

Insiemi

La notazione matematica	2
L'assioma della scelta	3
L'insieme delle parti	4

Le funzioni

Le funzioni	2
Uguaglianza di funzioni	3
Composizione di funzioni	3
Associatività	3
La funzione identica	3
L'immagine	3
Funzioni iniettive	3
Funzioni suriettive	4
Iniettività categoriale	4
Suriettività categoriale	4
Funzioni biiettive	4
Spazi di funzioni	4
Proprietà funtoriali	4

Algebra lineare

Equazioni lineari in una incognita	6
Sistemi astratti	7
Due sistemi lineari in due incognite	7
Esempi	8
La forma generale della regola di Cramer	8
Determinanti	8
L'algoritmo di eliminazione di Gauß	9
Sistemi con più di una soluzione	10
L'insieme delle soluzioni di un sistema lineare	10

Trigonometria

Trigonometria oggi	11
Un problema di geodesia	11
Il triangolo	12
Il triangolo rettangolo	12
Triple pitagoree	12
Le funzioni trigonometriche	12
La dimostrazione indiana	13
Il triangolo isoscele	13
Angoli sul cerchio	13
Il teorema del coseno	14
Il grafico della funzione seno	14
La periodicità di seno e coseno	14
Altre proprietà di seno e coseno	14
arcsin, arccos e arctan	14

Geometria

Grafica al calcolatore e geometria	11
Distanze in \mathbb{R}^n	15
Il prodotto scalare	15
Ortogonalità	15
Coordinate polari nel piano	21
Coordinate cilindriche nello spazio	21
Coordinate polari nello spazio	21
Rotazioni nel piano	21
Disuguaglianze fondamentali	24
Il segno del prodotto scalare	24
Ellissi	29
Iperboli	29
Rette e segmenti	30
Equazione di una retta	30
Proiezione su una retta	30
Riflessione in un punto	36
Riflessione in una retta	36

Numeri complessi

Numeri complessi in \mathbb{R}	22
I numeri complessi	23
La formula di Euler	23
Il campo dei numeri complessi	24
La formula di de Moivre	24
Parte reale e parte immaginaria	24
Radici di un polinomio	25
Radici n -esime dell'unità	25
Radici di un numero complesso	25
Perturbazione dei coefficienti	36
I cammini delle radici	37
Il teorema di Rouché	37
Continuità delle radici	37

Programmazione in R

R	16
Le funzioni d'aiuto	16
Installazione di R	16
Il libro di Crawley	16
Programmare in R	17
Programmi autonomi	17
Nomi in R	17
Assegnamento	17
Successioni	18
Angoli espressi in gradi	18
I commenti	18
Funzioni in R	22
expression ed eval	29
ifelse	32
NA	32
if ... else	33
identical	33
Operazioni insiemistiche	33
dimnames	33
for, while e repeat	40
Abbreviazioni	45
Options e print(x,n)	45
any e all	46
Ordinamento	46
La libreria	46

Grafica con R

Figure di Lissajous	18
La grafica di R	19
plot e lines	19
Il comando postscript	19
points	22
symbols e rect	22
Grafici di funzioni	26
Curve parametrizzate nel piano	26
Octobrina elegans	26
Octobrinidae I	27
Octobrinidae II	27
abline	28
Parabrinidae	28
Rotazioni	28
Testi matematici	29
Curve di livello	31
$x^2 - y^2 = k$	31
Una collana	32
$y^2 = x^3 + x^2$	33
$y^2 = x^3$	33
$y^2 = x^3 - x^2$	33
$y^2 = x^3 + x$	34
$y^2 = x^3 - x$	34
Il foglio di Cartesio	34
La chiocciola di Pascal	34
La lemniscata	34
Una curva trascendente	34
La funzione polygon	35
Il file figure	35
Tratteggi	35
Una modifica in Postscript	36
Parallele	38
Come nasce una forma	38

Vettori e matrici con R

outer	32
Creazione di matrici	38
Matrici in R	39
Esempi per le operazioni matriciali	39
Piccoli operatori	39
Indici vettoriali	40
$\sqrt[n]{v \% p0}$	40
Assegnazione vettoriale	41
Indici matriciali	41
L'opzione drop	41
rbind e cbind	41
Sistemi lineari con R	42
length e dim	42
det (determinante) e traccia	42
Matrici diagonali	42
abs (valore assoluto) e sign	42
Autovalori	42
Matrici simmetriche	43
eigen (autovalori e autovettori)	43
Matrici reali	43
I cerchi di Gershgorin	43

Analisi

I numeri binomiali	20
La formula di Stirling	21
Funzioni iperboliche	28
Derivazione simbolica	44
La derivata	44
La funzione esponenziale	44
Esempi per l'uso di D	45
La serie di Taylor	45

La professione del matematico

Che cos'è la matematica?	1
La professione	2
Dall'università all'azienda	2
Geometria applicata	5
La matematica in azienda	5
La statistica matematica	5
La matematica degli ingegneri	5
Matematica e chimica	5
La dinamica dei fluidi	5
Geomatematica	5
Malattie tropicali	5
La matematica del futuro	18

Varia

Il re dei matematici	6
Le basi di Gröbner	6
Il sito CRAN di Ferrara	30
Il crivello di Eratostene	40

Strumenti

L'alfabeto greco	1
------------------	---

Esercizi per gli scritti

Esercizi 1-13	10
Esercizi 14-17	15
Esercizi 18-22	18
Esercizi 23-27	22
Esercizi 28-43	25
Esercizi 44-53	30
Esercizi 54-58	34
Esercizi 59-66	38
Esercizi 67-76	43

ALGORITMI E STRUTTURE DI DATI

Corso di laurea in matematica

Anno accademico 2004/05

Numero 1

Che cos'è la matematica?

Dividiamo questa domanda in due sottodomande, cercando di indicare prima i costituenti elementari della matematica, poi come la matematica deve essere usata.

I componenti elementari del ragionamento matematico sono enunciati della forma **ipotesi implica tesi**; in questo senso la matematica non conosce affermazioni assolute, ma soltanto proposizioni che si compongono ogni volta di un preciso elenco delle ipotesi che vengono fatte, e poi di una altrettanto precisa specificazione dell'enunciato che secondo quella proposizione ne consegue. A questo punto non è detto che la proposizione sia valida, bisogna ancora dimostrarla, e ciò significa, nella matematica, dimostrare che la tesi segue dalle ipotesi unite agli assiomi e ai risultati già ottenuti e alle regole logiche che dobbiamo applicare. Gli assiomi sono enunciati che vengono messi all'inizio di una teoria, senza dimostrazione; ogni altro enunciato deve essere invece dimostrato.

È importante che bisogna sempre dimostrare una proposizione - che è sempre della forma **ipotesi implica tesi**! - nella sua interezza, cioè che si tratta di dimostrare la validità dell'implicazione e non la validità della tesi. L'enunciato **A implica B** può essere vero, anche se **B** non è vero. Ad esempio in logica si impara che, se l'ipotesi **A** è falsa, allora la proposizione **A implica B** è sempre vera. Quindi l'affermazione **se 3 è uguale a 3.1, allora io mi chiamo Piero** è sempre vera, indipendentemente da come mi chiamo io. Nella pratica matematica ciò significa che da una premessa errata si può, con un po' di pazienza, dedurre qualunque cosa.

La validità si riferisce quindi sempre a tutta la proposizione **A implica B**.

Mentre il matematico puro cerca soprattutto di arricchire l'edificio delle teorie matematiche con nuovi concetti o con dimostrazioni, talvolta assai difficili, di teoremi, il matematico applicato deve anche saper usare la matematica. Nelle scienze naturali e sociali, le quali pongono problemi molto complessi, uno dei compiti più importanti è spesso la separazione degli elementi essenziali di un fenomeno dagli aspetti marginali. In queste scienze le informazioni disponibili sono quasi sempre incomplete, cosicché possiamo ogni volta descrivere soltanto una piccola parte della realtà. Anche quando disponiamo di conoscenze dettagliate, queste si presentano in grande quantità, sono complesse e multiformi e richiedono concetti ordinatori per poterle interpretare. Ciò significa che bisogna estrarre e semplificare.

Un modello matematico di un fenomeno ha soprattutto lo scopo di permettere di comprendere meglio quel fenomeno, quindi di metterne in evidenza cause e effetti e comportamenti quantitativi, di individuarne i tratti essenziali e i meccanismi fondamentali. In un certo senso la matematica consiste di tautologie, e nel modello matematico si tenta di evidenziare le tautologie contenute nel fenomeno studiato. La teoria cerca di comprendere i processi e legami funzionali di un campo del sapere.

La mente umana pensa in modelli. Anche quando non facciamo matematica della natura, cerchiamo di comprendere la natura mediante immagini semplificate. La teoria inizia già nell'istante in cui cominciamo a porci la domanda quali siano gli aspetti essenziali di un oggetto o di un fenomeno. La matematica non è dunque altro che un modo sistematico e controllato di eseguire questi processi di astrazione e semplificazione costantemente attivi nel nostro intelletto.

Il modello matematico, una volta concepito, se sviluppato correttamente, si mostra poi di una esattezza naturale che spesso impressiona l'utente finale che è tentato di adottare gli enunciati matematici come se essi corrispondessero precisamente ai fenomeni modellati. Ma ciò non è affatto vero: La precisione del modello matematico è soltanto una precisione interna, tautologica, e la semplificazione, quindi verità approssimata e parziale, che sta all'inizio del modello, si conserva, e più avanza lo sviluppo matematico, maggiore è il pericolo che iterando più volte l'errore, questo sia cresciuto in misura tale da richiedere un'interpretazione estremamente prudente dei risultati matematici. Proprio le teorie più avanzate, più belle quindi per il matematico puro, sono spesso quelle più lontane dalla realtà. Questo automatismo della matematica può essere però anche fonte di nuovi punti di vista e svelare connessioni nascoste.

Un modello matematico è perciò solo un ausilio per la riflessione, per controllare il contenuto e la consistenza logica di un pensiero o di una ricerca. In altre parole, modelli sono strumenti intellettuali e non si possono da essi aspettare descrizioni perfette della realtà. Essi non forniscono risposte complete, ma indicano piuttosto quali siano le domande che bisogna porre.

L'astrattezza intrinseca della matematica comporta da un lato che essa rimanga sempre diversa dalla realtà, offre però dall'altro lato la possibilità di generalizzare i risultati ottenuti nelle ricerche in un particolare campo applicativo o anche uno strumento della matematica pura a problemi apparentemente completamente diversi, se questi hanno proprietà formali in comune con il primo campo.

In questo numero

- 1 Che cos'è la matematica?
L'alfabeto greco
- 2 La professione
Dall'università all'azienda
La notazione matematica
Le funzioni
- 3 Uguaglianza di funzioni
Composizione di funzioni
Associatività
La funzione identica
L'immagine
Funzioni iniettive
L'assioma della scelta
- 4 Funzioni suriettive
Iniettività categoriale
Suriettività categoriale
Funzioni biiettive
Spazi di funzioni
Proprietà funtoriali
L'insieme delle parti
- 5 Geometria applicata
La matematica in azienda
La statistica matematica
La matematica degli ingegneri
Matematica e chimica
La dinamica dei fluidi
Geomatematica
Malattie tropicali

L'alfabeto greco

alfa	α	A
beta	β	B
gamma	γ	Γ
delta	δ	Δ
epsilon	ϵ	E
zeta	ζ	Z
eta	η	H
theta	θ	Θ
iota	ι	I
kappa	κ	K
lambda	λ	Λ
mi	μ	M
ni	ν	N
xi	ξ	Ξ
omikron	\omicron	O
pi	π	Π
rho	ρ	P
sigma	σ, ς	Σ
tau	τ	T
upsilon	υ	Υ
fi	φ	Φ
chi	χ	X
psi	ψ	Ψ
omega	ω	Ω

Sono 24 lettere. La sigma minuscola ha due forme: alla fine della parola si scrive ς , altrimenti σ . In matematica si usa solo la σ .

Mikros ($\mu\kappa\rho\varsigma$) significa piccolo, megas ($\mu\epsilon\gamma\alpha\varsigma$) grande, quindi la omikron è la \omicron piccola e la omega la ω grande.

Le lettere greche vengono usate molto spesso nella matematica, ad esempio $\sum_{k=0}^3 a_k$ è un'abbreviazione per la somma $a_0 + a_1 + a_2 + a_3$ e $\prod_{k=0}^3 a_k$ per il prodotto $a_0 a_1 a_2 a_3$, mentre A_α^i è un oggetto con due indici, per uno dei quali abbiamo usato una lettera greca.

La professione

- * In genere lavoro in un team con ingegneri, biologi o bancari. Lavoro autonomo o in team nella programmazione (chi conosce il C e il Perl può realizzare le costruzioni e tecniche astratte matematiche imparate nel corso di laurea in modo creativo e utile per l'azienda).
- * *Il matematico deve essere in grado di comprendere e analizzare i problemi che si pongono nella prassi aziendale, di tradurli nel linguaggio formale della matematica e di risolverli, spesso con l'aiuto di tecniche e mezzi informatici.*
- * Matematica finanziaria e attuariale (calcolo delle probabilità, statistica, serie temporali, previsione dei valori di borsa, calcolo dei rischi e delle tariffe).
- * Matematica aziendale (ottimizzazione, statistica, teoria dei grafi). Il matematico può lavorare nei reparti di ricerca operativa in grandi ditte, nello sviluppo di software aziendale, nel controllo della produzione, nel servizio statistico, nelle banche, nelle assicurazioni, nella pubblica amministrazione.
- * Matematica industriale (equazioni differenziali, fisica matematica, ma anche statistica, serie temporali, ottimizzazione e controllo automatico, analisi numerica).
- * Elaborazione delle immagini (applicazioni nell'industria, nella medicina, nella robotica). Geometria (discreta e differenziale) computazionale, grafica al calcolatore).
- * Informatica teorica: algebra, strutture ordinate, funzioni booleane (metodi classici di semplificazione, diagrammi di decisione, proprietà matematiche-strutturali), linguaggi formali, automi. Algoritmi. Analisi formale dei concetti.
- * Bioinformatica (confronto di sequenze, studio dell'espressione genica, reti metaboliche). Statistica multivariata di dati clinici di grandi dimensioni.
- * Possibilità di posizioni anche superiori (industria farmaceutica, ramo attuariale-statistico).

