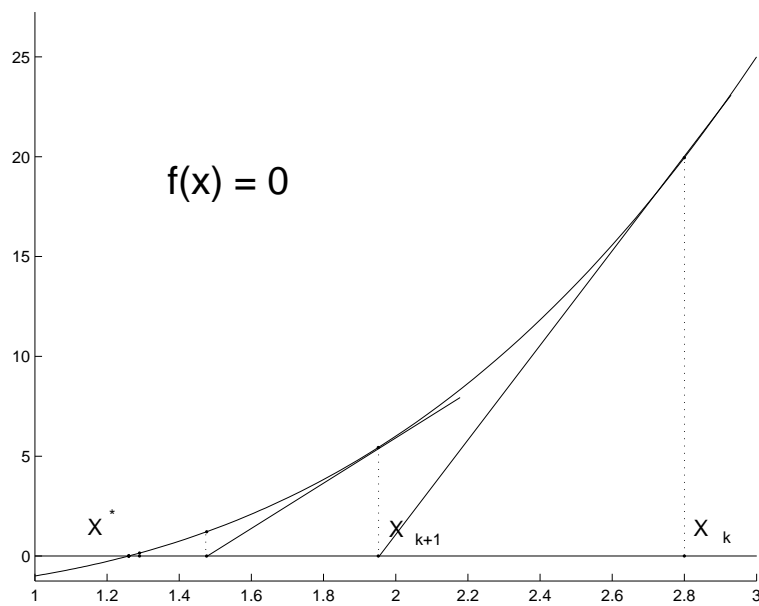


ARGOMENTI DEL CORSO CALCOLO NUMERICO

A.A. 2009/10



Equazioni non Lineari

Giulio Casciola

(novembre 2003, rivista e corretta novembre 2009)

Indice

1	Equazioni non Lineari	1
1.1	Metodo di Bisezione	1
1.2	Errore Inerente	3
1.3	Metodo di Newton	4
1.3.1	Metodi di Iterazione Funzionale	5
1.3.2	Convergenza del metodo di Newton	7
1.3.3	Propagazione degli errori	8
1.3.4	Test di arresto	9
1.3.5	Applicazioni: \sqrt{a} e $1/a$	11
1.3.6	Ordine di Convergenza	12
1.4	Metodo delle Secanti	14
1.5	Zeri di polinomi	16
1.5.1	Root-Isolate	18
1.5.2	Root	19
1.5.3	Metodo di Newton per polinomi	19
1.6	Applicazioni	20
1.6.1	Punto interno/esterno ad una curva	21
1.6.2	Punti estremi di una curva	22
1.6.3	Distanza di un punto da una curva	23
1.6.4	Intersezioni fra due curve	23
	Bibliografia	29

Capitolo 1

Equazioni non Lineari

Il problema di determinare le radici di un'equazione non lineare

$$f(x) = 0$$

nella variabile $x \in \mathbb{R}$, si presenta frequentemente nelle applicazioni. In generale non sono disponibili formule esplicite per calcolare le radici di tali equazioni, per cui si deve ricorrere a tecniche che consentono di approssimare le soluzioni con un prestabilito grado di precisione. In questa dispensa verranno dati gli elementi di base necessari per affrontare tale problema nel caso generale, quindi si particolarizzerà al caso di funzioni polinomiali per cui si discuterà uno fra i tantissimi approcci presenti in letteratura e si presenteranno alcune applicazioni significative.

1.1 Metodo di Bisezione

Sia $f(x) \in C_{[a,b]}$ tale che $f(a)f(b) < 0$. Allora esiste almeno una soluzione x^* di $f(x) = 0$ con $a < x < b$. Il metodo più semplice e intuitivo per approssimare x^* è il metodo di bisezione; questo procede suddividendo ripetutamente l'intervallo $[a, b]$ a metà e determinando in quale dei due sottointervalli si trova la soluzione. Così facendo l'ampiezza dell'intervallo che contiene x^* risulta ad ogni passo dimezzata fino ad essere piccola come la tolleranza desiderata. Più in dettaglio si considera il punto medio di $[a, b]$, $x_m = (a + b)/2$; si valuta $f(x_m)$ e si considera quale dei due intervalli $[a, x_m]$ o $[x_m, b]$ è nelle condizioni per contenere x^* ($f(a)f(x_m) < 0$, allora $x^* \in [a, x_m]$, altrimenti $x^* \in [x_m, b]$). Il procedimento viene ripetuto (metodo iterativo) sull'intervallo che è risultato contenere x^* e verrà arrestato quando l'ampiezza dell'intervallo in esame risulterà minore di una prefissata tolleranza tol .

Osservazione 1.1 *La costante tol non può essere scelta arbitrariamente piccola, perché lavorando con numeri finiti la condizione di arresto*

$$\left| \frac{b_k - a_k}{b_k} \right| < tol,$$

dove con $[a_k, b_k]$ si intende l'intervallo ottenuto al k -esimo passo del metodo, potrebbe non essere mai soddisfatta; infatti si ricorda che vale

$$\left| \frac{\alpha - fl(\alpha)}{\alpha} \right| < u$$

con u unità di arrotondamento. Da ciò si deduce che presi due numeri finiti consecutivi a e b con $a < b$ vale

$$\left| \frac{b - a}{b} \right| < 2u.$$

Questo deriva dal fatto che se $x \in \mathbb{R}$ fosse il punto medio fra a e b positivi, allora

$$\left| \frac{b - x}{x} \right| \simeq u \quad e \quad \left| \frac{x - a}{x} \right| \simeq u$$

da cui

$$2u \simeq \left| \frac{b - a}{x} \right| > \left| \frac{b - a}{b} \right|.$$

Ne consegue che tol non può essere scelto arbitrariamente, ma deve essere maggiore di $2u$; a tal fine il test di arresto deve essere così formulato:

$$|b_k - a_k| < tol + 2u|b_k|$$

dove ora tol può essere scelto liberamente.

Osservazione 1.2 *Si noti cosa accade se $x^* \equiv 0$; in questo caso sarà sempre $a_k < 0$ e $b_k > 0$, per cui*

$$\left| \frac{b_k - a_k}{b_k} \right| > 1;$$

per evitare questo caso è bene controllare se $f(0) \equiv 0$ quando $a < 0 < b$.

Di seguito viene presentato un semplice codice MATLAB che implementa quanto descritto.

```

function [xm,n]=sbisez(fun,a,b,tol)
% Questa routine determina una radice di una funzione con
% il metodo di bisezione
% fun --> funzione di cui det. uno zero
% a,b --> intervallo di definizione della funzione
% tol --> tolleranza richiesta
% xm <-- approssimazione desiderata della radice
% Questa routine fa uso della variabile eps di MATLAB che
% corrisponde a 2*u con u unita' di arrotondamento
    n=0;
    if ((a<0) & (b>0) & (feval(fun,0)==0)) then
        xm=0;
    else
        fa=feval(fun,a);
        fb=feval(fun,b);
        while (abs(b-a)>=(tol+eps*abs(b)));
            n=n+1;
            xm=(a+b)/2;
            fxm=feval(fun,xm);
            if (sign(fa)==sign(fxm))
                a=xm;
                fa=fxm;
            else
                b=xm;
                fb=fxm;
            end
        end
        xm=(a+b)/2;
    end
end

```

Osservazione 1.3 *Si noti che se nell'intervallo $[a, b]$ è contenuto un numero dispari di radici di $f(x) = 0$, allora il metodo di bisezione è ancora applicabile, ma determinerà l'approssimazione di una sola di esse.*

1.2 Errore Inerente

L'input per il problema di trovare le radici di una equazione non lineare $f(x) = 0$ consiste nell'espressione assegnata per definire la f ; nel caso polinomiale si tratterebbe della combinazione di certi coefficienti numerici per una certa base di rappresentazione. Lavorando in aritmetica finita,

la funzione che verrà considerata sarà differente dalla f , chiamiamola \hat{f} (nel caso polinomiale nella base delle potenze la \hat{f} sarà il polinomio che si ottiene considerando i coefficienti di f approssimati con numeri finiti). In definitiva il problema che andremo a risolvere sarà determinare le radici di $\hat{f}(x) = 0$ e non di $f(x) = 0$. Sia x^* una radice di $f(x) = 0$ e \hat{x}^* una radice di $\hat{f}(x) = 0$. Si indica l'errore sui dati con $E(x) = \hat{f}(x) - f(x)$ e si cerca una relazione con l'errore sul risultato $\hat{x}^* - x^*$ (si usa l'errore assoluto perché $f(x^*) = 0$). Si può dimostrare che

$$\hat{x}^* - x^* = \frac{E(\hat{x}^*)}{f'(\xi)} \quad \text{con } \xi \in (x^*, \hat{x}^*); \quad (1.1)$$

infatti per il teorema del valor medio vale

$$\frac{f(\hat{x}^*) - f(x^*)}{\hat{x}^* - x^*} = f'(\xi) \quad \xi \in (x^*, \hat{x}^*).$$

Ricordando che $f(x^*) = 0$ e che per definizione $E(\hat{x}^*) = f(\hat{x}^*) - f(x^*) = f(\hat{x}^*)$, risulterà

$$\frac{E(\hat{x}^*)}{\hat{x}^* - x^*} = f'(\xi)$$

da cui la 1.1.

