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, in certi casi, è più conveniente dell'altra dal punto di vista numerico e può anche dare risultato differente.

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 $1/x = 1/(10^5 + 1)$ come 10^{-5} per cui il risultato sarà 0.

Si può scrivere la funzione nel modo equivalente $f(x) = \dfrac{x - 10^5}{10^5 x}$. 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 relativamente piccolo) è relativamente grande e ciò può talora rendere inaffidabile un calcolo.

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

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 $\sqrt{1-x^2}/\sqrt{2-x^2}$ non è integrabile elementarmente, per cui occorreranno metodi approssimati e quindi apparentemente il primo integrale è molto più semplice del secondo.
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. Quindi, se non è richiesta un'alta precisione, il calcolo del secondo integrale mediante formule approssimate di quadratura, può risultare più semplice di quello del primo calcolato esplicitamente. Questo senza togliere valore alla formula esplicita che, se disponibile, è importante in svariate questioni.

Esempio 1.6: Tabulare un polinomio.

Sia per esempio $P(x) = 1 + 5x - 2x^2 + 3x^3 + 6x^4$
Scrivere al 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  ;  t(4)=t(3)*x
P(x)=1 +5*t(1) -2*t(2) +3*t(3) +6*t(4)
```

con solo 4 somme, 7 prodotti

La procedura migliore è però quella descritta dallo schema di Ruffini-Hörner 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 $x_0 = 2$.

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

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

1.2.1 Errore assoluto e errore relativo

Definizione: Se $x \in \mathbb{R}$ e $\tilde{x}$ è il suo valore "calcolato", definiamo

Errore assoluto:	$\tilde{x} - x$	(o meglio $\mid \tilde{x} - x \mid$)
Errore relativo ($x \neq 0$):	$\dfrac{\tilde{x} - x}{x}$	$\left(\text{ o meglio } \dfrac{\mid \tilde{x} - x \mid}{\mid x \mid} \right)$

Osserviamo che, se il risultato di un calcolo è $\tilde{x} \neq 0$ e si sa che $x = 0$ o viceversa, non ha molto senso calcolare l'errore relativo.

Quando non si conosce x, ma si è in grado di calcolare in qualche modo $\tilde{x}$ e si sa maggiorare l'errore assoluto come $\mid \tilde{x} - x \mid < \varepsilon$, si può scrivere $x = \tilde{x} \pm \varepsilon_1$ con $\varepsilon_1 < \varepsilon$; spesso si scrive, con abuso di notazione, semplicemente $x = \tilde{x} \pm \varepsilon$.

Se invece si ha $\dfrac{\tilde{x} - x}{x} < \varepsilon$, si può scrivere $\tilde{x} = x(1 + \varepsilon_1)$ con $\varepsilon_1 < \varepsilon$.

1.2.2 Possibili sorgenti di errore

Quando il risultato di un calcolo è diverso da quello che ci si attende, occorre determinare la sorgente dell'errore. Le principali cause da prendere in considerazione sono:

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 da arrotondamento o troncamento**. Ne discuteremo più a lungo in seguito. Per esempio sostituendo 1/3 con 0.3333, per quante numerose siano le cifre decimali non si potrà mai avere errore nullo. Scrivendo 0.6667 in luogo di 2/3 si ha un errore inferiore a quello che si ha scrivendo 0.6666.
4. **Errore da cancellazione**. Quando si calcola la differenza $a - b$ tra due numeri positivi a, b molto prossimi, cambia repentinamente l'ordine di grandezza e questo procura spesso errori. Ne discuteremo più a lungo in seguito.
5. **Errore da troncamento del calcolo**. Un algoritmo indefinito di approssimazione deve cessare ad un certo punto senza aver necessariamente raggiunto il risultato esatto. Questo accade in algoritmi tipo quello delle tangenti di Newton o nell'integrazione numerica.
6. **Errore umano** (o più raramente di macchina). La possibilità di aver per esempio toccato un tasto inavvertitamente e aver letto male un dato va sempre presa in considerazione.

1.3 Basi numeriche e rappresentazione di numeri interi

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*

$$n = s \cdot (d_0 + d_1 b + \cdots + d_r b^r) \qquad 0 \leq d_i < b \qquad d_r \neq 0 \qquad s = \pm 1$$

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. Il numero $s = \pm 1$ è il segno.

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

$$4073 = 1 \cdot (3 + 7 \cdot 10 + 0 \cdot 10^2 + 4 \cdot 10^3) \qquad d_0 = 3 \quad d_1 = 7 \quad d_2 = 0 \quad d_3 = 4 \neq 0$$

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 pronuncia $quatre - vingt - douze$ 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 $0, 1, ..., 9$. 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 di un numero intero in base 2

Ricordiamo brevemente l'algoritmo più semplice per passare dalla usuale rappresentazione di un numero intero in base 10 a quella in base 2.
A titolo di esempio usiamo il numero 1354. Dividiamo successivamente per 2 e 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 **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 $1354_{10} = 54A_{16} = 10 + 4 \cdot 16 + 5 \cdot 16^2$

Attenzione: Questo noto algoritmo in realtà non può essere utilizzato da un computer. Il computer lavora in base 2 quindi per eseguire il conto precedente dovrebbe avere il numero già scritto in base 2. Viene quindi usato un altro algoritmo che si basa sullo schema di Ruffini-Hörner.

1.3.3 Conversione macchina di un numero intero in base 2

Usiamo come sopra il numero 1354.
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 il numero decimale 1354 viene scritto, usando lo schema di Ruffini-Hörner, come

$$1354 = 4 + 5 \cdot 10 + 3 \cdot 10^2 + 1 \cdot 10^3 \qquad = \qquad 4 + 10 \cdot (5 + 10 \cdot (3 + 1 \cdot 10))$$

Dato che nell'ultima espressione compaiono solo numeri compresi tra 0 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.

Osservazione: Il passaggio dalla rappresentazione binaria a quella decimale, necessario per visualizzare il risultato di un calcolo eseguito dal computer in base 2, è ancora più complesso. Occorre dividere il numero successivamente per 10 (in binario 1010). Le cifre decimali si ricavano a partire dalla meno significativa come resto delle divisioni; la conversione termina quando l'ultimo quoziente è 0. Dato che una divisione richiede al computer un tempo più che doppio rispetto a un prodotto, questa conversione è di norma più onerosa.

1.3.4 Rappresentazione macchina di un numero intero

Vedremo tra poco come il computer rappresenta internamente un numero reale qualunque, ma spesso, quando ha a che fare solo con numeri interi, dopo averli convertiti in base 2, la macchina li può rappresentare, a seconda delle esigenze, usando un byte (8 bit), due byte (16 bit) oppure quattro byte (32 bit) (o anche 11 bit, come vedremo nella rappresentazione dell'esponente di un numero reale).

Descriviamo per semplicità la rappresentazione con un solo byte (che comunque ha evidentemente uso limitato).

Se non usiamo numeri negativi, mediante un byte (8 bit) si possono rappresentare 2^8, cioè 256 numeri, quelli da 0 a 255.

Se invece vogliamo rappresentare anche i negativi, ci sono due possibilità:

Numeri e segno: Si usano 7 bit per il numero e un bit per il segno. Si possono rappresentare i numeri da -127 a 127 e ovviamente non ha senso il numero 0 col segno meno, quindi si possono rappresentare 255 numeri. Questa rappresentazione è però scarsamente usata.

Numeri con segno: I numeri negativi vengono rappresentati in forma complementare col primo bit uguale a 1.

Il più grande numero positivo rappresentabile è 127 che, in forma binaria, è [0111 1111]. Il numero -1 viene rappresentato come [1111 1111]; infatti la sua somma con [0000 0001] è [0000 0000] (dato che va perso il nono bit di riporto). Il numero più piccolo è -128 che si scrive come [1000 0000]. E in effetti $127 + (-128)$ dà -1.

Usando due byte, i numeri rappresentabili vanno da -32768 a 32767 (*short integers*).

Usando quattro byte: da -2147483648 a 2147483647 (*long integers*).

1.4 Basi numeriche e rappresentazione di numeri reali

1.4.1 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... ovvero $0.\overline{3}$ (periodico), quindi non potrà mai essere scritto in modo esatto in base 10. 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 in base 2 la sua rappresentazione è $0.0\,0011\,0011\,00... = 0.0\,\overline{0011}$ (periodico).

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

Proposizione 2 *Fissiamo la* base *b, dove* $b \in \mathbb{N}$ *e* $b \geq 2$.
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*
$m = d_1, d_2, d_3, ...$	è una successione (spesso infinita) detta *mantissa*
	I numeri d_i sono numeri interi tali che $0 \leq d_i < b$.

Nella successione m si ha $d_1 \neq 0$ e d_1 è detto *prima cifra significativa.*
La successione m non è mai definitivamente $b-1\,,\ b-1\,,\ b-1\,, ...$
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 di potenze 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:

$24.31 =$	$1 \cdot (0.2431) \cdot 10^2$	segno 1	mantissa 2 4 3 1	(o 0.2431)	caratt. 2
$-0.0349 =$	$-1 \cdot (0.349) \cdot 10^{-1}$	segno -1	mantissa 3 4 9	(o 0.349)	caratt. -1
$1/3 =$	$1 \cdot (0.333...) \cdot 10^0$	segno 1	mantissa 3 3 3 ...	(o 0.333...)	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 $1 \cdot (0.29999999...) \cdot 10^2$ non è valida perché le cifre della mantissa sono tutte $b-1=9$ da un certo punto in poi; il numero periodico $29.\overline{9} = 29.999...$ è infatti una rappresentazione errata del numero 30, dato che la mantissa, intesa come sommatoria, vale 0.3.

Esempio 1.9: Determiniamo la mantissa di $(0.9)_{10}$ usando la base 16.

Poniamo innanzitutto $a_1 = (0.9)_{10} = 1 \cdot (d_1 \cdot 16^{-1} + d_2 \cdot 16^{-2} + \cdots)$.

Si ha: $a_1 = (0.9)_{10} = \dfrac{d_1}{16} + \dfrac{d_2}{16^2} + \cdots$. Moltiplichiamo per 16:

$16 \cdot a_1 = d_1 + \dfrac{d_2}{16} + \cdots$. Ma $16 \cdot a_1 = 16 \cdot 0.9 = 14.4$ quindi:

$16 \cdot 0.9 = 14.4 = 14 + 0.4 = d_1 + \dfrac{d_2}{16} + \dfrac{d_3}{16^2} + \cdots$ pertanto $d_1 = (14)_{10} = E_{16}$

Poniamo quindi $a_2 = (0.4)_{10} = \dfrac{d_2}{16} + \dfrac{d_3}{16^2} + \cdots$. Moltiplichiamo per 16:

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

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:
$(0.9)_{10} = 0.1110\,0110\,0110\cdots = 0.1\,1100\,1100\,1100\cdots = 1 \cdot (0.1\,\overline{1100}) \cdot 10^0$

Osservazione 1: Come nel caso dei numeri interi, l'algoritmo dell'esempio precedente non è eseguibile dal computer, che dovendo lavorare in base 2, esegue di fatto la divisione tra 9 e 10 dopo aver rappresentato 9 e 10 in base 2.

In effetti, quando deve rappresentare per esempio il numero 1234.56, il computer converte in binario il numero intero 123456 e poi lo divide (usando vari accorgimenti) per 100.

Osservazione 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 nella notazione scientifica si pone la cifra più significativa prima della virgola invece che dopo, quindi l'esponente in notazione scientifica è inferiore di un'unità.
Vediamo la differenza nei quattro esempi precedenti (si suppone una calcolatrice con visore a 8 cifre):

Numero		virgola variabile		notazione scientifica		visore calcolatrice
24.31	=	$1\cdot(0.2431)\cdot10^{2}$	=	2.431×10^{1}	=	24.31
-0.0349	=	$-1\cdot(0.349)\cdot10^{-1}$	=	-3.49×10^{-2}	=	-0.0349
$1/3$	=	$1\cdot(0.333...)\cdot10^{0}$	=	3.3333333×10^{-1}	=	0.3333333
125 000 000	=	$1\cdot(0.125)\cdot10^{9}$	=	1.25×10^{8}	=	1.25 E08

1.4.2 Numeri macchina

Mentre i numeri reali sono infiniti, i numeri macchina disponibili sono in numero finito, quindi, a seconda dell'architettura e dei limiti della macchina, occorre fissare 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

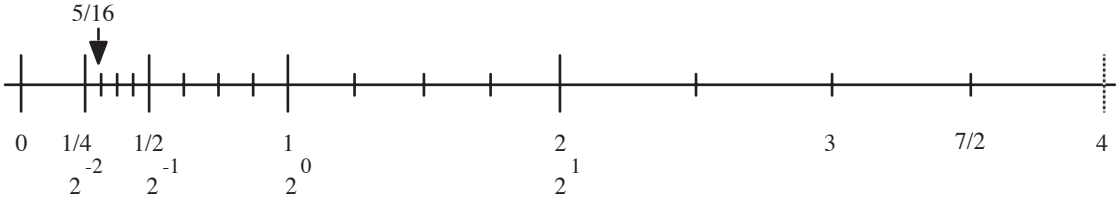
$$x = s \cdot (d_1 \cdot b^{-1} + d_2 \cdot b^{-2} + \cdots + d_t \cdot b^{-t}) b^p$$

cioè numeri a virgola variabile in cui però $L \leq p \leq U$ e la mantissa ha un numero fissato t di cifre.

Per capire come sono disposti i numeri macchina, li elenchiamo tutti supponendo, solo per ragioni di semplicità descrittiva, che:
$b = 2 \qquad t = 3 \qquad L = -2 \qquad U = 1$
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. Inoltre $-2 \leq p \leq 1$.

Per esempio:	$0.100\cdot10^{-1}$	=	$1/4_{10}$	= 2_{10}^{-2}	il più piccolo positivo
	$0.101\cdot10^{-1}$	=	$5/16_{10}$		
	$0.111\cdot10^{10}$	=	$7/2_{10}$		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 sulla retta reale, ma tra 2^t e 2^{t+1} sono equidistanti. Notiamo inoltre il grosso vuoto tra 0 e il minimo numero macchina.

Per memorizzare questi numeri occorrono 6 bit: 3 bit per la mantissa, 1 bit per il segno e 2 bit per l'esponente che può assumere quattro valori: $00 = 0_{10}$; $01 = 1_{10}$; $10 = -2_{10}$; $11 = -1_{10}$, gli ultimi due rappresentati come interi con segno, in 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: Nella pratica, una delle rappresentazioni macchina più usate è quella dei cosiddetti numeri *a doppia precisione*. La base è 2 e ogni numero è memorizzato con 8 byte cioè con 64 bit. Solitamente la suddivisione dei bit è la seguente:
• 1 bit per il segno
• 11 bit per la caratteristica che è un intero con segno. Quindi il range va da $-2^{-10} = -1024$ a $2^{10} - 1 = 1023$.
• 6 byte e mezzo per la mantissa che ha quindi al massimo 52 cifre.

In questo modo il massimo numero rappresentabile è 2^{1024} "meno un bit" (vedi più avanti la definizione del numero eps) che è circa 1.7977×10^{308} e le 52 cifre binarie della mantissa consentono di rappresentare i numeri in base 10 con circa 16 cifre decimali.
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.

I numeri BCD: Benché la cosa sembri a prima vista strana, è possibile rappresentare i numeri reali in macchina usando la base 10 anziché la base 2. Questi numeri sono comunemente detti BCD (*binary coded decimals*). Si usano, come nella doppia precisione, 6 byte e mezzo per la mantissa. Ogni mezzo byte è una cifra della mantissa decimale (e quindi può assumere valori solo da 0000 a 1001). In questo modo si possono memorizzare 13 cifre decimali. Ci sono meno numeri rappresentabili rispetto alla doppia precisione e il tempo di calcolo è leggermente superiore, ma, da un certo punto di vista, si ha maggior precisione perché non occorre una doppia conversione di base. Comunque attualmente i BCD sono di uso sempre meno frequente.

1.4.3 Rappresentazione in macchina di un numero reale

Sia $x \in \mathbb{R}$, $x \neq 0$; rappresentiamo il numero x con segno s, mantissa $m = d_1\,, d_2\,, \ldots$ e caratteristica p. Una volta fissati i parametri $b, t, [L, U]$, ci sono 4 possibilità:

1. Si ha $L \leq p \leq U$ e $d_i = 0$ per $i > t$. Quindi è possibile rappresentare x esattamente in macchina.
2. $p > U$: *overflow*. È praticamente impossibile rappresentare il numero. Diverse macchine si arrestano ed emettono segnale di errore. Altre restituiscono un numero speciale **`Inf`** (o $-$**`Inf`** se x è negativo) che significa appunto numero con esponente maggiore di U. Se questo non succede, ciò comporta grave errore e risultato inaffidabile.
3. $p < L$: *underflow*. Alcune macchine si arrestano ed emettono segnale di errore. Altre sostituiscono x con 0, ma in certi casi ciò è pericoloso perché sostituire un numero, anche molto piccolo con 0, significa generare un errore relativo teoricamente infinito. Si tenga anche presente che la distanza tra 0 e il minimo numero macchina è relativamente grande.
4. $L \leq p \leq U$, ma la mantissa ha più di t cifre (spesso è 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. Il numero x non può essere rappresentato esattamente, quindi occorre decidere in che modo approssimarlo con un numero macchina e sapere quale errore si commette.

1.4.4 Arrotondamento e troncamento

Siamo nel caso 4.

Per semplicità supponiamo $x > 0$. Per rappresentare x in macchina ci sono due tecniche.

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

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 viene rappresentato come $0.666\cdots 667 \cdot 10^0$.

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
$0.011\|0 +$	$0.0110\|1 +$	$0.01101\|1 +$
$\underline{0.000\|1 =}$	$\underline{0.0000\|1 =}$	$\underline{0.00000\|1 =}$
$0.011\|1$	$0.0111\|0$	$0.01110\|0$
quindi 0.011	quindi 0.0111	quindi 0.01110

Come vedremo al paragrafo successivo, l'arrotondamento, benché leggermente più complesso, è di norma preferito al troncamento perché l'errore risulta dimezzato.

1.5 Errori macchina

1.5.1 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*.

La proposizione che segue fornisce una maggiorazione dell'errore relativo commesso sostituendo x con $\mathrm{fl}(x)$:

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 $\frac{1}{2}\, b^{1-t}$ (o b^{1-t} se si usa il troncamento) è denotato eps ed è detto *precisione macchina*. Conviene però definirlo in modo indipendente da troncamento o arrotondamento.

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 la maggiorazione dell'errore relativo commesso rappresentando un numero in macchina. 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: In base 10. 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: Il maggior numero rappresentabile in macchina usando i numeri a doppia precisione è $(2 - \text{eps}) \cdot 2^{1023}$
Inoltre, usando numeri in doppia precisione e arrotondamento 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
```

1.5.2 Operazioni macchina

Date le 4 operazioni aritmetiche elementari, definiamo le corrispondenti operazioni macchina in questo modo:

$x \oplus y = \text{fl}(x + y)$ $\qquad$ $x \ominus y = \text{fl}(x - y)$
$x \otimes y = \text{fl}(x \cdot y)$ $\qquad$ $x \oslash y = \text{fl}(x/y)$

Gli input x, y saranno numeri macchina, ma non è detto che il risultato delle operazioni usuali $x+y$ etc. sia un numero macchina, per cui spesso $x \oplus y$ è diverso da $x + y$ e così per le altre operazioni. In effetti, si ha per esempio $x \oplus y = (x + y)(1 + \varepsilon) = x(1 + \varepsilon) + y(1 + \varepsilon)$ con $\varepsilon <$ eps.

Le quattro operazioni macchina non godono sempre di proprietà elementari valide nelle operazioni usuali. Per esempio, spesso si ha $(x \oplus y) \oplus z \neq x \oplus (y \oplus z)$.

Possono succedere cose abbastanza strane tipo il fatto che $x \oplus y = x$ se $\mid y \mid < \dfrac{\text{eps}}{b} \mid x \mid$, cioè se si somma a x un numero al di là della precisione macchina in confronto a x.

Per esempio in base 10 con 4 cifre decimali e range 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.
Da questo esempio si intuisce il fenomeno della *cancellazione*.

1.5.3 La cancellazione

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

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

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

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, ma il primo addendo aveva già subito conversione a numero macchina, 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. È meglio calcolare nel seguente modo:

$e^{30} = 1 + 30 + \frac{900}{2} + \frac{27000}{6} + \cdots$ 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 dare una possibile spiegazione teorica del 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: $\delta = \dfrac{(\tilde{a} + \tilde{b})(1+\varepsilon) - (a+b)}{a+b} = \cdots = \varepsilon + \left(\dfrac{a\varepsilon_1 + b\varepsilon_2}{a+b}\right)(1+\varepsilon) \qquad \text{con } \mid \varepsilon, \varepsilon_1, \varepsilon_2 \mid < \text{eps}$

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: $a = 0.123456 \qquad b = -0.123454$. Se $t = 5$ (numero di cifre decimali), allora
$\tilde{a} = 0.12346 \qquad \tilde{b} = -0.12345 \qquad a + b = 0.2 \cdot 10^{-5} \qquad \tilde{a} \oplus \tilde{b} = 0.1 \cdot 10^{-4}$
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 $b > 0$. 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 $x_1 = x_2 \cdot a \cdot c$.

1.5.4 L'errore nelle operazioni macchina

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

È possibile effettuare un conto analogo per il prodotto e si scopre che

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

Un conto analogo per la divisione mostra che

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

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

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.
Ci occuperemo dei vari modi di approssimare una soluzione, una volta stabilito che ne esista almeno una in un qualche sottoinsieme di $\mathbb{R}$.

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 il noto teorema degli zeri, nell'intervallo $[a, b]$ esiste almeno un x_0 tale che $f(x_0) = 0$; se poi $f(x)$ è strettamente monotona, il numero x_0 è unico.

Per approssimare x_0 si divide l'intervallo a metà: $\left[a\,,\ \frac{a+b}{2}\,,\ b\right]$ e si sostituisce l'intervallo $[a, b]$ con quello tra i due intervalli $\left[a\,,\ \frac{a+b}{2}\right]$ e $\left[\frac{a+b}{2}\,,\ b\right]$ nel quale la funzione ha ancora valori discordi negli estremi e così via.