Dall'università all'azienda

- ☉ Per inserirsi e crescere aziendalimente nel modo migliore, è importante capire sin dall'inizio che cosa le imprese vogliono dai laureati appena assunti.
- ☉ Allenarsi al lavoro in team vuol dire l'opposto che vedersi con i propri simili: vuol dire sviluppare l'interdisciplinarietà, la capacità di farsi capire da chi ha una cultura e un gergo differenti, di trovare soluzioni a problemi che toccano tutti in modo diverso.
- ☉ L'università spinge invece ad aggregarsi per omogeneità, a verificare di sapere tutti esattamente le stesse cose. Purtroppo l'università abitua spesso a un rapporto passivo, spersonalizzato, burocratico e disincentiva l'iniziativa individuale, la capacità di costruirsi sentieri e schemi propri.

Gian Battista Rosa (ed.): Dall'università all'azienda. ACTL 2002, 350p. Euro 30.

La notazione matematica

Matematica e linguaggi di programmazione hanno in comune la quasi perfetta precisione e allo stesso tempo la grande complessità degli enunciati. È necessario quindi definire attentamente gli oggetti con cui si vuole lavorare e le operazioni che si vogliono effettuare. Spesso questi oggetti e queste operazioni vengono utilizzati più volte nello stesso ragionamento o in una teoria. È quindi necessario introdurre abbreviazioni e, siccome talvolta le stesse operazioni possono essere effettuate su oggetti diversi, variabili. Ci limitiamo qui ad alcuni dei più comuni concetti insiemistici.

L'insieme che consiste degli elementi a_1, \dots, a_n è denotato con $\{a_1, \dots, a_n\}$. $\{x \mid P(x)\}$ è l'insieme di tutti gli x che godono della proprietà P o che comunque possono essere descritti dall'espressione $P(x)$. Se x è elemento di un insieme X , allora si scrive $x \in X$. Il simbolo $:=$ significa *uguale per definizione*. Alcuni insiemi di numeri:

$$\mathbb{N} := \{0, 1, 2, 3, \dots\}$$

$$\mathbb{Z} := \{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$$

$$\mathbb{Q} := \left\{ \frac{a}{b} \mid a, b \in \mathbb{Z} \text{ e } b \neq 0 \right\}$$

Gli elementi di \mathbb{N} si chiamano *numeri naturali*, quelli di \mathbb{Z} *numeri interi*, gli elementi di \mathbb{Q} *numeri razionali*. L'insieme dei *numeri reali* viene definito nei corsi di analisi ed è denotato con \mathbb{R} .

Molto importante e versatile è il concetto di **prodotto cartesiano** di insiemi: dati due insiemi X ed Y , con $X \times Y$ si denota l'insieme delle coppie (ordinate) di elementi che si possono formare prendendo come prima componente un elemento di X e come seconda un elemento di Y :

$$X \times Y := \{(x, y) \mid x \in X, y \in Y\}$$

Si possono anche formare prodotti cartesiani di più di due fattori, in particolare si può formare l'insieme $X^n := \underbrace{X \times \dots \times X}_{n \text{ volte}}$.

\mathbb{R}^2 è quindi il piano reale, \mathbb{R}^3 lo spazio tridimensionale. In statistica una tabella di n righe ed m colonne di numeri può essere considerata come una collezione di n punti nello spazio \mathbb{R}^m .

Quando un insieme X è contenuto in un insieme Y (ciò significa, per definizione, che ogni elemento di X è anche elemento di Y), allora si scrive $X \subset Y$. Così abbiamo $\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R}$. Due insiemi si chiamano *uguali* se hanno gli stessi elementi. Ciò si può formulare anche così:

$$X = Y \text{ se e solo se } X \subset Y \text{ e } Y \subset X.$$

L'**unione** $A \cup B$ di due insiemi A e B è l'insieme di tutti gli elementi che appartengono ad almeno uno dei due insiemi, mentre l'**intersezione** $A \cap B$ è l'insieme dei loro elementi comuni, cioè degli elementi che appartengono sia ad A che a B .

Come in aritmetica è utile avere un numero 0, così nell'insiemistica si introduce un insieme *vuoto* apparentemente artificiale definito dalla proprietà di non avere alcun elemento. Esso viene denotato con \emptyset ed è sottoinsieme di ogni altro insieme: $\emptyset \subset X$ per ogni insieme X (infatti ogni elemento di \emptyset , cioè nessuno, appartiene a X).

\iff è un'abbreviazione per *se e solo se*.

Le funzioni

Il concetto singolo più importante della matematica è certamente quello di funzione. Mediante funzioni possiamo trasformare informazioni da una forma all'altra, possiamo semplificare informazioni complesse o immergere informazioni in contesti più generali. Una funzione del tempo può essere studiata per scoprire proprietà di periodicità, funzioni complicate possono essere decomposte in funzioni più semplici. Se per esempio sappiamo che il prodotto di due funzioni *continue* è ancora continuo e se accettiamo per certo che la funzione che manda un numero reale in se stesso è continua, possiamo immediatamente concludere che la funzione che manda un numero reale x in x^2 è anch'essa continua.

Il concetto di funzione in matematica è molto generale. Una **funzione** (o **applicazione**) è definita da tre componenti: un insieme X su cui la funzione è definita (il *dominio* della funzione), un insieme Y (il *codominio*) di valori possibili (ogni valore della funzione deve essere un elemento di Y , ma non necessariamente ogni elemento di Y è valore della funzione), e un sottoinsieme $G \subset X \times Y$ (il *grafico* della funzione) che deve avere la proprietà che per ogni $x \in X$ esiste esattamente un $y \in Y$ tale che $(x, y) \in G$.

La tripla $f := (X, Y, G)$ si chiama allora una funzione da X in Y e si scrive anche $f: X \rightarrow Y$ oppure $X \xrightarrow{f} Y$. Per ogni $x \in X$ con $f(x)$ si denota quell'unico $y \in Y$ per cui $(x, y) \in G$.

Se f può essere espressa mediante una formula, per f si scrive anche $f = \bigcirc_x f(x)$,

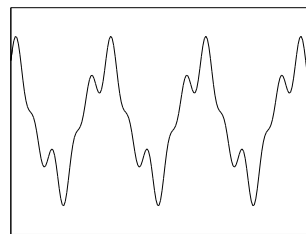
ad esempio $\bigcirc_x \sin(x^2 + 1): \mathbb{R} \rightarrow \mathbb{R}$.

Quando X è un insieme finito e non troppo grande, una funzione $f: X \rightarrow Y$ può essere rappresentata anche da una tabella:

x	$f(x)$
1	4
2	0
3	3
4	2
5	1
6	2

In questo caso $X = \{1, 2, 3, 4, 5, 6\}$ e per Y possiamo prendere ad esempio l'insieme $Y = \{0, 1, 2, 3, 4\}$.

Nell'analisi di una variabile reale si studiano funzioni definite su un intervallo di \mathbb{R} a valori reali. Il grafo di queste funzioni è un sottoinsieme di \mathbb{R}^2 e può quindi essere rappresentato nel piano.



$$y = \sin 3x + \sin 5x + 4 \cos x$$

Funzioni di questa forma, rappresentabili come somme (finite o infinite) di funzioni trigonometriche, vengono studiate nell'*analisi armonica* (o *analisi di Fourier*).

Uguaglianza di funzioni

Quand'è che due funzioni sono uguali?

Per definizione una funzione è una tripla. Due triple di oggetti matematici (a_1, b_1, c_1) e (a_2, b_2, c_2) sono uguali, se coincidono in ogni componente, cioè se $a_1 = b_1, a_2 = b_2$ e $a_3 = b_3$. Perciò due funzioni $f_1 = (X_1, Y_1, G_1)$ e $f_2 = (X_2, Y_2, G_2)$ sono uguali se e solo se $X_1 = X_2, Y_1 = Y_2$ e $G_1 = G_2$. Le prime due condizioni significano che le due funzioni hanno lo stesso dominio e lo stesso codominio che a questo punto possiamo chiamare X e Y ; analizziamo la terza condizione, cioè $G_1 = G_2$. Ciò significa che, per $x \in X$ e $y \in Y$ si ha $(x, y) \in G_1$ se e solo se $(x, y) \in G_2$ e quindi $y = f_1(x)$ se e solo se $y = f_2(x)$. In altre parole, la terza condizione è equivalente a $f_1(x) = f_2(x)$ per ogni $x \in X$.

Due funzioni f_1 e f_2 sono perciò uguali se e solo se hanno lo stesso dominio X e lo stesso codominio Y e se inoltre $f_1(x) = f_2(x)$ per ogni $x \in X$.

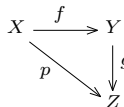
Composizione di funzioni

Se $f : X \rightarrow Y$ e $g : Y \rightarrow Z$ sono due funzioni, si può definire la funzione composta $g \circ f : X \rightarrow Z$ ponendo

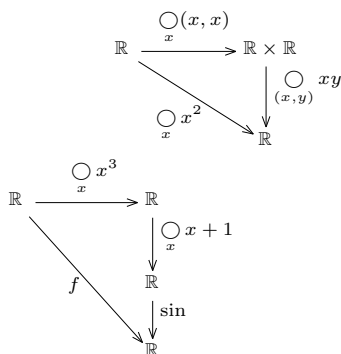
$$g \circ f(x) := g(f(x))$$

per ogni $x \in X$. La funzione composta trasforma quindi prima x in $f(x)$ e applica poi la g ad $f(x)$. Questa operazione è molto importante, sia per costruire nuove funzioni da funzioni note, sia per studiare funzioni complicate scomponendole in funzioni più semplici, rappresentando cioè la funzione complicata come composizione di funzioni più semplici.

Si usano allora spesso **diagrammi commutativi**: Un diagramma di funzioni



si chiama commutativo, se $p = g \circ f$. Siccome p e $g \circ f$ hanno, per definizione, lo stesso dominio X e lo stesso codominio Z , l'uguaglianza significa che $p(x) = g(f(x))$ per ogni $x \in X$. Esempi:



In analisi si dimostra che la composizione di funzioni continue è continua; quindi se sappiamo che $\bigcirc x^3$, $\bigcirc x + 1$ e \sin sono continue, possiamo dedurre che anche la funzione $f = \bigcirc \sin(x^3 + 1)$ è continua.

Associatività

Assumiamo che siano date tre funzioni $X \xrightarrow{f} Y, Y \xrightarrow{g} Z$ e $Z \xrightarrow{h} U$. Allora possiamo formare prima $g \circ f$ e comporre il risultato con h ottenendo così $r_1 := h \circ (g \circ f)$, oppure comporre f con $h \circ g$, ottenendo la funzione $r_2 := (h \circ g) \circ f$. Dimostriamo che si ottiene in entrambi i casi la stessa funzione. In primo luogo r_1 ed r_2 hanno lo stesso dominio X e lo stesso codominio U . Dobbiamo dimostrare che $r_1(x) = r_2(x)$ per ogni $x \in X$. Ma

$$\begin{aligned} r_1(x) &= h \circ (g \circ f)(x) = \\ &= h((g \circ f)(x)) = h(g(f(x))) \end{aligned}$$

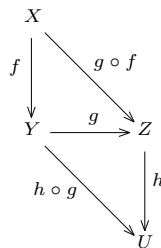
$$\begin{aligned} r_2(x) &= (h \circ g) \circ f(x) = \\ &= (h \circ g)(f(x)) = h(g(f(x))) \end{aligned}$$

e quindi $r_1(x) = r_2(x)$. Abbiamo così ottenuto la **legge di associatività** per la composizione di funzioni:

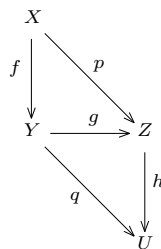
$$h \circ (g \circ f) = (h \circ g) \circ f$$

Possiamo perciò tralasciare le parentesi e scrivere semplicemente $h \circ g \circ f$ per una composizione di tre funzioni. In un diagramma commutativo la legge di associatività ci permette di percorrere il diagramma seguendo le frecce, il risultato dipenderà solo dall'inizio e dalla fine del cammino. Verificare questa regola nell'ultimo diagramma in basso a sinistra!

La legge di associatività corrisponde alla commutatività del diagramma

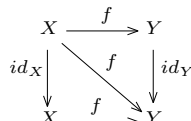


In verità, in un diagramma commutativo non useremo il cerchietto per denotare la composizione, ma daremo dei nuovi nomi a tutte le frecce; che nel diagramma che segue $p = g \circ f$ segue proprio dalla commutatività.



La funzione identica

Per ogni insieme X esiste la funzione identica (o identità) $id_X := \bigcirc x : X \rightarrow X$. Per ogni funzione $f : X \rightarrow Y$ abbiamo un diagramma commutativo



L'immagine

$f : X \rightarrow Y$ sia un'applicazione. Allora l'**immagine** di f è l'insieme

$$\begin{aligned} \text{Imm } f &:= \{f(x) \mid x \in X\} \\ &= \{y \in Y \mid \text{esiste } x \in X \text{ con } y = f(x)\} \end{aligned}$$

$\text{Imm } f$ è quindi un sottoinsieme di Y e consiste di quegli $y \in Y$ che sono immagine sotto f di qualche elemento di X .