Osservazione 1.4 *La differenza $\hat{x}^* - x^*$ dipende criticamente sia dalla derivata della f in prossimità della radice, quanto dall'errore di valutazione della f .*

1.3 Metodo di Newton

Supponiamo che $f(x) \in C_{[a,b]}^2$; sia $\bar{x} \in [a, b]$ un'approssimazione di x^* tale che $f'(\bar{x}) \neq 0$ e $|\bar{x} - x^*|$ sia piccolo. Si consideri il polinomio di Taylor per $f(x)$ in un intorno di \bar{x} e lo si valuti in x^*

$$f(x^*) = f(\bar{x}) + (x^* - \bar{x})f'(\bar{x}) + \frac{(x^* - \bar{x})^2}{2!}f''(\bar{x}) + \dots$$

Poiché $|\bar{x} - x^*|$ si è assunto piccolo, $(x^* - \bar{x})^2$ sarà ancora più piccolo e ancor di più i successivi termini; trascurando allora i termini non lineari si ha:

$$0 \simeq f(\bar{x}) + (x^* - \bar{x})f'(\bar{x});$$

risolvendo per x^* si ottiene:

$$x^* \simeq \bar{x} - \frac{f(\bar{x})}{f'(\bar{x})}$$

che viene ad essere un'approssimazione di x^* migliore di \bar{x} .

Questa relazione fornisce l'idea per il **metodo di Newton**, che consiste nel generare una successione $\{x_k\}$ definita da:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \quad k \geq 0 \quad \text{e} \quad f'(x_k) \neq 0 \quad \forall k. \quad (1.2)$$

L'interpretazione geometrica indica che ogni nuovo iterato x_{k+1} è dato dall'intersezione fra la retta tangente alla $y = f(x)$ nel punto x_k e l'asse x (ciò deriva dalla linearizzazione della $f(x)$ in x_k , da cui anche il nome **metodo delle tangenti**).

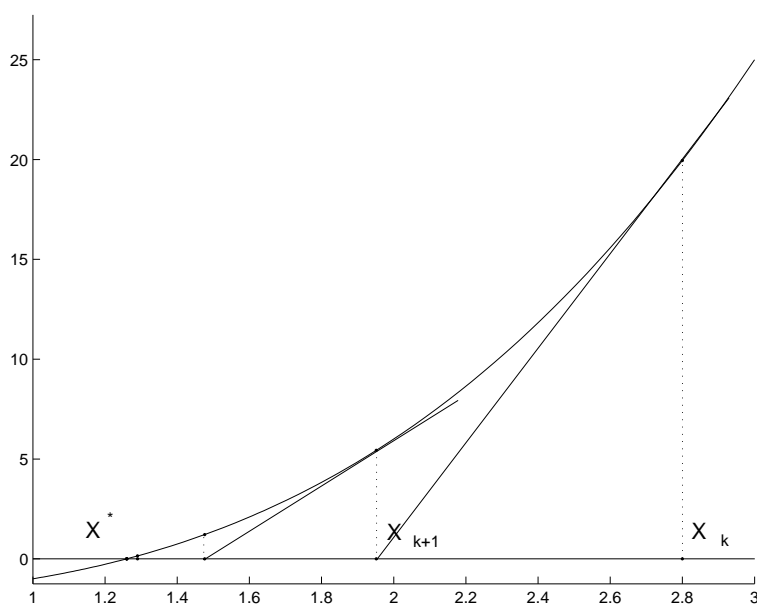


Figura 1.1: Iterazione del metodo di Newton.

Infatti, con riferimento alla Fig.1.1, se chiamiamo $\Delta y = f(x_k)$ e $\Delta x = x_k - x_{k+1}$ si ha che vale

$$\Delta x = \frac{\Delta y}{\tan \alpha} \quad \text{con} \quad \tan \alpha = f'(x_k).$$

Sostituendo si ottiene proprio la relazione 1.2 che dà il metodo di Newton.

1.3.1 Metodo di Iterazione Funzionale

Il metodo di Newton fa parte di una classe di metodi iterativi detti metodi di iterazione funzionale. Questi sono della forma

$$x_{k+1} = g(x_k) \quad k = 0, 1, \dots \quad (1.3)$$

con cui, a partire da un valore iniziale x_0 è possibile approssimare le soluzioni dell'equazione

$$x = g(x). \quad (1.4)$$

Le soluzioni di 1.4 sono dette **punti fissi** di $g(x)$. L'equazione $f(x) = 0$ viene trasformata in una equazione equivalente della forma 1.4. Uno dei modi con cui questa trasformazione può essere fatta è quella di usare una funzione $h(x)$ tale che l'equazione

$$x = x - \frac{f(x)}{h(x)}$$

abbia le stesse soluzioni di $f(x) = 0$ in un opportuno intervallo contenente la soluzione che si vuole approssimare.

Teorema 1.1 *Se $x = g(x)$ possiede un punto fisso x^* e se $g(x) \in C^1_{[x^* - \rho, x^* + \rho]}$ con $\rho > 0$ e soddisfa la condizione*

$$|g'(x)| \leq \lambda < 1 \quad \text{per } x \in [x^* - \rho, x^* + \rho],$$

allora per ogni $x_0 \in [x^ - \rho, x^* + \rho]$, tutti gli iterati x_k generati dalla 1.3 appartengono a questo intervallo e la successione converge ad x^* .*

dim.

Diremo che $x_k \rightarrow x^*$ per $k \rightarrow \infty$ se $\forall k$ vale

$$|x_{k+1} - x^*| < |x_k - x^*|.$$

Si ha

$$x_{k+1} - x^* = g(x_k) - g(x^*)$$

e per il teorema del valor medio

$$= g'(\xi_k)(x_k - x^*) \quad \text{con } \xi_k \in (x^*, x_k);$$

passando ai moduli si ha

$$|x_{k+1} - x^*| \leq \lambda |x_k - x^*| \quad \forall k$$

da cui, riapplicando lo stesso procedimento

$$|x_{k+1} - x^*| \leq \lambda |x_k - x^*| \leq \dots \leq \lambda^k |x_0 - x^*|$$

e quindi

$$\lim_{k \rightarrow \infty} |x_{k+1} - x^*| \leq \lim_{k \rightarrow \infty} \lambda^k |x_0 - x^*| = 0.$$

Le Fig.1.2 e 1.3 mostrano graficamente i possibili andamenti della successione $\{x_k\}$ a seconda della pendenza della funzione $g(x)$ in un intorno di x^* ; si osservi come le successioni $\{x_k\}$ convergono o meno a x^* a seconda che la $g(x)$ soddisfi o meno le condizioni di convergenza.