È praticamente rarissimo che a un certo punto il punto medio dell'intervallo sia proprio lo zero cercato, per cui l'algoritmo normalmente non ha termine e occorre arrestarlo a un certo punto mediante un qualche criterio. Il criterio di solito è uno di questi tre

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

Una delle peculiarità dell'algoritmo di bisezione è il fatto che tra i primi due criteri c'è una relazione.

Si ha: $\varepsilon < \frac{| b-a |}{2^{p+1}}$, quindi $p < 1 + \log_2 \frac{| b-a |}{\varepsilon}$.

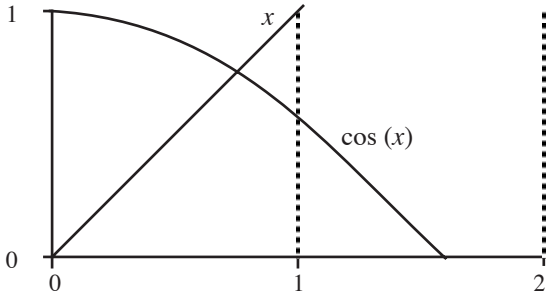
Invece per quanto riguarda il terzo criterio, è a priori impossibile prevedere il numero di passi, a meno di non avere informazioni sulla derivata di $f(x)$ (se esiste).

L'algoritmo è relativamente lento, dato che $\log_2(10) \simeq 3.32$, quindi occorrono circa tre passi per avere una cifra decimale esatta; in compenso, nelle ipotesi date, l'algoritmo converge sicuramente ed è 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 $\cos(0.7) = 0.7648$. 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 almeno quattro cifre decimali esatte della soluzione del problema.

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

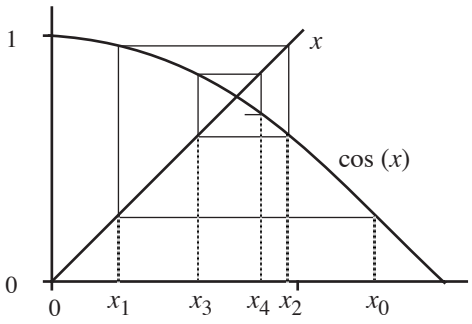
Nell'esempio $\alpha = 0.7391$ è un punto fisso di $\cos(x)$ nell'intervallo $[0, 1]$ e l'algoritmo consente di approssimarlo facilmente.
Benché sembri un caso particolare del problema iniziale di risolvere un'equazione $f(x) = 0$, l'algoritmo di punto fisso è alla base di molti altri metodi.

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

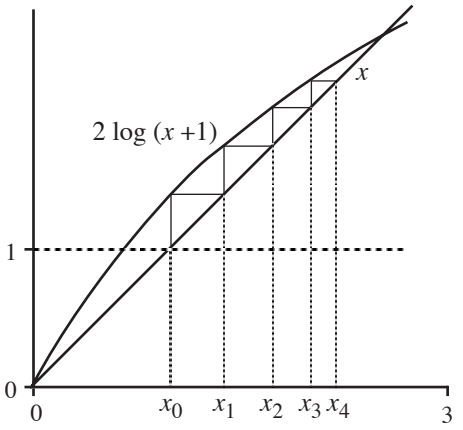
Non sempre la successione è convergente, però basta una condizione sulla derivata di $g(x)$:

Proposizione 5 *(condizione sufficiente) Sia α 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 $|\ g'(x)\ | < 1$ in I e $x_0 \in I$, allora la successione $x_i = g(x_{i-1})$ determinata dall'algoritmo di punto fisso è convergente e si ha $\lim_{i\to\infty} x_i = \alpha$. Inoltre α è unico nell'intervallo.*

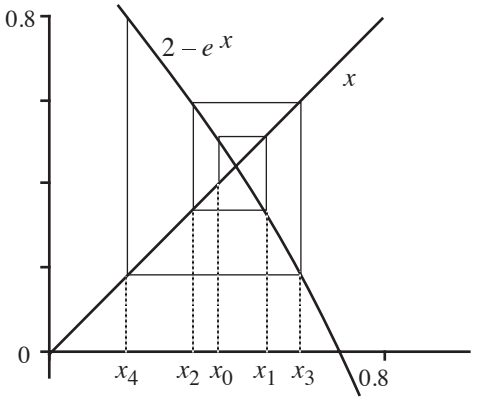
Graficamente:



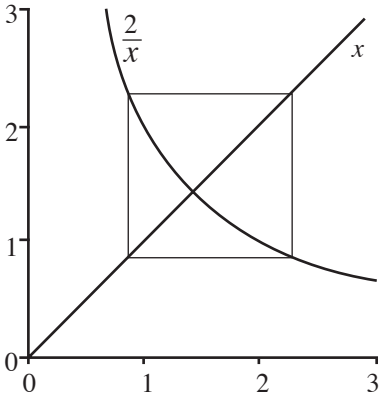
Equazione $x = \cos(x)$ come sopra, ma partendo con un x_0 lontano da α per chiarezza; la successione converge a segno alterno ad α.



Equazione $x = 2 \cdot \log(x + 1)$ con $x_0 = 1$. La successione converge monotonamente ad α, ma la convergenza è lenta.



Equazione $x = 2 - e^x$. La successione 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 Criteri d'arresto dell'algoritmo di punto fisso

Dato che l'algoritmo non ha quasi mai termine, occorre porre un criterio di arresto:
Il criterio viene di solito scelto tra uno di questi due

- Quando $\mid x_i - x_{i+1} \mid < \varepsilon$ con ε prefissato. Il criterio può essere poco valido quando $g'(x)$ è positivo, perché può capitare che $\mid x_i - x_{i+1} \mid$ sia più piccolo di $x_{i+i} - \alpha$, come succede nel secondo esempio grafico sopra.
- Quando $\dfrac{\mid x_i - x_{i+1} \mid}{\min\{\mid x_i \mid, \mid x_{i+1} \mid\}} < \varepsilon$ con ε prefissato. Come criterio può essere più affidabile, come ora vedremo.

Vediamo 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 possibile valutare l'errore assoluto, dato che α è compreso tra x_i e x_{i+1}.

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 $g(x_{i-1}) = x_i$

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

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:

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

Quindi il criterio di arresto maggiora l'errore assoluto se $g'(x) < 0$. Non si può invece dare una maggiorazione di $\mid x_{i-1} - \alpha \mid$ se $g'(x) > 0$.

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 $\min\{\mid x_i \mid, \mid x_{i+1} \mid\} \simeq \mid \alpha \mid$, quindi l'ultima espressione è circa ε.

Se $g'(x) > 0$ e soprattutto se $g'(x) \simeq 1$ l'errore può essere ancora grande anche in presenza di ε piccolo. Per valutare la distanza assoluta o relativa di x_i da α occorre conoscere almeno approssimativamente il valore di $g'(x)$ in un intorno di α.

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 si ha $x_i \neq \alpha$ per ogni i), allora il numero $\gamma = \lim_{i \to \infty} \dfrac{\mid x_{i+1} - \alpha \mid}{\mid x_i - \alpha \mid}$ è detto *fattore di convergenza*.

Se l'algoritmo converge ad α, 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 $\lim_{i\to\infty} \frac{|\, x_{i+1} - \alpha \,|}{|\, x_i - \alpha \,|^p} = \ell$ con $\ell \neq 0$, si dice che p è l'*ordine di convergenza.*

Se l'ordine di convergenza è 1, ciò significa che le cifre decimali (o binarie) esatte del risultato aumentano all'incirca linearmente ad ogni passo dell'algoritmo. Se l'ordine di convergenza è invece 2, ad ogni passo il numero di cifre decimali (o binarie) esatte viene circa raddoppiato.

L'ordine di convergenza è strettamente legato alla derivata prima e alle successive:

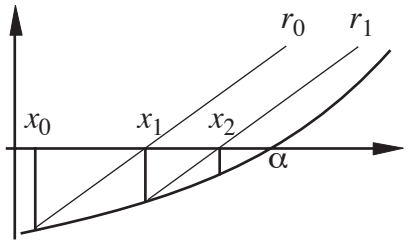
Proposizione 6 *Nelle ipotesi precedenti, supponiamo che $g(x)$ sia di classe C^p in $[\alpha - \varrho\,,\ \alpha + \varrho]$ e $x_0 \in [\alpha - \varrho\,,\ \alpha + \varrho]$. Se l'ordine di convergenza della successione di punto fisso è p, allora:*

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

In parole povere, se $p \geq 2$, la tangente a $g(x)$ in α è orizzontale.

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 $x_1 = x_0 - \dfrac{f(x_0)}{h}$.
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 $|\, g'(x) \,| < 1$.

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.

Per quanto riguarda i criteri d'arresto, essenzialmente ce ne sono tre, di cui due sono quelli dell'algoritmo di punto fisso cioè

- $|\, x_i - x_{i+1} \,| < \varepsilon$ con ε prefissato.
- $\dfrac{|\, x_i - x_{i+1} \,|}{\min\{|\, x_i \,|, |\, x_{i+1} \,|\}} < \varepsilon$ con ε prefissato.

Si può inoltre usare il seguente criterio:

- $|\, f(x_i) \,| < \varepsilon$ con ε prefissato.

Quest'ultimo criterio è il più semplice e osserviamo che, dato che si ha $f(x_i) = (x_i - g(x_i)) \cdot h(x_i)$ e $g(x_i) = x_{i+1}$, esso equivale a chiedere che

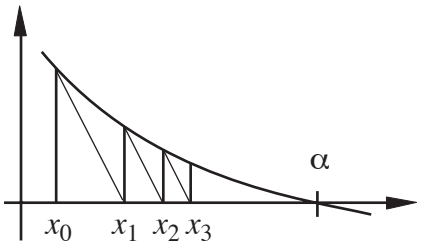
$$|\, f(x_i) \,| = |\, x_i - x_{i+1} \,| \cdot |\, h(x_i) \,| < \varepsilon \qquad \text{ovvero} \qquad |\, x_i - x_{i+1} \,| < \varepsilon \,/\, |\, h(x_i) \,|$$

Quindi sarà in pratica equivalente al primo, se si dispone di una maggiorazione di $|\, 1/h(x) \,|$ in un intorno di α.

2.1.6 Metodo delle corde

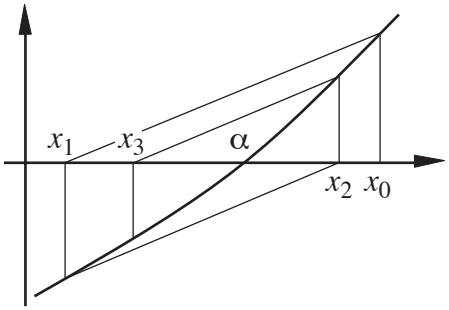
È il metodo più semplice ed è quello con la scelta $h(x) = m$ (m inclinazione costante).

Si cerca quindi il punto fisso della funzione $g(x) = x - \dfrac{f(x)}{m}$,
ovvero l'algoritmo è $x_{i+1} = x_i - \dfrac{f(x_i)}{m}$.
Condizione sufficiente affinché l'algoritmo converga è che $| g'(x) | = \left|1 - \dfrac{f'(x)}{m}\right| < 1$.



Equivalentemente la condizione è che in un intorno di α comprendente x_0 valgano le tre seguenti:

- $f'(x) \neq 0$
- $f'(x) \cdot m > 0$ (devono avere lo stesso segno)
- $| m | > \dfrac{1}{2} \max\{f'(x)\}$



Il metodo delle corde, se converge, converge di ordine 1.
I due esempi grafici mostrano come la convergenza possa essere monotona o alternante.

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: $x_{i+1} = x_i - \dfrac{f(x_i)}{f'(x_i)}$.

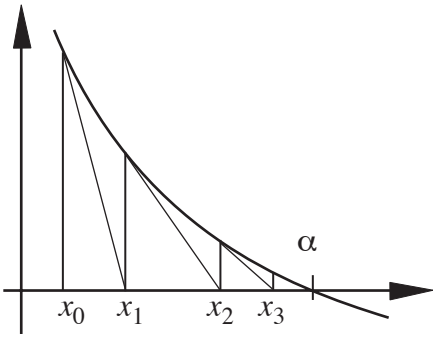
Condizione sufficiente affinché l'algoritmo converga è che $| g'(x) | = \left|\dfrac{f(x)f''(x)}{(f'(x))^2}\right| < 1$.

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.

Diversamente dal metodo delle corde, perché sia garantita la convergenza, occorre anche scegliere opportunamente il punto iniziale x_0 a destra o a sinistra di α, a seconda delle circostanze.
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: Si può 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 da un 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, non così se si inizia invece con $x_0 < 0$.

2.1.8 Metodo delle secanti

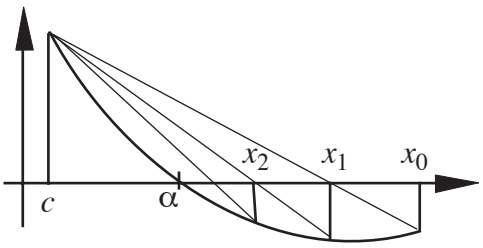
Sia α lo zero da cercare; fissiamo $x = c$ prossimo ad α e scegliamo come valore di partenza dell'algoritmo un x_0 tale che α sia compreso tra c e x_0.
Consideriamo la retta congiungente i due punti $(c, f(c))$ $(x_0, f(x_0))$. L'intersezione tra la retta e l'asse x è il nuovo punto x_1.
L'algoritmo 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:

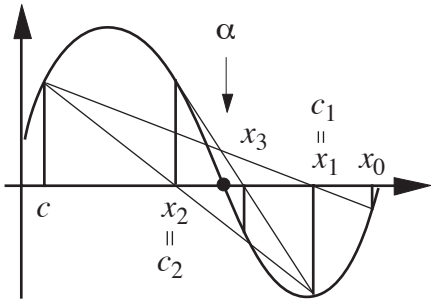
Proposizione 8 *Supponiamo che la funzione $f(x)$ definita nell'intervallo $I = [a, b]$ sia di classe C^2 e si abbia $f'(x), f''(x) \neq 0$.*
Se si scelgono nell'intervallo c, x_0 tali che $f(c) \cdot f''(c) \geq 0$ e inoltre $f(x_0) \cdot f''(x_0) \leq 0$, allora l'algoritmo delle secanti converge (monotonamente).

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 $f(x)$. Per certe funzioni non definite mediante formule esplicite il calcolo della derivata può essere estremamente difficoltoso. 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, converge di ordine 1 e ha il pregio di convergere 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.

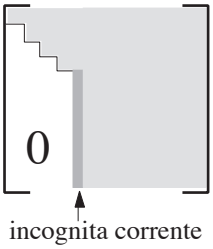


3.1 Algebra lineare numerica

3.1.1 Le varianti dell'algoritmo di Gauss

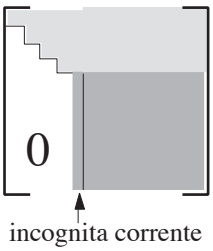
Dato un sistema lineare *quadrato* $Ax = b$ con A matrice invertibile (c'è sempre il problema di scoprire se lo sia), vediamo quali sono i metodi di risoluzione. L'algoritmo di Gauss è il metodo base, ma ha parecchie varianti. Elenchiamo le principali varianti:

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 in un insieme più vasto richiede 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. **Pivotizzazione scalata**: Osserviamo che, se una riga di un sistema viene moltiplicata per una costante non nulla, il sistema ottenuto è equivalente, ma può cambiare la scelta del pivot nella strategia della pivotizzazione parziale.

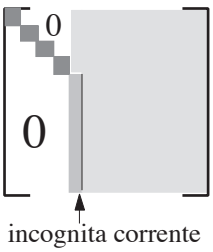
   Questo è il motivo per cui a volte si usa, in luogo della pivotizzazione totale, a parità di tempo, la cosiddetta *pivotizzazione parziale scalata.* In questo caso un elemento di una matrice viene considerato grande, non se è grande in assoluto, ma se lo è rispetto al resto della riga.

   Per la precisione, in ognuna delle righe interessate alla ricerca del pivot, si calcola la grandezza della riga i-esima che è definita come $d_i = \max_j \{ \mid a_{ij} \mid \}$.

   Quindi la grandezza del possibile pivot p_i della i-esima riga è calcolata come $\frac{\mid p_i \mid}{d_i}$ e viene scelto come pivot quello per cui il rapporto è maggiore.

4. **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 dalla memoria del computer i 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.

5. **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 $A = L \cdot U$, 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 $L \cdot U \cdot x = b$.
   La matrice L è (quasi) triangolare inferiore, nel senso che lo è a meno di un riordinamento delle righe e inoltre gli elementi della diagonale (una volta riordinata) sono tutti 1.
   Pertanto è facile risolvere il sistema lineare $L \cdot t = b$ 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 $L \cdot t = b$. Risulta pertanto $L \cdot b_1 = b$.
   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 LU della matrice A, qualunque altro sistema $Ax = c$, avente la stessa matrice dei coefficienti, ma diverso termine noto, può essere risolto in tempo brevissimo usando la fattorizzazione LU già trovata, dato che il calcolo di L e U è la parte più onerosa dell'intero processo e si può evitare di ripeterlo.

6. **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 la fattorizzazione LU 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, salvo casi particolarissimi, 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.

7. **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 l'algoritmo 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 confrontare la grandezza di δx con quella di δb.
Usualmente la grandezza di un vettore si misura mediante la *norma euclidea*:

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

La norma ha tre proprietà

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

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: $\dfrac{\| \delta b \|}{\| b \|}$
La "piccolezza" di δb sarà quindi misurata da $\| \delta b \| \, / \, \| b \|$ e analogamente 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 $\| A \delta x \|$ o meglio individuare una relazione tra $\| x \|$ e $\| Ax \|$ e una tra $\| \delta x \|$ e $\| A \delta x \|$. Queste relazioni dipenderanno ovviamente da proprietà della 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 si ricavano immediatamente le due relazioni

$$\| Ax \| \leq \| A \| \cdot \| x \| \qquad\qquad \| A \delta x \| \leq \| A \| \cdot \| \delta x \|$$

Queste ci consentono di procedere nel nostro conto.

Prima di continuare osserviamo che rimane il problema di calcolare $\| A \|$, dato che la definizione precedente non può essere usata per calcolare direttamente la norma di una matrice. Questo è un problema più complesso che rinviamo al paragrafo successivo.

Si ha:

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

Per il confronto tra δb e δx conviene scrivere diversamente la relazione $A\delta x = \delta b$ usando la matrice inversa A^{-1}

$$\delta x = A^{-1}\delta b \qquad \text{da cui} \qquad \| \delta x \| = \| A^{-1}\delta b \| \leq \| A^{-1} \| \, \| \delta b \|$$

Le due relazioni ottenute

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

si possono scrivere:

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

Moltiplicando le due diseguaglianze membro a membro, si ha la relazione cercata:

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

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 quindi la definizione

Definizione: Se A è una matrice quadrata e invertibile, il numero

$$\text{cond}(A) = \| A \| \, \| A^{-1} \|$$

è detto *numero di condizionamento* di A.

La relazione precedente si scrive quindi usualmente come

$$\frac{\| \delta x \|}{\| x \|} \leq \text{cond}(A) \, | \, \frac{\| \delta b \|}{\| b \|}$$

Rimane il problema di calcolare $\| A \|$ e $\| A^{-1} \|$ e quindi cond(A).

Prima però esaminiamo un esempio.

Esempio 3.1: Siano $A = \begin{pmatrix} -1 & 2 & 2 \\ 2 & 1 & 3 \\ 2 & 3 & 6 \end{pmatrix}$ e $b = \begin{pmatrix} 3 \\ 4 \\ 8 \end{pmatrix}$ La soluzione del sistema lineare $Ax = b$ è $x = \begin{pmatrix} 1 \\ 2 \\ 0 \end{pmatrix}$.

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 $b + \delta b = \begin{pmatrix} 3.1 \\ 4.2 \\ 7.9 \end{pmatrix}$ si scopre che la soluzione è $x + \delta x = \begin{pmatrix} 2.9 \\ 5.3 \\ -2.3 \end{pmatrix}$ assai differente da x.

Esaminiamo le norme. Si ha: $\| x \| \simeq 2.24$, $\| b \| \simeq 9.4$, $\| \delta x \| \simeq 4.45$, $\| \delta b \| \simeq 0.24$

Quindi $\dfrac{\| \delta b \|}{\| b \|} \simeq 0.03$ mentre $\dfrac{\| \delta x \|}{\| x \|} \simeq 1.99$

La norma di b è variata circa del 3%, mentre quella di x ha subito una variazione del 199% !

Dato che $\dfrac{\| \delta x \|}{\| x \|} \Big/ \dfrac{\| \delta b \|}{\| b \|} \leq \text{cond}\, A$, questo significa che cond(A) è superiore a $199/3 \simeq 66$.

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 $\lambda_1, \lambda_2, ..., \lambda_n$. 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 di A sono circa 8.18 , -2.24 , 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 facilmente che $A^T A$ è una matrice simmetrica e definita positiva, quindi i suoi autovalori $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ 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 $\text{cond}(A)$ fornisce lo stesso risultato nei due casi.

Osservazioni:

- Dal calcolo di $\text{cond}(A)$ si deduce che $\text{cond}(A) \geq 1$ per ogni matrice, mentre non esiste un limite superiore.
- Se A è una matrice *ortogonale*, ovvero le colonne di A sono a due ortogonali e di norma 1, come è noto, A è una matrice isometrica, cioè per ogni x si ha $\| Ax \| = \| x \|$.
  Per come è stata definita la norma di A, si deduce quindi che, se A è ortogonale, allora $\| A \| = 1$. Dato che anche A^{-1} è ortogonale, anche $\| A^{-1} \| = 1$, quindi $\text{cond}(A) = 1$. Le matrici ortogonali sono quindi sempre ben condizionate.

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

$$A = \begin{pmatrix} 3 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 3 \end{pmatrix} \quad b = \begin{pmatrix} 5 \\ 4 \\ -7 \end{pmatrix}$$ Il sistema $Sx - Tx = b$ 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 $(1.14...\ ,\ 1.85...\ ,\ -2.85...)$ 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 possibile, 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 *per righe, 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* diagonalmente dominante *per colonne, se cioè in ogni colonna l'elemento a_{ii} è in modulo strettamente maggiore della somma dei moduli degli altri elementi della colonna.*

In effetti la matrice dell'esempio sopra è diagonalmente dominante sia per righe che per colonne.

