*Scritti da Gabriel: gli appunti seguono l’ordine di tutte le lezioni.*

*Per eventuali completamenti anche di procedimenti di calcolo, si segua il file riassunto già presente su MEGA, da cui sono estrapolate le immagini presenti.*

*02/03/2022: Lezione 1 e 2: Numeri e floating point*

Parliamo dei numeri con segno. In particolare per un numero si intende una parte con segno, una parte intera ed una parte frazionaria, solitamente realizzata con una codifica binaria.

Le cifre e gli indici corrispondono alle potenze, positive e negative.

Noi ci focalizzeremo sulla base *b=10,* nonostante siano rimaste altre basi come la base 60 (sessagesimale).

All’interno del calcolatore si usa invece la base 2, corrispondente ai 2 possibili stati assunti da una macchina.

In merito ad un numero sappiamo essere composto da una parte intera e una serie frazionaria, definita come serie convergente. Ciò è visibile come:

Immagine che contiene testo, orologio

Descrizione generata automaticamente

La serie della parte frazionaria (che sta tra 0 ed 1) è convergente e ciò si dimostra per confronto per serie a termini non negativi, da cui viene maggiorata.

Genericamente tutto si riduce alla serie geometrica di ragione .

In generale la successione delle somme parziali con diverge con a > 1, mentre per a < 1 la serie della parte frazionaria è maggiorata da una serie geometrica convergente e quindi converge.

Per vedere che la parte frazionaria sta effettivamente tra 0 ed 1, ciò viene visto in base 10 si considera come esempio 0.99 periodico in base 10. Usando la serie geometrica e raccogliendo la cifra massima si dimostra che si ottiene proprio 1.

Alcune osservazioni:

* *i numeri irrazionali hanno parte frazionaria infinita* in qualsiasi base, dato che i numeri con parte frazionaria finita in una base sono necessariamente razionali, perché somma della parte intera che è un numero naturale e di una combinazione lineare
* i numeri irrazionali possono avere una base finita od infinita a seconda della base, per esempio:

= (0,100)3

Ma in base 10 diventa: (0,333…)10 = 1/3

Fondamentale nel calcolo numerico è il concetto di *errore*, fondamentale negli algoritmi e in particolare il *troncamento* dei numeri, tagliando n cifre nella parte frazionaria ma mantenendo la parte intera.

Introduciamo quindi due concetti chiave: l’approssimazione e l’errore stesso.

In merito al troncamento abbiamo:

Immagine che contiene testo, orologio

Descrizione generata automaticamente

La quantità errore è il modulo della distanza tra due reali, a ed a tilda, (quindi una differenza, vista come errore assoluto). Quindi:

Immagine che contiene orologio, calibro

Descrizione generata automaticamente

Vediamo di stimarla ed approssimarla correttamente.

Basta calcolare il resto della serie geometrica per avere una stima (usando i conti di dell’effettivo errore di troncamento generabile dai calcoli. L’errore di troncamento di n cifre dopo la virgola non supera , dove il *b* varia a seconda delle cifre utilizzate.

Attenzione: l’errore è stato solo stimato, che è una situazione tipica del calcolo numerico, dove gli errori non sono noti a priori, naturalmente.

Quante cifre devo prendere dopo la virgola per non superare una certa soglia di tolleranza lo determina la risoluzione di una disuguaglianza logaritmica, che porta come risultato

Il fatto di avere una stima permette di controllare l’errore, che dovrà necessariamente essere ≤ ε, che sarebbe la soglia di tolleranza.

La tecnica generalmente più usata è *l’arrotondamento*, che corrisponde alla serie infinita, considerando l’arrotondamento per difetto (quindi prima cifra va da 0 a 4 nella base 10) oppure per eccesso (da 5 a 9 base 10). Noi ci limitiamo a vedere basi pari. L’esempio è:

Immagine che contiene testo

Descrizione generata automaticamente

Il massimo errore nel caso di basi pari è metà del massimo errore di troncamento ad n cifre, quindi:

Immagine che contiene orologio, antenna, calibro

Descrizione generata automaticamente

Ad esempio nel caso del numero 0,44:

* nel caso del troncamento tutti i numeri dell’intervallo [0.44 e 0.449) tendono a 0.45
* nel caso dell’arrotondamento approssima tutti i numeri dell’intervallo (0.435,0.445) seguendo le regole di prima dell’arrotondamento per eccesso/per difetto.