Funzioni iniettive

Una funzione $f : X \rightarrow Y$ è detta **iniettiva**, se trasforma elementi distinti di X in elementi distinti di Y ; quindi la funzione f è iniettiva se e solo se l'uguaglianza $f(x_1) = f(x_2)$ implica $x_1 = x_2$.

Proposizione 3.1. Sia $X \neq \emptyset$. Allora la funzione $f : X \rightarrow Y$ è iniettiva se e solo se esiste una funzione $g : Y \rightarrow X$ tale che $g \circ f = id_X$.

Dimostrazione: (1) f sia iniettiva. L'idea della costruzione di g è di mandare ogni $y \in \text{Imm } f$ in quell' $x \in X$ (unico per la iniettività di f) per cui $y = f(x)$. Se $\text{Imm } f = Y$, abbiamo già finito. Cosa facciamo però, in caso generale, con quegli $y \in Y$ che non appartengono ad $\text{Imm } f$? A questo scopo scegliamo un elemento qualsiasi $x_0 \in X$ (ciò è possibile perché per ipotesi $X \neq \emptyset$) e mandiamo tutti gli elementi di $\{y \in Y \mid y \notin \text{Imm } f\}$ in x_0 . In altre parole

$$g(y) = \begin{cases} x & \text{se } y = f(x) \\ x_0 & \text{altrimenti} \end{cases}$$

In verità avremmo dovuto definire g mediante un grafo. È però chiaro che basta porre $g = (Y, X, H)$ con $H = H_1 \cup H_2$, dove

$$H_1 := \{(f(x), x) \mid x \in X\}$$

e

$$H_2 := \{(y, x_0) \mid y \in Y, y \notin \text{Imm } f\}.$$

(2) Molto più semplice, ma in un certo senso anche più interessante è la seconda direzione della dimostrazione, perché dà un primo saggio dell'efficacia di questo modo astratto di ragionare. Assumiamo che esista una funzione $g : Y \rightarrow X$ tale che $g \circ f = id_X$. Per dimostrare che f allora è iniettiva, prendiamo due elementi x_1 e x_2 di X per cui $f(x_1) = f(x_2)$.

Ma, per ipotesi, $g(f(x)) = x$ per ogni $x \in X$, e quindi $x_1 = g(f(x_1)) = g(f(x_2)) = x_2$.

L'assioma della scelta

Quando si sviluppa assiomaticamente la teoria degli insiemi, si scopre che un enunciato che per molti versi sembrerebbe naturale, non può essere dedotto dagli altri assiomi. È quindi necessario aggiungerlo al sistema degli assiomi insiemistici. Questo enunciato prende il nome di **assioma della scelta** e può essere formulato in questo modo:

Sia Y un insieme e per ogni $y \in Y$ sia dato un insieme non vuoto F_y in modo tale che per $y \neq z$ gli insiemi F_y ed F_z non abbiano elementi in comune. Allora esiste un insieme H contenuto nell'unione degli F_y tale che per ogni $y \in Y$ l'intersezione $H \cap F_y$ possieda esattamente un elemento.

Funzioni suriettive

Una funzione $f : X \rightarrow Y$ è detta *suriettiva* se ogni elemento di Y è immagine di qualche elemento di X , cioè se $\text{Imm } f = Y$.

Proposizione 4.1. *La funzione $f : X \rightarrow Y$ è suriettiva se e solo se esiste una funzione $h : Y \rightarrow X$ tale che $f \circ h = \text{id}_Y$.*

Dimostrazione: (1) f sia suriettiva. Stavolta la costruzione di h non è affatto elementare e richiede l'assioma della scelta.

Per ogni $y \in Y$ definiamo

$$F_y := \{(y, x) \in Y \times X \mid f(x) = y\}$$

Siccome la f è suriettiva, $F_y \neq \emptyset$ per ogni $y \in Y$. Inoltre F_y ed F_z per $y \neq z$ non hanno elementi in comune, come è evidente dalla definizione. Per l'assioma della scelta esiste un insieme H contenuto nell'unione degli F_y (e quindi contenuto in $Y \times X$ perché $F_y \subset Y \times X$ per ogni y) tale che l'intersezione $H \cap F_y$ per ogni $y \in Y$ possiede esattamente un elemento.

Quest'ultima condizione significa però per ogni $y \in Y$ esiste esattamente un $x \in X$ tale che $(y, x) \in H$.

Perciò la tripla $h := (Y, X, H)$ è una funzione ben definita.

Dobbiamo però dimostrare che $f \circ h = \text{id}_Y$, cioè che $f(h(y)) = y$ per ogni $y \in Y$. Ma $h(y)$ è proprio scelto in modo tale che $f(h(y)) = y$!

Riassumiamo la dimostrazione di questa prima parte: L'idea intuitiva è semplicemente che per ogni $y \in Y$ scegliamo un $x_y \in X$ tale che $f(x_y) = y$ (un tale x_y esiste sempre per l'ipotesi della suriettività della f). Poi formiamo $H := \{(y, x_y) \mid y \in Y\}$. Ma senza l'assioma della scelta non riusciamo a dimostrare che in questo modo abbiamo veramente definito un insieme H .

(2) La seconda direzione è di nuovo facile e istruttiva. Assumiamo che esista una funzione $h : Y \rightarrow X$ tale che $f \circ h = \text{id}_Y$. Ciò significa che per ogni $y \in Y$ abbiamo $y = f(h(y))$, per cui $y \in \text{Imm } f$.

Iniettività categoriale

L'algebra dei diagrammi commutativi è così importante in alcuni campi avanzati della matematica, che è argomento di una disciplina apposita, la *teoria delle categorie*. Formuliamo le proposizioni 3.1 e 4.1 in un modo un po' più vicino al linguaggio delle categorie.

Proposizione 4.2. *La funzione $f : X \rightarrow Y$ è iniettiva se e solo se per ogni insieme T e ogni coppia di funzioni $p : T \rightarrow X$ e $q : T \rightarrow X$ l'uguaglianza $f \circ p = f \circ q$ implica $p = q$.*

Dimostrazione: Il caso $X = \emptyset$ è banale; sia quindi $X \neq \emptyset$. (1) f sia iniettiva. Per la prop. 3.1 esiste una funzione $g : Y \rightarrow X$ tale che $g \circ f = \text{id}_X$. Allora, usando l'ipotesi e l'associatività della composizione di funzioni, abbiamo

$$\begin{aligned} p &= \text{id}_X \circ p = g \circ f \circ p = \\ &= g \circ f \circ q = \text{id}_X \circ q = q \end{aligned}$$

(2) Viceversa siano $x_1, x_2 \in X$ tali che $f(x_1) = f(x_2)$. Sia $T := \{0\}$. Definiamo le funzioni $T \xrightarrow{p} X$ e $T \xrightarrow{q} X$ ponendo $p(0) := x_1$ e $q(0) := x_2$. È chiaro che $f \circ p = f \circ q$. L'ipotesi implica $p = q$ e ciò a sua volta è possibile solo se $x_1 = x_2$.

Suriettività categoriale

Proposizione 4.3. *La funzione $f : X \rightarrow Y$ è suriettiva se e solo se per ogni insieme U e ogni coppia di funzioni $p : Y \rightarrow U$ e $q : Y \rightarrow U$ l'uguaglianza $p \circ f = q \circ f$ implica $p = q$.*

Dimostrazione: (1) f sia suriettiva. Per la prop. 4.1 esiste una funzione $h : Y \rightarrow X$ tale che $f \circ h = \text{id}_Y$. Di nuovo, usando l'ipotesi e l'associatività della composizione, abbiamo

$$\begin{aligned} p &= p \circ \text{id}_Y = p \circ f \circ h = \\ &= q \circ f \circ h = q \circ \text{id}_Y = q \end{aligned}$$

(2) Viceversa sia $U := \{1, 2\}$. La funzione $p : Y \rightarrow U$ sia la funzione costante che manda ogni y in 1. La funzione $q : Y \rightarrow U$ sia definita così:

$$q(y) = \begin{cases} 1 & \text{se } y \in \text{Imm } f \\ 2 & \text{se } y \notin \text{Imm } f \end{cases}$$

È chiaro che $p(f(x)) = q(f(x)) = 1$ per ogni $x \in X$, perché $f(x)$ appartiene sempre ad $\text{Imm } f$. Quindi $p \circ f = q \circ f$. Per ipotesi ciò implica $p = q$. Ma questo non sarebbe possibile se $\text{Imm } f \neq \emptyset$. Quindi necessariamente $\text{Imm } f = \emptyset$ e ciò significa che f è suriettiva.

Funzioni biiettive

Un'applicazione $f : X \rightarrow Y$ si chiama *biiettiva* se è allo stesso tempo iniettiva e suriettiva. Tralasciamo di nuovo il caso banale (ma un po' intricato) che X sia l'insieme vuoto e assumiamo quindi che $X \neq \emptyset$.

(1) f sia biiettiva. Per le prop. 3.1 e 4.1 esistono funzioni $g : Y \rightarrow X$ e $h : Y \rightarrow X$ tali che $g \circ f = \text{id}_X$ e $f \circ h = \text{id}_Y$. Dimostriamo prima che $g = h$. Infatti

$$g = g \circ \text{id}_Y = g \circ f \circ h = \text{id}_X \circ h = h$$

Abbiamo quindi un'applicazione $g : Y \rightarrow X$ tale che $g \circ f = \text{id}_X$ e $f \circ g = \text{id}_Y$. Dalle prop. 4.2 e 4.3 segue inoltre che g è univocamente determinata. Essa si chiama la *funzione inversa* di f e viene denotata con f^{-1} .

(2) Se viceversa esiste una funzione $Y \xrightarrow{g} X$ tale che $g \circ f = \text{id}_X$ e $f \circ g = \text{id}_Y$, allora f è iniettiva e suriettiva, quindi biiettiva, per le prop. 3.1 e 4.1. Abbiamo quindi il seguente risultato.

Proposizione 4.4. *La funzione $f : X \rightarrow Y$ è biiettiva se e solo se esiste una funzione $g : Y \rightarrow X$ t.c. $g \circ f = \text{id}_X$ e $f \circ g = \text{id}_Y$. g è allora univocamente determinata.*

Spazi di funzioni

Per insiemi X ed U definiamo U^X come l'insieme di tutte le funzioni $X \rightarrow U$. Un insieme della forma U^X o un suo sottoinsieme si chiama uno spazio di funzioni. Sia $f : X \rightarrow Y$ una funzione.

(1) Per ogni insieme T ed ogni funzione $r \in X^T$ possiamo formare la composizione $f_* (r) := f \circ r \in Y^T$, ottenendo così una funzione $f_* : X^T \rightarrow Y^T$.

(2) Per ogni insieme U ed ogni funzione $p \in U^Y$ possiamo formare la composizione $f^* (p) := p \circ f \in U^X$, ottenendo così una funzione $f^* : U^Y \rightarrow U^X$.

\mathbb{R}^n può essere identificato con $\mathbb{R}^{\{1, \dots, n\}}$.

Proprietà functoriali

Proposizione 4.5. *Siano date funzioni $X \xrightarrow{f} Y$ e $Y \xrightarrow{g} Z$. T sia un insieme. Come prima abbiamo applicazioni $X^T \xrightarrow{f_*} Y^T$ e $Y^T \xrightarrow{g_*} Z^T$. Allora*

$$(f \circ g)_* = f_* \circ g_*$$

Dimostrazione: Sia $r \in X^T$. Allora $(f \circ g)_* (r) = (f \circ g) \circ r = f \circ (g \circ r) = f_* (g \circ r) = f_* (g_* (r))$.

Proposizione 4.6. *Siano date funzioni $X \xrightarrow{f} Y$ e $Y \xrightarrow{g} Z$. U sia un insieme. Abbiamo applicazioni $U^Y \xrightarrow{f^*} U^X$ e $U^Z \xrightarrow{g^*} U^Y$. Allora*

$$(f \circ g)^* = g^* \circ f^*$$

Dimostrazione: Sia $p \in U^Z$. Allora $(f \circ g)^* (p) = p \circ (f \circ g) = (p \circ f) \circ g = g^* (p \circ f) = g^* (f^* (p))$.

Proposizione 4.7. *Sia $f : X \rightarrow Y$ una funzione. Allora:*

f è iniettiva $\iff f_* : X^T \rightarrow Y^T$ è iniettiva per ogni insieme T .

f è suriettiva $\iff f^* : U^Y \rightarrow U^X$ è iniettiva per ogni insieme U .

Dimostrazione: Ciò non è altro che una riformulazione delle prop. 4.2 e 4.3.

L'insieme delle parti

L'insieme delle parti di un insieme X è definito come l'insieme di tutti i sottoinsiemi di X . Ne fanno parte almeno X stesso e l'insieme vuoto \emptyset che sappiamo essere sottoinsieme di ogni insieme. Se X è vuoto e quindi coincide con \emptyset , l'insieme delle parti ha un solo elemento ed è uguale a $\{\emptyset\}$; se X possiede esattamente un elemento, $X \neq \emptyset$ e non ci sono sottoinsiemi di X diversi da X e \emptyset , quindi l'insieme delle parti è uguale all'insieme $\{\emptyset, X\}$ e possiede due elementi. In genere, se X è un insieme finito con n elementi, l'insieme delle parti consiste di 2^n elementi. L'insieme delle parti di X si denota con $\mathcal{P}(X)$.

X	$\mathcal{P}(X)$
\emptyset	$\{\emptyset\}$
$\{0\}$	$\{\emptyset, \{0\}\}$
$\{0, 1\}$	$\{\emptyset, \{0\}, \{1\}, \{0, 1\}\}$
$\{0, 1, 2\}$	$\{\emptyset, \{0\}, \{1\}, \{2\}, \{0, 1\}, \{0, 2\}, \{1, 2\}, \{0, 1, 2\}\}$

Una *funzione booleana* di n variabili è una funzione $f : \{0, 1\}^n \rightarrow \{0, 1\}$ o, equivalentemente, una funzione $f : \mathcal{P}(X) \rightarrow \{0, 1\}$, dove X è un insieme con n elementi.

