Idee su errore
e precisione arbitraria

github.com/nrizzo

12 giugno 2018

1 Introduzione

Nello studio del calcolo della variante di Ackermann su insiemi ereditariamente finiti (e non) definita come

$$\mathbb{R}_A(x) = \sum_{y \in x} 2^{-\mathbb{R}_A(y)},$$

sorgono questioni di analisi numerica, tra le quali come modellare l'incertezza su numeri reali e come rappresentare numeri razionali di precisione arbitraria con il calcolatore.

2 Considerazioni su errore e precisione

Definizione 1. Dati $x, \tilde{x} > 0$, diciamo che $\tilde{x}$ è un numero razionale che approssima (in modo forte) x fino a $k \in \mathbb{N}$ cifre binarie dopo la virgola se e solo se

$$x = \tilde{x} + e_x, \qquad \text{con} \quad 0 \leq e_x < 2^{-k} \quad \text{e} \quad \tilde{x} = \frac{j}{2^k},\ j \in \mathbb{N}$$

cioè è un'approssimazione per difetto.

Tuttavia, questo modello risulta poco maneggevole nel calcolare i risultati di operazioni aritmetiche tra dati approssimati. La prossima definizione è più generale.

Definizione 2. Dati $x, \tilde{x}, e_x^{\max} > 0$ con $x_{\min}$, $x_{\max}$ razionali tali che $x_{\min} \leq x_{\max}$, diciamo che $\tilde{x}$ approssima (in modo debole) x se e solo se

$$x = \tilde{x} + e_x, \qquad \text{con} \quad 0 \leq e_x < e_x^{\max}.$$

Però nel calcolo dell'approssimazione di $2^{-\tilde{x}}$ deve necessariamente venire introdotto errore *negativo* (indipendentemente dall'algoritmo utilizzato, un errore negativo compare nel resto di Lagrange o nel polinomio di MacLaurin), quindi la prossima rappresentazione alternativa può essere più comoda nella stima dell'errore.

Definizione 3. Alternativamente alla definizione 2, si può dire che x è approssimato dall'intervallo $[x_{\min}, x_{\max})$ se e solo se $x \in [x_{\min}, x_{\max})$, chiamando $\tilde{x} = x_{\min}$, $\tilde{x} + e_x^{\max} = x_{\max}$.

Semplici operazioni tra questi intervalli, come ad esempio la differenza, escludono gli estremi inclusi. Forse sarebbe utile fornire definizioni alternative, in base all'inclusione o meno degli estremi.

La definizione di intervalli "nei paraggi" di x semplifica il calcolo e lo studio dell'errore, poiché calcolando operazioni aritmetiche sugli estremi si lavorerà su dati esatti.

Teorema 1. Dato $x \in [x_{\min}, x_{\max}]$, il prefisso in comune delle rappresentazioni binaria dei due estremi (se esiste) è anche prefisso di x.

Teorema 2. Dato $x \in (b, c)$ e dato $(a, d) \supseteq (b, c)$ (cioè tali che $a \leq b \leq c \leq d$), il prefisso in comune tra b e d (se esiste) è anche prefisso di a, d e di x.

3 La propagazione dell'errore assoluto

3.1 Somma

$$(x_{\min}, x_{\max}) + (y_{\min}, y_{\max}) = (x_{\min} + y_{\min}, x_{\max} + y_{\max})$$
$$(x_{\min}, x_{\max}) + a = (x_{\min} + a, x_{\max} + a), \quad a \in \mathbb{R}$$

3.2 Immagine di funzione monotona

$$f\big((x_{\min}, x_{\max})\big) = \begin{cases} \big(f(x_{\min}), f(x_{\max})\big) & \text{se } f \text{ è non decrescente} \\ \big(f(x_{\max}), f(x_{\min})\big) & \text{se } f \text{ è non crescente} \end{cases}$$

3.3 Differenza

$$(x_{\min}, x_{\max}) - (y_{\min}, y_{\max}) = (x_{\min}, x_{\max}) + (-y_{\max}, -y_{\min}) =$$
$$= (x_{\min} - y_{\max}, x_{\max} - y_{\min})$$

3.4 Moltiplicazione

$$(x_{\min}, x_{\max}) \cdot (y_{\min}, y_{\max}) = (x_{\min} \cdot y_{\min}, x_{\max} \cdot y_{\max})$$
$$(x_{\min}, x_{\max}) \cdot b = (b \cdot x_{\min}, b \cdot x_{\max}), \quad b > 0$$

4 Approssimazione della potenza di due con esponente negativo

In questa sezione studiamo l'approssimazione, da parte di un calcolatore, del calcolo della potenza di due con esponente reale negativo, avendo un dato incerto rappresentato da un intervallo. Come numero di macchina supponiamo di avere a disposizione numeri frazionari binari a precisione arbitraria.