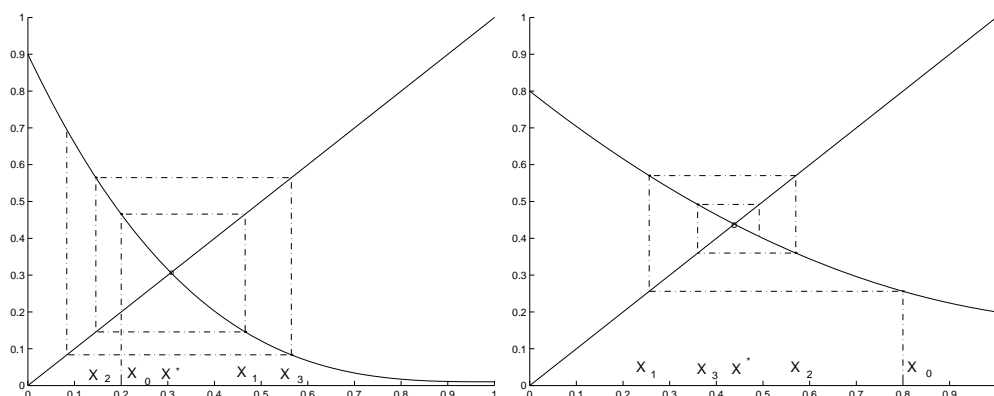


Figura 1.2: $g'(x) < -1$ (sinistra) e $-1 < g'(x) < 0$ (destra) per $x \in [x^* - \rho, x^* + \rho]$.

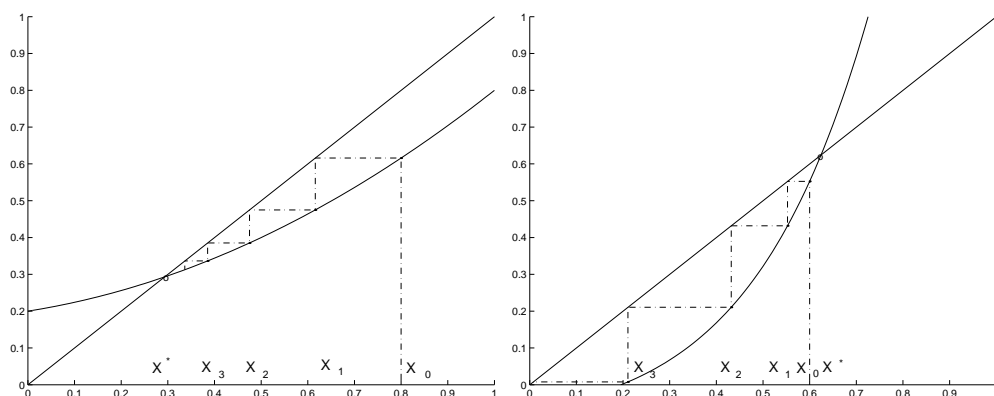


Figura 1.3: $0 < g'(x) < 1$ (sinistra) e $1 < g'(x)$ (destra) per $x \in [x^* - \rho, x^* + \rho]$.

Osservazione 1.5 Si può dimostrare che in $[x^* - \rho, x^* + \rho]$ non possono esistere due punti fissi distinti.

1.3.2 Convergenza del metodo di Newton

Tornando al metodo di Newton si vuole sfruttare il Teorema 1.1 per dimostrare il seguente risultato:

Teorema 1.2 Se $f(x) \in C_{[a,b]}^2$, $f(x^*) = 0$ e $f'(x^*) \neq 0$, allora esiste un intervallo $[x^* - \rho, x^* + \rho]$ tale che se $x_0 \in [x^* - \rho, x^* + \rho]$ il metodo di

Newton converge ad x^* .

dim.

Se si definisce

$$g(x) := x - \frac{f(x)}{f'(x)}$$

gli iterati del metodo di Newton sono dati da

$$x_{k+1} = g(x_k) \quad k = 0, 1, \dots$$

Allora il Teorema 1.1 regola la convergenza di tale successione ad x^* e dice che questo avviene se $|g'(x)| < 1$ per $x \in [x^* - \rho, x^* + \rho]$; nel nostro caso è:

$$g'(x) = \frac{f(x)f''(x)}{[f'(x)]^2}$$

ed è facile vedere che nelle ipotesi date sulla f , la $g'(x)$ è continua e si annulla in x^* ; allora esiste un intorno di x^* per cui $|g'(x)| < 1$ c.v.d.

1.3.3 Propagazione degli errori

A causa degli errori di arrotondamento che si commettono nel calcolo di $f(x)$ e conseguentemente di $g(x)$, la successione effettivamente calcolata $\{\tilde{x}_k\}$ può non essere convergente anche quando sono soddisfatte le ipotesi del Teorema 1.1.

Sia δ_k l'errore assoluto introdotto nel calcolo effettivo, in aritmetica finita, alla k -esima iterazione

$$\tilde{x}_k = g(\tilde{x}_{k-1}) + \delta_k \quad \text{con} \quad |\delta_k| < \delta \quad k = 0, 1, \dots,$$

allora vale il seguente

Teorema 1.3 *Sia x^* punto fisso di $g(x) \in C^1_{[x^*-\rho, x^*+\rho]}$ con $\rho > 0$ e sia $|g'(x)| \leq \lambda < 1$ per $x \in [x^* - \rho, x^* + \rho]$. Allora per ogni $\tilde{x}_0 \in [x^* - \rho, x^* + \rho]$, se*

$$\frac{\delta}{1 - \lambda} < \rho,$$

gli iterati \tilde{x}_k con gli errori δ_k (tutti limitati da δ) apparterranno all'intervallo $[x^ - \rho, x^* + \rho]$ e soddisfano la*

$$|\tilde{x}_k - x^*| \leq \frac{\delta}{1 - \lambda} + \lambda^k \left(\rho - \frac{\delta}{1 - \lambda} \right). \quad (1.5)$$

Osservazione 1.6 *Il secondo membro della 1.5 si compone di due termini; il primo può essere molto grande quando $\lambda \simeq 1$ e questo indipendentemente dall'indice di iterazione k ; il secondo termine tende a zero per $k \rightarrow \infty$, essendo $\lambda < 1$. Pertanto se $\lambda \simeq 1$ non si può essere certi che la successione $\{\tilde{x}_k\}$ converga ad x^* .*

Osservazione 1.7 *Si consideri la differenza di due iterati consecutivi; sfruttando il risultato del Teorema 1.3 si ha:*

$$\begin{aligned} |\tilde{x}_{k+1} - \tilde{x}_k| &= |(\tilde{x}_{k+1} - x^*) + (x^* - \tilde{x}_k)| \\ &\leq |\tilde{x}_{k+1} - x^*| + |x^* - \tilde{x}_k| \\ &\leq \frac{2\delta}{1-\lambda} + (\lambda^{k+1} + \lambda^k) \left(\rho - \frac{\delta}{1-\lambda} \right) \\ &= \frac{2\delta}{1-\lambda} + \lambda^k(1+\lambda) \left(\rho - \frac{\delta}{1-\lambda} \right). \end{aligned}$$

Ciò mostra che, comunque grande sia l'indice di iterazione k , non è possibile ottenere una maggiorazione della differenza di due iterati consecutivi più piccola di $\frac{2\delta}{1-\lambda}$ a causa degli errori di arrotondamento.

Questa stima è una misura dell'incertezza con cui è possibile determinare la soluzione per effetto degli errori di arrotondamento. Di questo è opportuno tenere conto nella scelta della tolleranza per le condizioni di arresto del metodo.

