

ottenuto dal metodo delle potenze, senza normalizzazione. Si ponga $x_k = \mathbf{q}_{n_0}^{(k)}$ per un qualche n_0 , con $1 \leq n_0 \leq n$. Si dimostri che

$$\rho^2 = \frac{x_k x_{k+2} - x_{k+1}^2}{x_{k-1} x_{k+1} - x_k^2} + \mathcal{O}\left(\left\|\frac{\lambda_3}{\rho}\right\|^k\right),$$

$$\cos(\theta) = \frac{\rho x_{k-1} + \rho^{-1} x_{k+1}}{2x_k} + \mathcal{O}\left(\left\|\frac{\lambda_3}{\rho}\right\|^k\right).$$

[Suggerimento : si mostri innanzitutto che

$$x_k = C(\rho^k \cos(k\theta + \alpha)) + \mathcal{O}\left(\left\|\frac{\lambda_3}{\rho}\right\|^k\right),$$

dove α dipende dalle componenti del vettore iniziale lungo le direzioni individuate dagli autovettori associati a λ_1 e λ_2 .]

14. Si applichi il metodo delle potenze modificato dell'Esercizio 13 alla matrice

$$A = \begin{bmatrix} 1 & -\frac{1}{4} & \frac{1}{4} \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix},$$

e si confrontino i risultati ottenuti con quelli generati dal metodo delle potenze tradizionale.

6. Ricerca di radici di equazioni non lineari

In questo capitolo esponiamo le principali questioni relative all'approssimazione numerica delle radici (o zeri) di una funzione reale di una variabile. Il caso di sistemi di equazioni non lineari sarà trattato nel Capitolo 7.

Consideriamo il problema

$$\text{data } f : \mathcal{I} = (a, b) \subseteq \mathbb{R} \rightarrow \mathbb{R}, \text{ si cerca } x \in \mathcal{I} \text{ tale che } f(x) = 0. \quad (6.1)$$

L'approssimazione numerica di una radice α di f si basa in genere sull'uso di metodi iterativi; si costruisce cioè una successione di valori $x^{(k)}$ tali che

$$\lim_{k \rightarrow \infty} x^{(k)} = \alpha. \quad (6.2)$$

La convergenza di un metodo iterativo si può caratterizzare mediante la seguente definizione:

Definizione 6.1 Si dice che la successione $\{x^{(k)}\}$ generata da un metodo numerico converge ad α con ordine p se

$$\exists C > 0 : \frac{|x^{(k+1)} - \alpha|}{|x^{(k)} - \alpha|^p} \leq C, \forall k \geq k_0 \quad (6.3)$$

dove $k_0 \geq 0$ è un intero opportuno. In tal caso si dirà che il metodo è di ordine p . Si osservi che nel caso particolare in cui p sia uguale ad 1, per avere convergenza di $x^{(k)}$ ad α dovrà necessariamente essere $C < 1$ nella (6.3). In quest'ultimo caso, la costante C prende il nome di *fattore di convergenza*. ■

A differenza di quanto accade nel caso dei sistemi lineari, la convergenza dei metodi iterativi per l'approssimazione di radici di funzioni non lineari in generale dipende dalla scelta del dato iniziale $x^{(0)}$. Ciò consentirà di stabilire risultati di

convergenza di validità solo *locale*, ovvero per $x^{(0)}$ appartenente ad un intorno opportuno della radice α . Metodi per i quali si abbia invece convergenza ad α per ogni scelta di $x^{(0)}$ nell'insieme di definizione I , si diranno *globalmente convergenti* alla radice α .

6.1 Condizionamento di un'equazione non lineare

Consideriamo l'equazione non lineare $f(x) \equiv \varphi(x) - d = 0$ e supponiamo che f sia almeno di classe C^1 . Intendiamo studiare la sensibilità del calcolo delle radici di f rispetto a variazioni del dato d . Il problema è ben definito solo se la funzione φ è invertibile. In tal caso, potremo scrivere $x = \varphi^{-1}(d)$ da cui, con le notazioni del Capitolo 2, si ha che il risolvete G è φ^{-1} . D'altra parte $(\varphi^{-1})'(d) = 1/\varphi'(x)$ e quindi la formula (2.7) per il calcolo approssimato del numero di condizionamento (relativo e assoluto), fornisce

$$K(d) \simeq \frac{|d|}{|x||f'(x)|}, \quad K_{abs}(d) \simeq \frac{1}{|f'(x)|}. \tag{6.4}$$

Il problema è dunque mal condizionato quando $f'(x)$ è "piccola" e ben condizionato se $f'(x)$ è "grande". L'analisi che conduce alla (6.4) si può generalizzare al caso in cui x sia una radice di f di molteplicità $m > 1$ come segue. Sviluppando φ in serie di Taylor in un intorno della radice x fino all'ordine m , si ha

$$d + \delta d = \varphi(x + \delta x) = \varphi(x) + \sum_{k=1}^m \frac{\varphi^{(k)}(x)}{k!} (\delta x)^k + o((\delta x)^m). \tag{6.5}$$

Grazie al fatto che $\varphi^{(k)}(x) = 0$ per $k = 1, \dots, m - 1$, si ricava

$$\delta d = f^{(m)}(x)(\delta x)^m/m!$$

e, quindi, il numero di condizionamento assoluto diventa

$$K_{abs}(d) \simeq \left| \frac{m! \delta d}{f^{(m)}(x)} \right|^{1/m} \frac{1}{|\delta d|}. \tag{6.6}$$

Si noti che la (6.4) è un caso particolare della (6.6) quando $m = 1$. Da quest'ultima segue inoltre che, se anche δd fosse sufficientemente piccolo da rendere $|m! \delta d / f^{(m)}(x)| < 1$, $K_{abs}(d)$ potrebbe risultare comunque grande. In generale dunque il problema del calcolo delle radici di un'equazione non lineare è ben condizionato se x è radice semplice e $|f'(x)|$ è sensibilmente diverso da zero, mentre risulta mal condizionato in caso contrario.

Un problema strettamente collegato all'analisi precedente è il seguente. Si supponga che α sia la radice di f , ovvero $f(\alpha) = 0$, e che in corrispondenza di $\hat{\alpha} \neq \alpha$ si abbia $f(\hat{\alpha}) = \hat{r} \neq 0$. Ci si chiede come migliorare la differenza tra \hat{x} ed

x in funzione di \hat{r} . Applicando la (6.4) in cui $\delta \alpha = \hat{\alpha} - \alpha$ e $\delta d = \hat{r}$, e ricordando la definizione di K_{abs} (2.5), si trova

$$\frac{|\hat{\alpha} - \alpha|}{|\alpha|} \lesssim K_{abs}(0) \simeq \frac{1}{|f'(\alpha)|}.$$

(Qui e nel seguito adottiamo questa convenzione: se $a \leq b$ e $b \simeq c$, allora scriveremo $a \lesssim c$). Pertanto

$$\frac{|\hat{\alpha} - \alpha|}{|\alpha|} \lesssim \frac{|\hat{r}|}{|f'(\alpha)|}. \tag{6.7}$$

Nel caso in cui α abbia molteplicità $m > 1$, utilizzando la (6.6) anziché la (6.4) e procedendo come sopra si perviene a

$$\frac{|\hat{\alpha} - \alpha|}{|\alpha|} \lesssim \left(\frac{m!}{|f^{(m)}(\alpha)||\alpha|^m} \right)^{1/m} |\hat{r}|^{1/m}. \tag{6.8}$$

Riprenderemo questi risultati nell'analisi dei test d'arresto di metodi iterativi (si veda la Sezione 6.5).