Il numero delle funzioni booleane è esorbitante. Esistono 2^{2^n} funzioni booleane di n variabili, quindi 65536 funzioni booleane con 4 argomenti, più di 4 miliardi con 5 argomenti, più di 18 miliardi di miliardi con 6 argomenti (ogni volta che si aggiunge una variabile il numero delle funzioni booleane è il quadrato di quello precedente).

Le funzioni booleane appaiono, sotto vesti distinte, nella matematica pura, nello sviluppo di circuiti elettronici, nella ricerca medica. Su Google con *boolean gene expression cancer filetype:pdf* si trovano 900 files.

Geometria applicata

La geometria viene utilizzata in molti campi della tecnologia moderna: nella tomografia computerizzata, nella pianificazione di edifici, nella creazione di animazioni per film e pubblicità, nell'analisi dei movimenti di robot e satelliti.

La matematica è di grande aiuto nella grafica al calcolatore; conoscere le operazioni fondamentali della geometria (traslazioni, rotazioni, riflessioni, coordinate baricentriche, i vari tipi di proiezioni) permette di creare facilmente programmi di grafica con caratteristiche che talvolta mancano anche nei programmi di grandi produttori quando non sono stati sviluppati da matematici.

A livello più avanzato la geometria studia le varie rappresentazioni di curve, superficie e varietà geometriche di dimensione superiore, mediante rappresentazioni parametriche oppure sistemi di equazioni.

Si impara allora come passare da una rappresentazione parametrica a un sistema di equazioni che descrive la stessa varietà e viceversa; si studiano le funzioni che trasformano una varietà in un'altra.

Nella statistica medica i dati sono spesso punti in spazi di alta dimensione (se in una analisi di prove di sangue con la spettrometria di massa vengono rilevate le concentrazioni di 80 molecole, ogni prova corrisponde a un punto di \mathbb{R}^{80}). Se adesso vogliamo suddividere i pazienti in gruppi caratteristici (sani e malati nel caso più semplice) abbiamo bisogno di metodi geometrici per definire validi criteri di separazione.

La matematica in azienda

La matematica aziendale comprende da un lato la ricerca operativa, cioè l'ottimizzazione delle risorse di un'azienda o di un ente, una disciplina che si è evoluta dall'ottimizzazione lineare a metodi sempre più avanzati (ottimizzazione quadratica, convessa, dinamica, geometria dei numeri e ottimizzazione intera, topologia algebrica, programmazione logica), e dall'altro, soprattutto in campo bancario, la moderna sofisticata matematica finanziaria che deriva dalla matematica attuariale, ma utilizza strumenti matematici molto complicati.

Processi stocastici e serie temporali oltre che in matematica finanziaria hanno molte altre applicazioni in economia: osservazioni del carico elettrico della rete ENEL, cambiamenti demografici, andamento di mercati e borse.

La statistica matematica

Lo statista che lavora in un'azienda, nell'amministrazione pubblica o nella ricerca clinica, deve comprendere i compiti che gli vengono posti e deve essere in grado di interagire con i committenti. Nonostante ciò la statistica è di sua natura una disciplina matematica che si basa sul calcolo delle probabilità e richiede conoscenze tecniche in altri campi della matematica come analisi reale e complessa, analisi armonica, calcolo combinatorio (ad esempio per la pianificazione di esperimenti).

La matematica degli ingegneri

Problemi ingegneristici hanno quasi sempre una forte componente matematica: dalla teoria dei materiali all'elaborazione dei segnali, dall'interpretazione di misurazioni al controllo di qualità, da modelli per il corpo umano e i suoi movimenti (ad esempio nell'industria automobilistica, ma anche nell'industria tessile - in sartoria!) all'analisi strutturale di edifici e ponti, dai modelli matematici per i processi fisici e chimici in un altoforno all'ottimizzazione dell'illuminazione in impianti industriali, dallo studio dell'erosione nel letto di un fiume ai problemi inversi della geofisica (importanti per esempio anche nell'analisi strutturale di monumenti e edifici), dappertutto si utilizza la matematica.

Potremmo elencare tanti altri campi di applicazione: la geometria dei movimenti (cinematica) in robotica e nella costruzione di macchine, teoria dei sistemi e controllo ottimale nell'automazione, modellazione di reazioni chimiche nella chimica industriale, ottimizzazione strutturale di componenti di macchine o della composizione di punte per trapani da dentista, microstruttura di metalli, costruzione di autoveicoli, treni ed aerei, ottimizzazione di orari ferroviari, pianificazione urbanistica, telecomunicazioni.

Matematica e chimica

Il ruolo della matematica in chimica è in rapida crescita, essa viene applicata ad esempio nel disegno razionale di farmaci, nella selezione e sintesi di nuovi materiali, come guida nella ricerca di nuovi catalizzatori, nello sviluppo di algoritmi per la dinamica molecolare, nella risoluzione di problemi di ottimizzazione di conformazioni, per la comprensione del ripiegamento delle proteine, nello studio del trasporto di sostanze attraverso le membrane esterne ed interne della cellula (fondamentale per la farmacologia), nell'analisi del complicato avvolgimento delle molecole di DNA, nello studio della struttura di cristalli e quasicristalli, nella chimica quantistica.

La geometria e la topologia possono contribuire alla comprensione della struttura tridimensionale delle molecole, la teoria dei grafi permette non solo la visualizzazione dei legami chimici, ma può essere applicata alla rappresentazione di reazioni chimiche oppure nell'organizzazione di banche dati di molecole o della letteratura chimica; il calcolo combinatorio e la teoria dei gruppi intervengono nella chimica combinatoria, una tecnica sempre più utilizzata dall'industria farmaceutica. Il matematico può lavorare nello sviluppo di algoritmi per la trasformazione di Fourier per le applicazioni nella spettroscopia molecolare oppure nella chimica quantistica computazionale.

Equazioni differenziali parziali, analisi armonica, processi stocastici, statistica, analisi numerica, teoria combinatoria dei gruppi finiti, teoria dei grafi, quasiordini, teoria dei numeri (generazione di numeri casuali), geometria computazionale e grafica al calcolatore nella modellistica molecolare (computer aided molecular design e computer aided drug design): sono poche le discipline matematiche che non hanno interessanti applicazioni in chimica.

La dinamica dei fluidi

Uno dei campi più classici e allo stesso tempo più attuali della fisica matematica è la dinamica dei fluidi e dei gas. Essa richiede un ricco bagaglio di tecniche matematiche, soprattutto dall'analisi (equazioni differenziali parziali) e dalla geometria differenziale (calcolo tensoriale), oltre a solide conoscenze in meccanica dei continui e termodinamica (densità e viscosità e altre caratteristiche di un fluido o di un gas dipendono dalla temperatura e viceversa - un gas si scalda se viene compresso). È una disciplina molto vasta con tantissime applicazioni: costruzione di macchine (iniettori, turbine, ventilatori, pompe), ali di aerei, eliche di aerei e di navi, ruote a vento, modelli per nuovi materiali, flussi in medi porosi, raffreddamento di vetri, produzione di fibre plastiche, serbatoi di olio, ottimizzazione del caffè espresso, studio della formazione di vortici e turbolenze, combustione, detonazioni, modelli per il movimento di animali (pesci, serpenti, uccelli), modelli aerodinamici per la meteorologia (circolazioni e turbolenze atmosferiche, moto dei venti attorno a grandi catene di montagne, uragani, convezione termica nell'atmosfera) e l'agricoltura (moto dell'aria in piantagioni o foreste), aerodinamica di edifici, pianificazione di esperimenti aerodinamici e idrodinamici (costruzione di canali aerodinamici), previsione delle interazioni con l'aria di treni ad alta velocità, stima delle esposizioni al vento di un ponte. In medicina la fluidodinamica del sangue è un campo importante ma ancora piuttosto difficile (prevenzione di aneurismi e patologie circolatorie).

Geomatematica

Questo è un campo nuovo, molto bello e difficile della matematica. Funzioni speciali della geofisica matematica, funzioni armoniche sulla sfera, operatori pseudodifferenziali della geodesia matematica, metodi di approssimazione multivariata, splines, wavelets, metodi degli elementi finiti nella geodesia, determinazione del campo gravitazionale della Terra, analisi delle deformazioni della superficie terrestre, effetti della rifrazione atmosferica, determinazione del campo magnetico della Terra mediante l'analisi di dati trasmessi da satelliti, sono solo alcuni dei temi di questa interessante disciplina.

Malattie tropicali

È tipico per la natura viva che essa pone dei problemi che difficilmente possono essere perfettamente modellati con i metodi matematici classici sviluppati in genere per la fisica o l'ingegneria. Ciò da un lato vale naturalmente anche per le malattie tropicali come malaria, bilharziosi, filariosi, leishmaniosi, dall'altro lato queste malattie colpiscono ogni anno centinaia di milioni di persone, sono trascurate dalle ditte farmaceutiche (i pazienti non possono pagare) e richiedono quindi interventi ecologici o politici molto impegnativi. Alla pianificazione di questi interventi anche i modelli matematici possono contribuire e sicuramente la medicina tropicale è attraente per il suo fascino e per il lato umano.

Il re dei matematici



Carl Friedrich Gauß (1777-1855) è considerato il re dei matematici. La lettera β alla fine del nome è una s tedesca antica; il nome (talvolta scritto Gauss) si pronuncia gaos, simile a caos, ma con la g invece della c e con la o molto breve e legata alla a in modo che le due vocali formino un dittongo. Nessun altro matematico ha creato tanti concetti profondi ancora oggi importanti nelle discipline matematiche più avanzate (teoria dei numeri, geometria differenziale e geodesia matematica, teoria degli errori e statistica, analisi complessa). Il ritratto lo mostra a ventisei anni.

È stato forse il primo a concepire le geometrie non euclidee, ha dato una semplice interpretazione dei numeri complessi come punti del piano reale con l'addizione vettoriale e la moltiplicazione

$$(a, b) \cdot (c, d) = (ac - bd, bc + ad)$$

e ha dimostrato il teorema fondamentale dell'algebra (che afferma che ogni polinomio con coefficienti complessi possiede, nell'ambito dei numeri complessi, una radice), ha introdotto la distribuzione gaussiana del calcolo delle probabilità, ha conseguito importanti scoperte nella teoria dell'elettromagnetismo; è stato direttore dell'osservatorio astronomico di Göttinga.

L'algoritmo di eliminazione era noto nel 1759 a Lagrange (1736-1813) e già 2000 anni fa in Cina; Gauß lo ha usato nel suo lavoro sui moti celesti del 1809, in cui descrisse il metodo dei minimi quadrati, una tecnica di approssimazione ancora oggi universalmente utilizzata.

Le basi di Gröbner

Se si prova ad imitare l'algoritmo di eliminazione nella soluzione di sistemi polinomiali di grado maggiore di uno, ad esempio di

$$3x^2 + 5xy + 6x - 7y = 4$$

$$8x^2 + y^2 + x + 10y = 3$$

si incontrano molte difficoltà (provare). Il problema è stato risolto solo nel 1965 con l'introduzione delle basi di Gröbner (Wolfgang Gröbner, 1899-1980, era un matematico austriaco) e dell'algoritmo di Buchberger (Bruno Buchberger, nato 1942), molto più profondo e complicato dell'algoritmo di eliminazione nel caso lineare. Sistemi di equazioni polinomiali appaiono in molti campi della matematica con numerose applicazioni in ingegneria e statistica. Per questa ragione la geometria algebrica computazionale (compresa la geometria algebrica reale, importantissima e molto difficile) è oggi un campo estremamente attivo della matematica, intera-



Wolfgang Gröbner

gendo con la teoria dell'ottimizzazione, la teoria dei poliedri convessi, la crittografia, le equazioni differenziali parziali, la fisica teorica e, se ci si fida, la matematica finanziaria.

Bruno Buchberger nel 1987 ha fondato il RISC (Research Institute for Symbolic Computation, www.risc.uni-linz.ac.at/), che ha sede nel castello di Hagenberg a 25 km da Linz e di cui è stato direttore fino al 2003.

Il RISC è un istituto dell'università di Linz e ospita circa 70 collaboratori, tra cui molti studenti. Le attività, iniziate con la geometria algebrica algoritmica nell'intento di sfruttare le possibilità offerte dall'algoritmo di Buchberger, sono molto varie, ma hanno tutte in qualche modo da fare con la risoluzione di equazioni e disequazioni, talvolta in senso molto lato confinando con la logica computazionale, la dimostrazione automatica di teoremi, l'intelligenza artificiale, la robotica e la chimica industriale.



Il castello di Hagenberg

In questo numero

- 6 Il re dei matematici
Le basi di Gröbner
Equazioni lineari in una incognita
- 7 Sistemi astratti
Due equazioni lineari in due incognite
- 8 Esempi
La forma generale della regola di Cramer
Determinanti
- 9 L'algoritmo di eliminazione di Gauß
- 10 Sistemi con più di una soluzione
L'insieme delle soluzioni di un sistema lineare
Esercizi 1-13

Equazioni lineari in una incognita

Siano dati numeri reali a e b . Cercare di risolvere l'equazione $ax = b$ nell'incognita x significa cercare tutti i numeri reali x per i quali $ax = b$. Per $a \neq 0$ la soluzione è unica e data da $x = \frac{b}{a}$.

Dimostrazione: È chiaro che le seguenti equazioni sono equivalenti, cioè se x soddisfa una di esse, allora le soddisfa tutte:

$$ax = b$$

$$\frac{ax}{a} = \frac{b}{a}$$

$$x = \frac{b}{a}$$

È evidente che nel nostro ragionamento solo le proprietà algebriche formali dei numeri reali sono state usate e che rimane quindi valido, così come le considerazioni successive, se lavoriamo con numeri razionali o numeri complessi o altri insiemi di numeri con quelle corrispondenti proprietà. Si vede comunque anche che abbiamo avuto bisogno di poter dividere per un numero $\neq 0$, e quindi il risultato non è vero nell'ambito dei numeri naturali o interi.