1.3.4 Test di arresto

Come test di arresto per un metodo di iterazione funzionale, ed in particolare per il metodo di Newton, si vorrebbe richiedere

$$\left| \frac{x_k - x^*}{x^*} \right| \leq tol,$$

ma non conoscendo x^* , ci si limiterà a chiedere

$$\left| \frac{x_k - x_{k-1}}{\max(|x_k|, |x_{k-1}|)} \right| \leq tol \quad \text{con} \quad tol > \frac{2\delta}{1-\lambda}.$$

Ancora però, pur pensando di approssimare δ con u , non è noto λ . Si osservi che nel caso del metodo di Newton, $\lambda \simeq 1$ può essere interpretato come un indice di mal condizionamento del problema cosa che si verifica, come abbiamo visto, sia a causa di errori nella valutazione della funzione

($E(\hat{x}^*)$) che del valore che assume la $f'(x)$ nell'intorno di x^* . In definitiva, nell'implementare il metodo di Newton, è bene prevedere almeno tre test al verificarsi di uno dei quali ci si deve arrestare:

1. $|x_k - x_{k-1}| \leq tol_1 + 2u \max(|x_k|, |x_{k-1}|)$
2. $|f(x_k)| \leq tol_2$
3. $k > MaxIter$

Con tol_1 tolleranza richiesta di approssimazione, tol_2 per stimare il mal-condizionamento e $MaxIter$ come limite massimo del numero di iterazioni. Di seguito viene presentato un semplice codice MATLAB che implementa quanto descritto.

```
function [xstar,n]=stangmet(fun,funp,x0,tol)
% Questa routine determina una radice di una funzione con
% il metodo di Newton o delle tangenti
% fun --> funzione
% funp--> funzione derivata
% x0 --> iterato iniziale
% tol --> tolleranza richiesta
% xstar <-- approssimazione desiderata della radice
% n <-- iterazioni effettuate
% Questa routine fa uso della variabile eps di MATLAB che
% corrisponde a 2*u con u unita' di arrotondamento e
% della variabile realmin di MATLAB che corrisponde al piu'
% piccolo numero finito positivo
n=0;
maxiter=50;
smallreal=100*realmin;
xstar=x0;
x0=x0+1;
while ((abs(xstar-x0)>=(tol+eps*abs(xstar))) & (n<maxiter))
    x0=xstar;
    fx0=feval(fun,x0);
    if (abs(fx0)>=smallreal)
        xstar=x0-fx0/feval(funp,x0);
        n=n+1;
    end
end
```

1.3.5 Applicazioni: \sqrt{a} e $1/a$

La radice quadrata positiva di un numero reale positivo o il reciproco di un reale (operazioni a tutt'oggi previste in hardware dallo standard IEEE) possono essere calcolate risolvendo rispettivamente le equazioni:

$$x^2 - a = 0 \quad \text{e} \quad \frac{1}{x} - a = 0.$$

Si osservi che agli inizi del calcolo automatico, la radice quadrata e la divisione erano realizzate via software trovando le radici delle equazioni suddette; le sole operazioni implementate in hardware erano l'addizione, la sottrazione e la moltiplicazione.

Procediamo nel trovare le radici di queste due equazioni applicando in entrambi i casi il metodo di Newton.

Equazione: $x^2 - a = 0$.

Sarà $f(x) = x^2 - a$ e $f'(x) = 2x$, allora

$$x_{k+1} = x_k - \frac{x_k^2 - a}{2x_k} = \frac{x_k^2 + a}{2x_k} = \frac{1}{2}\left(x_k + \frac{a}{x_k}\right).$$

Nell'innescare il metodo iterativo si può assumere come valore iniziale $x_0 = 1$; infatti essendo interessati alla radice positiva vale $|g'(x)| < 1$ in tutto il semiasse positivo (vedi Fig.1.4 sinistra). Se il valore iniziale 1 è molto distante dal valore cercato \sqrt{a} , il metodo impiegherà solo più iterazioni.

Equazione: $\frac{1}{x} - a = 0$.

Sarà $f(x) = \frac{1}{x} - a$ e $f'(x) = -\frac{1}{x^2}$, allora

$$x_{k+1} = x_k - \frac{\frac{1}{x_k} - a}{-\frac{1}{x_k^2}} = x_k + \frac{1 - ax_k}{x_k} x_k^2 = x_k(2 - ax_k).$$

Nell'innescare il metodo iterativo è importante assumere un opportuno valore iniziale x_0 (vedi Fig.1.4 destra). Il metodo infatti converge sicuramente se $0 < x_0 < 1/a$, ma negli altri casi potrebbe divergere. Assegnato a in forma normalizzata si consideri la sua parte esponente; se per esempio $a = 15000 = 0.15 \times 10^5$, l'esponente sarà 10^5 . Banalmente risulta $a \leq 10^5$ e $0 < 10^{-5} \leq \frac{1}{a}$. Allora si può assumere come valore iniziale il reciproco della parte esponente di a .

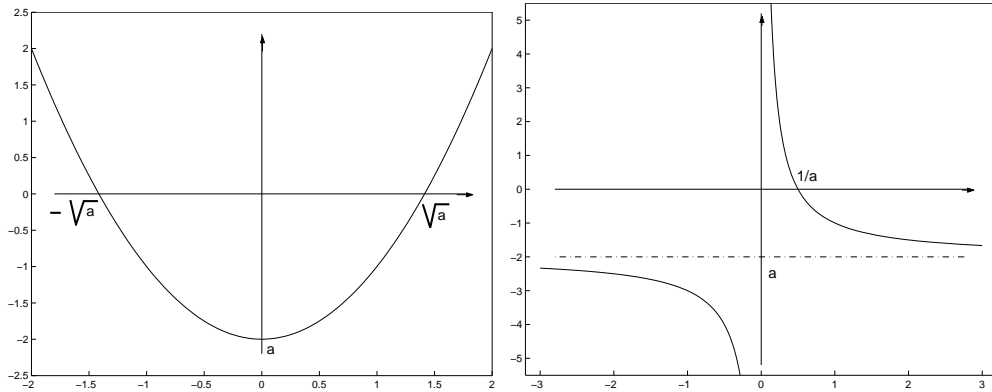


Figura 1.4: Equazione $x^2 - a = 0$ (sinistra) $\frac{1}{x} - a = 0$ (destra).

1.3.6 Ordine di Convergenza

Per confrontare differenti metodi iterativi che approssimano la stessa soluzione x^* di una equazione $f(x) = 0$, si può considerare la velocità con cui le successioni ottenute convergono alla soluzione. Lo studio della velocità di convergenza viene generalmente ricondotto a quello dell'ordine di convergenza del metodo.

Definizione 1.1 Sia $\{x_k\}$ una successione convergente ad x^* e sia $x_k \neq x^* \quad \forall k$. Se esiste un numero reale $p \geq 1$, tale che

$$\lim_{k \rightarrow \infty} \frac{|e_{k+1}|}{|e_k|^p} = \gamma \quad \text{dove} \quad e_k = x_k - x^*$$

con

$$\begin{cases} 0 < \gamma \leq 1 & \text{se } p = 1 \\ \gamma > 0 & \text{se } p > 1 \end{cases}$$

si dice che la successione ha **ordine di convergenza** p . La costante γ è detta **fattore di convergenza**.

Osservazione 1.8 Terminologia:

se $p = 1$ e $0 < \gamma < 1$ si dice che la convergenza è **lineare**;

se $p = 1$ e $\gamma = 1$ si dice che la convergenza è **sublineare**;

se $p > 1$ e $\gamma > 0$ si dice che la convergenza è **superlineare**;

Osservazione 1.9 Dalla definizione si ha che esiste una costante c tale che da un certo k in poi vale

$$|e_{k+1}| \leq c|e_k|^p$$

che misura la riduzione dell'errore assoluto ad ogni iterazione.

Teorema 1.4 *Nelle ipotesi del Teorema 1.2 si ha che il metodo di Newton ha ordine di convergenza 2.*
dim.

Si consideri lo sviluppo di $g(x)$ in serie di Taylor nell'intorno di x^* , e si utilizzi tale sviluppo nella relazione seguente:

$$\begin{aligned} x_{k+1} - x^* &= g(x_k) - g(x^*) \\ &= g(x^*) + g'(x^*)(x_k - x^*) + \frac{1}{2}g''(\xi_k)(x_k - x^*)^2 - g(x^*) \end{aligned}$$

ed essendo $g'(x^*) = 0$ si ha

$$|e_{k+1}| = \frac{1}{2}|g''(\xi_k)||e_k|^2.$$

Se si pone $c = \frac{1}{2}|g''(\xi_k)|$ si ha $p = 2$.

