Lez. 2 (26-02-25)

Operazioni aritmetiche di base

- `3+2` somma
- `3/4` divisione
- `3\4` divisione inversa ⇒ 4/8
- `3*5` moltiplicazione
- `3^4` elevamento a potenza
- `0/0` prodotto tra numeri complessi, restituisce `NaN`
- `1/0` restituisce `Inf` (divisione per zero)

Formattazione della visualizzazione dei numeri

- `format long` mostra i numeri con una precisione elevata
- `format short` mostra i numeri con una precisione più breve
- `format rational` mostra i risultati sotto forma di frazione quando possibile

Costanti e funzioni matematiche

- `pi` restituisce il valore π
- `realmax` restituisce il valore massimo rappresentabile
- `realmin` restituisce il valore minimo rappresentabile
- `eps` restituisce la precisione della macchina
- `exp(1)` calcola il valore di e
- `log(10)` logaritmo naturale di `10`
- `log10(10)` logaritmo base 10
- `sqrt(4)` calcola la radice quadrata

Funzioni trigonometriche

- `sin(9)` seno di 9 radianti
- `cos(-2)` coseno di -2 radianti
- `atan(2)` arcotangente di 2
- `asin(x)` arcoseno di x

Variabili e assegnazioni

- `s=3` assegna il valore `3` alla variabile `s`
- `s==4` confronta `s` con `4`, restituisce `false` (`0`)
- `d=4; c=3; e=d+s` assegna valori e somma `d` e `s`

Numeri complessi

- `c=2+3i` definizione di un numero complesso

- `i^2` restituisce `-1`, dal momento che non ho utilizzato `i` per altri scopi lo considera subito un numero immaginario
- `d=4+6i; d*c` moltiplicazione di numeri complessi

Controllo dell'errore numerico

- `1+eps>1` testa se `1+eps` è maggiore di `1`
- `abs(b-c)` calcola la differenza assoluto tra `b` e `c`

Cicli e array

- `S(1)=0; for i=1:10000 S(i+1)=S(i)+0.0001; end` crea un array `S` con incremento di `0.0001`
- `T=[0:0.0001:1];` crea un vettore `T` con valori da `0` a `1` con passo `0.0001`
- `F=[1:2:4]` crea un vettore `[1,3]` (valori tra `1` e `4` con passo `2`)

Comandi di pulizia

- `clc` cancella la command window
- `clear` cancella tutte le variabili
- `clear d` cancella solo la variabile `d`

Comandi di aiuto

- `help [comando]` mostra la documentazione del comando specificato

Def. L'EPSILON MACCHINA è il più piccolo numero macchina positivo x tale che $1 + x > 1$

OSS. Non è il più piccolo numero macchina positivo rappresentabile (REALMIN)

OSS. Definisce una stima di quanto possa variare l'errore relativo, indica quindi la sensibilità del sistema floating point adottato.

ES. **ERRORE DI RAPPRESENTAZIONE**

-
```
S(1) = 0            % S vettore con 1 riga e, dopo il for, 10001 colonne
for i = 1:10000     % i non è più un'unità immaginaria
    S(i+1) = S(i) + 0.0001;   % ';' per non vedere l'output
end                           di ogni iterazione
S(end)   % stampo l'ultimo numero memorizzato
```

=> OUTPUT: 0.9...906 (!= 1

Quello che vorrei dal for:

1° elemento = 0
2° elemento = 0.0001
3° elemento = 0.0002
e così via

fino ad arrivare all'ultimo elemento che dovrebbe essere 1, ma di fatto non lo è.

=> ERRORE DI RAPPRESENTAZIONE

- T = [0 : 0.0001 : 1]

vado da 0 a 1

voglio che in ogni elemento del vettore T ci sia:

$0 \cdot 10^{-4}$
$1 \cdot 10^{-4}$
$2 \cdot 10^{-4}$
e così via fino ad arrivare a 1

T(end) = 1

Restituisce 1 perché sotto questa sintassi c'è un algoritmo più PERFORMANTE della somma iterata con il ciclo for

OSS. Benché 0.0001 sia espresso esattamente in base decimale non e' rappresentato esattamente in aritmetica floating point.

ES. ERRORI DI CANCELLAZIONE NUMERICA : perdita di informazioni significative dovuta alla sottrazione di numeri molto vicini tra loro

$$x = 77\,777\,777$$

$$y = \sqrt{x^2+1} - x = \underline{0}$$

La precisione del calcolatore non è sufficiente per rappresentare un valore così piccolo dopo la sottrazione di due numeri molto simili

$$z = \underline{\frac{1}{\sqrt{x^2+1}+x}} = 6.4 \ldots \cdot 10^{-9}$$

z e' la stessa quantità che volevamo calcolare con y, ma senza la sottrazione problematica
=> z e' in una forma numericamente più stabile.

