Lezioni di Calcolo Numerico
Lezione 02: Aritmetica, errori

Alberto Tibaldi

5 maggio 2018

Indice

1 Operazioni di macchina **1**

2 Cancellazione numerica **3**
2.1 Esempio e definizione di cancellazione numerica 4
2.1.1 Esempio 1 5
2.1.2 Esempio 2 5
2.1.3 Un esempio di cancellazione numerica 5
2.2 Idee per risolvere la cancellazione numerica 6
2.2.1 Esempio 1: razionalizzazione -se possibile- 6
2.2.2 Esempio 2: espansione di Taylor 6
2.2.3 Esempio 3: relazioni trigonometriche -quando disponibili- 7

3 Condizionamento di un problema numerico **8**
3.1 Definizione di problema numerico; forma implicita, forma esplicita 8
3.2 Definizione di condizionamento di un problema numerico 9
3.3 Esempio: cancellazione numerica 9

4 Stabilità di un algoritmo **10**

1 Operazioni di macchina

Nella precedente lezione abbiamo appreso cosa accade ogni volta che arrotondiamo un numero reale, al fine di associargli un numero macchina che sia trattabile dall'aritmetica del nostro calcolatore. Tuttavia, questa operazione è solo il preludio: il motivo per cui ci interessa memorizzare questi numeri è poter sfruttare il calcolatore per eseguire le operazioni aritmetiche al nostro posto! Di conseguenza, nonostante abbiamo la garanzia che l'arrotondamento *di per sé* non possa influenzare più di tanto l'accuratezza dei nostri calcoli, dobbiamo ancora capire cosa succede quando eseguiamo delle operazioni aritmetiche a partire dai numeri di macchina ottenuti dall'arrotondamento.

In particolare, una prima domanda che potremmo farci è:

> «Eseguire un'operazione aritmetica su due numeri di macchina, ci dà la garanzia che il risultato sia un numero di macchina?»

La risposta a questa domanda, purtroppo, è negativa. Consideriamo, per esempio, $N = 10$, $t = 4$, $\overline{a}_1 = 0.5823$, $\overline{a}_2 = 0.6214$. Il fatto di aver indicato i numeri con una barretta ($\overline{a}$) è legato al fatto che questi sono già numeri arrotondati, e che infatti sono numeri di macchina per l'aritmetica specificata da N e t. La somma di questi numeri è:

$$\begin{array}{r} {\scriptstyle 1\ 1} \\ 0.5\,8\,2\,3 \\ +\ 0.6\,2\,1\,4 \\ \hline 1.2\,0\,3\,7. \end{array}$$

Il numero 1.2037, nella notazione floating-point normalizzata, è scritto come 0.12037×10^1 ma, quindi, la sua mantissa non sarebbe memorizzabile nella nostra aritmetica (richiederebbe 5 cifre, ma noi abbiamo $t = 4$). Questo semplicissimo esempio, quindi, permette di dimostrare che **il risultato di un'operazione aritmetica tra due numeri di macchina generalmente non fornisce un numero di macchina**. Questo fatto ci costringe ad aggiungere una nuova definizione: quella di **operazione di macchina**.

Un'operazione di macchina $\odot$ associa a due numeri di macchina un terzo numero di macchina, il quale è ottenuto arrotondando il risultato dell'operazione esatta eseguita sui numeri di macchina di partenza. A partire quindi da due numeri $\overline{a}_1$ e $\overline{a}_2$, una generica operazione di macchina $\odot$ si definisce come:

$$\overline{a}_1 \odot \overline{a}_2 = \overline{\overline{a}_1 \cdot \overline{a}_2} = (\overline{a}_1 \cdot \overline{a}_2)(1 + \varepsilon_\odot).$$

Per esempio, per la somma, si avrebbe

$$\overline{a}_1 \oplus \overline{a}_2 = \overline{\overline{a}_1 + \overline{a}_2} = (\overline{a}_1 + \overline{a}_2)(1 + \varepsilon_\oplus).$$

Queste ultime espressioni si possono leggere come:

> «L'operazione macchina $\odot$ eseguita tra i numeri di macchina $\overline{a}_1$ e $\overline{a}_2$ è uguale all'arrotondamento del risultato di $\overline{a}_1 \cdot \overline{a}_2$. Di conseguenza, l'errore $\varepsilon_\odot$ introdotto dall'arrotondamento del risultato è, proprio come in passato, minore della precisione di macchina ε_m.»

Riprendendo l'esempio di prima, una volta ottenuto $\overline{a}_1 + \overline{a}_2 = 0.12037 \times 10^1$, il risultato di $\overline{a}_1 \oplus \overline{a}_2$ sarebbe quindi 0.1204×10^1, ovvero l'arrotondamento alla quarta cifra.

