Corso di Calcolo Numerico I e Laboratorio

Anno Accademico 2006-2007
C.d.L. in Informatica

Docente:	Dott.ssa Federica Tinti
e-mail:	tntfrc@unife.it
Pagina web del corso:	http://dm.unife.it/~tinti/Didattica/cn1.htm

Calcolo Numerico

Il Calcolo Numerico si occupa di progettare ed analizzare metodi numerici per la risoluzione di problemi del mondo reale, sfruttando al meglio le risorse di un sistema di calcolo.

L'uso di una strumento di calcolo presuppone limitazioni che riguardano:

TEMPO Analisi della Complessità Computazionale:
numero di operazioni necessarie

SPAZIO Memoria limitata:
analisi della quantità di memoria,
numeri finiti e analisi dell'errore

La maggior parte dei problemi può essere decomposta in sottoproblemi riconducibili all'insieme dei problemi fondamentali del Calcolo Scientifico.

I numeri e la rappresentazione posizionale

NUMERO

ENTITA' ASTRATTA	RAPPRESENTAZIONE
"sette"	(7, VII, 111_2, ...)
univocamente determinata	molteplice a seconda dei criteri di rappresentazione adottati.

- RAPPRESENTAZIONE POSIZIONALE

Se leggiamo la stringa 287, noi in realtà stiamo usando una convenzione per intendere le seguenti operazioni:

$$(287)_{10} = 2 \cdot 10^2 + 8 \cdot 10^1 + 7 \cdot 10^0$$

Abbiamo cioè una base $\beta = 10$ che determina il peso della posizione occupata e un insieme di $\beta - 1$ simboli: dati dalle cifre arabe $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$.

Con lo stesso principio possiamo ricavare altre rappresentazioni scegliendo una diversa base:

$$\begin{aligned}(101110100)_2 &= 1 \cdot 2^8 + 0 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + \\ &+ 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0\end{aligned}$$

base = 2, simboli 0,1

$$(10133)_4 = 1 \cdot 4^4 + 0 \cdot 4^3 + 1 \cdot 4^2 + 3 \cdot 4^1 + 3 \cdot 4^0$$

base = 4, simboli 0,1,2,3

$$(11F)_{16} = 1 \cdot 16^3 + 1 \cdot 16^2 + 15 \cdot 16^0$$

base = 16, simboli 0,1,2,...,9,A,B,C,D,E,F

In generale, in un sistema di numerazione in base $\beta > 1$, ogni naturale N si rappresenta come:

$$N = (d_n d_{n-1} ... d_0)_\beta = \text{ord}(d_n)\beta^n + \text{ord}(d_{n-1})\beta^{n-1} + ... + \text{ord}(d_1)\beta^1 + \text{ord}(d_0)\beta^0$$

ove ord(d_i) è il valore dell'i–esimo naturale; se si usano come simboli le cifre arabiche e se $\beta \leq 10$, ord$(d_i) = d_i$.

$$(d_n d_{n-1} ... d_0)_\beta$$

è la forma sintetica di N in base β.

SCELTA DELLA BASE

- numero dei simboli
- lunghezza delle stringhe
- complessità dell'aritmetica

BASE piccola $\longrightarrow$ pochi simboli
stringhe lunghe
aritmetica semplice

OPERAZIONI ARITMETICHE NELLE DIFFERENTI BASI

Valgono le stesse regole e proprietà formali dell'aritmetica decimale, ma si devono usare tavole diverse da quelle pitagoriche per addizione e moltiplicazione. La somma può dare un riporto 1; il prodotto può dare un riporto compreso tra 1 e $\beta - 2$. Base 2

+	0	1
0	0	1
1	1	10

·	0	1
0	0	0
1	0	1

Base 8

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

·	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	10	12	14	16
3	0	3	6	11	14	17	22	25
4	0	4	10	14	20	24	30	34
5	0	5	12	17	24	31	36	43
6	0	6	14	22	30	36	44	52
7	0	7	16	25	34	43	52	61

CASO BINARIO

L'aritmetica binaria è particolarmente semplice.

Somma: $(25)_{10} + (19)_{10} = (44)_{10}$.

		1	1	0	0	1	$+$
		1	0	0	1	1	$=$
Riporto	1			1	1		
	1	0	1	1	0	0	

Differenza: $(24)_{10} - (13)_{10} = (11)_{10}$.