Un caso notevole di problema non lineare è quello in cui f sia un polinomio p_n di grado n . Il teorema fondamentale dell'Algebra assicura che il problema del calcolo degli zeri di p_n ammette esattamente n soluzioni α_i , reali o complesse, ciascuna contata con la propria molteplicità. È naturale in questo caso indagare la sensibilità delle radici di p_n al variare dei suoi coefficienti. A questo scopo, sia $\hat{p}_n = p_n + q_n$, dove q_n è un polinomio di perturbazione di grado n , e siano $\hat{\alpha}_i$ le corrispondenti radici di \hat{p}_n . Una diretta applicazione della (6.8) conduce alla seguente stima:

$$\frac{|\hat{\alpha}_i - \alpha_i|}{|\alpha_i|} \lesssim \left(\frac{m!}{|p_n^{(m)}(\alpha_i)||\alpha_i|^m} \right)^{1/m} |q_n(\hat{\alpha}_i)|^{1/m}, \tag{6.9}$$

per $1 \leq m \leq n$ e $i = 1, \dots, n$, dove $q_n(\hat{\alpha}_i) = -p_n(\hat{\alpha}_i)$ è il "residuo" del polinomio p_n nella radice perturbata.

Osservazione 6.1 Un'analogia formale si può cogliere fra le stime a priori sin qui sviluppate per il problema non lineare $\varphi(x) = d$ e quelle della Sezione 3.1.2 per i sistemi lineari, pur di far corrispondere A a φ e b a d . Più precisamente, la (6.7) è l'analogia della (3.10) nel caso particolare in cui $\delta A = 0$, mentre la (6.9) (per $m = 1$) lo è nel caso in cui $\delta b = 0$.

Esempio 6.1 Sia $p_4(x) = (x - 1)^4$, con radici $\alpha_i = 1$ di molteplicità $m = 4$, e sia $\hat{p}_4 = (x - 1)^4 - \epsilon$, con $0 < \epsilon \ll 1$. Le radici del polinomio perturbato sono $\hat{\alpha}_i = 1 + \sqrt[4]{\epsilon}$, disposte ad intervalli di $\pi/2$ sul cerchio di raggio $\sqrt[4]{\epsilon}$ e centro in $z = (1, 0)^T$ nel piano

complesso. Si ha dunque *stabilità* (infatti $\lim_{\epsilon \rightarrow 0} \hat{\alpha}_i = 1$), ma anche *malcondizionamento*. Infatti, lo scarto relativo è

$$\frac{|\hat{\alpha}_i - \alpha_i|}{|\alpha_i|} = \sqrt[4]{\epsilon}, \quad i = 1, \dots, 4,$$

ovvero, per $\epsilon = 10^{-4}$ si ha una variazione relativa di 10^{-1} . Si noti che il termine a destra nella (6.9) coincide esattamente con $\sqrt[4]{\epsilon}$, e dunque in questo caso la (6.9) vale con il segno uguale!

Esempio 6.2 (di Wilkinson). Consideriamo il seguente polinomio

$$p_{10}(x) = \prod_{k=1}^{10} (x + k) = x^{10} + a_1 x^9 + \dots + a_{10}$$

con $a_1 = 55$ e $a_{10} = 10!$. Sia $\hat{p}_{10} = p_{10} + \epsilon x^9$, con $\epsilon = 2^{-23} \simeq 1.2 \cdot 10^{-7}$. Studiamo il condizionamento del calcolo delle radici di p_{10} . Riportiamo a tal fine in Tabella 6.1, per $i = 1, \dots, 10$ ed $m = 1$ nella (6.9), gli errori relativi (termine di sinistra nella (6.9) e indicati con E_{rel}^i) e le corrispondenti stime (termine a destra nella (6.9) e indicate con S^i).

Facciamo due osservazioni. La prima è che il problema è mal condizionato, come si osserva dall'errore relativo massimo che risulta, in corrispondenza della radice $\alpha_8 = -8$, di tre ordini di grandezza superiore al valore della perturbazione assoluta impressa. La seconda è che si ha un eccellente accordo tra la stima a priori (6.9) e l'errore relativo effettivamente commesso.

i	E_{rel}^i	S^i	i	E_{rel}^i	S^i
1	$3.03979 \cdot 10^{-13}$	$3.28508 \cdot 10^{-13}$	6	$6.95604 \cdot 10^{-5}$	$6.95663 \cdot 10^{-5}$
2	$7.56219 \cdot 10^{-10}$	$7.56884 \cdot 10^{-10}$	7	$1.58959 \cdot 10^{-4}$	$1.58850 \cdot 10^{-4}$
3	$7.75853 \cdot 10^{-8}$	$7.75924 \cdot 10^{-8}$	8	$1.98423 \cdot 10^{-4}$	$1.98767 \cdot 10^{-4}$
4	$1.80845 \cdot 10^{-6}$	$1.80847 \cdot 10^{-6}$	9	$1.27376 \cdot 10^{-4}$	$1.27125 \cdot 10^{-4}$
5	$1.61660 \cdot 10^{-5}$	$1.61664 \cdot 10^{-5}$	10	$3.28300 \cdot 10^{-5}$	$3.28605 \cdot 10^{-5}$

Tabella 6.1. Errore relativo e stima a priori fornita dalla (6.9) nell'analisi di sensitività per il polinomio di Wilkinson di grado 10

6.2 Un approccio geometrico per la ricerca delle radici

Introduciamo in questa sezione i metodi di bisezione, delle corde, delle secanti, di falsa posizione (o *Regula Falsi*) e di Newton. L'ordine di presentazione riflette la crescente complessità degli algoritmi: nel caso del metodo di bisezione, infatti, l'unica informazione utilizzata è il segno assunto dalla funzione f agli estremi di ogni (sotto)intervallo di bisezione, mentre nei restanti algoritmi si tiene conto anche dei valori assunti dalla funzione e/o dalla sua derivata prima.

6.2.1 Il metodo di bisezione

Il metodo di bisezione prende spunto dalla seguente proprietà

Proprietà 6.1 (teorema degli zeri per le funzioni continue) *Data una funzione $f: \bar{I} = [a, b] \rightarrow \mathbb{R}$, continua in \bar{I} e tale che $f(a)f(b) < 0$, allora $\exists \alpha \in \bar{I}$ tale che $f(\alpha) = 0$.*

A partire da $\mathcal{I}_0 = [a, b]$, il metodo di bisezione genera una successione di sottointervalli $\mathcal{I}_k = [a^{(k)}, b^{(k)}]$, $k \geq 0$, ciascuno dei quali è contenuto nel precedente ($\mathcal{I}_k \subset \mathcal{I}_{k-1}$, $k \geq 1$) e gode della proprietà che $f(a^{(k)}) \cdot f(b^{(k)}) < 0$. Precisamente, una volta individuato un intervallo $[a, b]$ tale che $f(a) \cdot f(b) < 0$, si pone $a^{(0)} = a$, $b^{(0)} = b$ e $x^{(0)} = (a^{(0)} + b^{(0)})/2$; quindi, per $k \geq 0$:

$$\begin{aligned} \text{si pone } a^{(k+1)} &= a^{(k)}, \quad b^{(k+1)} = x^{(k)} && \text{se } f(x^{(k)})f(a^{(k)}) < 0 \\ \text{si pone } a^{(k+1)} &= x^{(k)}, \quad b^{(k+1)} = b^{(k)} && \text{se } f(x^{(k)})f(b^{(k)}) < 0 \\ \text{si pone infine } x^{(k+1)} &= (a^{(k+1)} + b^{(k+1)})/2. \end{aligned} \quad (6.10)$$