Una particolarità delle operazioni di macchina, rispetto alle operazioni in aritmetica esatta, è l'assenza di alcune delle *proprietà di base*. In particolare, valgono le proprietà commutative della somma e del prodotto

$$\begin{aligned} \overline{a}_1 \oplus \overline{a}_2 &= \overline{a}_2 \oplus \overline{a}_1 \\ \overline{a}_1 \otimes \overline{a}_2 &= \overline{a}_2 \otimes \overline{a}_1, \end{aligned}$$

ma **non** valgono varie altre proprietà, tra cui le proprietà associative

$$\begin{aligned} \overline{a}_1 \oplus (\overline{a}_2 \oplus \overline{a}_3) &\neq (\overline{a}_1 \oplus \overline{a}_2) \oplus \overline{a}_3 \\ \overline{a}_1 \otimes (\overline{a}_2 \otimes \overline{a}_3) &\neq (\overline{a}_1 \otimes \overline{a}_2) \otimes \overline{a}_3, \end{aligned}$$

e le proprietà distributive

$$\begin{aligned} \overline{a}_1 \otimes (\overline{a}_2 \oplus \overline{a}_3) &\neq (\overline{a}_1 \otimes \overline{a}_2) \oplus (\overline{a}_1 \otimes \overline{a}_3) \\ (\overline{a}_1 \otimes \overline{a}_2) \oslash \overline{a}_3 &\neq (\overline{a}_1 \oslash \overline{a}_3) \otimes \overline{a}_2. \end{aligned}$$

Inoltre, un'altra relazione molto strana per chi è abituato a pensare in aritmetica esatta è

$$\overline{a}_1 \oplus \overline{a}_2 = \overline{a}_1,$$

che, detto brutalmente, significa che *se un termine è molto più piccolo dell'altro, allora non conta nell'operazione macchina*. Cosa vuol dire *molto* più piccolo? In questo caso, dire che $|\overline{a}_2| \ll |\overline{a}_1|$ si può tradurre come $|\overline{a}_2| < \varepsilon_m |\overline{a}_1|$. Per esempio, si potrebbe provare a introdurre i seguenti comandi sulla command window di MATLAB®:

```
>> format long e
>> 5+1.0e-18
ans =
5
>> 2017.3+1.0e-14
ans =
2.017300000000000e+03
```

Sommando questi due numeri, si ha che 10^{-14} è trascurabile a 2017.3, proprio perché il primo è più piccolo del secondo moltiplicato per ε_{m}.

Più in generale, quando si opera con un'aritmetica di macchina, è possibile che due espressioni siano **equivalenti**, per quanto invece non lo sarebbero in aritmetica esatta. In particolare, due quantità $\overline{a}_1$ e $\overline{a}_2$ si dicono equivalenti quando

$$e_{\mathrm{r}} = \frac{|\overline{a}_1 - \overline{a}_2|}{\overline{a}_1} \quad \text{oppure} \quad e_{\mathrm{r}} = \frac{|\overline{a}_1 - \overline{a}_2|}{\overline{a}_2}$$

è dello stesso ordine di grandezza (o possibilmente minore), della precisione di macchina ε_{m}. Per dare un po' di *feeling numerico* in più sull'argomento, consideriamo per esempio $N = 10$, $t = 5$, e $\overline{a}_1 = 0.99999$, $\overline{a}_2 = 1$. Se calcoliamo

$$e_{\mathrm{r}} = \frac{|0.99999 - 1|}{1} = 0.00001,$$

vediamo che $0.00001 = 0.1 \times 10^{-4}$ è minore di ε_{m}, che invece sarebbe:

$$\varepsilon_{\mathrm{m}} = \frac{1}{2} \times 10^{1-5} = 0.5 \times 10^{-4},$$

e quindi, per l'aritmetica considerata in questo esempio, i numeri 0.99999 e 1 sono del tutto equivalenti.

2 Cancellazione numerica

Finora non abbiamo trovato punti critici: l'errore introdotto da un'operazione di macchina, applicata quindi a dei numeri già arrotondati, è semplicemente un errore di arrotondamento sul risultato; di conseguenza, questo è sicuramente più piccolo di ε_{m}. È vero, abbiamo notato cose un po' strane per quanto riguarda le proprietà delle operazioni di macchina, o per il fatto che due espressioni che sarebbero diverse in aritmetica esatta risultino esattamente *equivalenti* in aritmetica di macchina, quindi *dal punto di vista del calcolatore*; tutti questi errori, però, sono assolutamente minuscoli: sempre inferiori alla precisione di macchina! Ma a questo punto, è d'obbligo chiedersi:

> «Se né arrotondare un numero né le operazioni di macchina introducono problemi, allora da dove nascono 'sti problemi? Cioè, insomma, cos'è che causa i disastri di cui si parlava nell'introduzione?!»

