 0.0666667\,x_3 = 2.61667 \qquad (7.a)$$

2. *Moltiplichiamo la (7a) per 0.1.*
   $$0.1\,x_1 - 0.00333333\,x_2 - 0.00666667\,x_3 = 0.261667 \qquad (8.a)$$

3. *Sottraiamo la (8a) dalla (6b):*
   $$7.00333333\,x_2 - 0.293333\,x_3 = -19.5617 \qquad (9.a)$$

4. *Moltiplichiamo la (7a) per 0.3 e sottraiamo il risultato dalla (6c). Si perviene al sistema:*
   $$3\,x_1 - 0.1\,x_2 - 0.2\,x_3 = 7.85 \qquad (10.a)$$
   $$7.00333\,x_2 - 0.293333\,x_3 = -19.5617 \qquad (10.b)$$
   $$-0.190000\,x_2 + 10.02000\,x_3 = 70.6150 \qquad (10.c)$$

5. *Si divide la (10.b) per 7.00333333, si moltiplica per -0.190000 e si sottrae il risultato alla (10c). In tal modo si elimina l'incognita x_2.*
   $$3\,x_1 - 0.1\,x_2 - 0.2\,x_3 = 7.85 \qquad (11.a)$$
   $$7.00333\,x_2 - 0.293333\,x_3 = -19.5617 \qquad (11.b)$$
   $$10.0120\,x_3 = 70.0843 \qquad (11.c)$$

Soluzione: Sostituzione all'indietro:

La (11.c) ci fornisce $x_3 = 70.0843/10.0120 = 7.00003$

Dalla (11.b) si ottiene $x_2 = (-19.5617 + 0.293333\,x_3)/7.00333 = -2.50000$

Dalla (11.a)risulta: $x_1 = (7.85 + 0.2\,x_3 + 0.1\,x_2)/3 = 3.00000$

I risultati sono prossimi alla soluzione esatta: $x_1 = 3;\ x_2 = -2.5;\ x_3 = 7$

(Si può verificare che questa è la soluzione esatta sostituendo nelle eq. 6).

Programma in fortran per l'eliminazione gaussiana semplificata

```
      DIMENSION A(15,16), X(15)          ! A = matrice dei coeff. e dei termini noti
        READ(5,1)N                       ! N= numero di equazioni
1       FORMAT(I5)
        M=N+1
        DO 160 I=1,N
                DO 150 J=1,M
                    READ(5,2)A(I,J)
2                FORMAT(F10.0)
150             CONTINUE
160   CONTINUE
        CALL GAUSS(N,A,X)                ! X= VETTORE DELLE INCOGNITE
        DO 200 I=1,N
                WRITE(6,3)X(I)
3               FORMAT(' ',F10.3)
200 CONTINUE
        STOP
        END

        SUBROUTINE GAUSS(N,A,X)
        DIMENSION A(15,16), X(15)
        M=N+1
        L=N-1
        DO 1140 K=1,L
                JJ=K
                KP=K+1
                DO 1100 I=KP,N
                    QT=A(I,K)/A(K,K)
                    DO 1090 J=KP,N
                        A(I,J)=A(I,J)-QT*A(K,J)
1090            CONTINUE
1100  CONTINUE
                DO 1130 I=KP,N
                    A(I,K)=0.
1130  CONTINUE
1140 CONTINUE
         X(N)=A(N,M)/A(N,N)
        DO 1240 NN=1,L
                SUM=0.0
                I=N-NN
                IP=I+1
                DO 1220 J=IP,N
                    SUM=SUM+A(I,J)*X(J)
1220            CONTINUE
                X(I)=(A(I,M)-SUM)/A(I,I)
1240 CONTINUE
        RETURN
        END
```

Punti deboli del metodo di eliminazione Gaussiana

- Divisione per zero e per numeri molto prossimi a zero
- Errori di arrotondamento
- Sistemi mal condizionati

Divisione per zero

Il motivo principale per il quale il metodo presentato viene detto semplificato è che non prende in considerazione l'ipotesi che possano essere tentate delle divisioni per zero. Si consideri a titolo di esempio il sistema:

$$\begin{aligned} 2\,x_2 + 3\,x_3 &= 8 \\ 4\,x_1 + 6\,x_2 + 7\,x_3 &= -3 \\ 2\,x_1 + x_2 + 6\,x_3 &= 5 \end{aligned}$$

La normalizzazione della prima equazione comporta una divisione per zero. Difficoltà simili insorgono anche quando un coefficiente è molto vicino a zero. Per evitare questi problemi si introdurrà la tecnica del pivoting

Errori di arrotondamento

Si è visto nell'esempio precedente che usando 6 cifre significative, la soluzione ottenuta non coincideva con quella esatta. Si intuisce come aumentando il numero di cifre significative esatte, l'accuratezza del risultato cresca.

D'altra parte, se si risolve il sistema dell'esempio precedente con 3 cifre significative, si perviene al sistema triangolare:

$$\begin{aligned} 3\,x_1 - 0.1\,x_2 - 0.2\,x_3 &= 7.85 && (12.a) \\ 7.00\,x_2 - 0.293\,x_3 &= -19.6 && (12.b) \\ 9.99\,x_3 &= 70.1 && (12.c) \end{aligned}$$

Che risolto fornisce : $x_1 = 3.17$; $|\varepsilon_r| = 5.7\%$

$x_2 = -2.51$; $|\varepsilon_r| = 0.4\%$

$x_3 = 7.02$; $|\varepsilon_r| = 0.29\%$

Si noti che sostituendo i valori trovati nelle equazioni, queste sono ben lontane dall'essere soddisfatte: $8.36 \neq 7.85$; $-19.4 \neq -19.3$; $71.7 \neq 71.4$;

Sebbene i calcolatori impiegano nei calcoli un numero di cifre significative ben più alto di 3, il problema dato dall'arrotondamento diventa reale per i sistemi di grandi dimensioni (con 7 cifre significative da 25-50 equazioni in poi).

Sistemi mal condizionati

Gli errori di arrotondamento comportano inevitabilmente piccole variazioni nei coefficienti del sistema di equazioni o nei termini noti. Se il sistema è mal condizionato, per definizione, queste piccole variazioni producono grandi differenze tra la soluzione trovata e quella esatta.

Esempio

Risolvere il sistema.

$$x_1 + 2\,x_2 = 10 \qquad (13.a)$$
$$1.1\,x_1 + 2\,x_2 = 10.4 \qquad (13.b)$$

Risolvere successivamente il sistema ottenuto modificando il coefficiente di x_1 nella seconda equazione (da 1.1 a 1.05).

Soluzione 1: Con le espressioni trovate nel metodo di eliminazione delle incognite

$$x_1 = \frac{a_{22}c_1 - a_{12}c_2}{a_{11}a_{22} - a_{12}a_{21}} = \frac{2*10 - (2)*10.4}{1*2 - 2*(1.1)} = 4; \quad x_2 = \frac{a_{11}c_2 - a_{21}c_1}{a_{11}a_{22} - a_{12}a_{21}} = \frac{1*10.4 - 1.1*10}{1*2 - 2*(1.1)} = 3;$$

Soluzione 2: Il sistema modificato porta a risultati sensibilmente diversi:

$$x_1 = \frac{a_{22}c_1 - a_{12}c_2}{a_{11}a_{22} - a_{12}a_{21}} = \frac{2*10 - (2)*10.4}{1*2 - 2*(1.05)} = 8; \quad x_2 = \frac{a_{11}c_2 - a_{21}c_1}{a_{11}a_{22} - a_{12}a_{21}} = \frac{1*10.4 - 1.05*10}{1*2 - 2*(1.05)} = 1;$$

Va notato che la causa del mal-condizionamento è dovuta principalmente al fatto che al denominatore compare la differenza di due termini quasi uguali (vale a dire che **il determinante del sistema è quasi zero**).

Si noti che in questi casi la sostituzione del risultato nelle equazioni di partenza non serve ad evidenziare il problema. Difatti nel nostro caso, si ottiene:

$$8 + 2*1 = 10 \quad = 10 \qquad (13.a)$$
$$1.1*8 + 2*1 = 10.8 \cong 10.4 \qquad (13.b)$$

cioè l'errore relativamente piccolo commesso induce erroneamente a credere nella bontà della soluzione.

Come accorgersi del mal-condizionamento di un sistema?

Per accorgersi del mal-condizionamento di un sistema conviene riferirsi al valore del suo determinante.
In effetti, se il determinante fosse esattamente zero, non esisterebbe soluzione del sistema (o ce ne sarebbero infinite).
Un determinante prossimo a zero è pertanto indice di un sistema mal-condizionato.
Va immediatamente detto che questo criterio va preso con le molle in quanto se si moltiplica un'equazione per una costante il determinante scala nello stesso modo.
Il valore del determinante è pertanto relativo e risulta influenzato dall'ordine di grandezza dei suoi coefficienti.
Esempio: Vogliamo vedere come il valore del determinante risulta influenzato dall'ordine di grandezza dei suoi coefficienti.
Calcoliamo il determinante dei seguenti sistemi:

1) $3\,x_1 + 2\,x_2 = 18$ (14.a) $D = 3*2-2*(-1)=8$
   $-\,x_1 + 2\,x_2 = 2$ (14.b)

2) $x_1 + 2\,x_2 = 10$ (15.a) $D=1*2-2*(1.1)=-0.2$
   $1.1x_1 + 2\,x_2 = 10.4$ (15.b)

3) $10\,x_1 + 20\,x_2 = 100$ (16.a) $D=10*20-20*11=20$
   $11\,x_1 + 20\,x_2 = 104$ (16.b)

Si noti che il primo sistema (il cui determinante vale 8) è ben condizionato.
Il secondo (determinante vale -0.2) è mal condizionato.
Sembrerebbe pertanto che i sistemi che presentano un determinante prossimo a zero sono mal condizionati.
Questa conclusione è smentita dal terzo sistema ottenuto dal secondo moltiplicando le due equazioni per la costante 10. Si noti che la moltiplicazione per una costante influisce sul valore del determinante (che ora vale 20 cioè più del doppio di quello associato al primo sistema), ma non ha effetto sulla soluzione (basta pensare alla soluzione grafica: l'angolo tra le rette non cambia).

Per superare questo effetto di scala che complica la relazione tra il condizionamento di un sistema e il suo determinante si possono **normalizzare** le equazioni in modo che l'elemento più grande di ogni riga sia uguale a 1.

Esempio: Normalizzare i sistemi precedenti e calcolare i determinanti

1) $x_1 + 0.667\, x_2 = 6$ (14.a) D= 1*1-0.667*(-0.5)=1.333
$-0.5\, x_1 + x_2 = 1$ (14.b)

2) $0.5\, x_1 + x_2 = 5$ (15.a) D=0.5*1-1*0.55=-0.05
$0.55 x_1 + x_2 = 5.2$ (15.b)

3) Il terzo sistema dopo la normalizzazione coincide col secondo e il determinante è sempre -0.05.

Si è affermato in precedenza che il calcolo del determinante è difficile per sistemi con N>3. Sembrerebbe dunque che l'idea di stimare il buon condizionamento di una matrice sulla base del suo determinante non sia utilizzabile. In realtà, come si vedrà tra breve esiste un semplice algoritmo basato sull'eliminazione gaussiana che si può utilizzare per valutare il determinante di un sistema di grande dimensione.

Va tuttavia detto che il condizionamento di una matrice può essere valutato anche in modo diverso. Ad esempio

- si può utilizzare l'inversa della matrice.
- si possono modificare leggermente i coefficienti e ricalcolare la soluzione. Se questa è molto diversa è probabile che il sistema sia malcondizionato.

CALCOLO DEL DETERMINANTE PER MATRICI CON N>3

Alla fine della fase di eliminazione in avanti, il sistema equivalente cui si perviene è di tipo triangolare superiore. Per matrici di questo tipo, il determinante si può ottenere facilmente come prodotto degli elementi sulla diagonale:

$$D = a_{11} * a_{22} * a_{33} \ldots a_{NN}$$

MIGLIORAMENTO DELLE TECNICHE DI RISOLUZIONE

Con alcuni accorgimenti è possibile elevare la qualità della soluzione di un sistema lineare ottenuta attraverso il metodo dell'eliminazione gaussiana.

Uso di un maggior numero di cifre significative

Il metodo più semplice per migliorare la qualità della soluzione consiste nell'elevare il numero di cifre significative. Ad esempio in Fortran è consentito usare variabili in doppia precisione.

Pivoting

L'algoritmo di eliminazione gaussiana fallisce se il coefficiente di pivot è nullo o prossimo allo zero (in quest'ultimo caso sorgono problemi legati alla fase di arrotondamento).
Prima di normalizzare l'equazione di pivot vale allora la pena di decidere preventivamente in quale tra le righe rimanenti contenga il coefficiente di pivot più grande. A questo punto si opera uno scambio tra la riga corrente e quella che contiene il pivot vengono scambiate di posto così da avere il pivot più grande possibile.
Questa tecnica va sotto il nome di *pivoting parziale.*

Se la ricerca dell'elemento adatto e il successivo scambio avvengono sia secondo le righe che secondo le colonne si parla di *pivoting completo.*
Si noti che il pivoting completo modifica l'ordine delle incognite e per questo viene utilizzato di rado.
Il pivoting parziale invece presenta diversi vantaggi:

- permette di evitare le divisioni per zero.
- permette di minimizzare gli errori di arrotondamento e per questo contribuisce a superare le difficoltà dovute ad un cattivo condizionamento del sistema.

Esempio

Utilizzando l'eliminazione gaussiana semplificata risolvere il sistema

$0.0003\ x_1 + 3.0000\ x_2 = 2.0001$ (15.a)

$1.0000\ x_1 + 1.0000\ x_2 = 1.0000$ (15.b)

Come si vede il primo pivot (0.0003) è prossimo a zero.

Ripetere i calcoli utilizzando il pivoting parziale (in pratica invertendo le righe).

La soluzione esatta del sistema è $x_1=1/3$; $x_2=2/3$

Soluzione 1

Senza pivoting si ha, dopo la fase di eliminazione in avanti:

$x_1 + 10000\ x_2 = 6667$ (16.a)

$-9999\ x_2 = -6666$ (16.b)

La soluzione è pertanto: $x_2=2/3$; $x_1 = \dfrac{2.0001 - 3*(2/3)}{0.0003}$

Si noti che il valore di x_1 dipende in modo sensibile dal numero di cifre significative utilizzate.

Ad esempio con 3 cifre significative $x_1=-3.33$;

invece con 7 cifre significative si ottiene $x_1=0.3330000$.

Soluzione 2

Se si utilizza il pivoting parziale (si scambiano le righe) le equazioni diventano:

$1.0000\ x_1 + 1.0000\ x_2 = 1.0000$ (16.b)

$0.0003\ x_1 + 3.0000\ x_2 = 2.0001$ (16.a)

Normalizzando ed eliminando l'incognita si ottiene:

La soluzione è pertanto: $x_2=2/3$; $x_1 = \dfrac{1-(2/3)}{1}$

In questo caso il valore di x_1 dipende in modo meno marcato dal numero di cifre significative utilizzate. Ad esempio con 3 cifre significative $x_1=0.333$; invece con 7 cifre significative si ottiene $x_1=0.3333333$.

Implementazione della strategia di pivoting parziale

```
      SUBROUTINE PARPIV(A)
      DIMENSION A(15,16)
      COMMON N,K
      JJ=K
      KP=K+1
      N1=N+1
      B=ABS(A(K,K))
      DO I=KP,N
            BP=ABS(A(I,K))
            IF(B-BP.GE.0)GOTO 3090
            B=BP
            JJ=I
      ENDDO
3090 IF(JJ-K.EQ.0) GOTO 3150
      DO J=K,N1
            TE=A(JJ,J)
            A(JJ,J)=A(K,J)
            A(K,J)=TE
      ENDDO
      RETURN
      END
```

Normalizzazione

La normalizzazione oltre che a standardizzare la relazione tra il condizionamento di una matrice e il determinante a questa associato, consente di ridurre gli errori di arrotondamento quando nel sistema compaiono equazioni con coefficienti che differiscono gli uni dagli altri di diversi ordini di grandezza. Situazioni del genere sono molto diffuse nei modelli matematici associati ai problemi ingegneristici in quanto nelle equazioni compaiono grandezze le cui unità di riferimento sono molto diverse dal punto di vista quantitativo. Per esempio, con riferimento ad una LKT in un problema di teoria dei circuiti, verranno sommate cadute di tensione ai capi di resistori, induttori e condensatori. A seconda delle frequenze e dei valori dei parametri queste tensioni possono variare dai microVolt ai chiloVolt.

