1.1 L'analisi numerica

1.1.1 Introduzione

Il problema di fondo dell'analisi numerica è quello che i calcoli fatti a macchina e anche a mano non sono quasi mai esatti; dipendono da come vengono fatti e da come vengono inseriti i dati. A seconda poi di come vengono fatti possono richiedere più o meno tempo, essere più o meno precisi.

Spesso il tempo e la precisione sono inversamente proporzionali.

L'analisi numerica si propone quindi di elaborare le migliori tecniche di calcolo e di studiare questi fenomeni.

Comunque, anche quando si usano le migliori tecniche, e non ci si preoccupa del tempo, può darsi che i risultati di un calcolo siano comunque imprecisi. Questo può dipendere dalla natura stessa del problema: se il problema è *mal condizionato* i risultati del calcolo sono suscettibili di grandi variazioni a fronte di piccole variazione nei dati iniziali, il che li rende comunque poco affidabili.

L'analisi numerica ha quindi due aspetti strettamente collegati:

- Analisi del problema e tecniche di soluzione
- Analisi dell'errore e tecniche per renderlo minimo.

1.1.2 Alcuni esempi elementari

Esempio 1.1: Tabulare la funzione $f(x) = \dfrac{1+\sqrt{1+x^2}}{\sqrt{1+x^2}}$.

Se si scrive con un computer qualcosa come:

```
f(x)=(1+sqrt(1+x^2))/sqrt(1+x^2)
```

il valore di `sqrt(1+x^2)` viene calcolato due volte, con evidente cattivo impiego del tempo, quindi conviene un approccio in due passi, solo in apparenza più complesso, del tipo:

```
t=sqrt(1+x^2)
f(x)=(1+t)/t
```

Esempio 1.2: Se a, b, c sono numeri reali, allora, come è ben noto, si ha: $a(b+c) = ab + ac$

Però $a(b+c)$ richiede una somma e un prodotto, mentre $ab + ac$ richiede due prodotti e una somma, quindi maggior tempo di calcolo.

Vedremo anche che tra le due espressioni equivalenti $(a+b)+c$ e $a+(b+c)$, che richiedono lo stesso tempo di calcolo, una di esse possa essere, in certi casi, più conveniente dell'altra dal punto di vista numerico.

Esempio 1.3: Consideriamo un sistema lineare *quadrato* $Ax = b$ con A matrice invertibile che quindi ha, come ben noto, una e una sola soluzione

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ & \cdots & \cdots & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \quad b = \begin{pmatrix} b_1 \\ \cdots \\ b_n \end{pmatrix} \text{ cioè } \begin{cases} a_{11}x_1 & +\cdots & +a_{1n}x_n & = & b_1 \\ & \cdots & \cdots & & \\ a_{n1}x_1 & +\cdots & +a_{nn}x_n & = & b_n \end{cases}$$

Ci sono almeno tre metodi elementari per risolverlo:

1 Il noto algoritmo di eliminazione di Gauss che richiede circa $\dfrac{n^3}{3}$ moltiplicazioni.

2 L'uso dell'espressione $x = A^{-1} \cdot b$ che però richiede circa n^3 moltiplicazioni per il solo calcolo di A^{-1}.

3 La nota regola di Cramer: $x_i = \dfrac{\det(A_i)}{\det(A)}$ che richiede circa $\dfrac{n^3}{3}$ moltiplicazioni per incognita se i determinanti delle $n+1$ matrici vengono calcolati con l'algoritmo di Gauss.
Se poi i determinanti vengono calcolati mediante lo sviluppo di Laplace, ognuno richiede circa $n!$ prodotti.

Quindi il metodo in apparenza peggiore, perché richiede un'elaborazione abbastanza complessa (l'algoritmo di Gauss), è in realtà di gran lunga il più conveniente dal punto di vista del tempo di calcolo.

Esempio 1.4: Calcolare la funzione $f(x) = \dfrac{1}{10^5} - \dfrac{1}{x}$ per $x = 10^5 + 1 = 100001$.

Se la nostra calcolatrice si limita a cinque cifre decimali significative, essa sarà in grado di scrivere correttamente $1/10^5$ come 10^{-5}, ma scriverà anche $1/(10^5 + 1)$ come 10^{-5} per cui il risultato sarà 0.

Se la funzione viene scritta nel modo equivalente $\dfrac{x - 10^5}{10^5 x}$, quando si sostituisce a x il valore $10^5 + 1$ si ottiene 1 a numeratore e $10^5 \cdot (10^5 - 1)$ a denominatore, ovvero in totale circa 10^{-10}. La differenza tra un risultato che è 0 e un risultato diverso da zero (benché apparentemente piccolo) è enorme e spesso rischia di vanificare del tutto un calcolo.

Esempio 1.5: Calcolare gli integrali definiti $\displaystyle\int_0^1 \frac{x+1}{x^2+1}dx \quad \int_0^1 \frac{x}{\cos(x)}dx$

La prima funzione integranda ammette primitiva elementare, e si scrive facilmente

$$\int_0^1 \frac{x+1}{x^2+1}dx = \left[\frac{1}{2}\ln(x^2+1) + \arctan(x)\right]_0^1$$

La funzione $x/\cos(x)$ non è integrabile elementarmente, per cui apparentemente il primo integrale è molto più semplice del primo.
In realtà, anche se del primo integrale abbiamo una formula esplicita, il computo delle funzioni logaritmo e arcotangente può essere relativamente lungo anche per un computer, per cui, usando formule approssimate di quadratura, il secondo integrale può risultare più semplice del primo.
L'avere una formula sintetica per il primo integrale può comunque essere d'aiuto in talune questioni.

Esempio 1.6: Tabulare un polinomio.

Sia per esempio $P(x) = 1 + 5x - 2x^2 + 3x^3 + 6x^4$
Scrivere con un computer qualcosa come:

```
P(x)=1+5*x-2*x^2+3*x^3+6*x^4
```

non è la cosa più conveniente (4 somme, 4 prodotti, tre potenze)

Leggermente meglio sarebbe

```
P(x)=1+5*x-2*x*x+3*x*x*x+6*x*x*x*x
```

con solo 4 somme, 10 prodotti (i prodotti richiedono meno tempo di calcolo delle potenze)

Un certo miglioramento si avrebbe mediante l'uso di un'array ausiliaria `t()`

```
t(1)=x  ;  t(2)=t(1)*x  ;  t(3)=t(2)*x  ;  (4)=t(3)*x
P(x)=1+5*t(1)-2*t(2)+3*t(3)+6*t(3)
```

con solo 4 somme, 7 prodotti

La cosa migliore è però quella di usare lo schema di Ruffini-Hörner esposto qui di seguito.

1.1.3 Lo schema di Ruffini-Hörner

Consideriamo come sopra il polinomio $P(x) = 1 + 5x - 2x^2 + 3x^3 + 6x^4$

Lo schema di Ruffini-Hörner consiste nello scrivere il polinomio come

$$1 + x\Big(5 + x\big(-2 + x\,(3 + x \cdot 6)\big)\Big)$$

Calcoliamo per esempio $1 + 5x - 2x^2 + 3x^3 + 6x^4$ per $x_0 = 2$.

$$\begin{array}{rcrcr|lcr}
6 & & & \to & 6 & x_0 \cdot 6 & \to & 12 \\
3 & + & 12 & \to & 15 & x_0 \cdot 15 & \to & 30 \\
-2 & + & 30 & \to & 28 & x_0 \cdot 28 & \to & 56 \\
5 & + & 56 & \to & 61 & x_0 \cdot 61 & \to & 122 \\
1 & + & 122 & \to & 123 & & \Rightarrow & P(2) = 123
\end{array}$$

Richiede solo 4 somme e 4 prodotti.
Non è difficile scrivere la formula generale per un polinomio qualunque.
Vedremo poi che lo schema è utile in altre circostanze.

1.2 Errori

Definizione: Se $x \in \mathbb{R}$ e $\tilde{x}$ è il suo valore "calcolato", definiamo;
Errore assoluto $\tilde{x} - x$ (o anche $\mid \tilde{x} - x \mid$)
Errore relativo $(x \neq 0)$ $\dfrac{\tilde{x} - x}{x}$ (o anche $\dfrac{\mid \tilde{x} - x \mid}{\mid x \mid}$)

Questo significa per esempio che non ha molto senso calcolare $x = 0$ con un certo errore relativo (o è 0 o non lo è).

Quando non si conosce x, ma si conosce $\tilde{x}$ e si sa maggiorare l'errore assoluto come $\mid \tilde{x} - x \mid < \varepsilon$, si usa scrivere $x = \tilde{x} \pm \varepsilon$.

Possibili sorgenti di errore:

1. **Modello troppo semplice**. Per esempio voler rappresentare con un modello lineare un fenomeno che è molto più complesso.
2. **Errore nei dati**. Dipendono da informazioni e/o misurazioni.
3. **Errore di arrotondamento**. Ne discuteremo più a lungo in seguito. Per esempio scrivendo $1/3 = 0,3333333$, per quante numerose siano le cifre decimali non si potrà mai avere una eguaglianza.
4. **Errore di troncamento del calcolo**. Un algoritmo indefinito di approssimazione deve cessare ad un certo punto. Per esempio algoritmi tipo tangenti di Newton o altri che vedremo.
5. **Errore umano** (o più raramente di macchina). La possibilità di aver per esempio toccato un tasto inavvertitamente e aver cambiato un dato va sempre presa in considerazione.

1.3 Basi numeriche e rappresentazione di numeri

1.3.1 Numeri interi

Fissiamo $b \in \mathbb{N}$, $b \geq 2$, detto *base*.

Proposizione 1 *Sia ora $n \in \mathbb{Z}, n \neq 0$ un qualunque numero intero, allora esiste un'unica* rappresentazione di n in base b, *cioè un'espressione del tipo*

$$n = d_0 + d_1 b + \cdots + d_r b^r \qquad 0 \leq d_i < b \qquad d_r \neq 0$$

La cifra d_r è detta *cifra più significativa*, mentre la cifra d_0 è detta *cifra meno significativa* del numero n rappresentato in base b.

La cifra più significativa è sempre diversa da 0. Teniamo presente che il numero 0, che non ha cifre significative, è sempre un caso particolare.

Esempio 1.7: $b = 10$ è la base comunemente usata (solo per motivi storici).
Il numero 4073 si scrive quindi come

$$4073 = 3 + 7 \cdot 10 + 0 \cdot 10^2 + 4 \cdot 10^3 \qquad d_0 = 3 \quad d_1 = 7 \quad d_2 = 0 \quad d_3 = 4 \neq 0$$

Le basi comunemente usate oltre al 10 sono 2, 8, 16.
In informatica la base fondamentale è 2, ma le rappresentazioni di numeri in base 2 sono di solito molto lunghe, quindi, negli usi pratici si usano la base 8 (ottale) e soprattutto la base 16 (esadecimale), solo per il motivo che è immediato il passaggio dalla rappresentazione in base 2 a quella in base 16 e viceversa, mentre è complicato il passaggio dalla base 10 alla base 2 e viceversa.