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

Se le diseguaglianze nelle due definizioni precedenti non sono strette, se cioè a_{ii} è maggiore o uguale alla somma dei moduli degli altri elementi, allora la matrice si dice *debolmente diagonalmente dominante*. In questo caso la convergenza non è garantita, anche se si verifica in moltissimi casi.

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.
Quindi, dal punto di vista dell'onerosità del calcolo, il metodo di Gauss-Seidel è del tutto equivalente a quello di Jacobi, ma in generale, la convergenza è molto più veloce.

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 di diagonale dominanza, enunciate nel paragrafo precedente, in cui converge quello di Jacobi.
Aggiungiamo che l'algoritmo di Gauss-Seidel converge anche nel caso in cui la matrice A sia simmetrica e definita positiva. Comunque, se la matrice non è diagonalmente dominante, anche nel caso in cui l'algoritmo converga, la convergenza può essere assai lenta.

Per terminare aggiungiamo che i due metodi non sempre convergono, ma convergono in diversi casi che capitano nella pratica.

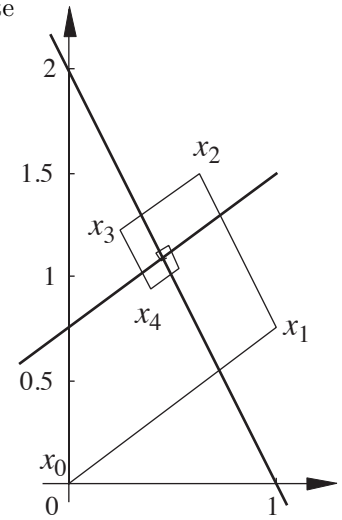
Cenno sul metodo di rilassamento: È possibile accelerare la convergenza di un metodo iterativo, "correggendo" ad ogni passo la soluzione ottenuta in modo da renderla più prossima a quella esatta.
L'idea base è la seguente: se x_{k-1} e x_k sono le soluzioni approssimate ottenute al $(k-1)^{\text{mo}}$ 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 spesso consente di far convergere l'algoritmo di Gauss-Seidel, anche in casi nei casi in cui il metodo base non converge.

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
$\begin{cases} 2x + y = 2 \\ -3x + 4y = 3 \end{cases}$ 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):
$x_1 = (1.0000, 0.7500)$ $x_2 = (0.6250, 1.5000)$
$x_3 = (0.2500, 1.2188)$ $x_4 = (0.3906, 0.9375)$
$x_5 = (0.5312, 1.0430)$ $x_6 = (0.4785, 1.1484)$
$x_7 = (0.4258, 1.1089)$ $x_8 = (0.4456, 1.0693)$
È interessante disegnare le due rette e la spezzata $x_0, x_1, ...$ che converge alla soluzione esatta.



3.1.6 Criteri d'arresto degli algoritmi iterativi

Non è facile dare una stima dell'errore commesso sostituendo alla soluzione esatta di un sistema una soluzione della successione ottenuta con un metodo iterativi e quindi non è facile dare un criterio d'arresto affidabile per un metodo che non fornisce praticamente mai la soluzione esatta.
In pratica ci si accontenta di un valore *stimato* dell'errore.
Se x_i e x_{i+1} fanno parte della successione, si può ritenere che, se si ha

$$\| x_{i+1} - x_i \| < \varepsilon$$

dove ε è un valore prefissato, l'errore assoluto sia minore di ε.
Analogamente, se

$$\frac{\| x_{i+1} - x_i \|}{\| x_{i+1} \|} < \varepsilon$$

si può ritenere che l'errore relativo sia minore di ε.
Il problema è che, come nei metodi di punto fisso in cui la convergenza è monotona, anche in questo caso il criterio non è del tutto affidabile.

Un'altro criterio di arresto è quello per cui si può reputare che x_i sia prossimo alla soluzione se Ax_i e b sono prossimi, se cioè $\| Ax_i - b \| < \varepsilon$, con ε valore prefissato.
Questo criterio è più affidabile dei precedenti, ma richiede un tempo di calcolo superiore.
Nella pratica si possono combinare i due criteri.

3.2 Interpolazione, approssimazione, modellazione

Il problema generale è quello di determinare un'espressione analitica o grafica per una funzione $f(x)$ di cui si conoscono un numero finito di punti del grafico (x_i, y_i).
Quindi si cerca una funzione $f(x)$ tale che

$$f(x_0) = y_0 \quad ; \quad f(x_1) = y_1 \quad ; \quad \cdots \quad ; \quad f(x_n) = y_n$$

x_0	y_0
x_1	y_1
x_2	y_2
$\dots$	$\dots$
x_n	y_n

Si vuole che la $f(x)$ sia *facilmente calcolabile* e che soddisfi le $n+1$ eguaglianze o *precisamente* o *approssimativamente* o *modellandosi* su di esse secondo un concetto che vedremo più avanti. Il problema si pone di solito in uno di questi due casi

- I dati sono ottenuti sperimentalmente, per cui $f(x)$ è da costruire.
- La $f(x)$ è nota ed è possibile calcolarla anche in altri punti, ma non è facilmente calcolabile (per esempio è la soluzione numerica di un'equazione differenziale) o la sua espressione è comunque assai complessa.

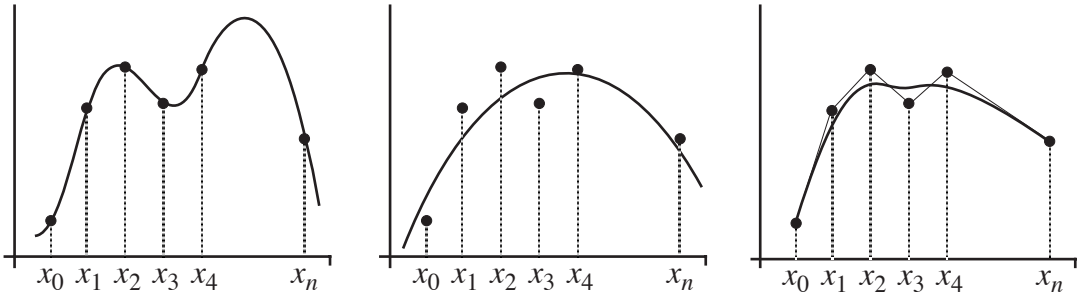
Come detto, le tecniche sono sostanzialmente tre: l'interpolazione, l'approssimazione e la modellazione. Ognuna di esse ha parecchie varianti che possono condurre a risultati diversi. Cerchiamo di dare per ora una definizione intuitiva, che poi approfondiremo, delle tre tecniche.

Interpolare significa determinare una funzione che soddisfi precisamente i dati.

Approssimare significa determinare una funzione che non soddisfi precisamente i dati, ma se ne discosti il meno possibile.

Modellare significa grosso modo determinare una funzione che nel modo più dolce "si inserisca nella poligonale dei dati".

Nel disegno sotto gli stessi dati interpolati, approssimati e "modellati" con qualche tecnica.



3.2.1 Interpolazione polinomiale, la matrice di Vandermonde

La prima idea è quella di determinare una funzione polinomiale

$$P(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$$

con $P(x)$ polinomio di grado minore o uguale a n. Il polinomio si dirà *polinomio interpolatore dei dati.*
Introducendo i dati si ottiene:

$$\left.\begin{array}{lcl} P(x_0) = y_0 & \Rightarrow & a_0 + a_1 x_0 + a_2 x_0^2 + \cdots + a_n x_0^n = y_0 \\ \cdots & & \cdots \\ P(x_n) = y_n & \Rightarrow & a_0 + a_1 x_n + a_2 x_n^2 + \cdots + a_n x_n^n = y_n \end{array}\right\}$$

Questo è un sistema lineare $(n+1)\times(n+1)$ nelle incognite $a_0, ..., a_n$, la cui matrice dei coefficienti è detta *matrice di Vandermonde* della successione $x_0, ..., x_n$. Questa matrice ha determinante diverso da zero se gli x_i sono distinti; pertanto in tal caso esiste un'unico polinomio di grado minore o uguale a n che soddisfa i dati (non è detto che abbia grado esattamente n perché non è detto che si abbia $a_n \neq 0$).

Risolvere il sistema lineare non è però conveniente dal punto di vista calcolativo, sia per la mole dei conti, sia perché la matrice di Vandermonde è particolarmente sensibile agli errori da arrotondamento avendo un numero di condizionamento elevato.

Esempio 3.6: Determinare la parabola $y = a + bx + cx^2$ passante per tre punti (x_0, y_0), (x_1, y_1), (x_2, y_2) (x_i distinti). Si ha il sistema lineare nelle incognite a, b, c:

$$\begin{cases} y_0 = a + bx_0 + cx_0^2 \\ y_1 = a + bx_1 + cx_1^2 \\ y_2 = a + bx_2 + cx_2^2 \end{cases} \quad \text{asociato alla matrice} \quad \left(\begin{array}{ccc|c} 1 & x_0 & x_0^2 & y_0 \\ 1 & x_1 & x_1^2 & y_1 \\ 1 & x_2 & x_2^2 & y_2 \end{array}\right)$$

La matrice dei coefficienti è la matrice di Vandermonde della successione x_0, x_1, x_2.
La soluzione (a, b, c) del sistema fornisce i coefficienti del polinomio. Ricavare il polinomio in questo modo è elementare, ma presenta dei problemi.
Si può per esempio calcolare che per $x_0 = 1$, $x_1 = 2$, $x_2 = 3$ la matrice di Vandermonde ha già numero di condizionamento circa 70.92.

3.2.2 Interpolazione polinomiale, il polinomio di Lagrange

Esiste una semplicissima formula dovuta a Lagrange per determinare il polinomio in questione:

$$P(x) = y_0 \frac{(x-x_1)(x-x_2)\cdots(x-x_n)}{(x_0-x_1)(x_0-x_2)\cdots(x_0-x_n)} +$$
$$+ y_1 \frac{(x-x_0)(x-x_2)\cdots(x-x_n)}{(x_1-x_0)(x_1-x_2)\cdots(x_1-x_n)} + \cdots + y_n \frac{(x-x_0)\cdots(x-x_{n-1})}{(x_n-x_0)\cdots(x_n-x_{n-1})}$$

È evidente che il polinomio ha grado non superiore a n ed è pure evidente il fatto che esso soddisfa i dati.
La formula di Lagrange, benché elegante ed elementare, non è in generale di uso pratico. Il polinomio non è infatti scritto in una forma che si presti a una semplice algoritmizzazione tipo schema di Ruffini-Hörner per calcolare il polinomio in un punto diverso dagli x_i.

3.2.3 Interpolazione polinomiale, il polinomio di Newton

Esiste un'altra formula, dovuta a Newton, per determinare il polinomio in modo algoritmico ed è la seguente:

$$P(x) = b_0 + b_1(x-x_0) + b_2(x-x_0)(x-x_1) + \cdots + b_n(x-x_0)(x-x_1)\cdots(x-x_{n-1})$$

Prima di spiegare come si calcolano i coefficienti b_i, osserviamo che, dopo averli determinati, è facile calcolare $P(x)$ in qualunque punto *senza sviluppare la formula*, in modo simile allo schema di Ruffini-Hörner:

$$P(x) = b_0 + (x-x_0)\Big(b_1 + b_2(x-x_1) + \cdots + b_n(x-x_1)\cdots(x-x_{n-1})\Big)$$

$$P(x) = b_0 + (x-x_0)\bigg(b_1 + (x-x_1)\Big(b_2 + \cdots + b_n(x-x_2)\cdots(x-x_{n-1})\Big)\bigg) \text{ etc.}$$

Per quanto riguarda i coefficienti b_i, un conto non difficile, ma laborioso, mostra che essi si possono determinare ricorsivamente nel modo seguente:

$$\begin{aligned} b_0 &= f(x_0) && \stackrel{\text{def}}{=} f[x_0] \\ b_1 &= \frac{f(x_1)-f(x_0)}{x_1-x_0} && \stackrel{\text{def}}{=} f[x_1, x_0] \\ b_2 &= \frac{f[x_2,x_1]-f[x_1,x_0]}{x_2-x_0} && \stackrel{\text{def}}{=} f[x_2, x_1, x_0] \\ &\cdots \\ b_n &= \frac{f[x_n, x_{n-1}, ..., x_1]-f[x_{n-1}, ..., x_0]}{x_n-x_0} && \stackrel{\text{def}}{=} f[x_n, x_{n-1}, ..., x_0] \end{aligned}$$

I b_i si calcolano quindi in modo algoritmico mediante un procedimento detto *calcolo alle differenze finite*. Esplicitiamo l'algoritmo nel caso in cui gli x_i formino una progressione aritmetica di ragione costante d cioè si abbia:

$$x_0 \qquad x_1 = x_0 + d \qquad x_2 = x_1 + d \qquad \cdots \qquad x_n = x_{n-1} + d$$

In questo caso si possono calcolare i b_i usando lo schema

$$\begin{array}{l|l|l|l|l}
y_0 = y_0^{(0)} & y_1 - y_0 = y_0^{(1)} & y_1^{(1)} - y_0^{(1)} = y_0^{(2)} & \cdots & \\
y_1 & y_2 - y_1 = y_1^{(1)} & y_2^{(1)} - y_1^{(1)} = y_1^{(2)} & \cdots & \\
y_2 & y_3 - y_2 = y_2^{(1)} & & & \\
\cdots & \cdots & \cdots & \cdots & y_1^{(n-1)} - y_0^{(n-1)} = y_0^{(n)} \\
y_{n-1} & & & & \\
y_n & y_n - y_{n-1} = y_{n-1}^{(1)} & & &
\end{array}$$

e, come si verifica subito, si ha:

$$b_0 = y_0^{(0)} \ ; \ b_1 = \frac{y_0^{(1)}}{1!\, d^1} \ ; \ b_2 = \frac{y_0^{(2)}}{2!\, d^2} \ ; \ \ldots \ ; \ b_n = \frac{y_0^{(n)}}{n!\, d^n}$$

Esempio 3.7: Determiniamo il polinomio $P(x)$ di grado minore o uguale a 3 tale che:

$$P(1) = 1 \quad P(1.5) = 3 \quad P(2) = 4 \quad P(2.5) = -1$$

In questo caso $d = 0.5$. Lo schema delle differenze finite unitarie è:

$$\begin{array}{r|r|r|l}
\mathbf{1} & 3 - 1 = \mathbf{2} & 1 - 2 = \mathbf{-1} & \\
3 & 4 - 3 = 1 & -5 - 1 = -6 & -6 + 1 = \mathbf{-5} \\
4 & -1 - 4 = -5 & & \\
-1 & & &
\end{array}$$

Si ha pertanto

$$b_0 = \mathbf{1} \ ; \ b_1 = \frac{\mathbf{2}}{1!(0.5)} = 4 \ ; \ b_2 = \frac{\mathbf{-1}}{2!(0.5)^2} = -2 \ ; \ b_3 = \frac{\mathbf{-5}}{3!(0.5)^3} = -20/3$$

Il polinomio è quindi:

$$P(x) = 1 + 4(x-1) - 2(x-1)(x-1.5) - \frac{20}{3}(x-1)(x-1.5)(x-2)$$

Se la successione non è a passo costante, lo schema delle differenze finite subisce una semplice modifica che non riportiamo in questa sede.

3.2.4 Il resto nell'interpolazione di Newton

È evidente l'analogia tra la formula di interpolazione di Newton e la nota formula di Taylor. In effetti, come il polinomio di Taylor approssima una funzione con un polinomio avente stesso valore e stesse derivate in un punto x_0, il polinomio di Newton approssima una funzione con un polinomio che assume nei punti x_i gli stessi valori della funzione.
In analogia alla formula del resto di Lagrange per il polinomio di Taylor, si ha:

Proposizione 10 *Sia $f(x)$ una funzione continua nell'intervallo $[a, b]$, ivi dotata di derivate continue fino all'ordine $n + 1$.*
Se $a = x_0 < x_1 < \cdots < x_n = b$ è una suddivisione dell'intervallo e P(x) è il polinomio di Newton che interpola *$f(x)$ nei punti x_i (nel senso che $P(x_i) = f(x_i)$ per ogni i), allora, per ogni $x \in [a, b]$ esiste un punto ξ nell'intervallo $[a, b]$ tale che:*

$$f(x) = P(x) + \frac{f^{(n+1)}(\xi)}{(n+1)!}(x - x_0) \cdots (x - x_n)$$

3.2.5 Interpolazione "spline"

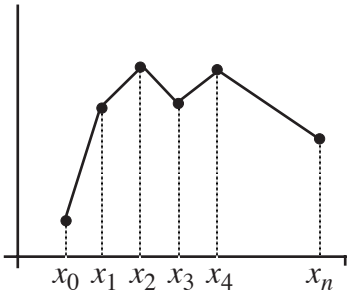
L'interpolazione polinomiale può non essere conveniente per vari motivi. Il primo è che per una lunga serie di dati il polinomio risulta di grado troppo alto, il secondo è che il problema è comunque mal condizionato, nel senso che basta una piccola variazione dei dati per cambiare anche di parecchio i coefficienti del polinomio.

Un modo spesso più efficiente e perciò maggiormente diffuso per interpolare una serie di dati consiste nell'usare una funzione definita a pezzi i cui pezzi siano polinomi di grado basso. Questa interpolazione si dice "spline" (dal nome inglese delle bacchette di legno flessibile usate per l'interpolazione meccanica di una serie di dati).

Si abbia la solita serie di dati da interpolare:

$$f(x_0) = y_0 \ , \ \dots \ , \ f(x_n) = y_n \quad \text{con} \quad x_0 < \cdots < x_n$$

Descriveremo tre tipi di interpolazione spline.

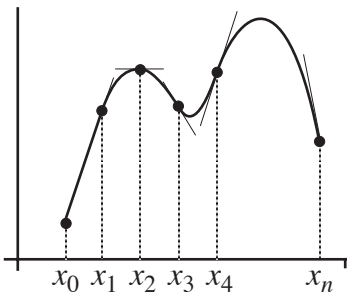


Interpolazione spline lineare

La più semplice interpolazione spline è quella mediante funzioni lineari. Si può scrivere per ogni i $(1 \leq i \leq n)$ l'equazione della retta passante per i due punti $(x_{i-1}, y_{i-1})\,(x_i, y_i)$.
La funzione così definita a pezzi sugli intervalli $[x_{i-1}, x_i]$ è evidentemente continua in $[x_0, x_n]$ e soddisfa le condizioni date.

Interpolazione spline quadratica

Scriviamo per ogni i $(1 \leq i \leq n)$ le parabole e cioè le funzioni del tipo $y = a_i + b_i x + c_i x^2$ passanti per i due punti $(x_{i-1}, y_{i-1})\,(x_i, y_i)$. Ne esistono ∞^1 per ogni i, quindi n dei $3n$ coefficienti sono arbitrari. Si può approfittare di questo fatto per imporre che le derivate prime delle parabole coincidano nei punti $x_1, \dots x_{n-1}$ e quindi la funzione sia dotata di derivata prima.



Queste sono $n-1$ condizioni su n parametri. Resta pertanto una scelta arbitraria.È uso imporre che la parabola dell'intervallo $[x_0, x_1]$ degeneri in una retta perché questo modo è facile costruire successivamente le varie parabole.
Si ha pertanto una funzione definita a pezzi sugli intervalli $[x_{i-1}, x_i]$ che è evidentemente continua e derivabile in $[x_0, x_n]$ e soddisfa i dati.
La spline quadratica non è molto usata perché spesso fornisce un risultato "saltellante" e quindi poco soddisfacente.

Interpolazione spline cubica

La più usata delle interpolazioni spline è quella con funzioni polinomiali di grado tre in quanto consente un calcolo semplice e una approssimazione più che soddisfacente. Inoltre si riesce a fare in modo che la funzione sia di classe C^2.

Si determina per ogni i $(1 \leq i \leq n)$ la funzione cubica ovvero del tipo $y = a_i + b_i x + c_i x^2 + d_i x^3$ passante per i due punti $(x_{i-1}, y_{i-1})\,(x_i, y_i)$. Il problema ha ∞^2 soluzioni per ogni i $(i = 1, \dots, n)$, quindi $2n$ dei $4n$ coefficienti sono arbitrari.
Imponendo che sia le derivate prime che quelle seconde coincidano nei punti $x_1, \dots x_{n-1}$ si hanno altre $2n-2$ condizioni lineari; rimangono ancora due scelte arbitrarie ed è uso imporre che la prima e l'ultima parabola cubica abbiano un flesso rispettivamente in x_0 e in x_n (spline naturale). A volte si danno due condizioni sulle derivate prime in x_0 e in x_n (spline vincolata).

Si ha pertanto una funzione definita a pezzi sugli intervalli $[x_{i-1}, x_i]$ che è evidentemente continua e derivabile due volte in $[x_0, x_n]$ e soddisfa i dati.

Accenniamo brevemente al procedimento per calcolare in maniera relativamente veloce la spline cubica naturale.

L'idea base è quella di fare in modo che le incognite siano solo le derivate seconde delle parabole cubiche nei punti $x_1, ..., x_{n-1}$. Poniamo per semplicità di notazione:

$$q_1 = f''(x_1)\ ,\ \dots\ ,\ q_{n-1} = f''(x_{n-1})$$

Scriviamo la *derivata seconda* della funzione cubica $f_i(x)$ che congiunge il punto (x_{i-1}, y_{i-1}) col punto (x_i, y_i). È una funzione lineare che possiamo scrivere così (usando la formula di Lagrange) in modo da evidenziare i valori che la derivata seconda stessa assume nei punti x_i:

$$f_i''(x) = q_{i-1}\frac{x - x_i}{x_{i-1} - x_i} + q_i\frac{x - x_{i-1}}{x_i - x_{i-1}}$$

Integriamo due volte rispetto a x e scriviamo opportunamente le due costanti di integrazione h_i e k_i ottenendo così le $f_i(x)$:

$$f_i(x) = \frac{q_{i-1}(x - x_i)^3}{6(x_{i-1} - x_i)} + \frac{q_i(x - x_{i-1})^3}{6(x_i - x_{i-1})} + h_i(x_i - x) + k_i(x - x_{i-1})$$

Le due costanti così scritte h_i e k_i si determinano imponendo che $f_i(x_i) = y_i$ e che $f_i(x_{i-1}) = y_{i-1}$. Svolti i conti si ottiene:

$$h_i = \frac{y_{i-1}}{x_i - x_{i-1}} - \frac{q_{i-1}(x_i - x_{i-1})}{6} \qquad k_i = \frac{y_i}{x_i - x_{i-1}} - \frac{q_i(x_i - x_{i-1})}{6}$$

Rimangono da determinare tutti i q_i. Imponendo che le derivate prime coincidano in tutti i punti x_i $(i \neq 0, n)$ si ottiene:

$$(x_i - x_{i-1})q_{i-1} + 2(x_{i+1} - x_{i-1})q_i + (x_{i+1} - x_i)q_{i+1} = 6\left(\frac{y_{i+1} - y_i}{x_{i+1} - x_i} + \frac{y_{i-1} - y_i}{x_i - x_{i-1}}\right)$$

Queste sono $n - 1$ relazioni lineari tra i q_i. Se si impone che la spline sia naturale, si ha $q_0 = q_n = 0$; le incognite sono quindi $n - 1$ e la matrice delle $n - 1$ relazioni lineari è tridiagonale, per cui la risoluzione del sistema è particolarmente agevole. Inoltre la forma delle singole $f_i(x)$ è particolarmente adatta al calcolo con uno schema tipo Ruffini-Hörner.

