CALCOLO NUMERICO

Il calcolo numerico è uno strumento che ci permette di sviluppare algoritmi numerici per studiare problemi di tipo matematico.

Gli algoritmi devono essere adatti ad essere scritti su un calcolatore.

Devono quindi avere una data

PORTABILITÀ → devono essere efficienti, devono avere tempi di realizzazione buoni, poche operazioni da svolgere

ROBUSTI → non devono propagare notevolmente gli errori

ARITMETICA DI MACCHINA

Modello di alto livello che ci permette di valutare l'algoritmo migliore specificando quello che è il modello di funzionamento della macchina.

Creare un modello semplice della macchina ci permette di capire perché alcuni risultati sono più accurati degli altri meno.

ALGORITMI

- Algoritmi di algebra lineare (sistemi lineari e calcolo di autovalori e autovettori)
- Algoritmi di approssimazione (di funzioni e di dati, integrali definiti, equazioni non lineari)
- Problemi matematici che vengono poi da specialisti funzionali

NUMERO

È il primo dato per capire l'aritmetica di macchina.

La prima cosa da fare è cercare di definire una rappresentazione unica del numero per avere una rappresentazione in macchina

TEOREMA DI RAPPRESENTAZIONE IN BASE

$B \in \mathbb{N}$ $B \geq 2$ $B \rightarrow$ base (non è più contare in base 1)

Tra la rappresentazione razionale (parte frazionaria) e quella a frazione di ufo la rappresentazione dei numeri in frazione di ufo è di tipo più generale perché si rappresentano, ad esempio, non sono rappresentati come

rapporto. Andiamo quindi a vedere la rappresentazione per eccesso:

ENUNCIATO RAPPRESENTAZIONE NORMALIZZATA IN VIRGOLA MOBILE

Sia x un numero reale, $x \neq 0$ (numero complesso è visto come una coppia di reali) allora:

1) $\exists!\ p \in \mathbb{Z}$ detto esponente della rappresentazione

2) $\exists!\ \{d_i\}_{d_i \in \mathbb{N}}$, $0 \leq d_i \leq B-1$, $d_1 \neq 0$, d_i non definitivamente uguale a $B-1$}

t.c. $x = \mathrm{segno}(x)\, B^p \sum_{i=1}^{+\infty} d_i B^{-i}$ (1 = Mantissa)

Aritmetic floating point rappresentazione normalizzata in singola mobile.

PAROLA CHIAVE: una successione è definitivamente uguale a $B-1$ se $\forall n > m$ tutti gli elementi della successione dell'indice m in poi sono uguali a $B-1$.

Normalizzata perché la prima cifra della mantissa deve essere $\neq 0$.

Normalizzare serve per avere l'unicità della rappresentazione. Se no non era una rappresentazione normalizzata perché non riesco a normalizzare la prima cifra.

Mentre la successione dei d_i non deve essere definitivamente $= B-1$ in quanto altrimenti l'unicità: $9 \sum_{i=1}^{\infty} \frac{1}{10^i} = 9 \cdot \sum_{i=1}^{\infty} \left(\frac{1}{10}\right)^i$ $a = \frac{1}{10}$

Raccolgo $\frac{1}{10}$ per poter ricondurre la successione da 0

$$0,\overline{9} = \sum_{i=1}^{\infty} 9 \cdot 10^{-i} = 9 \cdot 10^{-1} + 9 \cdot 10^{-2} + \ldots = \frac{9}{10} \sum_{i=0}^{\infty} 10^{-i}$$

serie geometrica

$$= \frac{9}{10} \cdot \frac{1}{1 - \frac{1}{10}} = 1$$ nella l'unicità $0,\overline{9} = 1$

PAROLA CHIAVE: SERIE GEOMETRICA

$\sum_{i=0}^{+\infty} a^i$ $0 \leq |a| < 1$ (perché la serie converga)

$\lim_{k \to +\infty} \sum_{i=0}^{k} a^i$ limite delle ridotte parziali:

Pongo (a posteriori) che che:

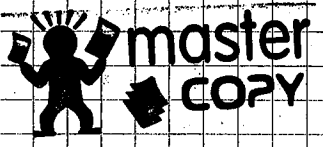
$(1+\dots+a^k)(1-a) = 1-a^{k+1}$ perché tutti i termini intermedi si annullano

$$1+\dots+a^k = \frac{1-a^{k+1}}{1-a}$$

Quindi:

$$\lim_{k\to+\infty} \sum_{i=0}^{+k} a^i = \lim_{k\to+\infty} \frac{1-a^{k+1}}{1-a} = \frac{1}{1-a}$$

($a^{k+1} \to 0$ perché $0 \le a < 1$)



REGISTRO

È dove andiamo ad allocare le varie informazioni. È il nostro secondo stato dell'architettura di macchina.

Può contenere in uno la rappresentazione in macchina.

Il registro è una sequenza finita di celle di memoria in cui può allocare dei numeri.

Se la mantissa è una sequenza infinita devo fare un approssimazione, questa è la prima parte di errore.

$$X \neq 0 \qquad X = (\pm)\, B^{(p)} \sum_{i=1}^{\infty} (d_i)\, B^{-i}$$

Ogni numero è caratterizzato da segno, esponente e sequenza di cifre.

Cella di memoria e spaziatura

$\pm$	exp	cifre della mantissa

Esempio:

$B=2$ lavoriamo in binario e il registro ha 32 bit

- 1 cella per il segno
- 8 celle per l'esponente
- 23 celle per le cifre della mantissa.

