01/03

→ parte dei problemi matematici non hanno soluzione in forma chiusa
→ gran quantità di informazione → sistemi lineari a più incognite
→ ricerca internet: matrici di ordine il numero di info trovate (milioni)
→ sperimentazione: ricerco di un approccio ottimale

MATEMATICA: analisi + algebra (attenzione alle MOLTI DI MATRICI
INFORMATICA: calcolatore

AMBIENTE DI CALCOLO: MATLAB → risoluzione di - SISTEMI LINEARI
- AUTOVALORI
- AUTOVETTORI

ESAME: scritto 2h + orale (teoremi CON dimostrazione)

1 costruire un modello dell'aritmetica in macchina → algoritmi adatti all'implementazione in macchina
2 come rappresentare i numeri → traslare la rappresentazione nel linguaggio macchina

RAPPRESENTAZIONE NUMERI E ARITMETICA MACCHINA

$134 = 1\cdot 10^{2} + 3\cdot 10^{1} + 4\cdot 10^{0}$ RAPPRESENTAZIONE DECIMALE (base 10)
cifre della rappresentazione: $0 \le a_i \le 9$

Considero una base $\beta \in \mathbb{N}$, $\beta > 1$ in quanto le cifre di rappresentazione vanno da 1 a $\beta - 1$
come si può rappresentare in modo univoco i numeri reali?
(i numeri complessi sono visti come $a+ib$, essa univocamente determinati come coppia di numeri reali)

TEOREMA DI RAPPRESENTAZIONE DEI NUMERI IN BASE β

Sia $\beta \in \mathbb{N}$, $\beta > 1$ una base di numerazione, sia $x \in \mathbb{R}$, $x \neq 0$
allora $\exists!$: un esponente $p \in \mathbb{Z}$
∴ una successione di cifre di $\{a_i\}_{i \in \mathbb{N}}$ $i \ge 1$
con $a_1 \neq 0$ e a_i non definitivamente uguali a $\beta - 1$
tali per cui

$$x = \mathrm{sgn}(x)\, \beta^{p} \underbrace{\sum_{i=1}^{+\infty} a_i \beta^{-i}}_{\text{mantissa}} \quad \text{dove } \mathrm{sgn}(x) = \begin{cases} +1 & x > 0 \\ -1 & x < 0 \end{cases}$$

RAPPRESENTAZIONE NORMALIZZATA IN VIRGOLA MOBILE
$a_1 \neq 0$ dipende dall'esponente

$(0.01)_{10} = 10^{-1}\cdot(0.01) = 10^{-1}\cdot(0.1)$

virgola mobile = utilizzando l'esponente cambio la posizione della virgola

$(0,111\ldots)_2$, $(0.9999\ldots)_{10}$ non sono ammesse, vediamo perché:

Voglio l'unicità della rappresentazione! per averla in ambito scolastico si porta il numero periodico in frazione.
SERIE NUMERICHE CON RAGIONE 1/10

$$0.\bar{9} = 0.9999\ldots = \sum_{i=1}^{+\infty} 9\cdot 10^{-i} = 9\sum_{i=1}^{+\infty} \cdot 10^{-i} = 9\sum_{i=1}^{+\infty} \left(\frac{1}{10}\right)^{i}$$

in particolare: $9\sum_{i=1}^{+\infty}\left(\frac{1}{10}\right) = 9\lim_{k\to+\infty}\sum_{i=1}^{k}\left(\frac{1}{10}\right)$ limite delle somme parziali