Differenze cosi marcate tra i coefficienti delle equazioni influenzano negativamente la scelta del pivoting e possono condurre a risultati poco accurati.

Esempio:

a) Utilizzando l'eliminazione gaussiana con pivoting e tre cifre significative risolvere il sistema:

$$2\,x_1 + 100000\,x_2 = 100000$$
$$x_1 + x_2 = 2$$

b) Risolvere lo stesso sistema dopo avere normalizzato le equazioni in modo che il coefficiente più grande di ciascuna riga sia 1.

La soluzione corretta è (x_1=1.00002 e x_2=0.99998) che con tre cifre significative diventa (x_1=1.00 e x_2=1.00)

Soluzione

a) L'eliminazione in avanti senza normalizzazione conduce a:

$$2\,x_1 + 100000\,x_2 = 100000$$
$$-\ 50000x_2 = -50000$$

che risolto con la sostituzione all'indietro fornisce la soluzione errata (x_1=0.00 e x_2=1.00)

b) Normalizzando le equazioni originali del sistema si perviene al sistema:

$$0.00002\quad x_1 + x_2 = 1$$
$$x_1 + x_2 = 2$$

L'applicazione del pivoting si riduce allo scambio delle righe:

$$x_1 + x_2 = 2$$
$$0.00002\quad x_1 + x_2 = 1$$

L'eliminazione in avanti conduce a

$$x_1 + x_2 = 2$$
$$x_2 = 1.00$$

che risolto con la sostituzione all'indietro porta alla soluzione corretta (x_1=1.00 e x_2=1.00)

Va notato che sebbene la procedura di normalizzazione permetta di ridurre gli errori di arrotondamento, comporta essa stessa un arrotondamento e per questo va eseguita solo se ce n'è veramente bisogno, ovvero se i coefficienti del sistema sono molto diversi.

Compensazione degli errori

Quando il pivoting parziale e la normalizzazione non sono sufficienti per ottenere soluzioni accurate (ad esempio, per il sistema 12), si può ricorrere alla procedura descritta di seguito. Consideriamo il generico sistema:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1j}x_j + \cdots + a_{1N}x_N = c_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2j}x_j + \cdots + a_{2N}x_N = c_2 \\ a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{ij}x_j + \cdots + a_{iN}x_N = c_j \\ a_{N1}x_1 + a_{N2}x_2 + \cdots + a_{Nj}x_j + \cdots + a_{NN}x_N = c_N \end{cases} \qquad (17a)$$

Sia $(\bar{x}_1, \bar{x}_2, \cdots \bar{x}_j, \cdots \bar{x}_N)$ una soluzione approssimata. Sostituendo nella (17a) si ha:

$$\begin{cases} a_{11}\bar{x}_1 + a_{12}\bar{x}_2 + \cdots + a_{1j}\bar{x}_j + \cdots + a_{1N}\bar{x}_N = \bar{c}_1 \\ a_{21}\bar{x}_1 + a_{22}\bar{x}_2 + \cdots + a_{2j}\bar{x}_j + \cdots + a_{2N}\bar{x}_N = \bar{c}_2 \\ a_{i1}\bar{x}_1 + a_{i2}\bar{x}_2 + \cdots + a_{ij}\bar{x}_j + \cdots + a_{iN}\bar{x}_N = \bar{c}_j \\ a_{N1}\bar{x}_1 + a_{N2}\bar{x}_2 + \cdots + a_{Nj}\bar{x}_j + \cdots + a_{NN}\bar{x}_N = \bar{c}_N \end{cases} \qquad (17b)$$

La soluzione esatta può essere espressa come somma della soluzione approssimata e di fattori di correzione incogniti: $x_j = \bar{x}_j + \Delta x_j \quad j = 1,..., N$ (17c)

Sostituendo (17c) nel sistema originario e sottraendo (17b) si ottiene un nuovo sistema che risolto fornisce i fattori di correzione:

$$\begin{cases} a_{11}\Delta x_1 + a_{12}\Delta x_2 + \cdots + a_{1j}\Delta x_j + \cdots + a_{1N}\Delta x_N = c_1 - \bar{c}_1 = E_1 \\ a_{21}\Delta x_1 + a_{22}\Delta x_2 + \cdots + a_{2j}\Delta x_j + \cdots + a_{2N}\Delta x_N = c_2 - \bar{c}_2 = E_2 \\ a_{i1}\Delta x_1 + a_{i2}\Delta x_2 + \cdots + a_{ij}\Delta x_j + \cdots + a_{iN}\Delta x_N = c_j - \bar{c}_j = E_j \\ a_{N1}\Delta x_1 + a_{N2}\Delta x_2 + \cdots + a_{Nj}\Delta x_j + \cdots + a_{NN}\Delta x_N = c_N - \bar{c}_N = E_N \end{cases} \qquad (17d)$$

Esempio: Si riconsideri il sistema (6) che con tre cifre significative forniva risultati affetti da un margine di errore significativo

$$3\,x_1 - 0.1\,x_2 - 0.2\,x_3 = 7.85 \qquad (6.a)$$
$$0.1\,x_1 + 7\,x_2 - 0.3\,x_3 = -19.3 \qquad (6.b)$$
$$0.3\,x_1 - 0.2\,x_2 + 10\,x_3 = 71.4 \qquad (6.c)$$

soluzione esatta:(x_1=3, x_2=-2.5, x_3=7)

soluzione approssimata con 3 cifre significative: ($\bar{x}_1 = 3.17, \bar{x}_2 = -2.51, \bar{x}_3 = 7.02$)

errore relativo percentuale:($|\varepsilon_{t,x1}|$=5.7%, $|\varepsilon_{t,x2}|$=0.4%, $|\varepsilon_{t,x3}|$=0.29%)

Usando la compensazione degli errori è possibile migliorare queste stime.

Sostituendo le soluzioni approssimate nelle eqs. (6) si ottiene un vettore dei termini noti diverso da quello originale: $\bar{c}_1 = 8.36,\ \bar{c}_2 = -19.4,\ \bar{c}_3 = 71.7$.
La differenza tra il vettore dei termini noti originale e quello testé calcolato rappresenta il termine noto del sistema (17c) da risolvere per ottenere i fattori di correzione:

$$3\,x_1 - 0.1\,x_2 - 0.2\,x_3 = 7.85 - 8.36 = -0.51$$
$$0.1\,x_1 + 7\,x_2 - 0.3\,x_3 = -19.3 + 19.4 = 0.1$$
$$0.3\,x_1 - 0.2\,x_2 + 10\,x_3 = 71.4 - 71.7 = -0.3$$

Risolvendo nella maniera usuale con tre cifre significative si ottengono i fattori di correzione: Δx_1=-0.171, Δx_2=0.0157, Δx_3=-0.0246;
Usando la (17c) si determina una soluzione più accurata del sistema originale:

soluzione più accurata:

$$x_1' = \bar{x}_1 + \Delta x_1 = 3.17 - 0.171 = 3.00;$$
$$x_2' = \bar{x}_2 + \Delta x_2 = -2.51 - 0.0157 = -2.49;$$
$$x_3' = \bar{x}_3 + \Delta x_3 = 7.02 - 0.0246 = 7.00;$$

Per ottenere dei fattori di correzione accurati nel caso di sistemi mal condizionati è necessario che il termine noto modificato sia calcolato nel modo più preciso possibile. Ad esempio, in Fortran utilizzando la doppia precisione.

Metodi diretti: Metodo di Gauss-Jordan

Il metodo di Gauss- Jordan è una variante dell'eliminazione Gaussiana e consente di calcolare numericamente l'inversa di una matrice in maniera immediata.
Come si è già detto, la conoscenza della matrice inversa permette di stabilire se un sistema assegnato è o meno ben condizionato.

A differenza dell'eliminazione Gaussiana, **nel metodo di Gauss-Jordan un'incognita viene eliminata da tutte le equazioni del sistema** e non solo da quelle che si trovano sotto la riga corrente.
Pertanto, alla **fine del processo, si perviene ad una matrice unitaria** piuttosto che ad una matrice triangolare superiore.
Non è, dunque, **necessario eseguire la back-substitution per determinare la soluzione.**

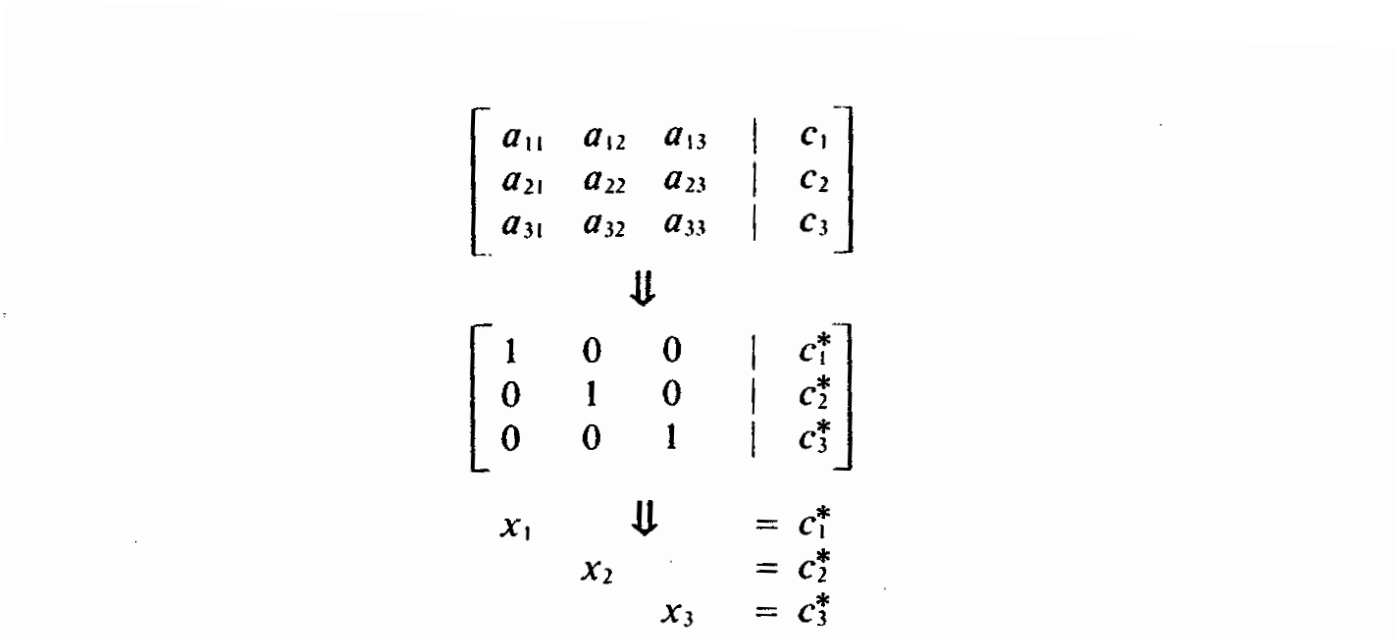


Figura 8.1 Rappresentazione grafica del metodo di Gauss-Jordan. Confrontata con la Figura 7.3 chiarisce la differenza tra questo metodo e l'eliminazione gaussiana. Gli asterischi indicano che gli elementi del vettore dei termini noti sono stati ottenuti con più modifiche successive.

Esempio: Usando il metodo di Gauss-Jordan risolvere il sistema dell'esempio precedente. Si lavori utilizzando 6 cifre significative.

Il sistema risolto precedentemente viene riportato per comodità:

$$\begin{aligned} 3\,x_1 - 0.1\,x_2 - 0.2\,x_3 &= 7.85 \\ 0.1\,x_1 + 7\,x_2 - 0.3\,x_3 &= -19.3 \\ 0.3\,x_1 - 0.2\,x_2 + 10\,x_3 &= 71.4 \end{aligned} \qquad \left[\begin{array}{cccc} 3 & -0.1 & -0.2 & 7.85 \\ 0.1 & 7 & -0.3 & -19.3 \\ 0.3 & -0.2 & 10 & 71.4 \end{array}\right]$$

Aggiungiamo alla matrice dei coefficienti il vettore dei termini noti e normalizziamo la prima riga:

$$\left[\begin{array}{cccc} 1 & -0.0333333 & -0.0666667 & 2.61667 \\ 0.1 & 7 & -0.3 & -19.3 \\ 0.3 & -0.2 & 10 & 71.4 \end{array}\right]$$

Utilizzando a_{11} come pivot per eliminare l'incognita x_1 dalle righe 2 e 3 si ottiene:

$$\left[\begin{array}{cccc} 1 & -0.0333333 & -0.0666667 & 2.61667 \\ 0 & 7.03333 & -0.293333 & -19.5617 \\ 0 & -0.190000 & 10.0200 & 70.6150 \end{array}\right]$$

Si utilizzi ora come pivot a_{22} per eliminare l'incognita x_2 dalle righe 1 e 3:

$$\left[\begin{array}{cccc} 1 & 0 & -0.0680570 & 2.52356 \\ 0 & 1 & -0.0417061 & -2.79320 \\ 0 & 0 & 10.0121 & 70.0843 \end{array}\right]$$

Infine utilizzando come pivot a_{33} si elimina l'incognita x_3 dalle righe 1 e 2 e si ottiene automaticamente la soluzione:

$$\left[\begin{array}{cccc} 1 & 0 & 0 & 3.00000 \\ 0 & 1 & 0 & 2.50001 \\ 0 & 0 & 1 & 7.00003 \end{array}\right]$$

Come si vede non è necessario eseguire la sostituzione all'indietro.

Tutto quanto detto riguardo ai punti deboli e ai metodi per limitare gli errori di arrotondamento nel caso dell'eliminazione gaussiana vale anche per il metodo di Gauss- Jordan.

ALGORITMO PER IL METODO DI GAUSS-JORDAN

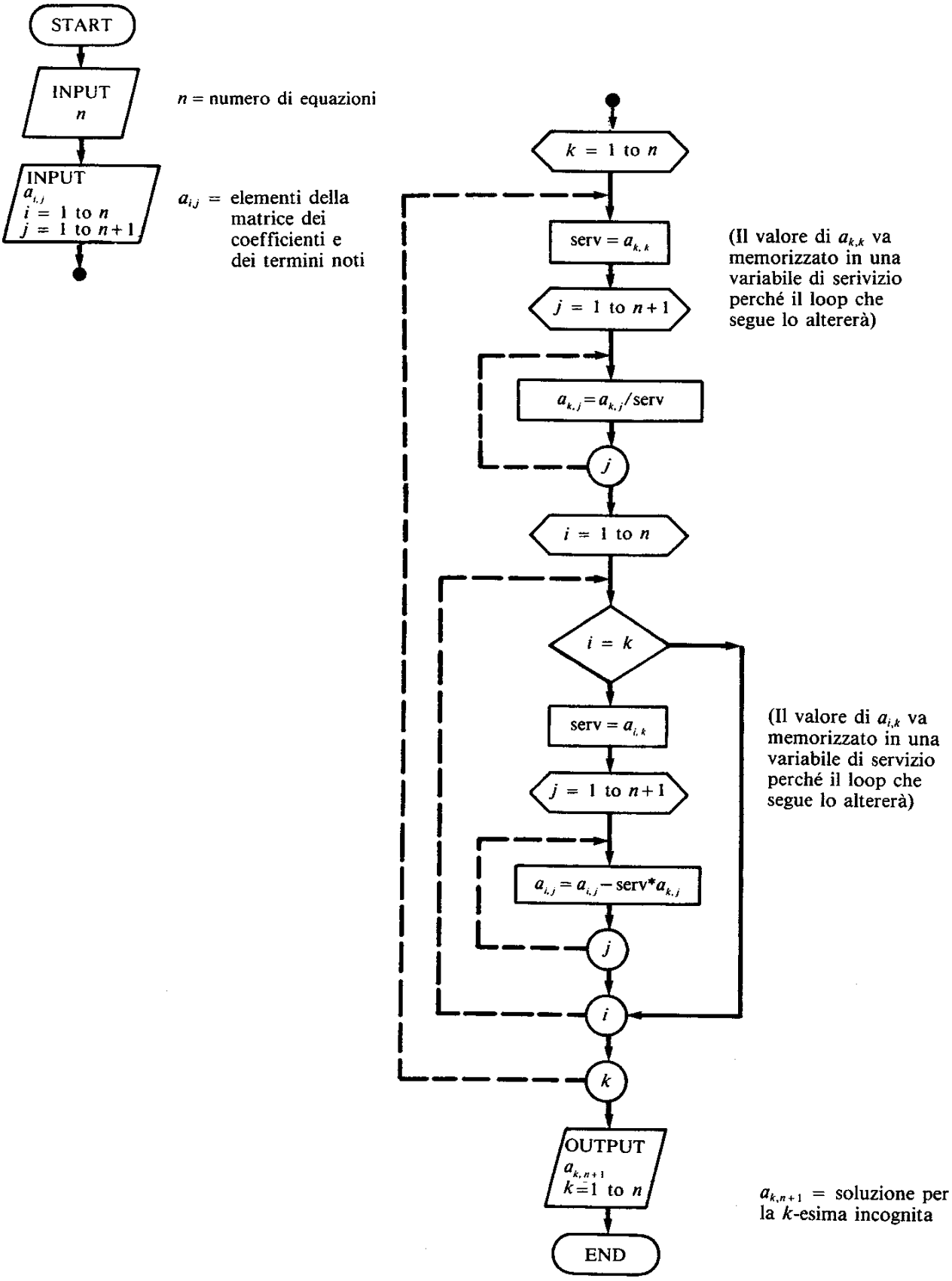


Figura 8.2 Diagramma di flusso per il metodo di Gauss-Jordan senza pivoting parziale.