La risposta alla domanda è nascosta nella domanda stessa. Infatti, se leggiamo con attenzione, possiamo notare che:

- l'operazione di arrotondamento coinvolge solo i numeri reali/razionali che si desidererebbe rappresentare in aritmetica macchina, ma non considera in alcun modo ciò che succederà dopo, ovvero, come *useremo* i numeri arrotondati;
- le operazioni di macchina producono un numero di macchina a partire da operandi, assumendo che essi siano già dei numeri di macchina; tuttavia, quindi, non prende in alcun modo in considerazione l'operazione di arrotondamento che li produce.

Se però pensiamo all'intero procedimento, questo è:

1. partire da due numeri che in generale possono essere reali, a_1 e a_2;
2. arrotondarli, ottenendo due numeri $\overline{a}_1$, $\overline{a}_2$;
3. eseguire l'operazione $\overline{a}_1 \cdot \overline{a}_2$ in aritmetica esatta;
4. arrotondare questo risultato: $\overline{\overline{a}_1 \cdot \overline{a}_2}$, concludendo l'operazione di macchina cominciata nel punto precedente.

Nonostante presi singolarmente arrotondamento e operazioni di macchina non generino grossi errori, **usarli assieme**, come si deve di fatto fare, può portare a disastri. Per capire meglio questo punto, è opportuno concentrarsi su un esempio.

2.1 Esempio e definizione di cancellazione numerica

Consideriamo $N = 10$, $t = 5$. In questo caso, si può vedere facilmente con le solite formule che $\varepsilon_{\mathrm{m}} = 0.5 \times 10^{-4}$. Consideriamo quindi i numeri $a_1 = 0.157824831$, $a_2 = 0.157348212$. Il nostro obiettivo sarà calcolare la loro differenza in aritmetica esatta e in aritmetica di macchina, e capire se le cose continuano ad andar bene come finora è successo.

Prima di tutto, proviamo a calcolare quindi $a_1 - a_2$, ovvero la differenza dei due numeri esatti, **non arrotondati**:

$$\begin{array}{r} 0.1\,5\,7\,8\,2\,4\,8\,3\,1 \\ -\ 0.1\,5\,7\,3\,4\,8\,2\,1\,2 \\ \hline 0.0\,0\,0\,4\,7\,6\,6\,1\,9, \end{array}$$

ovvero, $a_1 - a_2 = 0.476619 \times 10^{-3}$. Questo non è un numero di macchina, ma, se immaginiamo di arrotondare questo risultato, non notiamo nulla di devastante; infatti, si otterrebbe

$$\overline{a_1 - a_2} = 0.47662 \times 10^{-3},$$

dove l'errore commesso come al solito sarebbe minore di ε_{m}.

Tuttavia, l'operazione che abbiamo appena descritto non è davvero quella che si esegue quando si lavora con un calcolatore, è in qualche modo *idealizzata*. Infatti, come descritto poco fa nel testo, **prima si arrotonda**, **poi si esegue l'operazione di macchina a partire dai numeri arrotondati**: non è possibile eseguire l'operazione di macchina su numeri *esatti*: il calcolatore in questione non è stato progettato per memorizzarli!

Volendo quindi seguire la procedura *corretta*, prima dovremmo arrotondare i due numeri, ottenendo $\overline{a}_1 = 0.15782$, $\overline{a}_2 = 0.15735$. Come al solito, l'errore di arrotondamento è inferiore a ε_{m}: potremmo calcolarlo, ma non è necessario, dal momento che siamo sicuri della nostra stima della lezione precedente! Quindi, ora, dovremmo calcolare, a partire dai numeri arrotondati,

$$\begin{array}{r} 0.1\,5\,7\,8\,2 \\ -\ 0.1\,5\,7\,3\,5 \\ \hline 0.0\,0\,0\,4\,7, \end{array}$$

che è già un numero di macchina, e che si può scrivere come

$$\overline{a}_1 \ominus \overline{a}_2 = \overline{\overline{a}_1 - \overline{a}_2} = 0.47 \times 10^{-3}.$$