$\sum_{i=1}^{k} a^{i} = a + a^2 + a^3 + a^4 + \ldots + a^k$, osservo che:

$$(1-a)(a+a^2+\ldots+a^k) = (a+a^2+\ldots+a^k) - (a^2+a^3+\ldots+a^{k+1}) = a - a^{k+1}$$

$$\Rightarrow \sum_{i=1}^{k} a^{i} = \frac{a - a^{k+1}}{1-a}$$

$$\Rightarrow \lim_{k\to+\infty} \sum_{i=1}^{k} \left(\frac{1}{10}\right)^i = \lim_{k\to+\infty} \frac{\left(\frac{1}{10}\right) - \left(\frac{1}{10}\right)^{k+1}}{1-\frac{1}{10}} = \frac{\frac{1}{10}}{1-\frac{1}{10}} = \frac{1}{9}$$

$$\Rightarrow 0.\overline{9} = 9 \sum_{i=1}^{+\infty} \left(\frac{1}{10}\right)^i = 9 \cdot \frac{1}{9} = 1$$

⇒ per l'umidità non posso accettare tali rappresentazioni altrimenti troverei 2 rappresentazioni per lo stesso numero:

$10^0 \cdot (0.9)$ e $10^1 \cdot (0.1)$

Nel teorema viene escluso lo zero perchè non può avere una rappresentazione normalizzata (cioè si trova $a_1 \neq 0$)

Un numero $x \in \mathbb{R}$ è caratterizzato univocamente da:

$\pm$ (segno) $\quad p$ (esponente) $\quad \{a_i\}_{i\in\mathbb{N}, i\geq 1}$ (mantissa)

In macchina devo registrare numeri su registri di lunghezza finita (STACK finito di celle di memoria)

REGISTRO DI LUNGHEZZA FINITA ↔ non è possibile rappresentare tutti i numeri

numero caratterizzato da 3 parti ⇒ 3 sottoregistri

STANDARD IEEE 754 (Standard in doppia precisione)

per convenzione fu stabilito $\beta = 2$ e registro costituito da 64 bit (inizialmente erano 32 → singola precisione)

$\pm$	$\hat{p}$	$\{a_i\}_{i\in\mathbb{N}, i\geq 1}$	64 bits	$p = \hat{p} - 1022$
1 bit	11 bits	52 bits		

Esponente $p \in \mathbb{N}$ ⇒ 1 bit per il segno, 10 per le cifre

ATTENZIONE! DOPPIA RAPPRESENTAZIONE dello ZERO → si opera in traslazione mantenendo 11 bit (non si usa un bit per il segno)

Si considera tutti i naturali da 0.....0 a 1.....1 $= 1\cdot 2^0 + \dots + 1\cdot 2^{10} = 2^{11} - 1 = 2047$ in maniera standard.

Considero ora $\hat{p}$ = numero naturale registrato negli 11 bit, tra 0 e 2047.

La macchina è programmata per sapere che l'esponente p è pari a $\hat{p} - 1022$

⇒ l'esponente $p \in [-1022, 1025]$, ma si decide di tenere da parte le rappresentazioni $\hat{p} = \begin{cases} 0.....0 \\ 1.....1 \end{cases}$

⇒ $p \in [-1021, 1024]$

Dato che i numeri sono in forma normalizzata $d_1 = 1$ necessariamente ⇒ d_1 non viene rappresentato

$$\Rightarrow x = \pm 2^p \sum_{i=1}^{53} d_i 2^{-i} \quad \text{t.c.} \quad d_1 = 1, \; p \in [-1021, 1024]$$

NUMERI NORMALIZZATI DI MACCHINA

$\hat{p} = 0\dots0 \Rightarrow x = \pm 2^{-1022} \sum_{i=1}^{52} d_i 2^{-i}$ NON HO $d_1 = 1$ rappresento numeri non nec. normalizzati

→ vengono chiamati NUMERI DENORMALIZZATI

$\hat{p} = 1 \ldots\ldots 1$, in tal caso dipende dal campo cifre:

$\begin{cases} d_1 = \ldots = d_{52} = 0 & \pm\infty \\ \text{altrimenti} & \text{Not A Number (NaN) forme ind.} \end{cases}$

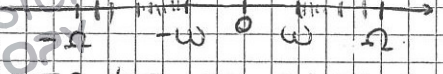
NUMERI NORMALIZZATI: $X = \pm 2^p \sum_{i=1}^{53} d_i 2^{-i}$ $d_1 = 1$, $p \in [-1021, 1024]$