Nel caso particolarmente frequente in cui gli x_i siano in progressione aritmetica di ragione d, il sistema nelle incognite $q_1, ..., q_{n-1}$ è associato alla matrice tridiagonale simmetrica

$$\left(\begin{array}{ccccccc|c} 4d & d & 0 & 0 & \cdots & 0 & 0 & 6(y_0 - 2y_1 + y_2)/d \\ d & 4d & d & 0 & \cdots & 0 & 0 & 6(y_1 - 2y_2 + y_3)/d \\ 0 & d & 4d & d & \cdots & 0 & 0 & 6(y_2 - 2y_3 + y_4)/d \\ \cdots & & & & & & & \dots \\ 0 & 0 & 0 & 0 & & d & 4d & 6(y_{n-2} - 2y_{n-1} + y_n)/d \end{array}\right)$$

La spline cubica naturale è in un certo senso la miglior funzione di classe C^2 che interpoli i dati. Più precisamente si dimostra che:

Proposizione 11 *Siano $x_0 < x_1 < \cdots < x_n$. Consideriamo tutte le funzioni $g(x)$ di classe C^2 nell'intervallo $[x_0, x_n]$, tali che $g(x_i) = y_i$, per $i = 1, ..., n$; allora la quantità*

$$\int_{x_0}^{x_n} (g''(x))^2 dx$$

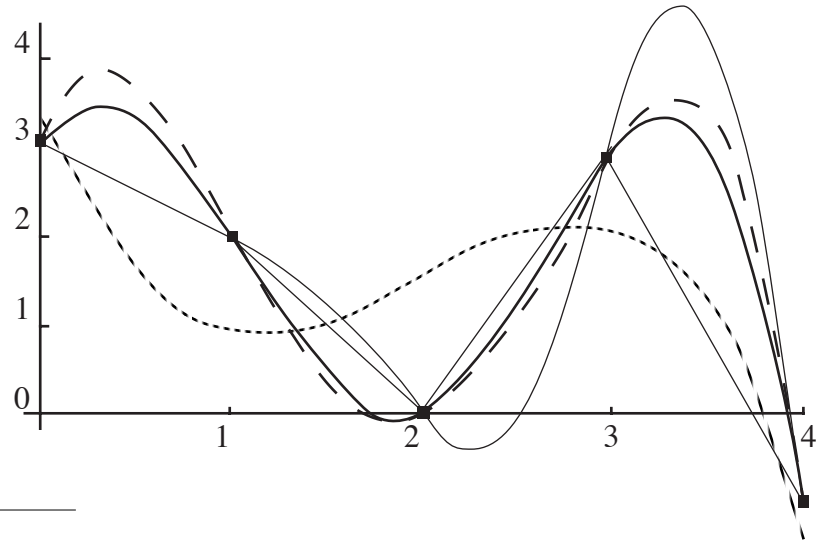
è minima qualora $g(x)$ sia la spline cubica naturale.

Esempio 3.8: Un esempio di costruzione di interpolazioni polinomiali con splines o altro comporterebbe solo una serie di lunghi calcoli e non sarebbe di aiuto alla comprensione dei metodi esposti. Ci accontentiamo perciò di fornire un grafico elaborato al calcolatore che illustra varie interpolazioni polinomiali della serie di dati sotto.

Nel grafico (che per motivi di chiarezza non è monometrico), sono disegnati con vari tipo di tratto:

- La spline lineare che interpola i dati.
- La spline quadratica che interpola i dati (il primo tratto coincide con quello lineare)
- La spline cubica che interpola i dati.
- Il polinomio di Newton (quindi di grado 4) che interpola i dati.
- Il polinomio di grado 3 che *approssima* i dati *ai minimi quadrati* (vedi paragrafo successivo).

x	y
0	3
1	2
2	0
3	3
4	−1



3.2.6 Approssimazione ai minimi quadrati

Invece di trovare un polinomio *di grado* n che interpoli $n+1$ dati, se ne può trovare uno *di grado inferiore* che non passi esattamente per i punti dati, ma se ne discosti per poco "nel senso dei minimi quadrati".

Più precisamente non si pretende che il polinomio $P(x)$ di grado $d \leq n$ soddisfi precisamente le eguaglianze $P(x_0) = y_0 \quad ; \quad P(x_1) = y_1 \quad ; \cdots\cdots ; \quad P(x_n) = y_n$, ma ci si accontenta che la quantità

$$\Big(P(x_0) - y_0\Big)^2 \quad + \quad \Big(P(x_1) - y_1\Big)^2 + \cdots\cdots + \Big(P(x_n) - y_n\Big)^2$$

sia la minima possibile.

Si dimostra che esiste unico un polinomio di grado $d < n$ con questa proprietà ed è detto polinomio che approssima i dati *ai minimi quadrati*.

Questo modo di approssimare i dati è usato soprattutto quando i dati sono frutto di osservazioni sperimentali e quindi soggetti a probabile errore. Particolarmente noto è il caso in cui il polinomio ha grado 1, quindi si ha l'approssimazione lineare ai minimi quadrati.

Ci sarebbe molto da dire, ma ci accontentiamo di riportare la tecnica più semplice per trovarlo (anche se non è la più numericamente stabile). Come nel caso di Vandermonde si cerca un polinomio

$$P(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_d x^d$$

di grado minore o uguale a d. Introducendo i dati si ottiene:

$$\left.\begin{array}{lcll} P(x_0) = y_0 & \Rightarrow & a_0 + a_1 x_0 + a_2 x_0^2 + \cdots + a_d x_0^d & = \quad y_0 \\ \cdots & & \cdots & \\ P(x_n) = y_n & \Rightarrow & a_0 + a_1 x_n + a_2 x_n^2 + \cdots + a_d x_n^d & = \quad y_n \end{array}\right\}$$

Osserviamo che questa volta la matrice dei coefficienti A, dato che $d \leq n$, è una matrice rettangolare con più righe che colonne e che quindi il sistema $Au = y$ è un sistema con più equazioni che incognite

e quindi quasi sicuramente senza soluzioni. La *soluzione ai minimi quadrati* è però l'unica soluzione del sistema quadrato con matrice invertibile

$$A^T A\, u = A^T\, y$$

dove la matrice $A^T A$ è una matrice $d \times d$ simmetrica. Se il polinomio cercato è di grado 1, la matrice è 2×2 e in questo caso la retta è la famosa *regressione lineare* spesso usata in statistica.

Esempio 3.9: Determinare la retta e la parabola $y = a + bx + cx^2$ che approssimino ai minimi quadrati i dati a lato.

1	0
3	3
4	2
6	3

Per l'approssimazione lineare si scrivono la matrice di Vandermonde dei numeri $1, 3, 4, 6$ arrestata alla potenza 1 e la matrice dei dati y in colonna:

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 3 \\ 1 & 4 \\ 1 & 6 \end{pmatrix} \quad y = \begin{pmatrix} 0 \\ 3 \\ 2 \\ 3 \end{pmatrix}$$

Il sistema lineare $Au = y$ non ha soluzioni, ma il sistema lineare 2×2 $(A^T A)\, u = (A^T\, y)$ ha l'unica soluzione $u = (0.1154\ ,\ 0.5385)$. Quindi la retta $y = 0.1154 + 0.5385\, x$ approssima i dati ai minimi quadrati.

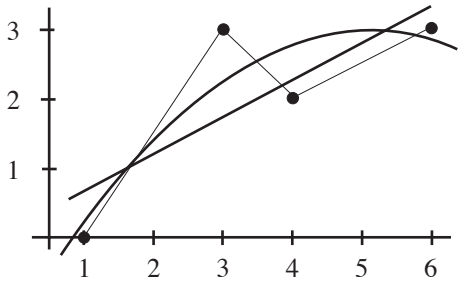
Analogamente, per l'approssimazione quadratica si scrivono la matrice di Vandermonde dei numeri $1, 3, 4, 6$ arrestata ai quadrati e la matrice dei dati y in colonna:

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 3 & 9 \\ 1 & 4 & 4 \\ 1 & 6 & 36 \end{pmatrix} \quad y = \begin{pmatrix} 0 \\ 3 \\ 2 \\ 3 \end{pmatrix}$$

Come sopra, ma il sistema lineare 3×3 $(A^T A)\, u = (A^T\, y)$ ha l'unica soluzione $u = (-1.3846\ 1.7051\ -0.1667)$.

Quindi la parabola $y = -1.3846 + 1.7051\, x - 0.1667\, x^2$ approssima i dati ai minimi quadrati.

Nel disegno sono riportati i punti, la retta e la parabola.



3.3 Curve di Bézier e B-spline

Se invece di interpolare o approssimare i dati, vogliamo "modellare" una curva sui dati, lo strumento base è quello delle curve di Bézier.

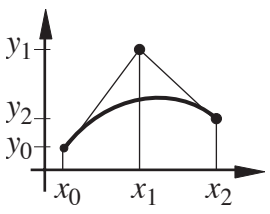
Grosso modo *modellare* significa determinare una curva "dolce" che si inserisca nella poligonale, detta *poligono di controllo* che interpola i punti dati.

3.3.1 Polinomi di Bézier

Iniziamo con le *funzioni* polinomiali di Bézier. Esse rappresentano un caso assai particolare e limitato, ma costituiscono in un certo senso il passaggio dall'interpolazione alla modellazione. Queste funzioni "modellano", *mediante una funzione*, dati del tipo $f(x_0) = y_0\ ;\ ...\ ;\ f(x_n) = y_n$ nel caso in cui $x_0, x_1, x_2, ...$ sia una successione a passo costante.

Polinomio quadratico di Bézier

Siano x_0, x_1, x_2 tre numeri *equidistanti* e y_0, y_1, y_2 tre numeri qualunque. Chiamiamo P_0 il punto (x_0, y_0) e così via.
Esiste una e una sola funzione quadratica $f(x) = ax^2 + bx + c$, il cui grafico è una parabola passante per P_0, passante per P_2, tangente alla retta $\overline{P_0\, P_1}$ in P_0 e tangente alla retta $\overline{P_1\, P_2}$ in P_2.

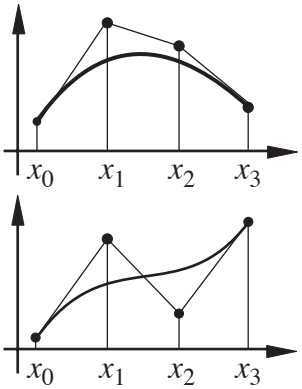


In realtà sembra strano che esista una parabola che soddisfi 4 condizioni, perché i coefficienti sono solo tre, ma ciò è dovuto al fatto che x_1 è il punto medio tra x_0 e x_2.
Se P_0, P_1, P_2 sono allineati, la parabola degenera in una retta.

Polinomio cubico di Bézier

Siano x_0, x_1, x_2, x_3 quattro numeri *equidistanti* e y_0, y_1, y_2, y_3 quattro numeri.
Esiste una e una sola *funzione cubica* $f(x) = ax^3 + bx^2 + cx + d$, passante per i punti $P_0(x_0, y_0), ..., P_3(x_3, y_3)$ e tangente alla retta $\overline{P_0\,P_1}$ in P_0 e alla retta $\overline{P_2\,P_3}$ in P_3.
La cubica è unica perché i coefficienti sono quattro di fronte a quattro condizioni indipendenti e potrebbe anche avere un flesso o degenerare in una parabola quadratica o in una retta.

È possibile costruire allo stesso modo i polinomi di Bézier di grado n che passano per $n+1$ punti *equidistanti* anche se non è facile dare un'interpretazione geometrica delle proprietà di queste curve.



La costruzione dei polinomi di Bézier viene di solito effettuata mediante una delle due tecniche seguenti: quella analitica (i polinomi di Bernstein) o quella grafica (l'algoritmo di de Casteljau).

Ci limitiamo per ora a introdurre i polinomi di Bernstein, riservando la descrizione dell'algoritmo di de Casteljau al caso ben più interessante delle *curve di Bézier*.

3.3.2 I polinomi di Bernstein

Sia $[a, b]$ un intervallo della retta reale e sia $n \geq 2$. Definiamo i polinomi di Bernstein di grado n nell'intervallo $[a, b]$.
Gli $n + 1$ polinomi di Bernstein di grado n si denotano normalmente con $B_i^n(x)$, ma li scriveremo semplicemente $B_i(x)$ per non appesantire la notazione.

Definizione: I polinomi di Bernstein di grado n nell'intervallo $[a, b]$ sono

$$B_i(x) = \binom{n}{i} \frac{(b-x)^{n-i}(x-a)^i}{(b-a)^n} \qquad i = 0, 1, ..., n$$

I polinomi di Bernstein di grado n dipendono dall'intervallo $[a, b]$ e si ha:

$$B_0(a) = 1 \quad \text{e} \quad B_i(a) = 0 \text{ per } i > 0 \qquad B_i(b) = 0 \text{ per } i < n \quad \text{e} \quad B_n(b) = 1$$

I polinomi di Bernstein costituiscono una base per lo spazio vettoriale costituito dai polinomi di grado $\leq n$, nel senso che ogni altro polinomio di grado minore o uguale a n si può scrivere *in modo unico* come loro combinazione lineare.

Si ha poi: $B_0(x) + B_1(x) + \cdots + B_n(x) \equiv 1$ (per ogni x).

Il polinomio di Bézier di grado n generato dai punti (x_0, y_0) , ... , (x_n, y_n) (x_i equidistanti) è, come si può facilmente verificare, combinazione lineare a coefficienti $y_0, ..., y_n$ dei polinomi di Bernstein di grado n nell'intervallo $[a, b] = [x_0, x_n]$

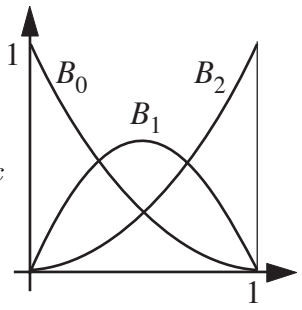
$$\text{Bez}(x) = y_0 B_0(x) + \cdots + y_n B_n(x)$$

Osservazione: Nel seguito, per generare le curve di Bézier, useremo esclusivamente i polinomi di Bernstein nell'intervallo $[0, 1]$.

I polinomi di Bernstein di grado 2 e di grado 3 in $[0, 1]$ sono

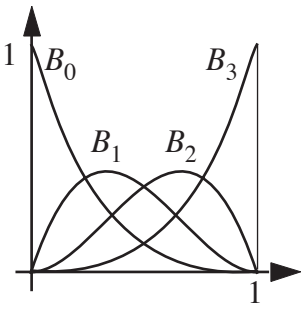
Grado 2:

$B_0(x) = (1-x)^2$
$B_1(x) = 2(1-x)x$
$B_2(x) = x^2$



Grado 3:

$B_0(x) = (1-x)^3$
$B_1(x) = 3(1-x)^2x$
$B_2(x) = 3(1-x)x^2$
$B_3(x) = x^3$



Osserviamo che: $B_0(x) + B_1(x) + B_2(x) \equiv 1$ (vale 1 per ogni x) nel caso quadratico.

Analogamente: $B_0(x) + B_1(x) + B_2(x) + B_3(x) \equiv 1$ nel caso cubico.

3.3.3 Le curve di Bézier

Svincoliamoci ora dall'ipotesi che $x_0, x_1, ..., x_n$ siano equidistanti. Esiste una curva, detta *curva di Bézier*, che modella i punti, anche se si non può pretendere che sia una semplice funzione polinomiale $y = a_0 + a_1x + \cdots + a_nx^n$.

Occorre lavorare bidimensionalmente ed esprimere la curva in forma parametrica.

Si può supporre che le funzioni $x(t)$ e $y(t)$ siano definite nell'intervallo $[0, 1]$ e che in 0 e in 1 assumano i valori (x_0, y_0) e (x_n, y_n) rispettivamente.

$$\begin{cases} x = x(t) \\ y = y(t) \end{cases}$$

La successione dei punti $P_0(x_0, y_0), ... , P_n(x_n, y_n)$ verrà detta *poligono di controllo* della curva di Bézier.

A questo punto non è più neanche necessario che gli x_i siano né ordinati né distinti, basta che siano distinti i punti $P_i(x_i, y_i)$.

Siano quindi $P_0(x_0, y_0)$, ... , $P_n(x_n, y_n)$, $n + 1$ punti distinti nel piano.

Definizione: È detta *curva di Bézier* generata dal poligono di controllo (x_0, y_0), ... , (x_n, y_n) la curva avente come rappresentazione parametrica

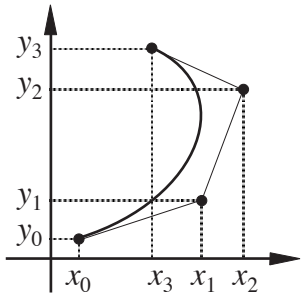
$$\begin{cases} x(t) = x_0B_0(t) + \cdots + x_nB_n(t) \\ y(t) = y_0B_0(t) + \cdots + y_nB_n(t) \end{cases}$$

dove i $B_i(t)$ sono i polinomi di Bernstein di grado n nell'intervallo $[0, 1]$.

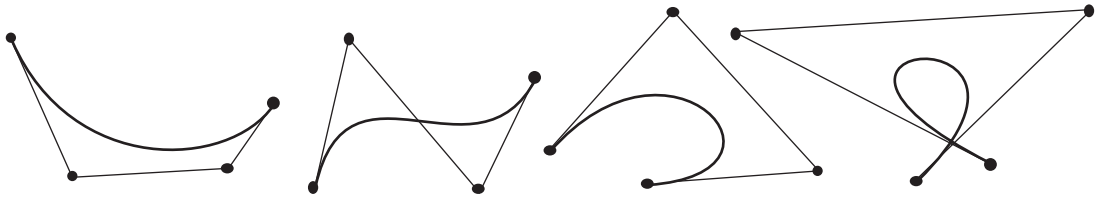
Per esempio nel caso cubico, che è anche uno dei più usati in pratica, il poligono di controllo sarà costituito da quattro punti (x_0, y_0), (x_1, y_1), (x_2, y_2), (x_3, y_3) e la curva avrà rappresentazione parametrica

$$\begin{cases} x(t) = x_0B_0(t) + x_1B_1(t) + x_2B_2(t) + x_3B_3(t) \\ y(t) = y_0B_0(t) + y_1B_1(t) + y_2B_2(t) + y_3B_3(t) \end{cases} \qquad t \in [0, 1]$$

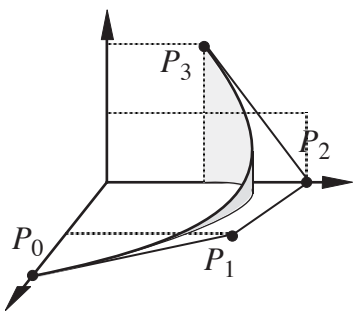
dove i $B_i(t)$ sono i polinomi cubici di Bernstein definiti nell'intervallo $[0, 1]$.



Qui di seguito alcuni esempi di curve di Bézier cubiche con il loro poligono di controllo; la quarta è addiritura nodata, cosa che può capitare se il poligono è intrecciato, del resto anche la terza è nodata, anche se il nodo cade esternamente alla porzione utile.



Questa generalizzazione permette anche di costruire curve di Bézier nello spazio. Si dovrà aggiungere una terza funzione $z(t)$, ma tutto funziona esattamente come nel caso planare. Si tenga presente che, mentre una curva di Bézier quadratica è sempre un arco di parabola e perciò una curva piana giacente nel piano dei tre punti del poligono di controllo, una curva di Bézier cubica può essere una curva sghemba e quindi dotata di vera torsione tridimensionale se i quattro punti del poligono di controllo non sono complanari.



3.3.4 Algoritmo di de Casteljau per costruire curve di Bézier

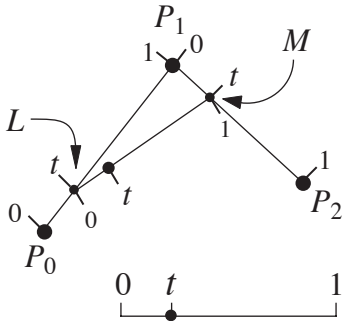
L'algoritmo di de Casteljau permette di costruire quanti punti si vuole di una curva di Bézier con un semplice procedimento che ha un'immediata interpretazione grafica.
Il concetto base è semplicemente la parametrizzazione segmentaria della retta. Data una retta r passante due punti A e B, si considera la parametrizzazione

$$P(t) = A + t(B - A)$$

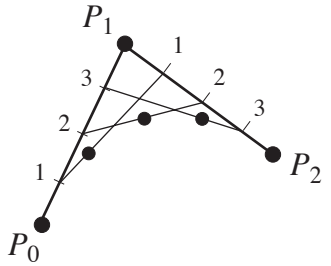
che pone il segmento $\overline{AB}$ in corrisponenza biunivoca con l'intervallo $[0, 1]$ e fornisce A per $t = 0$ e B per $t = 1$.

