LE RADICI SUL COMPUTER

come fanno i computer a calcolare la radice quadrata?

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Davide Palitta, Germana Landi
{davide.palitta, germana.landi}@unibo.it

Dipartimento di Matematica, Centro AM^2
Alma Mater Studiorum - Università di Bologna

Ancora approssimazioni?

Ieri abbiamo parlato di

- **quadrati perfetti**: la loro radice quadrata è un numero intero (soluzione esatta)
- **quadrati imperfetti**: la loro radice quadrata è un numero decimale, illimitato e non periodico (soluzione approssimata)

Spesso possiamo scrivere la radice quadrata di un quadrato imperfetto in *forma implicita*

$$\sqrt{\frac{25}{49}} = \frac{5}{7}$$

ma se vogliamo esplicitarne il valore su un foglio di carta dobbiamo per forza introdurre un'approssimazione

$$\sqrt{\frac{25}{49}} = \frac{5}{7} \approx 0.7142857$$

Ancora approssimazioni?

Questo tipo di approssimazioni sono necessarie in tantissimi ambiti

$$\pi = ?$$

Ancora approssimazioni?

Questo tipo di approssimazioni sono necessarie in tantissimi ambiti

$$\pi \approx 3.14$$

Ancora approssimazioni?

Questo tipo di approssimazioni sono necessarie in tantissimi ambiti

$$\pi \approx 3.14$$

$$g =? \text{ (accelerazione di gravità)}$$

Ancora approssimazioni?

Questo tipo di approssimazioni sono necessarie in tantissimi ambiti

$$\pi \approx 3.14$$

$$g \approx 9.81 \text{ (accelerazione di gravità)}$$

non abbiamo abbastanza fogli sull'intero pianeta per riportare tutte le infinite cifre decimali

e sul calcolatore?

Un problema molto simile si riscontra anche nell'utilizzo di un calcolatore:

nessun calcolatore al mondo ha abbastanza memoria per memorizzare tutte le infinite cifre decimali di π

e sul calcolatore?

Un problema molto simile si riscontra anche nell'utilizzo di un calcolatore:

nessun calcolatore al mondo ha abbastanza memoria per memorizzare tutte le infinite cifre decimali di π

- il calcolatore memorizza (e quindi usa!) solo delle approssimazioni dei numeri reali
- queste approssimazioni non possono essere evitate e devono essere tenute in grande considerazione dai matematici numerici

Numeri floating point

L'insieme dei numeri floating point è indicato con $\mathbb{F}$ e i suoi elementi sono spesso rappresentati in forma esponenziale

$$x = (-1)^s \cdot (0.a_1 a_2 \dots a_t) \cdot \beta^e, \quad a_1 \neq 0$$

- s vale 0 o 1 e quindi $(-1)^s$ stabilisce il **segno** di x
- β è la **base** di rappresentazione dei numeri ed è un numero naturale $\beta \geqslant 2$ ($\beta = 2, 8, 10, 16, \dots$)
- $a_1 a_2 \dots a_t$ è un intero detto **mantissa**
- a_1, $a_2, \dots, a_t$ sono le cifre della mantissa e sono numeri naturali compresi tra 0 e $\beta - 1$ con il vincolo che $a_1 \neq 0$
- t è il massimo numero di cifre della mantissa che possono essere memorizzate nella memoria del calcolatore
- e è un numero intero detto **esponente** che assume valori tra $L < 0$ e $U > 0$: $e \in [L, U]$

Numeri floating point

Un insieme $\mathbb{F}$ di numeri floating point è quindi caratterizzato dai seguenti parametri

- la base β
- il numero di cifre significative t della mantissa
- gli estremi $L < 0$ e $U > 0$ dell'intervallo in cui prende valori l'esponente e

$$\mathbb{F}(\beta, t, L, U)$$

è un insieme

- limitato, cioè non contiene numeri arbitariamente grandi/piccoli in valore assoluto
- finito, cioè ha un numero finito di elementi