Osservazione 1.10 *Se $f'(x^*) \neq 0$, cioè x^* è una radice semplice di $f(x) = 0$, ogni iterazione dà un errore che è circa il quadrato di quello dell'iterazione precedente. Per $e_k \rightarrow 0$ il numero di cifre decimali esatte (anche cifre binarie) vengono approssimativamente raddoppiate ad ogni iterazione.*

Osservazione 1.11 *Se $f'(x^*) = 0$ (o x^* è una radice multipla), viene violata l'ipotesi del Teorema 1.2 e la velocità di convergenza risulta solo lineare.*

Esempio 1.1 *Sia $f(x) = x^3 - 3x + 2$; si vuole applicare il metodo di Newton per determinare le due radici $x_1^* = -2$ e $x_2^* = 1$.*

Si calcola la derivata prima $f'(x) = 3x^2 - 3$ e si costruisce il metodo iterativo:

$$x_{k+1} = x_k - \frac{x_k^3 - 3x_k + 2}{3x_k^2 - 3} = \frac{2x_k^3 - 1}{3x_k^2 - 1}.$$

Partendo da $x_0 = -2.4$ si calcola la successione per determinare $x_1^ = -2$; la Tab.1.1 mostra quanto si ottiene.*

Si osservi che numericamente si ha

$$|e_{k+1}| \simeq \frac{2}{3}|e_k|^2.$$

Partendo da $x_0 = 1.2$ si calcola la successione per determinare la radice doppia $x_2^ = 1$; La Tab.1.2 mostra quanto si ottiene.*

Si osservi che numericamente si ha

$$|e_{k+1}| \simeq \frac{1}{2}|e_k|.$$

k	x_k	$e_k = 2 - x_k$	$ e_{k+1} / e_k ^2$
0	-2.4	0.4	0.476190475
1	-2.076190476	0.076190476	0.619469086
2	-2.003596011	0.003596011	0.664202613
3	-2.000008589	0.000008589	
4	-2.000000000	0.000000000	

Tabella 1.1: Esempio di esecuzione

k	x_k	$e_k = 2 - x_k$	$ e_{k+1} / e_k $
0	1.2	0.2	0.515151515
1	1.103030303	0.103030303	0.508165253
2	1.052356420	0.052356420	0.496751115
3	1.026400811	0.026400811	0.509753688
4	1.013257730	0.013257730	0.501097775
5	1.006643419	0.006643419	

Tabella 1.2: Esempio di esecuzione

Osservazione 1.12 *La difficoltà nel metodo di Newton è trovare un valore iniziale x_0 sufficientemente prossimo ad x^* così da soddisfare le ipotesi del Teorema 1.2. Per questo motivo, il metodo di Newton deve essere preceduto dall'applicazione di metodi globalmente convergenti come quello di bisezione. In definitiva ed in pratica, il metodo di Newton deve essere visto come un metodo terminale veloce (che agisce localmente) a seguito di un metodo iniziale più lento (che agisce globalmente).*

1.4 Metodo delle Secanti

Ci sono funzioni per cui è molto economica la valutazione della $f'(x)$ una volta valutata la $f(x)$. Per altre funzioni, la valutazione della $f'(x)$ è tanto costosa quanto una seconda valutazione della $f(x)$. Per altre funzioni ancora, valutare $f'(x)$ è quasi impossibile.

Quando si cercano gli zeri di una funzione per la quale la valutazione della $f'(x)$ è difficoltosa, il metodo delle secanti è spesso una scelta migliore che quella di Newton. Tale metodo parte da due valori x_0 e x_1 . Ad ogni iterazione si determina x_{k+1} come l'intersezione fra la retta passante per i punti $(x_k, f(x_k))$ e $(x_{k+1}, f(x_{k+1}))$ e l'asse x (vedi Fig.1.5).

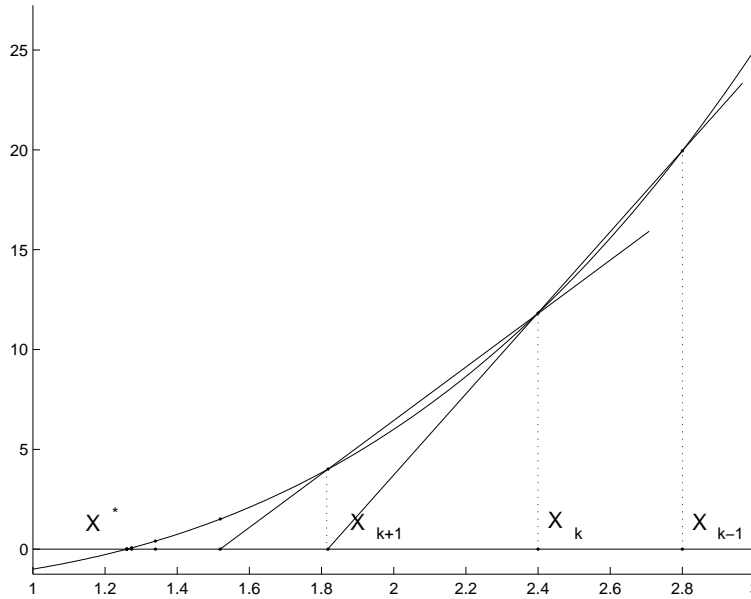


Figura 1.5: Iterazione del metodo delle secanti.

Da cui si ha la relazione:

$$x_{k+1} = x_k - \frac{f(x_k)}{f(x_k) - f(x_{k-1})}(x_k - x_{k-1}).$$

Questa si può anche vedere considerando il metodo di Newton in cui alla $f'(x_k)$ si sostituisce il rapporto incrementale

$$\frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}.$$

Teorema 1.5 Se $f(x) \in C_{[a,b]}^2$, $f(x^*) = 0$, $f'(x^*) \neq 0$ e $f''(x^*) \neq 0$, allora esiste un intervallo $[x^* - \rho, x^* + \rho]$ tale che se $x_0, x_1 \in [x^* - \rho, x^* + \rho]$ il metodo delle secanti converge ad x^* . Inoltre il

$$\lim_{k \rightarrow \infty} \frac{|e_{k+1}|}{|e_k|^p} = c \neq 0$$

dove $p = \frac{1}{2}(\sqrt{5} + 1) \simeq 1.618 \dots$

Questo Teorema afferma che, per buoni valori iniziali, il metodo delle secanti converge ad una radice semplice con ordine di convergenza $1.618 \dots$. Come già nel metodo di Newton, il più grosso problema del metodo delle

secanti è determinare x_0 e x_1 abbastanza prossimi ad x^* così da garantire la convergenza.

Di seguito viene presentato un semplice codice MATLAB che implementa quanto descritto.

```
function [xstar,n]=ssecmet(fun,x0,x1,tol)
% Questa routine determina una radice di una funzione
% con il metodo delle secanti
% fun --> funzione analitica
% x0,x1 --> iterati iniziali
% tol --> tolleranza richiesta
% xstar <-- approssimazione desiderata della radice
% n <-- iterazioni effettuate
% Questa routine fa uso della variabile eps di MATLAB che
% corrisponde a 2*u con u unita' di arrotondamento e
% della variabile realmin di MATLAB che corrisponde al piu'
% piccolo numero finito positivo
n=0;
maxiter=50;
smallreal=100*realmin;
xstar=x1;
x1=x0;
fx1=feval(fun,x1);
while ((abs(xstar-x1)>=(tol+eps*abs(xstar))) & (n<maxiter));
    x0=x1;
    fx0=fx1;
    x1=xstar;
    fx1=feval(fun,x1);
    if (abs(fx1)>=smallreal)
        xstar=x1-fx1*(x1-x0)/(fx1-fx0);
        n=n+1;
    end
end
end
```

1.5 Zeri di polinomi

In questa sezione si affronta il problema di determinare le radici di una equazione polinomiale od anche zeri di un polinomio. Come è noto, il teorema fondamentale dell'algebra, assicura che ogni polinomio di grado n ha n radici reali o complesse. Se il polinomio è a coefficienti reali e una radice è complessa anche la complessa coniugata sarà una radice.

In letteratura sono stati proposti numerosi metodi numerici per la determinazione di tutte le radici di un polinomio e ogni libreria scientifica è dotata di una tale routine. Per esempio il sistema MATLAB prevede la routine **roots**.

Nelle applicazioni pratiche, soprattutto della Computer Graphics e Modellazione Geometrica, è molto utile determinare tutte e sole le radici reali in un intervallo. Questa è la motivazione che ci spinge a presentare alcuni risultati per questo particolare problema a partire da un polinomio espresso nella base di Bernstein. Anche in questo particolare caso sono molti i metodi noti in letteratura e tutti sfruttano il significato geometrico che hanno i coefficienti di un polinomio di Bernstein (vedi dispensa su Interpolazione con Funzioni Polinomiali), dando luogo ad una classe di metodi geometrico/numerici.