In generale è meglio l’arrotondamento perché considera un intervallo più ampio e in generale è effettivamente migliore.

*Lezione 2: Floating point*

Un numero viene scritto come numero con segno, mantissa (vista come serie da 1 ad inf delle cifre) moltiplicato per una potenza della base.

L’esponente può essere positivo, nullo o negativo. Si adotta la convenzione che la prima cifra dopo la virgola *d1* sia diversa da 0 in maniera tale da non avere infinite rappresentazioni dello stesso numero.

Questa è la classica scrittura *floating point*, utile ad introdurre il concetto di *precisione macchina* ma anche di definizione dell’insieme dei *reali macchina*.

Per esempio:

diventa oppure:

Semplicemente si sposta la virgola con un opportuno spostamento della base.

La mantissa sta in 0 ed 1 (con 0 escluso), avendo in particolare la mantissa compressa tra 0.1 ad 1 (entrambi inclusi).

In generale un numero reale ha mantissa infinita, perché comprendiamo anche gli irrazionali; i numeri con mantissa finita sono invece i razionali (un esempio può essere 1/3 di prima con base 3 e base 10)

A questo punto si è in grado di definire *l’insieme dei reali-macchina* che, in un sistema di calcolo, lavora un insieme di numeri con una quantità finita di cifre di mantissa e un esponente che varia in un intervallo finito di numeri. Il fatto che il numero di cifre di mantissa sia finito e l’intervallo di esponenti sia finito permette la memorizzazione utilizzando sequenze di bit in base 2.

Per avere un’idea (poi faremo un modellino) con variabili (zone di memoria) a 64 bit per i reali-macchina (come accade ad es. in Matlab) si ha b= 2, t= 53 e in un modello semplificato L=−1023,U= + 1023. In una parte successiva andremo a studiare più in dettaglio la struttura dell’insieme dei reali-macchina.

Ora vogliamo stimare l’errore in cui l’approssimazione avviene per arrotondamento della mantissa al numero di cifre disponibili. Quindi:

Immagine che contiene testo

Descrizione generata automaticamente

Come detto, l’errore massimo sarà , da cui sappiamo che non supera ,

Subito si nota che l’errore dipende da *p*, cioè dall’ordine di grandezza del numero.

Numeri grandi in modulo avranno errori grandi, numeri piccoli in modulo avranno errori piccoli.

Allontandosi oppure avvicinandosi agli estremi, è accettabile un errore variabile con l’ordine di grandezza del numero? Questo succede se ci spostiamo dall’*errore assoluto*, quindi dati due numeri *a* e  *̃a* in modulo, *all’errore relativo*, definito invece come il rapporto dell’errore assoluto per il modulo di a. In generale l’errore relativo è il più importante in campo sperimentale e viene infatti espresso in percentuale.

Quindi:

Immagine che contiene testo

Descrizione generata automaticamente

Per questo motivo andremo ad eseguire una stima da sotto, con

Di fatto viene poi eseguito un calcolo in cui, usando la tecnica di arrotondamento, mettiamo a frazione la stima da sopra e da sotto, si calcola il massimo errore relativo di arrotondamento possibile, cosiddetta precisione di macchina, che sarebbe .

Esso dipende solo da *b* e da *t.*

Un esempio del concetto del floating point sono overflow ed underflow, arrotondando a seconda della situazione a t cifre di mantissa. Nell’intervallo di rappresentazione ogni “tacchetta” nella rappresentazione è un intorno di precisione e, unendo tutti questi intorni, ogni numero se non è numero macchina viene arrotondato per difetto.

*07/03/2022: Lezione 3*