Numeri floating point

Un insieme $\mathbb{F}$ di numeri floating point è quindi caratterizzato dai seguenti parametri

- la base β
- il numero di cifre significative t della mantissa
- gli estremi $L < 0$ e $U > 0$ dell'intervallo in cui prende valori l'esponente e

$$\mathbb{F}(\beta, t, L, U)$$

è un insieme

- limitato, cioè non contiene numeri arbitariamente grandi/piccoli in valore assoluto
- finito, cioè ha un numero finito di elementi

Uno degli insiemi di numeri floating point più comunemente usato è

$$\mathbb{F}(2, 53, -1021, 1024)$$

Perchè $\beta = 2$

$$\mathbb{F}(2, 53, -1021, 1024)$$

L'utilizzo della base 2 è naturale su un calcolatore perchè è possibile rappresentare i due simboli 0 e 1 rispettivamente con un circuito elettrico aperto o chiuso.

Esempio

Consideriamo un circuito elettrico che alimenta una lampadina: se l'interruttore è aperto non passa corrente e la lampadina è spenta; se è chiuso la lampadina è accesa. Associando alla lampadina spenta il valore 0 e alla lampadina accesa il valore 1, possiamo rappresentare i numeri per mezzo di sequenze di lampadine, spente o accese.

Spento  Acceso 

Numeri decimali	0	1	2	3	4	5	6	7	8	9
Serie di lampadine in stato di acceso/spento										
Numeri binari	0	1	10	11	100	101	110	111	1000	1001

Ed. Atlas

Perchè $\beta = 2$

L'utilizzo della base 2 riserva però qualche problemino...

Esempio
Il numero $t = 0.1$ non ha una rappresentazione esatta in base binaria e richiede infinite cifre:

$$0.1 = 1\cdot 2^{-4}+1\cdot 2^{-5}+0\cdot 2^{-6}+0\cdot 2^{-7}+1\cdot 2^{-8}+1\cdot 2^{-9}+0\cdot 2^{-10}+0\cdot 2^{-11}+1\cdot 2^{-12}\ldots$$

ossia

$$0.1 = 0.0001100110011001100110011001100110011001100110011001100110\ldots$$

Possibili conseguenze?

Un disastro "numerico"

Durante la guerra del Golfo (1991) una batteria di missili Patriot americana non riuscì ad intercettare uno Scud iracheno, a causa di un problema di precisione numerica. Come conseguenza, lo Scud uccise 28 americani. **Cosa è successo?**

- Il computer del sistema Patriot, per eseguire i calcoli, doveva moltiplicare per 1/10 il tempo registrato dall'orologio interno del sistema.
- Poichè il numero 1/10 in base 2 ha infinite cifre decimali, la rappresentazione di 1/10 era troncata alla 23-esima cifra introducento un errore di circa 0.000000095 in base 10.
- Gli errori di arrotondamento nella conversione del tempo causarono un errore nel calcolo della traiettoria.
- Difatti, il tempo di 100 ore calcolato in secondi diede il valore 359999.6567 invece di 360000, un errore di 0.3433 secondi che portò il Patriot 569 metri fuori della traiettoria del missile Scud!

Errori di arrotondamento

Dato $x \in \mathbb{R}$, si presentano due casi

1. $x \in \mathbb{F}(2, 53, -1021, 1024)$, quindi possiamo rappresentare x sul calcolatore senza bisogno di nessuna approssimazione
   ESEMPIO: tutti i numeri naturali possono essere rappresentati esattamente come numeri floating point

2. $x \notin \mathbb{F}(2, 53, -1021, 1024)$, quindi abbiamo bisogno di approssimare x per riuscire a rappresentarlo sul calcolatore. Il calcolatore genera $fl(x) \in \mathbb{F}(2, 53, -1021, 1024)$ tale che l'errore di arrotondamento è

$$\frac{|x - fl(x)|}{|x|} \leqslant \frac{1}{2}\varepsilon_M, \quad \varepsilon_M = 2^{-52} : \text{epsilon macchina}$$

   Per $\mathbb{F}(\beta, t, L, U)$ generico si ha $\varepsilon_M = \beta^{1-t}$

E quando faccio delle operazioni?