Storicamente sono state usate anche la base 60 (in Babilonia, ne abbiamo un ricordo nella divisione di angoli e ore in 60 minuti e secondi) e la base 20. Come curiosità notiamo per esempio che in francese 92 si dice $quatre - vingt - douze$ che corrisponde a una rappresentazione in base 20: $92 = 12 + 4 \cdot 20 \qquad d_0 = 12 \quad d_1 = 4 \neq 0.$

Per rappresentare quindi un numero in base b occorrono b simboli che rappresentino le b cifre. In base 10 le cifre sono notoriamente $0, 1, \ldots, 9$, in base 16 occorrono altre 6 cifre che vengono denotate A, B, C, D, E, F. Vediamo i primi 16 numeri naturali in base 10, 8, 16 e 2.

Base 10	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Base 8	0	1	2	3	4	5	6	7	10	11	12	13	14	15	16	17
Base 16	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Base 2	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

1.3.2 Rappresentazione in base 2

Come esempio rappresentiamo il numero 1354 che dividiamo successivamente per 2 tenendo conto del resto:

$1354 : 2$	$=$	677	resto 0
$677 : 2$	$=$	338	resto 1
$338 : 2$	$=$	169	resto 0
$169 : 2$	$=$	84	resto 1
$84 : 2$	$=$	42	resto 0
$42 : 2$	$=$	21	resto 0
$21 : 2$	$=$	10	resto 1
$10 : 2$	$=$	5	resto 0
$5 : 2$	$=$	2	resto 1
$2 : 2$	$=$	1	resto 0
$1 : 2$	$=$	0	resto $\mathbf{1}$

La cifra più significativa è in neretto per chiarezza.
La rappresentazione è
$\mathbf{1}0101001010 = 0 + 1 \cdot 2 + 0 \cdot 2^2 + \cdots + \mathbf{1} \cdot 2^{10}$.
È facile passare alla rappresentazione in base 16 dividendo le cifre binarie in gruppi di 4 partendo da destra e assegnando a ogni gruppo di 4 la sua cifra esadecimale:

101	0100	1010
5	4	A

Infatti $1354_{10} = 54A_{16} = 10 + 4 \cdot 16 + 5 \cdot 16^2$

In realtà il computer, dato che lavora in base 2, non può eseguire il conto precedente che presuporrebbe il numero già scritto in base 2, quindi usa un altro algoritmo basato sullo schema di Ruffini-Hörner.

Il computer ha già in memoria la rappresentazione binaria dei numeri da 0 a 10:

$0_{10} = 0000_2 \qquad 1_{10} = 0001_2 \qquad 2_{10} = 0010_2 \qquad \cdots \qquad 10_{10} = 1010_2$

ed è in grado di eseguire le operazioni aritmetiche con i numeri in base 2, quindi scrive il numero decimale 1354 come

$1354 = 4 + 5 \cdot 10 + 3 \cdot 100 + 1 \cdot 1000 \qquad = \qquad 4 + 10 \cdot (5 + 10 \cdot (3 + 1 \cdot 10))$

Dato che nell'ultima espressione compaiono solo numeri compresi tra 1 e 10 di cui il computer conosce la rappresentazione binaria e con cui è in grado di eseguire i conti, questo permette di determinare la rappresentazione binaria del numero.

1.3.3 Numeri reali

La scelta della base numerica influisce in modo notevole sulla rappresentazione dei numeri reali non interi.

Per esempio il numero reale 1/3 ha la nota rappresentazione decimale $0.3333\ldots$ ovvero $0.\overline{3}$ (periodico), quindi non potrà mai essere scritto in modo esatto in base 10, ma usando per esempio la base 12 (di raro uso) la sua rappresentazione sarebbe 0.4 (non periodico), cioè una rappresentazione esatta.

Viceversa il numero reale 1/10 ha la rappresentazione decimale esatta 0.1 (non periodico), ma usando la base 2 la sua rappresentazione è $0.0\,0011\,0011\,00\ldots = 0.0\,\overline{0011}$ (periodico).

Per chiarire questi concetti introduciamo la rappresentazione dei numeri reali a virgola variabile (*floating-point representation* in inglese).

Fissiamo $b \in \mathbb{Z}$, $b \geq 2$, detto *base*.

Proposizione 2 *Sia $x \in \mathbb{R}, x \neq 0$ un numero reale non nullo, allora esiste un'unica* rappresentazione di x in base b, *cioè un'espressione del tipo*

$$x = s \cdot (d_1 \cdot b^{-1} + d_2 \cdot b^{-2} + \cdots) b^p$$

$s = \pm 1$	è detto *segno*
$p \in \mathbb{Z}$	è un intero detto *caratteristica* o *esponente*
$m = d_1, d_2, d_3, \ldots$	è una successione (spesso infinita) di interi tali che $0 < d_i < b$ detta *mantissa* La successione m non è mai definitivamente $b-1, b-1, b-1, \ldots$ Inoltre si ha $d_1 \neq 0$ e d_1 è detto *prima cifra significativa*

A volte viene detta mantissa non la successione, ma la sommatoria $m = d_1 \cdot b^{-1} + d_2 \cdot b^{-2} + \cdots$ che, quando è infinita, è sempre una serie convergente.
La mantissa m così definita è un numero compreso tra $1/b$ e 1: più precisamente $1/b \leq m < 1$.

Esempio 1.8: Qualche esempio in base 10 per familiarizzare:

$24.3 =$	$1 \cdot (0.243) \cdot 10^2$	segno 1	mantissa 2 4 3 (o 0.243)	caratt. 2
$-0.034 =$	$-1 \cdot (0.34) \cdot 10^{-1}$	segno -1	mantissa 3 4 (o 0.34)	caratt. -1
$1/3 =$	$1 \cdot (0.333\cdots) \cdot 10^0$	segno 1	mantissa $3\,3\,3\cdots$ (o $0.333\cdots$)	caratt. 0
$125\,000\,000 =$	$1 \cdot (0.125) \cdot 10^9$	segno 1	mantissa 1 2 5 (o 0.125)	caratt. 9

La rappresentazione $1 \cdot (0.299999\cdots) \cdot 10^2$ non è valida perché le cifre della mantissa sono tutte $b-1 = 9$ da un certo punto in poi; il numero $29.999\cdots$ è una rappresentazione non valida del numero 30.

Esempio 1.9: Rappresentiamo $(0.9)_{10}$ in base 16:

$a_1 = (0.9)_{10} = \dfrac{d_1}{16} + \dfrac{d_2}{16^2} + \cdots$. Raccolgo 16

$16 \cdot a_1 = (16 \cdot 0.9)_{10} = d_1 + \dfrac{d_2}{16} + \cdots$. Ma $16 \cdot 0.9 = 14.4$ da cui $d_1 = (14)_{10} = E_{16}$

$a_2 = (0.4)_{10} = \dfrac{d_2}{16} + \dfrac{d_3}{16^2} + \cdots$. Raccolgo 16

$16 \cdot a_2 = (16 \cdot 0.4)_{10} = d_2 + \dfrac{d_3}{16} + \cdots$. Ma $16 \cdot 0.4 = 6.4$ da cui $d_2 = (6)_{10} = 6_{16}$

In definitiva $(0.9)_{10} = 0.E6666\cdots$ e quindi $(0.9)_{10} = 1 \cdot (0.E66\cdots) \cdot (16)^1_{10}$
È facile passare alla rappresentazione binaria e vedere la periodicità dello sviluppo del numero:
$(0.9)_{10} = 0.1110\,0110\,0110\cdots = 1 \cdot (0.11\,\overline{1001}) \cdot 10^0$

L'algoritmo dell'esempio precedente non è in realtà esattamente quello che fa il computer, che dovendo lavorare in base 2, esegue di fatto la divisione tra 9 e 10 dopo aver rappresentato i due numeri in base 2.

La rappresentazione a virgola variabile è analoga alla cosiddetta notazione scientifica, usata in fisica e in tecnica e anche sul display delle macchine calcolatrici specialmente per numeri molto grandi o molto piccoli.
La differenza è che la notazione scientifica mette prima della virgola la cifra più significativa e non lo zero, quindi l'esponente in notazione scientifica è inferiore di un'unità.
Vediamo la differenza nei tre esempi precedenti (si suppone una calcolatrice con visore a 8 cifre):

Numero		virgola variabile		notazione scientifica		calcolatrice
24.3	=	$1 \cdot (0.243) \cdot 10^2$	=	2.43×10^1	=	24.3
-0.034	=	$-1 \cdot (0.34) \cdot 10^{-1}$	=	-3.4×10^{-2}	=	-0.034
$1/3$	=	$1 \cdot (0.333\cdots) \cdot 10^0$	=	$3.333\cdots \times 10^{-1}$	=	0.3333333
125 000 000	=	$1 \cdot (0.125) \cdot 10^9$	=	1.25×10^8	=	1.25 e08

1.4 Rappresentazione di numeri su computer

1.4.1 Numeri macchina

Fissiamo i seguenti numeri interi positivi

- $b \geq 2$, la *base*.
- t il numero di cifre della mantissa.
- $[L, U]$ il range (minimo e massimo esponente consentiti).

I numeri macchina sono numeri del tipo $x = s \cdot (d_1 \cdot b^{-1} + d_2 \cdot b^{-2} + \cdots + d_r \cdot b^{-r}) b^p$ cioè numeri a virgola variabile in cui però $L \leq p \leq U$ e la mantissa ha un numero limitato r di cifre con $r \leq t$.
Ovviamente, mentre i numeri reali sono infiniti, i numeri macchina disponibili sono in numero finito.
Per capire come sono disposti i numeri macchina li elenchiamo tutti supponendo, solo per ragioni di semplicità descrittiva, che:

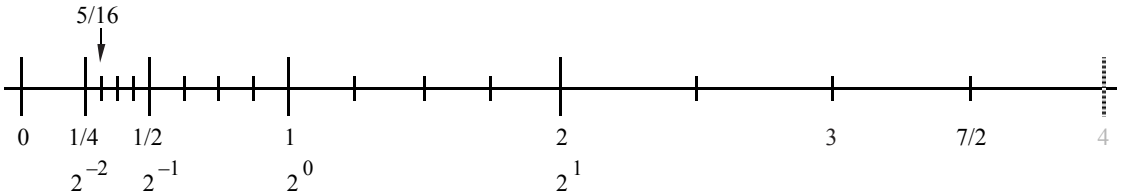
$b = 2 \qquad t = 3 \qquad L = -1 \qquad U = 2$

I numeri rappresentabili con queste limitazioni sono solo 32 (33 se comprendiamo anche lo zero) e cioè $\pm 0.d_1d_2d_3 \cdot 10^p$ (10 è il numero 2 in rappr. binaria)
Si deve avere $d_1 = 1$ (cifra più significativa), mentre d_2, d_3 possono valere 0 o 1 e $-1 \leq p \leq 2$.

