Lezione 2 del 25/02/2021

Titolo nota 25/02/2021

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 $x^{(k)}$ tale che

$$\lim_{k \to +\infty} x^{(k)} = \alpha$$

METODO di BISEZIONE

Si basa su

Teorema di esistenza degli zeri: Data una funzione $f : [a,b] \to \mathbb{R}$, continua in $[a,b]$ e tale che

$$f(a) f(b) < 0$$

allora $\exists\, \alpha \in (a,b)$ tale che $f(\alpha)=0$

Avviamo l'algoritmo

$I_0 = [a,b]$ $\qquad a^{(0)} = a \qquad b^{(0)} = b \qquad x^{(0)} = \dfrac{a^{(0)} + b^{(0)}}{2}$

$f(x^{(0)})f(a^{0}) < 0$ $\qquad\qquad$ $f(x^{(0)})\, f(b^{(0)}) < 0$

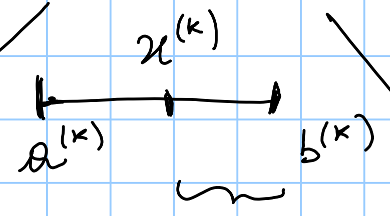
$a^{(0)}$ —— $x^{(0)}$ —— $b^{(0)}$

$a^{(1)} = a^{(0)} \qquad b^{(1)} = x^{(0)}$ $\qquad\qquad$ $a^{(1)} = x^{(0)} \qquad b^{(1)} = b^{(0)}$

$$x^{(1)} = \frac{a^{(1)} + b^{(1)}}{2}$$

...

al passo k $\quad I_k = [a^{(k)}, b^{(k)}]$ $\quad x^{(k)} = \dfrac{a^{(k)} + b^{(k)}}{2}$



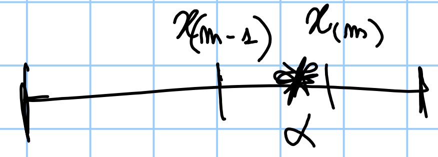
$f(x^{(k)})\, f(a^{(k)}) < 0$ $\qquad\qquad$ $f(x^{(k)})\, f(b^{(k)}) < 0$

$a^{(k+1)} = a^{(k)} \quad b^{(k+1)} = x^{(k)}$ $\qquad\qquad$ $a^{(k+1)} = x^{(k)} \quad b^{(k+1)} = b^{(k)}$

$$x^{(k+1)} = \frac{a^{(k+1)} + b^{(k+1)}}{2}$$

Oss: Il metodo di bisezione si arresta al passo m tale che $m>0$

$$e^{(m)} := |x^{(m)} - \alpha| \le |I_m| = |b^{(m)} - a^{(m)}|$$



e con ε tolleranza prefissata : $\boxed{e^{(m)} \leq \varepsilon}$

$$e^{(m)} \leq |I_m| \leq \varepsilon$$

$$|I_m| = \frac{|I_0|}{2^m} = \frac{b-a}{2^m} \xrightarrow[m \to +\infty]{} 0$$

Il metodo di bisezione è convergente? Ossia $\lim_{m \to +\infty} x^{(m)} = \alpha$?

Ossia

$$\lim_{m \to +\infty} \underbrace{|x^m - \alpha|}_{= e^{(m)}} = 0 \ ?$$ Sì perché

$$\boxed{e^{(m)} \leq |I_m| = \frac{b-a}{2^m} \xrightarrow[m \to +\infty]{} 0}$$

$\boxed{\text{Il metodo di bisezione è globalmente convergente}}$

Oss: $e^{(m)} \leq \boxed{\frac{b-a}{2^m} \leq \varepsilon}$ se $m > \frac{\log(b-a) - \log(\varepsilon)}{\log(2)}$

Def: I NUMERI MACCHINA sono i numeri rappresentabili esattamente in un calcolatore

Oss: Hanno un numero finito di cifre

Def: L'insieme dei numeri macchina è chiamato SISTEMA FLOATING POINT

Es.: Notazione scientifica

31.415

$\boxed{0.31415 \cdot 10^{5}}$

$3.1415 \cdot 10^{4}$

-0.00000123

$\boxed{-0.123 \cdot 10^{-5}}$

$-123 \cdot 10^{-8}$

<u>Oss</u>: Matlab rappresenta i numeri utilizzando 64 bit

$$x = \underset{\substack{\text{segno} \\ (1\,\text{bit})}}{s} \cdot \underbrace{\left(\frac{d_1}{\beta} + \frac{d_2}{\beta^2} + \ldots + \frac{d_t}{\beta^t} \right)}_{(d_1 \ldots d_t)\ \text{mantissa (52 bit)}} \cdot \beta^{p}$$

$p \to$ esponente (11 bit) $p \in [-1022, 1023]$

$\beta \to$ base

$0 \leq d_i \leq \beta - 1 \qquad i = 1 \dots t$

<u>Oss</u>: nella notazione posizionale $\boxed{x = \pm\, 0.d_1 d_2 \dots d_t \cdot \beta^p}$

<u>Oss</u>: Il più grande numero positivo memorizzabile è

REALMAX

al di sopra della quale si ha il livello di overflow

Il minimo numero $\boxed{\text{MAGGIORE di } 0}$ rappresentabile è

REALMIN

Altri valori più piccoli compresi tra 0 e il più piccolo positivo rappresentabile possono essere gestiti rinunciando al primo bit diverso da 0, ...

__Oss__: Alcuni valori speciali richiedono una codifica particolare: 0, +∞, −∞, NaN
eps

__Def__: L'EPSILON MACCHINA è il più piccolo numero macchina positivo x tale che

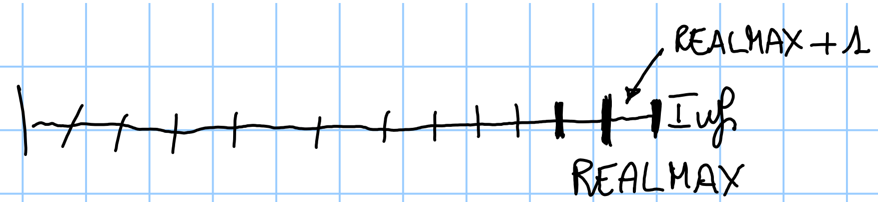
$$(1+x) > 1$$

L'epsilon macchina definisce una stima di quanto possa variare al più l'errore relativo, indica la sensibilità del sistema floating point adottato

__Def__: ERRORE ASSOLUTO $\quad E_{a} = |\text{valore esatto} - \text{valore approssimato}|$

ERRORE RELATIVO $\quad E_{rel} = \dfrac{E_a}{|\text{valore esatto}|}$

__Oss__: Matlab esegue tutti i calcoli in doppia precisione



REALMAX + 1
$\overline{I}$nf
REALMAX