INVERSIONE DI UNA MATRICE CON GAUSS-JORDAN

Nella rivisitazione dei concetti fondamentali relativi alle matrici si è definita **inversa** di una matrice quadrata **A** di ordine N, la matrice quadrata di ordine N, $\mathbf{A}^{-1}$, tale per cui risulta:

$$\mathbf{A}^{-1}\,\mathbf{A} = \mathbf{A}\,\mathbf{A}^{-1} = \mathbf{I}$$

La matrice inversa può essere utilizzata per risolvere sistemi di equazione attraverso la formula:

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

Questa espressione risulta particolarmente utile nel caso in cui si chiede di risolvere diversi sistemi di equazioni che differiscono l'uno dall'altro per il solo termine noto.

In questi casi dopo aver determinato la matrice inversa $\mathbf{A}^{-1}$, la soluzione di ciascun sistemi richiede una semplice moltiplicazione matrice-vettore.
Si noti che utilizzando una delle tecniche precedentemente esposte, sarebbe stato necessario eseguire l'algoritmo completo di soluzione per ciascuno dei sistemi.
Per il calcolo della matrice inversa ci si avvale del metodo di Gauss- Jordan. Allo scopo bisogna aggiungere alla matrice originale una matrice unità della stessa dimensione. Alla fine dell'algoritmo, al posto della matrice originale vi sarà una matrice unità, mentre al posto della matrice unità comparirà la matrice inversa.

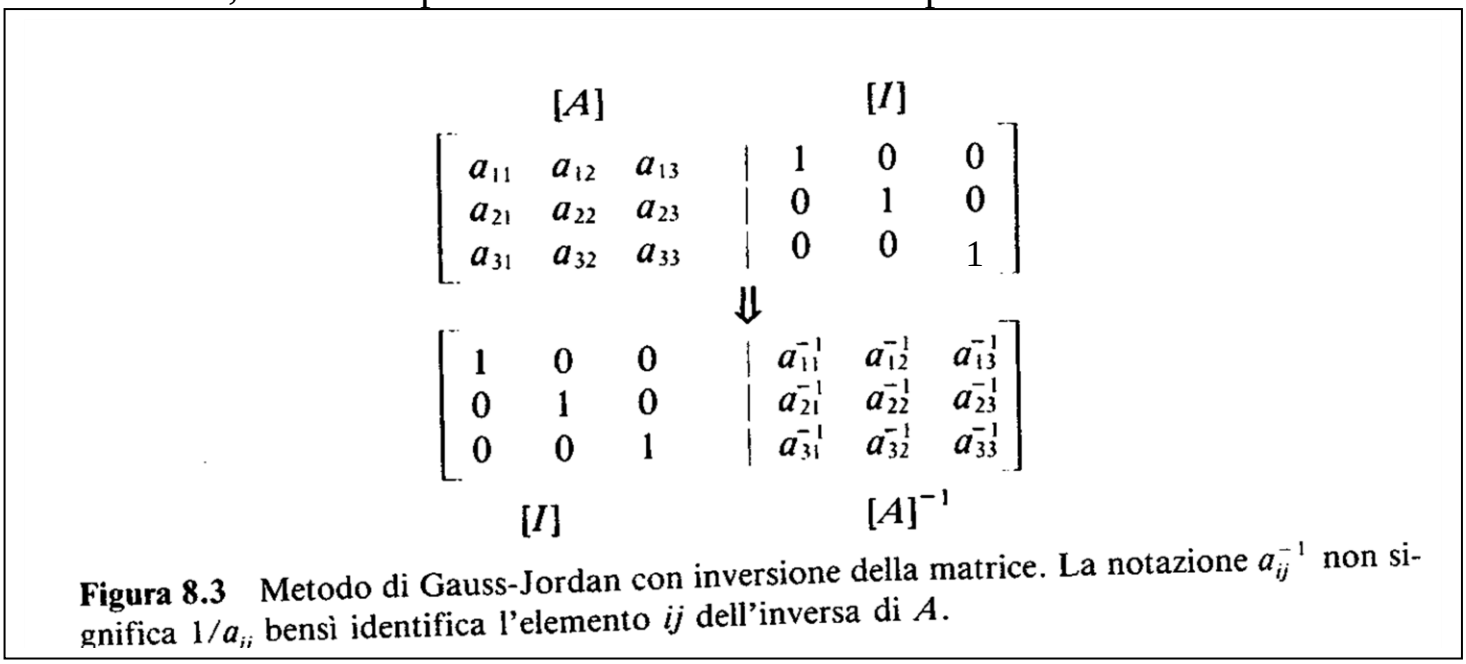


Figura 8.3 Metodo di Gauss-Jordan con inversione della matrice. La notazione a_{ij}^{-1} non significa $1/a_{ij}$ bensì identifica l'elemento ij dell'inversa di A.

Esempio: Usando il metodo di Gauss-Jordan risolvere il sistema dell'esempio precedente. Calcolare la soluzione anche per un secondo vettore dei termini noti: C_1=[20 50 15]. Si lavori utilizzando 6 cifre significative.
Il sistema risolto precedentemente viene riportato per comodità:

$$\begin{aligned} 3\,x_1 - 0.1\,x_2 - 0.2\,x_3 &= 7.85 \\ 0.1\,x_1 + 7\,x_2 - 0.3\,x_3 &= -19.3 \\ 0.3\,x_1 - 0.2\,x_2 + 10\,x_3 &= 71.4 \end{aligned}$$

Aggiungiamo alla matrice dei coefficienti la matrice unità:

$$[A] = \left[\begin{array}{ccc|ccc} 3 & -0.1 & -0.2 & 1 & 0 & 0 \\ 0.1 & 7 & -0.3 & 0 & 1 & 0 \\ 0.3 & -0.2 & 10 & 0 & 0 & 1 \end{array}\right]$$

Utilizzando a_{11} come pivot per eliminare l'incognita x_1 dalle righe 2 e 3 si ottiene:

$$\left[\begin{array}{ccc|ccc} 1 & -0.0333333 & -0.0666667 & 0.333333 & 0 & 0 \\ 0 & 7.03333 & -0.293333 & -0.0333333 & 1 & 0 \\ 0 & -0.190000 & 10.0200 & -0.0999999 & 0 & 1 \end{array}\right]$$

Si utilizzi ora come pivot a_{22} per eliminare l'incognita x_2 dalle righe 1 e 3:

$$\left[\begin{array}{ccc|ccc} 1 & 0 & -0.0680570 & 0.333175 & 0.00473933 & 0 \\ 0 & 1 & -0.0417061 & -0.00473933 & 0.142180 & 0 \\ 0 & 0 & 10.0121 & -0.10090 & 0.0270142 & 1 \end{array}\right]$$

Infine utilizzando come pivot a_{33} si elimina l'incognita x_3 dalle righe 1 e 2 e si ottiene automaticamente la matrice inversa:

$$\left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0.332489 & 0.00492297 & 0.00679813 \\ 0 & 1 & 0 & -0.0051644 & 0.142293 & 0.00418346 \\ 0 & 0 & 1 & -0.0100779 & 0.00269816 & 0.0998801 \end{array}\right]$$

Moltiplicando la $\underline{\underline{\mathbf{A}}}^{-1}$ per i due termini noti, si ottengono le soluzioni corrispondenti: $\underline{\mathbf{x}}=\underline{\underline{\mathbf{A}}}^{-1}\underline{\mathbf{c}}$.

Per il primo problema la soluzione numerica è $\underline{\mathbf{x}}=[3.00041,-2.48810,7.00025]^T$ mentre quella esatta è: (x_1=3, x_2=-2.5, x_3=7)
Per il secondo problema la soluzione è $\underline{\mathbf{x}}=[6.99790,7.07411,1.43155]^T$

Interpretazione della matrice inversa:
Calcolo della risposta di un sistema fisico

Come detto, molte delle equazioni che modellano i problemi ingegneristici nascono da una qualche legge di conservazione, ovvero di bilancio, di una certa grandezza di natura fisica (es. conservazione della massa, del calore, della quantità di moto, della carica elettrica, dell'energia). Scrivendo per ogni parte del sistema una equazione di bilancio si perviene ad un sistema di equazioni che modella il comportamento statico (le grandezze di interesse assumono valore costante nel tempo) o dinamico (le grandezze sono funzione della variabile temporale) del sistema: $\underline{\underline{\mathbf{A}}}\,\underline{\mathbf{x}}=\underline{\mathbf{b}}$

In queste equazioni gli elementi di $\underline{\mathbf{x}}$ rappresentano lo stato o la risposta del sistema, cioè le quantità da determinare.

La matrice dei coefficienti $\underline{\underline{\mathbf{A}}}$ contiene i parametri che descrivono l'interazione tra i componenti del sistema o le sue caratteristiche geometriche e/o fisiche.

Il termine al secondo membro $\underline{\mathbf{b}}$ invece dà ragione degli stimoli ovvero dei forzamenti esterni che vengono applicati al sistema e sono, pertanto, quantità note.

Sinora abbiamo visto che esistono diverse tecniche per determinare la risposta del sistema. A quella che fa uso della matrice inversa è possibile attribuire un significato fisico piuttosto interessante:

$$\underline{\mathbf{x}}=\underline{\underline{\mathbf{A}}}^{-1}\underline{\mathbf{b}}$$

e, in modo esplicito, facendo riferimento per fissare le idee ad un sistema con N=3:

$$\begin{aligned} x_1 &= a_{11}^{-1}b_1 + a_{12}^{-1}b_2 + a_{13}^{-1}b_3 \\ x_2 &= a_{21}^{-1}b_1 + a_{22}^{-1}b_2 + a_{23}^{-1}b_3 \\ x_3 &= a_{31}^{-1}b_1 + a_{32}^{-1}b_2 + a_{33}^{-1}b_3 \end{aligned} \qquad (18)$$

Si vede cioè che il generico elemento i-j della matrice inversa rappresenta la risposta della variabile i^{-ma} del sistema quando agisce il solo forzamento j^{-mo}.

Si noti inoltre che le equazioni (18) dipendono in modo lineare dai forzamenti applicati. Vale cioè l'importante **principio di sovrapposizione degli effetti** secondo il quale la risposta di un sistema sottoposto ad un certo numero di forzamenti **indipendenti** può essere ottenuta sommando le risposte ottenute quando i forzamenti agiscono uno per volta.

Per i sistemi lineari vale anche il **principio di proporzionalità** (**omogeneità**) in base al quale se si considera un sistema con un unico ingresso, moltiplicando il forzamento per un valore costante *k*, si ottiene una risposta scalata di un identico fattore *k*.

La matrice inversa nell'analisi del condizionamento dei sistemi

Come accennato, la matrice inversa permette di riconoscere i sistemi mal condizionati.

Allo scopo si possono utilizzare tre metodi:

1. Si normalizza la matrice dei coefficienti $\underline{\underline{\mathbf{A}}}$ in modo che l'elemento più grande di ogni riga sia uguale a 1 e, come visto in precedenza, si determina contemporaneamente la matrice inversa $\underline{\underline{\mathbf{A}}}^{-1}$. Se questa contiene elementi che sono di diversi ordini di grandezza superiori a quelli della matrice originale è probabile che il sistema sia mal-condizionato.

   Se, per una qualunque norma risulta, $\|\underline{\underline{A}}^{-1}\| >> 1$ ***il sistema è mal condizionato***

2. Si moltiplica la matrice inversa per la matrice originale. Se il risultato è molto diverso dalla matrice unità il sistema è mal-condizionato.

   Se $\underline{\underline{\mathbf{A}}}^{-1}\ \underline{\underline{\mathbf{A}}} \neq \underline{\underline{\mathbf{I}}}$ ***il sistema è mal condizionato***

3. Si inverte la matrice inversa e si verifica che la matrice ottenuto sia simile a quella originale. In caso contrario il sistema è mal condizionato.

   Se $(\underline{\underline{\mathbf{A}}}^{-1})^{-1} \neq \underline{\underline{\mathbf{A}}}$ ***il sistema è mal condizionato***

Numero di condizionamento di una matrice

Indichiamo con $\underline{\delta\mathbf{b}}$ il vettore delle perturbazioni (dovuto, come noto, ad errori nei dati o nella rappresentazione dei numeri) del termine noto $\underline{\mathbf{b}}$. Ad esso corrisponderà un vettore di perturbazione nei dati $\underline{\delta\mathbf{x}}$ tale che:

$$\underline{\underline{\mathbf{A}}}\ (\underline{\mathbf{x}} + \underline{\delta\mathbf{x}}) = \underline{\mathbf{b}} + \underline{\delta\mathbf{b}} \quad \rightarrow \quad \underline{\underline{\mathbf{A}}}\ \underline{\delta\mathbf{x}} = \underline{\delta\mathbf{b}} \quad \rightarrow \quad \underline{\underline{\mathbf{A}}}^{-1}\ \underline{\delta\mathbf{b}} = \underline{\delta\mathbf{x}}$$

(si assuma la matrice $\underline{\underline{\mathbf{A}}}$ imperturbata ed invertibile). Considerando una qualunque norma matriciale risulta

Algoritmo per l'inversione di una matrice

L'algoritmo di Gauss-Jacobi può essere modificato per permettere il calcolo della matrice inversa. Allo scopo alla matrice dei coefficienti va aggiunta una matrice unità e alcuni contatori di cicli devono essere modificati.

Nel caso in cui l'algoritmo preveda il pivoting parziale occorrono modifiche più profonde. In pratica se due righe della matrice originaria vengono permutate, occorre eseguire la stessa operazione sulle due colonne corrispondenti della matrice inversa.

Il programma deve essere inoltre in grado di calcolare le soluzioni del sistema per un numero arbitrario di termini noti. Allo scopo occorre aggiungere alla fine del calcolo dell'inversa un nuovo ciclo che ad ogni passaggio

1) richiede un nuovo vettore di termini noti
2) moltiplica la matrice inversa per questo vettore
3) restituisce la soluzione.

METODI DIRETTI: FATTORIZZAZIONE DI UNA MATRICE

Una **matrice sparsa** è una matrice nella quale i termini nulli sono prevalenti su quelli diversi da zero. Una matrice viene detta **bandata** se i termini non nulli sono addensati attorno alla diagonale principale. Il **pattern** ovvero la **struttura** della matrice è una rappresentazione della matrice nella quale vengono evidenziati i termini non nulli.

Da un punto di vista pratico capita spesso di incontrare problemi che richiedono la soluzione di diversi sistemi lineari di tipo sparso per i quali **la struttura della matrice non subisce modifiche**, **mentre possono variare sia i coefficienti del sistema che i termini noti.**

In questi casi l'approccio risolutivo generalmente adoperato passa attraverso tre stadi che vanno sotto i nomi di *fattorizzazione simbolica*, *fattorizzazione numerica* della matrice e *soluzione del sistema*.

La **fattorizzazione simbolica** viene eseguito una volta sola, all'inizio del processo solutivo e il suo scopo è quello di determinare i legami tra le caratteristiche di sparsità delle matrici **A** e quelle delle matrici risultato del processo di fattorizzazione.

Il processo di **fattorizzazione numerica** va invece ripetuto ogni volta che un nuovo sistema deve essere risolto. Tuttavia le informazioni acquisite durante la fase di fattorizzazione simbolica fanno sì che quest'ultima operazione risulti estremamente veloce

Per i metodi a banda o di inviluppo, caratterizzati da sparsità strutturali, i blocchi contenenti elementi non nulli sono semplicemente identici e, pertanto, il processo di fattorizzazione simbolica può essere evitato.