Un insieme, su cui sono date due operazioni di addizione e di moltiplicazione che soddisfano le familiari regole di calcolo e in cui si può sempre dividere per un elemento $\neq 0$, in matematica si chiama un *campo*. Quindi l'algoritmo di eliminazione di Gauß rimane valido per sistemi di equazioni lineari i cui coefficienti appartengono a un campo qualsiasi.

Elenchiamo le regole che devono valere in un campo; verranno stabilite e studiate più dettagliatamente negli altri corsi.

$$a + (b + c) = (a + b) + c$$

$$a + b = b + a$$

$$a + 0 = a$$

$$a + (-a) = 0$$

$$a(bc) = (ab)c$$

$$ab = ba$$

$$a1 = a$$

$$\frac{1}{a}a = a \text{ per } a \neq 0$$

$$a(b + c) = ab + ac$$

Sistemi astratti

Definizione 7.1. Siano dati un insieme X ed una funzione $f : X \rightarrow \mathbb{R}$. Denotiamo con $(f = 0)$ l'insieme degli zeri di f :

$$(f = 0) := \{x \in X \mid f(x) = 0\}.$$

Più in generale, se sono date m funzioni $f_1, \dots, f_m : X \rightarrow \mathbb{R}$, con

$$(f_1 = 0, \dots, f_m = 0)$$

denotiamo l'insieme

$$(f_1 = 0) \cap \dots \cap (f_m = 0)$$

degli zeri comuni di queste funzioni. Questa notazione è molto usata in calcolo delle probabilità.

Osservazione 7.2. X sia un insieme. Abbiamo visto a pagina 4 che con \mathbb{R}^X si denota l'insieme di tutte le funzioni a valori reali definite su X . Possiamo formare somme di funzioni, moltiplicare funzioni con un numero reale o con un'altra funzione, e costruire combinazioni lineari di funzioni in questo spazio nel modo seguente.

Siano $f, g, f_1, \dots, f_m \in \mathbb{R}^X$ ed $\alpha, \alpha_1, \dots, \alpha_m \in \mathbb{R}$. Allora:

$$f + g := \bigcirc_x f(x) + g(x)$$

$$\alpha f := \bigcirc_x \alpha f(x)$$

$$\alpha_1 f_1 + \dots + \alpha_m f_m := \bigcirc_x \alpha_1 f_1(x) + \dots + \alpha_m f_m(x)$$

Teorema 7.3. Siano dati un insieme X ed m funzioni $f_1, \dots, f_m : X \rightarrow \mathbb{R}$. Siano $\alpha_1, \dots, \alpha_m$ numeri reali con $\alpha_1 \neq 0$. Allora gli insiemi

$$(f_1 = 0, \dots, f_m = 0)$$

e

$$(\alpha_1 f_1 + \alpha_2 f_2 + \dots + \alpha_m f_m = 0, f_2 = 0, \dots, f_m = 0)$$

coincidono.

Dimostrazione. Per dimostrare l'uguaglianza tra i due insiemi, dobbiamo dimostrare che ogni elemento del primo insieme è anche elemento del secondo, e che ogni elemento del secondo insieme è elemento del primo.

Sia quindi x un elemento fissato di X .

(1) Sia $f_1(x) = 0, \dots, f_m(x) = 0$. È chiaro che allora anche

$$\begin{aligned} \alpha_1 f_1(x) + \alpha_2 f_2(x) + \dots + \alpha_m f_m(x) &= 0 \\ f_2(x) &= 0 \\ \dots \\ f_m(x) &= 0. \end{aligned}$$

(2) Sia viceversa

$$\begin{aligned} \alpha_1 f_1(x) + \alpha_2 f_2(x) + \dots + \alpha_m f_m(x) &= 0 \\ f_2(x) &= 0 \\ \dots \\ f_m(x) &= 0. \end{aligned}$$

Dobbiamo dimostrare che $f_1(x) = 0$. Ma se sostituiamo $f_2(x) = 0, \dots, f_m(x) = 0$ nella prima equazione, vediamo che

$$\alpha_1 f_1(x) = 0.$$

Qui possiamo adesso applicare la nostra ipotesi che $\alpha_1 \neq 0$. Essa implica $f_1(x) = 0$.

Attenzione: Il ragionamento non vale più, se non sappiamo che $\alpha_1 \neq 0$!

Nota 7.4. Nel seguito avremo $X = \mathbb{R}^2$ oppure, più in generale, $X = \mathbb{R}^n$. Nel primo caso scriveremo gli elementi di X nella forma (x, y) , cosicché x denoterà la prima coordinata di un elemento e non l'elemento stesso.

Gli elementi di \mathbb{R}^3 saranno scritti nella forma (x, y, z) , gli elementi di \mathbb{R}^n nella forma $x = (x_1, \dots, x_n)$. Questo passaggio da una notazione all'altra è frequente e diventerà presto familiare.

Due equazioni lineari in due incognite

Siano dati numeri reali $a_1, b_1, c_1, a_2, b_2, c_2$. Risolvere il sistema lineare

$$\begin{aligned} a_1 x + b_1 y &= c_1 \\ a_2 x + b_2 y &= c_2 \end{aligned}$$

significa trovare tutte le coppie (x, y) di numeri reali che soddisfano entrambe le equazioni. Per poter applicare il teorema 7.3 introduciamo le funzioni $f_1, f_2 : \mathbb{R}^2 \rightarrow \mathbb{R}$ definite da

$$f_1(x, y) := a_1 x + b_1 y - c_1$$

$$f_2(x, y) := a_2 x + b_2 y - c_2$$

cosicché l'insieme delle soluzioni cercate coincide con $(f_1 = 0, f_2 = 0)$. Lasciamo come esercizio il caso molto facile che $a_1 = 0$.

Assumiamo quindi che $a_1 \neq 0$ e definiamo la funzione

$$f_3 := a_1 f_2 - a_2 f_1$$

Per il teorema 7.3 abbiamo

$$(f_1 = 0, f_2 = 0) = (f_1 = 0, f_3 = 0)$$

perché è f_2 che sicuramente appare con un coefficiente $\neq 0$ in f_3 . Scritta per esteso l'equazione $f_3 = 0$ diventa

$$\underline{a_1 a_2 x} + a_1 b_2 y - a_1 c_2 - \underline{a_2 a_1 x} - a_2 b_1 y + a_2 c_1 = 0$$

cioè

$$(a_1 b_2 - a_2 b_1) y = a_1 c_2 - a_2 c_1,$$

e quindi le soluzioni del sistema originale coincidono con le soluzioni del sistema

$$\begin{aligned} a_1 x + b_1 y &= c_1 \\ (a_1 b_2 - a_2 b_1) y &= a_1 c_2 - a_2 c_1 \end{aligned}$$

Il numero $a_1 b_2 - a_2 b_1$ si chiama il **determinante** del sistema; lasciamo ancora come esercizio il caso che il determinante si annulli; se è invece $\neq 0$, allora la seconda equazione significa che

$$y = \frac{a_1 c_2 - a_2 c_1}{a_1 b_2 - a_2 b_1}.$$

Se per numeri reali a, b, c, d poniamo $\begin{vmatrix} a & b \\ c & d \end{vmatrix} := ad - bc$, possiamo scrivere

$$y = \frac{\begin{vmatrix} a_1 & c_1 \\ a_2 & c_2 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}}$$

Vediamo che anche il numeratore ha la forma di un determinante; infatti si ottiene dal denominatore sostituendo per la seconda colonna la colonna che costituisce il lato destro del sistema.

A questo punto possiamo calcolare anche x . Ricordando che $a_1 \neq 0$, otteniamo

$$\begin{aligned} x &= \frac{c_1 - b_1 y}{a_1} = \frac{c_1 - b_1 \frac{a_1 c_2 - a_2 c_1}{a_1 b_2 - a_2 b_1}}{a_1} = \\ &= \frac{a_1 b_2 c_1 - a_2 b_1 c_1 - b_1 a_1 c_2 + b_1 a_2 c_1}{a_1 (a_1 b_2 - a_2 b_1)} = \\ &= \frac{a_1 b_2 c_1 - b_1 a_1 c_2}{a_1 (a_1 b_2 - a_2 b_1)} = \frac{b_2 c_1 - b_1 c_2}{a_1 b_2 - a_2 b_1} = \frac{\begin{vmatrix} c_1 & b_1 \\ c_2 & b_2 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}} \end{aligned}$$

Quindi nel caso che il determinante del sistema sia $\neq 0$, il sistema possiede un'unica soluzione data da

$$x = \frac{\begin{vmatrix} c_1 & b_1 \\ c_2 & b_2 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}} \quad y = \frac{\begin{vmatrix} a_1 & c_1 \\ a_2 & c_2 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}}$$

Si osservi che il numeratore di x si ottiene anch'esso dal determinante del sistema, sostituendo stavolta la prima colonna con il lato destro del sistema. Questo risultato è molto importante per l'algebra lineare e può essere generalizzato a più dimensioni; è noto come *regola di Cramer*.

Esempi

Risolviamo con la regola di Cramer il sistema

$$\begin{aligned} 3x - 2y &= 8 \\ x + 6y &= 5 \end{aligned}$$

Il determinante del sistema è $\begin{vmatrix} 3 & -2 \\ 1 & 6 \end{vmatrix} = 18 + 2 = 20$, quindi diverso da 0, per cui

$$x = \frac{\begin{vmatrix} 8 & -2 \\ 5 & 6 \end{vmatrix}}{20} = \frac{48 + 10}{20} = \frac{58}{20}$$

$$y = \frac{\begin{vmatrix} 3 & 8 \\ 1 & 5 \end{vmatrix}}{20} = \frac{15 - 8}{20} = \frac{7}{20}$$

Esercizio. Risolvere da soli

$$4x + 3y = 10$$

$$2x + 9y = 7$$

Esercizio. Perché non si può applicare la regola di Cramer al sistema

$$x + 3y = 1$$

$$2x + 6y = 4$$

Eppure non è difficile trovare “tutte” le soluzioni. Perché ho messo “tutte” tra virgolette? E perché è anche (quasi) facile trovare tutte le soluzioni di

$$x + 3y = 2$$

$$2x + 6y = 4$$

La forma generale della regola di Cramer

Sia dato un sistema di n equazioni lineari in n incognite (quindi il numero delle equazioni è uguale al numero delle incognite):

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= c_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= c_2 \\ &\vdots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= c_n \end{aligned}$$

Anche in questo caso più generale si può definire il *determinante* del sistema, un numero che viene denotato con

$$D := \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix}$$

e si dimostrerà nel corso di Geometria I che questo determinante è $\neq 0$ se e solo se il sistema possiede un'unica soluzione che in tal caso è data da

$$x_1 = \frac{\begin{vmatrix} c_1 & a_{12} & \dots & a_{1n} \\ c_2 & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ c_n & a_{n2} & \dots & a_{nn} \end{vmatrix}}{D}$$

$$x_2 = \frac{\begin{vmatrix} a_{11} & c_1 & \dots & a_{1n} \\ a_{21} & c_2 & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & c_n & \dots & a_{nn} \end{vmatrix}}{D}$$

$$x_n = \frac{\begin{vmatrix} a_{11} & a_{12} & \dots & c_1 \\ a_{21} & a_{22} & \dots & c_2 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & c_n \end{vmatrix}}{D}$$

x_i è quindi un quoziente il cui numeratore si ottiene dal determinante del sistema, sostituendo la i -esima colonna con il lato destro del sistema.

Determinanti

Conosciamo già i determinanti 2×2 : $\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix} = a_1b_2 - a_2b_1$.

Definizione 8.1. Per induzione definiamo i determinanti di ordine superiore:

$$\begin{aligned} \begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix} &:= a_1 \begin{vmatrix} b_2 & c_2 \\ b_3 & c_3 \end{vmatrix} - a_2 \begin{vmatrix} b_1 & c_1 \\ b_3 & c_3 \end{vmatrix} + a_3 \begin{vmatrix} b_1 & c_1 \\ b_2 & c_2 \end{vmatrix} \\ \begin{vmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \\ a_4 & b_4 & c_4 & d_4 \end{vmatrix} &:= a_1 \begin{vmatrix} b_2 & c_2 & d_2 \\ b_3 & c_3 & d_3 \\ b_4 & c_4 & d_4 \end{vmatrix} - a_2 \begin{vmatrix} b_1 & c_1 & d_1 \\ b_3 & c_3 & d_3 \\ b_4 & c_4 & d_4 \end{vmatrix} + \\ &+ a_3 \begin{vmatrix} b_1 & c_1 & d_1 \\ b_2 & c_2 & d_2 \\ b_4 & c_4 & d_4 \end{vmatrix} - a_4 \begin{vmatrix} b_1 & c_1 & d_1 \\ b_2 & c_2 & d_2 \\ b_3 & c_3 & d_3 \end{vmatrix} \end{aligned}$$

e così via. Si noti l'alternanza dei segni. I determinanti hanno molte proprietà importanti che verranno studiate nel corso di Geometria. Qui ci limiteremo a determinanti 2×2 e 3×3 , per i quali mostriamo alcune semplici regole, valide anche per determinanti $n \times n$, se riformulate in modo naturale.

Lemma 8.2. Se in un determinante 2×2 scambiamo tra di loro due righe o due colonne, il determinante si moltiplica con -1 .

Dimostrazione. Immediata.