Es. studiamo il CONDIZIONAMENTO della SOMMA

$x, y \in \mathbb{R}$ $\quad \bar{x} = x(1+\varepsilon_1)$ $\quad \varepsilon_1$ errore sul dato x

$$e_a = |\bar{x} - x| = |x(1+\varepsilon_1) - x| =$$

$$= |x + x\varepsilon_1 - x|$$

$$= |x\varepsilon_1|$$

ε_1 non e' l'errore assoluto perchè non sono riuscita ad isolarlo

$$e_r = \frac{e_a}{|x|} = \frac{|x|\,|\varepsilon_1|}{|x|} = \varepsilon_1$$ abbiamo isolato ε_1

$\bar{y} = y(1+\varepsilon_2)$ $\quad\varepsilon_2$ errore relativo sul dato y

$$\frac{|\bar{x}+\bar{y}-(x+y)|}{|x+y|} = \frac{|\cancel{x}+x\varepsilon_1+\cancel{y}+y\varepsilon_2-\cancel{x}-\cancel{y}|}{|x+y|} \leq$$

disuguaglianza triangolare $|a+b| \leq |a|+|b|$

$$\leq \underbrace{\left[\frac{|x|}{|x+y|}\right.}_{K} \overbrace{|\varepsilon_1|}^{e_r(x)} + \underbrace{\left[\frac{|y|}{|x+y|}\right.}_{K} \overbrace{|\varepsilon_2|}^{e_r(y)}$$

$K \to +\infty$ se $x \to -y$

Se K e' "grande" la somma è mal condizionata
(denominatore circa 0 => x e y sono circa opposti)
(denominatore < numeratore)

OSS. Quando x e y sono circa opposti la somma è mal condizionata (es. cancellazione numerica)

Es. Studiamo il CONDIZIONAMENTO del PRODOTTO

$x, y \in \mathbb{R}$ $\quad \bar{x} = x(1+\varepsilon_1) \quad \bar{y} = y(1+\varepsilon_2)$

$$\frac{|\bar{x}\bar{y}-xy|}{|xy|} = \frac{|x(1+\varepsilon_1)\,y(1+\varepsilon_2)-xy|}{|xy|} =$$

$$= \frac{|(x+x\varepsilon_1)(y+y\varepsilon_2)-xy|}{|xy|} =$$

$$= \frac{|\cancel{x}y + x\cancel{y}\varepsilon_2 + x\cancel{y}\varepsilon_1 + x\cancel{y}\varepsilon_1\varepsilon_2 - \cancel{x}y|}{|x\cancel{y}|}$$

disuguaglianza triangolare

$$= |\varepsilon_1 + \varepsilon_2 + \varepsilon_1\varepsilon_2| \le |\varepsilon_1| + |\varepsilon_2| + |\varepsilon_1\varepsilon_2|$$

$0 \le (a-b)^2$

(a^2+b^2-2ab)

$ab \le \frac{1}{2}(a^2+b^2)$

$$\le |\varepsilon_1| + |\varepsilon_2| + \frac{1}{2}(\varepsilon_1^2 \ \varepsilon_2^2)$$

$$\le \underbrace{\frac{3}{2}}_{K}|\varepsilon_1| + \underbrace{\frac{3}{2}}_{K}|\varepsilon_2|$$

se $|\varepsilon| < 1$

$\varepsilon^2 < |\varepsilon|$

Oss. Il prodotto è "ben condizionato".

Oss. Definiamo: $\odot$ = op. aritmetica

$\boxed{\odot}$ = op. in aritmetica di macchina

$$x \boxed{\odot} y = fl(fl(x) \odot \boxed{fl}(y))$$

rappresentazione nel sistema floating point

In generale vale la proprietà commutativa (se vale in aritmetica esatta), ma non le altre proprietà.

es. $(a \boxplus b) \boxplus c \neq a \boxplus (b \boxplus c)$

RISOLUZIONE DI EQUAZIONI NON LINEARI

Consideriamo il problema $f: (a,b) \subset \mathbb{R} \to \mathbb{R}$

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

Sia $f \in C^m(a,b) = \{$funzioni derivabili m volte in $(a,b)\}$

Def.
- α si dice RADICE SEMPLICE se $f(\alpha)=0$ e $f'(\alpha) \neq 0$
- α si dice RADICE di ORDINE m se

$$f(\alpha) = f'(\alpha) = f''(\alpha) = \dots = f^{m-1}(\alpha) = 0$$

e $f^m(\alpha) \neq 0$

In tal caso $f(x) = (x-\alpha)^m h(x)$ con $h(\alpha) \neq 0$

due radici

ES. $f(x) = x^2 - 3x + 2$ $\qquad f'(x) = 2x - 3$

$= (x-2)(x-1) \qquad \alpha_1 = 1 \quad \alpha_2 = 2$

α_1 e' radice semplice $f(\alpha_1) = 1^2 - 3 \cdot 1 + 2 = 0$

$f'(\alpha_2) = 2 \cdot 1 - 3 = 2 - 3 \neq 0$

α_2 e' radice semplice $f(\alpha_2) = 2^2 - 3 \cdot 2 + 2 = 0$

$f'(\alpha_2) = 2 \cdot 2 - 3 \neq 0$

ES. $x^2 - 2x + 1 = 0$ $\qquad h(x) = 1$ $\qquad f'(x) = 2x - 2$

$= \underline{(x-1)}^2 \qquad \alpha = 1$ $\qquad f''(x) = 2$

$f(\alpha) = 1^2 - 2 \cdot 1 + 1 = 0 \qquad f'(\alpha) = 2 \cdot 1 - 2 = 0$

$f''(\alpha) = 2 \neq 0$ (α ha ordine 2)