Informatica Medica a.a. 2012-2013

1 Esercitazione n^o 2: Errori

1.1 Errori di arrotondamento

Il calcolatore opera su numeri rappresentati per mezzo di una sequenza *finita* di cifre: pertanto numeri reali come π o $\sqrt{2}$ non possono essere trattati nelle elaborazioni numeriche se non commettendo un errore. Lo stesso problema si ha anche con numeri razionali: a causa della rappresentazione in base 2 del calcolatore, un numero come 0.1 viene effettivamente elaborato usando un valore che non è 0.1. Questo problema suggerisce di utilizzare *variabili intere* come *contatori*.

1.1.1 Propagazione degli errori

Tali piccoli errori si propagano durante le usuali operazioni aritmetiche diventando in alcuni casi macroscopici. Si ricorda che per le operazioni macchina non valgono le note proprietà delle quattro operazioni aritmetiche.

Esercizio proposto Scrivere un programma che sommi il valore `float` 0.1 un numero di volte pari a 100000 e visualizzi la somma finale.

Provare ad utilizzare il valore `float` 0.5. Confrontare i due output.

1.1.2 Eccezioni

La rappresentazione dei numeri per mezzo di una sequenza finita di cifre pone il problema legato alla rappresentazione di numeri molto piccoli o molto grandi. A causa di questa limitazione durante l'elaborazione possono verificarsi particolari condizioni: per esempio, se il risultato di una elaborazione ha un valore troppo elevato si genera una condizione di *overflow*, se il valore è troppo piccolo si genera una condizione di *underflow*. Esistono altre eccezioni legate a particolari condizioni di calcolo: per esempio, l'indeterminazione (e.g. 0/0) che viene segnalata con un `nan` (not a number) e la divisione per zero che viene segnalata con un `inf`.

1.1.3 Precisione di macchina

Si ricorda che la *precisione di macchina* è la massima precisione di calcolo raggiungibile su un dato calcolatore ed è una misura relativa.

$$\frac{|x - \bar{x}|}{|x|} \leq B^{1-t}$$

Sviluppiamo un algoritmo per individuare tale quantità su un calcolatore. La precisione macchina ϵ_m può essere definita nel modo seguente:

$$\epsilon_m = min\{\epsilon \in F,\ \epsilon > 0 :\ 1 \oplus \epsilon > 1\}$$

dove F è l'insieme di tutti i numeri floating-point di macchina. I passi dell'algoritmo sono

1. $\epsilon = 1$
2. $\epsilon = \epsilon/2$
3. $temp = 1 + \epsilon$

4. se $temp > 1$ torna a 2

5. altrimenti $\epsilon_m = 2\epsilon$

Esercizio proposto Scrivere un programma che implementi l'algoritmo esposto, trovare la precisione macchina dei numeri `float` e dei numeri `double`. Che differenza c'è tra la precisione macchina e il più piccolo numero rappresentabile?

1.2 Errori di troncamento

Gli errori di troncamento derivano dall'uso di un procedimento di approssimazione in sostituzione di operazioni matematiche esatte. Per comprendere le caratteristiche di tali errori, prendiamo in esame una formulazione matematica molto spesso utilizzata nel calcolo numerico per esprimere una funzione in maniera approssimata: la serie di Taylor.

1.2.1 Serie di Taylor

Si ricorda che la serie di Taylor è descritta dalla seguente espressione

$$f(x+h) = f(x) + f'(x)h + \frac{f''(x)}{2!}h^2 + \frac{f'''(x)}{3!}h^3 + \cdots + \frac{f^{(n)}(x)}{n!}h^n + R_n$$

h indica la distanza da x a cui vogliamo valutare la funzione e il resto R_n è il valore dell'errore di troncamento che si ha se la serie viene troncata all'n-esimo ordine.

Esercizio proposto Data la seguente serie

$$e^x = 1 + x + \frac{1}{2!}x^2 + \frac{1}{3!}x^3 + \frac{1}{4!}x^4 + \frac{1}{5!}x^5 + \frac{1}{6!}x^6 + R_n$$

scrivere un programma che calcoli l'errore relativo e l'errore relativo approssimato di $e^{0.5}$ al crescere dell'ordine della serie: cioè considerando la serie dall'ordine 0 all'ordine 6.

```
ord.  0:  val=1.000000  er=39.346931%  era=100.000000%
ord.  1:  val=1.500000  er=9.020398%   era=33.333332%
ord.  2:  ...           ...            ...
```