L'utilizzo dei numeri floating point ha un grosso impatto anche sulle operazioni aritmetiche che utilizziamo nei nostri algoritmi

Dati $x, y \in \mathbb{R}$ e una qualsiasi operazione aritmetica $\circ$, avremo, in generale,

- $fl(x) \neq x$, $fl(y) \neq y$
- $fl(fl(x) \circ fl(y)) \neq fl(x) \circ fl(y)$

e quindi

$$fl(fl(x) \circ fl(y)) \neq x \circ y$$

E quando faccio delle operazioni?

Esempio 1

- Prendendo spunto dal nostro Patriot, proviamo a sommare 10 volte 0.1
- Quale dovrebbe essere il risultato?

E quando faccio delle operazioni?

Esempio 1

- Prendendo spunto dal nostro Patriot, proviamo a sommare 10 volte 0.1
- Quale dovrebbe essere il risultato?
- Proviamo a verificare se

$$1 - (0.1 + 0.1 + \ldots + 0.1) = 0$$

E quando faccio delle operazioni?

Esempio 2

- Consideriamo l'espressione

$$y = \frac{(-1 + (1 + x))}{x}$$

- Quale dovrebbe essere il risultato?

E quando faccio delle operazioni?

Esempio 2

- Consideriamo l'espressione

$$y = \frac{(-1 + (1 + x))}{x}$$

- Quale dovrebbe essere il risultato?
- Verifichiamo se

$$y = \frac{(-1 + (1 + x))}{x} = 1$$

per diversi valori di x

E quando faccio delle operazioni?

Esempio 2

- Consideriamo l'espressione

$$y = \frac{(-1 + (1 + x))}{x}$$

- Quale dovrebbe essere il risultato?
- Verifichiamo se

$$y = \frac{(-1 + (1 + x))}{x} = 1$$

  per diversi valori di x
- per x piccolo abbiamo dei problemi!

E quando faccio delle operazioni?

Esempio 2

- Consideriamo l'espressione

$$y = \frac{(-1 + (1 + x))}{x}$$

- Quale dovrebbe essere il risultato?
- Verifichiamo se

$$y = \frac{(-1 + (1 + x))}{x} = 1$$

  per diversi valori di x
- per x piccolo abbiamo dei problemi!
- e se invece facessimo

$$y = \frac{(-1 + 1 + x)}{x}$$

Torniamo ad Erone

$$x = \sqrt{a}$$

Algoritmo: dato x_0 e fissato $M > 0$, l'iterazione del metodo di Erone è data da

$$x_{k+1} = \frac{1}{2}\left(x_k + \frac{a}{x_k}\right), \quad k = 0, \ldots, M$$

Come possiamo implementare questo algoritmo? cioè, come possiamo "tradurlo" in modo che sia eseguibile dal calcolatore?

Diagramma di flusso

Iniziamo con lo scrivere il diagramma di flusso del metodo di Erone

Simbolo	Blocco	Descrizione
	**Inizio/Fine**	Segnala l'inizio (o la fine dell'algoritmo).
	**Assegnazione**	Contiene un'istruzione con cui viene assegnato un determinato valore ad una variabile.
	**Test**	Presenta una condizione. A seconda che la condizione sia verificata o meno, l'algoritmo indica operazioni differenti da compiere.
	**Lettura**	Acquisisce i dati in *input*.
	**Scrittura**	Restituisce l'*output* indicato.

