Capitolo 1
Introduzione

Queste note raccolgono le lezioni di calcolo numerico tenute dal docente a Pisa nei corsi di studio di laurea triennale di informatica e di ingegneria biomedica. L'insegnamento intende fornire un'introduzione al calcolo scientifico. Vengono illustrati metodi numerici di base per la risoluzione approssimata con il calcolatore di problemi matematici di interesse applicativo che per dimensioni o complessità non possono essere risolti con carta e penna. Tra questi citiamo il calcolo degli zeri di funzioni, la risoluzione di sistemi di equazioni lineari, il calcolo degli autovalori di matrici, l'integrazione definita e l'approssimazione di funzioni e di dati. La trattazione di ogni argomento ricerca un equilibrio tra il rigore matematico e le approssimazioni introdotte di volta in volta per definire metodi ed algoritmi di risoluzione numerica. L'enfasi è posta sull'analisi degli aspetti computazionali, quali il condizionamento dei problemi esaminati e la stabilità e la complessità dei metodi proposti, che assumono rilevanza una volta si proceda al calcolo approssimato mediante calcolatore. Le esercitazioni con l'ausilio dello strumento di calcolo MATLAB introducono lo studente all'analisi sperimentale degli algoritmi e alla validazione dei risultati.

ALGORITMI → PORTABILI ~> li rende efficienti
→ ESEGUIBILI
→ ROBUSTI ~> non sensibili ad errori

~> riguardano → ALGEBRA LINEARE
→ APPROSSIMAZIONE di FUNZIONI

costruzione di un modello di funzionamento della macchina, al fine di definire quale algoritmo sia il migliore. Il modello deve essere: di alto livello, semplice e flessibile

I° PUNTO: definizione di una RAPPRESENTAZIONE UNICA del NUMERO in macchina per svolgere operazioni aritmetiche, unico tipo di funzioni svolgibili dal calcolatore

II° PUNTO: definizione di un' ARITMETICA di MACCHINA, che mi permetta di capire le operazioni svolgibili dal calcolatore, gli errori e le approssimazioni che si generano

master COPY

ESEMPIO di rappresentazione normalizzata: può essere infinita

• 0,01 in $B=10$ $\rightsquigarrow$ $= +10^{p}(d_1 \cdot 10^{-1} + d_2 \cdot 10^{-2} + \dots) = +d_1 10^{p-1} + d_2 10^{p-2} + \dots$

$d_1 \neq 0 \Rightarrow$ la prima cifra $\neq 0$ è $d_2 \Rightarrow 1 \cdot 10^{-2} \Rightarrow p-1=-2 \Rightarrow p=-1$ allora $0{,}01 = 10^{-1}(1 \cdot 10^{-1})$

•

Capitolo 2

L'Aritmetica del Calcolatore

→ fornisce un metodo di RAPPRESENTAZIONE in BASE di un numero

Lezione 2.1: Rappresentazione in Base e Numeri di Macchina.

(binario)
↳ i calcolatori ragionano in base 2 $\rightsquigarrow$ le cifre sono 0,1 (solo numeri reali)

Sia $B \in \mathbb{N}$, $B > 1$. Il seguente teorema caratterizza la rappresentazione di un numero $x \in \mathbb{R}$, $x \neq 0$, in base B.

TEOREMA di RAPPRESENTAZIONE in BASE

Teorema 2.1.1. Dato $x \in \mathbb{R}$, $x \neq 0$, esistono e sono univocamente determinati

1. un intero $p \in \mathbb{Z}$ detto *esponente* della rappresentazione;
2. una successione di numeri naturali $\{d_i\}_{i \geq 1}$, con $d_1 \neq 0$, $0 \leq d_i \leq B-1$ e d_i non definitivamente uguali a $B-1$, dette *cifre* della rappresentazione; = MANTISSA

tali per cui si ha

$$x = \text{sign}(x) B^{p} \sum_{i=1}^{+\infty} d_i B^{-i} \tag{2.1}$$

La rappresentazione (2.1) di un numero reale x si dice rappresentazione *normalizzata* in *virgola mobile* (*floating point*) in quanto l'esponente p determinato in modo da avere la prima cifra dopo la virgola non nulla.