Viene presentato un metodo, dovuto a Lane e Riesenfeld [LaRi81] per isolare e approssimare le radici reali di un polinomio nella base di Bernstein. Tale metodo è basato sulla proprietà di *Variation Diminishing* dei coefficienti del polinomio nella base di Bernstein e nell'uso di una tecnica di bisezione ricorsiva (vedi Suddivisione nella sopra citata dispensa).

La proposta consiste in una procedura *Root-Isolate* ricorsiva per determinare degli intervalli contenenti una sola radice controllando che il polinomio, ristretto a quell'intervallo nella base di Bernstein, abbia una sola variazione di segno dei coefficienti. Praticamente si esamina il numero di variazioni di segno dei coefficienti del polinomio: se questo è zero non ci sono radici; se è uno c'è una radice; se è > 1 si suddivide il polinomio nel punto medio e si chiama ricorsivamente la procedura *Root-Isolate* sui due polinomi. Alla fine si avranno in uno stack tanti polinomi che hanno una sola radice nell'intervallo di definizione. Viene inoltre proposta una procedura *Root* che per ogni polinomio memorizzato nello stack, cerca la sua unica radice. Si tratta ancora di una routine ricorsiva che suddivide il polinomio fino a che uno dei due coefficienti estremi non sia così piccolo (tolleranza richiesta), da indicare nell'estremo corrispondente la radice. La tecnica di suddivisione non avviene nel punto medio, ma nel punto in cui la poligonale interseca l'asse x . Questa scelta è dettata dal fatto che per suddivisione la poligonale converge alla funzione e quindi lo zero della poligonale deve convergere allo zero della funzione.

Osservazione 1.13 *Si osservi che una situazione come quella presentata in Fig.1.6, dove la radice del polinomio coincide con un estremo dell'intervallo di definizione, non presenta variazioni di segno dei coefficienti, e quindi la radice non viene localizzata. Una implementazione corretta deve accorgersi di una tale situazione.*

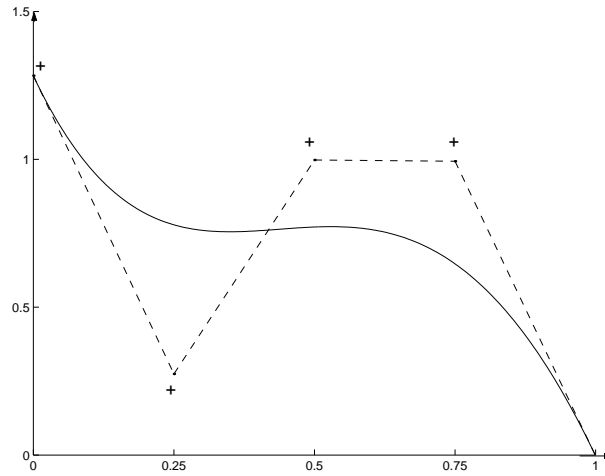


Figura 1.6: Zero in un estremo dell'intervallo.

Di seguito viene riportata la pseudocodifica dell'algoritmo, che è composto appunto da due procedure:

1.5.1 Root-Isolate

Calcola gli intervalli in cui vi è una sola variazione di segno del polinomio e isola gli zeri.

Parametri di **input**:

b_0, \dots, b_n coefficienti del polinomio,

x_0, x_1 estremi dell'intervallo del polinomio; alla prima chiamata i valori iniziali sono $x_0 = 0$ e $x_1 = 1$.

procedure *Root-Isolate*($b_0, \dots, b_n, x_0, x_1$)

Passo 1. [Test sul segno dei coefficienti].

if $V^-[b_0, \dots, b_n] = 0$

delete (b_0, \dots, b_n) e **return**.

if $V^-[b_0, \dots, b_n] = 1$

save il polinomio (si mette nello stack) e **return**.

Passo 2. [Suddivide usando de Casteljau e Ritesta].

Si calcoli $[b_0^{[0]}, \dots, b_n^{[n]}, b_n^{[n-1]}, \dots, b_n^{[0]}]$ per $\alpha = 1/2$;

Si calcoli $x_m \leftarrow (x_0 + x_1)/2$;

Root-Isolate($b_0^{[0]}, \dots, b_n^{[n]}, x_0, x_m$);

Root-Isolate($b_n^{[n]}, \dots, b_n^{[0]}, x_m, x_1$);

return.

1.5.2 Root

Questa procedura calcola gli zeri dei polinomi aventi una sola variazione di segno dei coefficienti.

Parametri di **input**:

b_0, \dots, b_n coefficienti del polinomio,

α_0, α_1 estremi dell'intervallo del polinomio.

procedure $Root(b_0, \dots, b_n, \alpha_0, \alpha_1, tol)$

Passo 1. [Test per la convergenza].

if $|b_0| < tol$ **output** α_0 e **return**.

if $|b_n| < tol$ **output** α_1 e **return**.

Passo 2. [Si determini la variazione di segno].

Si trovi k tale che $V^-[b_k, b_{k+1}] = 1$;

Passo 3. [Suddivide usando de Casteljau e Ritest].

Si calcoli

$$[b_0^{[0]}, \dots, b_n^{[n]}, b_n^{[n-1]}, \dots, b_n^{[0]}]$$

per $\alpha = (kb_{k+1} - (k+1)b_k) / (n(b_{k+1} - b_k))$;

Si calcoli il punto α_m dell'intervallo come

$$\alpha_m = (1 - \alpha) \cdot \alpha_0 + \alpha \cdot \alpha_1;$$

if $V^-[b_0^{[0]}, \dots, b_n^{[n]}] = 1$

$Root(b_0^{[0]}, \dots, b_n^{[n]}, \alpha_0, \alpha_m, tol)$;

else

$Root(b_n^{[n]}, \dots, b_n^{[0]}, \alpha_m, \alpha_1, tol)$;

return.

1.5.3 Metodo di Newton per polinomi

Nel caso polinomiale, il metodo di Newton, come procedura locale per convergere velocemente ad una radice, è molto attraente in quanto, una volta valutato il polinomio, è poco costoso valutare anche la sua derivata prima. Il vero problema, come precedentemente sottolineato, è sapere quando si sia nelle ipotesi per poter applicare il metodo, certi che converga od anche sapere quando ci si è localizzati ad un intervallo da un punto del quale, innescando Newton, si ha sicuramente convergenza della successione. A tal fine è molto interessante il seguente risultato:

Teorema 1.6 *Sia $f(x) \in C_{[a,b]}^2$ e siano soddisfatte le seguenti condizioni:*

1. $f(a)f(b) < 0$;

2. $f'(x) \neq 0 \quad \forall x \in [a, b]$;

3. $f''(x) \geq 0 \quad o \quad \leq 0 \quad \forall x \in [a, b]$;

4. sia c l'estremo di $[a, b]$ nel quale $|f'(x)|$ risulta minore e sia

$$\left| \frac{f(c)}{f'(c)} \right| \leq b - a.$$

Sotto queste ipotesi il metodo di Newton converge all'unica radice di $f(x) = 0$ per un qualsiasi iterato iniziale x_0 in $[a, b]$.

Osservazione 1.14 Se la $f(x)$ è un polinomio nella base di Bernstein e cioè

$$f(x) = \sum_{i=0}^n b_i B_{i,n}(x) \quad x \in [a, b]$$

è banale verificare le ipotesi del teorema e quindi stabilire se nell'intervallo di definizione si può applicare il metodo di Newton con la certezza di convergere all'unica radice presente; infatti avremo:

1. (segno di b_0) \neq (segno di b_n);
2. $b'_i := b_{i+1} - b_i \quad i = 0, \dots, n-1$ tutti dello stesso segno;
3. $b''_i := b'_{i+1} - b'_i = b_{i+2} - 2b_{i+1} + b_i \quad i = 0, \dots, n-2$ tutti dello stesso segno;
4. ($|b'_0| < |b'_{n-1}|$ e $|b_0/b'_0| \leq b-a$) oppure ($|b'_{n-1}| < |b'_0|$ e $|b_n/b'_{n-1}| \leq b-a$).
se queste sono verificate si consideri $x_0 = (a+b)/2$ e si applichi il metodo di Newton, il valore del polinomio e della sua derivata possono essere calcolati applicando l'algoritmo di de Casteljau una sola volta.