Per esempio:

$0.100 \cdot 10^{-1}$	$=$	$1/4_{10}$	$=$ 2_{10}^{-2}	il più piccolo positivo
$0.101 \cdot 10^{-1}$	$=$	$5/16_{10}$		
$0.111 \cdot 10^{10}$	$=$	$7/2_{10}$		il più grande positivo
0.000	$=$	0		eccezione

È possibile disegnare tutti i 16 numeri positivi (scritti qui in decimale):



Come si vede non sono egualmente spaziati, ma equidistanti tra 2^t e 2^{t+1}.

Come memoria occorono 6 bit: 3 bit per la mantissa, 1 bit per il segno e 2 bit per l'esponente che può assumere quattro valori: $00 = 0_{10}$; $01 = 1_{10}$; $10 = 2_{10}$; $11 = -1_{10}$, l'ultimo rappresentato nella forma "complementare".
Teniamo presente che la prima cifra della mantissa è necessariamente 1. Unica eccezione è il numero 0 che viene rappresentato con mantissa tutta nulla.

I numeri *a doppia precisione* che vengono usati da molti computer sono memorizzati con 8 bytes cioè con 64 bit. Solitamente la suddivisione dei bit è la seguente:
• 1 bit per il segno
• 11 bit per la caratteristica (che va quindi da $-2^{-10} + 1 = -1023$ a $2^{10} = 1024$)
• 6 bytes e mezzo per la mantissa che ha quindi al massimo 52 cifre significative.

In questo modo il massimo numero rappresentabile è $2^{1024} - 2^{971}$ che è circa 1.7×10^{308} e le 52 cifre binarie della mantissa consentono di rappresentare i numeri in base 10 con circa 16 cifre significative.
Sono rappresentabili in modo esatto tutti i numeri interi positivi fino a 2^{53}; il numero $2^{53} + 1$ è il primo intero positivo non rappresentabile esattamente.

1.4.2 Rappresentazione in macchina di un numero reale

Sia $x \in \mathbb{R}, x \neq 0$, quindi dotato di segno s, mantissa $m = d_1, d_2, \ldots$, caratteristica p. Ci sono 3 possibilità:

1. Si ha $L \leq p \leq U$ e $d_i = 0$ per $i > t$. Quindi la rappresentazione macchina di x è esatta.
2. (a) $p > U$: overflow. Di solito la macchina si arresta ed emette segnale di errore. Se questo non succede ciò comporta grave errore e perdita di dati.
   (b) $p < L$: underflow. A volte si può sostituire x con 0, ma più spesso questo comporta grave errore e perdita di dati.
   L'underflow non segnalato è spesso più grave dell'overflow perché sostituisce un numero, anche molto piccolo con 0, quindi l'errore è teoricamente infinito.
3. $L \leq p \leq U$, ma la successione della mantissa è più lunga di t (spesso infinita). per esempio $1/3_{10} = 0,010101 \cdots \cdot 2_{10}^{-1}$ o meglio $1/3_{10} = 0,10101 \cdots \cdot 2_{10}^{-2}$.
   Questo è di gran lunga il caso più frequente. La rappresentazione macchina di x non è perfetta, quindi bisogna sapere cosa si perde e in che modo.

1.4.3 Arrotondamento e troncamento

Siamo nel caso 3.
Per semplificare poniamo $x > 0$. Per rappresentare in macchina x ci sono due tecniche.

- *Troncamento*: si omettono nella mantissa tutte le cifre oltre la t-esima.
- *Arrotondamento*: si omettono nella mantissa tutte le cifre oltre la $t+1$-esima, ma, visto che al momento della memorizzazione le cifre devono essere t, si scrive come mantissa $d_1 \cdots d_{t+1} + \frac{1}{2} b^{-t}$ e si tronca

Esempio 1.10: Un tipico arrotondamento in base 10:

$\frac{1}{3} = 0.3333 \cdots$ troncamento e arrotondamento coincidono.

$\frac{2}{3} = 0.6666 \cdots$ il troncamento è $0.66666 \cdots 6$, l'arrotondamento è

$$\begin{array}{l} 0.666 \cdots 666|6 + \\ \underline{0.000 \cdots 000|5 =} \\ 0.666 \cdots 667|1 \end{array}$$

quindi $2/3 = 0.66 \cdots 667 \cdot 10^0$.

Esempio 1.11: Un arrotondamento in base 2:

Il numero decimale 3/7 ha la rappresentazione binaria infinita periodica $0.011\,011 = 0.\overline{011}$. Volendolo rappresentare con un numero finito di cifre binarie dopo il punto mediante arrotondamento si ottiene

con 3 cifre binarie	con 4 cifre binarie	con 5 cifre binarie
$0.011\|0 +$	$0.0110\|1 +$	$0.01101\|1 +$
$\underline{0.000\|1 =}$	$\underline{0.0000\|1 =}$	$\underline{0.00000\|1 =}$
$0.011\|1$	$0.0111\|1$	$0.01110\|0$
quindi 0.011	quindi 0.0111	quindi 0.01110

1.4.4 La precisione macchina

Poniamo la seguente

Definizione: Fissati i parametri $b, t, [L, U]$ e un metodo di rappresentazione tra troncamento e arrotondamento, si indica con

$$\mathrm{fl}(x)$$

la rappresentazione macchina del numero reale x

L'abbreviazione fl sta per *floating*. Vale la seguente:

Proposizione 3 *Se non c'è overflow, allora*

$$\left|\frac{\mathrm{fl}(x)-x}{x}\right| \le b^{1-t} \qquad \left|\frac{\mathrm{fl}(x)-x}{x}\right| \le \frac{1}{2}\, b^{1-t}$$

La prima in caso di troncamento, la seconda in caso di arrotondamento.

Il numero $\mathrm{eps} = \frac{1}{2}\, b^{1-t}$ (o b^{1-t} se si usa il troncamento) è detto *precisione macchina*.
Conviene però definirlo in modo indipendente dal metodo usato.

Definizione: E' detto eps il più piccolo numero tale che

$$\mathrm{fl}(1+\mathrm{eps}) > 1$$

Il numero eps non è il minimo numero rappresentabile in macchina, ma è il limite della precisione cui può arrivare la macchina, nel senso che è possibile chiedere alla macchina un errore relativo tra $\mathrm{fl}(x)$ e x fino a eps, ma non inferiore. Si ha infatti:

$$\left|\frac{\mathrm{fl}(x)-x}{x}\right| \le \mathrm{eps} \qquad \text{che si può scrivere} \qquad \mathrm{fl}(x) = x(1+\varepsilon) \qquad \text{dove} \qquad |\,\varepsilon\,| \le \mathrm{eps}$$

Esempio 1.12: $\sqrt{2} = 1.414\cdots$. Tronchiamo a due cifre dopo il punto. Allora $\left|\frac{\sqrt{2}-1.41}{\sqrt{2}}\right| \le \frac{1}{10^2}$

Sia $\pi = 3.14159\cdots$.
Se tronchiamo a 4 cifre dopo il punto $\left|\frac{\pi-3.1415}{\pi}\right| \le \frac{1}{10^4}$
Se invece arrotondiamo a 4 cifre dopo il punto $\left|\frac{\pi-3.1416}{\pi}\right| \le \frac{1}{2}\frac{1}{10^4}$

Esempio 1.13: Nel caso di numeri in doppia precisione si ha eps$= 2^{-52} \simeq 2.22\cdot 10^{-16}$

Una semplice routine per il calcolo di eps:

```
eps = 1
while 1+ eps > 1
        eps = eps /2
end
eps = eps *2
```

1.5 Errori macchina

1.5.1 Operazioni macchina

Date le 4 operazioni aritmetiche elementari, definiamo le corrispondenti operazioni macchina in questo modo:
$x \oplus y = \mathrm{fl}(x+y)$
$x \ominus y = \mathrm{fl}(x-y)$ etc.
Gli input x, y saranno o già numeri macchina o anche numeri ancora da convertire. In realtà le cose sono quindi un po' più complicate, perché la macchina potrebbe aver bisogno di convertire gli input x e y a numeri macchina prima di calcolare $x+y$ etc., ma per per quanto segue può bastare questa definizione.

Quindi, si ha per esempio $x \oplus y = (x+y)(1+\varepsilon) = x(1+\varepsilon) + y(1+\varepsilon)$ con $\varepsilon <$ eps.
Le quattro operazioni non godono sempre di proprietà elementari quali $(x \oplus y) \oplus z = x \oplus (y \oplus z)$. Possono succedere cose abbastanza strane tipo il fatto che $x \oplus y = x$ se $|\,y\,| < \frac{\mathrm{eps}}{b}\,|\,x\,|$, cioè se

sommo a x un numero al di là della precisione macchina in confronto a x. Per esempio in base 10 con quattro cifre decimali e range di esponenti abbastanza grande si ha: $1 + 0.00002 = 1$.
Il numero 0.00002 non è al di fuori dei numeri rappresentabili in macchina (è $0.2 \cdot 10^{-4}$), ma è troppo piccolo in confronto a 1 o meglio è oltre la precisione della macchina in confronto a 1.
Questo esempio introduce il fenomeno della *cancellazione*.

1.5.2 La cancellazione

La cancellazione è tra le più frequenti sorgenti di errore nelle operazioni macchina.

Esempio 1.14: Lavoriamo in base $b = 10$ con $t = 8$ cifre. Siano

$a = 0.23371258 \cdot 10^{-4}$ $b = 0.33678429 \cdot 10^{2}$ $c = -0.33677811 \cdot 10^{2}$

Calcoliamo
$(a \oplus b) \oplus c = 0.33678452 \cdot 10^{2} \ominus 0.33677811 \cdot 10^{2} = 0.6410000 \cdot 10^{-3}$
$a \oplus (b \oplus c) = 0.23371258 \cdot 10^{-4} \oplus 0.6180000 \cdot 10^{-3} = 0.64137126 \cdot 10^{-3}$
Il risultato esatto è $a + b + c = 0.64137126 \cdot 10^{-3}$.

Nel primo conto la cancellazione è avvenuta alla seconda somma e il primo addendo aveva già subito arrotondamento, quindi con perdita di dati, nel secondo caso la cancellazione è avvenuta subito tra due numeri vicini e quindi con minore perdita di dati. Meglio quindi prima sommare i due numeri di grandezza simile.

Un altro esempio famoso

Esempio 1.15: Scriviamo il noto sviluppo di MacLaurin $e^x = 1 + x + \dfrac{x^2}{2!} + \dfrac{x^3}{3!} + \cdots$ e usiamolo per calcolare e^{-30}: $e^{-30} = 1 - 30 + \dfrac{900}{2} - \dfrac{27000}{6} + \cdots$
Nella sommatoria ci sono numeri di diverso ordine di grandezza, per cui si verifica la cancellazione. meglio e calcolare nel seguente modo;
$e^{30} = 1 + 30 + \dfrac{900}{2} + \dfrac{27000}{6} + \cdots$ e poi eseguire $e^{-30} = 1/e^{30}$. Calcolando con un computer ci si può render conto di come il primo metodo porti a gravi errori di conto dopo un certo numero di passi.