Cominciamo col caso quadratico

Il poligono di controllo sarà costituito da tre punti P_0, P_1, P_2.
Si pone il segmento $\overline{P_0\,P_1}$ in corrispondenza biunivoca coll'intervallo $[0, 1]$ in modo che P_0 corrisponda a 0 e P_1 a 1, ovvero con la parametrizzazione $P(t) = P_0 + t(P_1 - P_0)$. Allo stesso modo anche $\overline{P_1\,P_2}$ è posto in corrispondenza con $[0, 1]$.
Si fissa un numero t compreso tra 0 e 1 e si considerano sui segmenti $P_0\,P_1$ e $P_1\,P_2$ i due punti corrispondenti a t che chiamiamo per ora L e M.
Si costruisce il segmento che ha come estremi questi due punti L e M e lo si pone in corrispondenza biunivoca coll'intervallo $[0, 1]$. Il punto del segmento LM corrispondente a t fa parte della curva quadratica di Bézier. Facendo variare t nell'intervallo $[0, 1]$ si ottengono tutti i punti della parabola.

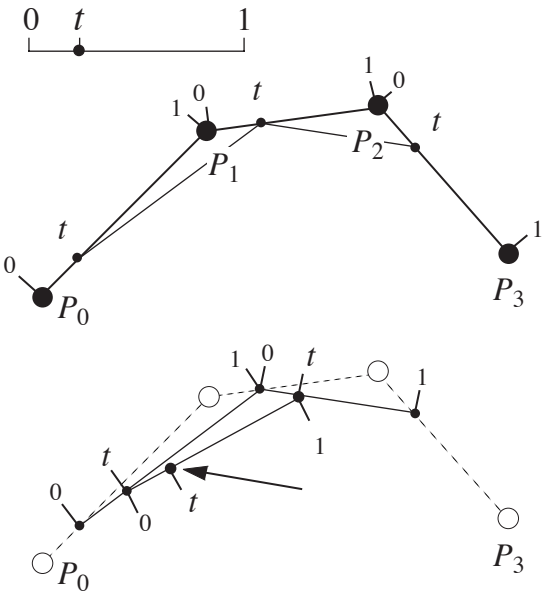


Esempio 3.10: I due segmenti sono stati divisi in 4 parti uguali, quindi si usano i tre valori $t = 1/4, 2/4, 3/4$ compresi tra 0 e 1. I punti su P_0P_1 e su P_1P_2 sono stati chiamati $1, 2, 3$. I segmenti $\overline{1\,1}, \overline{2\,2}, \overline{3\,3}$ sono posti in corrispondenza con $[0, 1]$ e sul segmento $\overline{1\,1}$ viene considerato il punto corrispondente a $t = 1/4$, sul segmento $\overline{2\,2}$ il punto corrispondente a $t = 2/4$, sul segmento $\overline{3\,3}$ il punto corrispondente a $t = 3/4$. I 5 punti così trovati (si aggiungono i due estremi) appartengono alla parabola di Bézier.



Proseguiamo col caso cubico

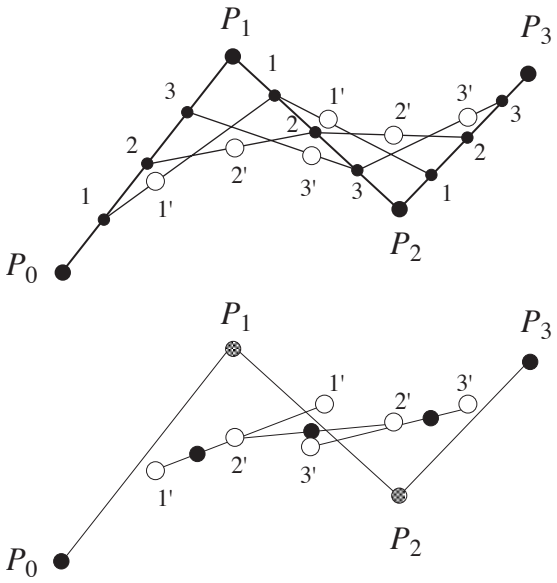
Il poligono di controllo sarà costituito da quattro punti P_0, P_1, P_2, P_3.
Come nel caso quadratico, si pongono i segmenti $P_0\,P_1$, $P_1\,P_2$, $P_2\,P_3$ in corrispondenza biunivoca coll'intervallo $[0, 1]$ in modo che rispettivamente P_0 corrisponda a 0, P_1 a 1 etc.
Si fissa un numero t compreso tra 0 e 1 e si cercano sui segmenti $P_0\,P_1$, $P_1\,P_2$, $P_2\,P_3$ i tre punti corrispondenti a t.
Si costruiscono quindi i due segmenti che hanno come estremi questi tre punti nell'ordine. Come nella seconda figura si pongono i due segmenti in corrispondenza biunivoca coll'intervallo $[0, 1]$. A questo punto si prosegue come per l'algoritmo di de Casteljau nel caso quadratico, cercando sui due segmenti i punti corrispondenti a t e congiungendoli con un segmento che va posto in corrispondenza biunivoca con $[0, 1]$. In corrispondenza di questo t si ha il punto della cubica di Bézier.



Esempio 3.11: È dato un poligono di controllo P_0, P_1, P_2, P_3. Si dividono i lati del poligono di controllo in 4 parti uguali, ovvero si usano tre valori $t = 1/4, 2/4, 3/4$ compresi tra 0 e 1. I punti vengono chiamati egualmente $1, 2, 3$ su ciascuno dei tre segmenti.

I 6 segmenti $\overline{1\,1}$, $\overline{2\,2}$, $\overline{3\,3}$, $\overline{1\,1}$, $\overline{2\,2}$, $\overline{3\,3}$ vengono divisi in 4 parti uguali, ma su ciascuno dei due segmenti $\overline{1\,1}$ viene considerato il punto corrispondente a $t = 1/4$ e abbiamo i due punti di nome $1'$. Su ciascuno dei due segmenti $\overline{2\,2}$ viene considerato il punto corrispondente a $t = 2/4$ e abbiamo i due punti di nome $2'$. Su ciascuno dei due segmenti $\overline{3\,3}$ viene considerato il punto corrispondente a $t = 3/4$ e abbiamo i due punti di nome $3'$. A questo punto consideriamo il secondo disegno identico al primo, ma dove, per chiarezza, sono stati eliminati i segmenti $\overline{1\,1}$ etc.

I segmenti $\overline{1'\,1'}$, $\overline{2'\,2'}$, $\overline{3'\,3'}$ vengono divisi in 4 parti uguali, ma sul segmento $\overline{1'\,1'}$ viene considerato il punto corrispondente a $t = 1/4$, sul segmento $\overline{2'\,2'}$ il punto per $t = 2/4$ e sul segmento $\overline{3'\,3'}$ quello per $t = 3/4$. I 5 punti così trovati (si aggiungono i due estremi P_0 e P_3) appartengono alla cubica di Bézier.



3.3.5 Le curve B-spline

Se il poligono da modellare è costituito da molti punti, non conviene costruire una curva di Bézier di grado elevato, ma è meglio costruire diverse curve di Bézier di ordine basso (3 è il più usato) e raccordarle insieme nel modo migliore possibile.
La curva ottenuta in questo modo e che quindi è una curva costituita da varie curve di Bézier è detta *curva B-spline*.

La teoria è assai vasta; ci limitiamo ai due casi più semplici, le B-spline quadratiche uniformi e non uniformi e le B-spline cubiche uniformi e non, avvertendo che anche sulle curve non uniformi si possono fare variazioni di rilievo rispetto alla semplice trattazione che segue.

3.3.6 Le B-spline quadratiche

La cosa più complicata è capire quale uso fare dei dati iniziali, perché la B-spline modella una serie di punti, senza necessariamente passare per essi, ma "raddolcendo" il loro andamento.

Nel caso più elementare sono assegnati $n + 1$ punti distinti (o meglio tali che tre consecutivi siano distinti) che costituiscono il cosiddetto *poligono di de Boor*

$$P_0, P_2, P_4, ..., P_{2n}$$

I punti del poligono hanno indici pari. Definiremo ora i punti P_i con i dispari e costruiremo una B-spline in cui ogni pezzo è una curva quadratica di Bézier con poligono di controllo $P_{2i-1}, P_{2i}, P_{2i+1}$ (il centrale ha indice pari).
Per costruire i punti di indice dispari esistono vari criteri.

Caso uniforme

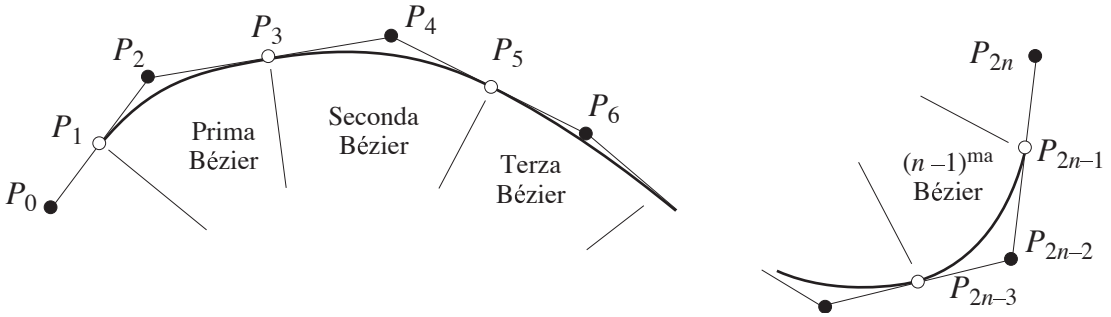
Nel caso più elementare i punti di indice dispari saranno semplicemente i punti medi

Quindi porremo: $P_1 = \dfrac{P_0 + P_2}{2} \quad P_3 = \dfrac{P_2 + P_4}{2} \quad \cdots \quad P_{2n-1} = \dfrac{P_{2n-2} + P_{2n}}{2}$

e costruiremo le $n-2$ curve di Bézier con poligono di controllo $P_{2i-1}, P_{2i}, P_{2i+1}$ mediante i polinomi di Bernstein o l'algoritmo di de Casteljau. Vedremo nel caso non uniforme che è possibile parametrizzare tutta la B-spline partendo un apposito intervallo esattamente come l'algoritmo di de Casteljau parametrizza una curva di Bézier usando l'intervallo $[0, 1]$.

La B-spline così costruita è detta *B-spline quadratica uniforme*. L'aggettivo uniforme si riferisce al fatto che i punti di indice dispari sono presi come *punti medi* dei segmenti.

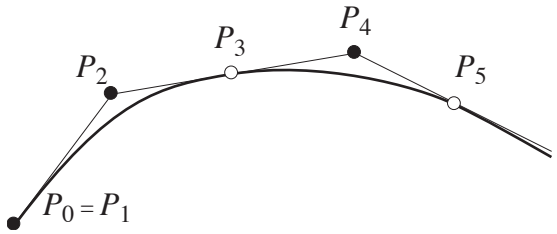
La B-spline è continua e di classe C^1 per costruzione.



Osservazioni: :

1. La B-spline quadratica ha un controllo *semi-locale* dei punti, nel senso che cambiando uno dei P_i subiscono variazioni solo due pezzi di Bèzier della curva e non l'intera curva.

2. La curva non passa per i punti iniziali della spezzata. Se si vuole ottenere questo, si può agire in due modi: o, come fanno alcuni, facendo semplicemente coincidere P_1 con P_0 e P_{2n-1} con P_{2n} (e definendo tutti gli altri punti come sopra) oppure mediante l'uso di opportune successioni nodali come vediamo nel seguito.



Caso non uniforme e successioni nodali

Invece di prendere i punti medi dei segmenti $\overline{P_0\,P_2}$ etc., si possono prendere altri punti più o meno distanti dagli estremi ed avere una curva più o meno aderente al poligono di de Boor e quindi più adatta a certe esigenze. Si tenga presente che cambiando un punto si cambiano solo due curve di Bézier della B-spline, e si mantiene quindi un controllo *semi-locale* su tutta la curva.
Potremmo semplicemente dire quale punto prendiamo su ogni segmento, ma conviene introdurre le *successioni nodali*, perché più utili per il seguito e indispensabili, come vedremo, nel caso delle B-spline cubiche.

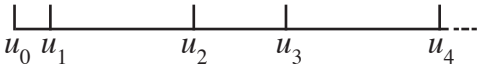
Supponiamo di avere il poligono di de Boor $P_0, P_2, ..., P_{2n}$ costituito da $n+1$ punti. Si chiama successione nodale una successione non decrescente di $n+2$ numeri $u_0 \le u_1 \le \cdots \le u_{n+1}$.
Per definire i punti $P_1, P_3, ...$ si adopera la successione nodale nel seguente modo:

$$P_1 = \frac{(u_2 - u_1)P_0 + (u_1 - u_0)P_2}{u_2 - u_0} \qquad P_3 = \frac{(u_3 - u_2)P_2 + (u_2 - u_1)P_4}{u_3 - u_1} \qquad \text{etc.}$$

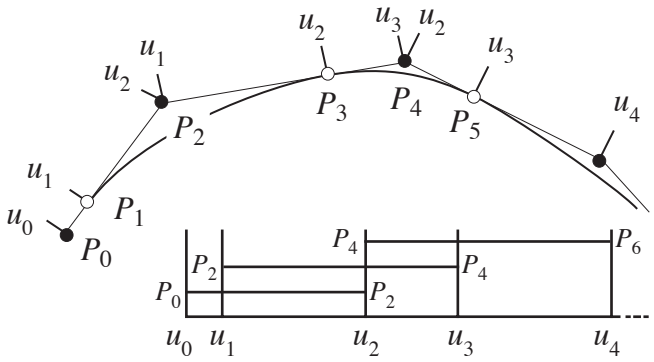
Notiamo che, se la successione nodale è a passo costante, per esempio $0 < 1 < 2 < 3 < \cdots$, si riottiengono gli stessi P_i del caso uniforme.

Per illustrare graficamente il funzionamento della successione nodale, conviene visualizzare la

successione mediante un righello.



I segmenti del poligono di de Boor vengono messi in corrispondenza biunivoca con i sottosegmenti del righello nel seguente modo:



Il segmento $P_0\,P_2$ con la porzione $[u_0, u_2]$ del righello, il segmento $P_2\,P_4$ con la porzione $[u_1, u_3]$ del righello e così via.
Analiticamente ciò equivale a parametrizzare la retta P_0P_2 nel seguente modo

$$P(t) = P_0 + \frac{t - u_0}{u_2 - u_0}(P_2 - P_0)$$

Questa parametrizzazione di P_0P_2 fa ottenere P_0 per $t = u_0$ e P_2 per $t = u_2$. Su P_2P_4 si ottiene in modo analogo P_2 per $t = u_1$ e P_4 per $t = u_3$ e così via per gli altri segmenti $P_i\,P_{i+2}$.

Il segmento $P_0\,P_2$ viene ad avere un punto intermedio corrispondente a u_1, il segmento $P_2\,P_4$ un punto intermedio corrispondente a u_2 etc.
Questi punti, come si vede in figura, sono i punti $P_1, P_3, ...$ e ciò chiarisce il significato geometrico delle formule sopra che definiscono i P_i dispari.

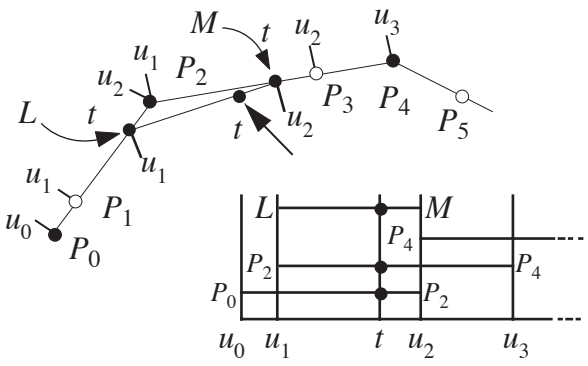
La B-spline varia cambiando la successione nodale. La variazione di un elemento della successione nodale ha effetto solo su tre curve di Bézier della curva (una o due se siamo agli estremi).

Osservazione: Se nella successione nodale si ha $u_0 = u_1$, allora $P_0 = P_1$ e la curva passa per il primo punto del poligono.
In generale però è bene che nella successione nodale non ci siano coincidenze fuori dagli estremi perché queste causano punti angolosi nella B-spline e comunque u_i troppo ravvicinati causano bruschi cambiamenti di curvatura.

L'algoritmo di De Casteljau nel caso quadratico non uniforme

La successione nodale funge anche da parametro per la parametrizzazione della curva risultante nel senso che ogni punto dell'intervallo $[u_1, u_n]$ (escludendo cioè gli estremi) fornisce un punto della B-spline in modo analogo all'algoritmo di De Casteljau per le Bézier quadratiche.

Per esempio scegliamo un t nell'intervallo $[u_1, u_2]$. Questo determina un punto della prima curva di Bézier della B-spline nel seguente modo:
Sul segmento P_0P_2 che è parametrizzato dall'intervallo $[u_0, u_2]$, si considera il punto $t \in [u_1, u_2]$ che chiamiamo L.
Analogamente sul segmento P_2P_4 che è parametrizzato dall'intervallo $[u_1, u_3]$, si considera il punto $t \in [u_1, u_2]$ che chiamiamo M.
Il segmento LM viene ora parametrizzato dall'intervallo $[u_1, u_2]$. In corrispondenza di t su LM si determina il punto della B-spline corrispondente a t.
Facendo variare t nell'intervallo $[u_1, u_2]$ si ottengono tutti i punti della B-spline compresa tra P_1 e P_3, facendolo variare in $[u_2, u_3]$ si ottengono i punti compresi tra P_3 e P_5 e così via.



3.3.7 Le B-spline cubiche

Le B-spline quadratiche danno risultati abbastanza soddisfacenti e, grazie alla flessibilità data dalle successioni nodali, si adattano facilmente a molte esigenze.

Le B-spline più usate comunque sono quelle cubiche. A tal proposito riportiamo due osservazioni:

Osservazione 1: Anche se le B-spline quadratiche sono curve di classe C^1 e quindi non presentano spigoli nei punti di giunzione delle varie curve di Bézier che la compongono, esse non sono quasi mai di classe C^2.
Le curve non di classe C^2, pur non avendo spigoli, risultano spesso sgradevoli per il fatto che *il raggio di curvatura*, che dipende dalla derivata seconda, può variare bruscamente nei punti di giunzione.
Le B-spline cubiche saranno invece costruite in modo da essere di classe C^2.

Osservazione 2: Le parabole sono curve piane. Se si crea una curva nello spazio raccordando tra loro rami di parabola, il passaggio da una parabola all'altra può risultare assai brusco perché cambia di colpo il piano di giacenza.
Le cubiche nello spazio invece sono dotate di *torsione* e quindi la giacitura sul piano osculatore varia con continuità nello spazio e nelle B-spline cubiche la variazione è continua anche nei punti di giunzione.

Nel seguito, i punti del poligono di de Boor saranno non necessariamente nel piano e quindi potranno avere una terza coordinata z, e ciò senza alcuna variazione degli algoritmi di costruzione delle B-spline.

Il poligono di de Boor

Come dati iniziali sono assegnati $n+1$ punti distinti (o meglio tali che tre consecutivi siano distinti) che costituiscono il *poligono di de Boor*.

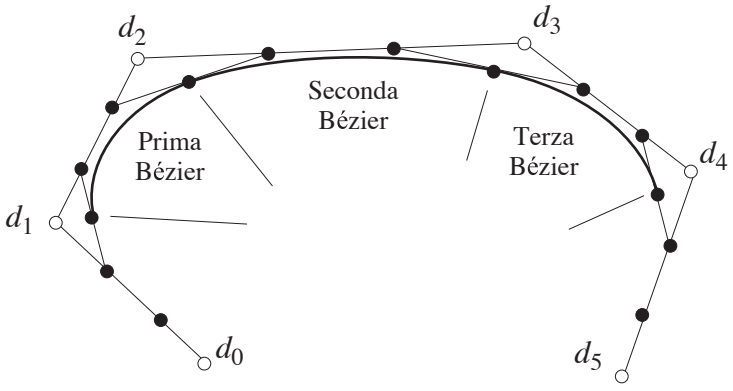
$$d_0, d_1, d_2, ..., d_{n-1}, d_n$$

Costruiremo una curva B-spline costituita da $n-2$ curve cubiche di Bézier ognuna delle quali ha un poligono di controllo A, B, C, D col secondo e terzo punto situati sui segmenti $d_i\, d_{i+1}$ $(i \neq 0, n)$.

I punti dei poligoni di controllo andranno scelti osservando certe regole se si vuole fare in modo che la B-spline risultante sia di classe C^2. Questo rende la costruzione leggermente più complessa che nel caso quadratico.

Caso uniforme

Nel caso più semplice divideremo i segmenti $d_i\, d_{i+1}$ in tre parti uguali. Poi considereremo i segmenti che hanno estremi due punti consecutivi delle divisioni e li divideremo in due parti uguali. La B-spline sarà costituita dalle curve di Bézier cubiche che hanno come punti di controllo questi. La figura dovrebbe chiarire quali sono i poligoni di controllo.



La curva così costruita è detta *B-spline cubica uniforme*. L'aggettivo uniforme si riferisce al fatto che i punti dei poligoni di controllo sono presi con suddivisioni uniformi dei segmenti del poligono di de Boor.

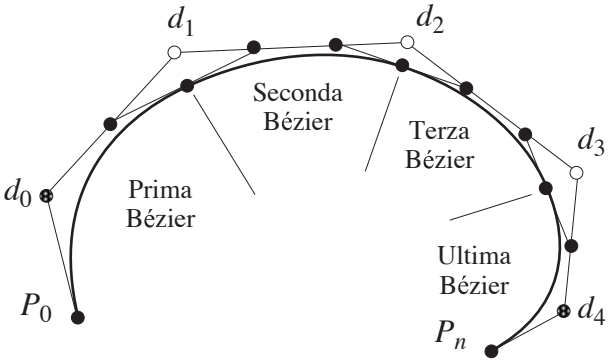
La B-spline è continua e di classe C^1 per costruzione. *Si può dimostrare che, se i punti P_i sono costruiti come sopra, e cioè dividendo in tre e in due parti uguali, essa è anche di classe C^2.*

Osserviamo che la curva non passa per nessuno dei punti del poligono di de Boor ed è anche lontana dagli estremi.

Esiste una costruzione alternativa che consente di far passare la curva per gli estremi del poligono, mantenendo la classe C^2.
Questa costruzione privilegia gli estremi del poligono, quindi è bene chiamare il poligono

$$P_0, d_0, d_1, ..., d_{n-1}, d_n, P_n$$

Il primo lato $d_0\, d_1$ viene diviso in due parti e non in tre e così pure l'ultimo. I punti estremi P_0 e P_n fanno parte del poligono di controllo delle Bézier estreme.