Vedere solo due cifre nel risultato dovrebbe già far nascere un qualche senso di inquietudine. A questo punto, però, proviamo a calcolare l'errore relativo tra il risultato della prima operazione e quello della versione *corretta*:

$$e_{\mathrm{r}} = \frac{|(a_1 - a_2) - (\overline{a}_1 \ominus \overline{a}_2)|}{|a_1 - a_2|} = \frac{0.47662 - 0.47}{0.47662}$$

se lo calcoliamo con MATLAB®, per esempio,

```
>> abs(0.47662-0.47)/(0.47662)
ans =
    0.0139
```

ossia, $e_{\mathrm{r}} \simeq 0.1 \times 10^{-1}$: **questo è molto, molto maggiore rispetto alla precisione di macchina!!!** Abbiamo provato a calcolare l'arrotondamento solo del risultato, ottenuto a partire dall'operazione eseguita sui numeri esatti e non abbiamo riscontrato problemi; l'arrotondamento di per sé non introduce problemi; ciononostante, **unire le due sotto-operazioni può introdurre un'amplificazione degli errori di arrotondamento!**

Il fatto di aver *perso* cifre significative nel risultato dell'operazione di macchina *corretta*, *non idealizzata*, è la conseguenza del cosiddetto fenomeno di **cancellazione numerica**. Visto che si parla di *sparizione di cifre*, possiamo intuire che l'operazione che permette un fenomeno del genere debba essere la sottrazione: è l'unica che possa far *sparire* qualcosa[1]! Inoltre, guardando con

[1] questa intuizione verrà formalizzata meglio nella prossima sezione

attenzione l'esempio, notiamo che i due numeri sono abbastanza simili tra loro. Questo ci permette di capire che **la cancellazione numerica avviene quando si effettua la sottrazione tra due numeri simili tra loro**.

Si noti che **simili** significa che l'esponente dei numeri di partenza deve essere lo stesso (altrimenti i due numeri non sarebbero confrontabili), così come il segno (altrimenti la differenza diventerebbe una somma), e almeno uno dei due sottraendi deve essere affetto da errori di arrotondamento (altrimenti, la *perdita di cifre* sarebbe in verità il risultato corretto). Consideriamo alcuni esempietti in cui **non** avviene cancellazione numerica.

2.1.1 Esempio 1

Consideriamo $N = 10$, $t = 5$, $a_1 = 0.15782$, $a_2 = 0.15735$. Questi due numeri sono numeri di macchina, quindi $\overline{a}_1 = a_1$, $\overline{a}_2 = a_2$. Se si calcolano $a_1 - a_2$, o $\overline{a}_1 - \overline{a}_2$, si otterranno esattamente gli stessi risultati e, quindi, non si avrà cancellazione numerica.

2.1.2 Esempio 2

Consideriamo $N = 10$, $t = 5$, $a_1 = 0.157824831 \times 10^0$, $a_2 = 0.157348212 \times 10^{-1}$. Questi due numeri non sono numeri di macchina, quindi andranno arrotondati, ottenendo $\overline{a}_1 = 0.15782 \times 10^0$, $\overline{a}_2 = 0.15735 \times 10^{-1}$. Tuttavia, se ora calcoliamo la loro sottrazione esatta $a_1 - a_2$, avremo

$$\begin{array}{r} 0.1\,5\,7\,8\,2\,4\,8\,3\,1\,0 \\ -\ 0.0\,1\,5\,7\,3\,4\,8\,2\,1\,2 \\ \hline 0.1\,4\,2\,0\,9\,0\,0\,0\,9\,8 \end{array},$$

e, per quanto riguarda la sottrazione di macchina,

$$\begin{array}{r} 0.1\,5\,7\,8\,2\,0 \\ -\ 0.0\,1\,5\,7\,3\,5 \\ \hline 0.1\,4\,2\,0\,8\,5 \end{array},$$

e, chiaramente, non si hanno errori clamorosi. Questo perché **in questo caso, l'esponente è diverso**.

2.1.3 Un esempio di cancellazione numerica

Abbiamo quindi capito che i problemi devastanti di cui si parlava nascono semplicemente dal sottrarre due numeri simili tra loro.

> «Ma, cioè, ma chi se ne frega! Capita così spesso?!»