Lemma 8.3. Un determinante 3×3 può essere calcolato anche secondo la regola

$$\begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix} := a_1 \begin{vmatrix} b_2 & c_2 \\ b_3 & c_3 \end{vmatrix} - b_1 \begin{vmatrix} a_2 & c_2 \\ a_3 & c_3 \end{vmatrix} + c_1 \begin{vmatrix} a_2 & b_2 \\ a_3 & b_3 \end{vmatrix}$$

Dimostrazione. Le due espansioni si distinguono in

$$-a_2 \begin{vmatrix} b_1 & c_1 \\ b_3 & c_3 \end{vmatrix} + a_3 \begin{vmatrix} b_1 & c_1 \\ b_2 & c_2 \end{vmatrix} = -a_2b_1c_3 + a_2b_3c_1 + a_3b_1c_2 - a_3b_2c_1$$

e

$$-b_1 \begin{vmatrix} a_2 & c_2 \\ a_3 & c_3 \end{vmatrix} + c_1 \begin{vmatrix} a_2 & b_2 \\ a_3 & b_3 \end{vmatrix} = -b_1a_2c_3 + b_1a_3c_2 + c_1a_2b_3 - c_1a_3b_2$$

che però, come vediamo, danno lo stesso risultato.

Lemma 8.4. Se si scambiano due righe o due colonne in una matrice 3×3 , il determinante si moltiplica per -1 .

Dimostrazione. Ciò, per il lemma 8.2, è evidente per lo scambio della seconda e della terza colonna e, per il lemma 8.3, anche per lo scambio della seconda e della terza riga. Se invece scambiamo la prima e la seconda colonna, otteniamo il determinante

$$\begin{vmatrix} b_1 & a_1 & c_1 \\ b_2 & a_2 & c_2 \\ b_3 & a_3 & c_3 \end{vmatrix} := b_1 \begin{vmatrix} a_2 & c_2 \\ a_3 & c_3 \end{vmatrix} - a_1 \begin{vmatrix} b_2 & c_2 \\ b_3 & c_3 \end{vmatrix} + c_1 \begin{vmatrix} b_2 & a_2 \\ b_3 & a_3 \end{vmatrix}$$

uguale, come si vede subito, al determinante originale moltiplicato per -1 . Gli altri casi seguono adesso applicando le regole già dimostrate.

Lemma 8.5. Se in un determinante appaiono due righe o due colonne uguali, allora il determinante è uguale a 0.

Dimostrazione. Ciò per un determinante 2×2 è ovvio, e se ad esempio sono uguali le ultime due colonne, l'enunciato segue (usando il caso 2×2) dalla formula di espansione anche per i determinanti 3×3 , e poi dal caso 3×3 anche per i determinanti 4×4 ecc.

Esempio. Verificare con calcoli a mano che

$$\begin{vmatrix} a_1 & a_1 & c_1 \\ a_2 & a_2 & c_2 \\ a_3 & a_3 & c_3 \end{vmatrix} = 0$$

e che

$$\begin{vmatrix} a_1 & b_1 + a_1 & c_1 \\ a_2 & b_2 + a_2 & c_2 \\ a_3 & b_3 + a_3 & c_3 \end{vmatrix} = \begin{vmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{vmatrix}$$

L'ultima uguaglianza è un caso speciale di un'altra proprietà fondamentale dei determinanti.

L'algoritmo di eliminazione di Gauß

La teoria dei determinanti e la regola di Cramer hanno una grandissima importanza teorica, ma non possono essere utilizzate se non per sistemi in due o al massimo tre incognite. Inoltre la regola di Cramer si applica solo al caso di un sistema quadratico. Esiste invece un metodo molto efficiente (anche nel calcolo a mano) per la risoluzione di sistemi di equazioni lineari, che viene detto *algoritmo di eliminazione di Gauß* e che consiste nella sistematica applicazione del teorema 7.3.

Esempio 9.1. Consideriamo il sistema

$$\begin{aligned} 3x + 2y - z &= 10 \dots f_1 \\ 4x - 9y + 2z &= 6 \dots f_2 \\ x + y - 14z &= 2 \dots f_3 \end{aligned}$$

In analogia a quanto abbiamo fatto a pagina 7 per il sistema 2×2 le funzioni f_1, f_2 ed f_3 sono definite da

$$\begin{aligned} f_1(x, y) &:= 3x + 2y - z - 10 \\ f_2(x, y) &:= 4x - 9y + 2z - 6 \\ f_3(x, y) &:= x + y - 14z - 2 \end{aligned}$$

Le indichiamo alla destra delle equazioni corrispondenti. Con la notazione che abbiamo introdotto nell'osservazione 7.2 poniamo

$$f_4 := 4f_1 - 3f_2$$

Per il teorema 7.3 allora

$$(f_1 = 0, f_2 = 0, f_3 = 0) = (f_4 = 0, f_2 = 0, f_3 = 0)$$

perché il coefficiente con cui f_1 appare in f_4 è diverso da 0. Esplicitamente $f_4 = 0$ equivale a

$$4(3x + 2y - z) - 3(4x - 9y + 2z) = 4 \cdot 10 - 3 \cdot 6$$

cioè a

$$35y - 10z = 22.$$

Se chiamiamo due sistemi **equivalenti** quando hanno le stesse soluzioni, possiamo dire che il sistema originale è equivalente al sistema

$$\begin{aligned} 4x - 9y + 2z &= 6 \dots f_2 \\ x + y - 14z &= 2 \dots f_3 \\ 35y - 10z &= 22 \dots f_4 \end{aligned}$$

Nell'ultima equazione la variabile x in f_4 è sparita, è stata *eliminata*. Ripetiamo questa operazione sostituendo la funzione f_2 con

$$f_5 := f_2 - 4f_3$$

Ciò è possibile perché in f_5 la f_2 appare con un coefficiente $\neq 0$. Esplicitamente $f_5 = 0$ significa

$$4x - 9y + 2z - 4(x + y - 14z) = 6 - 4 \cdot 2$$

cioè

$$-13y + 58z = -2.$$

Perciò il sistema originale ha le stesse soluzioni come il sistema

$$\begin{aligned} x + y - 14z &= 2 \dots f_3 \\ 35y - 10z &= 22 \dots f_4 \\ -13y + 58z &= -2 \dots f_5 \end{aligned}$$

Adesso formiamo $f_6 := 13f_4 + 35f_5$ che può sostituire sia la f_4 che la f_5 . Possiamo togliere la f_5 . $f_6 = 0$ è equivalente a

$$13(35y - 10z) + 35(-13y + 58z) = 13 \cdot 22 + 35 \cdot (-2),$$

cioè a

$$1900z = 216.$$

Otteniamo così il sistema

$$\begin{aligned} x + y - 14z &= 2 \dots f_3 \\ 35y - 10z &= 22 \dots f_4 \\ 1900z &= 216 \dots f_6 \end{aligned}$$

che è ancora equivalente a quello originale. Ma adesso vediamo che nell'ultima equazione è stata eliminata anche la y ed è rimasta solo la z che possiamo così calcolare direttamente:

$$z = \frac{216}{1900} = 0.11368,$$

poi, usando $f_4 = 0$, otteniamo

$$y = \frac{22 + 10z}{35} = 0.66105,$$

e infine dal $f_3 = 0$

$$x = -y + 14z + 2 = 2.930526.$$

Nella pratica si userà uno schema in cui vengono scritti, nell'ordine indicato dall'ordine delle variabili, solo i coefficienti. Nell'esempio appena trattato i conti verrebbero disposti nel modo seguente:

3	2	-1	10	f_1	✓
4	-9	2	6	f_2	✓
1	1	-14	2	f_3	
0	35	-10	22	$f_4 = 4f_1^* - 3f_2$	
0	-13	58	-2	$f_5 = f_2^* - 4f_3$	✓
0	0	1900	216	$f_6 = 13f_4 + 35f_5^*$	

L'asterisco indica ogni volta l'equazione cancellata in quel punto; l'uncino a destra di un'equazione significa che questa equazione è stata cancellata. Nei conti a mano spesso si preferirà forse cancellare la riga con un tratto orizzontale piuttosto di usare l'uncino.

Come si vede, nell'algoritmo cerchiamo prima di ottenere un sistema equivalente all'originale in cui tutti i coefficienti tranne al massimo uno nella prima colonna sono $= 0$, poi, usando le equazioni rimaste, applichiamo lo stesso procedimento alla seconda colonna (non modificando più però quella riga a cui corrisponde quell'eventuale coefficiente $\neq 0$ nella prima colonna), ecc. È chiaro che il procedimento termina sempre: alle m equazioni iniziali si aggiungono prima $m - 1$, poi $m - 2$, poi $m - 3$, ecc.

L'insieme delle soluzioni rimane sempre lo stesso; le equazioni cancellate naturalmente sono superflue e non vengono più usate. Quindi, se il sistema non ha soluzioni o più di una soluzione, riusciamo a scoprire anche questo.

Esempio 9.2. Consideriamo il sistema

$$\begin{aligned} 2x_1 - 5x_2 + 3x_3 - x_4 &= 1 \\ x_1 + 4x_2 - x_3 + 2x_4 &= 3 \\ 3x_1 - x_2 + 2x_3 + x_4 &= 7 \end{aligned}$$

Applichiamo il nostro schema:

2	-5	3	-1	1	f_1	✓
1	4	-1	2	3	f_2	
3	-1	2	1	7	f_3	✓
0	-13	5	-5	-5	$f_4 = f_1^* - 2f_2$	✓
0	-13	5	-5	-2	$f_5 = f_3^* - 3f_2$	
0	0	0	0	-3	$f_6 = f_4^* - f_5$	

Il sistema dato è quindi equivalente al sistema

$$\begin{aligned} x_1 + 4x_2 - x_3 + 2x_4 &= 3 \\ -13x_2 + 5x_3 - 5x_4 &= -2 \\ 0 &= -3 \end{aligned}$$

In particolare siamo arrivati alla contraddizione $0 = -3$, quindi il sistema non ha soluzione.

Sistemi con più di una soluzione

Consideriamo il sistema

$$\begin{aligned} 4x - y + 3z &= 5 \\ x + 2y - 10z &= 4 \\ 6x + 3y - 17z &= 13 \end{aligned}$$

Usiamo di nuovo il nostro schema di calcolo:

4	-1	3	5	f_1	✓
1	2	-10	4	f_2	
6	3	-17	13	f_3	✓
0	-9	43	-11	$f_4 = f_1^* - 4f_2$	✓
0	9	-43	11	$f_5 = 6f_2 - f_3^*$	
0	0	0	0	$f_6 = f_4^* + f_5$	

Stavolta non abbiamo una contraddizione, ma un'ultima equazione $0 = 0$ superflua, quindi siamo rimasti con due equazioni per tre incognite:

$$\begin{aligned} x + 2y - 10z &= 4 \\ 9y - 43z &= 11 \end{aligned}$$

Per ogni valore t di z possiamo risolvere

$$\begin{aligned} y &= \frac{11 + 43t}{9} \\ x &= -2y + 10t + 4 = \frac{-22 - 86t}{9} + 10t + 4 = \\ &= \frac{-22 - 86t + 90t + 36}{9} = \frac{14}{9} + \frac{4}{9}t \end{aligned}$$

e vediamo che l'insieme delle soluzioni è una retta nello spazio \mathbb{R}^3 con la rappresentazione parametrica

$$\begin{aligned} x(t) &= \frac{14}{9} + \frac{4}{9}t \\ y(t) &= \frac{11}{9} + \frac{43}{9}t \\ z(t) &= t \end{aligned}$$

Per ogni numero reale t si ottiene un punto $(x(t), y(t), z(t)) \in \mathbb{R}^3$ che è una soluzione del nostro sistema, e viceversa ogni soluzione è di questa forma.

L'insieme delle soluzioni di un sistema lineare

Negli esempi visti finora abbiamo trovato sistemi che non avevano soluzioni, oppure un'unica soluzione (descrittivi cioè un unico punto nello spazio), oppure, nell'ultimo esempio, una retta di soluzioni.

Ciò vale per ogni sistema di equazioni lineari: l'insieme delle soluzioni è sempre o vuoto (nessuna soluzione), oppure un solo punto, oppure una retta, oppure un piano, oppure uno spazio affine tridimensionale ecc., e viceversa ogni insieme di questa forma può essere descritto da un sistema di equazioni lineari. La dimostrazione di questo teorema e la definizione precisa del concetto di spazio affine verranno date nel corso di Geometria I.

Nonostante l'efficienza dell'algoritmo di eliminazione che permette la risoluzione abbastanza agevole di sistemi lineari non troppo grandi (con un po' di pazienza si possono risolvere anche sistemi 10×10 a mano) la pratica è più complicata. Nelle applicazioni reali si affrontano sistemi con decine di migliaia di equazioni e variabili e non solo il tempo di calcolo, ma anche l'accumularsi di errori di arrotondamento nei calcoli approssimati che il software normalmente utilizza possono creare grandi problemi.

Piccoli errori, spesso inevitabili, nei dati in entrata (ad esempio nei coefficienti a_{ij} e b_i del nostro sistema) possono provocare in taluni casi, che bisogna riconoscere e controllare, grandi cambiamenti nelle soluzioni. Così il sistema

$$\begin{aligned} 4x + 1.99y &= 2.01 \\ 2x + y &= 1 \end{aligned}$$

possiede un'unica soluzione $x = 1, y = -1$, ma se lo cambiamo di poco,

$$\begin{aligned} 4x + 2y &= 2 \\ 2x + y &= 1 \end{aligned}$$

il determinante si annulla e l'insieme delle soluzioni è dato da $y = 1 - 2x$, e quindi le soluzioni non sono più univocamente determinate e possono essere arbitrariamente distanti dalla soluzione $(1, -1)$ del primo sistema.