Nel caso dei metodi di sparsità generali, invece i legami tra **A** e **U** sono molto meno intuitivi e dipendono oltre che dallo schema di fattorizzazione (che determina il cosiddetto *fill-in* o riempimento della matrice) anche dalla numerazione delle incognite interagenti del problema (si pensi alle formulazioni 2D e 3D di tipo FEM). Per questo motivo i metodi di sparsità generale vengono usualmente preceduti da algoritmi -il più famoso dei quali è quello di Cuthill-McKee- capaci di minimizzare il *fill-in* attraverso il riordinamento della numerazione delle incognite.
Questo insieme di operazioni può essere molto costoso dal punto di vista computazionale perciò la fase preliminare di fattorizzazione simbolica produce un consistente vantaggio.

Durante la soluzione di sistemi lineari ben difficilmente si opera direttamente sulla matrice di partenza. Molto più frequentemente, infatti, si preferisce decomporla in matrici di struttura più semplice -diagonale o triangolare-, per le quali esistono implementazioni del procedimento di eliminazione di Gauss particolarmente efficienti.

Alcuni modi per eseguire la fattorizzazione di una matrice

Tra le varie possibili strategie di fattorizzare una generica matrice **A** citiamo:

1) La decomposizione **LU** dove **L** è una matrice triangolare inferiore, e **U** una matrice triangolare superiore.

2) La decomposizione $\mathbf{LDM}^{\mathbf{T}}$, dove **L** ed **M** sono matrici triangolari inferiori con elementi unitari sulla diagonale principale e **D** è una matrice diagonale.

3) Le decomposizioni di Doolitle e di Crout, analoghe alla decomposizione **LU**, con la differenza che **L** nel metodo di Doolitle ed **U** nel metodo di Crout presentano elementi tutti unitari sulla diagonale principale.

4) La decomposizione, possibile solo se **A** è simmetrica, $\mathbf{LDL}^{\mathbf{T}}$, analoga alla *2)* con **M**=**L.**

5) La decomposizione $\mathbf{RR}^{\mathbf{T}}$, detta di Cholesky, possibile se **A** è simmetrica e definita positiva, con **R** matrice triangolare inferiore con elementi tutti positivi sulla diagonale.

LIMITI DEI METODI DIRETTI BASATI SULLE FATTORIZZAZIONI

Esistono due grossi limiti all'utilizzo dei metodi diretti:

1) Le decomposizioni e il pivoting, in generale, distruggono la struttura originaria della matrice, e ciò risulta particolarmente fastidioso per matrici di tipo sparso. Questa osservazione non è valida per le matrici bandate a dominanza diagonale e per le matrici bandate simmetriche e definite positive che non necessitano dell'operazione di pivoting e possono essere decomposte con algoritmi che conservano la struttura bandata anche nelle matrici fattori,
2) Un limite ulteriore all'impiego dei metodi diretti è l'ingente costo computazionale associato al processo di fattorizzazione della matrice.

   Per matrici piene esso risulta proporzionale a N^3 (in particolare, se si adotta la fattorizzazione di Cholesky vale $N^3/3$), dove N è il numero delle incognite, che, nei problemi FEM di interesse pratico, risulta tipicamente dell'ordine di 1E3-1E5. Una stima del tempo occorrente per risolvere un sistema lineare siffatto si ottiene facilmente tenendo conto che la velocità operativa degli attuali Pcs è attualmente dell'ordine dei Gigaflops (1 Gigaflop corrisponde ad un miliardo di moltiplicazioni per secondo). Il tempo di elaborazione richiesto per l'inversione di un problema di interesse pratico può quindi essere dell'ordine dei mesi, il che è, ovviamente, assolutamente inaccettabile. Se le matrici sono di tipo sparso, l'impiego di metodi di banda rende il costo del processo di fattorizzazione proporzionale a $M_A \cdot n^2$ dove M_A è l'ampiezza di banda della matrice il cui ordine è tipicamente inferiore a 100; seppure l'impiego di metodi di inviluppo renda il costo computazionale della fattorizzazione ancora minore, questa tuttavia conserva un peso notevole nell'economia complessiva del processo solutivo. Risulta per questo motivo conveniente adoperare i metodi diretti nei casi in cui l'ordine del sistema non sia particolarmente elevato e/o quando il problema richieda di risolvere un grande numero di sistemi di equazione, nei quali il termine noto cambi mentre la matrice dei coefficienti si conservi costante; in casi diversi, possono essere preferiti schemi iterativi di soluzione.

Metodi iterativi

Nel caso di matrici dotate di sparsità non strutturale e di dimensione elevata, un'alternativa ai metodi diretti è costituita dagli **schemi di soluzione iterativi** che lasciano inalterata la struttura originale della matrice e non richiedono il processo di fattorizzazione.
Essendo, tuttavia, in questo caso la soluzione ottenuta come limite di una successione, per essere un'alternativa valida ai metodi diretti, essi hanno bisogno di **opportune tecniche di accelerazione**.

L'idea comune a tutti i metodi iterativi è la seguente:

1) Si dà una stima del vettore soluzione $\underline{\mathbf{x}}^{(0)}$ del sistema $\underline{\underline{\mathbf{A}}}\,\underline{\mathbf{x}} = \underline{\mathbf{b}}$, dove $\underline{\underline{\mathbf{A}}}$ è una matrice sparsa di ordine N non singolare.
2) Si costruisce una successione di soluzioni $\underline{\mathbf{x}}^{(k)}$ attraverso la successiva risoluzione di sistemi lineari più semplici.

In forma generale consideriamo una decomposizione della matrice $\underline{\underline{\mathbf{A}}}$ della forma:

$$\underline{\underline{\mathbf{A}}} = \underline{\underline{\mathbf{M}}} - \underline{\underline{\mathbf{N}}} \qquad (1)$$ con $\underline{\underline{\mathbf{M}}}$ non singolare.

Si ha allora:

$$\underline{\underline{\mathbf{A}}}\,\underline{\mathbf{x}} = \underline{\mathbf{b}} \Leftrightarrow \underline{\underline{\mathbf{M}}}\,\underline{\mathbf{x}} = \underline{\underline{\mathbf{N}}}\,\underline{\mathbf{x}} + \underline{\mathbf{b}} \qquad (2)$$

da cui il procedimento iterativo:

$$\underline{\underline{\mathbf{M}}}\,\underline{\mathbf{x}}^{(k+1)} = \underline{\underline{\mathbf{N}}}\,\underline{\mathbf{x}}^{(k)} + \underline{\mathbf{b}} \qquad (3)$$

La matrice

$$\underline{\mathbf{B}} = \underline{\underline{\mathbf{M}}}^{-1}\,\underline{\underline{\mathbf{N}}} = \underline{\underline{\mathbf{M}}}^{-1}\,(\underline{\underline{\mathbf{M}}} - \underline{\underline{\mathbf{A}}}) = \underline{\underline{\mathbf{I}}} - \underline{\underline{\mathbf{M}}}^{-1}\,\underline{\underline{\mathbf{A}}} \qquad (4)$$

è detta *matrice di iterazione*; essa individua un particolare metodo e le sue proprietà sono fondamentali per la qualità e la rapidità della convergenza del processo iterativo.
Facendo riferimento alla decomposizione $\underline{\underline{\mathbf{A}}} = \underline{\underline{\mathbf{D}}} - \underline{\underline{\mathbf{E}}} - \underline{\underline{\mathbf{F}}}$:

$$\mathbf{A} = \begin{bmatrix} \ddots & & & \\ & \ddots & -\mathbf{F} & \\ & \mathbf{D} & & \\ -\mathbf{E} & & \ddots & \\ & & & \ddots \end{bmatrix} \qquad (5)$$

Fig.8.4: Decomposizione della matrice $\underline{\underline{\mathbf{A}}}$ secondo gli schemi di Jacobi e Gauss-Seidel

citiamo alcuni tra gli schemi iterativi più comunemente impiegati:
-Il metodo di Jacobi, o *delle sostituzioni simultanee* ove si assume

$$\underline{\underline{M}}=\underline{\underline{D}}; \qquad \underline{\underline{N}}=\underline{\underline{E}}+\underline{\underline{F}}; \qquad (6)$$

e quindi: $$\mathbf{B} = \underline{\underline{D}}^{-1}(\underline{\underline{E}}+\underline{\underline{F}}) = \underline{\underline{I}} - \underline{\underline{D}}^{-1}\underline{\underline{A}} \qquad (7)$$

-Il metodo di Gauss Seidel, o *delle sostituzioni successive* ove si assume

$$\mathbf{M}= \underline{\underline{D}} - \underline{\underline{E}}; \qquad \underline{\underline{N}}= \underline{\underline{F}}; \qquad (8)$$

e quindi: $$\underline{\underline{B}} = (\underline{\underline{D}} - \underline{\underline{E}})^{-1}\underline{\underline{F}} = (\mathbf{I} - \underline{\underline{D}}^{-1}\underline{\underline{E}})^{-1}\underline{\underline{D}}^{-1}\underline{\underline{F}} \qquad (9)$$

Metodi iterativi: Metodo di Gauss-Seidel

Analizziamo in maggior dettaglio il metodo iterativo di Gauss-Seidel.
Il sistema da risolvere sia :

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

Se gli elementi della diagonale principale sono tutti non nulli è possibile risolvere la prima equazione rispetto a x_1, la seconda rispetto a x_2, etc.:

$$x_1 = \frac{c_1 - a_{12}x_2 - a_{13}x_3 - a_{1N}x_N}{a_{11}} \qquad (10.1)$$

$$x_2 = \frac{c_2 - a_{21}x_1 - a_{23}x_3 - a_{2N}x_N}{a_{22}} \qquad (10.2)$$

$$x_N = \frac{c_N - a_{N2}x_2 - a_{N3}x_3 - a_{N,N-1}x_{N-1}}{a_{NN}} \qquad (10.N)$$

ALGORITMO

1. Si scelgano i valori iniziali delle variabili $x_j^{(0)}$ j=1,…,N. (Per esempio, assumiamo che siano tutte nulle).
2. Questi valori, sostituiti nella (10.1) forniscono il nuovo valore di $x_1^{(1)}$.
3. Sostituendo nella (10.2) $x_1^{(1)}$, $x_1^{(0)}$,…,$x_j^{(0)}$ …,$x_N^{(0)}$ si determina $x_2^{(1)}$.
4. Procedendo nello stesso modo si determinano uno per volta i valori di $x_3^{(1)}$,…,$x_N^{(1)}$
5. Si verifica se per ogni i l'errore relativo percentuale approssimato risulta minore di una tolleranza preassegnata:

$$\varepsilon_{a,i} = \frac{x_i^j - x_i^{j-1}}{x_i^j} < \varepsilon_S \,, \quad \forall\ i = 1,..., N$$

6. Se la condizione è verificata si è trovata la soluzione. In caso contrario si torna al punto 2.

N.B. A differenza del metodo di Seidel, nel metodo di Jacobi, si calcolano tutte le $\mathbf{x_i^{(1)}}$ i=1,…,N utilizzando le $\mathbf{x_i^{(0)}}$. In altre parole, i nuovi valori generati non vengono utilizzati immediatamente ma vengono conservati per l'iterazione successiva.

Esempio: Si risolva il sistema precedente utilizzando il metodo di Gauss-Seidel. Si lavori con 10 cifre significative. Il sistema è riportato di seguito.
La soluzione esatta è (x_1=3,x_2=-2.5,x_3=7)

$$\begin{aligned} 3\,x_1 - 0.1\,x_2 - 0.2\,x_3 &= 7.85 \\ 0.1\,x_1 + 7\,x_2 - 0.3\,x_3 &= -19.3 \\ 0.3\,x_1 - 0.2\,x_2 + 10\,x_3 &= 71.4 \end{aligned}$$

Soluzione: Applicando l'algoritmo precedentemente stabilito risulta:

$$x_1 = \frac{7.85 + 0.1*0 + 0.2*0}{3} = 2.616666667$$

$$x_2 = \frac{-19.3 - 0.1*2.616666667 + 0.3*0}{7} = -2.794523810$$

$$x_3 = \frac{71.4 - 0.3*2.616666667 + 0.2*(-2.794523810)}{10} = 7.005609524$$

Ovviamente l'errore è del 100%. Nella seconda iterazione risulta:

$$x_1 = \frac{7.85 + 0.1*(-2.794523810) + 0.2*7.005609524}{3} = 2.990556508$$

$$x_2 = \frac{-19.3 - 0.1*2.990556508 + 0.3*7.005609524}{7} = -2.499624684$$

$$x_3 = \frac{71.4 - 0.3*2.990556508 + 0.2*(-2.499624684)}{10} = 7.00029081$$

Gli errori relativi percentuali approssimati sono:

$$\varepsilon_{a,1} = \left|\frac{x_1^2 - x_1^1}{x_1^2}\right| *100\% = \frac{2.990556508 - 2.616666667}{2.990556508} *100\% = 12.5\%$$

$$\varepsilon_{a,2} = \left|\frac{x_2^2 - x_2^1}{x_2^2}\right| *100\% = \left|\frac{-2.499624684 + 2.794523810}{-2.499624684}\right| *100\% = 11.8\%$$

$$\varepsilon_{a,3} = \left|\frac{x_3^2 - x_3^1}{x_3^2}\right| *100\% = \left|\frac{7.000290811 - 7.005609524}{7.000290811}\right| *100\% = 0.076\%$$

Si noti che sono tutti superiori agli errori percentuali relativi veri che valgono rispettivamente: ε_{t1}=0.31%; ε_{t2}=0.015%; ε_{t3}=0.0042%.
La stima dell'errore è pertanto conservativa.

Criterio di convergenza per il metodo di Gauss-Seidel

Il metodo di Gauss-Seidel utilizza una strategia simile a quella impiegata nel metodo delle sostituzioni successive.
Ricordiamo che questa tecnica presenta due punti deboli:

- **non sempre converge**
- **se converge, la convergenza è spesso molto lenta**

Il metodo di Gauss-Seidel può presentare difetti analoghi.

Un criterio sufficiente per la convergenza è il seguente:

I coefficienti sulla diagonale principale devono essere maggiori in valore assoluto della somma dei valori assoluti dei termini fuori della diagonale principale che si trovano sulla stessa riga, ovvero, con espressione matematica, il sistema deve essere **a dominanza diagonale**:

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

Per fortuna molti sistemi che modellano problemi ingegneristici godono di questa proprietà

N.B. Se la proprietà non è soddisfatta il metodo può convergere, ma la convergenza non è garantita.

Tecniche di accelerazione dei metodi iterativi

Con l'obiettivo di accelerare il processo di convergenza i metodi precedenti possono essere modificati introducendo un parametro reale ω e definendo la componente i^{-ma} della soluzione al $(k+1)^{-mo}$ passo iterativo come:

$$\underline{x}_i^{(k+1)} = \omega\, \underline{x}_i^{(k+1/2)} + (1-\omega)\, \underline{x}_i^{(k)}$$

In questo modo, partendo dal metodo di Jacobi si perviene al seguente schema:

$$\underline{x}^{(k+1)} = (\mathbf{I} - \omega \underline{\underline{\mathbf{D}}}^{-1} \underline{\underline{\mathbf{A}}})\, \underline{x}^{(k)} + \omega\, \underline{\underline{\mathbf{D}}}^{-1} \underline{b}$$

Mentre partendo dal metodo di Gauss-Seidel si giunge al metodo di rilassamento (detto anche SOR) in cui si assume:

$$\underline{\underline{\mathbf{M}}} = \left(\frac{\underline{\underline{\mathbf{D}}}}{\omega} - \underline{\underline{\mathbf{E}}} \right); \qquad \underline{\underline{\mathbf{N}}} = \left(\frac{1}{\omega} - 1 \right) \underline{\underline{\mathbf{D}}} + \underline{\underline{\mathbf{F}}};$$

a cui corrisponde la seguente matrice di iterazione:

$$\mathbf{B}_\omega = (\underline{\underline{\mathbf{D}}} - \omega \underline{\underline{\mathbf{E}}})^{-1} [\, (1-\omega)\, \underline{\underline{\mathbf{D}}} + \omega\, \underline{\underline{\mathbf{F}}}\,]$$

Circa la convergenza di questi schemi iterativi della successione $x^{(k)}$ alla soluzione x_{TRUE} valgono i seguenti teoremi:

- -Se **A** è una matrice di ordine N a stretta predominanza diagonale, allora i metodi di Jacobi e di Gauss-Seidel sono convergenti.
- -Se **A** è una matrice hermitiana definita positiva, allora il metodo di rilassamento converge se e solo se $0<\omega<2$.