Sì. Per esempio, un calcolo con cui i reduci di Analisi Matematica I dovrebbero avere una certa confidenza, è

$$\lim_{h \to 0} \frac{f(x+h) - f(x)}{h},$$

ossia, il calcolo della derivata mediante limite del rapporto incrementale con incremento h. Molto spesso, capita di non conoscere un'espressione *in forma chiusa* della funzione $f(x)$, bensì solo per punti. In questo caso, dovendo calcolare la pendenza in un punto, si può usare questo limite calcolando esplicitamente questo rapporto. Il problema è che

- se mettiamo un h grande, la pendenza che si ottene non è significativa, poiché stiamo semplicemente calcolando la pendenza della retta che interseca i punti x e $x + h$ lontani tra loro;
- se mettiamo un h troppo piccolo, $f(x+h) \simeq f(x)$, che è proprio il caso sfortunato per cui avvengono i fenomeni di cancellazione numerica. :-(

2.2 Idee per risolvere la cancellazione numerica

Abbiamo intuito da dove nasce problema della cancellazione numerica, dunque ora il lettore si aspetta che gli si possa dare una *ricetta* in grado di risolverlo. Purtroppo, sarà allora piuttosto deluso, perché:

- una ricetta *unica* e *indolore* non esiste;
- esistono svariati *trucchetti* che possiamo usare nelle diverse situazioni che ci capitano, e che in effetti in quelle situazioni sono molto efficaci;
- nel caso non si disponga di trucchetti, è sempre possibile **aumentare la precisione di macchina**[2]; questo però è solo un palliativo: il problema della cancellazione continuerà a sussistere, ma apparirà con maggiore difficoltà.

L'obiettivo di questa sezione del testo non vuole essere spingere verso soluzioni *brute-force*, quale per esempio è aumentare la precisione di macchina. Il nostro scopo è imparare almeno alcuni dei trucchi che possono risolvere alcune specifiche situazioni in maniera *pulita*. Purtroppo, ogni situazione è diversa da tutte le altre e, quindi, solo *l'esperienza*, acquisita mediante l'esercizio, può aiutare a capire come districarsi nella giungla della cancellazione numerica.

2.2.1 Esempio 1: razionalizzazione -se possibile-

Si consideri il seguente esempio:

$$f(x) = \sqrt{x+\delta} - \sqrt{x},$$

in cui $x > 0$, $x + \delta > 0$. Evidentemente, abbiamo una sottrazione tra le due radici. Se abbiamo che $|\delta| \ll x$, si avrà che il primo termine sarà molto simile al secondo, e quindi potrebbero verificarsi fenomeni di cancellazione numerica. Un trucco che permette di risolvere questo caso è basato sulla **razionalizzazione**, idea usata spesso anche in Analisi Matematica I per risolvere limiti con forme indeterminate. In particolare, si può moltiplicare a numeratore e denominatore per il termine $\sqrt{x+\delta} + \sqrt{x}$, ottenendo:

$$\begin{aligned} f(x) = \sqrt{x+\delta} - \sqrt{x} &= (\sqrt{x+\delta} - \sqrt{x})\frac{\sqrt{x+\delta}+\sqrt{x}}{\sqrt{x+\delta}+\sqrt{x}} = \frac{x+\delta-x}{\sqrt{x+\delta}+\sqrt{x}} = \\ &= \frac{\delta}{\sqrt{x+\delta}+\sqrt{x}}. \end{aligned}$$

Scritta in questo modo, questa espressione non contiene più le sottrazioni *critiche*, e quindi non può essere affetta da problemi di cancellazione numerica per δ piccolo. Inoltre, dal momento che per passare dalla scrittura di partenza a quella di arrivo non sono state sfruttate né approssimazioni di alcun tipo, né proprietà dell'aritmetica di macchina, le due espressioni sono totalmente **equivalenti in aritmetica esatta**.

Tuttavia, questo esempio apparentemente banale contiene altre insidie. Infatti, se $\delta \simeq -x$, la differenza tra le due radici non è più un problema, ma all'interno di $\sqrt{x+\delta}$ si verificherebbe un **altro** problema di cancellazione, che in questo caso non sarebbe risolubile con la razionalizzazione (poiché razionalizzare farebbe apparire la radice in questione al denominatore, come abbiamo visto, e questa continuerebbe a essere affetta dal problema). Dunque, per $\delta \simeq -x$, **non abbiamo soluzioni banali al problema della cancellazione numerica**.

2.2.2 Esempio 2: espansione di Taylor

Si consideri l'espressione

$$f(x) = \frac{-1+e^x}{x}.$$

[2] per esempio esiste un tool a pagamento per MATLAB®, con informazioni disponibili presso https://www.advanpix.com/; in effetti questa è la soluzione che ho adottato per il mio simulatore di trasporto elettronico D1ANA usato per esempio in [1]

Quando $x \simeq 0$, si ha che $e^0 \simeq 1$, e quindi si verifica un fenomeno di cancellazione numerica. Per risolvere questo problema, una soluzione può riguardare l'utilizzo di un'espansione di Taylor nell'intorno del punto di interesse. Infatti, come noto da Analisi Matematica I, l'espansione di Maclaurin (Taylor in $x = 0$) è:

$$e^x \simeq 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots$$

In particolare, quindi, la funzione diverrebbe

$$\begin{aligned} f(x) &\simeq \frac{1}{x}\left(-1 + 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots\right) = \\ &= 1 + \frac{x}{2!} + \frac{x^2}{3!} + \dots \end{aligned}$$

che non contiene più differenze, e quindi nemmeno problemi di cancellazione numerica!