Esercizi per i compiti scritti

1. Sia $f := \bigcirc_x 3x^4 - 2x^2 + 10x - 6 : \mathbb{R} \rightarrow \mathbb{R}$. Calcolare $f(2)$.
2. Siano $f := \bigcirc_x 3x^2 + 4$ e $g := \bigcirc_x 6x - 1$ considerate come funzioni $\mathbb{R} \rightarrow \mathbb{R}$. Calcolare $f \circ g$ e $g \circ f$.
3. Siano $f := \bigcirc_{(x,y)} x + y^3 : \mathbb{R}^2 \rightarrow \mathbb{R}$ e $g := \bigcirc_x x^2 : \mathbb{R} \rightarrow \mathbb{R}$. Calcolare $g \circ f$.

4. Risolvere con la regola di Cramer il sistema

$$\begin{aligned} 3x - 2y &= 9 \\ 2x + 7y &= 5 \end{aligned}$$

5. Calcolare il determinante

$$\begin{vmatrix} 3 & 8 & 2 \\ 6 & 2 & 1 \\ 4 & 2 & 1 \end{vmatrix}$$

6. Calcolare il determinante

$$\begin{vmatrix} 2 & 8 & 12 \\ 5 & 1 & 11 \\ 4 & 3 & 11 \end{vmatrix}$$

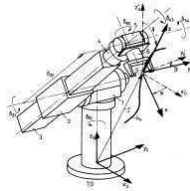
Risolvere i sistemi con l'algoritmo di Gauß usando lo schema.

7.
$$\begin{aligned} 2x_1 - 4x_2 + x_3 - x_4 &= 8 \\ x_1 + 5x_2 - x_3 + 2x_4 &= 0 \\ 2x_1 - x_2 + 4x_3 + x_4 &= 6 \\ 4x_1 + x_2 - x_3 + 3x_4 &= 10 \end{aligned}$$
8.
$$\begin{aligned} x_1 + 2x_2 + 3x_3 + 2x_4 &= 8 \\ 4x_1 + x_2 + 2x_3 + x_4 &= 5 \\ 4x_1 + 2x_2 + 3x_3 + 4x_4 &= 0 \\ x_1 - 2x_2 - 2x_3 - 2x_4 &= 4 \end{aligned}$$
9.
$$\begin{aligned} 8x + 7y + 15z &= 10 \\ 2x + 3y + 3z &= 4 \\ 3x + 2y + 6z &= 3 \end{aligned}$$
10.
$$\begin{aligned} 2x_1 + x_2 + x_3 + x_4 + 2x_5 &= 1 \\ 2x_1 + x_2 + x_3 + x_4 + x_5 &= 2 \\ x_1 - x_2 + 2x_3 + x_4 + 2x_5 &= 9 \\ x_1 + 2x_2 + 2x_3 + x_4 + 3x_5 &= 4 \\ 3x_1 + x_2 + x_4 + 2x_5 &= -2 \end{aligned}$$
11.
$$\begin{aligned} 4x_1 + 8x_2 + 17x_3 + 100x_4 + 2x_5 &= 0 \\ 3x_1 + 5x_2 + 32x_3 + 33x_4 + 34x_5 &= 1 \\ 2x_2 + x_3 + 100x_4 + 2x_5 &= 4 \\ 2x_1 + 3x_2 + 15x_3 + 6x_4 + 5x_5 &= -1 \\ 4x_1 + 8x_2 + x_3 + 7x_4 + 2x_5 &= 2 \end{aligned}$$
12.
$$\begin{aligned} 121x_1 + 58x_2 + 85x_3 + 45x_4 &= 1 \\ 2x_1 + 175x_2 + 619x_3 + 218x_4 &= 0 \\ 415x_1 + 26x_2 + 510x_3 + 77x_4 &= 2 \\ 8x_1 + 15x_2 + 315x_3 + 418x_4 &= -1 \end{aligned}$$
13.
$$\begin{aligned} 6x_1 + 2x_2 + 3x_3 + 5x_4 + 2x_5 + x_6 &= 1 \\ 3x_1 + x_2 + x_3 + 10x_4 + x_5 + 2x_6 &= 0 \\ 3x_1 + 2x_2 + x_3 + 2x_4 + x_5 + 2x_6 &= 0 \\ 2x_1 + x_3 + 5x_4 + x_5 + 4x_6 &= 0 \\ 3x_1 + x_2 + 2x_5 + x_6 &= 0 \\ 6x_1 + x_3 + 5x_4 + x_5 + 4x_6 &= 0 \end{aligned}$$

Trigonometria oggi

Dai piani di studio, soprattutto nell'università, la trigonometria è sparita da molto tempo. Ma questa disciplina, una delle più antiche della matematica, è ancora oggi una delle più importanti.

Mentre almeno gli elementi della trigonometria piana vengono insegnati nelle scuole, la trigonometria sferica è ormai conosciuta pochissimo anche tra i matematici di professione. Eppure le applicazioni sono tantissime: nautica, cartografia, geodesia e geoinformatica, astronomia, cristallografia, classificazione dei movimenti nello spazio, cinematica e quindi robotica e costruzione di macchine, grafica al calcolatore.

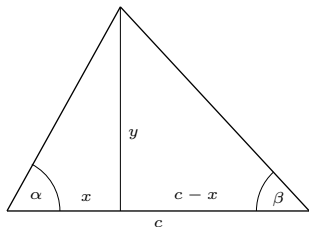


L'immagine rappresenta un robot con i suoi movimenti; trovata su www.igm.rwth-aachen.de/deutsch/lehre-lehrveranstaltungen/guk/index.php

Un problema di geodesia

Sia dato, come nella figura, un triangolo con base di lunghezza nota c e in cui anche gli angoli α e β siano noti e tali che $0 < \alpha, \beta < 90^\circ$.

Vogliamo calcolare x ed y .



Per le nostre ipotesi $\tan \alpha$ e $\tan \beta$ sono numeri ben definiti e > 0 (cfr. pag. 13). Inoltre abbiamo

$$\tan \alpha = \frac{y}{x}$$

$$\tan \beta = \frac{y}{c-x}$$

Queste equazioni possono essere riscritte come sistema lineare di due equazioni in due incognite:

$$x \tan \alpha - y = 0$$

$$x \tan \beta + y = c \tan \beta$$

Il determinante $\begin{vmatrix} \tan \alpha & -1 \\ \tan \beta & 1 \end{vmatrix}$ di questo sistema è uguale a

$$\tan \alpha + \tan \beta$$

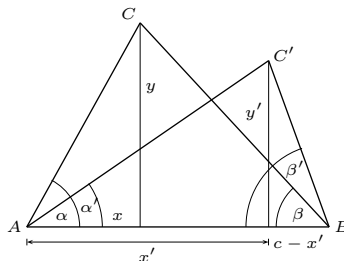
e quindi > 0 . Possiamo perciò applicare la regola di Cramer e otteniamo

$$x = \frac{\begin{vmatrix} 0 & -1 \\ c \tan \beta & 1 \end{vmatrix}}{\tan \alpha + \tan \beta} = \frac{c \tan \beta}{\tan \alpha + \tan \beta}$$

mentre per y possiamo, se calcoliamo prima x , usare direttamente la relazione $y = x \tan \alpha$.

Esercizio: Prendendo il centimetro come unità di misura e con l'uso di un goniometro verificare le formule con le distanze nella figura.

Con questo metodo possiamo adesso risolvere un compito elementare ma frequente di geodesia illustrato dalla figura seguente.



Assumiamo di conoscere la distanza tra i punti A e B , e mediante un teodolite, di essere in grado di misurare gli angoli α , β , α' e β' . Vorremmo conoscere la distanza tra i punti C e C' , ai quali però non possiamo accedere direttamente, ad esempio perché da essi ci separa un fiume che non riusciamo ad attraversare o perché si trovano in mezzo a una palude. Se le distanze sono molto grandi (maggiore di 50 km), dovremo appellarci alla trigonometria sferica, per distanze sufficientemente piccole invece possiamo utilizzare la tecnica vista sopra che ci permette di calcolare x , y , x' e y' , da cui la distanza tra C e C' si ottiene come

$$|C - C'| = \sqrt{(x - x')^2 + (y - y')^2}$$

Calcoliamo l'errore $d - c$ che si commette approssimando la distanza d sulla sfera terrestre tra due punti mediante la lunghezza c del segmento di retta che si ottiene utilizzando la trigonometria piana:

c	$d - c$
50 km	0.13 m
100 km	1 m
500 km	128 m
1000 km	1029 m

In questo numero

- 11 Trigonometria oggi
Un problema di geodesia
Grafica al calcolatore e geometria
- 12 Il triangolo
Il triangolo rettangolo
Triple pitagoroe
Le funzioni trigonometriche
- 13 La dimostrazione indiana
Il triangolo isolatero
Angoli sul cerchio
- 14 Il teorema del coseno
Il grafico della funzione seno
La periodicità di seno e coseno
Altre proprietà di seno e coseno
arcsin, arccos e arctan
- 15 Distanze in \mathbb{R}^n
Il prodotto scalare
Ortogonalità
Esercizi 14-17

Grafica al calcolatore e geometria

La grafica al calcolatore e le discipline affini come la geometria computazionale e l'elaborazione delle immagini si basano sulla matematica. È importante separare gli algoritmi dalla loro realizzazione mediante un linguaggio di programmazione. È importante separare la rappresentazione matematica delle figure nello spazio dalle immagini che creiamo sullo schermo di un calcolatore.

Il matematico è molto avvantaggiato in questo. Già semplici nozioni di trigonometria e di geometria affine e algebra lineare possono rendere facili o immediate costruzioni e formule di trasformazione (e quindi gli algoritmi che da esse derivano) che senza questi strumenti matematici risulterebbero difficoltose o non verrebbero nemmeno scoperte.

La geometria proiettiva, apparentemente una vecchia teoria astratta e filosofica, diventa di sorpresa una tecnica molto utile per trasformare compiti di proiezione in semplici calcoli.

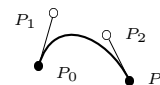
I concetti dell'analisi e della geometria differenziale portano all'introduzione e allo studio delle curve e superfici di Bézier, largamente utilizzate nei programmi di disegno al calcolatore (CAD, computer aided design).

Molte figure possono essere descritte mediante equazioni algebriche; per questa ragione la geometria algebrica assume notevole importanza nella grafica al calcolatore moderna. Curve e superficie possono essere date in forme parametriche oppure mediante un sistema di equazioni; le basi di Gröbner forniscono uno strumento per passare da una rappresentazione all'altra.

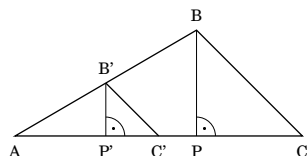
La topologia generale, una disciplina tra la geometria, l'analisi e l'algebra, è la base della morfologia matematica, mentre la topologia algebrica e la geometria algebrica reale possiedono applicazioni naturali in robotica.

H. Pottmann/J. Wallner: Computational line geometry. Springer 1999.

W. Böhm/H. Prautzsch: Geometric concepts for geometric design. Peters 1994.



Il triangolo



In questa figura i segmenti BC e $B'C'$ sono paralleli. Nella geometria elementare si dimostra che le *proporzioni* del triangolo più piccolo $AB'C'$ sono uguali alle proporzioni del triangolo grande ABC . Ciò significa che, se \overline{AB} denota la lunghezza del segmento AB , allora

$$\frac{\overline{AB'}}{\overline{AB}} = \frac{\overline{AC'}}{\overline{AC}} = \frac{\overline{B'C'}}{\overline{BC}}$$

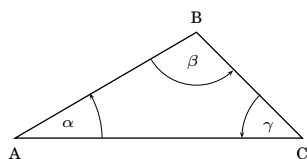
Se il valore comune di queste tre frazioni viene denotato con λ , abbiamo quindi

$$\begin{aligned}\overline{AB'} &= \lambda \cdot \overline{AB} \\ \overline{AC'} &= \lambda \cdot \overline{AC} \\ \overline{B'C'} &= \lambda \cdot \overline{BC}\end{aligned}$$

Una relazione analoga vale anche per le altezze:

$$\overline{B'P'} = \lambda \cdot \overline{BP}$$

Dati tre punti A, B, C denotiamo con $\angle(AC, AB)$ l'angolo α tra i segmenti AC e AB :



Evidentemente $0 < \alpha < 180^\circ$.

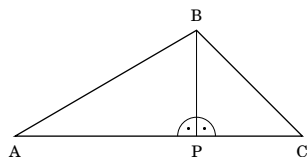
Con β e γ indichiamo gli altri due angoli come nella figura; spesso serve solo la grandezza assoluta degli angoli, allora si lasciano via le punte di freccia.

Nella prima figura il triangolo piccolo e il triangolo grande hanno gli stessi angoli, cioè

$$\begin{aligned}\angle(AC, AB) &= \angle(AC', AB') \\ \angle(BA, BC) &= \angle(B'A, B'C') \\ \angle(CB, CA) &= \angle(C'B', C'A)\end{aligned}$$

Si può dimostrare ed è chiaro intuitivamente che, dati due triangoli con gli stessi angoli, essi possono essere sovrapposti in maniera tale che si ottenga una figura simile alla nostra.

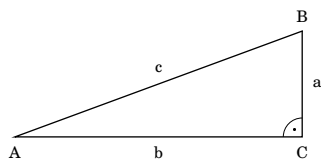
Ogni triangolo può essere considerato (talvolta anche in più modi - quando?) come unione di due triangoli rettangoli.



Le formule per i triangoli rettangoli sono particolarmente semplici; conviene quindi studiare separatamente i triangoli APB e BPC .

Il triangolo rettangolo

Il triangolo ABC sia rettangolo, ad esempio $\angle(CA, CB) = 90^\circ$.



Il lato più lungo è quello opposto all'angolo retto, cioè AB , e si chiama *ipotenusa*, i due altri lati sono più brevi e sono detti *cateti*.