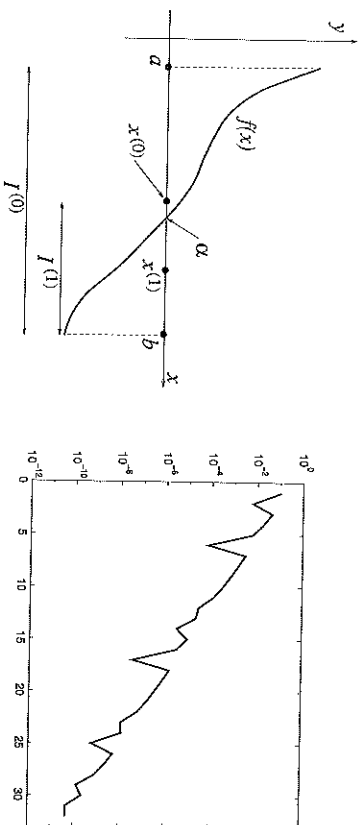


Fig. 6.1. Il metodo di bisezione: a sinistra, i primi due passi; a destra, la curva di convergenza relativa all'Esempio 6.3. In ascissa è riportato il numero di iterazioni, in ordinata l'errore assoluto in funzione di k

La strategia di bisezione si arresta al passo m -esimo in corrispondenza del quale si abbia $|x^{(m)} - \alpha| \leq |T_m| \leq \epsilon$, essendo ϵ una prefissata tolleranza e $|T_m|$ la misura di T_m . Per quanto riguarda la *velocità di convergenza* del metodo di bisezione, notiamo che $|T_0| = b - a$, mentre

$$|T_k| = |T_0|/2^k = (b - a)/2^k, \quad k \geq 0. \quad (6.11)$$

Introducendo l'errore assoluto $e^{(k)} \equiv x^{(k)} - \alpha$ al passo k , scende dalla (6.11) che $|e^{(k)}| \leq (b - a)/2^k$, $k \geq 0$, da cui risulta $\lim_{k \rightarrow \infty} |e^{(k)}| = 0$. Pertanto, il metodo di

bisezione è *globalmente convergente* e per ottenere $|x^{(m)} - \alpha| \leq \varepsilon$ bisogna prendere:

$$m \geq \log_2(b-a) - \log_2(\varepsilon) = \frac{\log((b-a)/\varepsilon)}{\log(2)} \approx \frac{\log((b-a)/\varepsilon)}{0.6931}. \quad (6.12)$$

In particolare, per guadagnare una cifra significativa sull'accuratezza dell'approssimazione della radice (ovvero per avere $|x^{(k)} - \alpha| = |x^{(j)} - \alpha|/10$), occorrono $(k-j) = \log_2(10) \approx 3.32$ bisezioni. Ciò caratterizza il metodo di bisezione come un algoritmo di sicura, ma lenta convergenza. Va sottolineato come il metodo di bisezione non garantisca, in generale, una riduzione *monotona* dell'errore assoluto tra due iterazioni consecutive, ovvero non si può assicurare *a priori* che valga, per ogni $k \geq 0$

$$|e^{(k+1)}| \leq M_k |e^{(k)}| \quad (6.13)$$

con $M_k < 1$. A tale proposito si consideri la situazione riportata in Figura 6.1(a), dove, evidentemente, risulta $|e^{(1)}| > |e^{(0)}|$. La mancata verifica della (6.13) impedisce di qualificare il metodo di bisezione come un metodo di ordine 1, nel senso della Definizione 6.1.

Esempio 6.3 Verifichiamo le proprietà di convergenza del metodo di bisezione per il calcolo della radice $\alpha = 0.9062$ dell'equazione di quinto grado $L_5(x) = 0$, dove

$$L_5(x) = \frac{x}{8}(63x^4 - 70x^2 + 15)$$

è il polinomio di Legendre di grado 5 le cui radici appartengono all'intervallo $(-1, 1)$ (si veda la Sezione 10.1.2). Il Programma 36 è stato eseguito prendendo $a = 0.6$, $b = 1$ ($L_5(a) \cdot L_5(b) < 0$), $nmax = 100$, $tol1 = 10^{-10}$ e ha raggiunto la convergenza impiegando 32 iterazioni, in accordo con la stima teorica (6.12) (risulta infatti $m \geq 31.8974$). La storia di convergenza, riportata in Figura 6.1(b), evidenzia una riduzione (media) dell'errore di un fattore 2, con un comportamento oscillatorio della successione $\{x^{(k)}\}$. •

Il lento abbattimento dell'errore, riscontrato anche sperimentalmente nell'esempio precedente, suggerisce di adottare il metodo di bisezione come una tecnica di avviamento alla radice. In pochi passi si ottiene infatti una stima ragionevole di α a partire dalla quale, utilizzando un metodo di ordine più elevato, è possibile convergere rapidamente alla soluzione cercata entro i limiti di accuratezza prestabiliti. Un esempio di tale procedura sarà illustrato nella Sezione 6.7.1.

L'algoritmo di bisezione è implementato in linguaggio MATLAB nel Programma 36. I parametri in ingresso, qui e nel seguito, hanno il seguente significato: a e b indicano gli estremi dell'intervallo di ricerca, fun la variabile contenente l'espressione della funzione f di cui si cerca uno zero, $tol1$ una tolleranza fissata e $nmax$ il numero massimo ammissibile di passi del processo iterativo. Nei vettori di uscita $xvect$, $xdiff$ e fx sono memorizzate rispettivamente le successioni $\{x^{(k)}\}$, $\{|x^{(k+1)} - x^{(k)}|\}$ e $\{f(x^{(k)})\}$, per $k \geq 0$, mentre nit indica il numero di iterazioni

impiegate dall'algoritmo per soddisfare il criterio di convergenza. Nel caso del metodo di bisezione, il codice si arresta non appena la semiampiezza dell'intervallo di ricerca risulta minore di $tol1$.

Programma 36 - bisect : Metodo di bisezione

```
function [xvect,xdiff,fx,nit]=bisect(a,b,toll,nmax,fun)
err=tol1+1; nit=0; xvect=[]; fx=[]; xdiff=[];
while (nit < nmax & err > toll)
    nit=nit+1; c=(a+b)/2; xc=c; fc=eval(fun); xvect=[xvect;x];
    fx=[fx;fc]; xa=a; if (fc*eval(fun) > 0), a=c; else, b=c; end;
    err=abs(b-a); xdiff=[xdiff;err];
end;
```

6.2.2 I metodi delle corde, secanti, Regola Falsi e Newton

Per costruire algoritmi con proprietà di convergenza migliori dei precedenti conviene fare uso anche dei valori di f' (nel caso in cui f sia differenziabile) o di una sua opportuna approssimazione. A tale scopo, sviluppando f in serie di Taylor in un intorno di α e arrestando lo sviluppo al prim'ordine, si ottiene una versione *linearizzata* del problema (6.1):

$$f(\alpha) = 0 = f(x) + (\alpha - x)f'(\xi), \quad (6.14)$$

per un opportuno ξ compreso tra α ed x . La (6.14) suggerisce il seguente metodo iterativo: per ogni $k \geq 0$, dato $x^{(k)}$, si determina $x^{(k+1)}$ risolvendo l'equazione $f(x^{(k)}) + (x^{(k+1)} - x^{(k)})g_k = 0$, dove g_k è una costante che rappresenta una opportuna approssimazione della derivata prima di f . Il metodo descritto equivale a trovare il punto di intersezione tra l'asse x e la retta di pendenza g_k passante per il punto $(x^{(k)}, f(x^{(k)}))$ e può essere quindi posto in modo più conveniente nella forma

$$x^{(k+1)} = x^{(k)} - q_k^{-1} f(x^{(k)}), \quad \forall k \geq 0. \quad (6.15)$$

Consideriamo nel seguito quattro particolari scelte di g_k .