In generale:

- se $\omega=1$ non c'è accelerazione
- se $\omega<1$ (sottorilassamento) si può rendere convergente un sistema che non lo è
- se $2>\omega>1$ (sovrarilassamento) si può accelerare la velocità convergenza di sistemi già convergenti.

Per quanto riguarda la velocità di convergenza, essa dipende in maniera critica (e questo è il più grande inconveniente del metodo SOR) dalla scelta del fattore di rilassamento. Young, a questo proposito, ha mostrato come la scelta ottimale sia:

$$\omega_b = \frac{2}{1 + (1 - z^2)^{\frac{1}{2}}}$$

dove *z* è il più grande autovalore di $\underline{\underline{\mathbf{D}}}^{-1}$ ($\underline{\underline{\mathbf{E}}}$+$\underline{\underline{\mathbf{F}}}$).

ALGORITMO DEL METODO DI GAUSS-SEIDEL CON RILASSAMENTO

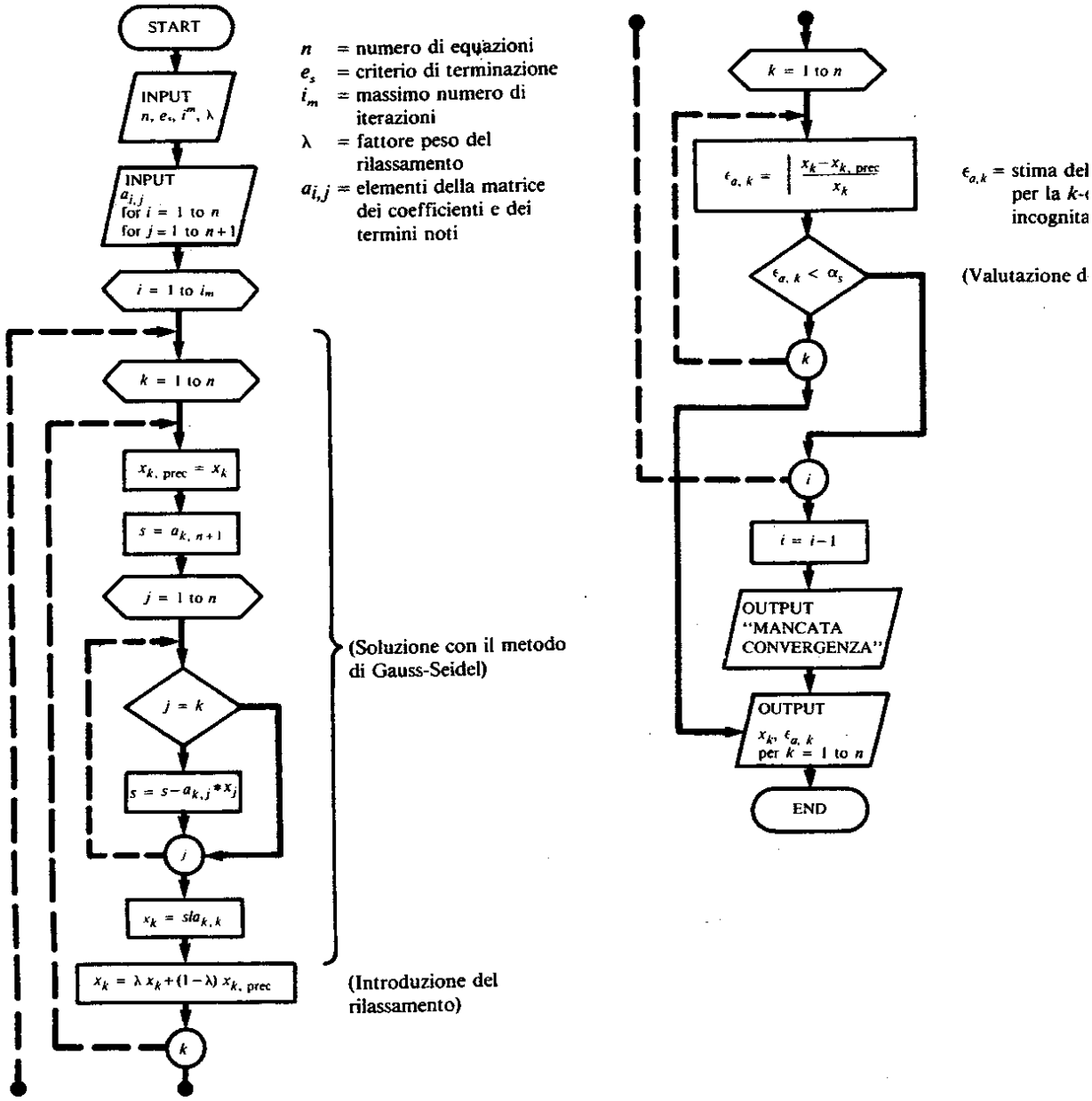


Figura 8.6 Diagramma di flusso per il metodo di Gauss-Seidel con rilassamento.

L'algoritmo di Gauss-Seidel funziona solo per sistemi a dominanza diagonale. Per cercare di superare questa limitazione, si cerca all'interno della prima equazione il coefficiente più grande e si risolve l'equazione rispetto all'incognita associata a questo coefficiente. Si usa lo stesso procedimento per le restanti equazioni.

Questa tecnica aumenta la probabilità di avere un sistema a dominanza diagonale, ma non garantisce la convergenza di sistemi lontani da questa condizione.

Esempio: Si usi il metodo di Gauss-Seidel con sovra-rilassamento per risolvere il seguente sistema hermitiano definito positivo. Si applichi la formula di Young per determinare il valore ottimale di ω.

La soluzione esatta è (x_1=10.13208, x_2=-12.46548, x_3=11.92355)

$$\begin{aligned} 3\,x_1 + 2.0\,x_2 + 0.2\,x_3 &= 7.85 \\ 2.0\,x_1 + 7\,x_2 + 4.\,x_3 &= -19.3 \\ 0.2\,x_1 + 4.\,x_2 + 10\,x_3 &= 71.4 \end{aligned}$$

L'auto-valore più grande della matrice $\underline{\underline{\mathbf{D}}}^{-1}(\underline{\underline{\mathbf{E}}}+\underline{\underline{\mathbf{F}}})$ è λ=0.66578 cui corrisponde un valore ottimale di ω=1.14538

Assumendo come soluzione iniziale x_0=(0,0,0), dopo 10 iterazioni

con Gauss-Seidel si ottiene	(10.12350,-12.46022,11.92162),
col metodo SOR si ha	(10.13122,-12.46495,11.92335)
La soluzione esatta è	(10.13208, -12.46548, 11.92355)

Si vede che il SOR fornisce le migliori performances

Metodi iterativi: Metodo del gradiente coniugato

Un altro schema iterativo che può essere utilizzato vantaggiosamente nella soluzione di problemi FEM è quello del **Gradiente Coniugato**.
Con questo metodo è teoricamente possibile risolvere esattamente un sistema di ordine N, al più in N iterazioni (in pratica, a causa degli errori di arrotondamento, questa proprietà viene persa, ed inoltre N iterazioni per pervenire alla soluzione sono troppe).
Il metodo del gradiente coniugato sfrutta il fatto che risolvere il sistema

$$\underline{\underline{\mathbf{A}}}\,\underline{\mathbf{x}} = \underline{\mathbf{b}} \qquad (1)$$

è equivalente a minimizzare il funzionale quadratico:

$$\mathbf{Q}(\underline{\mathbf{x}}) = \frac{1}{2}\underline{\mathbf{x}}^{\mathrm{T}}\underline{\underline{\mathbf{A}}}\underline{\mathbf{x}} - \underline{\mathbf{x}}^{\mathrm{T}}\underline{\mathbf{b}}$$

o equivalentemente

$$\mathbf{E}(\underline{\mathbf{x}}) = (\underline{\mathbf{x}} - \underline{\mathbf{x}}_{\mathbf{T}})^{\mathrm{T}}\underline{\underline{\mathbf{A}}}(\underline{\mathbf{x}} - \underline{\mathbf{x}}_{\mathbf{T}}) = \underline{\mathbf{e}}^{\mathrm{T}}\underline{\underline{\mathbf{A}}}\underline{\mathbf{e}} \qquad (2)$$

dove $\underline{\mathbf{x}}_{\mathbf{T}}$ è la soluzione esatta del sistema originale, ed $\underline{\mathbf{e}}=\underline{\mathbf{x}}-\underline{\mathbf{x}}_{\mathbf{T}}$.
$\underline{\underline{\mathbf{A}}}$ deve essere necessariamente definita positiva affinchè $\mathbf{E}(\underline{\mathbf{x}})$ sia minimo per $\underline{\mathbf{e}}=\mathbf{0}$.
La minimizzazione di (2) viene condotta per via iterativa scegliendo al passo k una direzione $\underline{\mathbf{p}}_{k-1}\neq 0$ e uno scalare α_k in maniera che, posto

$$\underline{\mathbf{x}}_k = \underline{\mathbf{x}}_{k-1} + \alpha_k \underline{\mathbf{p}}_{k-1} \qquad (3)$$

si abbia $\mathbf{E}(\underline{\mathbf{x}}_k) \leq \mathbf{E}(\underline{\mathbf{x}}_{k-1})$.
Fissata la direzione $\underline{\mathbf{p}}_{k-1}$ e definito il residuo $\mathbf{r}_k(\underline{\mathbf{x}}) = \underline{\mathbf{b}} - \underline{\underline{\mathbf{A}}}\underline{\mathbf{x}}_{k-1} = \underline{\underline{\mathbf{A}}}\,(\underline{\mathbf{x}}_{\mathbf{T}}-\underline{\mathbf{x}}_{k-1})$ si dimostra che la scelta ottimale per α_k è

$$\alpha_k = \frac{(\underline{\mathbf{r}}_k, \underline{\mathbf{p}}_{k-1})}{(\underline{\underline{\mathbf{A}}}\underline{\mathbf{p}}_{k-1}, \underline{\mathbf{p}}_{k-1})} \qquad (4)$$

Per quel che concerne la scelta delle direzioni $\underline{\mathbf{p}}_k$ ricordiamo che due vettori $\underline{\mathbf{x}}$ e $\underline{\mathbf{y}}$ si dicono coniugati rispetto all'applicazione $\underline{\underline{\mathbf{A}}}$ se vale la relazione $(\underline{\underline{\mathbf{A}}}\,\underline{\mathbf{x}}, \underline{\mathbf{y}})=0$.
Il nome dell'algoritmo è così legato al criterio di ortogonalità rispetto ai prodotto scalare $(\underline{\underline{\mathbf{A}}}\,\underline{\mathbf{p}}_i, \underline{\mathbf{p}}_k)$ $\forall i=1, k-1$ col quale, al $k^{\text{-mo}}$ passo di iterazione, viene scelta la direzione lungo le quali effettuare la minimizzazione.
Da un punto di vista pratico, se si adotta per α_k la scelta (4), la direzione $\underline{\mathbf{p}}_k$ è determinata attraverso le relazioni:

$$\underline{\mathbf{p}}_k = \underline{\mathbf{r}}_k + \beta_k \underline{\mathbf{p}}_{k-1} \; ; (5) \qquad \text{dove} \qquad \beta_\kappa = -\frac{(\underline{\underline{\mathbf{A}}}\underline{\mathbf{p}}_{\kappa-1}, \underline{\mathbf{r}}_\kappa)}{(\underline{\underline{\mathbf{A}}}\underline{\mathbf{p}}_{\kappa-1}, \underline{\mathbf{p}}_{\kappa-1})} \qquad (6)$$

Interpretazione geometrica del metodo

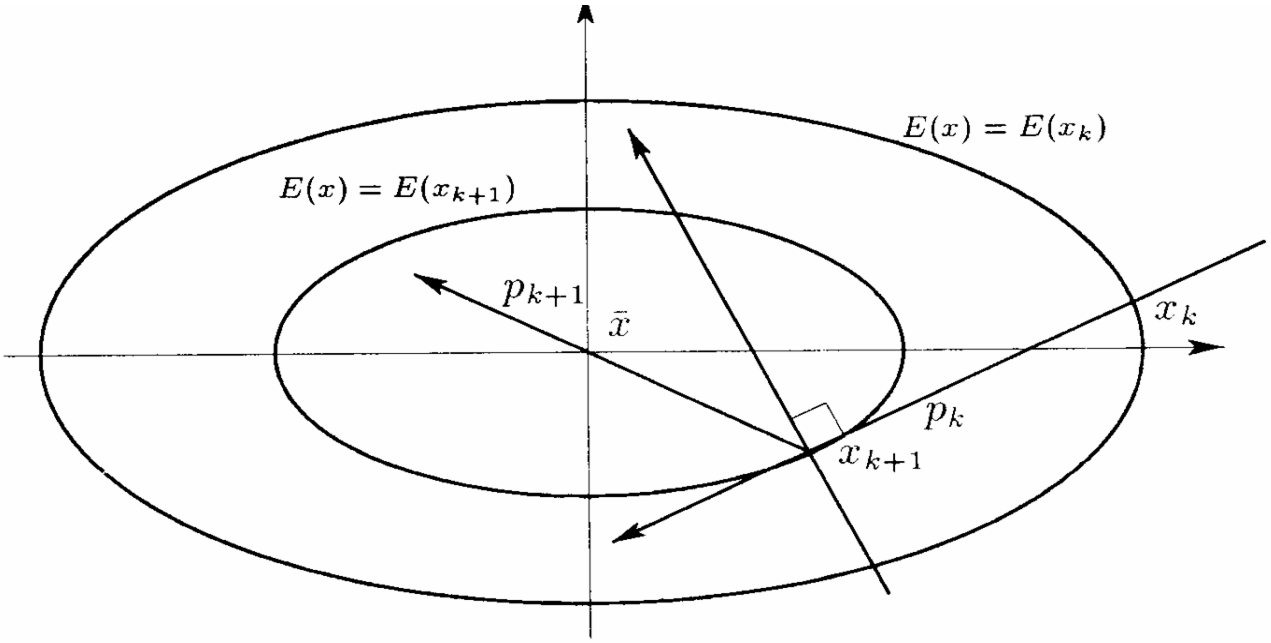


Figura 5.8. Direzioni del gradiente e del gradiente coniugato.

Nel caso N=2, $\underline{\mathbf{x}}=\{x_1,x_2\}$ ed è possibile dare un'interpretazione geometrica del metodo. L'equazione $E(\underline{\mathbf{x}})$=cost. è dal punto di vista geometrico un 'ellisse. Al variare quindi di $\underline{\mathbf{x}}_k$ si ottiene quindi una famiglia di ellissi concentriche $E(\underline{\mathbf{x}})=E(\underline{\mathbf{x}}_k)$, con centro in $\underline{\mathbf{x}}_T$ (punto nel quale si raggiunge il valore minimo per il funzionale $E(\underline{\mathbf{x}})$). Si scelga ad arbitrio un punto di partenza $\underline{\mathbf{x}}_1$ e un vettore direzione $\underline{\mathbf{p}}_1$.

Usando la relazione (4) si ottiene il valore di α_k per il quale si ottiene il valore minimo per il funzionale $E(\underline{\mathbf{x}}_2)$ tra tutte le nuovi possibili soluzioni $\underline{\mathbf{x}}_2 = \underline{\mathbf{x}}_1 + \alpha_1 \underline{\mathbf{p}}_1$ raggiungibili spostandosi da $\underline{\mathbf{x}}_1$ lungo la direzione $\underline{\mathbf{p}}_1$.

Si vede in figura che in questo punto, il vettore $\underline{\mathbf{p}}_1$ risulta tangente all'ellisse $E(\underline{\mathbf{x}})=E(\underline{\mathbf{x}}_2)$.

La nuova direzione di spostamento, $\underline{\mathbf{p}}_2$, deve essere coniugata rispetto a $\underline{\mathbf{p}}_1$, ovvero:

$$(\underline{\underline{\mathbf{A}}}\,\underline{\mathbf{p}}_1, \underline{\mathbf{p}}_2)=0.$$

È possibile dimostrare che, avendo supposti nulli gli errori di arrotondamento, la direzione $\underline{\mathbf{p}}_2$ deve puntare direttamente verso la soluzione $\underline{\mathbf{x}}_T$.

Allo scopo basta osservare che, poiché N=2, i due vettori coniugati (e pertanto indipendenti) $\underline{\mathbf{p}}_1,\underline{\mathbf{p}}_2 \in \Re^2$ costituiscono una base per la soluzione $\underline{\mathbf{x}}_T \in \Re^2$.

