- Pagina web del corso ...
- Programma ... ⊕ ARITMETICA del CALCOLATORE

1) ZERI di funzioni (es: $f(x)=0$ ...)

2) SISTEMI di EQUAZIONI LINEARI (es: $Ax=b$ ...)

   * metodi DIRETTI (fattorizz di A ...)
   * metodi ITERATIVI (matrici SPARSE)

3) EQUAZIONI DIFFERENZIALI ORDINARIE

   (es: $\ddot{x} = F(\dot{x}, x)$ ...)

$\boxed{1}$ ZERI di FUNZIONI & ARITMETICA del CALCOLATORE

Pb: data $f:[a,b]\to\mathbb{R}$ continua t.c.

$\exists\, \alpha \in \mathbb{R}$ t.c. $f(\alpha)=0$, determinare α.

"ZERO di f"

TEO (esistenza degli zeri)

$f:[a,b]\to\mathbb{R}$ continua t.c. $f(a)\,f(b)<0$

$\Rightarrow \exists\, \alpha \in (a,b)$ t.c. $f(\alpha)=0$

- Metodo di BISEZIONE

Idea: utilizza il Teo di esistenza degli zeri per ottenere una succes di intervalli $I_k = [a_k, b_k]$ t.c.

- $\forall k$, $\exists$ zero di f in I_k
- $I_{k+1} \subset I_k$
- $\lim_{k \to \infty} \text{mis } I_k = 0$

descrizione del METODO di BISEZIONE

---

dati: $f : [a, b] \to \mathbb{R}$ t.c. $f(a) f(b) < 0$

- $a_0 = a$; $b_0 = b$; $I_0 = [a_0, b_0]$; $x_0 = \dfrac{a_0 + b_0}{2}$;
- per $k = 1, 2, 3, \ldots$ ripeti:

  se $f(x_{k-1}) = 0$ allora: STOP, altrimenti

  - se $f(x_{k-1}) f(b_{k-1}) < 0$ allora: $a_k = x_{k-1}$, $b_k = b_{k-1}$;

    altrimenti: $a_k = a_{k-1}$, $b_k = x_{k-1}$;
  - $I_k = [a_k, b_k]$; $x_k = \dfrac{a_k + b_k}{2}$;

uscita: quando un opportuno criterio d'arresto è verificato: x_k (pto medio dell'ultimo intervallo determinato)

Oss: • $\text{mis}\, I_k = b_k - a_k = \dfrac{\text{mis}\, I_{k-1}}{2^1} = \dfrac{\text{mis}\, I_{k-2}}{2^2} = \dots$

$$\dots = \frac{\text{mis}\, I_0}{2^k} \Rightarrow \boxed{\lim_{k\to\infty} \text{mis}\, I_k = 0}$$

• SE f continua allora: $\forall k$, I_k contiene uno zero di f e

$$\boxed{\lim_{k\to\infty} x_k = \alpha \quad \text{t.c.} \quad f(\alpha)=0}$$

CRITERIO di ARRESTO (necessario: non è possibile costruire tutta una successione in tempo finito)

• di "tipo ASSOLUTO":

dato δ reale positivo...

... $\boxed{\text{se} \quad \text{mis}\, I_k < \delta \quad \text{allora} \quad \text{STOP}}$

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

2) disuguaglianza certamente verificata dopo un numero finito di iterazioni...

3) SE f continua:

- $\exists\, \alpha \in I_k$ zero di f
- $|x_k - \alpha| \leq \dfrac{\text{mis}\, I_k}{2} < \dfrac{\delta}{2}$

dunque: si ottiene un'approx di α (utilizzando x_k) con $\boxed{\text{errore assoluto} < \dfrac{\delta}{2}}$

```
    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] -> 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 kmax iterazioni.
//
// kmax: valore opzionale (valore predefinito: 50).
//
// z: approssimazione finale (zero di f oppure punto medio
//    dell'ultimo intervallo generato);
// v: valore di f in z;
// info = 0: individuato valore in cui f si annulla (f(z) = 0);
//      = 1: f(z) ~= 0 e l'ultimo intervallo considerato ha misura
//           minore di E (mis < E);
//      = 2: f(z) ~= 0, mis >= E e il numero di iterazioni ha raggiunto
//           il massimo consentito (k = kmax);
// k: numero di iterazioni effettuate;
// 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
```