2.2.3 Esempio 3: relazioni trigonometriche -quando disponibili-

Sembrerebbe che utilizzare espansioni di Taylor / Maclaurin possa essere la fantastica *ricetta definitiva* che ci piacerebbe avere: più o meno ogni funzione ammette uno sviluppo di Taylor! Purtroppo, però, questa strada ha due controindicazioni.

- Questi sviluppi di Taylor richiedono -in teoria- la somma di infiniti termini e, quindi, valutarli può richiedere un certo sforzo computazionale.
- Considerando per esempio il caso delle funzioni oscillanti come seno o coseno, gli sviluppi di Taylor sono a segni alterni, quindi introdurre Taylor non elimina necessariamente l'operazione di sottrazione dalle espressioni.

Considerando per esempio la funzione

$$f(x) = \frac{1 - \cos x}{x^2},$$

Taylor non rappresenta la via più indicata. Tuttavia, possiamo ricordarci le formule di bisezione del seno,

$$\sin\left(\frac{x}{2}\right) = \pm\sqrt{\frac{1 - \cos x}{2}}.$$

Manipoliamo quindi la nostra $f(x)$ al fine di poter sfruttare questa espressione; per far questo, moltiplichiamo e dividiamo per due, ottenendo

$$\frac{1 - \cos x}{x^2} = \frac{2}{x^2}\frac{1 - \cos x}{2}.$$

Possiamo quindi sostituire in questa espressione il quadrato dell'espressione appena trovata (facendo così anche sparire l'indeterminazione di segno $\pm$), ottenendo

$$\frac{2}{x^2}\frac{1 - \cos x}{2} = \frac{2}{x^2}\sin^2\left(\frac{x}{2}\right) = \frac{1}{2}\frac{4}{x^2}\sin^2\left(\frac{x}{2}\right) = \frac{1}{2}\left[\frac{\sin\left(\frac{x}{2}\right)}{\frac{x}{2}}\right]^2.$$

Questa funzione per $x \to 0$ non presenta alcun problema[3], e quindi la cancellazione numerica è stata risolta.

[3] se non una insignificante discontinuità eliminabile

3 Condizionamento di un problema numerico

Nella precedente sezione abbiamo capito, studiando un certo numero di esempi, che esiste la possibilità di avere fenomeni di amplificazione degli errori, ovvero, di crescita imprevista rispetto alle stime che siamo riusciti a ottenere sui semplici arrotondamenti. In effetti, il meccanismo alla base di questa amplificazione è una specie di *combo* di arrotondamento dei dati e arrotondamento delle operazioni di macchina. In aggiunta a tutto ciò, questi esempi ci hanno permesso di capire che alcune situazioni in cui si verifica la cancellazione numerica si possono risolvere mediante dei *trucchi*, ma altre no. Volendo quindi dirlo con parole diverse, alcuni problemi sono **intrinsecamente presenti nel problema**, mentre altri sono **contenuti nell'algoritmo**, ovvero, modificando la procedura di calcolo è possibile rimuoverli. Con riferimento all'esempio nella sezione 2.2.1, il problema per $|\delta| \ll x$ era risolubile modificando la procedura di calcolo della funzione, quindi l'algoritmo, mentre la cancellazione legata a $\delta \simeq -x$ non era trattabile, e quindi **intrinsecamente presente nel problema**. L'obiettivo di questa sezione sarà formalizzare un po' meglio queste definizioni, fino a spiegare, in termini diversi, cosa sia il problema della cancellazione numerica.

3.1 Definizione di problema numerico; forma implicita, forma esplicita

Un problema numerico in questo contesto è una relazione f che lega i dati di ingresso (input), ossia le x, ai dati di uscita (output), ovvero le y. A seconda della situazione, un problema numerico può essere fornito in forma **esplicita**, ovvero del tipo

$$y = f(x),$$

o in forma **implicita**, ovvero

$$f(x, y) = 0.$$

La particolarità della forma esplicita è che a partire dal valore dell'input x, è direttamente possibile ottenere gli output. Questo non è possibile nella forma implicita. Per capire questo concetto, si pensi a una retta:

- l'equazione della retta può essere scritta in forma esplicita, ovvero

  $$y = mx + q,$$

  e quindi, per vari valori di x, è immediatamente possibile trovare i valori di y;

- l'equazione della retta può anche essere scritta in forma implicita, ovvero

  $$ax + by + c = 0;$$

  in questo caso, non è possibile, a partire dai valori di x, trovare immediatamente quelli di y; tutto ciò che si può fare è, data una coppia di valori (x, y), capire se essi soddisfano l'equazione e quindi appartengono alla retta.