N.B.: "normalizzata" poiché la prima cifra $\neq 0$; "virgola mobile" perché l'esponente permette di spostare il punto ed eliminare eventuali 0 iniziali

Si osserva che

- Le condizioni $d_1 \neq 0$ (rappresentazione normalizzata) e d_i non definitivamente uguale a $B-1$ sono introdotte per garantire l'unicità della rappresentazione. A titolo di esempio in base $B=10$ si ha

$$1 = +10^{1}(1 \cdot 10^{-1}) = +10^{2}(0 \cdot 10^{-1} + 1 \cdot 10^{-2})$$

ma la seconda non risulta accettabile perchè ha la prima cifra nulla. Analogamente in base $B=10$ si ha

$$1 = +10^{1}(1 \cdot 10^{-1}) = 0.\overline{9} = +10^{0} \sum_{i=1}^{+\infty} 9 \cdot 10^{-i}$$

ma la seconda non risulta accettabile perchè ha le cifre tutte e quindi definitivamente uguali a $9 = 10-1$.

master COPY

- Il numero $x = 0$ non ammette rappresentazione normalizzata. In macchina verrà trattato e memorizzato in modo speciale.
- La rappresentazione floating point dei numeri reali si estende all'insieme dei numeri complessi $z = a + ib$ rappresentati come coppie di numeri reali.

Poichè i registri delle unità aritmetiche ed i dispositivi di memoria di un calcolatore consentono la memorizzazione di un numero finito di cifre binarie *l'insieme dei numeri di macchina*, cioè l'insieme dei numeri reali esattamente rappresentabili in macchina ha cardinalità finita. Risulta pertano essenziale determinare criteri e condizioni che consentano la rappresentazione quanto più accurata possibile dei numeri reali nel calcolatore.

Dal teorema di rappresentazione in base segue che la rappresentazione di un numero reale nel calcolatore può avvenire assegnando delle posizioni di memoria per il segno, per l'esponente e per le cifre della rappresentazione. In tal modo la rappresentazione di un numero in macchina assume la seguente struttura

Segno	Esponente	Cifre della rappresentazione

N.B. : l'esponente è un altro numero con il segno

Esempio 2.1.1. I personal computer che implementano lo standard IEEE 754-1985 prevedono la memorizzazione su registri lunghi 32 bit ripartiti come $1 + 8 + 23$ per la *singola precisione* e 64 bit ripartiti come $1 + 11 + 52$ per la *doppia precisione*.

Definizione 2.1.1. Si definisce *insieme dei numeri di macchina* in rappppresentazione floating point con t cifre, base B e range $(-m, M)$ l'insieme dei numeri reali

$$\mathbb{F}(B,t,m,M) = \{0\} \cup \{x \in \mathbb{R}\colon x = \text{sign}(x) B^p \textstyle\sum_{i=1}^{t} d_i B^{-i}, 0 \le d_i \le B-1, d_1 \neq 0, -m \le p \le M\}.$$

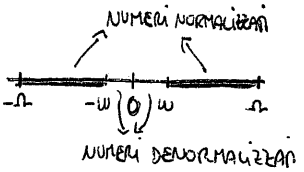
ha una rappresentazione a parte

Si osserva che

- L'insieme dei numeri di macchina $\mathbb{F}(B,t,m,M)$ ha cardinalità finita $N = 2B^{t-1}(B-1)(M+m+1)+1$ → insieme di numeri rappresentabili finito ⇝ i numeri andranno incontro ad approssimazioni
- Se $x \in \mathbb{F}(B,t,m,M)$ e $x \neq 0$ allora $\omega = B^{-m-1} \le |x| \le B^M(1-B^{-t}) = \Omega$. Ne segue che non è possibile rappresentare esattamente numeri non nulli di modulo minore ad ω. Per aggirare questa limitazione lo standard IEEE754 prevede anche una rappresentazione *denormalizzata*. Quando $p = -m$ la condizione $d_1 \neq 0$ può essere abbandonata e quindi vengono rappresentati numeri positivi e negativi compresi in modulo tra B^{-m-t} e $B^{-m}(B^{-1} - B^{-t})$. Analogamente se $p = M$ si introducono rappresentazioni speciali per i simboli $\pm\infty$ e NaN *not a number*.