I numeri sono simmetrici → insieme simmetrico ai numeri normalizzati (vale anche per i denormalizzati)

ω = numero più piccolo normalizzato positivo $= 2^{-1021} \cdot 2^{-1} = 2^{-1022}$

Ω = numero più grande normalizzato positivo $= 2^{1024} \sum_{i=1}^{53} 1 \cdot 2^{-i}$

⇒ i numeri di macchina normalizzati si organizzano nel seguente range, ossia sono compresi tra $-\Omega$ e $-\omega$ o ω e Ω



Consideriamo la distanza tra un numero di macchina normalizzato e il successivo:

$x = 2^p \sum_{i=1}^{53} d_i 2^{-i}$ $\quad y = 2^p \left(\sum_{i=1}^{53} d_i 2^{-i} + \underbrace{1 \cdot 2^{-53}}_{+1 \text{ unità all'ultima cifra}} \right)$

⇒ $dist(x,y) = 2^{p-53}$ → la distanza dipende da p:

⇒ i numeri di macchina non sono uniformemente distribuiti nell'intervallo ma sono più densi avvicinandosi a ω e si diradano andando verso Ω

Questo perché dal punto di vista del calcolo scientifico non importa la quantità ammessa di errore possibile, ma a quello RELATIVA (percentuale di errore).

NUMERI DENORMALIZZATI: $X = \pm 2^{-1022} \sum_{i=1}^{52} d_i 2^{-i}$, essi sono invece compresi tra $-\omega$ e ω, dove troviamo anche lo 0.

$\pm\infty$ e NaN servono a gestire le situazioni in cui il numero inserito non si trova nel range dei numeri normalizzati → ERRORE DI OVERFLOAT

L'insieme dei numeri normalizzati di macchina è rappresentato da:

$$F(\beta, t, u, M) = \{ x \in \mathbb{R} : x = \pm \beta^p \sum_{i=1}^{t} d_i \beta^{-i}, \; d_1 \neq 0, \; p \in [-u, M] \} \cup \{0\}$$

(t: cifre della mantissa)

Per lo standard IEEE 754 $p \in [-1021, 1024]$, $t = 53$ e lo zero vi viene introdotto con i numeri denormalizzati

La distanza tra numeri di macchina è in modo analogo β^{p-53}

Tale insieme è finito e vale:

$$\# F = 2 \underbrace{(u + M + 1)}_{\text{quanti modi da } p} \underbrace{(\beta - 1)}_{\text{varia } d_1} \underbrace{\beta^{t-1}}_{\text{le altre } t-1 \text{ cifre}} + 1$$

è simmetrico

a tale numero vanno aggiunti i denormalizzati (finiti)

Come viene approssimato $x \in \mathbb{R}$ ad un numero di macchina?
- Se è troppo grande è associato a $\pm\infty$
- Se è troppo piccolo è associato a un denormalizzato
- A cosa si associano i numeri normalizzati? ANALISI DEGLI ERRORI GENERATI IN MACCHINA → capire i buoni algoritmi in macchina

master COPY

COMMAND WINDOW → calcolatrice + ambiente di lavoro programmi
format long → cifre significative più esplicite (16, doppia precisione)

$a = [1, 2; 3, 4]$ → matrice $a = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ — inv(a) inversa, eig(a) autovaloni, det(a) determinante

[v, d] = eig(a) → matrice v di autovaloni, matrice d di autovettori (ossia a*v = v*d)

le assegnazioni hanno precedenza a sinistra : b = x*a.

y = a\b → data la matrice a si effettua l'inversa di a per b ($a^{-1}b$), ossia l'approssimazione del sistema lineare $ax = b$

ERRORI segnalati: errori aritmetici

determinante: non è usato (laplace) lo sviluppo ha un numero di operazioni troppo elevato (costo computazionale maggiore di n!)
→ Gauss/scalini

Matlab implementa IEEE 754: abbiamo visto che i numeri sono organizzati tra $-\Omega$ e $-\omega$ o ω e Ω.