Dalla minimizzazione del funzionale si ottengono i coefficienti α_1,α_2 tali per cui

$$\underline{\mathbf{x}}_T = \underline{\mathbf{x}}_1 + \alpha_1 \underline{\mathbf{p}}_1 + \alpha_2 \underline{\mathbf{p}}_2$$

COSTO COMPUTAZIONALE DEL GRADIENTE CONIUGATO

Per quanto riguarda il numero di operazioni richieste si vede che il costo essenziale è dovuto al calcolo di $\underline{\underline{\mathbf{A}}}\,\underline{\mathbf{p}}$, pari a N^2 per matrici piene ed al numero degli elementi non nulli della matrice per matrici sparse non strutturate.

Se il numero di iterazioni è N, il costo totale per matrici piene risulta pari a N^3 superiore a quello di $N^3/3$ richiesto nel processo di fattorizzazione di Cholesky.

Appare chiaro, dunque, che il metodo diventa interessante quando **la matrice è sparsa** e/o quando **il numero di iterazioni necessarie è decisamente inferiore a N**.

CONDIZIONAMENTO DI UNA MATRICE

Allo scopo di ridurre il numero di iterazioni risulta molto spesso di grande efficacia **precondizionare** la matrice $\underline{\underline{\mathbf{A}}}$ attraverso la moltiplicazione con una matrice non-singolare $\underline{\underline{\mathbf{R}}}$ quanto più possibile vicina ad $\underline{\underline{\mathbf{A}}}^{-1}$. In questo modo l'algoritmo del gradiente coniugato convergerà molto più velocemente alla soluzione del sistema equivalente:

$$\underline{\underline{\mathbf{R}}}^{-1}\underline{\underline{\mathbf{A}}}(\underline{\underline{\mathbf{R}}}^{-1})^{T}(\underline{\underline{\mathbf{R}}}^{T}\underline{\mathbf{x}})=(\underline{\underline{\mathbf{R}}}^{-1}\underline{\mathbf{b}}) \Leftrightarrow \quad \underline{\underline{\mathbf{R}}}^{-1}\underline{\underline{\mathbf{A}}}\,\underline{\mathbf{x}}=(\underline{\underline{\mathbf{R}}}^{-1}\underline{\mathbf{b}}) \Leftrightarrow \quad \underline{\underline{\mathbf{R}}}^{-T}\underline{\underline{\mathbf{R}}}^{-1}\underline{\underline{\mathbf{A}}}\,\underline{\mathbf{x}}=\underline{\underline{\mathbf{R}}}^{-T}(\underline{\underline{\mathbf{R}}}^{-1}\underline{\mathbf{b}})$$

Posto $\underline{\underline{\mathbf{M}}}=\underline{\underline{\mathbf{R}}}\,\underline{\underline{\mathbf{R}}}^{T}$, con $\underline{\underline{\mathbf{R}}}$ non singolare, $\underline{\underline{\mathbf{M}}}$ risulta simmetrica e definita positiva.

$$\underline{\underline{\mathbf{M}}}^{-1}\underline{\underline{\mathbf{A}}}\underline{\mathbf{x}}=(\underline{\underline{\mathbf{M}}}^{-1}\underline{\mathbf{b}})=\underline{\mathbf{s}} \qquad (7)$$

Il calcolo di $\underline{\mathbf{s}}$ deve essere eseguito ad ogni iterazione e non deve, naturalmente, passare per l'inversione di $\underline{\underline{\mathbf{M}}}$.

È pertanto essenziale scegliere $\underline{\underline{\mathbf{M}}}$ in modo tale che il sistema lineare $\underline{\underline{\mathbf{M}}}\,\underline{\mathbf{s}}=\underline{\mathbf{b}}$ possa essere risolto in maniera economica. Tra le varie possibili scelte di **M** si ricordano:

- $\underline{\underline{\mathbf{M}}}=\underline{\underline{\mathbf{D}}}$ che dà origine al *metodo del gradiente coniugato di Jacobi*.
- $\underline{\underline{\mathbf{M}}}=\dfrac{1}{\omega(2-\omega)}\left(\underline{\underline{\mathbf{D}}}-\omega\,\underline{\underline{\mathbf{F}}}^{T}\right)\underline{\underline{\mathbf{D}}}^{-1}\left(\underline{\underline{\mathbf{D}}}-\omega\,\underline{\underline{\mathbf{E}}}^{T}\right)$, usata nel metodo SSOR, ove, a differenza del metodo SOR, la scelta di ω non è più critica.
- Se per l'immagazzinamento di $\underline{\underline{\mathbf{A}}}$ si è adoperato un metodo di matrici sparse, allora può essere conveniente calcolare $\underline{\underline{\mathbf{R}}}$ attraverso una fattorizzazione parziale di Cholesky di $\underline{\underline{\mathbf{A}}}$ con un metodo di eliminazione Gaussiana, in cui il *fill-in* è trascurato, oppure portato in conto solo in zone limitate della matrice fattorizzata (p.e. all'interno di un fissato numero di diagonali localizzate attorno a quella principale).

Allo scopo di limitare l'insorgere di mal-condizionamenti per effetto degli arrotondamenti numerici e di evitare il fallimento del processo di fattorizzazione, può risultare conveniente scalare la matrice $\underline{\underline{\mathbf{A}}}$ in modo da riportare i termini diagonali a valori unitari e, eventualmente, moltiplicare i termini fuori diagonale per il fattore $1/(1+\alpha)$, ove α è un numero reale positivo.

Esempio: Risolvere il seguente sistema utilizzando il gradiente coniugato:

$$\begin{aligned} 10\,x_1 + 2\,x_2 + 5\,x_3 &= 34 \\ 2\,x_1 + 7\,x_2 + 3\,x_3 &= 28 \\ 5\,x_1 + 3\,x_2 + 9\,x_3 &= 47 \end{aligned}$$

La soluzione esatta è (x_1=1, x_2=2, x_3=4)

Si ponga x_0=(0,0,0) e p_0=(1,0,0)
Al primo step risulta $\mathbf{r}_1(\underline{\mathbf{x}}) = \underline{\mathbf{b}} - \underline{\underline{\mathbf{A}}}\underline{\mathbf{x}}_0$ =(34, 28, 47)

$$\alpha_1 = \frac{(\underline{\mathbf{r}}_1, \underline{\mathbf{p}}_0)}{(\underline{\underline{\mathbf{A}}}\underline{\mathbf{p}}_0, \underline{\mathbf{p}}_0)} = 3.4$$

$$\mathbf{x_1} = \mathbf{x_0} + \boldsymbol{\alpha}_1\mathbf{p_0} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + 3.4 \begin{bmatrix} 1. \\ 0 \\ 0 \end{bmatrix}$$

$$\beta_1 = -\frac{(\underline{\underline{\mathbf{A}}}\underline{\mathbf{p}}_0, \underline{\mathbf{r}}_1)}{(\underline{\underline{\mathbf{A}}}\underline{\mathbf{p}}_0, \underline{\mathbf{p}}_0)} = -63.1$$

$$\mathbf{p_1} = \mathbf{r_1} + \boldsymbol{\beta}_1\mathbf{p_0} = \begin{bmatrix} 34 \\ 28 \\ 47 \end{bmatrix} - 63.1 \begin{bmatrix} 1. \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -29.1 \\ 28 \\ 47 \end{bmatrix}$$

Dopo 7 iterazioni si perviene alla soluzione

$\underline{\mathbf{x}}_7$=(1.0006,2.0051,3.9962).

Si noti che se si continuasse ad iterare, per effetto delle approssimazioni numeriche, la soluzione peggiorerebbe.

Tabella riassuntiva dei metodi per la so a dominanza diagonale sistemi lineari

Tabella III.2 Confronto tra i vari metodi per la risoluzione di sistemi di equazioni lineari algebriche.

Metodo	Numero massimo approssimato di equazioni	Stabilità	Precisione	Applicabilità	Programmabilità	Commenti
Grafico	2	—	Scarsa	Limitata	—	Può essere più lento di un metodo numerico
Regola di Cramer	3	—	Soggetta a errore di arrotondamento	Limitata	—	Sforzo computazionale eccessivo per più di 3 equazioni
Manipolazione algebrica (eliminazione delle incognite)	2	—	Soggetta a errore di arrotondamento	Limitata		
Eliminazione di Gauss (con pivot parziale)	50	—	Soggetta a errore di arrotondamento	Generale	Moderata	
Gauss-Jordan (con pivot parziale)	50	—	Soggetta a errore di arrotondamento	Generale	Moderata	Permette il calcolo delle matrici inverse
Gauss-Seidel	1000	Può non convergere, se non è un sistema diagonale	Eccellente	Indicato solo per sistemi diagonali	Facile	
Gradiente Coniugato	**5000**	—	**Eccellente**	**Sistemi sparsi, simm.,def.pos.**	**Moderata**	

Soluzione di sistemi di equazioni non lineari con Newton-Raphson

Per trovare la soluzione di sistemi di equazioni non lineari del tipo

$$\underline{\mathbf{F}}(\underline{\mathbf{x}})=[f_1(\underline{\mathbf{x}}),...,f_N(\underline{\mathbf{x}})]=\underline{0}, \qquad \underline{\mathbf{x}}\in\mathbf{R}^N \rightarrow \mathbf{R}^N$$

è possibile utilizzare il metodo di Newton-Raphson (già introdotto nella sezione precedente con l'obiettivo di determinare le radici di una singola equazione non-lineare)
Posto $\mathbf{x}=\mathbf{x}^k+d\mathbf{x}$, si estende facilmente quanto detto nel caso monodimensionale.

L'idea di base dell'algoritmo consiste ancora una volta nell'espandere il sistema $\mathbf{F}(\mathbf{x}^{k+1})$ in serie di Taylor alla $k+1^{-ma}$ iterazione e imporre il suo annullamento.
Trascurando i termini del secondo ordine si ottiene:

$$\mathbf{F}(\mathbf{x}^k+d\mathbf{x}) \cong \mathbf{F}(\mathbf{x}^k) + \mathfrak{J}(\mathbf{x}^k)\, d\mathbf{x} = \mathbf{0} \qquad (1)$$
$$d\mathbf{x}= -\mathfrak{J}(\mathbf{x}^k)^{-1}\, \mathbf{F}(\mathbf{x}^k) \qquad (2)$$
$$\mathbf{x}^{k+1} = \mathbf{x}^k+d\mathbf{x} \qquad (3)$$

In questo caso è necessario saper valutare nel generico punto $\mathbf{x}$ oltre che le funzioni $f_i(\mathbf{x})$ anche lo Jacobiano $\mathfrak{J}(\mathbf{x})$ di $\mathbf{F}(\mathbf{x})$.
Il sistema lineare (2) può essere risolto utilizzando metodi utilizzando tra gli algoritmi presentati quello che risulta più vantaggioso in base alla struttura di $\mathfrak{J}(\mathbf{x})$.
Anche nel caso multidimensionale il metodo di Newton presenta un tasso di convergenza quadratico, vale a dire che *nelle vicinanze di una radice* il numero di cifre significative esatte approssimativamente raddoppia ad ogni step.

Anche questo metodo presenta, tuttavia, degli inconvenienti.

- In primo luogo la sua applicabilità dipende dal fatto che lo Jacobiano sia continuo e diverso da zero in prossimità delle radici.
- Inoltre, lontano dalle radici, dove i termini dello sviluppo di ordine superiore acquistano importanza rilevante, il metodo di Newton può dare grosse correzioni del tutto prive di significato.
- Infine, soprattutto nel caso in cui le funzioni $f_i(\mathbf{x})$ siano costruite per interpolazione non lineare di valori tabellati, possono insorgere patologie che determinano il completo fallimento dello schema solutivo.

Un modo per ridurre il rischio di insorgenza di situazioni patologiche e per migliorare ulteriormente il tasso di convergenza, è quello di assumere per la soluzione al passo k+1^{-mo}, anziché il valore fornita dalla (3), quello determinato dall'espressione:

$$\mathbf{x}^{\mathbf{k+1}} = \mathbf{x}^{\mathbf{k}} + a\, \mathrm{d}\mathbf{x} \qquad (4)$$

dove *a* è l'incognita scalare del problema monodimensionale non-lineare:

$$\mathbf{G}(\mathbf{x}^{\mathbf{k}}+a\, \mathrm{d}\mathbf{x}) = \mathbf{F}(\mathbf{x}^{\mathbf{k}}+a\, \mathrm{d}\mathbf{x})\, \mathrm{d}\mathbf{x}=\mathbf{0} \qquad (5)$$

che può essere risolto con il metodo *regula falsi*, oppure attraverso il *metodo delle secanti*.

Esempio: Utilizzando il metodo di Newton-Raphson, risolvere il sistema non lineare

$$\underline{\underline{\mathbf{F}}}(\underline{\mathbf{x}}) = \begin{cases} x^2 + y^3 + 5x - 63 = 0 \\ yx^3 + 4y - 204 = 0 \end{cases}$$

Lo Jacobiano del sistema è: $\quad \underline{\underline{\mathfrak{J}}}(\underline{\mathbf{x}}) = \begin{bmatrix} 2x+5 & 3y^2 \\ 3x^2 y & x^3+4 \end{bmatrix}$

Partiamo dal punto iniziale $\underline{\mathbf{x}}^0$=(3,4).
Alla prima iterazione si deve risolvere il sistema $-\underline{\underline{\mathfrak{J}}}(\underline{\mathbf{x}}^{\mathbf{0}})\underline{\mathrm{d}\mathbf{x}}^{\mathbf{1}}=\underline{\underline{\mathbf{F}}}(\underline{\mathbf{x}}^{\mathbf{0}})$, vale a dire:

$$\underline{\underline{\mathfrak{J}}}(x^0) = \begin{bmatrix} 2x+5 & 3y^2 \\ 3x^2 y & x^3+4 \end{bmatrix} = \begin{bmatrix} 11 & 48 \\ 108 & 31 \end{bmatrix}; \qquad \mathrm{F}(\mathrm{x}^0) = \begin{bmatrix} 25 \\ -80 \end{bmatrix}$$

Usando uno dei metodi precedenti si ottiene:

$$d\mathbf{x}^{\mathbf{1}} = \begin{bmatrix} 0.952921742721454 \\ -0.739211232707 \end{bmatrix}; \qquad \mathbf{x}^{\mathbf{1}} = \mathbf{x}^0 + d\mathbf{x}^{\mathbf{1}} = \begin{bmatrix} 3 \\ 4 \end{bmatrix} + \begin{bmatrix} 0.953 \\ -0.739 \end{bmatrix} = \begin{bmatrix} 3.953 \\ 3.261 \end{bmatrix}$$

Alla seconda iterazione risulta:

$$\underline{\underline{\mathfrak{J}}}(x^0) = \begin{bmatrix} 12.906 & 31.898 \\ 152.855 & 65.767 \end{bmatrix}; \qquad \mathrm{F}(\mathrm{x}^0) = \begin{bmatrix} 7.061 \\ 10.451 \end{bmatrix}$$

$$d\mathbf{x}^{\mathbf{1}} = \begin{bmatrix} 0.033 \\ -0.235 \end{bmatrix}; \qquad \mathbf{x}^{\mathbf{1}} = \mathbf{x}^0 + d\mathbf{x}^{\mathbf{1}} = \begin{bmatrix} 3.953 \\ 3.261 \end{bmatrix} + \begin{bmatrix} 0.033 \\ -0.235 \end{bmatrix} = \begin{bmatrix} 3.985 \\ 3.026 \end{bmatrix}$$

Una delle soluzioni è (4,3). Si vede come in poche iterazioni l'algoritmo converga verso la soluzione.

Circuiti semplici dinamici lineari

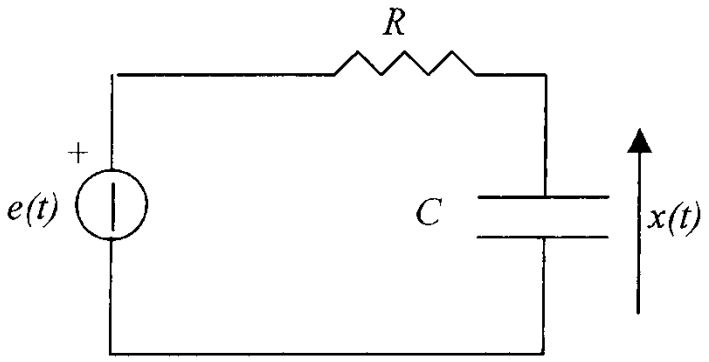


Figura 1.7: circuito RC.