Inoltre assumiamo $(x_{\min}, x_{\max}) = (\lfloor x \rfloor + \{x_{\min}\}, \lfloor x \rfloor + \{x_{\max}\})$, cioè i due estremi sono compresi in $(\lfloor x \rfloor, \lfloor x \rfloor + 1)$.

4.1 Approssimare $2^{-\tilde{x}}$

Il valore da calcolare è

$$2^{-(x_{\min}, x_{\max})},$$

che secondo le regole appena presentate è uguale a

$$\left(2^{-x_{\max}}, 2^{-x_{\min}}\right) = \left(2^{-\lfloor x \rfloor} 2^{-x_{\max}}, 2^{-\lfloor x \rfloor} 2^{-x_{\min}}\right).$$

Supponendo che il numero di macchina sia rappresentato in base due, il prodotto per $2^{-\lfloor x \rfloor}$ corrisponde ad un semplice `rshift`; mentre per il secondo fattore il cambio a base e è di fatto obbligatorio, poiché nel polinomio di MacLaurin per 2^{-x} compaiono potenze di $x \log 2$.

Definito $y = x \log 2$,

$$2^{-x} = e^{\log(2^{-x})} = e^{-x \log 2} = e^{-y},$$

quindi si possono calcolare $y_{\min}$ e $y_{\max}$ approssimando $x_{\min} \cdot \log 2$ per difetto e $x_{\max} \cdot \log 2$ per eccesso, con scarto $-\delta_{\min}$ e $\delta_{\max}$, per cui vale

$$\left(2^{-\lfloor x \rfloor} e^{-y_{\max}}, 2^{-\lfloor x \rfloor} e^{-y_{\min}}\right) \supseteq \left(2^{-\lfloor x \rfloor} 2^{-x_{\max}}, 2^{-\lfloor x \rfloor} 2^{-x_{\min}}\right).$$

L'elevamento a potenza e^{-y} viene approssimato dal polinomio di MacLaurin g di cui vanno implementate due versioni, che approssimino una per eccesso e una per difetto: $\tilde{g}_{\text{ecc}}$ e $\tilde{g}_{\text{dif}}$. Il calcolo finale è di

$$\left(2^{-\lfloor x \rfloor} \tilde{g}_{\text{dif}}(y_{\max}), 2^{-\lfloor x \rfloor} \tilde{g}_{\text{ecc}}(y_{\min})\right) \supseteq \left(2^{-\lfloor x \rfloor} e^{-y_{\max}}, 2^{-\lfloor x \rfloor} e^{-y_{\min}}\right)$$

4.2 Polinomio di MacLaurin

L'algoritmo che calcola g potrebbe essere il seguente, in pseudocodice, ricalcando l'algoritmo di Horner per il calcolo di un polinomio.

```
0  reciprocal_exp_aux(y,n)
1  {
2          res = 0;
3          for (i = n; i > 0; i--) {
4                  if (i%2 == 0)
5                          res += +1;
6                  else
7                          res += -1;
8
9                  res *= y;
10                 res /= i;
11         }
12         res += 1;
13
14         return res;
15 }
```

Computazione parziale "migliorabile"? Partire dal fondo del polinomio di Maclaurin permette di risparmiare operazioni e rendere il calcolo della serie lineare rispetto al grado del polinomio, a patto che ogni approssimazione sia per eccesso o per difetto (in base alla versione di $\tilde{g}$ calcolata). Si potrebbe mantenere (circa) lo stesso numero di operazioni semplici partendo anche dai primi addendi, potendo rendere il calcolo "migliorabile in seguito", salvandosi l'ultimo addendo calcolato, ma bisognerebbe maneggiare due copie: l'approssimazione per difetto e l'approssimazione per eccesso, perché gli addendi hanno segno alternato.

4.3 MacLaurin: quanto e dove approssimare?

4.3.1 Operazioni

Le operazioni in cui si restituisce un risultato approssimato sono:

1. moltiplicazione (si può);
2. divisione per intero (spesso si deve);

ed entrambe possono essere troncate o arrotondate ad una precisione arbitraria, per essere approssimate rispettivamente per difetto o per eccesso.