Cerchiamo di spiegare in modo teorico il fenomeno della cancellazione:
Siano a, b due numeri reali e poniamo $\tilde{a} = \mathrm{fl}(a)$, $\tilde{b} = \mathrm{fl}(b)$. Si ha

$$\tilde{a} = a(1+\varepsilon_1) \qquad \tilde{b} = b(1+\varepsilon_2) \qquad \tilde{a} \oplus \tilde{b} = (\tilde{a} + \tilde{b})(1+\varepsilon) \qquad \text{con } \mid \varepsilon, \varepsilon_1, \varepsilon_2 \mid < \text{eps}$$

Vogliamo calcolare δ, errore relativo tra $\tilde{a} \oplus \tilde{b}$ e $a + b$, cioè $\delta = \dfrac{(\tilde{a} \oplus \tilde{b}) - (a+b)}{a+b}$

Si ha: $\delta = \dfrac{(\tilde{a} + \tilde{b})(1+\varepsilon) - (a+b)}{a+b} = \cdots = \varepsilon + \left(\dfrac{a\varepsilon_1 + b\varepsilon_2}{a+b}\right)(1+\varepsilon)$ con $\mid \varepsilon, \varepsilon_1, \varepsilon_2 \mid <$eps

Quindi $\mid \delta \mid < \text{eps} + (1 + \text{eps})\,\text{eps}\, \dfrac{\mid a \mid + \mid b \mid}{\mid a + b \mid}$

Questo spiega il fenomeno della cancellazione che si verifica quando a e b sono di segno discorde, ma molto prossimi in valore assoluto, perché $\mid a+b \mid$ può essere molto piccolo rendendo $\mid \delta \mid$ grande.

Esempio 1.16: $a = 0.123456$ $b = -0.123454$. Se $t = 5$ (numero di cifre decimali), allora

$\tilde{a} = 0.12346$ $\tilde{b} = -0.12345$ $a + b = 0.2 \cdot 10^{-5}$ $\tilde{a} \oplus \tilde{b} = 0.1 \cdot 10^{-4}$

Quindi $\delta = \dfrac{(\tilde{a} \oplus \tilde{b}) - (a+b)}{a+b} = 4$

Esempio 1.17: Equazione di secondo grado. Sia $ax^2 - 2bx + c = 0$ una semplice equazione di grado 2, allora le soluzioni sono notoriamente

$$x_1 = \frac{b - \sqrt{b^2 - 4ac}}{a} \qquad x_2 = \frac{b + \sqrt{b^2 - 4ac}}{a}$$

Supponiamo che $b > 0$. Se c è prossimo a 0, allora in x_1 c'è una cancellazione, per cui l'errore può essere anche elevato. Conviene allora calcolare prima x_2 che non ha cancellazione e poi porre $x_1 = \dfrac{c}{b + \sqrt{b^2 - 4ac}}$, ovvero $x_1 = x_2 \cdot a \cdot c$.

1.5.3 L'errore nelle operazioni macchina

Abbiamo già visto che, se $\tilde{a} = a(1 + \varepsilon_1) \qquad \tilde{b} = b(1 + \varepsilon_2)$, nella somma l'errore è inferiore a $\text{eps} + (1 + \text{eps})\,\text{eps}\,\dfrac{|\,a\,| + |\,b\,|}{|\,a + b\,|}$.

Nel prodotto si vede che

$$\delta = \frac{\tilde{a}\tilde{b} - ab}{ab} = (1 + \varepsilon_1)(1 + \varepsilon_2) - 1 = \varepsilon_1 + \varepsilon_2 + \varepsilon_1\varepsilon_2 \simeq \varepsilon_1 + \varepsilon_2$$

Un conto analogo per la divisione mostra che

$$\delta = \frac{\tilde{a}/\tilde{b} - a/b}{a/b} = \frac{\varepsilon_1 - \varepsilon_2}{1 + \varepsilon_2} \simeq \varepsilon_1 - \varepsilon_2$$

La conclusione è che nella somma e nel prodotto c'è più controllo nell'errore rispetto a quanto avviene con le somme dove si può verificare il fenomeno della cancellazione.

Un'altra osservazione è quella che se uno dei due fattori è noto con una certa incertezza, non si può sperare di migliorare la precisione cercando di conoscere con migliore precisione l'altro fattore.

2.1 Equazioni non lineari

Capita sovente di dover risolvere un'equazione in un'incognita

$$f(x) = 0$$

dove $f(x)$ è un qualche funzione più o meno semplice.
Se in qualche modo è noto che il problema ha una soluzione in un qualche sottoinsieme di $\mathbb{R}$, esistono vari modi di determinarla con una certa precisione.

2.1.1 Il metodo di bisezione

Il più semplice algoritmo è quello ben noto di bisezione.

Se $f(x)$ è continua in un intervallo $[a, b]$ e $f(a) \cdot f(b) < 0$ (ovvero assume valori di segno discorde negli estremi), allora, per un noto teorema (quello degli zeri), nell'intervallo $[a, b]$ esiste almeno un x_0 tale che $f(x_0) = 0$.

Per approssimarlo si divide l'intervallo a metà: $a\,,\ \dfrac{a+b}{2}\,,\ b$ e si sostituisce l'intervallo $[a, b]$ con quello tra i due intervalli $\left[a\,,\ \dfrac{a+b}{2}\right]$ e $\left[\dfrac{a+b}{2}\,,\ b\right]$ dove la funzione ha ancora valori discordi negli estremi e così via.

L'algoritmo ha la caratteristica di non avere (quasi mai) termine, per cui occorre fermarlo a un certo punto con un qualche criterio. Il criterio di solito è uno di questi tre

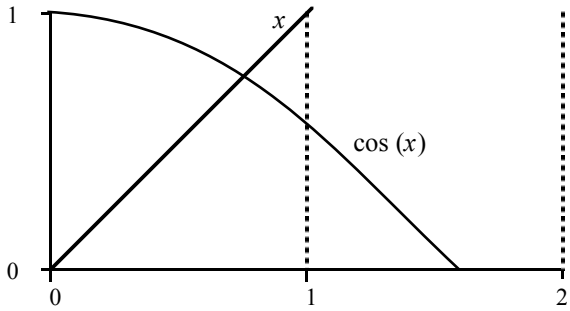
- Dopo un certo numero p di passi.
- Quando $\mid \tilde{x} - x_0 \mid < \varepsilon$ con ε valore prefissato ($\tilde{x}$ il valore trovato in quel momento).
- Quando $f(\tilde{x}) < \varepsilon$ con ε valore prefissato.

Tra i primi due criteri c'è una relazione: $\varepsilon < \dfrac{\mid b-a \mid}{2^{p+1}}$, quindi $p < 1 + \log_2 \dfrac{\mid b-a \mid}{\varepsilon}$.
Invece per quanto riguarda il terzo è a priori impossibile prevedere il numero di passi, a meno di non avere informazioni sulla derivata di $f'(x)$ (se esiste).
L'algoritmo è piuttosto lento, dato che $\log_2(10) \simeq 3.32$, quindi i passi per avere molte cifre decimali possono essere parecchi, in compenso è abbastanza sicuro e di facilissima implementazione.

2.1.2 L'algoritmo di punto fisso

Cominciamo con un semplice esempio facilmente eseguibile con calcolatrice scientifica tascabile.

Esempio 2.1: Risolvere l'equazione $x = \cos(x)$ ($\cos(x)$ in radianti !).
Dalla figura si può dedurre che $x_0 \simeq 0.7$, e quindi calcoliamo $\cos(0.7) = 0.7648$, e di seguito calcoliamo $\cos(0.7648)$ e così via



$$\begin{aligned}
\cos(0.7) &= 0.7648\\
\cos(0.7648) &= 0.7215\\
\cos(0.7215) &= 0.7508\\
\cos(0.7508) &= 0.7311\\
\cos(0.7311) &= 0.7444\\
\cos(0.7444) &= 0.7355\\
&\text{venti volte...}\\
\cos(0.7391) &= 0.7391
\end{aligned}$$

Quindi dopo una ventina di passi si trovano in modo elementare quattro cifre decimali esatte della soluzione del problema.

Definizione: Si dice che α è un punto fisso della funzione $f(x)$ se $f(\alpha) = \alpha$

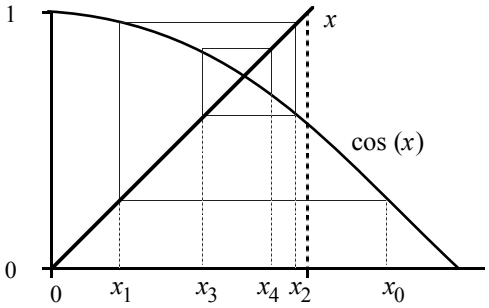
Nell'esempio $\alpha = 0.7391$ è un punto fisso di $\cos(x)$ nell'intervallo $[0, 1]$ e l'algoritmo consente di determinarlo facilmente.
Benché sembri un caso particolare del problema iniziale di risolvere un'equazione $f(x) = 0$, l'algoritmo di punto fisso è alla base di molti altri metodi.

Proposizione 4 *Sia $g(x)$ continua in $[a, b]$ e sia $x_0 \in [a, b]$ e $x_0, x_1, \ldots$ la successione determinata dall'algoritmo di punto fisso, cioè $\forall i\ x_i = g(x_{i-1})$. Se la successione converge e $\lim_{i\to\infty} x_i = \alpha$, allora α è punto fisso di $g(x)$.*

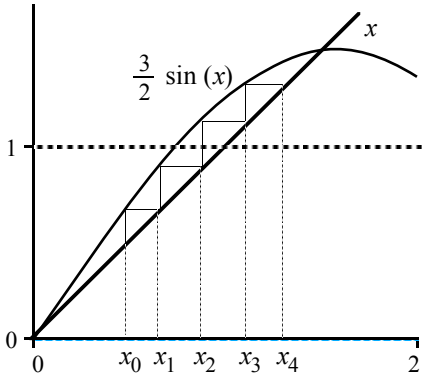
Non sempre l'algoritmo converge all'eventuale punto fisso, però basta una condizione sulla derivata di $g(x)$:

Proposizione 5 *(condizione sufficiente) Sia α è un punto fisso di $g(x)$ e supponiamo che $g(x)$ sia derivabile in un intervallo $I = [\alpha - \varrho\ ,\ \alpha + \varrho]$ con $\varrho > 0$. Se $\mid g'(x) \mid < 1$ e $x_0 \in I$, allora la successione $x_i = g(x_{i-1})$ determinata dall'algoritmo di punto fisso converge e $\lim_{i\to\infty} x_i = \alpha$. Inoltre α è unico nell'intervallo.*

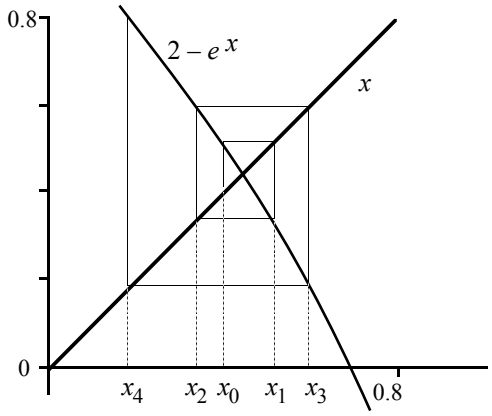
Graficamente:



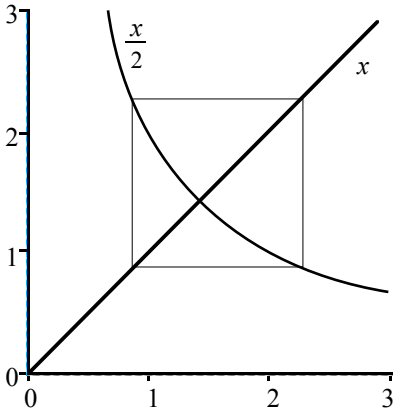
Equazione $x = \cos(x)$ come sopra, ma partendo con un x_0 più lontano per chiarezza; la successione converge a segno alterno ad α.