Si consideri il circuito R-C rappresentato in figura, nel quale il generatore di tensione e(t) ha una forma d'onda assegnata nel tempo, e supponiamo di voler determinare la tensione ai capi del condensatore in ogni istante di tempo successivo a t_0 nel quale supponiamo noto tale valore $x(t_0)=x_0$.
L'equazione risolvente è un'equazione differenziale ordinaria del primo ordine:

$$\frac{dx}{dt}+\frac{x}{\tau}=\frac{e}{\tau}=b; \qquad x(t_0)=x_0; \qquad \tau=RC \quad (1)$$

dove $\tau=RC$ è la costante di tempo del circuito e x_0 è la condizione iniziale che rende unica la soluzione.
l'equazione differenziale (1) può in linea di principio essere risolta per via analitica, ma la cosa può diventare complicata quando l'espressione della forma d'onda in ingresso e(t) non è di tipo canonico.
Ci proponiamo di risolvere la (1) numericamente attraverso il metodo delle differenze finite.

METODO DELLE DIFFERENZE FINITE: Eulero esplicito

- La prima operazione da compiere è un campionamento della funzione incognita x(t) in istanti prefissati. In altre parole nel prosieguo non determineremo il valore di x(t) per ogni istante t, ma solo in determinati istanti t_k , k=0,1,2.
  Si supporrà nel seguito che gli istanti di campionamento siano equispaziati, ovvero distanziati l'uno dall'altro di un intervallo costante $h=t_k-t_{k-1}$, detto passo temporale. Si ha pertanto: $t_k=t_0+ k\,h$
  I valori corrispondenti della soluzione saranno definiti come $x_k=x(t_k)$
  In seguito verrà affrontato il problema dell'interpolazione dei dati, cioè della determinazione del valore della funzione x(t) in istanti differenti da t_k a partire dai valori x_k calcolati.

- La seconda operazione è la cosiddetta algebrizzazione della derivata temporale presente nella (1), ossia una sua approssimazione in termini di operazioni algebriche. Questo passaggio è necessario, in quanto sono queste le sole operazioni che un calcolatore è in grado di eseguire direttamente.
  Allo scopo, supposto *x(t)* sufficientemente regolare, può essere utilizzato uno sviluppo in serie di Taylor

$$x(t_k) = x(t_{k-1}) + \left.\frac{dx}{dt}\right|_{t_{k-1}} (t_k - t_{k-1}) + \frac{1}{2} \left.\frac{d^2x}{dt^2}\right|_{t_{k-1}} (t_k - t_{k-1})^2 + \ldots$$

Troncando al primo ordine, risulta $\left.\frac{dx}{dt}\right|_{t_{k-1}} = \frac{x(t_k) - x(t_{k-1})}{h} + O(h)$ (2)

dove *O(h)* indica una quantità che va a zero almeno come *h*.
In altre parole, trascurando *O(h)* (e conseguentemente accettando di commettere un errore che va a zero almeno come *h*), è possibile approssimare una derivata temporale con un rapporto incrementale.
Dalla natura stessa dell'approssimazione, deriva il nome del metodo, detto, appunto, *delle differenze finite*.
Sostituendo la (2) nella (1), si ottiene all'istante t_{k-1}

$$\frac{x(t_k) - x(t_{k-1})}{h} + \frac{x(t_{k-1})}{\tau} = b(t_{k-1}); \quad \Leftrightarrow x(t_k) = x(t_{k-1})\left(1 - \frac{h}{\tau}\right) + hb(t_{k-1}); \quad \forall k > 0 \qquad (3)$$

L'espressione (3) permettedi determinare **esplicitamente** il valore di x_k, una volto noto il valore di x_{k-1}. Da qui il nome del metodo.
A titolo di esempio si supponga di voler determinare la soluzione della (1) per *e(t)=0*.
In questo caso la soluzione analitica è: $x(t)=x_0\, e^{-t/\tau}$.
Nelle figure sono riportate le soluzioni numeriche per vari valori di h/τ.
Si può osservare come la soluzione numerica tenda a quella analitica quando questo rapporto tende a zero e, all'opposto, come le soluzioni tendano ad allontanarsi se il rapporto cresce, finché per $h/\tau>2$ la soluzione numerica risulti addirittura crescente in valore assoluto

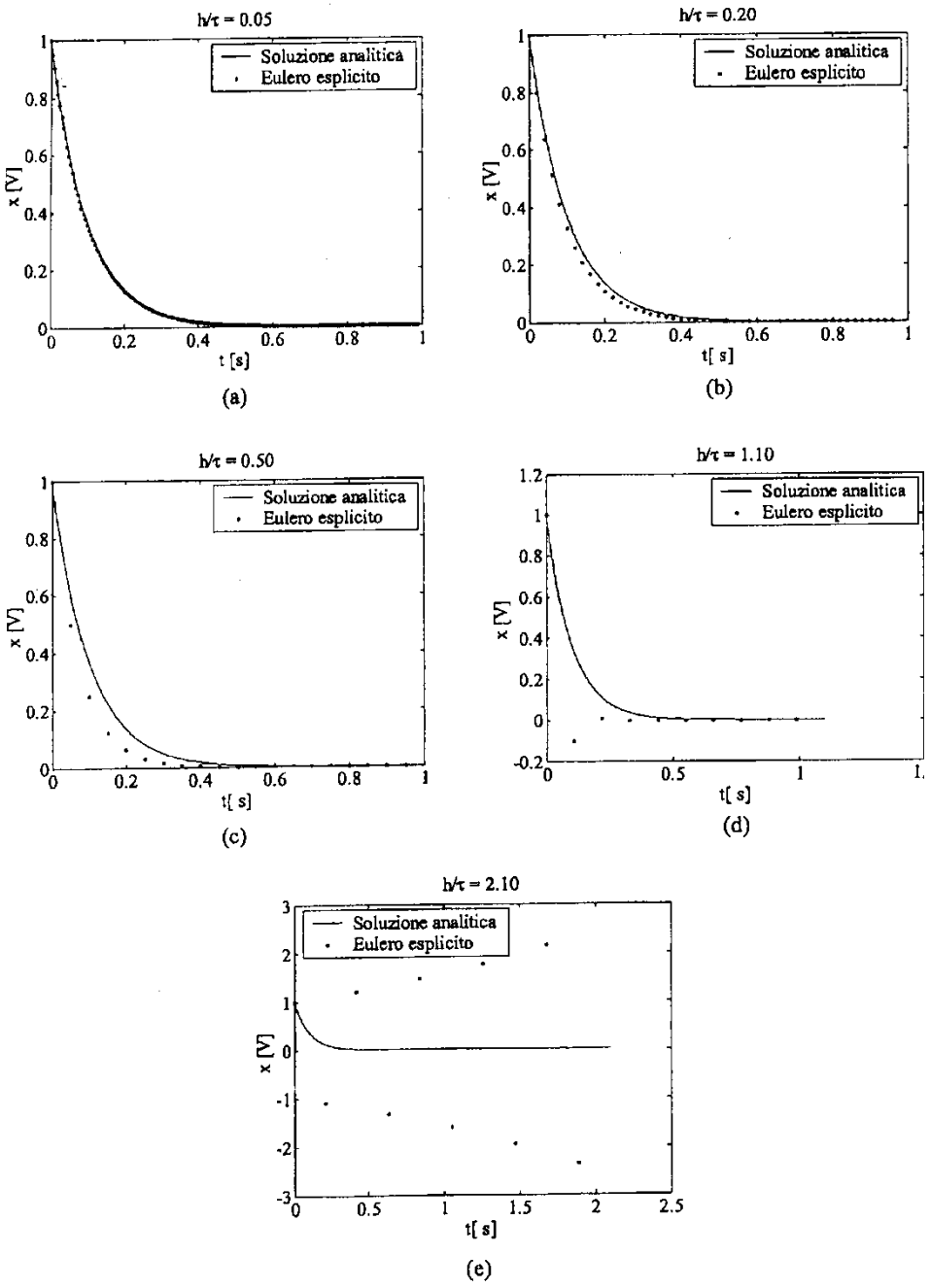


Figura 1.8: Soluzioni ottenute con il metodo di Eulero esplicito: ($h/\tau = 1/20$, (b) $h/\tau = 1/5$, (c) $h/\tau = 1/2$, (d) $h/\tau = 1.1$, ($h/\tau = 2.1$.

Convergenza,consistenza,stabilità

L'esempio precedente ha mostrato quanto la scelta del passo temporale *h* sia critica nella ricerca di una buona soluzione numerica.
Cerchiamo di quantificare questa osservazione.

Errore globale e convergenza

Sia $x^*(t)$ la soluzione della (1); si definisce *errore globale* al passo *k* la quantità:

$$e_k = x_k - x^*(t_k)$$

ossia la differenza, all'istante t_k, tra la soluzione approssimata e quella esatta.
Si dice che il metodo numerico è *convergente* se risulta:

$$\lim_{h\to 0} e_k = 0, \forall k > 0,$$

ossia se l'errore numerico globale tende a zero all'annullarsi del passo temporale (time-step). Si noti come la convergenza implichi che, attraverso un'opportuna scelta del time-step, si possa rendere l'errore piccolo a piacere.
Affinché un metodo numerico risulti di una qualche proprietà pratica questa proprietà deve essere obbligatoriamente soddisfatta.
Si può dimostrare che il metodo di Eulero esplicito deve essere convergente.

Risulta in particolare: $$e_k = e_0\left(1-\frac{h}{\tau}\right)^k + O(h^2) \qquad (4).$$

E cioè se $e_0=0$ (la condizione iniziale è nota esattamente), il metodo di Eulero esplicito risulta convergente.
Si noti che attraverso la (4) è possibile capire i risultati ottenuti nell'esempio precedente. Infatti per $h/\tau > 2$ si vede immediatamente che il termine in parentesi nella (4) risulta maggiore di 1 e dunque, nel tempo si ha una continua amplificazione dell'errore numerico.

Errore locale e consistenza

L'errore globale introdotto precedentemente rappresenta l'errore complessivo da cui la soluzione numerica risulta affetta.
Allo scopo di giudicare la bontà di uno schema alle differenze finite, può essere interessante studiare anche l'errore introdotto passo dopo passo.
A tale scopo si introduce la quantità $\tilde{x}_k$, definita come la soluzione che si ottiene al passo k-mo, supponendo che al passo *(k-1)-mo* la soluzione sia esatta.
Tale ipotesi viene chiamata localizzazione.

Con questa definizione la quantità $\tau_k = \tilde{x}_k - x^*(t_k)$ può senz'altro essere definito *errore locale di troncamento*, vale a dire l'errore introdotto al passo *k-mo* prescindendo dall'errore introdotto nei passi precedenti.

Se risulta: $\lim_{h\to 0} \frac{\tau_k}{h} = 0, \forall k > 0$, il metodo numerico viene detto *consistente* con l'equazione differenziale di partenza.
È facile convincersi che la proprietà di consistenza garantisce che, al limite per $h \to 0$, l'equazione algebrica approssimata restituisce l'equazione differenziale di partenza.

Infatti risulta:

$$0 = \lim_{h\to 0} \frac{\tau_k}{h} = \lim_{h\to 0} \frac{\tilde{x}_k - x^*(t_k)}{h} = \lim_{h\to 0} \frac{\tilde{x}_k - x^*(t_{k-1})}{h} - \lim_{h\to 0} \frac{x^*(t_k) - x^*(t_{k-1})}{h} \to$$

$$\left.\frac{dx^*}{dt}\right|_{t_{k-1}} = \lim_{h\to 0} \frac{\tilde{x}_k - x^*(t_{k-1})}{h}$$

Nel caso particolare di Eulero esplicito, sviluppando i calcoli risulta:

$$\lim_{h\to 0} \frac{\tau_k}{h} = \lim_{h\to 0} \left(\left.\frac{dx^*}{dt}\right|_{t_{k-1}} - \lim_{h\to 0} \frac{x^*(t_k) - x^*(t_{k-1})}{h} \right) = 0$$

e dunque, come anticipato la quantità $\frac{\tau_k}{h}$ risulta essere pari all'errore che si commette rimpiazzando la derivata temporale col rapporto incrementale, e dunque è perfettamente logico chiedere che l'errore vada a zero al tendere a zero del passo incrementale.
In particolare si definisce ordine dello schema numerico l'intero *p* tale che:

$$\frac{\tau_k}{h} = O(h^p)$$

e dunque lo schema di Eulero-esplicito è di ordine 1.

In parole povere quanto più è elevato l'ordine del metodo, tanto più rapidamente l'errore locale va a zero, quando $h \to 0$. Pertanto con un *h* sufficientemente piccolo i metodi di ordine superiore forniscono approssimazioni migliori di quelli di ordine inferiore.

Stabilità numerica

Si è visto nel caso precedente come, in assenza di forzamento, la soluzione numerica, per *h* sufficientemente grande, risulti in modulo indefinitamente crescente, anche quando la soluzione analitica è un'esponenziale decrescente (come è giusto data la stabilità della rete).

In altre parole la soluzione numerica ottenuta indicherebbe erroneamente che il circuito ha un comportamento di tipo instabile.

Da questa incongruenza nasce l'esigenza di chiarire la proprietà di stabilità numerica degli algoritmi, ossia la proprietà di fornire soluzioni limitate in corrispondenza di eccitazioni limitate per sistemi intrinsecamente stabili.

La letteratura abbonda di definizioni di stabilità. Il motivo di tanto proliferare è legato all'esigenza di ricavare dalle proprietà di stabilità e di consistenza, la proprietà di **convergenza**, ossia la proprietà fondamentale di un metodo alle differenze finite che purtroppo **non è di semplice verifica.**

In particolare citiamo il teorema di Lax che asserisce:

- Per un metodo alle differenze finite **consistente**, associato ad un'equazione differenziale, **la stabilità** è equivalente **alla convergenza.**

Definizione: zero-stabilità

Consideriamo un assegnato metodo numerico ed operiamo una perturbazione sul valore iniziale e, ad ogni passo *k*, sul forzamento δ_k' (risp. δ_k"). La soluzione corrispondente ottenuta sia x_k' (risp. x_k"); Il metodo si dice zero-stabile se

$$\forall \varepsilon > 0, \text{esistono } C \text{ ed } h_0 > 0 \text{ tali che per } h < h_0, \quad \left|\delta_k^{'} - \delta_k^{''}\right| < \varepsilon \Rightarrow \left|x_k^{'} - x_k^{''}\right| < C\varepsilon \ \forall k > 0 \quad (5)$$

In altre parole un metodo è zero-stabile se, scegliendo il passo h sufficientemente piccolo, la variazione della soluzione tende a zero, mandando a zero la perturbazione sul termine noto e sul forzamento ad ogni passo *k*.

Questa circostanza è fondamentale per l'applicabilità di un metodo numerico. Si ricordi infatti che tutti i numeri introdotti nel calcolatore sono soggetti inevitabilmente ad errori di arrotondamento a causa dell'utilizzo di una aritmetica finita. Se il metodo non è zero-stabile, gli errori vengono amplificati ad ogni passo.

Con questa osservazione possiamo dare una lettura intuitiva del teorema di Lax:

Se l'errore locale tende a zero con *h* (consistenza) ed esso non è amplificato indefinitamente nella soluzione (stabilità) allora l'errore globale tende a zero con *h* (convergenza).

Nel metodo di Eulero-esplicito il comportamento instabile indesiderato compare solo se *h* è troppo grande. Vale pertanto la pena di dare una definizione di stabilità per valori di *h* fissati.

Definizione: stabilità

Consideriamo un metodo numerico con passo *h* assegnato ed operiamo una perturbazione sul valore iniziale e, ad ogni passo *k*, sul forzamento δ_k' (risp. δ_k"). La soluzione corrispondente ottenuta sia x_k' (risp. x_k"); Il metodo si dice (numericamente) stabile per il passo *h* se

$$\forall \varepsilon > 0, \text{esiste} C > 0 \text{ tale che } \left|\delta_k^{'} - \delta_k^{''}\right| < \varepsilon \Rightarrow \left|x_k^{'} - x_k^{''}\right| < C\varepsilon \ \forall k > 0 \qquad (6)$$

In altre parole si richiede che la proprietà illustrata precedentemente valga per il passo h assegnato.

La (6) afferma che rendendo sufficientemente piccola la differenza delle perturbazioni, si rende piccola a piacere la differenza delle soluzioni.