Il metodo delle corde. Si pone:

$$g_k = q = \frac{f(b) - f(a)}{b - a}, \quad \forall k \geq 0 \quad (6.16)$$

ottenendo, una volta assegnato il valore iniziale $x^{(0)}$, la relazione ricorsiva:

$$x^{(k+1)} = x^{(k)} - \frac{b-a}{f(b)-f(a)} f(x^{(k)}), \quad k \geq 0 \quad (6.17)$$

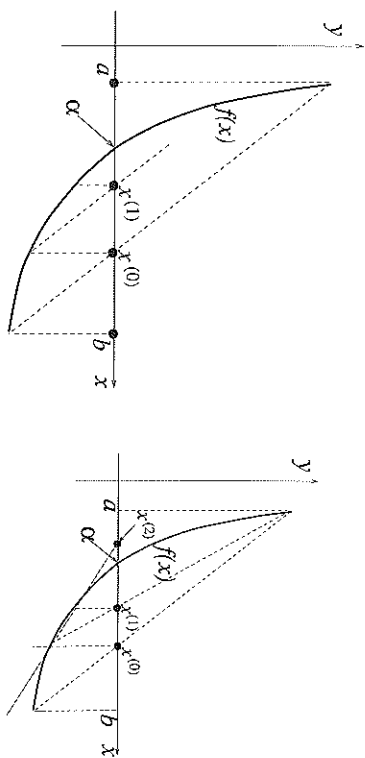


Fig. 6.2. I primi due passi dei metodi delle corde (a sinistra) e delle secanti (a destra)

Si vedrà nella Sezione 6.3.1 che la successione $\{x^{(k)}\}$ generata dalla (6.17) converge alla radice α con ordine di convergenza $p = 1$.

Il metodo delle secanti. Si pone:

$$q_k = \frac{f(x^{(k)}) - f(x^{(k-1)})}{x^{(k)} - x^{(k-1)}}, \quad \forall k \geq 0 \quad (6.18)$$

ottenendo, una volta assegnati i due valori iniziali $x^{(-1)}$ e $x^{(0)}$, il metodo

$$x^{(k+1)} = x^{(k)} - \frac{x^{(k)} - x^{(k-1)}}{f(x^{(k)}) - f(x^{(k-1)})} f(x^{(k)}), \quad k \geq 0 \quad (6.19)$$

Rispetto al metodo delle corde, il processo iterativo (6.19) richiede l'assegnazione del punto iniziale $x^{(-1)}$ ed il calcolo del corrispondente valore $f(x^{(-1)})$, nonché, per ogni k , il calcolo del rapporto incrementale (6.18). Tale costo addizionale è confortato da un incremento della velocità di convergenza, come enunciato nella seguente proprietà, che può essere considerata come un primo esempio di teorema di *convergenza locale* (si veda per la dimostrazione [97], pag. 99-101)

Proprietà 6.2 Sia $f \in C^2(\mathcal{J})$, essendo \mathcal{J} un intorno opportuno della radice α e si assuma $f''(\alpha) \neq 0$. Allora, se i dati iniziali $x^{(-1)}$ e $x^{(0)}$ sono scelti sufficientemente vicini ad α (in \mathcal{J}), la successione (6.19) converge ad α con ordine $p = (1 + \sqrt{5})/2 \approx 1.63$.

Il metodo Regula Falsi (o della falsa posizione). È una variante del metodo delle secanti in cui, anziché scegliere la secante attraverso i valori $(x^{(k)}, f(x^{(k)}))$ e

$(x^{(k-1)}, f(x^{(k-1)}))$, si sceglie quella attraverso $(x^{(k)}, f(x^{(k)}))$ e $(x^{(k')}, f(x^{(k')}))$, essendo k' il massimo indice minore di k per cui si abbia $f(x^{(k')}) \cdot f(x^{(k)}) < 0$. Precisamente, individuati due valori $x^{(-1)}$ e $x^{(0)}$ tali per cui $f(x^{(-1)}) \cdot f(x^{(0)}) < 0$, si pone

$$x^{(k+1)} = x^{(k)} - \frac{x^{(k)} - x^{(k')}}{f(x^{(k)}) - f(x^{(k')})} f(x^{(k)}), \quad k \geq 0 \quad (6.20)$$

Fissata una tolleranza assoluta ϵ , la successione (6.20) viene arrestata all'iterazione m -esima per cui risulta $|f(x^{(m)})| < \epsilon$. Si osserva che la successione degli indici k' è non decrescente; pertanto, al fine di determinare al passo k il nuovo valore di k' , non è necessario scandire tutta la successione a ritroso, ma è sufficiente arrestarsi al valore di k' determinato al passo precedente. Sono riportati a sinistra nella Figura 6.3 i primi due passi della (6.20) nella situazione particolare in cui $x^{(k)}$ coincide con $x^{(-1)}$ per ogni $k \geq 0$.

Il metodo *Regula Falsi*, pur avendo una complessità computazionale simile a quella del metodo delle secanti, ha ordine di convergenza lineare (si veda ad esempio [31], pag. 339-340). Peraltro, le iterate generate dalla (6.20) sono tutte contenute nell'intervallo di partenza $[x^{(-1)}, x^{(0)}]$, contrariamente a quanto può accadere per il metodo delle secanti. A tale proposito si veda l'esempio illustrato a destra in Figura 6.3, dove sono riportate le prime due iterazioni dei metodi delle secanti e della *Regula Falsi*, a partire da medesimi dati iniziali $x^{(-1)}$ e $x^{(0)}$. Si nota come l'iterata $x^{(1)}$ calcolata dal metodo delle secanti coincida con quella calcolata dal metodo *Regula Falsi*, mentre il valore $x^{(2)}$ calcolato dal primo metodo (e indicato nella figura con $x^{Sec(2)}$) cade fuori dall'intervallo iniziale $[x^{(-1)}, x^{(0)}]$.

Anche il metodo *Regula Falsi*, come il metodo di bisezione, può pertanto considerarsi un metodo *globalmente convergente*.

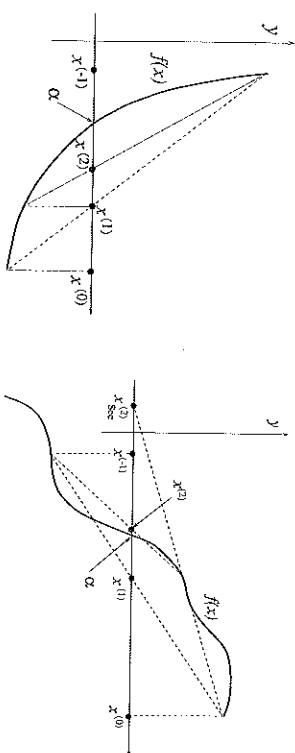


Fig. 6.3. I primi due passi del metodo *Regula Falsi* per due diverse funzioni

Il metodo di Newton (o di Newton-Raphson o delle tangenti).