Spesso, in pratica, un test per controllare se siamo nelle ipotesi di applicazione del metodo di Newton può risultare più costoso che innescare Newton, fare un paio di iterazioni e controllare se queste stanno convergendo o meno. Più precisamente localizzate le radici con la procedura **Root-Isolate**, in ogni intervallo individuato si può innescare un Newton a partire dal punto medio dell'intervallo e controllare semplicemente che ogni iterato resti interno all'intervallo di partenza; se questo accade la successione sta convergendo e soddisfatto il test di arresto si restituisce la soluzione trovata, altrimenti si suddivide ricorsivamente l'intervallo in due parti e si innesca Newton su ciascuno.

1.6 Applicazioni

Vengono di seguito presentate alcune semplici, ma significative applicazioni della determinazione delle radici di una equazione polinomiale in un intervallo, per problemi di modellazione e interrogazione per curve di Bézier piane.

1.6.1 Punto interno/esterno ad una curva

Data una curva chiusa piana $\mathbf{C}(t)$ $t \in [0, 1]$ nella forma di Bézier (o Bézier a tratti) e un punto Q del piano, si vuole determinare se il punto è interno o esterno alla regione di piano delimitata dalla curva. La soluzione di questo problema potrebbe consistere nel determinare le intersezioni fra una semiretta, che per semplicità possiamo considerare orizzontale, uscente da Q e la curva $\mathbf{C}(t)$; se il numero di intersezioni risulta dispari, il punto è interno, se pari o zero, il punto è esterno (vedi Fig.1.7 sinistra).

Analizziamo, in generale, come poter determinare le intersezioni fra una

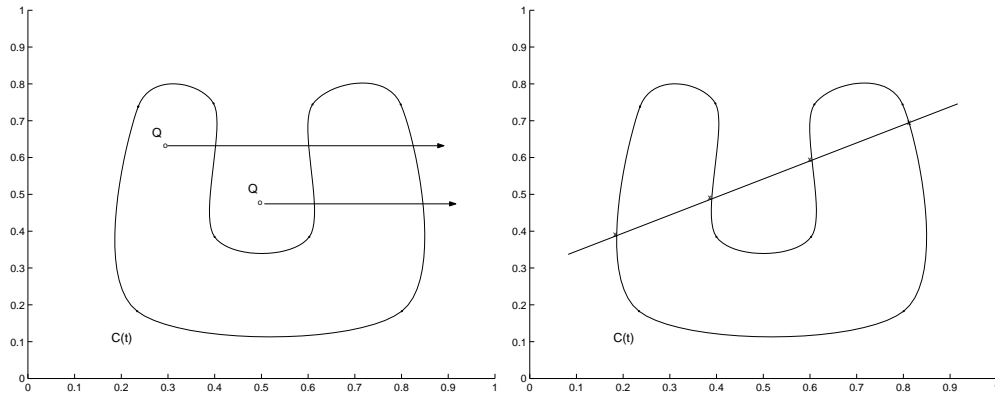


Figura 1.7: Punto interno/esterno ad una curva (sinistra); intersezione retta/curva (destra).

curva ed una retta del piano (vedi Fig.1.7 destra). Siano

$$\mathbf{C}(t) = \begin{pmatrix} C_1(t) \\ C_2(t) \end{pmatrix} \quad \text{e} \quad ax + by + c = 0$$

rispettivamente l'espressione della curva in forma parametrica con $t \in [0, 1]$ e l'equazione della retta in forma cartesiana. Sostituendo le componenti x ed y della curva nell'equazione della retta si ha:

$$aC_1(t) + bC_2(t) + c = 0.$$

Se $\mathbf{C}(t) = \sum_{i=0}^n P_i B_{i,n}(t)$ con $P_i = (x_i, y_i)^T$, sostituendo

$$a \sum_{i=0}^n x_i B_{i,n}(t) + b \sum_{i=0}^n y_i B_{i,n}(t) + c \sum_{i=0}^n B_{i,n}(t) = 0$$

e quindi

$$\sum_{i=0}^n (ax_i + by_i + c) B_{i,n}(t) = 0.$$

Si tratta di determinare le radici di un polinomio di grado n nell'intervallo $[0, 1]$. Le soluzioni saranno i parametri della curva in corrispondenza dei quali la curva e la retta si intersecano. La valutazione della curva in corrispondenza di questi parametri fornisce le coordinate cartesiane dei punti di intersezione.

Tornando al problema di determinare se un punto $Q = (qx, qy)^T$ è interno o esterno, è sufficiente considerare la retta orizzontale $y = qy$ e determinare le radici di

$$\sum_{i=0}^n (y_i - qy) B_{i,n}(t) = 0.$$

Si valuti la componente x della curva in corrispondenza delle soluzioni trovate, e si contino i valori trovati maggiori di qx (numero di intersezioni fra la curva e la semiretta orizzontale di origine Q e diretta come l'asse x positivo).

1.6.2 Punti estremi di una curva

Data una curva piana $\mathbf{C}(t)$ $t \in [0, 1]$ nella forma di Bézier (o Bézier a tratti), si vogliono determinare i punti estremi (vedi Fig.1.8 sinistra). Siano

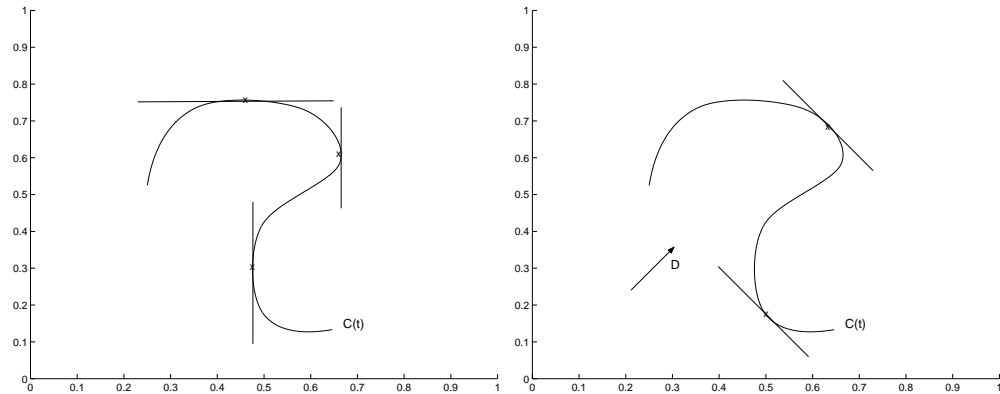


Figura 1.8: Punti estremi di una curva (sinistra); punti estremi secondo una fissata direzione D (destra).

$$\mathbf{C}(t) = \begin{pmatrix} C_1(t) \\ C_2(t) \end{pmatrix} \quad \text{e} \quad \mathbf{C}'(t) = \begin{pmatrix} C'_1(t) \\ C'_2(t) \end{pmatrix}$$

rispettivamente l'espressione della curva in forma parametrica con $t \in [0, 1]$ e l'espressione della curva derivata prima che rappresenta per ogni

punto t il vettore tangente alla curva $\mathbf{C}(t)$; i punti estremi saranno i punti della curva che hanno vettore tangente orizzontale o verticale e che quindi possono essere individuati come i punti della $\mathbf{C}'(t)$ con una delle componenti nulla e l'altra no. In pratica si cercano i punti t tali che:

$$C'_1(t^*) = 0 \quad \text{e} \quad C'_2(t^*) \neq 0$$

e

$$C'_2(t^*) = 0 \quad \text{e} \quad C'_1(t^*) \neq 0$$

Si tratta di determinare le radici di equazioni polinomiali e verificare che in corrispondenza di tali soluzioni l'altra componente non si annulli.