Equazione $x = (3/2) \cdot \sin(x)$ partendo con un x_0 lontano da α per chiarezza; la successione è crescente e converge ad α.



Equazione $x = 2 - e^x$ che diverge anche partendo da x_0 prossimo ad α perché la derivata è in modulo maggiore di 1.



Equazione $x^2 = 2$ che può essere pensata come punto fisso di $f(x) = 2/x$, ma la successione è stabile e non converge perché la derivata in α è proprio -1.

2.1.3 Velocità di convergenza dell'algoritmo di punto fisso

Vediamo quanto deve durare l'algoritmo:
Il criterio di solito è uno di questi tre

- Quando $\mid x_i - x_{i+1} \mid < \varepsilon$ con j prefissato. Può essere un criterio assai poco valido quando $\mid g'(x) \mid$ è prossimo a 1 e quindi l'algoritmo è lento, perché può capitare che $\mid x_i - x_{i+1} \mid$ sia assai piccolo pur essendo lontani da α.

- Quando $\dfrac{\mid x_i - x_{i+1} \mid}{\min\{\mid x_i \mid, \mid x_{i+1} \mid\}} < \varepsilon$ con ε valore prefissato. Come criterio è più affidabile, come adesso vedremo.

- Quando $f(x_i) < \varepsilon$ con ε valore prefissato.

Osserviamo ora (vedi anche gli esempi precedenti) che il segno della derivata di $g(x)$ consente di stabilire in che modo la successione x_i converge.
Quando $g'(x)$ è positivo e quindi $0 < g'(x) < 1$, la successione è *monotona*; quando $g'(x)$ è negativo e quindi $-1 < g'(x) < 0$, la successione è *alternante*. In quest'ultimo caso è più facile valutare quanto si è lontani da α.

Per capire come vadano le cose, applichiamo il noto teorema di Lagrange all'intervallo $[x_{i-1}, \alpha]$ (o all'intervallo $[\alpha, x_{i-1}]$):

$$\frac{g(x_{i-1}) - g(\alpha)}{x_{i-1} - \alpha} = g'(\xi) \qquad \text{con } \mid \xi - \alpha \mid < \mid x_{i-1} - \alpha \mid$$

Quindi, dato che $g(\alpha) = \alpha$ e $g(x_{i-1}) = x_i$

$x_i - \alpha = g'(\xi)(x_{i-1} - \alpha)$ da cui $x_i - x_{i-1} = (x_i - \alpha) - (x_{i-1} - \alpha) = (x_{i-1} - \alpha)(g'(\xi) - 1)$

In conclusione

$$\mid x_{i-1} - \alpha \mid = \frac{\mid x_i - x_{i-1} \mid}{\mid g'(\xi) - 1 \mid}$$

Col primo criterio di arresto si ha:

$$\mid x_{i-1} - \alpha \mid < \frac{\varepsilon}{\mid g'(\xi) - 1 \mid} \qquad \text{Se } g'(\xi) < 0, \text{ allora} \qquad \mid x_{i-1} - \alpha \mid < \varepsilon$$

Col secondo criterio di arresto si ha:

$$\frac{\mid x_{i-1} - \alpha \mid}{\mid \alpha \mid} < \frac{\varepsilon \cdot \min\{\mid x_i \mid, \mid x_{i+1} \mid\}}{\mid \alpha \mid \mid g'(\xi) - 1 \mid}$$

Se $g'(\xi) < 0$, allora $\min\{\mid x_i \mid, \mid x_{i+1} \mid\} \simeq \alpha$, quindi l'ultima espressione è circa ε.

Se $g'(x) > 0$ occorre conoscere almeno approssimativamente il valore di $g'(x)$ in un intorno di α per valutare la distanza assoluta o relativa di x_i da α.
Se $\mid g'(x) \mid \simeq 1$, questo significa che $g(x)$ è quasi tangente alla retta $y = x$.

2.1.4 Ordine di convergenza dell'algoritmo di punto fisso

Definizione: Se in un algoritmo di punto fisso si ha $\lim_{i\to\infty}(x_i) = \alpha$ (e $x_i \neq \alpha \ \forall i$), allora il numero $\gamma = \lim_{i\to\infty} \frac{\mid x_{i+1} - \alpha \mid}{\mid x_i - \alpha \mid}$ è detto fattore di convergenza

Si ha sempre $\gamma \leq 1$.
Se $0 < \gamma < 1$ si dice che la convergenza è *lineare* (caso normale)
Se $\gamma = 1$ si dice che la convergenza è *sublineare* (caso lento)
Se $\gamma = 0$ si dice che la convergenza è *superlineare* (caso veloce)

Definizione: Nelle ipotesi precedenti, se esiste p, $p \geq 1$ tale che $\lim_{i\to\infty} \frac{\mid x_{i+1} - \alpha \mid}{\mid x_i - \alpha \mid^p}$ si dice che p è l'ordine di convergenza

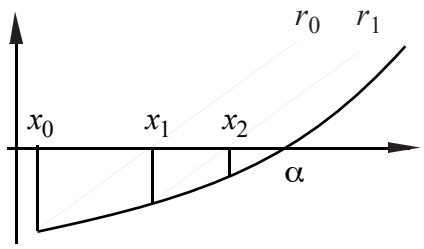
Un ordine di convergenza maggiore di 1 implica velocità di convergenza alta. L'ordine di convergenza è strettamente legato alla derivata prima e alle successive:

Proposizione 6 *Nelle ipotesi precedenti, supponiamo che $g(x)$ sia di classe C^p in $[\alpha - \varrho\,,\ \alpha + \varrho]$ e $x_0 \in [\alpha - \varrho\,,\ \alpha + \varrho]$ e che l'ordine di convergenza della successione di punto fisso sia p, allora:*

$$g'(\alpha) = g''(\alpha) = \cdots = g^{(p-1)}(\alpha) = 0 \qquad g^{(p)}(\alpha) \neq 0$$

2.1.5 Riduzione di un'equazione ad algoritmo di punto fisso

In generale l'equazione $f(x) = 0$ può essere trasformata in vari modi in un problema di punto fisso:



Sia α uno zero di $f(x)$. Scegliamo un punto x_0 prossimo ad α e consideriamo la retta r_0 con coefficiente angolare h passante per $(x_0, f(x_0))$ con h scelto in qualche modo.
La retta è $r_0 :\ y - f(x_0) = h(x - x_0)$
L'intersezione tra la retta r_0 e l'asse x ha ascissa $x_1 = x_0 - \frac{f(x_0)}{h}$.

Proseguiamo con la retta $r_1 : \; y - f(x_1) = h(x - x_1)$.

L'intersezione tra la retta r_0 e l'asse x ha ascissa $x_2 = x_1 - \dfrac{f(x_1)}{h}$

In pratica stiamo cercando il punto fisso della funzione $g(x) = x - \dfrac{f(x)}{h}$.

Naturalmente non è detto che l'algoritmo converga ad α, ma in molti casi, attraverso un'opportuna scelta di h è possibile riuscirci. In generale, anche h non andrà scelto costante, ma verrà fatto variare in funzione dell'x via via trovato. Quindi dobbiamo cercare di studiare la convergenza dell'algoritmo di punto fisso della funzione

$$g(x) = x - \frac{f(x)}{h(x)}$$

con $h(x)$ scelta opportunamente in modo che $\mid g'(x) \mid < 1$.

A seconda della scelta di $h(x)$ si ottengono vari algoritmi. I più noti sono quelli delle corde, delle tangenti, delle secanti e quello della falsa posizione.

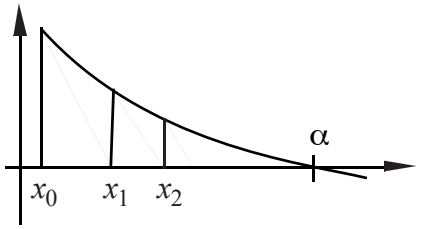
2.1.6 Metodo delle corde

È il più semplice ed è quello con la scelta $h(x) = m$ (m inclinazione costante).

Quindi si cerca il punto fisso della funzione $g(x) = x - \dfrac{f(x)}{m}$, ovvero l'algoritmo è $x_{i+1} = x_i - \dfrac{f(x_i)}{m}$.

Affinché l'algoritmo converga occorre che $\mid g'(x) \mid = \left| 1 - \dfrac{f'(x)}{m} \right| < 1$. La diseguaglianza equivale alle tre condizioni:

- $f'(x) \neq 0$
- $f'(x) \cdot m > 0$ (devono avere lo stesso segno)
- $\mid m \mid > \dfrac{1}{2} \max\{f'(x)\}$ (in un intorno di α)



Il metodo delle corde, se converge, converge di ordine 1.

2.1.7 Metodo delle tangenti

Detto anche metodo di Newton-Raphson è sicuramente il più noto e presuppone il calcolo di $f'(x)$. Infatti come h si usa il valore della derivata nel punto, cioè la retta tangente al grafico. In pratica $h(x) = f'(x)$

L'algoritmo consiste nel determinare un punto fisso della funzione $g(x) = x - \dfrac{f(x)}{f'(x)}$, quindi la successione x_i è così definita: $\quad x_{i+1} = x_i - \dfrac{f(x_i)}{f'(x_i)}$.