Supponendo $f \in C^1(\mathcal{I})$ e assumendo $f'(\alpha) \neq 0$ (ovvero α radice semplice), se si

pone

$$q_k = f'(x^{(k)}), \quad \forall k \geq 0 \quad (6.21)$$

una volta assegnato il valore iniziale $x^{(0)}$, si ottiene il cosiddetto *metodo di Newton*:

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})}, \quad k \geq 0 \quad (6.22)$$

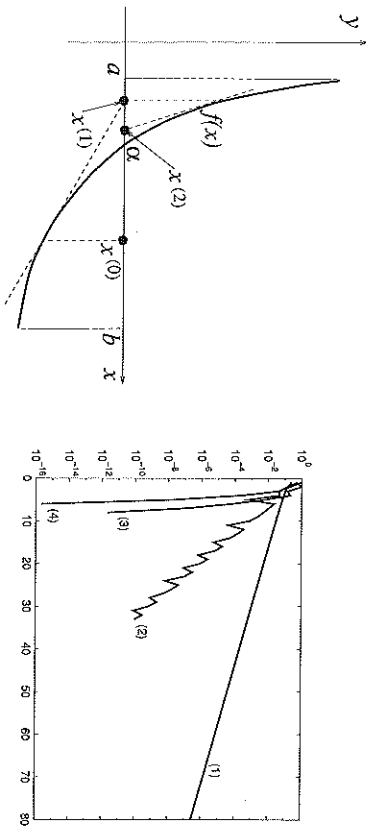


Fig. 6.4. A sinistra: i primi due passi del metodo di Newton; a destra: curve di convergenza nell'Esempio 6.4 per i metodi (1) delle corde, (2) di bisezione, (3) delle secanti, (4) di Newton. In ascissa è riportato il numero di iterazioni, in ordinata l'errore assoluto al variare di k

Alla k -esima iterazione, il metodo di Newton richiede *due* valutazioni funzionali ($f(x^{(k)})$ e $f'(x^{(k)})$). L'aumento del costo computazionale rispetto ai metodi sopra introdotti è però compensato da un incremento nell'ordine di convergenza, essendo questo un metodo di ordine 2 (si veda la Sezione 6.3.1).

Esempio 6.4 Confrontiamo il comportamento dei metodi sopra introdotti nel caso del calcolo della radice $\alpha \approx 0.5149$ della funzione $f(x) = \cos^2(x) - x^2$ nell'intervallo $0 \leq x \leq 1.5$. La tolleranza e sull'errore assoluto è stata presa uguale a 10^{-10} e le storie di convergenza dei vari metodi sono illustrate a destra in Figura 6.4.

L'analisi dei risultati mette in particolare evidenza la lenta convergenza del metodo delle corde; per ottenere un sensibile miglioramento delle prestazioni del metodo, è necessario scegliere $x^{(0)}$ molto vicino alla radice α . Per ottenere una soluzione accurata alla quindicesima cifra significativa sono necessari 53 passi del metodo di bisezione, come risulta dalla (6.12) prendendo $\epsilon = 2.2 \cdot 10^{-16}$ (pari all'unità di *roundoff*), a fronte delle 6 iterazioni impiegate dal metodo di Newton. Anche il metodo *Regula Falsi* risulta superiore al metodo di bisezione, impiegando 7 iterazioni per raggiungere la convergenza, esattamente come nel caso del metodo delle secanti. La curva dell'errore per il metodo *Regula Falsi*

è assai simile a quella del metodo delle secanti e per tale motivo non è stata riportata nella Figura 6.4.

È infine interessante confrontare le prestazioni dei metodi di Newton e delle secanti (entrambi aventi ordine $p > 1$) in relazione allo sforzo computazionale da essi richiesto. Si può infatti dimostrare che è più conveniente utilizzare il metodo delle secanti ogni volta che il numero di operazioni macchina per valutare la f' risulta circa doppio delle operazioni necessarie per valutare f (si veda [6], pag. 71-73). In effetti, nell'esempio qui considerato il metodo di Newton converge alla radice α in 6 iterazioni anziché in 7, tuttavia il metodo delle secanti impiega 94 *flops* rispetto alle 177 *flops* del metodo di Newton.

I metodi delle corde, delle secanti, *Regula Falsi* e di Newton sono implementati in linguaggio MATLAB nei programmi 37, 38, 39 e 40. Qui e nel seguito $x0$ e $xm1$ indicano i dati iniziali, $x^{(0)}$ e $x^{(-1)}$. Nel caso del metodo *Regula Falsi*, $tol1$ è la tolleranza per il test d'arresto $|f(x^{(k)})| < tol1$, mentre per il metodo di Newton la variabile $dnum$ contiene l'espressione di f' .

Programma 37 - chord : Metodo delle corde

```
function [xvect,xdif,fx,nit]=chord(a,b,x0,nmax,tol1,fun)
x=a; fa=eval(fun); x=b; fb=eval(fun); r=(fb-fa)/(b-a);
err=tol1+1; nit=0; xvect=x0; x=x0; fx=eval(fun); xdif=[];
while (nit<nmax & err>tol1),
    nit=nit+1; x=xvect(nit); xn=x-fx(nit)/r; err=abs(xn-x);
    xdif=[xdif; err]; x=xn; xvect=[xvect;x]; fx=[fx;eval(fun)];
end;
```

Programma 38 - secant : Metodo delle secanti

```
function [xvect,xdif,fx,nit]=secant(xm1,x0,nmax,tol1,fun)
x=xm1; fxm1=eval(fun); xvect=[x]; fx=[fxm1]; x=x0; fx0=eval(fun);
xvect=[xvect;x]; fx=[fx;fx0]; err=tol1+1; nit=0; xdif=[];
while (nit<nmax & err>tol1)
    nit=nit+1; x=x0-fx0*(x0-xm1)/(fx0-fxm1); xvect=[xvect;x];
    fx=eval(fun); fx=[fx;fxnew]; err=abs(x0-x); xdif=[xdif;err];
    xm1=x0; fxm1=fx0; x0=x; fx0=fxnew;
end;
```

Programma 39 - regfalsi : Metodo *Regula Falsi*

```
function [xvect,xdif,fx,nit]=regfalsi(xm1,x0,tol1,imax,fun)
nit=0; x=xm1; f=eval(fun); fx=[f]; x=x0; f=eval(fun); fx=[fx, f];
xvect=[xm1,x0]; xdif=[]; f=tol1+1; kprime=1;
while (nit < imax) & (abs(f) > tol1)
nit=nit+1; dim=length(xvect); x=xvect(dim); fxk=eval(fun); xk=x; i=dim;
while (i >=kprime), i=i-1; x=xvect(i); fxkpr=eval(fun);
if ((fxkpr*fxk)<0), xkpr=x; kprime=i; break; end;
end;
x=xk-fxk*(xk-xkpr)/(fxk-fxkpr); xvect=[xvect, x]; f=eval(fun);
fx=[fx, f]; err=abs(x-xkpr); xdif=[xdif, err];
end;
```

Programma 40 - newton : Metodo di Newton

```
function [xvect,xdif,fx,nit]=newton(x0,imax,tol1,fun,dfun)
err=tol1+1; nit=0; xvect=x0; x=x0; fx=eval(fun); xdif=[];
while (nit<imax & err>tol1)
nit=nit+1; x=xvect(nit); dfx=eval(dfun);
if (dfx == 0), err=tol1*1.e-10;
disp(' Arresto per azzerramento di dfun');
else,
xn=x-fx(nit)/dfx; err=abs(xn-x); xdif=[xdif, err];
x=xn; xvect=[xvect;x]; fx=[fx;eval(fun)];
end;
end;
```

6.2.3 Il metodo di Dekker-Brent

L'algoritmo di Dekker-Brent utilizza in modo combinato i metodi di bisezione e delle secanti, riunendo i pregi di entrambi a vantaggio di una notevole efficienza computazionale.

Individuato un intervallo $[a^{(0)}, b^{(0)}]$ tale che $f(a^{(0)})f(b^{(0)}) < 0$ e contenente almeno una radice α della funzione $y = f(x)$, l'algoritmo costruisce per $k \geq 0$ una successione di valori $a^{(k)}$, $b^{(k)}$ e $c^{(k)}$ tali che α si trovi sempre tra $b^{(k)}$ e $c^{(k)}$ e, a convergenza, la semiampiezza $|c^{(k)} - b^{(k)}|/2$ risulti minore di una tolleranza prefissata. Il punto $b^{(k)}$ è scelto in modo tale che risulti $|f(b^{(k)})| < |f(c^{(k)})|$. Inoltre, per ogni $k \geq 0$ si ha $a^{(k+1)} = b^{(k)}$. Se la funzione f è sufficientemente regolare, l'ordine di convergenza dell'algoritmo è più che lineare, come accade nel caso del metodo delle secanti (si veda [48], [22] Capitolo 4 e [6], pag. 91-93). Indicata con ε una tolleranza opportuna e posto $c^{(0)} = a^{(0)}$, per ogni $k \geq 0$ il metodo di Dekker-Brent procede come segue.