① $\omega = 2^{-1022} \cdot 2^{-1} = 2^{-1022} = 2.225e^{-308} = 2.225 \cdot 10^{-308}$

help realmin = + piccolo numero normalizzato positivo

posso usare il numero 2^{-1073} ma è denormalizzato ma è comunque considerato un numero di macchina: i numeri denormalizzati possono anche usati impiegati fino a $2^{-1022} \cdot 2^{-52}$, ossia il più piccolo n° di macchina

se calcolo numeri più piccoli essi sono approssimati a 0

② $\Omega = 2^{+1024} \cdot \left(2^{-1} + 2^{-2} + \dots + 2^{-53}\right) = 2^{1024} \cdot \left(\frac{1}{2} + \dots + \left(\frac{1}{2}\right)^{53}\right) =$

$= 2^{+1024} \cdot \frac{2}{2^{53}} \left(\frac{1}{2} + \dots + \left(\frac{1}{2}\right)^{53}\right) = 2^{+1024} \cdot 1$

$= 2^{+1024} \cdot \frac{2^{53} - 1}{2^{53}} = 2^{+1024} \left(1 - 2^{-53}\right)$

se lo inserisco in macchina mi dà $+\infty$ in quanto inserendo 2^{1024} va in overflow, posso considerare $1024 - 53 = 971$ $\Rightarrow \Omega = 2^{971} \cdot (2^{53} - 1) = 1.798 \cdot 10^{308}$

realmax = $1.798 \cdot 10^{308}$

In macchina espressioni equivalenti possono portare a risultati diversi, a prescindere da come è strutturata l'espressione.
Essendo l'aritmetica finita e con errori il modo per organizzare i calcoli influisce sul risultato (sensibilità del calcolo)

ESEMPIO

$((x - 1) + 1)/x$, $x = 1 + 2^{-52} = 2^{1} \cdot (0.100\dots1)$ (1 in 52°)

risultato = 1

$((x + 1) - 1)/x$, $x = 2^{-53}$ → non normalizzato

risultato = 0 → effetto legato all'aritmetica di macchina e alle approssimazioni.

Osservo che facendo $(x-1)/x$ e $1-1/x$ ottengo due calcoli con le ultime 7 cifre decimali differenti (ordine della $10^{-8/9}$)

Quale quantità è più corretta per $x = 1 + 2^{-32}$?

$2^{-32} / x = (x-1)/x$ → tale espressione mi dà il corretto risultato, senza fare approssimazioni che conducono ad errori

per $x = 1 + 2^{-52}$ le due espressioni danno uguale risultato → aritmetica con precisione finita

Distanza tra due numeri consecutivi $= \beta^{p-t} = 2^{p-53}$

$x = 1 \rightarrow p = 1 \Rightarrow$ successivo numero : $1 + 2^{-52}$

eps → $2.220446 \cdot 10^{-16} = 2^{-52}$ → l'errore relativo che commettiamo nella rappresentazione dei numeri è maggiorato da tale numero

PRECISIONE DI MACCHINA : l'errore

help eps : distanza tra numeri floating point positiva ⇒ dà un bound per la precisione di macchina. : tra 1 e il numero successivo.

$2^{1024} = +\infty$, $-2^{1024} = -\infty$ → gestire le situazioni di OVERFLOW!

$1/2^{+1024} = 0$, $2^{1024}/2^{1024} = \text{NaN}$, $\log(0) = -\infty$

$2^{-1022} \cdot 2^{-52} = 4.94 \cdot 10^{-324}$ → cifre tutte 0, l'ultima = 1

$2^{-1022} \cdot 2^{-53} = 0$ sotto il più piccolo numero denormalizzato

$2^{-1022} \cdot (2^{-53} + 2^{-52}) = 9.88 \cdot 10^{-324}$ → numero più vicino denormalizzato: $2^{-1022} \cdot 2^{-51}$