1	1	0	0	0	-
0	1	1	0	1	$=$
0	1	0	1	1	

Prodotto: $(13)_{10} \cdot (14)_{10} = (182)_{10}$.

$$
\begin{array}{cccccccccc}
 & & & & 1 & 1 & 0 & 1 & \cdot \\
 & & & & 1 & 1 & 1 & 0 & = \\
\hline
 & & & & 0 & 0 & 0 & 0 & \\
 & & & 1 & 1 & 0 & 1 & & \\
 & & 1 & 1 & 0 & 1 & & & \\
 & 1 & 1 & 0 & 1 & & & & \\
\hline
1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 &
\end{array}
$$

Quoziente: $(28)_{10} : (9)_{10} = (3)_{10}$ con resto 1.

$$
\begin{array}{ccccc|cccc}
1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
\cline{6-9}
1 & 0 & 0 & 1 & & 1 & 1 & & \\
\cline{1-4}
0 & 1 & 0 & 1 & 0 & & & & \\
 & 1 & 0 & 0 & 1 & & & & \\
\cline{1-5}
 & 0 & 0 & 0 & 1 & & & &
\end{array}
$$

Quoziente e prodotto sono riportati a differenze o somme e traslazioni di numeri. La scelta della base 2 comporta la manipolazione di lunghe stringhe di numeri ma la complessità dell'aritmetica è bassa. Le operazioni possono essere realizzate con semplici circuiti elettronici.

ESEMPIO

La somma di due cifre con riporto fornisce il risultato e il successivo riporto. Il numero delle possibili combinazioni degli impulsi in entrata è basso.

c_1	c_0	riporto	s	riporto
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Teorema di rappresentazione dei numeri reali

Teorema. Sia $\alpha \in \mathbb{R}, \alpha \neq 0$; fissato un intero $\beta > 1$, α si rappresenta in modo unico come:

$$\begin{aligned} \alpha &= \text{segno}(\alpha)(a_1\beta^{-1} + a_2\beta^{-2} + a_3\beta^{-3} + \ldots)\beta^p \\ &= \text{segno}(\alpha) \sum_{i=1}^{\infty} (a_i\beta^{-i})\beta^p \\ &= \text{segno}(\alpha)\ m\beta^p \end{aligned}$$

ove segno$(\alpha) = \pm 1$ (a seconda che $\alpha >, < 0$), $0 \leq a_i \leq \beta - 1$, con a_i interi e $a_1 \neq 0$ e p è un intero; può esistere un indice k tale che $a_i = 0, k \leq i$ (rappresentazione degli interi o dei razionali finiti), ma non esiste un indice k tale che $a_i = \beta - 1, k \leq i$.

- Il numero reale 0 si rappresenta con 0.
- Poichè $\beta > 1$, la serie $\sum_{i=1}^{\infty}(a_i\beta^{-i})$ è convergente.
- m si dice **mantissa** e vale che $\frac{1}{\beta} \leq m < 1$.
- β^p si dice **parte esponente**; p si dice **esponente o caratteristica**.
- $\cdot$ si dice **punto radice**, $+$ o $-$ si dice **segno** del numero e può essere omesso se il numero è positivo.

Un numero reale $\alpha \neq 0$ si esprime in notazione posizionale in base $\beta > 1$ in **forma scientifica** nel seguente modo:

$$\alpha = \pm .a_1 a_2 \ldots \beta^p;$$

si dice normalizzata se $a_1 \neq 0$. In base 2 , in forma normalizzata, $a_1 = 1$.

ESEMPI.

$.372\ 10^3$	normalizzata
$.0372\ 10^4$	scientifica
$.3141592\ 10^1$	normalizzata
$.3243F\ldots\ 16^1$	normalizzata

Algoritmi di conversione di base

Conversione di un intero positivo α da base 10 a base $\beta > 1$.
METODO DELLE DIVISIONI SUCCESSIVE:

$$\begin{aligned}
\alpha &= (a_m a_{m-1} ... a_0)_\beta \\
&= a_m \beta^m + a_{m-1}\beta^{m-1} + ... + a_1\beta + a_0 = \\
&= (a_m\beta^{m-1} + a_{m-1}\beta^{m-2} + ... + a_1)\beta + a_0 = \gamma_1\beta + a_0
\end{aligned}$$

a_0, ossia la cifra meno significativa della rappresentazione cercata, è il resto della divisione intera di α per β.

$$\begin{aligned}
\gamma_1 &= (a_m\beta^{m-2} + a_{m-1}\beta^{m-3} + ... + a_3\beta + a_2)\beta + a_1 = \gamma_2\beta + a_1 \\
\gamma_2 &= (a_m\beta^{m-3} + a_{m-1}\beta^{m-4} + ... + a_4\beta + a_3)\beta + a_2 = \gamma_3\beta + a_2 \\
&\dots \\
\gamma_{m-1} &= a_m\beta + a_{m-1} = \gamma_m\beta + a_{m-1} \\
\gamma_m &= 0\beta + a_m
\end{aligned}$$

$m+1 =$ numero delle divisioni eseguite fino ad avere un quoziente 0.