Nel caso della retta, con semplici manipolazioni, è possibile ottenere una rappresentazione dall'altra; questo non è il caso di molti altri problemi. Per esempio, si consideri il problema matriciale[4]

$$\underline{\underline{A}}\,\underline{y} - \underline{x} = 0.$$

Questa è un'equazione matriciale: un sistema lineare. Questo è scritto in forma implicita, dal momento che si dovrebbero provare vari vettori $\underline{x}$ e $\underline{y}$ nell'equazione, per verificare se è soddisfatta. Al contrario, è possibile a certe condizioni scrivere questo problema in forma implicita come

$$\underline{y} = \underline{\underline{A}}^{-1}\underline{x},$$

dove $\underline{\underline{A}}^{-1}$ è detta *matrice inversa*.

[4] in queste note, per chiarezza, le matrici verranno sempre indicate con una doppia sottolineatura, e i vettori con una singola sottolineatura

3.2 Definizione di condizionamento di un problema numerico

In un problema numerico, da un ingresso x, si immagini di ottenere un numero $\overline{x}$, dopo averlo perturbato[5]. Quindi, si calcoli

$$\overline{y} = f(\overline{x}).$$

Questo significa che $\overline{y}$ è il risultato dell'operazione f, effettuata in precisione infinita (aritmetica esatta), a partire dai dati perturbati $\overline{x}$. Se l'errore su $\overline{y}$ è dello stesso ordine di grandezza di $\overline{x}$, il problema si dice **ben condizionato**. Volendo scrivere questa proposizione in *matematichese*, avremmo

$$\frac{|y-\overline{y}|}{|y|} \leq \kappa(f,x)\frac{|x-\overline{x}|}{|x|},$$

dove $y = f(x)$ è il risultato ottenuto a partire dal dato non perturbato (e lavorando sempre in aritmetica esatta). Il senso di questa operazione, quindi è semplicemente quello di concentrarsi sul fatto che perturbare un dato di ingresso non causi un'esplosione dell'errore. In altre parole, si desidera trovare un valore per κ che ci permetta di avere la certezza che il membro sinistro non sia più grande del membro destro; se però il κ che troviamo è un numero grande, allora, nel caso peggiore, la perturbazione dell'ingresso sarà fortemente amplificata sull'uscita. Il problema è dunque detto detto **ben condizionato** se è possibile trovare una costante di amplificazione dell'errore κ, la quale dipende dal problema f e dall'input x, che sia **piccola**; la costante κ viene per questo detta **numero di condizionamento**.

3.3 Esempio: cancellazione numerica

Al fine di applicare i concetti introdotti nella precedente sezione, ci poniamo l'obiettivo di studiare il condizionamento del problema

$$y = x_1 + x_2, \qquad x_1, x_2 \in \mathbb{R}.$$

Sfruttando la teoria prima introdotta, l'idea è perturbare i dati di ingresso x_1 e x_2, ottenendo $\overline{x}_1$, $\overline{x}_2$. Definiamo quindi gli errori relativi[6] ε_1, ε_2, come

$$\varepsilon_1 = \frac{\overline{x}_1 - x_1}{x_1} \qquad \varepsilon_2 = \frac{\overline{x}_2 - x_2}{x_2}.$$

Mediante semplici operazioni aritmetiche, è possibile riscrivere i valori perturbati degli ingressi come segue:

$$\begin{aligned}
\overline{x}_1 &= \overline{x}_1 + x_1 - x_1 = x_1 + \overline{x}_1 - x_1 = x_1\left(1 + \frac{\overline{x}_1 - x_1}{x_1}\right) = x_1(1+\varepsilon_1) \\
\overline{x}_2 &= \overline{x}_2 + x_2 - x_2 = x_2 + \overline{x}_2 - x_2 = x_2\left(1 + \frac{\overline{x}_2 - x_2}{x_2}\right) = x_2(1+\varepsilon_2).
\end{aligned}$$