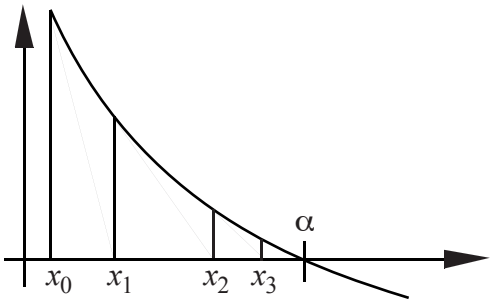
Affinché l'algoritmo converga occorre che $\mid g'(x) \mid = \left| \dfrac{f(x)f''(x)}{(f'(x))^2} \right| < 1$.

Non è facile verificare direttamente la diseguaglianza, quindi si ricorre a criteri sufficienti.
Il più noto è:

Proposizione 7 *Supponiamo che $f(x)$ sia di classe C^2 in $I = [\alpha \, , \, \alpha + \varrho]$ (o in $I = [\alpha - \varrho \, , \, \alpha]$)e $x_0 \in I$.*
Se nell'intervallo si ha $f(x)f''(x) > 0$ e $f'(x) \neq 0$ allora l'algoritmo converge.

Graficamente la situazione è quella a lato.
È chiaro che nella situazione disegnata l'algoritmo converge partendo da x_0 a sinistra perché la $f(x)$ è positiva e così pure $f''(x)$, mentre l'algoritmo non converge necessariamente partendo da x_0 a destra perché $f(x)$ è negativa e $f''(x) > 0$

Il metodo delle tangenti, se converge, converge di ordine 2 o superiore, quindi, quando è applicabile, è uno dei più veloci.



Esempio 2.2: Come esperimento, provare a calcolare $\sqrt{2}$ cercando lo zero positivo della funzione $x^2 - 2$. Basta eseguire l'algoritmo di punto fisso sulla funzione $g(x) = x - \dfrac{x^2-2}{2x} = \dfrac{x^2+2}{2x}$.
Se si parte per esempio da $x_0 = 2$ (o da qualunque $x_0 > 2$) converge perché soddisfa le condizioni sufficienti.
Se si parte da $x_0 = 1$ converge ugualmente, perché dopo il primo passo si trova $x_1 = 3/2$ e si rientra nelle condizioni sufficienti della proposizione.
Se si inizia invece con $x_0 < 0$ l'algoritmo non converge a $\sqrt{2}$.

2.1.8 Metodo delle secanti

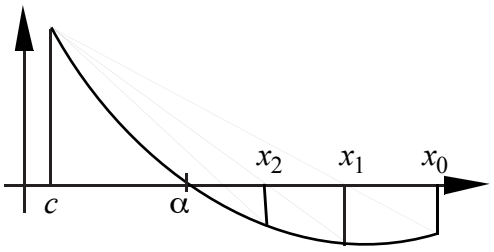
Sia α lo zero da cercare; fissiamo $x = c$ prossimo ad α e scegliamo x_0 il valore di partenza dell'algoritmo in modo che α sia compreso tra c e x_0.
Consideriamo la retta congiungente i due punti $(c, f(c))$ $(x_0, f(x_0))$. L'intersezione tra la retta e l'asse x è il nuovo punto x_1.
L'algoritmo è $x_{i+1} = x_i - \dfrac{f(x_i)(x_i - c)}{f(x_i) - f(c)}$. Quindi come funzione h si ha $h(x) = \dfrac{f(x) - f(c)}{x - c}$.
La funzione di cui trovare il punto fisso è

$$g(x) = \frac{c \cdot f(x) - x \cdot f(c)}{f(x) - f(c)} \qquad \text{e si ha} \qquad g'(x) = f(c)\frac{f'(x)(x-c) - f(x) + f(c)}{(f(x) - f(c))^2}$$

Una condizione sufficiente per la convergenza è $\left|\dfrac{f(c)}{c-\alpha}\right| > \dfrac{1}{2} \mid f'(\alpha) \mid$. Come per le tangenti si ha:

Proposizione 8 *Supponiamo che la funzione $f(x)$ definita nell'intervallo $I = [a, b]$ sia di classe C^2 e si abbia $f'(x), f''(x) \neq 0$.*
Se si scelgono nell'intervallo c, x_0 tali che $f(c) \cdot f''(c) \geq 0$ e inoltre $f(x_0) \cdot f''(x_0) \leq 0$, allora l'algoritmo delle secanti converge (monotonamente).

L'algoritmo delle secanti è talvolta preferito a quello delle tangenti, anche se la convergenza è di ordine 1, perché la funzione di cui calcolare il punto fisso può essere più semplice, non prevedendo il calcolo della derivata di $f(x)$. Inoltre l'algoritmo delle secanti è la premessa al metodo seguente.

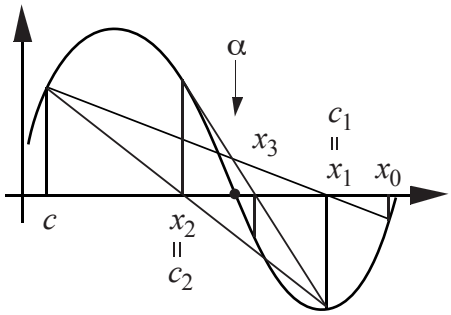


2.1.9 Metodo della falsa posizione (regula falsi)

Come nel metodo delle secanti fissa un punto c prossimo allo zero da cercare α e si scrive la retta congiungente i due punti $(c, f(c))$ $-$ $(x_0, f(x_0))$. Però ci si riserva di cambiare il punto c, quando sia necessario, se le condizioni della proposizione non sono più verificate. Nella fattispecie,

se x_{i+1} è tale che $f(x_{i+1}) \cdot f(c) > 0$, allora si pone $c = x_i$ e si prosegue l'algoritmo con il nuovo c. L'algoritmo delle secanti modificato con la regula falsi, ha convergenza di ordine 1 e ha il pregio di convergere sempre, nella sola ipotesi che $f(x)$ sia continua.

Come si vede nell'esempio, si comincia con c e x_0 tra cui è compreso α e si trova x_1.
Poi si continua con x_1 e c e si trova x_2. A questo punto $f(x_2)$ e $f(c)$ sono concordi, perciò α non è più compreso tra c e x_i .
Si sostituisce c con $c_1 = x_1$ e si prosegue con x_2 e c_1.
Si trova x_3 e, dato che $f(x_3)$ e $f(c_1)$ sono concordi, si deve di nuovo porre $c_2 = x_2$, dopodiché l'algoritmo dovrebbe procedere senza più cambiamenti.

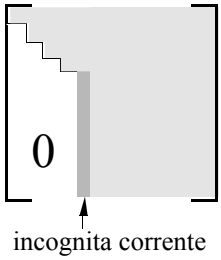


3.1 Algebra lineare numerica

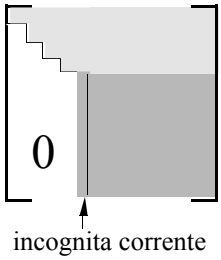
3.1.1 Le varianti dell'algoritmo di Gauss

Dato un sistema lineare *quadrato* $Ax = b$ con A matrice invertibile (c'è sempre il problema di scoprire se lo sia), vediamo quali sono i metodi di risoluzione. L'algoritmo di Gauss è il metodo base, ma ha parecchie varianti. Elenchiamo le principali varianti:

1. **Pivotizzazione parziale**: L'algoritmo di Gauss prevede la ricerca di un pivot per ogni incognita.
   Dopo alcuni passi dell'algoritmo di Gauss la matrice è parzialmente a scala. Il pivot va cercato nella zona grigio scuro tra i coefficienti della incognita corrente. La *pivotizzazione parziale* prevede che tra i possibili pivot si scelga sempre quello di valore assoluto più alto e poi si faccia uno scambio di righe per usarlo come pivot. La scelta di un pivot di valore assoluto alto riduce l'impatto degli inevitabili errori di arrotondamento.
   Non è facile dare una spiegazione di questo fatto, ma si può averne un'intuizione dal fatto che, se un pivot nullo è improponibile, un pivot piccolo è comunque sconsigliato.

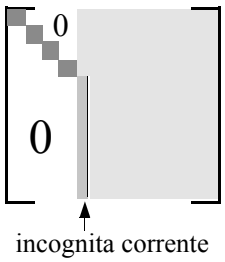


2. **Pivotizzazione totale**: Questa strategia prevede la ricerca di un pivot non solo tra i coefficienti dell'incognita su cui si sta lavorando, ma anche tra i coefficienti delle incognite successive (la zona grigio scuro della figura)
   Se viene trovato un buon pivot in un'altra incognita, si scambiano tra loro le incognite e poi si procede secondo l'algoritmo classico. Naturalmente si dovrà tenere conto di questi scambi al momento di scrivere il risultato finale, cioè la n-upla delle soluzioni.
   La ricerca di un pivot lungo tutta la matrice e non solo in una colonna può richiedere più tempo contro un vantaggio non sempre reale, quindi il metodo della pivotizzazione totale è scarsamente usato, mentre la pivotizzazione parziale è in pratica lo standard nell'algoritmo gaussiano.



3. **Algoritmo di Gauss-Jordan**: Usando l'algoritmo classico di Gauss, si produce una matrice ridotta, dopodiché occorre l'algoritmo retrogrado ovvero la sostituzione all'indietro per risolvere il sistema.
   La variante di Jordan dell'algoritmo gaussiano invece riduce immediatamente in modo totale la matrice.

Cioè il pivot viene usato non solo per annullare i coefficienti della sua colonna situati nelle righe inferiori, ma anche quelli situati nelle righe sopra. Nella figura in scuro i pivot già usati. L'algoritmo di Gauss-Jordan venne usato ai primordi del calcolo, perché riducendo immediatamente tutta la matrice, permetteva di liberare la memoria del computer dai dati delle colonne già ridotte. Oggi è meno usato, dato che comporta un tempo leggermente superiore all'algoritmo classico di Gauss, mentre la quantità di memoria disponibile non è più un problema.