4.3.2 MacLaurin

Esempio del calcolo di $\tilde{g}_{\text{dif}}(y_{\max})$: perché sia un'approssimazione per difetto, il resto di Lagrange deve essere **positivo**, cioè il grado del polinomio di MacLaurin deve essere dispari.

Approssimazione (difetto)	Approssimato e resto	Prossima operazione
-1	$= -1$	$\times y_{\max}$
$a(-y_{\max})$	$= -y_{\max}$	$/n$
$a\Big(-\frac{y_{\max}}{n}\Big)$	$= -\frac{y_{\max}}{n} - \alpha_n$	$+1$
$a\Big(-\frac{y_{\max}}{n} + 1\Big)$	$= -\frac{y_{\max}}{n} + 1 - \alpha_n$	$\times y_{\max}$
$a\Big(-\frac{y_{\max}^2}{n} + y_{\max}\Big)$	$= -\frac{y_{\max}^2}{n} + y_{\max} - \alpha_n \cdot y_{\max} - \beta_n$	$/(n-1)$
$a\Big(-\frac{y_{\max}^2}{n(n-1)} + \frac{y_{\max}}{n-1}\Big)$	$= -\frac{y_{\max}^2}{n(n-1)} + \frac{y_{\max}}{n-1} - \frac{\alpha_n}{n-1} y_{\max} - \frac{\beta_n}{n-1}$	-1
	$\vdots$	
$a\Big(\sum_{i=0}^{n}(-1)^i \frac{(y_{\max})^i}{(i)!}\Big)$	$= \sum_{i=0}^{n}(-1)^i \frac{(y_{\max})^i}{(i)!} - \sum_{i=2}^{n} \frac{(y_{\max})^{i-1}}{(i-1)!}\alpha_i - \sum_{i=2}^{n} \frac{(y_{\max})^{i-2}}{(i-1)!}\beta_i$	–

Dove a rappresenta una funzione che approssima per difetto e α_i e β_i sono gli errori introdotti nell'approssimazione di, rispettivamente, l'$n-i$-esima moltiplicazione e della divisione per i. Maggiorando questi errori con α e β, e notando

che le rispettive sommatorie sono maggiorate dal polinomio di MacLaurin per $y = 2^x$,

$$a\left(\sum_{i=0}^{n}(-1)^i\frac{(y_{\max})^i}{(i)!}\right) = \tilde{g}_{\text{dif}}(y_{\max}) \geq T_n(y_{\max}) - \alpha \cdot e^{y_{\max}} - \beta \cdot e^{y_{\max}},$$

e poiché $y_{\max} \in (0, \log 2)$[1],

$$\tilde{g}_{\text{dif}}(y_{\max}) \geq T_n(y_{\max}) - 2(\alpha + \beta).$$

Quest'ultima disequazione mostra che il polinomio di MacLaurin si può approssimare con precisione arbitraria, a seconda della precisione delle operazioni di moltiplicazione e divisione per intero.

5 Conclusione

In conclusione, i calcoli importanti da considerare su intervalli (dati affetti da errore) sono la somma e l'elevamento a potenza di due con esponente cambiato di segno, per cui dovrebbe valere

$$\begin{aligned}\left(2^{-\lfloor x \rfloor}2^{-x_{\max}}, 2^{-\lfloor x \rfloor}2^{-x_{\min}}\right) \subseteq \left(2^{-\lfloor x \rfloor}\tilde{g}_{\text{dif}}(y_{\max}), 2^{-\lfloor x \rfloor}\tilde{g}_{\text{ecc}}(y_{\min})\right) \subseteq \\ \subseteq \left(2^{-\lfloor x \rfloor} \cdot \left(\frac{2^{-x_{\max}}}{e^{\delta_{\max}}} - \frac{(\log 2 \cdot x_{\max})^{2n+1}}{(2n+1)!} - 2(\alpha+\beta)\right),\right. \\ \left. 2^{-\lfloor x \rfloor} \cdot \left(2^{-x_{\min}} \cdot e^{\delta_{\min}} + \frac{(\log 2 \cdot x_{\max})^{2m}}{(2m)!} + 2(\alpha+\beta)\right)\right),\end{aligned}$$

con $\delta_{\max}$, $\delta_{\min}$, n, m, α, β arbitrari.

[1] **Da rivedere**, $y_{\max}$ è un'approssimazione per eccesso...