Per il resto tutto è come nel caso sopra. La figura dovrebbe chiarire la costruzione.
Comunque vedremo che l'uso delle successioni nodali nel caso di B-spline non uniformi fornirà un modo più efficace per costruire una B-spline con questa proprietà.

Caso non uniforme

Invece di dividere i segmenti $\overline{d_i\, d_{i+1}}$ in tre parti uguali e i segmenti intermedi in due parti uguali, si possono fare altre scelte e ottenere una modellazione diversa con un controllo *semi-locale*.
Si tenga però presente che scelte casuali della suddivisione dei vari segmenti possono far sì che la B-spline risultante non sia più di classe C^2 e, come abbiamo detto, le curve non di classe C^2 sono da evitare.
Per ottenere ciò si fa uso di una *successione nodale* che consente di effettuare modifiche, mantenendo la classe C^2.

Se il poligono di de Boor è $d_0, ..., d_n$, la successione nodale è una successione non decrescente di $n+3$ numeri positivi $u_0 \le u_1 \le \cdots \le u_{n+2}$. Definiremo quindi i punti di controllo delle Bézier cubiche usando gli u_i. La costruzione è analoga al caso quadratico, anche se più complessa.

Per motivi di chiarezza illustreremo la procedura mediante un semplice poligono di de Boor di soli quattro vertici. Si otterrà una B-spline cubica elementare costituita da una sola curva di Bézier.

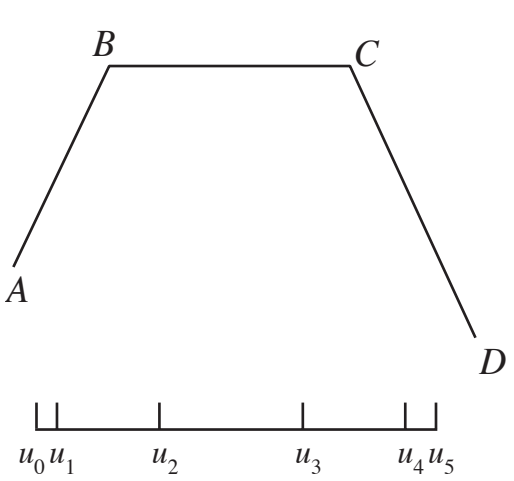
I quattro vertici per praticità saranno denotati A, B, C, D anziché d_0, d_1, d_2, d_3
La successione nodale sarà costituita da 6 numeri $u_0 \le u_1 \le \cdots \le u_5$.

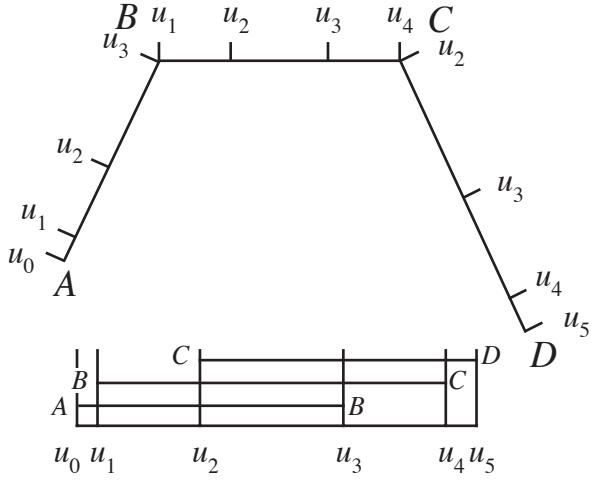
Visualizziamo, come prima, la successione

$$u_0, u_1, ..., u_5.$$

su un righello che useremo per parametrizzare i tre lati.

Parametrizziamo:
il segmento AB mediante $[u_0, u_3]$
il segmento BC mediante $[u_1, u_4]$
il segmento CD mediante $[u_2, u_5]$

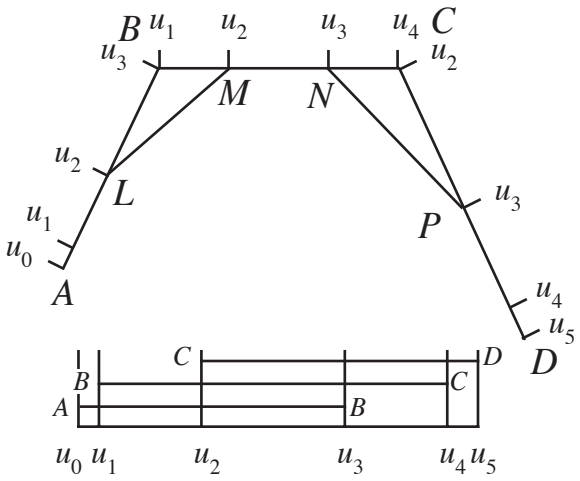


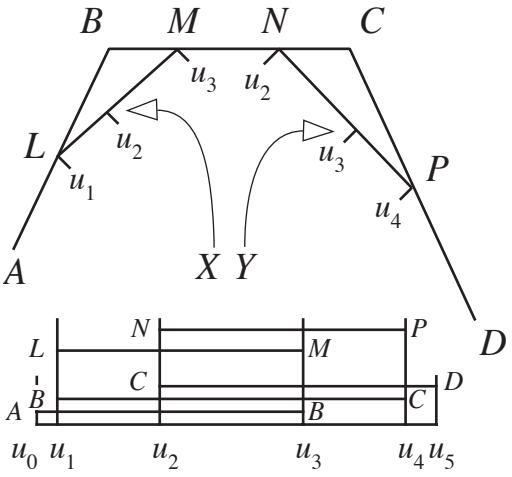


Su ognuno dei lati AB, BC, CD ci sono due punti ottenuti rispettivamente in corrispondenza di u_2 e di u_3.
Escludendo B ottenuto su AB per $t = u_3$ e C ottenuto su CD in corrispondenza di u_2, abbiamo 4 punti che chiamiamo L, M, N, P come nella prima figura sotto.

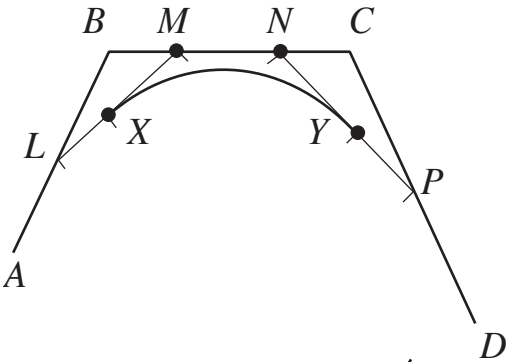
Analiticamente: $L = A + \dfrac{u_2 - u_0}{u_3 - u_0}(B - A)$ $\quad M = B + \dfrac{u_2 - u_1}{u_4 - u_1}(C - B)$ etc.

I due segmenti LM, NP vanno parametrizzati rispettivamente mediante l'intervallo $[u_1, u_3]$ e mediante l'intervallo $[u_2, u_4]$ come nella seconda figura sotto.

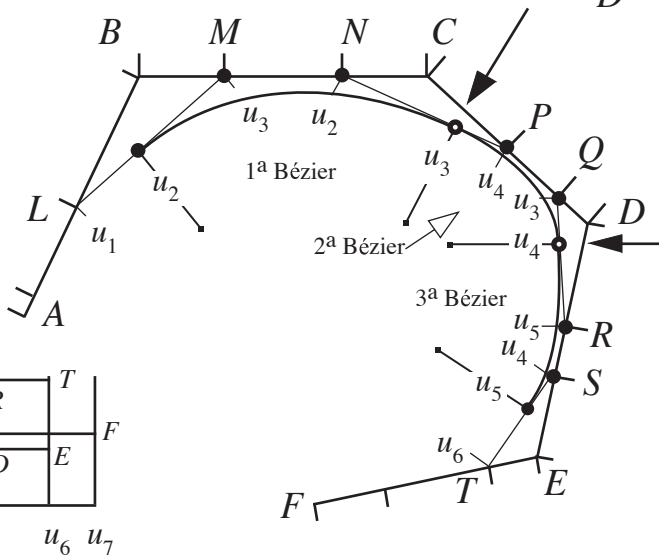


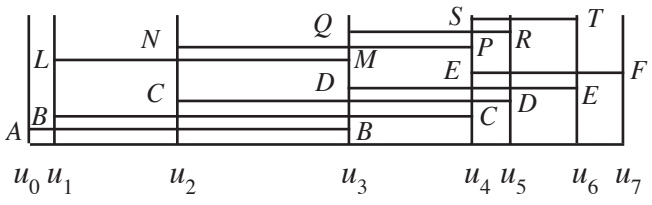


Fissiamo l'attenzione sul punto ottenuto su LM per $t = u_2$ che chiamiamo X e su quello ottenuto su NP per $t = u_3$ che chiamiamo Y (seconda figura sopra).
Come nel caso quadratico, costruiamo la curva di Bézier che ha come poligono di controllo quello costituito dai quattro punti X, M, N, Y segnati nella figura a lato. Osserviamo solo che, se la successione nodale è a passo costante, si riottiene il caso uniforme, con suddivisione dei tre segmenti AB, BC, CD in tre parti uguali e dei due segmenti LM, NP in due parti uguali.

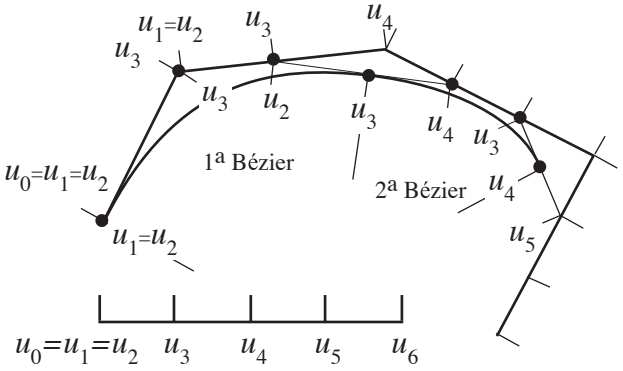


Nella figura accanto vediamo questo procedimento portato avanti per un poligono di 6 vertici $ABCDEF$ e quindi usando una successione nodale di 8 numeri $u_0, ..., u_7$. La B-spline conseguente è costituita da tre curve di Bèzier. Sono indicati i due punti di giunzione delle tre curve. Qui sotto il righello coi segmenti parametrizzati mediante la successione nodale.





Per concludere vediamo un esempio con poligono di de Boor di 5 vertici e una successione nodale $u_0, ..., u_6$ in cui $u_0 = u_1 = u_2$. Questo fa sì che la curva passi per il primo punto del poligono di de Boor. Per il resto abbiamo scelto che la successione nodale sia di passo costante. Fondamentalmente si ritrova la costruzione alternativa del caso uniforme che permette di far passare la B-spline per il primo vertice del poligono.

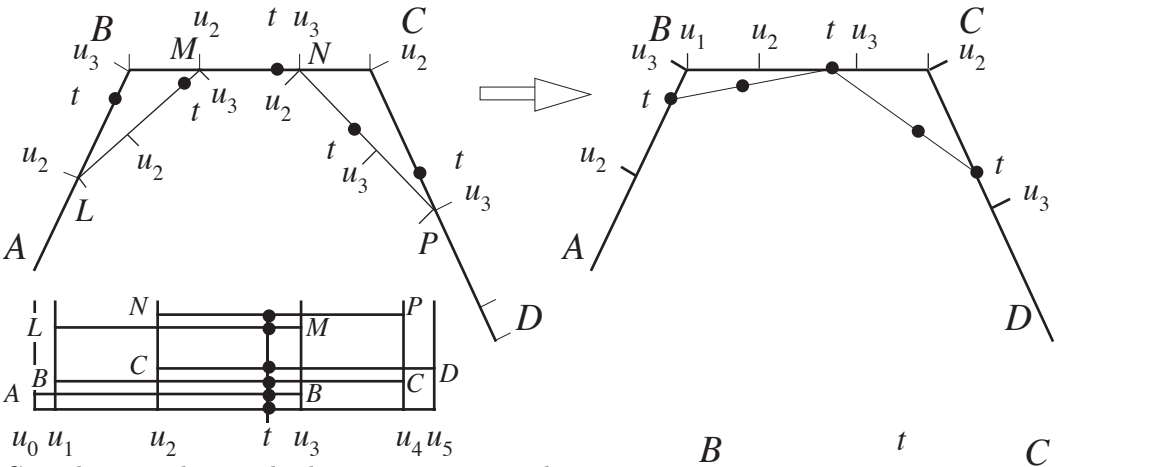


L'algoritmo di De Casteljau per le B-spline cubiche non uniformi

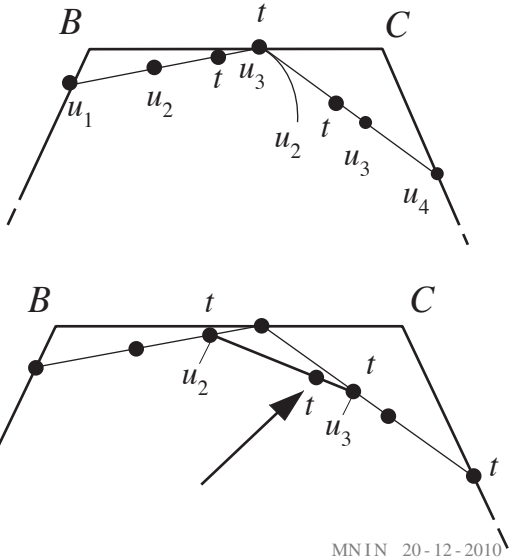
Invece di costruire le curve di Bézier dati i loro poligoni di controllo, può essere più conveniente parametrizzare la curva, come nel caso quadratico.

Riprendiamo quindi il caso del poligono $ABCD$ dal momento in cui sono stati trovati i punti L, M, N, P. Parametrizzeremo la curva con un parametro t che varia nell'intervallo $[u_2, u_3]$ della successione nodale.

Scegliamo $t \in [u_2, u_3]$ e riportiamolo su tutti i segmenti che sono stati posti in corrispondenza con questo intervallo. Ci sono 5 intervalli $[u_2, u_3]$, quindi fissiamo l'attenzione sui 5 punti corrispondenti a t. I tre punti su AB, su LM e su BC risultano allineati e così pure i tre su BC, su NP e su CD. Questo è conseguenza di un famoso teorema geometrico, noto come *Teorema di Menelao.*



Consideriamo dunque le due rette e poniamole in corrispondenza rispettivamente con gli intervalli $[u_1, u_3]$ e $[u_2, u_4]$ della successione nodale. Sempre mediante il teorema di Menelao, si può dimostrare che i punti intermedi per questa parametrizzazione degli intervalli sono proprio quelli che corrispondono rispettivamente a u_2 e a u_3. Su questi segmenti individuiamo ancora una volta il punto t compreso tra u_2 e u_3.



Per concludere consideriamo l'ultimo segmento che ha come estremi i due punti t e poniamolo in corrispondenza con l'intervallo $[u_2, u_3]$. Su questo segmento individuiamo il punto t compreso tra u_2 e u_3.

Questo è finalmente un punto della B-spline.

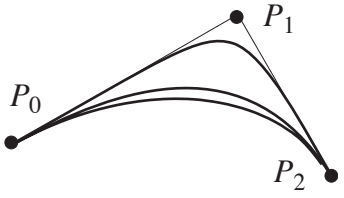
3.3.8 Cenno sulle curve di Bézier razionali

Le B-spline cubiche così definite degenerano correttamente in rette o in parabole se i punti del poligono sono disposti in maniera particolare, per esempio se sono allineati, ma non possono mai rappresentare correttamente un arco di circonferenza o di ellisse per il semplice motivo che né circonferenze, né ellissi ammettono una parametrizzazione mediante funzioni polinomiali, mentre le parametrizzazioni delle curve di Bézier sono polinomiali essendo combinazione lineare di polinomi di Bernstein. Questa è la ragione per cui spesso vengono utilizzate le curve B-spline razionali che sono a pezzi curve di Bézier razionali. Accenniamo brevemente a queste ultime.

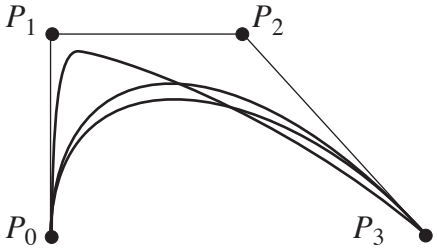
Per definire una curva di Bézier razionale di ordine n occorrono un poligono $P_0, P_1, ..., P_n$ e una successione di numeri positivi $w_1, ...w_n$ detti *pesi*. La curva ha rappresentazione parametrica

$$P(t) = \frac{w_0 P_0 B_0(t) + \cdots + w_n P_n B_n(x)}{w_0 B_0(t) + \cdots + w_n B_n(x)}$$

dove i B_i sono i polinomi di Bernstein di ordine n. Osserviamo che, per le proprietà dei polinomi di Bernstein, il denominatore vale 1 se tutti i pesi sono 1, per cui in questo caso si ottiene la solita curva di Bézier. Assegnando opportunamente i pesi si riesce a fare "aderire" più o meno la curva ai vertici del poligono ottenendo spesso risultati più flessibili di quelli delle curve di Bézier semplici e riuscendo per esempio a descrivere archi di coniche diversi dalle parabole nel caso di curve di Bézier quadratiche.
Senza entrare nei dettagli mostriamo due esempi.
Sul poligono $P_0P_1P_2$ sono state costruite le curve di Bézier quadratiche razionali con pesi [1 1 1] [1 1 2] [1 4 1] rispettivamente. La prima è la più distante da P_1 ed è la curva di Bézier normale (quindi una parabola), la seconda è la mediana ed è precisamente un quarto di ellisse inserito nel poligono di controllo, la terza "aderisce" al punto centrale del poligono di controllo ed è una porzione di iperbole. La curva con pesi [2 1 1] coincide con quella di pesi [1 1 2] anche se con diversa parametrizzazione.



Sul poligono $P_0P_1P_2P_3$ sono state costruite le curve di Bézier cubiche razionali con pesi [1 1 1 1] [3 2 2 3] [1 5 1 1] rispettivamente. La prima è la mediana ed è la cubica di Bézier normale. La seconda è la più bassa ed è un arco di ellisse, la terza "aderisce" al secondo punto del poligono di controllo.



5.1 Integrazione ed equazioni differenziali

È un capitolo assai vasto dell'analisi numerica, anche perché le equazioni differenziali sono uno strumento essenziale in moltissime questioni.
Ci limitiamo ai casi più semplici illustrando tecniche che comunque sono abbastanza antiche, anche se il loro studio ha ricevuto enorme impulso con l'avvento del calcolo automatico.
Il problema è che di raro è possibile risolvere le equazioni differenziali in modo esatto e quindi i metodi numerici sono quasi sempre indispensabili.

5.1.1 Richiami sugli integrali

Il più semplice problema differenziale è il seguente:

Data una funzione $f(t)$ definita in un punto t_0 e in un suo intorno (destro, sinistro o comprendente t_0), e un numero y_0, determinare una funzione $y(t)$ definita in un intervallo comprendente il numero t_0 tale che

$$y' = f(t) \qquad y(t_0) = y_0$$

La funzione $y(t)$ è detta *primitiva* di $f(t)$.
Come è ben noto, il teorema fondamentale del calcolo integrale fornisce la soluzione del problema:

Proposizione 12 *Se $f(t)$ è integrabile in un intorno di t_0, allora*

$$y(t) = y_0 + \int_{t_0}^{t} f(u)du$$

Il problema è quindi quello di calcolare l'integrale. Se la primitiva $y(t)$ è ricavabile mediante le note tecniche di integrazione indefinita, allora la proposizione fornisce semplicemente la nota formula

$$\int_{t_0}^{t} f(u)du = y(t) - y(t_0)$$

È noto che in molti casi $y(t)$ non è calcolabile elementarmente. In altri casi lo è, ma la sua espressione è comunque complessa, per cui ci proponiamo di ricavare degli algoritmi numerici per il calcolo dell'integrale definito.

5.1.2 Integrazione numerica: formule di Newton-Cotes

Vogliamo calcolare numericamente l'integrale definito

$$\int_{a}^{b} f(t)dt$$

dove $f(t)$ è una funzione continua nell'intervallo $[a, b]$ (ma basterebbe continua a tratti e limitata).

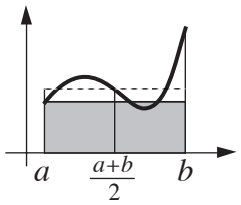
L'idea base è sempre quella di sostituire a $f(t)$ un polinomio di grado n passante per $n+1$ punti dell'intervallo $[a, b]$ e quindi di usare la primitiva del polinomio che è calcolabile elementarmente, avvertendo che, in genere, *non è necessario esplicitare il polinomio* per calcolare l'area sottesa.

A seconda del grado usato e del criterio di scelta dei punti si possono avere numerosissimi metodi di integrazione numerica.
Se tra gli $n+1$ punti ci sono gli estremi si parla di *metodo chiuso*.
Se gli $n+1$ punti sono scelti dividendo l'intervallo in parti uguali, le formule ricavate sono dette *formule di quadratura di Newton-Cotes*.

Ci limiteremo a quest'ultimo caso ed esamineremo in dettaglio i casi $n = 0, 1, 2$.

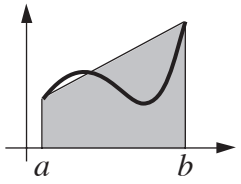
Metodo del rettangolo (o di Cauchy) ($n = 0$)

Il polinomio ha grado 0 è cioè una costante, quindi va scelto un solo punto dell'intervallo, per esempio il punto a.
Quindi $f(t)$ è sostituita dalla funzione costante $y = f(a)$ e notoriamente $\int_a^b f(a)dt = f(a)(b-a)$ (area del rettangolo).
Benché il metodo sia grossolano e banale, vedremo poi la sua controparte nel caso di un'equazione differenziale qualunque.
Osserviamo solo che, se invece di scegliere il punto a si sceglie il punto medio dell'intervallo $(a+b)/2$, si ottiene una formula assai simile alla successiva e in molti casi più precisa.