Figura: *Analisi Numerica*, Marco Rocco, Editrice La Scuola

Diagramma di flusso - Esempio 1

Consideriamo l'algoritmo per calcolare il massimo tra due numeri $x, y \in \mathbb{R}$

Diagramma di flusso - Esempio 1

Consideriamo l'algoritmo per calcolare il massimo tra due numeri $x, y \in \mathbb{R}$

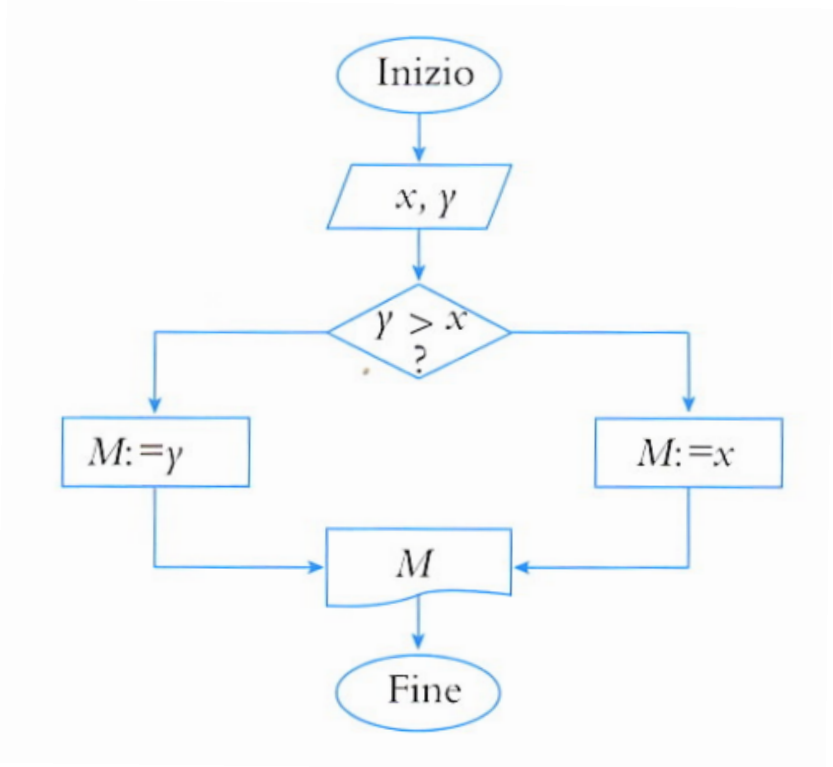

Figura: *Analisi Numerica*, Marco Rocco, Editrice La Scuola

Diagramma di flusso - Esempio 2

e se dovessi calcolare il massimo tra k numeri $a_1, a_2, \ldots, a_k \in \mathbb{R}$?

Diagramma di flusso - Metodo di Erone

Algoritmo: dato x_0 e fissato $M > 0$, l'iterazione del metodo di Erone è data da

$$x_{k+1} = \frac{1}{2}\left(x_k + \frac{a}{x_k}\right), \quad k = 0, \ldots, M$$

Siamo pronti per scrivere il diagramma di flusso del metodo di Erone!

Criterio di arresto

Algoritmo: dato x_0 e fissato $M > 0$, l'iterazione del metodo di Erone è data da

$$x_{k+1} = \frac{1}{2}\left(x_k + \frac{a}{x_k}\right), \quad k = 0, \ldots, M$$

Possiamo cambiare il criterio d'arresto per cercare di avere qualche informazione sulla bontà della approssimazione calcolata?

Proprietà del metodo di Erone

Algoritmo: dato x_0 e fissato $M > 0$, l'iterazione del metodo di Erone è data da

$$x_{k+1} = \frac{1}{2}\left(x_k + \frac{a}{x_k}\right), \quad k = 0, \ldots, M$$

Se $x_0 \geqslant \sqrt{a}$, il metodo di Erone gode delle seguenti proprietà

- tutti gli iterati x_k sono positivi
- $x_k > x_{k+1}$ per ogni $k \geqslant 0$
- $x_k \to \sqrt{a}$ per $k \to +\infty$
- $x_k - \sqrt{a} < \dfrac{x_0 - \sqrt{a}}{2^k}$
- $x_{k+1} - \sqrt{a} < x_k - x_{k+1}$

Giochiamo un po' col metodo di Erone

Divertiamoci col metodo di Erone!