Dato l'insieme di simboli $(d_0, d_1, \ldots d_{\beta-1})$

$q \leftarrow \alpha;$

$s \leftarrow$ (' ');

while $q \neq 0$

- $r \leftarrow$ resto(q/β);
- $q \leftarrow$ parte intera(q/β);
- $s \leftarrow$ concatena(d_r, s);

stampa s;

ESEMPIO. $(1972)_{10}$

$$
\begin{array}{lrl}
\text{base } 2 & 1972 : 2 = 986 & \text{resto } 0 \\
 & 986 : 2 = 493 & \text{resto } 0 \\
 & 493 : 2 = 246 & \text{resto } 1 \\
 & 246 : 2 = 123 & \text{resto } 0 \\
 & 123 : 2 = 61 & \text{resto } 1 \\
 & 61 : 2 = 30 & \text{resto } 1 \\
 & 30 : 2 = 15 & \text{resto } 0 \\
 & 15 : 2 = 7 & \text{resto } 1 \\
 & 7 : 2 = 3 & \text{resto } 1 \\
 & 3 : 2 = 1 & \text{resto } 1 \\
 & 1 : 2 = 0 & \text{resto } 1
\end{array}
$$

$$(1972)_{10} = (11110110100)_2$$

base 8	$1972 : 8 = 264$	resto 4
	$246 : 8 = 30$	resto 6
	$30 : 8 = 3$	resto 6
	$3 : 8 = 0$	resto 3

$$(1972)_{10} = (3664)_8$$

base 16	$1972 : 16 = 123$	resto 4
	$123 : 16 = 7$	resto $11 = B$
	$7 : 16 = 0$	resto 7

$$(1972)_{10} = (7B4)_{16}$$

Conversione di un reale positivo $\alpha < 1$ da base 10 a base $\beta > 1$.

METODO DELLE MOLTIPLICAZIONI SUCCESSIVE:

$$\begin{aligned} \alpha &= (.a_1 a_2 a_3 ...)_\beta = \\ &= a_1\beta^{-1} + a_2\beta^{-2} + a_3\beta^{-3}... \end{aligned}$$

- $\alpha\beta = a_1 + a_2\beta^{-1} + a_3\beta^{-2} + a_4\beta^{-3}... = a_1 + n_1$
  a_1 è la parte intera del risultato e n_1 la parte frazionaria.

$$n_1 = a_2\beta^{-1} + a_3\beta^{-2} + ...$$

- $n_1\beta = a_2 + a_3\beta^{-1} + a_4\beta{-2} + ... = a_2 + n_2$
- $n_2\beta = a_3 + a_4\beta^{-1} + a_5\beta^{-2} + ... = a_3 + n_3$
- .... Quando ci fermiamo?

Osservazione: Se la rappresentazione in base 10 di un numero finita, non è detto che la rappresentazione dello stesso numero in un'altra base β sia finita.
Si può dimostrare che: un numero $\alpha > 0$ ha rappresentazione finita in base $\beta \Leftrightarrow$ esistono interi positivi m, n tali che $\alpha = \frac{m}{\beta^n}$.
Altrimenti il numero nella nuova base ha rappresentazione periodica.

$$\Downarrow$$

Ci fermiamo quando la parte frazionaria diventa nulla oppure quando abbiamo raggiunto un numero di cifre sufficienti.

Dato l'insieme di simboli $(d_0, d_1, ...d_{\beta-1})$
e il numero massimo di cifre N_{max}

$p \leftarrow \alpha;$

$s \leftarrow$ '0.'

$i \leftarrow 0;$

while $s \neq 0$ and $i < N_{max}$

- $r \leftarrow$ parte intera$(s * \beta);$
- $p \leftarrow p * \beta -$ parte intera$(p * \beta);$
- $s \leftarrow$ concatena(s, d_r)
- $i \leftarrow i + 1;$

stampa $(s);$

ESEMPIO. $(0.1)_{10}$

base 2	$0.1 \times 2 = 0.2$	p. intera 0
	$0.2 \times 2 = 0.4$	p. intera 0
	$0.4 \times 2 = 0.8$	p. intera 0
	$0.8 \times 2 = 1.6$	p. intera 1
	$0.6 \times 2 = 1.2$	p. intera 1
	$0.2 \times 2 = 0.4$	p. intera 0
	$\vdots$	