4. **La fattorizzazione LU**: Per risolvere il sistema $Ax = b$ l'algoritmo classico di Gauss prevede che si riduca la matrice $[A \mid b]$.
   La fattorizzazione LU, che non descriviamo in dettaglio, prevede invece che si riduca solo la matrice A ottenendo quindi una matrice U triangolare superiore (U sta per "upper triangular"). Le operazioni elementari eseguite vengono memorizzate in una matrice (quasi) triangolare inferiore L. Il costo di questa operazione è praticamente nullo perché non richiede operazioni aritmetiche, ma solo spazio in memoria. Non stiamo qui a descrivere in dettaglio la costruzione di L, diciamo solo che tra le matrici A, L, U c'è la relazione $A = LU$, per cui si parla di fattorizzazione LU
   Vediamo ora come si usa la decomposizione $A = L \cdot U$ per risolvere il sistema $A \cdot x = b$.
   Il sistema diventa $L \cdot U \cdot x = b$.
   Risolviamo il sistema lineare $L \cdot t = b$. Dato che L è (quasi) triangolare inferiore, la soluzione si determina facilmente con una variante dell'algoritmo retrogrado di Gauss che consiste semplicemente nel partire dalla prima equazione e prima incognita anziché dall'ultima.
   Sia quindi b_1 la soluzione del sistema $L \cdot t = b$, ovvero $L \cdot b_1 = b$
   Il sistema originale $L \cdot U \cdot x = b$ si scrive $L \cdot U \cdot x = L \cdot b_1$ ed è equivalente al sistema ridotto $U \cdot x = b_1$ che si risolve con la sostituzione all'indietro.
   La x trovata è la soluzione del sistema originale.
   In pratica, una volta ridotta A, si trova la nuova matrice dei termini noti semplicemente risolvendo $L \cdot t = b$.
   Questo metodo, nonostante l'apparenza più macchinosa, è in realtà una variante dell'algoritmo gaussiano che richiede un numero di operazioni aritmetiche (somme, prodotti e divisioni) uguale a quello della riduzione totale di $[A \mid b]$ attraverso l'algoritmo gaussiano classico. Il grosso vantaggio di questo metodo sta però nel fatto che, una volta individuata la fattorizzazione $L \cdot U$ della matrice A, qualunque sistema avente A come matrice dei coefficienti si risolve in tempo brevissimo nel modo descritto.

5. **La matrice inversa e il metodo di Cramer**: Risolvere il sistema $Ax = b$ scrivendo $x = A^{-1}b$ è lecito, ma non conveniente. Infatti, mentre l'algoritmo di Gauss per ridurre A richiede *circa* $n^3/3$ prodotti, per calcolare l'inversa mediante l'algoritmo di Gauss occorrono invece circa n^3 prodotti.

   Nella soluzione dei sistemi lineari, non conviene quindi determinare l'inversa della matrice dei coefficienti, ma usare metodi tipo la fattorizzazione LU.

   La riduzione retrograda a partire dalla matrice ridotta U richiede un numero di prodotti dell'ordine di $n^2/2$, numero trascurabile, rispetto alle operazioni richieste per la riduzione della matrice.
   Del tutto da evitare in generale è la nota regola di Cramer.
   La regola dice che $x_i = \det(A_i)/\det(A)$ dove con A_i si indica la matrice ottenuta sostituendo in A la colonna C_i con la colonna b.
   Quindi la regola di Cramer richiede il calcolo di $n+1$ determinanti, ciascuno dei quali richiede circa $n^3/3$ prodotti.

6. **I metodi iterativi**: Assomigliano un po' agli algoritmi di punto fisso. Si parte da una stima della soluzione e, attraverso un algoritmo se ne trova (se converge) una stima più prossima. Non si trova praticamente mai la soluzione esatta (d'altra parte anche con l'algoritmo di

Gauss non la si ottiene mai, causa gli arrotondamenti), ma hanno certi tipi di vantaggi come vedremo più avanti.

3.1.2 Il condizionamento

Sia A una matrice invertibile. Consideriamo il sistema lineare $Au = b$ (con $b \neq 0$) e sia x la sua soluzione.
Consideriamo poi il sistema $Au = b + \delta b$ in cui il termine noto b ha subito una "piccola" perturbazione δb e sia $x+\delta x$ la sua soluzione. Ci proponiamo di studiare quanto sia piccola la perturbazione δx subita dalla soluzione del sistema.
Occorre misurare la grandezza di δb e di δx. Il modo più usato di misurare un vettore è la norma euclidea:

$$\text{Se } v = (x_1, \ldots, x_n): \qquad \| v \| = \sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}$$

La norma ha tre proprietà

1. $\| u + x \| \leq \| u \| + \| x \|$
2. $\| \lambda x \| = | \lambda | \, \| x \|$
3. $\| x \| \geq 0$ e $\| x \| = 0$ se e solo se $x = 0$

Avvertiamo che esistono altri modi di misurare la norma, o meglio altre norme, a volte più convenienti, comunque ci limitiamo alla norma euclidea.

Teniamo ora presente il fatto che è importante non tanto conoscere la norma della perturbazione $\| \delta x \|$ subita da x, quanto il rapporto $\dfrac{\| \delta x \|}{\| x \|}$, cioè la misura relativa della perturbazione e che questo rapporto va confrontato con quello analogo per b: $\dfrac{\| \delta b \|}{\| b \|}$.
Cioè δb è piccolo se lo è $\| \delta b \| / \| b \|$ e così per δx.

Consideriamo le due eguaglianze

$$Ax = b \qquad A(x + \delta x) = b + \delta b \qquad \text{da cui} \qquad A\,\delta x = \delta b$$

Pertanto $\| Ax \| = \| b \|$ $\| A\delta x \| = \| \delta b \|$.
Per confrontare $\| \delta b \| / \| b \|$ con $\| \delta x \| / \| x \|$ occorre quindi conoscere $\| Ax \|$ e $\| A\delta x \|$.
Ovviamente la norma dei vettori Ax e $A\delta x$ dipende dalla matrice A.
Poniamo quindi la seguente definizione.

Definizione: La *norma matriciale* di una matrice quadrata invertibile $A \in M_{nn}(\mathbb{R})$ è

$$\| A \| = \max_{x \neq 0} \frac{\| Ax \|}{\| x \|}$$

al variare di x in $\mathbb{R}^n$ (x è sempre un vettore colonna).

Dalla definizione è immediato che $\| Ax \| \leq \| A \| \cdot \| x \|$ e questo ci consente di procedere nel nostro problema.
Rimane il problema di calcolare $\| A \|$ dato che non si può ovviamente usare direttamente la definizione per calcolare la norma di una matrice. Questo è un fatto non banale di cui ci occupiamo successivamente.
Si ha:

$$\| b \| = \| Ax \| \leq \| A \| \, \| x \|$$

Per il confronto tra δb e δx conviene però usare A^{-1}: $\qquad A\delta x = \delta b \qquad \delta x = A^{-1}\delta b$

$$\| \delta x \| = \| A^{-1}\delta b \| \leq \| A^{-1} \| \, \| \delta b \|$$

Le due relazioni tra le norme $\| b \| \leq \| A \| \| x \|$ $\| \delta x \| \leq \| A^{-1} \| \| \delta b \|$ si possono scrivere:

$$\frac{1}{\| x \|} \leq \frac{\| A \|}{\| b \|} \qquad \| \delta x \| \leq \| A^{-1} \| \| \delta b \|$$

In definitiva:

$$\frac{\| \delta x \|}{\| x \|} \leq \| A \| \| A^{-1} \| \frac{\| \delta b \|}{\| b \|}$$

è la relazione cercata.

Quindi, se $\| \delta b \| / \| b \|$ è piccolo e anche il numero $\| A \| \| A^{-1} \|$ lo è, allora $\| \delta x \| / \| x \|$ rimane piccolo. Se invece il numero $\| A \| \| A^{-1} \|$ è grande, a fronte di una piccola perturbazione di b si può verificare una grossa perturbazione di x.

Si pone la definizione

Definizione: Se A è una matrice quadrata e $\| A \|$ è la sua norma matriciale, il numero $\text{cond}(A) = \| A \| \| A^{-1} \|$ si dice *numero di condizionamento* di A.

Rimane il problema di calcolare $\text{cond}(A)$. Prima però esaminiamo un esempio.

Esempio 3.1: Siano $A = \begin{pmatrix} -1 & 2 & 2 \\ 2 & 1 & 3 \\ 2 & 3 & 6 \end{pmatrix}$ $b = \begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix}$. La soluzione del sistema lineare $Ax = b$ è $x = \begin{pmatrix} -1 \\ -2 \\ 2 \end{pmatrix}$.

Apparentemente la matrice A non presenta inconvenienti: è simmetrica, ha elementi non troppo distanti tra loro e ha determinante -1. Se però consideriamo il sistema $Ax = b + \delta b$ con $b + \delta b = \begin{pmatrix} 1.1 \\ 2.1 \\ 3.9 \end{pmatrix}$ si scopre che la soluzione è $x + \delta x = \begin{pmatrix} 0.3 \\ 0.3 \\ 0.4 \end{pmatrix}$ assai differente da x.

Esaminiamo le norme. Si ha: $\| x \| = 3$, $\| b \| \simeq 4.5$, $\| \delta x \| \simeq 3.09$, $\| \delta b \| \simeq 0.17$

Quindi $\frac{\| \delta x \|}{\| x \|} \simeq 1.03$ mentre $\frac{\| \delta b \|}{\| b \|} \simeq 0.03$

La norma di b è variata circa del 3%, mentre quella di x ha subito una variazione del 103% ! Questo significa che $\text{cond}(A)$ è superiore a 30.

3.1.3 Calcolo di norme e condizionamenti

Caso simmetrico

Se A è simmetrica, ($A = A^T$), allora, come è noto (teorema *spettrale*), A ha solo autovalori reali $\lambda_1, \lambda_2, \ldots, \lambda_n$. Ordiniamo gli n autovalori secondo il loro modulo: $| \lambda_1 | \leq | \lambda_2 | \leq \cdots \leq | \lambda_n |$, per cui con λ_1 si intenderà un'autovalore (può non essere unico) di A minimo in modulo e con λ_n un'autovalore massimo in modulo.

Si dimostra che: $\| A \| = | \lambda_n |$

Gli autovalori di A^{-1} sono notoriamente i reciproci di quelli di A, per cui la successione degli autovalori sarà: $\left| \frac{1}{\lambda_1} \right| \geq \left| \frac{1}{\lambda_2} \right| \geq \cdots \geq \left| \frac{1}{\lambda_n} \right|$. Quindi $\| A^{-1} \| = \lambda_1^{-1}$. In conclusione:

$$\text{cond}(A) = \left| \frac{\lambda_n}{\lambda_1} \right|$$

Una matrice è *ben condizionata* se gli autovalori non sono troppo distanti tra loro in modulo.

Esempio 3.2: Nell'esempio precedente gli autovalori erano circa 8.18 $-$ 2.23 0.05, per cui $\text{cond}(A) = \frac{8.18}{0.05} \simeq 150$ piuttosto elevato, come si è visto.

Caso generale

Se A non è simmetrica, consideriamo la matrice $A^T \cdot A$.
Si verifica che $A^T A$ è simmetrica e definita positiva, quindi i suoi autovalori $0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ sono tutti positivi. Per ogni i poniamo $s_i = \sqrt{\lambda_i}$.
Le radici quadrate $s_1 \leq s_2 \leq \cdots \leq s_n$ si dicono *valori singolari* di A.
Si dimostra che $\| A \| = s_n = \sqrt{\lambda_n}$. Analogamente $\| A^{-1} \| = 1/s_1 = \sqrt{1/\lambda_1}$, per cui

$$\text{cond}(A) = \frac{\text{max valore singolare di } A}{\text{min valore singolare di } A} = \frac{s_n}{s_1} = \sqrt{\frac{\lambda_n}{\lambda_1}}$$

Notiamo che per matrici simmetriche il calcolo di cond(A) fornisce lo stesso risultato nei due casi.