Se si cercano i punti estremi in una data direzione $D = (dx, dy)$ (vedi Fig.1.8 destra), basta determinare le radici dell'equazione polinomiale

$$\mathbf{C}'(t) \cdot D = 0$$

cioè

$$C'_1(t)dx + C'_2(t)dy = 0.$$

Se $\mathbf{C}(t) = \sum_{i=0}^n P_i B_{i,n}(t)$ con $P_i = (x_i, y_i)^T$, sarà $\mathbf{C}'(t) = n \sum_{i=0}^{n-1} (P_{i+1} - P_i) B_{i,n-1}(t)$ e si cercheranno le radici di

$$\sum_{i=0}^{n-1} [(x_{i+1} - x_i)dx + (y_{i+1} - y_i)dy] B_{i,n-1}(t) = 0.$$

1.6.3 Distanza di un punto da una curva

Data una curva piana $\mathbf{C}(t)$ $t \in [0, 1]$ nella forma di Bézier (o Bézier a tratti) e un punto Q del piano, si vuole determinare il punto della curva più prossimo a Q (vedi Fig.1.9). Si tratta di trovare quel punto della curva per cui

$$\mathbf{C}(t) - Q$$

risulta ortogonale alla tangente alla curva, cioè

$$(\mathbf{C}(t) - Q) \cdot \mathbf{C}'(t) = 0.$$

Si devono determinare gli zeri di un polinomio di grado $n(n-1)$.

1.6.4 Intersezioni fra due curve

Date due curve $\mathbf{C}(t)$ $t \in [0, 1]$ e $\mathbf{D}(s)$ $s \in [0, 1]$ si vogliono determinare i punti di intersezione (vedi Fig.1.10), che consistono a seconda delle applicazioni nelle coppie (t^*, s^*) , nei punti $Q^* = (qx^*, qy^*)$ o in entrambi. I

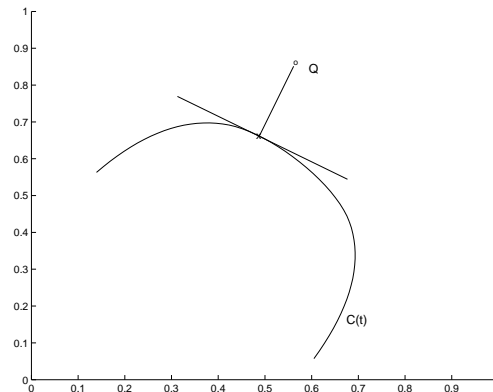


Figura 1.9: Distanza di un punto da una curva.

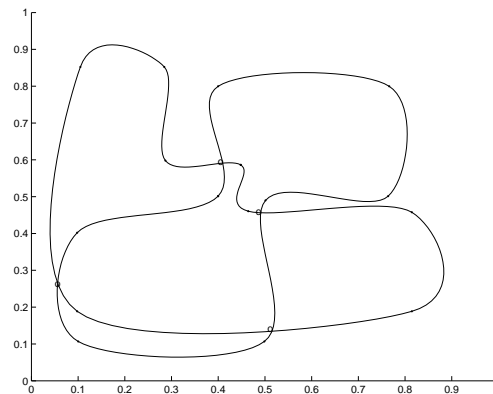


Figura 1.10: Intersezione fra due curve di Bézier a tratti.

metodi noti in letteratura sono numerosi e tutti interessantissimi; qui ne vedremo uno molto semplice che necessita in fase di preprocessing di trovare i punti estremi delle curve assegnate e nella fase finale di determinare l'intersezione fra due segmenti retti.

Il metodo consiste nel suddividere le due curve date in corrispondenza dei loro punti estremi ottenendo così due insiemi di tratti di curve che rappresentano rispettivamente la $\mathbf{C}(t)$ e la $\mathbf{D}(s)$; l'algoritmo continua nel confrontare ogni tratto della $\mathbf{C}(t)$ con quelli della $\mathbf{D}(s)$. Si osservi che ognuno di tali segmenti di curva è sicuramente contenuto nel rettangolo di vertici opposti gli estremi del segmento di curva (vedi Fig 1.11 sinistra). Questa osservazione è tanto semplice quanto potente. Infatti, in questo modo si può procedere con un test molto semplice e veloce per controllare

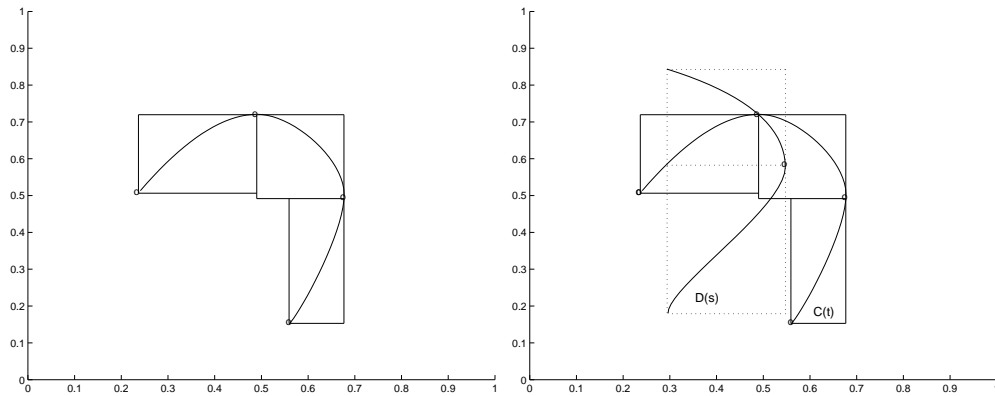


Figura 1.11: Punti estremi e rettangoli definiti dagli estremi dei tratti di curva (sinistra); intersezione di due curve (destra).

se due tratti di curva si intersecano o meno. Il controllo consiste nel determinare se il rettangolo definito dagli estremi di un tratto si interseca con il rettangolo di un altro (vedi Fig.1.11 destra). Se non si intersecano i rettangoli, non si intersecheranno nemmeno i tratti di curva; se si intersecano i rettangoli si procede suddividendo i due tratti a metà e ricontrollando i quattro accoppiamenti. Si osservi ulteriormente che anche eventuali suddivisioni di un tratto saranno sicuramente contenute nel rettangolo di vertici opposti gli estremi. In questo caso i rettangoli così determinati vengono utilizzati come convessi contenenti i segmenti di curva e la suddivisione, che consiste nel valutare la curva e memorizzare i coefficienti intermedi, serve solo per controllare quando arrestare il processo. Infatti se due tratti che si intersecano risultano essere dei segmenti retti a meno di una prefissata tolleranza (test di linearità dei punti di controllo associati), allora si arresta il processo di test e suddivisione e si intersecano i due segmenti (vedi algoritmo di intersezione fra due segmenti).

Elenco delle figure

1.1	Iterazione del metodo di Newton.	5
1.2	$g'(x) < -1$ (sinistra) e $-1 < g'(x) < 0$ (destra) per $x \in [x^* - \rho, x^* + \rho]$	7
1.3	$0 < g'(x) < 1$ (sinistra) e $1 < g'(x)$ (destra) per $x \in [x^* - \rho, x^* + \rho]$	7
1.4	Equazione $x^2 - a = 0$ (sinistra) $\frac{1}{x} - a = 0$ (destra).	12
1.5	Iterazione del metodo delle secanti.	15
1.6	Zero in un estremo dell'intervallo.	18
1.7	Punto interno/esterno ad una curva (sinistra); intersezione retta/curva (destra).	21
1.8	Punti estremi di una curva (sinistra); punti estremi secondo una fissata direzione D (destra).	22
1.9	Distanza di un punto da una curva.	24
1.10	Intersezione fra due curve di Bézier a tratti.	24
1.11	Punti estremi e rettangoli definiti dagli estremi dei tratti di curva (sinistra); intersezione di due curve (destra).	25

Bibliografia

- [BBCM92] R. Bevilacqua, D. Bini, M. Capovani, O. Menchi. *Metodi Numerici*. Zanichelli, 1992.
- [FaPr87] I. D. Faux, M. J. Pratt. *Computational Geometry for Design and Manufacture*. John Wiley & Sons, 1987.
- [LaRi81] J.M.Lane, R.F.Riesenfeld, “*Bounds on a Polynomial*”, *BIT*, Vol. 21, (1981), pp. 112-117