$$(0.1)_{10} = (0.000\overline{1100})_2$$

base 5	$0.1 \times 5 = 0.5$	p. intera 0
	$0.5 \times 5 = 2.5$	p. intera 2
	$0.5 \times 5 = 2.5$	p. intera 2
	$\vdots$	

$$(0.1)_{10} = (0.0\overline{2})_5$$

base 7	$0.1 \times 7 = 0.7$	p. intera 0
	$0.7 \times 7 = 4.9$	p. intera 4
	$0.9 \times 7 = 6.3$	p. intera 6
	$0.3 \times 7 = 2.1$	p. intera 2
	$0.1 \times 7 = 0.7$	p. intera 0
	$\cdots$	

$$(0.1)_{10} = (0.\overline{0462})_7$$

Algoritmo di conversione di un reale α da base 10 a base $\beta > 1$.

1. Determinare $|\alpha|$, ricordando il segno.
2. Determinare $[|\alpha|]$ e eseguire la conversione con l'algoritmo delle divisioni successive.
3. Determinare $|\alpha| - [|\alpha|]$ e eseguire la conversione con l'algoritmo delle moltiplicazioni successive.
4. Scrivere il segno, la conversione della parte intera, il punto radice, la conversione della parte frazionaria.

ESEMPIO. $\alpha = (-25.375)_{10}$. Convertire in base 2.

1. $|\alpha| = 25.375$; segno='-'.
2. $[|\alpha|] = 25$; $(25)_{10} = (11001)_2$.
3. $|\alpha| - [|\alpha|] = .375$; $(.375)_{10} = (.011)_2$.
4. $\alpha = (-11001.011)_2$.

Conversione di un reale da base $\beta > 1$ a base 10.

$$\alpha = \pm(a_1 a_2 \ldots a_p . a_{p+1} a_{p+2} \ldots a_q)_\beta$$

Sfruttando la rappresentazione posizionale abbiamo che

$$\alpha = \pm(a_1\beta^{p-1} + a_2\beta^{p-2} + \ldots + a_p\beta^0 + a_{p+1}\beta^{-1} + a_{p+2}\beta^{-2} + \ldots + a_q\beta^{p-q})$$

$$\Downarrow$$

Valutazione di due polinomi :
$f(x) = a_1x^{p-1} + a_2x^{p-2} + \ldots + a_p$ in $x = \beta$,
$g(x) = a_qx^{-p+q} + a_{q-1}x^{-p+q-1} + \ldots + a_{p+1}x$ in $x = 1/\beta$.

$$(\alpha)_{10} = \pm(f(\beta) + g(1/\beta))$$

.

$$\Downarrow$$

Occorre un algoritmo conveniente per fare il calcolo di un polinomio a coefficienti reali in corrispondenza di un certo valore.

Conversione di un reale α da base β_1 a base β_2.

$$\beta_1 \quad \longrightarrow \quad 10 \quad \longrightarrow \quad \beta_2$$

$\uparrow$ rappresentazione posizionale

$\uparrow$ divisioni e moltiplicazioni successive

ESEMPIO

$\alpha = (1221)_7$, $\beta_2 = 2$.

$$\alpha = 1.7^3 + 2.7^2 + 2.7^1 + 1.7^0 \longrightarrow \alpha = (456)_{10} \longrightarrow \alpha = (111001000)_2$$

Valutazione di un polinomio reale in $x = \alpha$.

$$p_n(x) = a_0 x^n + a_1 x^{n-1} + \ldots + a_{n-1} x + a_n$$

$a_i \in \mathbb{R}, a_0 \neq 0$

ALGORITMO 1.

$p \leftarrow 1;$
$s \leftarrow a_n;$
for $\quad i = n-1, n-2, \ldots 0$
$\quad p \leftarrow p * \alpha;$
$\quad s \leftarrow p * a_i + s;$
stampa $s;$

La COMPLESSITA' COMPUTAZIONALE dell'algoritmo (ossia il numero totale di operazioni aritmetiche che devono essere fatte) è $2n$ moltiplicazioni e n addizioni.

ALGORITMO 2. Si basa sulla seguente riscrittura del polinomio:

$$p_n(x) = (\ldots(((a_0x + a_1)x + a_2)x + a_3)x + \ldots)x + a_n$$

Esempio:

$$\begin{aligned} p(x) &= 4x^3 + 5x^2 + x + 2 \\ &= ((4x + 5)x + 1)x + 2 \end{aligned}$$

```
s ← a_0;
for   i = 1, ..., n
      | s ← s * α + a_i;
stampa s;
```

La COMPLESSITA' COMPUTAZIONALE è pari a n moltiplicazioni e n addizioni.
L'algoritmo prende il nome di SCHEMA di RUFFINI-HORNER.