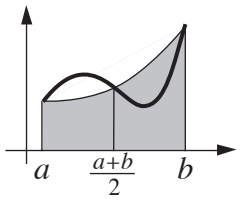
Metodo del trapezio (o di Bézout) ($n = 1$)

Il polinomio ha grado 1 e ha come grafico una retta, quindi vanno scelti due punti dell'intervallo che, nel caso chiuso di Newton-Cotes sono i due punti a, b.
Quindi $f(t)$ è sostituita dalla funzione che rappresenta la retta passante per i punti $(a, f(a))$ e $(b, f(b))$. L'integrale di questa funzione è l'area del trapezio in figura che vale $\dfrac{f(a)+f(b)}{2}(b-a)$



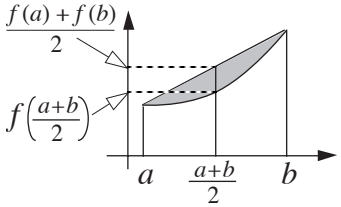
Metodo di Cavalieri-Simpson ($n = 2$)

Il polinomio ha grado 2 e ha come grafico una parabola, quindi vanno scelti tre punti dell'intervallo, che, nel caso chiuso di Newton-Cotes sono a, $\frac{a+b}{2}$, b.



Quindi $f(t)$ è sostituita dalla funzione che rappresenta la parabola $p(t)$ passante per tre punti. Un conto non difficile, anche se laborioso mostra che $\int_a^b p(t)dt = \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right)$ che è la classica formula di Cavalieri-Simpson.

Accenniamo a una semplice costruzione della formula di Cavalieri-Simpson: l'area sottesa dalla parabola è la differenza (o la somma se la parabola ha concavità verso il basso) dell'area del trapezio $(b-a)\frac{f(a)+f(b)}{2}$ e dell'area del settore parabolico che come è noto è $\frac{2}{3}(b-a)\left(\frac{f(a)+f(b)}{2} - f\left(\frac{a+b}{2}\right)\right)$.



Da qui con un semplice calcolo la formula.

È possibile ricavare formule analoghe per $n > 2$ e per altre scelte dei punti per i quali passa il polinomio, ma normalmente non ci si spinge oltre il grado due. Rileviamo solo che in diversi casi metodi aperti e con scelta non uniforme dei punti possono essere più convenienti dei metodi chiusi tipo Newton-Cotes.

5.1.3 Metodi generali di Cauchy, Bézout, Cavalieri-Simpson

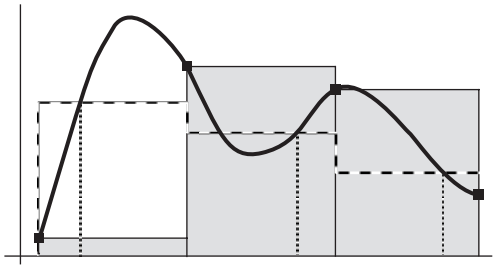
Non essendo convenienti le formule di Newton-Cotes per $n > 2$, la prassi usuale consiste nel suddividere l'intervallo $[a, b]$ in tanti sottointervalli in ciascuno dei quali viene applicato uno dei tre metodi esposti, tenendo presente che, se $a = x_0$, x_1 , ... , $x_n = b$ è una suddivisione dell'intervallo, si ha

$$\int_a^b f(t)dt = \int_{x_0}^{x_1} f(t)dt + \int_{x_1}^{x_2} f(t)dt + \cdots + \int_{x_{n-1}}^{x_n} f(t)dt$$

Supponiamo che la divisione dell'intervallo sia uniforme e poniamo $h = x_{i+1} - x_i$. Nei tre metodi citati (Cauchy, Bézout, Cavalieri-Simpson) le formule diventano:

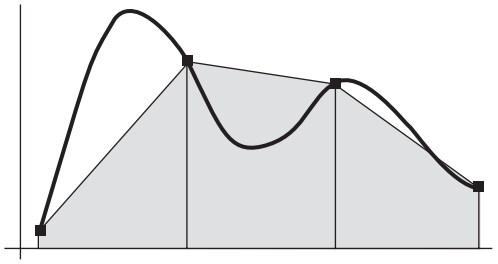
$\boxed{n = 0}$ Mediante il primo metodo, come integrale di $f(t)$ si ottiene proprio l'integrale definito mediante la definizione originale di Cauchy, cioè, come somma delle aree di plurirettangoli.

$$\int_a^b f(t)dt \simeq \sum_{i=0}^{n} f(x_i)\,(x_{i+1} - x_i) = h \cdot \sum_{i=0}^{n} f(x_i)$$



Nella definizione di integrale più spesso usata, quella di Riemann, il punto in cui si calcola f non è il primo punto di ogni intervallo, ma un qualunque punto ξ_i interno all'intervallo $[x_i\,,\, x_{i+1}]$.

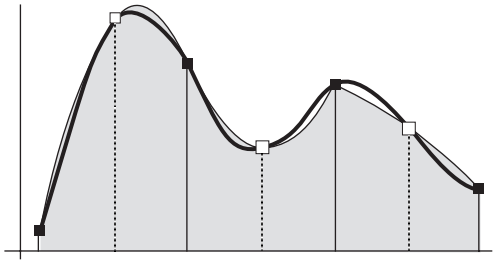
$\boxed{n = 1}$ Mediante il metodo dei trapezi, l'integrale approssimato di $f(t)$ si ottiene come somma delle aree di trapezi. In pratica si integra lo spline lineare di $f(t)$.



$$\int_a^b f(t)dt \simeq \sum_{i=0}^{n} \frac{f(x_i) + f(x_{i+1})}{2}\,(x_{i+1} - x_i) =$$

$$= \frac{h}{2} \cdot \Big(f(x_0) + 2\,f(x_1) + \cdots + 2f(x_{n-1}) + f(x_n)\Big)$$

$\boxed{n=2}$ Mediante il metodo di Cavalieri-Simpson l'integrale approssimato di $f(t)$ si ottiene come somma delle aree sottese da parabole.
La funzione formata da parabole è continua, ma non è lo spline quadratico della funzione nei punti x_i, sia perché vengono utilizzati anche i punti medi, sia perché non è detto sia dotata di derivata prima.



$$\int_a^b f(t)dt \simeq h \cdot \sum_{i=0}^{n} \frac{1}{6}\left(f(x_i) + 4f\left(\frac{x_i + x_{i+1}}{2}\right) + f(x_{i+1})\right) =$$

$$= \frac{h}{6}\left(f(x_0) + 4f\left(\frac{x_0 + x_1}{2}\right) + 2f(x_1) + 4f\left(\frac{x_1 + x_2}{2}\right) + \cdots + 2f(x_{n-1}) + f(x_n)\right)$$

5.1.4 L'errore nelle formule di integrazione numerica

Naturalmente è importante sapere quanto il valore calcolato con le formule numeriche si discosti dal vero valore dell'integrale. È possibile dare una valutazione dell'errore commesso solo nel caso in cui la funzione $f(t)$ presenti una certa regolarità. Senza addentrarci nei particolari, ci limitiamo a fornire le valutazioni per le formule di Bézout e Cavalieri-Simpson.

L'errore nelle formule di Bèzout

Nel caso semplice, cioè di un solo trapezio nell'intervallo $[a, b]$, si dimostra che:

Se $f(t)$ è dotata di derivata seconda *continua*, esiste un punto ξ dell'intervallo (a, b) tale che l'errore è in modulo uguale a $\left|\frac{(b-a)^3}{12} f''(\xi)\right|$

Nel caso generale di suddivisione dell'intervallo in sottointervalli di ampiezza h, esiste un punto ξ dell'intervallo (a, b) tale che l'errore è in modulo uguale a $\left|\frac{(b-a)\,h^2}{12} f''(\xi)\right|$

Quindi è possibile maggiorare l'errore, se è possibile maggiorare la derivata seconda di $f(t)$ nell'intervallo (a, b) e si può render piccolo quanto si vuole l'errore diminuendo l'ampiezza h dei sottointervalli. L'errore tende a zero al tendere a zero di h con ordine di infinitesimo pari a 2.

L'errore nelle formule di Cavalieri-Simpson

Nel caso semplice, cioè di una sola parabola nell'intervallo $[a, b]$, si dimostra che:

Se $f(t)$ è dotata di derivata quarta *continua*, esiste un punto ξ dell'intervallo (a, b) tale che l'errore è in modulo uguale a $\left|\frac{(b-a)^5}{2880} f^{(IV)}(\xi)\right|$

Nel caso generale di suddivisione dell'intervallo in sottointervalli di ampiezza h, esiste un punto ξ dell'intervallo (a, b) tale che l'errore è in modulo uguale a $\left|\frac{(b-a)\,h^4}{2280} f^{(IV)}(\xi)\right|$

È quindi possibile maggiorare l'errore se è possibile maggiorare la derivata quarta di $f(t)$ nell'intervallo (a, b) e si può render piccolo quanto si vuole l'errore diminuendo l'ampiezza h dei sottointervalli. L'errore tende a zero al tendere a zero di h con ordine di infinitesimo pari a 4.

Un'osservazione: alcuni testi danno formule di Cavalieri-Simpson e maggiorazioni differenti, ma solo perché considerano come ampiezza h, non quella degli intervalli $[x_i\,,\, x_{i+1}]$, ma la loro metà, visto che la funzione va calcolata anche nei punti medi.

5.2 Equazioni differenziali

5.2.1 Richiami sul problema di Cauchy

Il classico problema differenziale di Cauchy si può enunciare così:

Determinare una funzione $y(t)$ definita in un intervallo comprendente il numero t_0 tale che

$$y' = f(t, y) \qquad y(t_0) = y_0$$

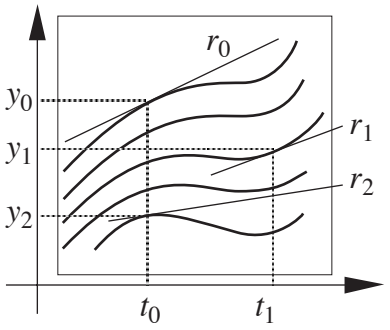
La differenza col problema dell'integrazione sta nel fatto che f è una funzione di due variabili e dipende anche dalla funzione y, anziché solo da t. Esistono varie condizioni sufficienti sulla funzione $f(t, y)$ che assicurano l'esistenza e unicità di una soluzione del problema. Una delle più semplici è la seguente (anche se in realtà spesso bastano condizioni assai meno restrittive).

Proposizione 13 *Se $f(t, y)$ è continua in un dominio rettangolare $D = \{[t_a, t_b] \times [y_a, y_b]\}$ con $t_0 \in (t_a, t_b)$ e $y_0 \in (y_a, y_b)$ e inoltre anche la funzione $\partial f/\partial y$ esiste ed è continua e quindi limitata in D, allora $y(t)$ esiste ed è unica in un intorno di t_0.*

Se f è una funzione elementare, in alcuni casi "speciali" esistono varie tecniche (variabili separabili etc.) per determinare esplicitamente $y(t)$. Ci occuperemo invece del caso in cui y non sia determinabile esplicitamente o comunque la sua espressione sia complessa.
In questi casi la soluzione va calcolata in modo approssimato mediante tecniche numeriche.

Osserviamo innanzitutto che il problema di Cauchy $y' = f(t, y)$ senza la condizione iniziale su $y(t_0)$, se ha soluzione, ne ha in generale infinite che costituiscono una famiglia di funzioni passanti per ognuna delle coppie (t_i, y_i) di punti interni al dominio.

L'osservazione è fondamentale perché la soluzione che troveremo sarà in qualche modo una mediazione tra molte di queste soluzioni.
Esiste una funzione $y(t)$ passante per (t_0, y_0) e tale che $f(t_0, y_0)$ sia il coefficiente angolare della retta r_0 e analogamente esiste una funzione $y(t)$ passante per (t_1, y_1) e tale che $f(t_1, y_1)$ sia il coefficiente angolare della retta r_1 e così pure per (t_0, y_2).
In generale non è detto che queste funzioni abbiano lo stesso campo di esistenza.



5.2.2 Il metodo di Eulero

Il metodo più semplice è quello di Eulero che è anche alla base di metodi più raffinati. Si fissa un passo h (che può anche essere negativo) e quindi si considerano diversi punti a partire da t_0:

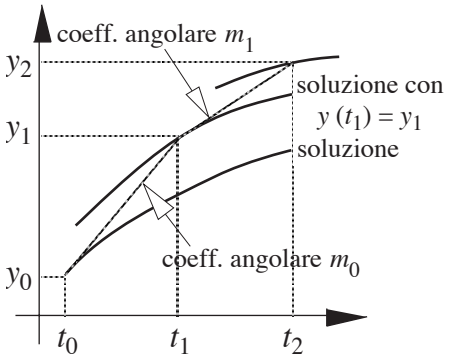
$t_0 \qquad t_1 = t_0 + h \qquad t_2 = t_1 + h \quad ...$

L'equazione differenziale ci fornisce il coefficiente angolare della soluzione in t_0 che è $y'(t_0) = f(t_0, y_0)$.

L'idea è quindi di sostituire la soluzione $y(t)$ con la retta passante per (t_0, y_0) di coefficiente angolare $f(t_0, y_0)$ che chiamiamo m_0 e che ha quindi equazione

$$y = y_0 + m_0(t - t_0)$$

Questo nell'intervallo $[t_0, t_1]$. Se è possibile proseguire oltre t_1, calcoliamo la funzione lineare in t_1: $y_1 = y_0 + m_0(t_1 - t_0)$, quindi nell'intervallo $[t_1, t_2]$ considereremo un'altra retta, quella passante per (t_1, y_1) con coefficiente angolare $m_1 = f(t_1, y_1)$ e cioè $y = y_1 + m_1 + (t - t_1)$



Si badi che comunque il punto (t_1, y_1) in generale non appartiene alla soluzione del problema originale, ma al grafico di un'altra funzione della famiglia dell'equazione differenziale. Man mano che l'algoritmo prosegue è possibile che ci si allontani sempre di più dalla soluzione del problema originale.
Alla fine si otterrà una funzione di cui si hanno i valori in $t_0, t_1, ...$ e si può eventualmente usare un metodo di interpolazione.

Osserviamo che se il problema di Cauchy è è semplicemente il problema di integrazione $\{y' = f(t) \quad ; \quad y(t_0) = y_0\}$, con f non dipendente da y, la soluzione fornita dal metodo di Eulero si riduce al metodo di Cauchy per gli integrali definiti con la suddivisione $t_0, t_1, ...$

5.2.3 Il metodo di Eulero quadratico

È detto in molti testi metodo dui Eulero modificato.
Il metodo di Eulero sopra descritto, consiste in pratica nel sostituire a $y(t)$ la sua linearizzazione, ovvero il suo sviluppo di Taylor arrestato al primo ordine in t_0 che è fornito direttamente dalla funzione $f(t, y)$.
Da qui nasce l'idea di sostituire a $y(t)$ il suo sviluppo di Taylor arrestato a un ordine superiore, per esempio due.
Esplicitamente, se $y'(t) = f(y, t)$, allora, usando note formule di derivazione delle funzioni composte, la sua derivata seconda è esprimibile in funzione delle derivate parziali di f ovvero si ha $y''(t) = \dfrac{df}{dt}(t, y_0) = f_t + f_y\, y' = f_t + f_y\, f$. Quindi la funzione quadratica che rappresenta il primo passo del metodo di Eulero quadratico è

$$y = y_0 + f(t_0, y_0)(t - t_0) + \frac{1}{2}\Big(f_t(t_0, y_0) + f_y(t_0, y_0)\, f(t_0, y_0)\Big)(t - t_0)^2$$

Di qui è possibile ricavare per $t = t_1$ il prossimo punto (t_1, y_1) da cui ricominciare l'algoritmo.
Il metodo non è di uso frequente, perché il calcolo delle derivate parziali può dare origine a formule assai complesse e vengono preferiti metodi che fanno uso di rette come quelli esposti di seguito.

5.2.4 I metodi di Eulero generalizzati

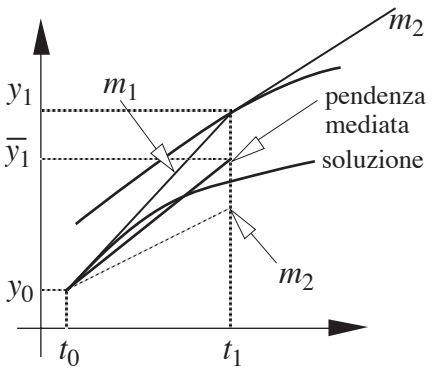
L'idea base dei metodi esposti qui di seguito è quella di sostituire la linearizzazione semplice di $y(t)$ con una funzione ugualmente lineare che tenga però già conto del comportamento della funzione nei punti successivi a t_0, cioè t_1 ed eventuali altri precedenti o successivi.

In generale partendo dalla formula elementare di Eulero
$y_1 = y_0 + f(t_0, y_0)(t_1 - t_0)$

si scelgono due numeri positivi c_1, c_2 tali che $c_1 + c_2 = 1$ e la formula viene così modificata

$$\overline{y_1} = y_0 + (t_1 - t_0)\Big(c_1\, f(t_0, y_0) + c_2\, f(t_1, y_1)\Big)$$

Quindi la funzione lineare ha una pendenza mediata tra quella nota in t_0 e quella calcolata in t_1 dopo il primo passo del metodo di Eulero. Vari accorgimenti suggeriscono i pesi c_1 e c_2 da usare.



5.2.5 Il metodo di Heun

Si tratta del metodo di Eulero generalizzato in cui $c_1 = c_2 = 1/2$. Quindi, come sopra

$$y_1 = y_0 + f(t_0, y_0)(t_1 - t_0) \qquad \overline{y_1} = y_0 + (t_1 - t_0)\left(\frac{f(t_0, y_0) + f(t_1, y_1)}{2}\right)$$

Quindi l'algoritmo procede con la coppia $(t_1, \overline{y_1})$. Si noti che la pendenza della retta è la media delle due pendenze calcolate in t_0 e in t_1, ma la nuova retta non è la bisettrice delle due.

Osserviamo ancora che se il problema di Cauchy è semplicemente il problema di integrazione $\{y' = f(t) \quad ; \quad y(t_0) = y_0\}$, con f non dipendente da y, la soluzione fornita dal metodo di Heun è il metodo di Bézout per gli integrali definiti con la suddivisione $t_0, t_1, ...$

5.2.6 Il metodo di Eulero modificato

Un'ulteriore generalizzazione della formula di Eulero può essere la seguente:

$$\overline{y_1} = y_0 + (t_1 - t_0)\left(c_1\, f(t_0, y_0) + c_2\, f\Big(t_0 + ha\ ,\ y_0 + b\, h\, f(t_0, y_0)\Big)\right)$$

dove $c_1 + c_2 = 1 \quad ; \quad ac_2 = 1/2 \quad ; \quad bc_2 = 1/2 \quad ; \quad h$ è il passo $t_1 - t_0$.
Quindi si ha una media pesata tra la pendenza calcolata in t_0 e quella calcolata in qualche punto compreso tra t_0 e t_1. I parametri c_1, c_2, a, b sono tutti da scegliere con vari criteri suggeriti dall'esperienza.

In molti testi è detto "metodo di Eulero modificato" quello che usa la formula precedente semplicemente con $c_1 = 0\ ;\ c_2 = 1$ e $a = b = 1/2$

$$\overline{y_1} = y_0 + (t_1 - t_0) f\left(t_0 + \frac{h}{2}\ ,\ y_0 + \frac{h}{2}\, f(t_0, y_0)\right)$$

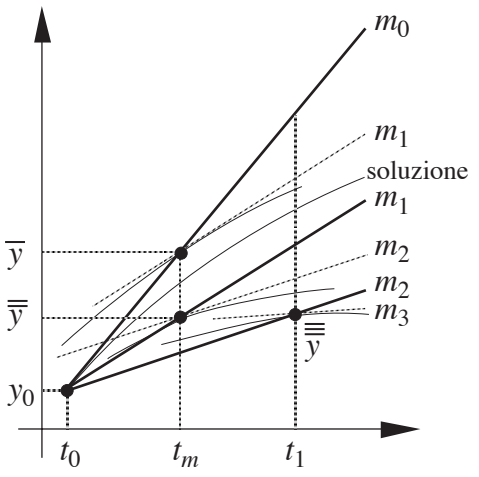
Quindi per determinare la nuova coppia $(t_1, \overline{y_1})$ si fa uso del valore della linearizzazione di $y(t)$ calcolato nel punto medio tra t_0 e t_1.

5.2.7 Il metodo di Runge-Kutta

Esistono diversi metodi detti di Runge-Kutta che fanno uso di varie medie delle pendenze in t_0, t_1 e in punti intermedi. Quello illustrato di seguito è il metodo classico di Runge-Kutta di ordine 4.

Si fa uso del punto medio tra i primi due punti della suddivisione $t_m = \dfrac{t_0 + t_1}{2}$.

Si inizia come nel metodo di Eulero con la retta passante per (t_0, y_0) di coefficiente angolare $m_0 = f(t_0, y_0)$. La retta è $y = y_0 + m_0(t - t_0)$.
Si trova il punto $\overline{y}$ in cui la retta ha ascissa t_m, ovvero $\overline{y} = y_0 + m_0(t_m - t_0)$.
Si calcola il valore di $f(t, y)$ nel punto $(t_m\,,\,\overline{y})$, quindi si pone $m_1 = f(t_m, \overline{y})$.



Si prosegue con la retta passante per (t_0, y_0) questa volta di coefficiente angolare m_1.
La retta è $y = y_0 + m_1(t - t_0)$.
Si trova il punto $\overline{\overline{y}}$ in cui la retta ha ascissa t_m ovvero $\overline{\overline{y}} = y_0 + m_1(t_m - t_0)$.
Si calcola il valore di $f(t, y)$ nel punto $(t_m\,,\,\overline{\overline{y}})$ quindi si pone $m_2 = f(t_m, \overline{\overline{y}})$.

Ancora una volta si considera la retta passante per (t_0, y_0), ma con coefficiente angolare m_2, cioè la retta $y = y_0 + m_2(t - t_0)$
Quest'ultima volta si trova il punto $\overline{\overline{\overline{y}}}$ in cui la retta ha ascissa t_1 (non t_m), ovvero $\overline{\overline{\overline{y}}} = y_0 + m_3(t_1 - t_0)$.
Si calcola il valore di $f(t, y)$ nel punto $(t_1\ ,\ \overline{\overline{\overline{y}}})$ quindi si pone $m_3 = f(t_1, \overline{\overline{\overline{y}}})$. Si osservi che i numeri m_i sono le pendenze di quattro diverse soluzioni dell'equazione differenziale $y' = f(t, y)$ che passano per quattro punti vicini a t_0.