ω = più piccolo numero di macchina positivo $\neq 0$
$= B^{-m}(1 \cdot B^{-1})$
↳ perchè $d_1 = 1$

Ω = più grande numero di macchina positivo $\neq 0$
$= B^M(1-B^{-t})$
↳ poichè si vuole considerare il più grande numero, le cifre considerate saranno $B-1$ ⇝ non ha più senso parlare di cifre def. $\neq B-1$ in quanto si considerano numeri finiti



- L'insieme dei numeri di macchina $\mathbb{F}(B,t,m,M)$ è simmetrico rispetto all'origine.
- Posto $x = \pm(.1)^s B^p \alpha \in \mathbb{F}(B,t,m,M)$ allora il successivo numero di macchina risulta essere $y = \pm(.1)^s B^p(\alpha + B^{-t})$ per cui in un sistema a virgola mobile la distanza $|y - x| = B^{p-t}$ varia con p ovvero con l'ordine di grandezza dei numeri considerati.

da $\sum_{k=0}^{K} a^n = \frac{a^K - 1}{a^1 - 1}$

DIM.: $\Omega = B^M(B-1)\sum_{i=1}^{t} B^{-i} = B^M(B-1)B^{-1}\sum_{i=0}^{t-1} B^{-i} = B^M(B-1)\frac{1}{B}\left(\frac{B^{-t}-1}{B^{-1}-1}\right) =$

$= B^M(B-1)\frac{1}{B}\frac{B^{-t}-1}{1/B-1} = B^M(B-1)\frac{1}{B}\frac{(-B^{-t}+1)B}{B-1} = B^M(1-B^{-t})$ OK

RAPPRESENTAZIONE sul CALCOLATORE

analizziamo questa organizzazione quanti numeri mi permette di rappresentare?

↳ p: poiché ha il segno, 1 bit sarà riservato al esso; rimangono allora 7 bit e posso quindi rappresentare:

$2^0 + 2^1 + 2^2 + \dots + 2^6 = 2^7 - 1 = 127$ valori

I numeri saranno scritti nella forma:

$$x = \pm 2^p \sum_{i=1}^{23} d_i 2^{-i} \qquad d_1 \neq 0,\ p \in [-127, 127],\ 0 \leq d_i \leq 1$$

In questo caso, l'insieme dei numeri macchina sarà del tipo:

$$F(2, 23, 127, 127) = \left\{ x \in \mathbb{R} \,/\, x = \pm 2^p \sum_{i=1}^{23} d_i 2^{-i},\ d_1 \neq 0,\ p \in [-127, 127],\ 0 \leq d_i \leq 1 \right\} \cup \{0\}$$

in realtà ciò non corrisponde alla vera rappresentazione, in quanto vi sono delle ridondanze:

- se rappresentiamo l'esponente con il segno, lo 0 sarà rappresentato due volte, cioè come ± 0 (DOPPIA RAPPRESENTAZIONE DELLO ZERO) ⇝ si parla sempre di 0 dell'esponente
- la quantità d_1 può anche non essere rappresentata in base 2, in quanto:
  $d_1 \neq 0$ ma poiché in base 2 $0 \leq d_1 \leq 1 \Rightarrow d_1 = 1$

Per eliminare queste ridondanze è stato definito lo standard IEEE 754 di organizzazione:

- L'esponente viene rappresentato senza segno, quindi solo con valori positivi. Il calcolatore sa però dov'è l'origine del sistema, ovvero sa distinguere i numeri negativi e i numeri positivi. Dato che utilizza 8 bit, l'esponente rappresenta 256 numeri che, escluse le rappresentazioni con tutti 0 e tutti 1, si riducono a 254.
  Questo modus operandi viene definito TECNICA di TRASLAZIONE e permette all'esponente di rappresentare gli interi nell'intervallo $p \in [-125, 128]$;
- d_1 non viene rappresentato, in quanto il calcolatore conosce già il suo valore, cioè $d_1 = 1$

↳ la rappresentazione sarà del tipo: $x = \pm 2^p \sum_{i=1}^{24} d_i 2^{-i}$, $p \in [-125, 128]$, $d_1 = 1$, $0 \leq d_i \leq 1$

N.B.: la mantissa ha 24 cifre in quanto d_1 si sa già il valore

Occorre ora considerare le rappresentazioni tenute da parte (dell'esponente):

- rappresentazione con tutti 1: a seconda del segno del numero x, il calcolatore darà delle rappresentazioni speciali, in quanto sarà un numero troppo grande:

  $+\infty$ $\quad -\infty$ $\quad$ NaN

- rappresentazione con tutti 0: si introducono numeri denormalizzati, dove viene eliminato il vincolo $d_1 \neq 0$, e servono per rappresentare numeri piccoli

$$x = \pm 2^{-126} \sum_{i=1}^{23} d_i 2^{-i}$$

master COPY

SEGNO 1 bit | ESPONENTE p 8 bit | CIFRE MANTISSA 23 bit

Esempio 2.1.2. Si consideri una rappresentazione su 32 bit in base 2 del tipo

$$\boxed{\pm}\ \boxed{a_1 \cdots a_8}\ \boxed{d_1 \cdots d_{23}}$$

In notazione normalizzata la prima cifra $d_0 = 1$ non viene rappresentata. Per l'esponente abbiamo 256 possibili valori che si riducono a 254 eliminando le configurazioni con tutti zeri e tutti 1. Mediante una tecnica di traslazione questi 254 valori sono utilizzati per rappresentare gli interi nell'intervallo $[-125, 128]$. Se $a_1 = \ldots = a_8 = 0$ allora il numero rappresentato è $x = \pm(0.0d_1d_2 \ldots d_{23})_2 \cdot 2^{-125}$ (rappresentazione denormalizzata con $d_0 = 0$). Se $a_1 = \ldots = a_8 = 1$ allora $x = \pm\infty$ se $d_1 = \ldots = d_{23} = 0$ e $x =$ NaN altrimenti. Per valori dell'esponente compresi tra 1 e 254 il numero rappresentato è $x = \pm(0.1d_1d_2 \ldots d_{23})_2 \cdot 2^{p-126}$ (rappresentazione normalizzata con $d_0 = 1$). Il più piccolo numero positivo normalizzato è $\omega = (1.00\ldots0)_2 \cdot 2^{-126} = 2^{-126}$. Il più grande numero normalizzato è $\Omega = (1.11\ldots1) \cdot 2^{127} = 2^{128}(1 - 2^{-24})$. Per il range si trova $m = 125$ e $M = 128$. Il numero di cifre è $t = 24$. Il primo numero di macchina più grande di $1 = (1.00\ldots0)_2 \cdot 2^0$ è $1 + 2^{-23}$.

Lezione 2.2: Aritmetica di Macchina.

→ cosa succede quando inserisco un numero in macchina?

Rappresentare un numero reale non nullo $x \in \mathbb{R}$, $x \neq 0$, in macchina significa *approssimare* x con un numero $\tilde{x} \in \mathbb{F}(B, t, m, M)$ commettendo un *errore relativo* di rappresentazione

$$\epsilon_x = \frac{\tilde{x} - x}{x} = \frac{\eta_x}{x}, \quad x \neq 0,$$

quanto più piccolo possibile in valore assoluto. La quantità

$$\eta_x = \tilde{x} - x$$

è detta *errore assoluto* di rappresentazione. In ambito ingegneristico e scientifico l'errore relativo è importante per la valutazione *qualitativa* del fenomeno considerato. Diversamente, un errore assoluto stimato in 1 cm assume differente significato se riferito alla misura di un tavolo o di una distanza astronomica. Il sistema floating point è costruito in modo da assicurare per il modulo dell'errore relativo di rappresentazione (e non per il modulo dell'errore assoluto) una maggiorazione indipendente dalla grandezza del numero rappresentato x.

numero reale

0 ω x̃ x Ω

→ x̃ = trn(x) ≠ x. Esso rappresenta sempre un numero più piccolo di x e può essere pensato come il più grande numero di macchina più piccolo di x, cioè il altri numeri di macchina tra x̃ e x

+∞ ⇑ x > Ω −∞ ⇑ x < −Ω

Dato $x \in \mathbb{R}$, $x \neq 0$, distinguiamo 2 casi

1. $|x| < \omega$ (*underflow*) o $|x| > \Omega$ (*overflow*);
2. $\omega \leq |x| \leq \Omega$. → si cercherà di rappresentarlo, approssimandolo

→ se |x| > Ω → numero troppo grande per essere rappr.

→ se |x| < ω → numero troppo piccolo per essere rappr.
↳ possibilità di associare 0 o numero denormalizzato

Nel secondo caso le tecniche di approssimazione previste dallo standard IEEE 754 sono:

1. *round to the nearest* (arrotondamento): il numero x viene approssimato con il numero rappresentabile $\tilde{x}$ più vicino; → in termini di precisione di macchina, la distanza tra x e x̃ e il suo arrotondamento posso maggiorare con la semiampiezza del suo intervallo, perché cadrà in una delle due metà
2. *round toward zero* (troncamento): il numero x viene approssimato con il più grande numero rappresentabile $\tilde{x}$ il cui valore assoluto risulti minore od uguale al valore assoluto di x;

$u = 2^{-24}$ in IEEE 754