3.1.4 Metodi iterativi, il metodo di Jacobi

La pratica mostra che il metodo di eliminazione di Gauss diventa inaffidabile quando il sistema sia troppo grosso anche usando tutte le cautele possibili.
Per questa ragione conviene in molti casi ricorrere ai metodi iterativi che consentono di usare sempre la matrice originale e modificano invece la soluzione fino a farla tendere a quella esatta. Questi metodi, quando funzionano, permettono di superare anche l'eventuale mal condizionamento della matrice che rende ancor più instabile l'algoritmo gaussiano
Inoltre in molti casi consentono di avere una soluzione accettabile in un tempo più breve di quello richiesto dall'eliminazione gaussiana.

Il metodo di Jacobi consiste nel decomporre A come $A = S - T$, dove S è la matrice diagonale di A e T è la matrice complementare con diagonale nulla.
Esplicitamente:

$$\text{Se } A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots \\ a_{21} & a_{22} & a_{23} & \cdots \\ a_{31} & a_{32} & a_{33} & \cdots \\ \cdots & \cdots & & \end{pmatrix}$$

Allora:

$$S = \begin{pmatrix} a_{11} & 0 & 0 & \cdots \\ 0 & a_{22} & 0 & \cdots \\ 0 & 0 & a_{33} & \cdots \\ \cdots & \cdots & & \end{pmatrix} \quad T = \begin{pmatrix} 0 & -a_{12} & -a_{13} & \cdots \\ -a_{21} & 0 & -a_{23} & \cdots \\ -a_{31} & -a_{32} & 0 & \cdots \\ \cdots & \cdots & & \end{pmatrix}$$

Si scrive:

$$Ax = Sx - Tx = b \quad \text{cioè} \quad Sx = Tx + b$$

Sia ora x_0 un qualunque vettore. Sostituiamo x_0 a secondo membro e otteniamo:

$$Sx = Tx_0 + b$$

Dato che il sistema nella matrice S è facilmente risolubile, è agevole trovare la soluzione x_1 di questo sistema. Ricominciamo dal sistema $Sx = Tx + b$, sostituendo x_1 a secondo membro:

$$Sx = Tx_1 + b$$

Risolviamo nuovamente il sistema determinando x_2 e così via. Descriviamo esplicitamente il metodo di Jacobi nel caso particolare di un sistema 3×3:

Esempio 3.3: Siano

$$A = \begin{pmatrix} 3 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 3 \end{pmatrix} \quad b = \begin{pmatrix} 5 \\ 4 \\ -7 \end{pmatrix} \text{ Il sistema } Sx - Tx = b \text{ è: } \begin{cases} 3x &= -y + 5 \\ 3y &= -x - z + 4 \\ 3z &= -y - 7 \end{cases}$$

Partiamo con la terna $x_0 = 0\ ;\ y_0 = 0\ ;\ z_0 = 0$.
Sostituiamo a secondo membro $(0, 0, 0)$ e otteniamo:
$x_1 = 5/3\ ;\ y_1 = 4/3\ ;\ z_1 = -7/3$
Sostituiamo a secondo membro $(5/3, 4/3, -7/3)$ e otteniamo:
$x_2 = 11/9\ ;\ y_2 = 14/9\ ;\ z_2 = -25/9$
Sostituiamo a secondo membro $(11/9, 14/9, -25/9)$ e otteniamo:
$x_3 = 31/27\ ;\ y_3 = 50/27,\ ;\ z_3 = -77/27$
L'ultima terna è $(1.14\ldots\ ,\ 1.85\ldots\ ,\ -2.85\ldots)$ che è discretamente vicina alla soluzione esatta: $(1, 2, -3)$.

Non sempre il metodo di Jacobi converge, in realtà si può dimostrare che la successione converge alla soluzione del sistema, quale che sia la scelta iniziale di x_0, se e solamente la matrice $S^{-1}T$ ha tutti autovalori minori di 1 in modulo. Non è praticamente mai facile, né conveniente verificare direttamente la condizione del teorema. Esistono però dei criteri *sufficienti* di facile uso che garantiscano che essa sia verificata.

Proposizione 9 *(condizione sufficiente) L'algoritmo di Jacobi converge nei seguenti due casi interessanti:*

- *Se A è* diagonalmente dominante*, se cioè in ogni riga l'elemento a_{ii} è in modulo strettamente maggiore della somma dei moduli degli altri elementi della riga.*
- *Se A è simmetrica e definita positiva.*

Soprattutto il primo criterio è interessante, in effetti la matrice dell'esempio sopra è diagonalmente dominante

$$\begin{pmatrix} 3 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 3 \end{pmatrix} \begin{array}{l} |\,3\,| > |\,1\,| + |\,0\,| \\ |\,3\,| > |\,1\,| + |\,1\,| \\ |\,3\,| > |\,0\,| + |\,1\,| \end{array}$$

3.1.5 Metodi iterativi, il metodo di Gauss-Seidel

Il metodo consiste nel decomporre A come $A = S - T$, dove S è la parte triangolare inferiore di A e T è la matrice complementare.
Esplicitamente:

$$\text{Se } A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots \\ a_{21} & a_{22} & a_{23} & \cdots \\ a_{31} & a_{32} & a_{33} & \cdots \\ \cdots & \cdots & & \end{pmatrix}$$

Allora:

$$S = \begin{pmatrix} a_{11} & 0 & 0 & \cdots \\ a_{21} & a_{22} & 0 & \cdots \\ a_{31} & a_{32} & a_{33} & \cdots \\ \cdots & \cdots & & \end{pmatrix} \quad T = \begin{pmatrix} 0 & -a_{12} & -a_{13} & \cdots \\ 0 & 0 & -a_{23} & \cdots \\ 0 & 0 & 0 & \cdots \\ \cdots & \cdots & & \end{pmatrix}$$

Descriviamo esplicitamente anche il metodo di Gauss-Seidel nel caso particolare di un sistema 3×3.
Si scrive : $Sx = Tx + b$, cioè:

$$\begin{cases} a_{11}x & = & -a_{12}y - a_{13}z + b_1 \\ a_{21}x + a_{22}y & = & -a_{23}z + b_2 \\ a_{31}x + a_{32}y + a_{33}z & = & +b_3 \end{cases}$$

Nella pratica non si scrivono le matrici S e T, ma si scrive il sistema come nel metodo di Jacobi:

$$\begin{cases} a_{11}x & = & -a_{12}y - a_{13}z + b_1 \\ a_{22}y & = & -a_{21}x - a_{23}z + b_2 \\ a_{33}z & = & -a_{31}x - a_{32}y + b_3 \end{cases}$$

e la differenza sta nel fatto che a secondo membro non viene sostituita la terna (x_i, y_i, z_i), ma vengono utilizzati i valori di x, y, z via via trovati. Anche qui illustriamo il metodo di Gauss-Seidel usando lo stesso sistema dell'esempio precedente:

Esempio 3.4: Sostituiremo, per semplicità, i risultati intermedi dell'esempio con i loro sviluppi decimali arrotondati alla seconda cifra decimale.

$$\begin{cases} 3x &= -y+5 \\ 3y &= -x-z+4 \\ 3z &= -y-7 \end{cases}$$

Partiamo con la terna $x_0 = 0\ ;\ y_0 = 0\ ;\ z_0 = 0$.
Sostituiamo y_0, z_0 a secondo membro della E_1 e otteniamo:
$x_1 = 5/3 = 1.67$
Sostituiamo x_1, z_0 a secondo membro della E_2 e otteniamo:
$y_1 = 7/9 = 0.78$
Sostituiamo x_1, y_1 a secondo membro della E_3 e otteniamo:
$z_1 = -70/27 = -2.59$
Si noti come per ricavare la terna (x_1, y_1, z_1) si siano usato i risultati intermedi.
Sostituiamo y_1, z_1 a secondo membro della E_1 e otteniamo $x_2 = 1.41$
Sostituiamo x_2, z_1 a secondo membro della E_2 e otteniamo $y_2 = 1.73$
Sostituiamo x_2, y_2 a secondo membro della E_3 e otteniamo $z_2 = -2.91$
Al terzo passo si otterrà la terna $(1.09\,,\ 1.94\,,\ -2.98)$ e come si vede la convergenza è più veloce che con il metodo di Jacobi.

Si dimostra che l'algoritmo di Gauss-Seidel converge in ciascuna delle due ipotesi sufficienti, enunciate nel paragrafo precedente, in cui converge quello di Jacobi.
Per terminare aggiungiamo che i due metodi non sempre convergono, ma convergono in diversi casi che capitano nella pratica.

Cenno sul metodo di rilassamento: È possibile accelerare la convergenza di un metodo iterativo, "correggendo" ad ogni passo la soluzione ottenuta in modo da renderla più prossima a quella esatta.
L'idea base è la seguente: se x_{k-1} e x_k sono le soluzioni approssimate ottenute al $(k-1)^{\text{mo}}$ e k^{mo} passo di un algoritmo, si può proseguire l'algoritmo sostituendo x_k con $x_k^* = (1-\omega)x_{k-1} + \omega x_k$ dove ω è un numero compreso tra 1 e 2 (di solito intorno a 1.1), detto *coefficiente di rilassamento*. Il reperimento del coefficiente di rilassamento corretto è la parte più difficile, ma se lo si riesce a trovare (occorrono esperienza e sperimentazione) di solito esso vale per una vasta classe di sistemi lineari e consente anche di *far convergere i metodi iterativi in casi in cui i metodi base non convergerebbero*.

Terminiamo con un semplicissimo esempio che mostra l'analogia tra gli algoritmi di punto fisso e i metodi iterativi di algebra lineare.

Esempio 3.5: Usiamo Jacobi sul sistema diagonalmente dominante

$$\begin{cases} 2x+y &= 2 \\ -3x+4y &= 3 \end{cases} \text{ riscritto come } \begin{cases} 2x &= 2-y \\ 4y &= 3+3x \end{cases}$$

Geometricamente è l'intersezione di due rette.
Partiamo con $x_0 = (0,0)$.
Sostituiamo $(0,0)$ a secondo membro e otteniamo:
$x_1 = (1, 3/4)$
Sostituiamo $(1, 3/4)$ a secondo membro e otteniamo:
$x_2 = (5/8, 3/2)$
I primi 8 passi sono (arrotondando):
$x_1 = (1.0000, 0.7500)$ $\quad$ $x_2 = (0.6250, 1.5000)$
$x_3 = (0.2500, 1.2188)$ $\quad$ $x_4 = (0.3906, 0.9375)$
$x_5 = (0.5312, 1.0430)$ $\quad$ $x_6 = (0.4785, 1.1484)$
$x_7 = (0.4258, 1.1089)$ $\quad$ $x_8 = (0.4456, 1.0693)$
È interessante disegnare le due rette e la spezzata $x_0, x_1, \ldots$ che converge alla soluzione esatta.