ZOOM ESPONENTE (8 celle)

$\pm$							

↓ segno esponente

+ 0 numeri per i numeri rimangono 7 celle

2^7 combinazioni

$\pm(1\cdot 2^0 + 1\cdot 2^1 + \dots + 1\cdot 2^6) = 2^7 - 1$ massimo numero rappresentabile

$p \in [-127, 127]$

ZOOM MANTISSA

$\sum d_i 2^{-i}$

$x = \pm 2^p \sum_{i=1}^{23} d_i 2^{-i}$ celle per la mantissa

Quanti sono i numeri che posso rappresentare all'interno dello spaccamonte.

INSIEME DEI NUMERI DI MACCHINA

$F(B, t, m, M) = \{ x = \pm B^p \sum_{i=1}^{t} d_i B^{-i},\ d_1 \neq 0;\ 0 \le d_i \le B-1 \} \cup \{0\}$

B: base mantissa; t: esponente nel range $-m, M$

$p \in [-m, M]$

$p \in \mathbb{Z} \cup \{0\}$

lo zero è un numero di macchina ma lo metto a parte a parte e unito.

Se un numero non è di macchina devo approssimarlo.

OPERARE IN TRASLAZIONE ($B=2$, 32 bit) E DENORMALI

Se lavoriamo in $B=2$ possiamo effettuare un'ulteriore ottimizzazione:

1) Tutto che lavoro in notazione normalizzata so già che $d_1 = 1$ (perché $d_1 \neq 0$ e sono in base 2) quindi posso avere un guadagno di 1 cella di memoria. DELL'ESPONENTE

2) La rappresentazione è ridondante perché ho 0 e ho rappresento due volte, +0 e -0 ho due rappresentazioni.

±	$0 \le p \le 255$	$d_1 \dots d_{23}$

8 celle

tutti 0 o tutti 1

Non posso =D so già per il segno dell'esponente

Approfitto un trucco per poter rappresentare anche gli esponenti negativi senza rappresentare due volte lo zero

Devo avere $q - 2^{t-1} \neq 0$ quadruplo un numero

- Se $1 \le p \le 254$ $\quad x = \pm 2^{p-126} \left(1 \cdot 2^{-1} + \sum_{i=1}^{23} d_{i+1} 2^{-(i+1)}\right)$

126 perché è la metà per avere metà esponenti positivi e metà negativi, però devi ricordare questo fatto che $M = 128$ ho guadagnato un esponente rispetto a prima $m = -125$

- Se $p = 0$ $\quad x = \pm 2^{-126} \left(\sum_{i=1}^{23} d_i 2^{-i}\right)$ Attenzione qua non vedendo $d_1 \neq 0$ è una rappresentazione denormalizzata

NORMALIZZATA-D

con la rappresentazione denormalizzata posto anche lo zero, questo se $d_i = 0 \ \forall i$

- Se $p = 255$ comprende anche il massimo numero rappresentabile

$p = 255$ → $+\infty$ se ho un + e le cifre della mantissa non sono tutte $= 0$

→ $-\infty$ se ho un − e le cifre della mantissa non sono tutte nulle

→ NAN not a number $d_1 = \dots = d_{23} = 0$ tutte vere

$\rightarrow 2^{\overbrace{255}^{\neq 0}} \cdot 0 = \text{NaN}$

DENORMALIZZATA →

Questa versione denormalizzata è stata introdotta per:

1) Per le divisioni per 0 ($+\infty, -\infty$)
2) NAN per le forme indeterminate $\frac{\pm\infty}{\pm\infty}$

INSIEME DEI NUMERI RAPPRESENTABILI IN MACCHINA NELLA FORMA NORMALIZZATA

L'insieme dei numeri esattamente rappresentabili in macchina è un insieme a cardinalità finita.

PIU' PICCOLO NUMERO DI MACCHINA POSITIVO

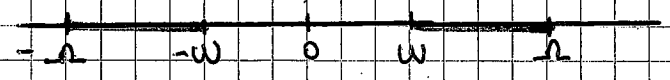
$$w = +\underbrace{B^{-m}}_{\substack{\text{B diverso da 0} \\ \text{questo numero} \\ \text{è diverso da 0.}}} \left(\underbrace{1 \cdot B^{-1}}_{\substack{\text{perché il primo} \\ \text{numero non può} \\ \text{essere = 0}}} + \underbrace{\underline{1}0)}_{\substack{\text{perché tutte le altre cifre} \\ \text{le prendo nulle} \\ \text{per avere il più} \\ \text{piccolo numero possibile}}}\right) = B^{-m-1}$$

PIU' GRANDE NUMERO DI MACCHINA

$$\Omega = +B^M \underbrace{(B-1)}_{\substack{\text{tutto} \\ \text{fuori dalla} \\ \text{sommatoria}}} \sum_{i=1}^{t} B^{-i}$$

Perché ho preso tutti i $d_i = B-1$ cifra massima? non è una coincidenza con il definitivamente?
NO! perché il definitivamente vale per una sequenza infinita qui la sommatoria arriva fino a t che è un numero finito, il numero di cifre della mantissa.

I numeri di macchina rimangono numerici rispetto allo 0



DISTANZA TRA UN NUMERO E IL SUCCESSIVO NUMERO DI MACCHINA

Prendiamo un numero di macchina positivo, tanto vale numerico: $x = B^p \sum_{i=1}^{t} d_i B^{-i} \rightarrow y = B^p \left(\sum_{i=1}^{t} d_i B^{-i} + \underline{1}B^{-t}\right) \Rightarrow |x-y| = B^{p-t}$