1. Si pone $\delta^{(k)} = 2\varepsilon \max\{|b^{(k)}|, 1\}$.

2. Se $f(b^{(k)})f(c^{(k)}) > 0$, ovvero $c^{(k)}$ si trova dalla stessa parte della radice α rispetto a $b^{(k)}$, si opera lo scambio $c^{(k)} = a^{(k)}$.
3. Se $|f(c^{(k)})| < |f(b^{(k)})|$, si operano gli scambi $a^{(k)} = b^{(k)}$, $b^{(k)} = c^{(k)}$ e $c^{(k)} = a^{(k)}$.
4. Se $c^{(k)} = a^{(k)}$, si calcola la radice z del polinomio di primo grado interpolante i punti $(c^{(k)}, f(c^{(k)}))$ e $(b^{(k)}, f(b^{(k)}))$. Questo equivale a eseguire un passo del *metodo delle secanti* sull'intervallo avente per estremi i punti $b^{(k)}$ e $c^{(k)}$.
5. Se invece $c^{(k)} \neq a^{(k)}$, ovvero i punti $a^{(k)}$, $b^{(k)}$ e $c^{(k)}$ sono distinti, si costruisce il polinomio di secondo grado rispetto a y , interpolante i punti $(a^{(k)}, f(a^{(k)}))$, $(b^{(k)}, f(b^{(k)}))$ e $(c^{(k)}, f(c^{(k)}))$ (*interpolazione quadratica inversa*) e se ne calcola il valore z in $y = 0$.

6. Si pone

$$(a) \quad b = z \text{ se } z \text{ si trova fra } b^{(k)} \text{ e } b^{(k)} + m^{(k)}, \text{ essendo } m^{(k)} = (c^{(k)} - b^{(k)})/2.$$

$$(b) \quad \text{In caso contrario, si pone } b = (b^{(k)} + c^{(k)})/2.$$

7. Si pone $a^{(k+1)} = b^{(k)}$.

8. Si pone infine $b^{(k+1)} = b^{(k)} + \delta^{(k)} \text{sign}(m^{(k)})$ se risulta $|b^{(k)} - b| \leq \delta^{(k)}$; diversamente, si pone $b^{(k+1)} = b$. In quest'ultimo caso, se il valore di b proviene dal passo 6(b) dell'algoritmo, l'assegnazione $b^{(k+1)} = b$ equivale ad applicare il *metodo di bisezione* sull'intervallo avente per estremi i punti $b^{(k)}$ e $c^{(k)}$.

Il processo di calcolo si arresta quando è verificata una delle seguenti condizioni

$$f(b^{(k)}) = 0 \quad \text{oppure} \quad |m^{(k)}| \leq \delta^{(k)}. \quad (6.23)$$

Esempio 6.5 Per verificare il comportamento dell'algoritmo di Dekker-Brent, è stata utilizzata la function `fzero` di MATLAB che implementa il metodo considerato. La function determina i valori $a^{(0)}$ e $b^{(0)}$ in modo automatico a partire da un assegnato dato iniziale ξ fornito dall'utente.

Consideriamo il calcolo della radice α della funzione f studiata nell'Esempio 6.4, assumendo ε pari all'unità di *roundoff*. Prendendo $\xi = 1.5$, l'algoritmo individua i valori $a^{(0)} = 0.3$ e $b^{(0)} = 2.1$; la convergenza è raggiunta in 5 iterazioni e le successioni dei valori $a^{(k)}$, $b^{(k)}$, $c^{(k)}$ e $f(x^{(k)})$ sono riportate nella Tabella 6.2. Osserviamo che i valori tabulati si riferiscono allo stato appena prima del punto 4. nell'algoritmo sopra illustrato, e quindi, in particolare, a posteriori di eventuali scambi fra $a^{(k)}$ e $b^{(k)}$.

6.3 Il metodo delle iterazioni di punto fisso

Consideriamo in questa sezione un approccio del tutto generale per l'approssimazione delle radici di una funzione non lineare. Il metodo è basato sull'osservazione

k	$a^{(k)}$	$b^{(k)}$	$c^{(k)}$	$f(x^{(k)})$
0	2.1	0.3	2.1	0.5912
1	0.3	0.5235	0.3	$-2.39 \cdot 10^{-2}$
2	0.5235	0.5148	0.5235	$3.11 \cdot 10^{-4}$
3	0.5148	0.5149	0.5148	$-8.8 \cdot 10^{-7}$
4	0.5149	0.5149	0.5148	$-3.07 \cdot 10^{-11}$

Tabella 6.2. Soluzione dell'equazione $\cos^2(2x) - x^2 = 0$ con l'algoritmo di Dekker-Brent

che, data $f : [a, b] \rightarrow \mathbb{R}$, è sempre possibile trasformare il problema $f(x) = 0$ in un problema equivalente $x - \phi(x) = 0$, dove la funzione ausiliaria $\phi : [a, b] \rightarrow \mathbb{R}$ deve essere scelta in modo tale che $\phi(\alpha) = \alpha$ ogni volta che $f(\alpha) = 0$. Il calcolo degli zeri di una funzione è dunque ricondotto a determinare i punti fissi (o punti uniti) dell'applicazione ϕ . Si considera poi il seguente algoritmo iterativo:

$$x^{(k+1)} = \phi(x^{(k)}), \quad k \geq 0 \quad (6.24)$$

Chiameremo la (6.24) *iterazione di punto fisso* e ϕ la *funzione di iterazione* ad essa associata. La (6.24) è anche detta *iterazione di Picard* o *iterazione funzionale* per risolvere il problema $f(x) = 0$ e la proprietà $\phi(\alpha) = \alpha$ si chiama *consistenza* dell'iterazione (6.24). La costruzione di ϕ non è unica. Ad esempio, fissata una costante arbitraria $\lambda \neq 0$, è sempre possibile assumere $\phi(x) = x + \lambda f(x)$, ma anche $\phi(x) = x + F(f(x))$, per ogni funzione continua F tale che $F(0) = 0$. Per caratterizzare le proprietà di ϕ affinché il metodo di punto fisso (6.24) risulti convergente alla radice α del problema (6.1) si ha il seguente risultato:

Teorema 6.1 (di convergenza delle iterazioni di punto fisso) Si consideri la successione $x^{(k+1)} = \phi(x^{(k)})$, per $k \geq 0$ con $x^{(0)}$ assegnato. Si supponga che:

1. $\exists \mathcal{J} \subseteq [a, b] : \forall x \in \mathcal{J}, \phi(x) \in \mathcal{J}$;
2. $\phi \in C^0(\mathcal{J})$;
3. $\exists K < 1 : \forall x \in \mathcal{J}, \phi'(x) \text{ esiste e } |\phi'(x)| \leq K$.