Dal momento che $y = x_1 + x_2$, e che $\overline{y} = \overline{x}_1 + \overline{x}_2$, è possibile scrivere l'errore relativo su $\overline{y}$ come:

$$\begin{aligned}
\frac{y-\overline{y}}{y} &= \frac{x_1 + x_2 - (\overline{x}_1 + \overline{x}_2)}{x_1 + x_2} = \frac{x_1 + x_2 - x_1(1+\varepsilon_1) - x_2(1+\varepsilon_2)}{x_1 + x_2} = \\
&= \frac{x_1 + x_2 - x_1 - x_1\varepsilon_1 - x_2 - x_2\varepsilon_2}{x_1 + x_2} = -\frac{x_1}{x_1 + x_2}\varepsilon_1 - \frac{x_2}{x_1 + x_2}\varepsilon_2.
\end{aligned}$$

A questo punto, definiamo

$$\kappa_1 \triangleq \left|\frac{x_1}{x_1 + x_2}\right| \qquad \kappa_2 \triangleq \left|\frac{x_2}{x_1 + x_2}\right|,$$

[5] per esempio arrotondare un numero reale a per ottenere il numero di macchina $\overline{a}$ si può vedere come una perturbazione: una piccola variazione dell'input!

[6] al fine di semplificare i conti si omette il valore assoluto; questo non riduce la generalità della dimostrazione

e così si può stimare l'errore relativo su $\overline{y}$ come

$$\left|\frac{y-\overline{y}}{y}\right| \leq \kappa_1\varepsilon_1 + \kappa_2\varepsilon_2 \leq \kappa(|\varepsilon_1| + |\varepsilon_2|),$$

che è stato maggiorato dall'ultima espressione, dove abbiamo definito κ come il massimo tra κ_1 e κ_2 (e quindi è certamente più grande), e i valori assoluti. Riassumendo, abbiamo quindi ricavato che

$$\left|\frac{y-\overline{y}}{y}\right| \leq \kappa \left(\underbrace{\left|\frac{\overline{x}_1 - x_1}{x_1}\right|}_{|\varepsilon_1|} + \underbrace{\left|\frac{\overline{x}_2 - x_2}{x_2}\right|}_{|\varepsilon_2|} \right),$$

con

$$\kappa = \left|\frac{\max\{|x_1|, |x_2|\}}{x_1 + x_2}\right|.$$

L'espressione di κ è scritta in modo tale da mettere immediatamente in evidenza quale sia la situazione di *mal condizionamento*. Infatti, κ, e dunque il fattore di amplificazione degli errori sugli ingressi, è grande quando $x_1 + x_2 \to 0$, ossia quando $x_1 \to -x_2$. Questo, guarda caso, è esattamente il caso in cui si effettua la sottrazione di due numeri molto vicini tra loro: **la cancellazione numerica**! Tutto questo conto, quindi, ha permesso di formalizzare meglio l'idea di cancellazione numerica.

4 Stabilità di un algoritmo

Per concludere l'argomento, dobbiamo introdurre un'idea un po' *duale* a quella di condizionamento. Se vogliamo, il *condizionamento* è legato ai *problemi del problema numerico*: a partire dal problema numerico, lo applichiamo -sempre con precisione infinita di calcolo- prima sui dati esatti, poi sui dati approssimati, e, studiando l'errore tra questi due risultati, definiamo il numero di condizionamento come una sorta di *cifra di merito* della *qualità* del problema, facendo finta che esso agisca con precisione infinita.

Quando si parla di **stabilità dell'algoritmo**, si fa qualcosa di molto diverso. Si parta da $\overline{x}$, quindi dai dati già approssimati, senza porsi ulteriori domande su cosa accadrebbe se si lavorasse con i dati esatti. Poi, ci chiediamo, focalizzandoci esclusivamente su questi $\overline{x}$, cosa succederebbe calcolando il risultato dell'algoritmo in precisione di macchina e in aritmetica esatta. A tal fine, è opportuno ricordare che **un algoritmo è una sequenza finita di operazioni che permette di calcolare l'output di un problema**. Ovviamente, **sequenza finita** è un sinonimo di **precisione finita**: non è possibile ottenere un risultato in aritmetica esatta usando un numero finito di passaggi (si pensi per esempio a sommare $\sqrt{2}$ e π: sarebbe necessario avere infinite operazioni!). In questo senso, dato $\widetilde{y}$ il risultato dell'algoritmo applicato con precisione finita, e $\overline{y}$ quello dell'algoritmo applicato in precisione infinita, viene detto **stabile** un algoritmo tale per cui

$$\frac{|\widetilde{y} - \overline{y}|}{|\overline{y}|} < \varepsilon_{\mathrm{m}}.$$

Riferimenti bibliografici

[1] M. Calciati, A. Tibaldi, F. Bertazzi, M. Goano, and P. Debernardi, "Many-valley electron transport in AlGaAs VCSELs," *Semicond. Sci. Technol.*, vol. 32, no. 5, 055007, Apr. 2017.