Si definisce come primo passo del metodo di Runge-Kutta la retta di equazione

$$y = y_0 + \frac{m_0 + 2m_1 + 2m_2 + m_3}{6}(t - t_0)$$

e il primo valore della soluzione approssimata dell'equazione differenziale sarà

$$y_1 = y_0 + \frac{m_0 + 2m_1 + 2m_2 + m_3}{6}(t_1 - t_0)$$

Dopodiché si calcolerà y_2 in t_2 allo stesso modo, usando il punto intermedio tra t_1 e t_2.

Per terminare osserviamo ancora che se il problema di Cauchy è semplicemente il problema integrale $\{y' = f(t) \quad ; \quad y(t_0) = y_0\}$, la soluzione fornita dal metodo di Runge-Kutta è il metodo di Cavalieri-Simpson per gli integrali definiti con la suddivisione $t_0, t_1, ...$, di cui Runge-Kutta classico può essere quindi considerato una generalizzazione.

5.3 Equazioni differenziali: alcuni problemi al contorno

5.3.1 Schemi alle differenze finite per funzioni di una variabile

Sia $f(x)$ una funzione definita in un intervallo $[x_0, x_n]$, di cui siano note $n + 1$ coppie di valori (x_i, y_i) con $i = 0, .., n$, ovvero si sappia che

$$f(x_0) = y_0 \quad ; \quad f(x_1) = y_1 \quad ; \quad \cdots \quad ; \quad f(x_n) = y_n$$

Vogliamo valutare (in modo approssimato) le derivate della funzione $f(x)$, che si suppone continua e con derivate continue. Per semplicità consideriamo solo il caso in cui gli x_i siano equidistanti, ovvero per ogni $i = 0, ..., n - 1$ si abbia $x_{i+1} = x_i + h$.
Per valutare la derivata prima nel generico punto x_i, usiamo lo sviluppo di Taylor con punto iniziale x_i $(i = 1, ..., n - 1)$ che fornisce la funzione in $x_{i+1} = x_i + h$ e in $x_{i-1} = x_i - h$, ottenendo quindi

$$f(x_{i+1}) = f(x_i) + f'(x_i)\, h + f''(x_i)\, \frac{h^2}{2} + f'''(x_i)\frac{h^3}{6} + f^{iv}(x_i)\frac{h^4}{24} + \cdots$$

$$f(x_{i-1}) = f(x_i) - f'(x_i)\, h + f''(x_i)\, \frac{h^2}{2} - f'''(x_i)\frac{h^3}{6} + f^{iv}(x_i)\frac{h^4}{24} + \cdots$$

Sottraendo membro a membro otteniamo

$$f(x_{i+1}) - f(x_{i-1}) = f'(x_i)\, 2h + f'''(x_i)\frac{h^3}{3} + \cdots$$

da cui

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h} - f'''(x_i)\frac{h^3}{6} + \cdots \qquad \text{e} \qquad f'(x_i) \simeq \frac{y_{i+1} - y_{i-1}}{2h}$$

Se quindi h è abbastanza piccolo, l'espressione data fornisce il valore della derivata prima calcolata *alle differenze finite centrate.*

Sommando invece i due sviluppi di Taylor sopra

$$f(x_{i+1}) + f(x_{i-1}) = 2f(x_i) + f''(x_i)\, h^2 + f^{iv}(x_i)\frac{h^4}{12} + \cdots$$

da cui

$$f''(x_i) = \frac{f(x_{i+1}) - 2f(x_i) + f(x_{i-1})}{h^2} - f^{iv}(x_i)\frac{h^4}{12} + \cdots \qquad \text{e} \qquad f''(x_i) \simeq \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}$$

che, ancora per h piccolo, fornisce il valore della derivata seconda calcolata *alle differenze finite centrate.*

Un modo alternativo per generare le stesse formule è il seguente: si considera la funzione quadratica passante per i tre punti (x_{i-1}, y_{i-1}), (x_i, y_i), (x_{i+1}, y_{i+1}). La sua derivata prima in x_i è esattamente

$\frac{y_{i+1}-y_{i-1}}{2h}$ e la sua derivata seconda è $\frac{y_{i+1}-2y_i+y_{i-1}}{h^2}$

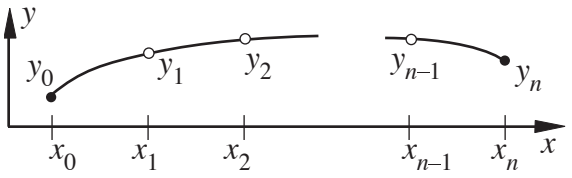
Le formule date consentono di stimare le derivate per tutti *i punti interni*, ovvero nei punti x_i con $i = 1, ..., n-1$, e si prestano a risolvere problemi con condizioni al contorno.
Gli schemi alle differenze discussi in precedenza valutano le derivate prime e seconde di una funzione di classe C^2 con un errore di ordine inferiore a h^3 e per questo sono esatte al secondo ordine. Attraverso procedure analoghe è possibile ottenere formule alle differenze di ordine superiore per la valutazione delle quali vengono coinvolti non solo x_{i-1} e x_{i+1}, ma altri punti precedenti e successivi.

5.3.2 Condizioni al contorno: problema di Dirichlet in una dimensione

Il più semplice problema di Dirichlet del secondo ordine in una dimensione è quello di determinare una funzione $y(t)$ definita in un intervallo $[x_0, x_n]$ tale che

$$\begin{cases} y'' = f(x, y, y') & \text{Sotto opportune ipotesi di regolarità di } f, \text{ il problema ha una soluzione.} \\ y(x_0) = y_0 & \text{Senza addentrarci nei particolari teorici del problema, ci proponiamo di} \\ y(x_n) = y_n & \text{approssimarne la soluzione facendo uso degli schemi alle differenze finite.} \end{cases}$$

Gli estremi dell'intervallo sono stati chiamati x_0, x_n perché, al fine di approssimare la soluzione, divideremo l'intervallo in n sottointervalli di ampiezza $h = \frac{x_n - x_0}{n}$ mediante i punti $x_0, x_1, ..., x_n$ con $x_{i+1} = x_i + h$ per $i = 0, ..., n-1$



Le incognite saranno $y_1, ..., y_{n-1}$, dato che il problema al contorno fornisce già y_0 e y_n.
Dalla discretizzazione alle differenze finite dell'equazione differenziale in corrispondenza del generico punto x_i si ottiene per $i = 1, ..., n-1$

$$\frac{y_{i+1}-2y_i+y_{i-1}}{h^2} = f\left(x_i, y_i, \frac{y_{i+1}-y_{i-1}}{2h}\right)$$

Queste sono $n-1$ equazioni in $n-1$ incognite. La risoluzione di questo sistema è elementare quando la funzione $f(x, y, y')$ è di primo grado rispetto a y e a y' perché in tal caso ognuna delle equazioni è lineare nelle y_i e quindi il problema consiste nel risolvere un sistema lineare.
Esaminiamo le equazioni:

Nella prima (per $i = 1$) compaiono solo y_0 (che è dato), y_1 e y_2.
Nella seconda (per $i = 2$) compaiono solo y_1, y_2 e y_3.
.....
Nella $n-2$-esima (per $i = n-2$) compaiono solo y_{n-3}, y_{n-2} e y_{n-1}.
Nella $n-1$-esima (per $i = n-1$) compaiono solo y_{n-2}, y_{n-1} e y_n (che è dato).

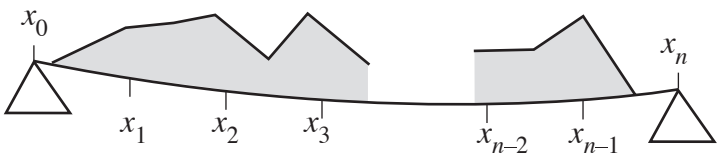
Quindi la matrice dei coefficienti del sistema è del tipo

$$\begin{pmatrix} a_{11} & a_{12} & 0 & \cdots & 0 & 0 \\ a_{21} & a_{22} & a_{23} & \cdots & 0 & 0 \\ 0 & a_{32} & a_{33} & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & & & \\ 0 & 0 & 0 & \cdots & a_{n-1,n-2} & a_{n-1,n-1} \end{pmatrix}$$

Ovvero una matrice tridiagonale e spesso diagonalmente dominante.

5.3.3 Condizioni al contorno in una dimensione: il problema della trave

Come semplice esempio consideriamo il caso di una trave appoggiata sulla quale insiste una distribuzione di carico come mostrato in figura



Senza entrare nel dettaglio, il problema è descritto da una equazione differenziale del tipo:

$$y''(x) + c(x)y(x) = p(x)$$

in cui y rappresenta lo spostamento verticale della trave causato dalla presenza del carico, $c(x)$ descrive caratteristiche locali della trave (materiale, sezione, forma) e $p(x)$ è il valore locale della pressione del carico. Nell'esempio che segue supporremo nota la distribuzione del carico per ogni i e porremo $p_i = p(x_i)$ e $c_i = c(x_i)$ (c è costante se la trave, come spesso accade, è omogenea).
Dalla discretizzazione alle differenze finite dell'equazione differenziale in corrispondenza di ogni i, si ottiene:

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + c_i y_i = p_i$$

Scriviamo le singole equazioni cominciando da quella per $i = 1$ e concludendo con $i = n - 1$:

$$\begin{aligned}
\frac{y_2 - 2y_1 + y_0}{h^2} + c_1 y_1 &= p_1 & (c_1h^2 - 2)y_1 + y_2 &= p_1h^2 - y_0 \\
\frac{y_3 - 2y_2 + y_1}{h^2} + c_2 y_2 &= p_2 & y_1 + (c_2h^2 - 2)y_2 + y_3 &= p_2h^2 \\
&\cdots & &\cdots \\
\frac{y_{n-2} - 2y_{n-1} + y_n}{h^2} + c_{n-1} y_{n-1} &= p_{n-1} & y_{n-2} + (c_{n-1}h^2 - 2)y_{n-1} &= p_{n-1}h^2 - y_n
\end{aligned}$$

La matrice completa del sistema lineare è pertanto

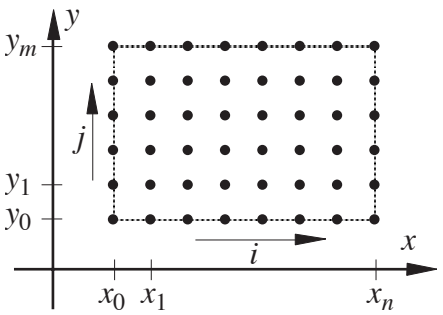
$$\left(\begin{array}{cccccccc|c}
c_1h^2-2 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & p_1h^2 - y_0 \\
1 & c_2h^2-2 & 1 & 0 & \cdots & 0 & 0 & 0 & p_2h^2 \\
0 & 1 & c_3h^2-2 & 1 & \cdots & 0 & 0 & 0 & p_3h^2 \\
\cdots & \cdots & & & \cdots & & & & \cdots \\
0 & 0 & 0 & 0 & \cdots & 1 & c_{n-2}h^2-2 & 1 & p_{n-2}h^2 \\
0 & 0 & 0 & 0 & \cdots & 0 & 1 & c_{n-1}h^2-2 & p_{n-1}h^2 - y_n
\end{array}\right)$$

La matrice è tridiagonale e può essere ridotta facilmente con l'algoritmo di Gauss (sono anche applicabili i metodi iterativi in quando diagonalmente dominante, anche se in questo caso non sempre sono convenienti)

5.3.4 Schemi alle differenze finite per funzioni di due variabili

Nel caso in cui la funzione dipenda da due o più variabili, le formule della sezione precedente possono essere estese alle derivate parziali.

Per semplicità consideriamo un dominio rettangolare $[x_0, x_n] \times [y_0, y_m]$ in cui sia definita una funzione $g(x, y)$. Per discretizzare il problema suddividiamo i lati del rettangolo in sottointervalli di passo costante: rispettivamente $\Delta x = x_{i+1} - x_i \quad \forall i$ e $\Delta y = y_{j+1} - y_j \quad \forall j$.
Quindi si hanno $(n+1)\times(m+1)$ punti ognuno dei quali è individuato da una coppia di indici (i, j) con $i = 0, ..., n$ e $j = 0, ..., m$.
Scriveremo $g(i, j)$ in luogo di $g(x_i, y_j)$.



Un tipico esempio è il problema di calcolare la funzione di più variabili che determina la distribuzione di temperatura nell'ambiente nel caso in cui la variabile in esame, la temperatura, sia nota al contorno cioè nei punti segnati in tratteggio.

È semplice la costruzione di schemi alle differenze finite per il calcolo di derivate parziali rispetto

a x o y. In pratica si tratta di utilizzare le stesse formule riportate in precedenza per le funzioni di una sola variabile, per la funzione di due variabili $g(i,j)$, facendo variare solo l'indice relativo alla variabile rispetto alla quale si deve effettuare la derivata mantenendo fisso l'altro. Il passo verrà poi sostituito da Δx o Δy, a seconda della direzione rispetto alla quale si effettua la derivata.
Riportiamo di seguito le espressioni delle derivate parziali prime e seconde rispetto a x e y, valutate *nei punti interni* al dominio mediante gli schemi alle differenze centrate riportati nel paragrafo precedente. La quantità

$$\frac{\partial g}{\partial x}(i,j) = \frac{g(i+1,j) - g(i-1,j)}{2\Delta x}$$

può essere assunta come valore della derivata parziale rispetto alla x nel punto (i,j). Analogamente

$$\frac{\partial g}{\partial y}(i,j) = \frac{g(i,j+1) - g(i,j-1)}{2\Delta y}$$

può essere assunta come valore della derivata parziale rispetto alla y nello stesso punto (i,j).
Per le derivate seconde si ha:

$$\frac{\partial^2 g}{\partial x^2}(i,j) = \frac{g(i+1,j) - 2g(i,j) + g(i-1,j)}{\Delta x^2}$$

$$\frac{\partial^2 g}{\partial y^2}(i,j) = \frac{g(i,j+1) - 2g(i,j) + g(i,j-1)}{\Delta y^2}$$

Gli errori relativi, trattandosi di schemi al secondo ordine, saranno rispettivamente inferiori a Δx^2 o Δy^2 per le derivate parziali rispetto a x o a y.

5.3.5 Equazioni di Laplace e Poisson e loro soluzione numerica

Molti problemi di fisica tecnica, fluidodinamica, strutture e teoria dei campi elettromagnetici sono descritti da una equazione differenziale del tipo

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = b(x,y)$$

nota come *equazione di Poisson*.
La funzione $f(x,y)$ rappresenta la distribuzione di una qualche variabile fisica, mentre la $b(x,y)$ rappresenta un termine sorgente.
Un tipico problema è quello del calcolo della temperatura $f(x,y)$ su una superficie consoscendo l'intensità $b(x,y)$ di una fonte di calore.
Un altro problema è quello di derminare la forma di una membrana che si incurva fino ad assumere una posizione d'equilibrio quando è sottoposta a una forza verticale di densità $b(x,y)$. La funzione forma f è una soluzione dell'equazione di Poisson.

In molti problemi il termine sorgente è nullo in tutto il dominio e l'equazione assume la forma

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} = 0$$

nota come *equazione di Laplace*.
Sottolineiamo che le forme riportate non si limitano a problemi bidimensionali. Forme completamente analoghe si possono avere in tre dimensioni aggiungendo la derivata seconda rispetto alla terza variabile z.
Le equazioni di Poisson o Laplace hanno soluzione unica in un dominio chiuso e limitato con opportune ipotesi su $b(x,y)$ e opportune condizioni al contorno. Le condizioni al contorno possono essere di due tipi: in alcuni casi viene assegnata la stessa funzione f nel contorno del dominio, mentre in altri casi viene assegnata la sua derivata nella direzione normale alla curva che delimita il dominio. Nel primo caso si parla di condizioni di Dirichlet nel secondo caso si parla di condizioni di Neumann.

Si hanno condizioni di Dirichlet quando ad esempio si conosce il valore della temperatura su tutto il contorno. Si hanno condizioni di Neumann quando sul contorno è posizionata una sorgente di calore o, nel caso della membrana, quando il bordo della membrana è fissato a una curva data Γ e sul bordo agisce una forza di densità lineare sempre in direzione verticale.

Come semplice esempio presentiamo la soluzione numerica dell'equazione di Poisson in un dominio rettangolare con condizioni di Dirichlet al contorno.
Ci limitiamo al caso in cui il problema sia quello di determinare la f in un dominio piano rettangolare, come quello descritto nel paragrafo precedente di cui conserviamo le notazioni.
I lati vengono discretizzati rispettivamente con n e m sottointervalli e supporremo, per semplicità anche che $\Delta x = \Delta y = h$.
Dato che la funzione f è nota nel contorno, sono noti i valori $f(0,j)$, $f(n,j)$ per ogni j e i valori $f(i,0)$, $f(i,m)$ per ogni i.
Le incognite sono $f(i,j)$ per $i = 1, ..., n-1$ e $j = 1, ..., m-1$. Per determinarle è necessario scrivere l'equazione di Poisson in forma discreta in ognuno di questi punti (i,j).
Le formule del paragrafo precedente sono

$$\frac{\partial^2 f}{\partial x^2}(i,j) = \frac{f(i+1,j) - 2f(i,j) + f(i-1,j)}{h^2}$$

$$\frac{\partial^2 f}{\partial y^2}(i,j) = \frac{f(i,j+1) - 2f(i,j) + f(i,j-1)}{h^2}$$

Sostituendo queste due espressioni delle derivate seconde nell'equazione di Poisson:

$$f(i,j-1) + f(i-1,j) - 4f(i,j) + f(i+1,j) + f(i,j+1) = h^2 \cdot b(i,j)$$

Scriviamo questa espressione per tutte le coppie di indici i, j con $i = 1, ..., n-1$, $j = 1, ..., m-1$ e teniamo conto del fatto che i valori nei punti del contorno sono noti.
Per esempio per $i, j = 1$

$$f(1,0) + f(0,1) - 4f(1,1) + f(2,1) + f(1,2) = h^2 b(1,1)$$

e poiché $f(1,0)$ e $f(0,1)$ sono noti, si ha

$$-4f(1,1) + f(2,1) + f(1,2) = -f(1,0) - f(0,1) + h^2 b(1,1)$$

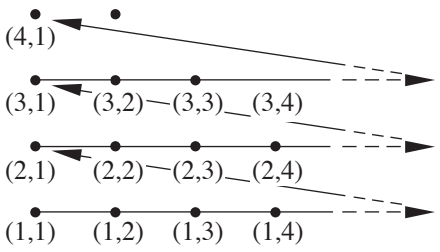
In modo simile, per il punto $(2,1)$ si ha

$$f(1,1) - 4f(2,1) + f(3,1) + f(2,2) = -f(2,0) + h^2 b(2,1)$$

e così via. Nei punti con $i = 2, ..., n-2$, $j = 2, ..., m-2$, nessuno dei punti che compaiono nella somma appartiene al contorno e quindi nell'equazione compaiono 5 incognite. Per esempio, nel punto $(3,2)$ la forma è

$$f(3,1) + f(2,2) - 4f(3,2) + f(4,2) + f(3,3) = h^2 b(3,2)$$

In conclusione abbiamo $(n-1)\cdot(m-1)$ equazioni lineari in $(n-1)\cdot(m-1)$ incognite.
Occorre ordinare in qualche modo le incognite $f(i,j)$.
È conveniente usare l'ordinamento raffigurato a lato, per cui le incognite sono, nell'ordine



$$f(1,1), f(1,2), f(1,3), \ldots, f(2,1), f(2,2), \ldots, f(n-1, m-1)$$

In questo modo le equazioni formano un sistema lineare quadrato $Ax = b$ la cui matrice dei coefficienti è di formato $(n-1)\cdot(m-1)$. La matrice completa è

$$\left(\begin{array}{cccccccccccc|l}
-4 & 1 & 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & h^2b(1,1) - f(1,0) - f(0,1) \\
1 & -4 & 1 & 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & 0 & 0 & \cdots & h^2b(2,1) - f(2,0) \\
0 & 1 & -4 & 1 & \cdots & 0 & 0 & 1 & \cdots & 0 & 0 & 0 & \cdots & h^2b(3,1) - f(3,0) \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & & & \\
1 & 0 & 0 & 0 & \cdots & -4 & 1 & 0 & \cdots & 1 & 0 & 0 & \cdots & h^2b(1,2) - f(0,2) \\
0 & 1 & 0 & 0 & \cdots & 1 & -4 & 1 & \cdots & 0 & 1 & 0 & \cdots & h^2b(2,2) \\
0 & 0 & 1 & 0 & \cdots & 0 & 1 & -4 & \cdots & 0 & 0 & 1 & \cdots & h^2b(3,2) \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & & & \\
0 & 0 & 0 & 0 & \cdots & 1 & 0 & 0 & \cdots & -4 & 1 & 0 & \cdots & h^2b(1,3) - f(0,3) \\
0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 & \cdots & 1 & -4 & 1 & \cdots & h^2b(2,3) \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & \cdots & 0 & 1 & -4 & \cdots & h^2b(3,3) \\
 & & & & \ddots & & & & & & & & & \\
 & & & & & \ddots & & & & & & & & \\
 & & & & & & \ddots & & & & & & & \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 1 & 0 & 0 & \cdots & -4 & 1 & 0 & h^2b(\,) - f(m-3,n) \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & 1 & 0 & \cdots & 1 & -4 & 1 & h^2b(\,) - f(m-2,n) \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & 0 & 1 & \cdots & 0 & 1 & -4 & h^2b(\,) - f(m-1,n) - \\
 & & & & & & & & & & & & & \quad -f(m,n-1)
\end{array}\right)$$

La matrice dei coefficienti ha una struttura tridiagonale a blocchi di questo tipo

$$A = \left(\begin{array}{cccccc}
\boxed{B} & \boxed{I} & 0 & 0 & 0 & \cdots \\
\boxed{I} & \boxed{B} & \boxed{I} & 0 & 0 & \cdots \\
0 & \boxed{I} & \boxed{B} & \boxed{I} & 0 & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{array}\right)$$

dove le $n-1$ matrici B sono a loro volta tridiagonali di ordine $(m-1)\times(m-1)$ e ognuna delle matrici I è la matrice identica di ordine $(m-1)$.
Inoltre la matrice è diagonalmente dominante (seppur debolmente) e questo consente l'impiego di metodi iterativi, indispensabili per la risolvere sistemi con un numero molto elevato di incognite.