Si osservi che se *k* rimanesse finito, si potrebbe sempre trovare una C tale da soddisfare la (6). Pertanto tale definizione è applicabile solo per $k \to \infty$, ossia se il dominio dell'equazione comprende tutto l'asse reale.

Si noti che, al contrario, nella definizione di zero-stabilità, il far tendere $h \to 0$ rende infinito il numero di passi anche se l'intervallo di soluzione è finito.

Applicando la definizione di stabilità (6) al metodo di Eulero-esplicito,

e definendo $\Delta_{\max} = \max(\left|\delta_n^{'} - \delta_n^{''}\right|); \quad r = \left(1 - \frac{h}{\tau}\right)$ si ottiene:

$$\left|x_k^{'} - x_k^{''}\right| \le \sum_{n=0}^{k} |r|^{k-n} \left|\delta_n^{'} - \delta_n^{''}\right| \le \Delta_{\max} \sum_{n=0}^{k} |r|^{k-n} = \Delta_{\max} \frac{1 - |r|^{k+1}}{1 - |r|} \qquad (7)$$

Pertanto condizione affinché il metodo sia stabile è che, come già si era trovato, risulti:

$$r = \left(1 - \frac{h}{\tau}\right) < 1 \Leftrightarrow 0 < h < 2\tau$$

Infatti, sotto questa condizione, e per $\Delta_{\max} < \varepsilon$ fissato ad arbitrio, si ottiene:

$$\left|x_k^{'} - x_k^{''}\right| \le \varepsilon \frac{1 - |r|^{k+1}}{1 - |r|} = C\varepsilon \ \ \forall k$$

che è proprio la condizione richiesta dalla (6).

Inoltre la (7) ci permette anche di dire che il metodo di Eulero è sicuramente stabile nel limite $h \to 0$, ossia che è zero stabile.

METODO DELLE DIFFERENZE FINITE: Eulero implicito

Una variante interessante del metodo di Eulero esplicito, può essere ottenuta considerando invece dello sviluppo in serie di Taylor precedente, il seguente:

$$x(t_{k-1}) = x(t_k) + \left.\frac{dx}{dt}\right|_{t_k} (t_{k-1} - t_k) + \frac{1}{2} \left.\frac{d^2x}{dt^2}\right|_{t_k} (t_{k-1} - t_k)^2 + \ldots$$

Arrestando ancora al primo ordine, risulta come prima:

$$\left.\frac{dx}{dt}\right|_{t_k} = \frac{x(t_k) - x(t_{k-1})}{h} + O(h) \qquad (8)$$

dove *O(h)* indica una quantità che va a zero almeno come *h*.

Tuttavia adesso, sostituendo la (8) nella (1), si ottiene all'istante t_k:

$$\frac{x(t_k) - x(t_{k-1})}{h} + \frac{x(t_k)}{\tau} = b(t_k); \;\; \Leftrightarrow \left(1 - \frac{h}{\tau}\right) x(t_k) = x(t_{k-1}) + hb(t_k); \forall k > 0$$

e quindi la formula: $x(t_k) = rx(t_{k-1}) + rhb(t_k); \forall k > 0; \text{con } r = \left(1 - \frac{h}{\tau}\right)^{-1}$ (9)

detto metodo di Eulero Implicito. Il nome deriva dal fatto che attraverso la (9) non si può esprimere direttamente $x(t_k)$ in funzione di $x(t_{k-1})$, ma, allo scopo, occorre invertire un'equazione, ovvero esplicitare il valore di *r*.

Se al posto di una singola equazione abbiamo a che fare con un sistema di equazioni, occorre risolvere ad ogni time-step un sistema di equazioni lineari.

A titolo di esempio si supponga di voler determinare con Eulero-implicito, la soluzione della (1) per *e(t)=0* (la soluzione analitica è: $x(t)=x_0\, e^{-t/\tau}$).
Si può notare come, in questo caso, anche per $h/\tau > 2$ la soluzione numerica si mantenga stabile.

Dall'esempio si intuiscono i risultati riportati di seguito:

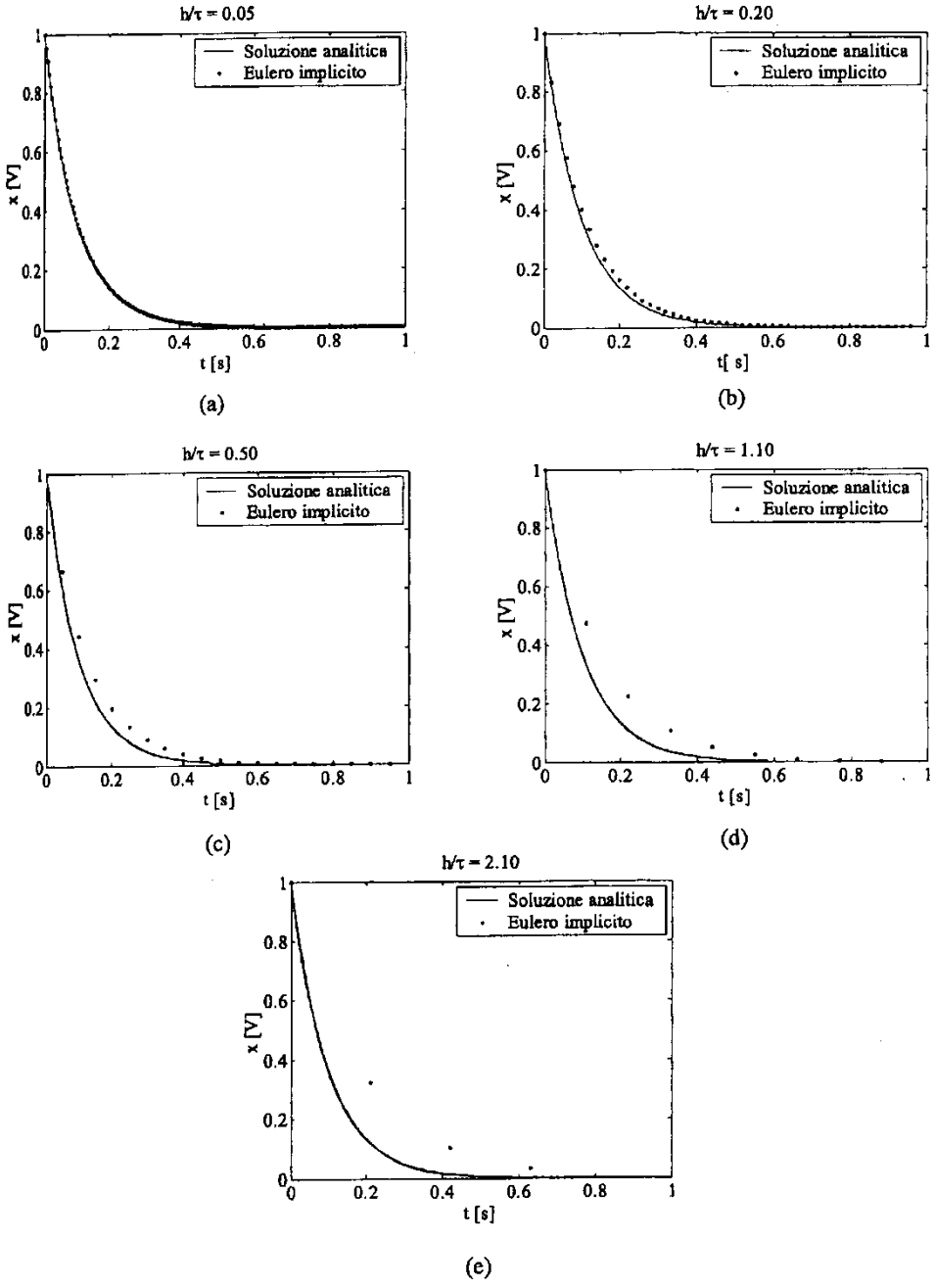


Figura 1.9: Soluzioni ottenute con il metodo di Eulero implicito: (a) $h/\tau = 1/20$, (b) $h/\tau = 1/5$, (c) $h/\tau = 1/2$, (d) $h/\tau = 1.1$, (a) $h/\tau = 2.1$.

Il metodo di Eulero-implicito è

- Consistente di ordine 1
- Incondizionatamente stabile, ossia stabile per ogni $h>0$, in quanto risulta sempre $r<1$

Sembrerebbe, da quanto sopra, che Eulero implicito sia sempre da preferirsi. Tuttavia non va dimenticato che la sua applicazione risulta decisamente più onerosa in termini computazionale rispetto a quella di Eulero Esplicito. Occorre dunque decidere caso per caso.

METODO DELLE DIFFERENZE FINITE: Metodo θ

Esistono molti altri possibili schemi per le differenze finite. A titolo di esempio, si riporta il seguente., dove si è posto $t_\theta = t_{k-1} + \theta h$
Dallo sviluppo in serie di Taylor, si ottiene:

$$x(t_k) = x(t_\theta) + \left.\frac{dx}{dt}\right|_{t_\theta} (1-\theta)h + \frac{1}{2}\left.\frac{d^2x}{dt^2}\right|_{t_\theta} ((1-\theta)h)^2 + O(h^3)... \quad (10)$$

$$x(t_{k-1}) = x(t_\theta) - \left.\frac{dx}{dt}\right|_{t_\theta} \theta h + \frac{1}{2}\left.\frac{d^2x}{dt^2}\right|_{t_\theta} (\theta h)^2 + O(h^3)... \quad (11)$$

Sottraendo membro a membro e dividendo per *h*, si ottiene:

$$\frac{x(t_k) - x(t_{k-1})}{h} = \left.\frac{dx}{dt}\right|_{t_\theta} + \frac{1}{2}\left.\frac{d^2x}{dt^2}\right|_{t_\theta} h((1-\theta)^2 - \theta^2) + O(h^2) \quad (12)$$

Sostituendo la (12) nella (1), si ottiene

$$\frac{x(t_k) - x(t_{k-1})}{h} + \frac{x(t_\theta)}{\tau} = b(t_\theta) + O(h)((1-\theta)^2 - \theta^2) + O(h^2); \quad (13)$$

Inoltre combinando linearmente la (9) e la (10), avendo moltiplicato la prima equazione θ, e la seconda per $(1-\theta)$, risulta

$$x(t_\theta) = \theta\, x(t_k) + (1-\theta)x(t_{k-1}) \qquad (14)$$

e, utilizzando per il coefficiente *b*, un'espressione analoga alla (14), la (13) diventa:

$$\frac{x(t_k) - x(t_{k-1})}{h} + \frac{\theta\, x(t_k) + (1-\theta)x(t_{k-1})}{\tau} = \theta\, b(t_k) + (1-\theta)b(t_{k-1}) + O(h)((1-\theta)^2 - \theta^2) + O(h^2); \quad (15)$$

La (15) porta al seguente schema alle differenze finite, detto metodo θ

$$x(t_k) = r x(t_{k-1}) + t h\left(\theta\, b(t_k) + (1-\theta)b(t_{k-1})\right);$$

$$\forall k > 0; \text{con } r = \left(1 + \theta\frac{h}{\tau}\right)^{-1}\left(1 - (1-\theta)\frac{h}{\tau}\right)^{-1}; t = \left(1 + \theta\frac{h}{\tau}\right)^{-1}$$

- Il metodo θ ha come casi particolari Eulero implicito ed esplicito, quando si rispettivamente $\theta=0$ e $\theta=1$.
- Se si sceglie $\theta=0.5$, il termine $O(h)$ si annulla e il metodo diventa del secondo ordine.
- Si può dimostrare che lo schema risulta incondizionatamente stabile per $\theta \geq 0.5$, mentre per $\theta<0.5$ è stabile per $h<2\tau/(1-2\theta)$

In figura viene riportato l'andamento di $|r|$ in funzione di h/τ per vari valori di θ
Si osservi come per $\theta \geq 0.5$ risulta *sempre* $|r|<1$, il che implica stabilità.

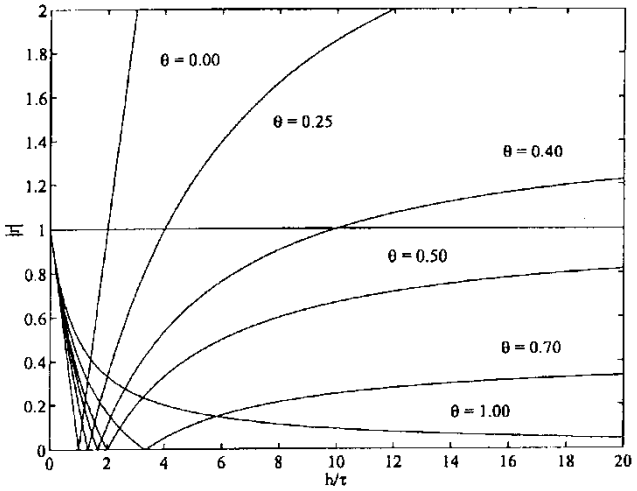


Figura 1.11: Andamento della quantità $|r|$ in funzione di h/τ per va valori di θ.

Con riferimento all'esempio precedente, si riporta l'andamento delle soluzioni per vari valori di θ, *avendo fissato* h/τ.
Risulta evidente come il valore $\theta=0.5$ fornisce la migliore approssimazione per la soluzione, come da attendersi visto che in questo caso il metodo risulta del secondo ordine.
Nella figura successiva, questa proprietà viene evidenziata riportando l'andamento dell'errore rispetto alla soluzione analitica in funzione di h per vari valori di θ.
Si può osservare che per $\theta=0$ *e per* $\theta=1$ l'errore è praticamente identico e va a zero come $O(h)$.
Per $\theta=0.5$ l'errore va a zero come $O(h^2)$ in quanto diminuendo h di un fattore 10, l'errore scala di un fattore 100.
Per $\theta=0.48$ l'errore è più basso rispetto ad entrambi i metodi di Eulero, ma va ancora a zero come $O(h)$ in quanto diminuendo h di un fattore 10, l'errore scala del medesimo fattore.

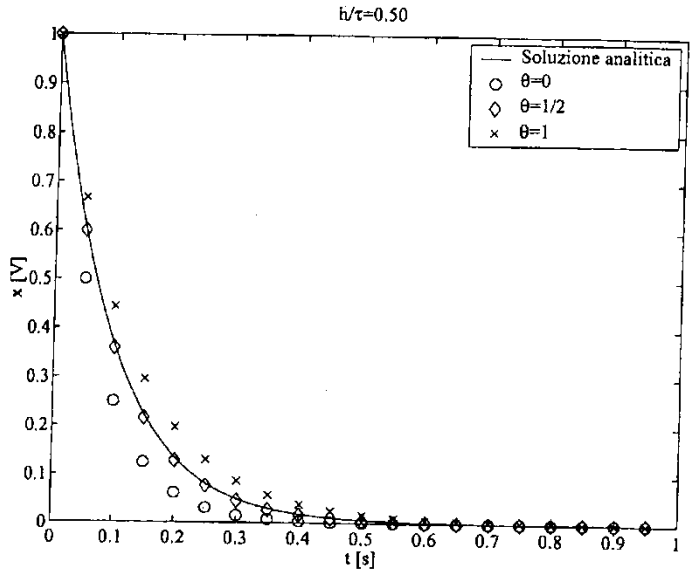


Figura 1.12: Soluzioni ottenute con il metodo θ, per $\theta = 0, 1/2, 1$, assumendo $h/\tau = 1/2$.

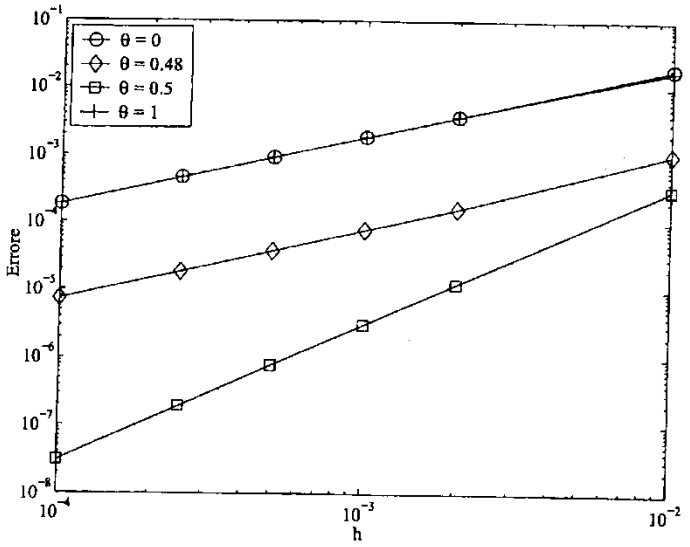


Figura 1.13: Andamento dell'errore in funzione di h, per vari valori di θ.