

(1.06) Osservazione (continua...)

(B) se f continua allora: per ogni k , $I(k)$ contiene almeno uno zero di f e

quando $k \rightarrow \infty$ si ha $x(k) \rightarrow \alpha$ e $f(\alpha) = 0$

(Dimostrazione ...)

(1.07) Criterio d'arresto.

Il metodo di bisezione è un *metodo iterativo*, ovvero un metodo che approssima l'oggetto cercato costruendo una *successione*. Poiché è materialmente impossibile costruire *tutti* gli elementi della successione, è *necessario* introdurre un criterio d'arresto, ovvero una condizione che, quando verificata, arresta la costruzione delle successione.

Un esempio di criterio d'arresto è: dato Δ numero reale *positivo* ...

se $\max I(k) < \Delta$ allora STOP

Proprietà del criterio d'arresto:

- (1) la *condizione* $\max I(k) < \Delta$ 'è *calcolabile*'
- (2) la condizione è *certamente verificata* dopo un numero *finito* di iterazioni (vedi l'Osservazione (B) in (1.06)): il criterio 'è *efficace*'
- (3) se f continua e k è tale che $\max I(k) < \Delta$ allora:

- esiste α in $I(k)$ zero di f
- $|x(k) - \alpha| < \max I(k) / 2 < \Delta/2 < \Delta$

ovvero la procedura restituisce un valore $x(k)$ che è un'approssimazione di uno zero di f , α , con *errore assoluto* $|x(k) - \alpha|$ minore di Δ : 'la procedura restituisce un'approssimazione *accurata quanto richiesto dall'utilizzatore*'.

Attenzione: Il criterio d'arresto utilizzato nella procedura *Bisezione* riportata in (1.05) *non* è efficace. La procedura *va modificata* inserendo un criterio efficace. Inoltre, il valore Δ va inserito tra le *variabili di ingresso* della procedura.

- Perché il criterio d'arresto utilizzato nella procedura riportata in (1.05) non è efficace?
- Sia α uno zero di f in $[a,b]$ e si consideri il criterio d'arresto seguente: dato Δ numero reale *positivo* ...

se $|x(k) - \alpha| < \Delta$ allora STOP

Quali delle tre proprietà elencate sopra sono verificate?

(1.08) Realizzazione Scilab.

```

function [z, v, info, k, mis] = bisezione(f, a, b, E, kmax)
//
// Uso:
//      [ z,v,info,[k,[mis]] ] = bisezione(f,a,b,E,kmax)
//
//
// Approssima uno zero della funzione  $f:[a,b] \rightarrow \mathbb{R}$ , che deve
// essere continua, con il metodo di bisezione. La funzione  $f$ 
// deve assumere valori non nulli e di segno opposto in  $a$  e  $b$ .
//
// L'iterazione si arresta quando:
// (*) la funzione  $f$  ha valore zero nel punto medio  $x_m$ 
// dell'intervallo considerato  $[a(k),b(k)]$ ;
// (*) l'intervallo considerato  $[a(k),b(k)]$  ha misura minore di
//  $E$ : in tal caso si ha, in teoria, che  $z$  approssima uno zero di
//  $f$  con errore assoluto non superiore ad  $E/2$ ;
// (*) dopo  $k_{\max}$  iterazioni.
//
//  $k_{\max}$ : valore opzionale (valore predefinito: 50).
//
//  $z$ : approssimazione finale (zero di  $f$  oppure punto medio
// dell'ultimo intervallo generato);
//  $v$ : valore di  $f$  in  $z$ ;
//  $\text{info} = 0$ : individuato valore in cui  $f$  si annulla ( $f(z) = 0$ );
//  $\text{info} = 1$ :  $f(z) \approx 0$  e l'ultimo intervallo considerato ha misura
// minore di  $E$  ( $\text{mis} < E$ );
//  $\text{info} = 2$ :  $f(z) \approx 0$ ,  $\text{mis} \geq E$  e il numero di iterazioni ha
// raggiunto il massimo consentito ( $k = k_{\max}$ );
//  $k$ : numero di iterazioni effettuate;
//  $\text{mis}$ : ampiezza dell'ultimo intervallo determinato.
//
//
// Inizializzazioni
//
if ~exists('kmax','l') then kmax = 50; end;
k_bis = 0; // contatore delle iterazioni eseguite
//
// Costruzione successioni
//
x_m = (a + b)/2;
f_m = f(x_m);
while (abs(b-a) >= E & f_m ~= 0 & k_bis < kmax),
    k_bis = k_bis+1;
    if sign(f_m) == sign(f(b)) then b = x_m; else a = x_m; end;
    x_m = (a + b)/2;
    f_m = f(x_m);
end;
//
// Fine costruzione: assegno variabili di uscita
//

```

```
z = x_m; v = f_m; k = k_bis; mis = abs(b-a);  
if f_m == 0 then info = 0;  
    else if abs(b-a) >= E then info = 2; else info = 1; end;  
end;  
//  
endfunction
```