

Oss (criterio d'arresto assoluto):

• $\delta > 0$ assegnato, criterio d'arresto: $\text{mis } I_k < \delta$

1) $\text{mis } I_k$ è calcolabile: $\text{mis } I_k = b_k - a_k$

2) di sup certamente verif dopo # finito iteraz

3) SE f continua:

• $\exists \alpha \in I_k$ t.c. $f(\alpha) = 0$ (teo \exists zeri...)

• $|x_k - \alpha| \leq \text{mis } I_k < \delta$

ovvero: è otten un'appross di α (zero di f) con
ERRORE ASSOLUTO inferiore a δ .

Es: dato $\delta > 0$, determ k t.c. $\text{mis } I_k < \delta$

(ovvero: determ il numero di it da fare perché
il cr d'arresto sia verificato)

(Sol. $k > \log_2 \text{mis } I_0 - \log_2 \delta$)

```

0001 // Esercitazione 3
0002 //
0003 // Argomenti:
0004 //
0005 // (1) Realizzazione del metodo di bisezione (while, printf).
0006 // (2) Uso della realizzazione per l'approssimazione di sqrt(2).
0007 //
0008 function [z, v]=Bisezione(f, a, b, delta)
0009 //
0010 // Applica il metodo di bisezione alla funzione f. L'iterazione si arresta
0011 // quando mis I(k) <= delta (criterio di arresto di tipo assoluto).
0012 //
0013 if f(a)*f(b) > 0 then
0014     error('la funzione non assume valori di segno opposto agli estremi');
0015 else
0016     x = (a+b)/2;
0017     while ((b-a) > delta & f(x) ~= 0),
0018         printf('a = %3.2e , b = %3.2e , ampiezza = %4.3e\n', a,b,b-a);
0019         if sign(f(a)) ~= sign(f(x)) then b = x;
0020         else a = x; end;
0021         x = (a+b)/2;
0022     end;
0023     z = x;
0024     v = f(z);
0025 end;
0026 endfunction
0027 //
0028 // Esercizio
0029 //
0030 // (A) Utilizzare opportunamente la procedura Bisezione per determinare
0031 // approssimazioni dello zero positivo della funzione fun(x) = x^2 - 2
0032 // con errore assoluto non superiore a:
0033 // (1) 10^(-4)
0034 // (2) 10^(-8)
0035 // (3) 10^(-16)
0036 // In ciascun caso visualizzare l'approssimazione trovata z ed il valore di
0037 // fun(z) con l'istruzione printf('z = %16.15e , fun(z) = %16.15e',z,v).
0038 //
0039 // (B) Modificare opportunamente la procedura Bisezione per realizzare una
0040 // procedura di intestazione
0041 //
0042 // function [z,v,ampiezza] = Bisezione(f,a,b,delta)
0043 //
0044 // che restituisce, oltre all'approssimazione z e v = f(z), l'ampiezza
0045 // dell'ultimo intervallo costruito.
0046 //

```