Allora ϕ ha un unico punto fisso α in \mathcal{J} , e la successione $\{x^{(k)}\}$ converge ad α per ogni scelta di $x^{(0)} \in \mathcal{J}$. Inoltre:

$$\lim_{k \rightarrow \infty} \frac{x^{(k+1)} - \alpha}{x^{(k)} - \alpha} = \phi'(\alpha). \quad (6.25)$$

La quantità $|\phi'(\alpha)|$ è chiamata *fattore asintotico di convergenza* e, in analogia con il caso dei metodi iterativi per la risoluzione di sistemi lineari, si definisce la

velocità asintotica di convergenza

$$R = \log \frac{1}{|\phi'(\alpha)|} \quad (6.26)$$

Dimostrazione. Le ipotesi 1. e 2. assicurano che la funzione di iterazione ϕ ammette almeno un punto fisso in \mathcal{J} . L'ipotesi 3. afferma che ϕ è una *contrazione* e garantisce l'unicità del punto fisso. Supponiamo infatti per assurdo che esistano due valori distinti $\alpha_1, \alpha_2 \in \mathcal{J}$ tali che $\phi(\alpha_1) = \alpha_1$ e $\phi(\alpha_2) = \alpha_2$. Dal teorema del valor medio segue allora che esiste $\eta \in (\alpha_1, \alpha_2)$ tale che:

$$|\alpha_2 - \alpha_1| = |\phi(\alpha_2) - \phi(\alpha_1)| = |\phi'(\eta)(\alpha_2 - \alpha_1)| \leq K|\alpha_2 - \alpha_1| < |\alpha_2 - \alpha_1|,$$

da cui segue necessariamente che $\alpha_2 = \alpha_1$. L'analisi della convergenza della successione $\{x^{(k)}\}$ si basa di nuovo sul teorema del valor medio; per ogni $k \geq 0$ esiste infatti un valore $\eta^{(k)}$ compreso tra α e $x^{(k)}$ tale che:

$$x^{(k+1)} - \alpha = \phi(x^{(k)}) - \phi(\alpha) = \phi'(\eta^{(k)})(x^{(k)} - \alpha) \quad (6.27)$$

da cui $|x^{(k+1)} - \alpha| \leq K|x^{(k)} - \alpha| \leq K^{k+1}|x^{(0)} - \alpha| \rightarrow 0$ per $k \rightarrow \infty$. Quindi $x^{(k)}$ converge ad α e dalla (6.27) segue

$$\lim_{k \rightarrow \infty} \frac{x^{(k+1)} - \alpha}{x^{(k)} - \alpha} = \lim_{k \rightarrow \infty} \phi'(\eta^{(k)}) = \phi'(\alpha),$$

ovvero la (6.25). \diamond

Il Teorema 6.1 garantisce la convergenza della successione $\{x^{(k)}\}$ alla radice α per qualsiasi scelta del valore iniziale $x^{(0)} \in \mathcal{J}$. Come tale, esso rappresenta un esempio di risultato di convergenza *globale*. Nella pratica è però spesso difficile delimitare *a priori* l'ampiezza dell'intervallo \mathcal{J} ; in tal caso è utile il seguente risultato di convergenza *locale*, per la cui dimostrazione si rimanda a [123]:

Proprietà 6.3 (teorema di Ostrowski) Sia α un punto fisso di una funzione ϕ continua e differenziabile in un intorno \mathcal{J} di α . Se risulta $|\phi'(\alpha)| < 1$, allora esiste $\delta > 0$ in corrispondenza del quale la successione $\{x^{(k)}\}$ converge ad α , per ogni $x^{(0)}$ tale che $|x^{(0)} - \alpha| < \delta$.

Esempio 6.6 Il Teorema di Ostrowski fornisce una condizione solo *sufficiente* per garantire la convergenza del metodo (6.24) come evidenziato dal seguente esempio. Sia $\phi(x) = x - x^3$, che ammette $\alpha = 0$ come punto fisso. Pur risultando $\phi'(\alpha) = 1$, se $x^{(0)} \in [-1, 1]$ allora la successione $x^{(k)} \in (-1, 1)$ per ogni $k \geq 1$ e converge asintoticamente ad α (se $x^{(0)} = \pm 1$ si ha addirittura $x^{(k)} = \alpha$ per ogni $k \geq 1$). Assumendo ad esempio $x^{(0)} = 1/2$ e come criterio di arresto per l'iterazione di punto fisso la condizione $|x^{(k+1)} - x^{(k)}| < 10^{-5}$, il metodo converge in 1072 iterazioni. \bullet

Diremo che un metodo di punto fisso ha *ordine* p (con p non necessariamente intero) se la successione da esso generata converge al punto fisso α con ordine p nel senso della Definizione 6.1.

Proprietà 6.4 Se $\phi \in C^{p+1}(\mathcal{J})$ per un opportuno intorno \mathcal{J} di α e per un intero $p \geq 0$, e se $\phi'(\alpha) = \phi''(\alpha) = \dots = \phi^{(p)}(\alpha) = 0$ e $\phi^{(p+1)}(\alpha) \neq 0$, allora il metodo di punto fisso ha ordine $p + 1$ e risulta inoltre

$$\lim_{k \rightarrow \infty} \frac{x^{(k+1)} - \alpha}{(x^{(k)} - \alpha)^{p+1}} = \frac{\phi^{(p+1)}(\alpha)}{(p+1)!}, \quad p \geq 0 \tag{6.28}$$

Dimostrazione. Verifichiamo la (6.28). A tal fine sviluppiamo ϕ in serie di Taylor nell'intorno di $x = \alpha$. Risulta

$$x^{(k+1)} - \alpha = \phi(x^{(k)}) - \phi(\alpha) = \phi'(\alpha)(x^{(k)} - \alpha) + \frac{\phi''(\alpha)}{2!}(x^{(k)} - \alpha)^2 + \dots + \frac{\phi^{(p)}(\alpha)}{p!}(x^{(k)} - \alpha)^p + \frac{\phi^{(p+1)}(\eta)}{(p+1)!}(x^{(k)} - \alpha)^{p+1},$$

per un certo η compreso tra $x^{(k)}$ e α . Si ha allora

$$\lim_{k \rightarrow \infty} \frac{(x^{(k+1)} - \alpha)}{(x^{(k)} - \alpha)^{p+1}} = \lim_{k \rightarrow \infty} \frac{\phi^{(p+1)}(\eta)}{(p+1)!} = \frac{\phi^{(p+1)}(\alpha)}{(p+1)!}.$$

A parità di ordine di convergenza, quanto più piccola risulterà la quantità a secondo membro nella (6.28), tanto più rapida sarà la convergenza della successione al punto fisso α . \diamond

Il metodo di punto fisso (6.24) è implementato in linguaggio MATLAB nel Programma 41. La variabile phi contiene l'espressione della funzione di iterazione ϕ .

Programma 41 - fixpoint : Metodo di punto fisso

```
function [xvect,xdif,fx,nit]=fixpoint(x0,mmax,toll,fun,phi)
err=toll+1; nit=0; xvect=x0; fx=eval(fun); xdif=[];
while (nit<mmax & err>toll),
    nit=nit+1; x=xvect(nit); xn=eval(phi); err=abs(xn-x);
    xdif=[xdif; err]; x=xn; xvect=[xvect;x]; fx=[fx;eval(fun)];
end;
```

6.3.1 Risultati di convergenza per alcuni metodi di punto fisso

È interessante analizzare, alla luce del Teorema 6.1, alcuni dei metodi iterativi introdotti sulla base di considerazioni geometriche nella Sezione 6.2.2.

Il metodo delle corde. La (6.17) è un caso particolare della (6.24) in cui si ponga $\phi(x) = \phi_{corde} = x - q^{-1}f(x) = x - (b-a)/(f(b) - f(a))f(x)$. La condizione $|\phi'_{corde}(\alpha)| < 1$ richiede che risulti

$$0 < q^{-1}f'(\alpha) < 2. \tag{6.29}$$

La pendenza q della corda deve dunque avere lo stesso segno di $f'(\alpha)$. Se quest'ultima è uguale a zero la (6.29) non può essere soddisfatta, mentre in caso contrario essa impone la seguente restrizione sulla scelta dell'ampiezza dell'intervallo di ricerca $[a, b]$

$$(b-a) < 2 \frac{f(b) - f(a)}{f'(\alpha)}.$$

Il metodo delle corde converge in una sola iterazione se f è una retta, altrimenti ha convergenza lineare, salvo il caso in cui risulti (fortuitamente) $f'(\alpha) = (f(b) - f(a))/(b-a)$, e dunque $\phi'_{corde}(\alpha) = 0$.

Il metodo di Newton. La (6.22) rientra nella forma (6.24) ponendo

$$\phi_{Newton}(x) = x - \frac{f(x)}{f'(x)}. \tag{6.30}$$

Supponendo $f'(\alpha) \neq 0$ (ovvero α radice semplice), si ha:

$$\phi'_{Newton}(\alpha) = 1 - \frac{[f'(\alpha)]^2 - f(\alpha)f''(\alpha)}{[f'(\alpha)]^2} = 0, \quad \phi''_{Newton}(\alpha) = \frac{f'''(\alpha)}{f'(\alpha)}. \tag{6.31}$$

Qualora la radice α abbia molteplicità $m > 1$ il metodo (6.22) non risulta più del second'ordine. In effetti, risulta (si veda l'Esercizio 2)

$$\phi'_{Newton}(\alpha) = 1 - \frac{1}{m}. \tag{6.32}$$

Qualora sia noto a priori il valore di m , è possibile ripristinare la convergenza quadratica del metodo di Newton introducendo il cosiddetto metodo di Newton modificato

$$x^{(k+1)} = \phi_{Newton}^m(x^{(k)}), \quad k \geq 0,$$

dove la funzione di iterazione è ora data da

$$\phi_{Newton}^m(x) = x - m \frac{f(x)}{f'(x)} \tag{6.33}$$

Per la verifica dell'ordine si veda l'Esercizio 2.

6.4 Radici di polinomi algebrici

Nel caso in cui f sia un polinomio algebrico di grado n a coefficienti reali, esistono per il calcolo delle sue radici metodi *ad hoc* che sfruttano la speciale struttura della funzione. Sia dunque $p_n \in \mathbb{P}_n$ un polinomio algebrico di grado $\leq n$. Come già osservato, esso ammette n radici, reali o complesse, ciascuna contata con la propria molteplicità. Ciò consente di rappresentare p_n nella forma "fattorizzata"

$$p_n(x) = a_n(x - \alpha_1)^{m_1} \dots (x - \alpha_k)^{m_k}, \quad \sum_{i=1}^k m_i = n$$

dove α_i e m_i indicano l' i -esima radice di p_n e la relativa molteplicità. Si nota che essendo i coefficienti a_k reali, se α è radice di p_n anche $\bar{\alpha}$ (il suo complesso coniugato) lo è.

Il teorema di Abelaffirma che per $n \geq 5$ non esistono rappresentazioni esplicite degli zeri di p_n (si veda ad esempio [115], Teorema 10.1). Ciò rende dunque indispensabile una valutazione numerica delle soluzioni dell'equazione $p_n(x) = 0$. Poiché i metodi numerici introdotti in precedenza necessitano di scegliere opportunamente un intervallo iniziale $[a, b]$ o un valore iniziale $x^{(0)}$, ricordiamo due risultati che possono essere utili per la localizzazione degli zeri di un polinomio.

Proprietà 6.5 (Regola dei segni di Cartesio) Sia $p_n \in \mathbb{P}_n$. Indichiamo con ν il numero di variazioni di segno nell'insieme dei coefficienti $\{a_j\}$, e con k il numero di radici reali positive di p_n (ciascuna contata con la propria molteplicità). Si ha allora che $k \leq \nu$ e $\nu - k$ è pari.

Proprietà 6.6 (Teorema di Cauchy) Tutti gli zeri di p_n sono inclusi nel cerchio Γ del piano complesso:

$$\Gamma = \{z \in \mathbb{C} : |z| \leq 1 + \eta_k\}, \quad \text{dove } \eta_k = \max_{0 \leq k \leq n-1} |a_k/a_n|. \quad (6.34)$$

Questa proprietà diventa di scarsa utilità qualora risulti $\eta_k \gg 1$. In tale circostanza è conveniente adottare una procedura di *traslazione* mediante un opportuno cambio di coordinate, come suggerito nell'Esercizio 10.

6.4.1 Il metodo di Horner e la deflazione

Illustriamo in questa sezione un metodo, noto come algoritmo di Horner, per la valutazione efficiente di un polinomio (e della sua derivata) in un punto assegnato z . Tale algoritmo consente di generare un procedimento automatico, detto metodo di *deflazione*, per la valutazione progressiva di tutte le radici di un polinomio.

Il metodo di Horner parte dall'osservazione che un polinomio $p_n \in \mathbb{P}_n$ può essere rappresentato mediante le seguenti due formule, del tutto equivalenti dal

punto di vista algebrico:

$$p_n(x) = \sum_{k=0}^n a_k x^k, \quad (6.35)$$

$$p_n(x) = a_0 + x(a_1 + x(a_2 + \dots + x(a_{n-1} + a_n x) \dots)). \quad (6.36)$$

La (6.35) richiede n addizioni e $2n - 1$ moltiplicazioni per calcolare $p_n(x)$, mentre la (6.36) richiede n addizioni più n moltiplicazioni. La seconda espressione, nota anche come algoritmo delle moltiplicazioni annidate (in inglese, *nested multiplications*), sta alla base del metodo di Horner. Quest'ultimo consente la valutazione efficiente del polinomio p_n in un punto z mediante il seguente algoritmo di *divisione sintetica*

$$\begin{aligned} b_n &= a_n, \\ b_k &= a_k + b_{k+1}z, \quad k = n-1, n-2, \dots, 0 \end{aligned} \quad (6.37)$$

Tutti i coefficienti b_k dipendono da z e si ha $b_0 = p_n(z)$. Il polinomio

$$q_{n-1}(x; z) = b_1 + b_2x + \dots + b_n x^{n-1} = \sum_{k=1}^n b_k x^{k-1} \quad (6.38)$$

di grado pari a $n - 1$ nella variabile x dipende dal parametro z (attraverso i coefficienti b_k) e si dice il *polinomio associato* a p_n .

È utile a questo punto ricordare il *teorema di divisione tra polinomi*:

dati due polinomi $h_n \in \mathbb{P}_n$ e $g_m \in \mathbb{P}_m$ con $m \leq n$, esistono due polinomi $\delta \in \mathbb{P}_{n-m}$ e $\rho \in \mathbb{P}_{m-1}$ tali che

$$h_n(x) = g_m(x)\delta(x) + \rho(x). \quad (6.39)$$

Dalla (6.39) si deduce che $p_n(x) = b_0 + (x - z)q_{n-1}(x; z)$, ovvero che $(x - z)$ è un divisore di p_n , q_{n-1} è il quoziente e b_0 è il resto. Se z è una radice di p_n , allora si ha $b_0 = p_n(z) = 0$ e quindi $p_n(x) = (x - z)q_{n-1}(x; z)$. In tal caso l'equazione algebrica $q_{n-1}(x; z) = 0$ fornisce le $n - 1$ radici restanti di $p_n(x)$. Questa osservazione suggerisce di adottare il seguente procedimento di *deflazione* per il calcolo di tutte le radici di p_n :

1. si trova una radice r di p_n con un opportuno metodo di approssimazione;
2. si calcola $q_{n-1}(r; r)$;
3. si pone $n = n - 1$, $p_n = q_n$ e si ritorna al punto 1.