La somma dei tre angoli α, β, γ di un triangolo è sempre uguale a 180° :

$$\alpha + \beta + \gamma = 180^\circ.$$

Ciò implica che un triangolo può avere al massimo un angolo retto (se ce ne fossero due, il terzo dovrebbe essere zero e non avremmo più un triangolo).

Teorema di Pitagora: Dato un triangolo rettangolo e posto $a := \overline{BC}$, $b := \overline{AC}$ e $c := \overline{AB}$ come nella figura, si ha

$$a^2 + b^2 = c^2.$$

Dimostrazione: Pag. 13.

Il teorema di Pitagora implica che l'ipotenusa è veramente più lunga di ciascuno dei due cateti (perché $a, b > 0$). La relazione $c^2 = a^2 + b^2$ può essere anche usata per il calcolo di uno dei lati di un triangolo rettangolo dagli altri due:

$$\begin{aligned}c &= \sqrt{a^2 + b^2} \\ a &= \sqrt{c^2 - b^2} \\ b &= \sqrt{c^2 - a^2}\end{aligned}$$

Triple pitagoree

Una tripla pitagorea è una tripla (a, b, c) di numeri naturali positivi tali che $a^2 + b^2 = c^2$. La tripla pitagorea si chiama *primitiva*, se a, b e c sono relativamente primi tra di loro. Diamo una tavola delle prime triple pitagoree primitive in ordine crescente di c .

3	4	5
5	12	13
8	15	17
7	24	25
20	21	29
12	35	37
9	40	41
28	45	53
11	60	61
33	56	65
16	63	65
48	55	73

Gli arabi possedevano già nel 972 tavole simili a questa.

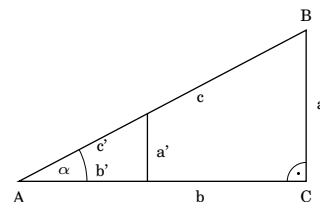
Per $n > 2$ non esistono invece soluzioni dell'equazione

$$a^n + b^n = c^n$$

con a, b, c interi > 0 . La dimostrazione di questo teorema (detto *ultimo teorema di Fermat*) è stata molto difficile; per circa tre secoli i matematici l'avevano cercata invano e solo intorno al 1995 Andrew Wiles ci è riuscito, utilizzando strumenti molto avanzati della geometria algebrica.

Le funzioni trigonometriche

Consideriamo la seguente figura,



in cui a, b, c sono come prima i lati del triangolo rettangolo più grande e a', b' e c' sono i lati del triangolo più piccolo, che è ancora rettangolo. Le proporzioni nella figura dipendono solo dall'angolo α , si ha cioè

$$\frac{c'}{c} = \frac{b'}{b} = \frac{a'}{a},$$

e da ciò anche

$$\begin{aligned}\frac{a'}{c'} &= \frac{a}{c} \\ \frac{b'}{c'} &= \frac{b}{c} \\ \frac{a'}{b'} &= \frac{a}{b}\end{aligned}$$

Questi rapporti sono perciò funzioni dell'angolo α che vengono dette *funzioni trigonometriche* e denotate come segue:

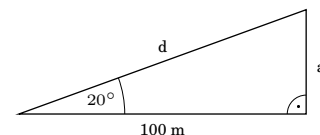
$$\begin{aligned}\sin \alpha &:= \frac{a}{c} \dots \text{ seno di } \alpha \\ \cos \alpha &:= \frac{b}{c} \dots \text{ coseno di } \alpha \\ \tan \alpha &:= \frac{a}{b} \dots \text{ tangente di } \alpha \\ \cot \alpha &:= \frac{b}{a} \dots \text{ cotangente di } \alpha\end{aligned}$$

Dalle definizioni seguono le relazioni

$$\begin{aligned}a &= c \sin \alpha = b \tan \alpha \\ b &= c \cos \alpha = a \cot \alpha \\ c &= \frac{a}{\sin \alpha} = \frac{b}{\cos \alpha}\end{aligned}$$

Esercizio. Calcolare $\sin 45^\circ$, $\cos 45^\circ$, $\tan 45^\circ$, $\cot 45^\circ$.

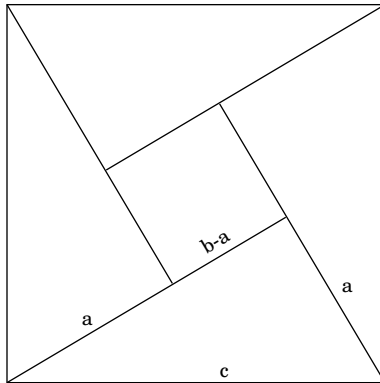
Esercizio. I valori delle funzioni trigonometriche si trovano in tabelle oppure possono essere calcolati con la calcolatrice tascabile oppure con una semplice istruzione in quasi tutti i linguaggi di programmazione. Ricavare in uno di questi modi i necessari valori per calcolare la distanza d e l'altezza a nella seguente figura:



Pierre de Fermat (circa 1607-1665) sostenne di conoscere una dimostrazione del teorema che poi portò il suo nome, ma non è mai stata trovata e si dubita molto che sia esistita.

La dimostrazione indiana

In una fonte indiana del dodicesimo secolo si trova il seguente disegno, con una sola parola in sanscrito: *guarda!*



Da esso si deduce immediatamente il teorema di Pitagora:

Il nostro triangolo rettangolo abbia i lati a, b, c con $a < b < c$. Allora l'area del quadrato grande è uguale a quella del quadrato piccolo più quattro volte l'area del triangolo, quindi

$$c^2 = (b-a)^2 + 4 \frac{ab}{2},$$

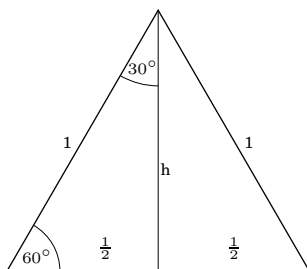
cioè

$$c^2 = b^2 - 2ab + a^2 + 2ab = b^2 + a^2.$$

Esercizio: Disegnare la figura nel caso che $a = b$ e convincersi che la dimostrazione rimane ancora valida.

Il triangolo isoscele

Consideriamo adesso un triangolo isoscele di lato 1. In esso anche gli angoli devono essere tutti uguali, quindi, dovendo essere la somma degli angoli 180° , ogni angolo è uguale a 60° .



Dalla figura otteniamo

$$h = \sqrt{1 - \frac{1}{4}} = \frac{\sqrt{3}}{2}$$

$$\sin 60^\circ = \cos 30^\circ = \frac{\sqrt{3}}{2}$$

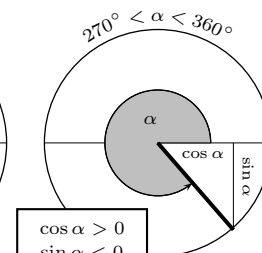
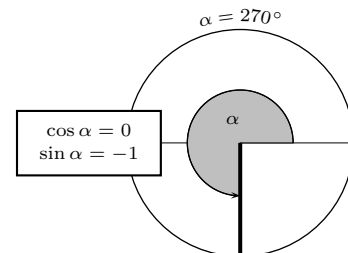
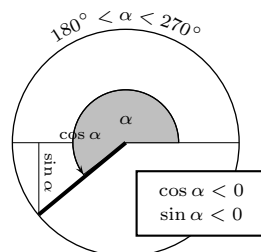
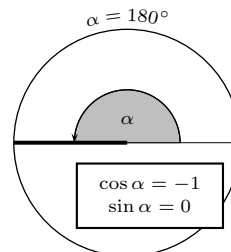
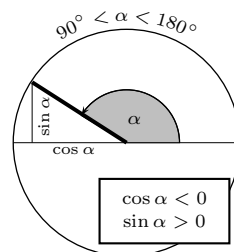
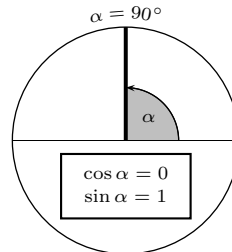
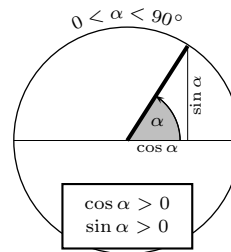
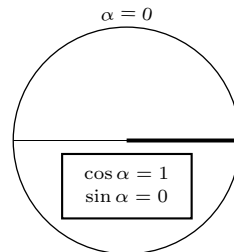
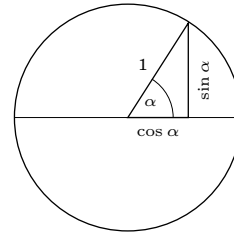
$$\sin 30^\circ = \cos 60^\circ = \frac{1}{2}$$

$$\tan 60^\circ = 2h = \sqrt{3}$$

$$\tan 30^\circ = \frac{1}{2h} = \frac{\sqrt{3}}{3}$$

Angoli sul cerchio

Siccome le lunghezze assolute non sono importanti, possiamo assumere che l'ipotenusa del triangolo rettangolo considerato sia di lunghezza 1 e studiare le funzioni trigonometriche sulla circonferenza di raggio 1. Questo ci permette inoltre di estendere la definizione delle funzioni trigonometriche a valori arbitrari di α , non necessariamente sottoposti, come finora, alla condizione $0 < \alpha < 90^\circ$. Definiamo prima $\sin \alpha$ e $\cos \alpha$ per ogni α con $0 \leq \alpha \leq 360^\circ$ come nelle seguenti figure:



Definiamo poi ogni volta

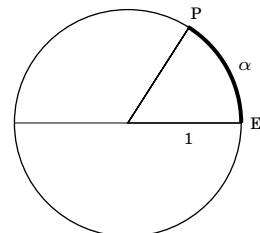
$$\tan \alpha := \frac{\sin \alpha}{\cos \alpha} \quad \cot \alpha := \frac{\cos \alpha}{\sin \alpha}$$

quando $\cos \alpha \neq 0$ risp. $\sin \alpha \neq 0$. Si vede subito che questa definizione coincide con quella data a pag. 12, quando $0 < \alpha < 90^\circ$.

Quindi $\tan \alpha = \frac{1}{\cot \alpha}$ quando entrambi i valori sono definiti.

Se α è infine un numero reale qualsiasi (non necessariamente compreso tra 0 e 360°), esiste sempre un numero intero n tale che $\alpha = n \cdot 360^\circ + \alpha_0$ con $0 \leq \alpha_0 < 360^\circ$ e possiamo definire $\cos \alpha := \cos \alpha_0$, $\sin \alpha := \sin \alpha_0$, $\tan \alpha := \tan \alpha_0$, $\cot \alpha := \cot \alpha_0$.

In matematica si identifica l'angolo con la lunghezza dell'arco descritto sulla circonferenza tra i punti E e P della figura a lato, aggiungendo però multipli del perimetro della circonferenza se l'angolo è immaginato ottenuto dopo essere girato più volte attorno al centro. Se il centro del cerchio è l'origine $(0, 0)$ del piano, possiamo assumere che $E = (1, 0)$. Siccome il perimetro della circonferenza di raggio 1 è 2π , si ha $360^\circ = 2\pi$.



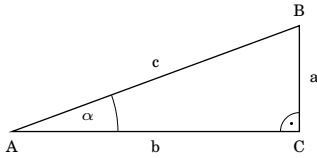
È chiaro che un angolo di g° è uguale a $\frac{g}{360} 2\pi$,

in altre parole $g^\circ = \frac{2\pi g}{360}$, e viceversa $\alpha = \frac{360^\circ}{2\pi} \alpha$ per ogni $\alpha \in \mathbb{R}$.

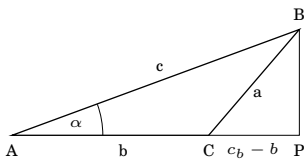
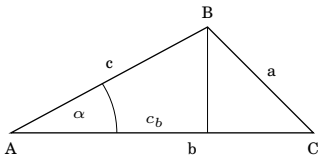
Infatti $1 = \frac{360^\circ}{2\pi} \sim 57.29577951^\circ$.

Il teorema del coseno

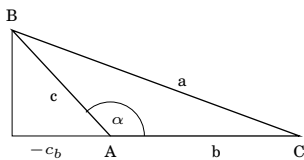
Dato un triangolo con i vertici A, B, C poniamo ancora $a := BC$, $b := AC$ e $c := AB$. Denotiamo inoltre con c_b la lunghezza della proiezione di AB su AC misurando a partire da A . In modo analogo sono definite le grandezze c_a, b_a ecc. Se l'angolo α è ottuso, c_b sarà negativo. Sono possibili quattro situazioni:



In questo caso $c_b = b$.



Si osservi che qui c_b è la lunghezza di tutto il segmento AP .



Teorema: In tutti i casi, quindi in ogni triangolo, vale la relazione

$$a^2 = b^2 + c^2 - 2bc_b.$$

Per simmetria vale anche

$$c^2 = a^2 + b^2 - 2ab_a.$$

Dimostrazione: Quando $c_b = b$, la formula diventa $a^2 = c^2 - b^2$ e segue direttamente dal teorema di Pitagora.

Nei rimanenti tre casi calcoliamo l'altezza del triangolo con il teorema di Pitagora in due modi.

Nella seconda figura abbiamo

$$c^2 - c_b^2 = a^2 - (b - c_b)^2,$$

cioè

$$c^2 - c_b^2 = a^2 - b^2 + 2bc_b - c_b^2,$$

per cui

$$c^2 = a^2 - b^2 + 2bc_b.$$

Similmente nella terza figura

$$c^2 - c_b^2 = a^2 - (c_b - b)^2,$$

la stessa equazione di prima.

Nella quarta figura infine abbiamo

$$c^2 - (-c_b)^2 = a^2 - (b - c_b)^2,$$

che è ancora la stessa equazione.

Teorema di Pitagora inverso: Un triangolo è rettangolo con l'ipotenusa c se e solo se

$$c^2 = a^2 + b^2.$$

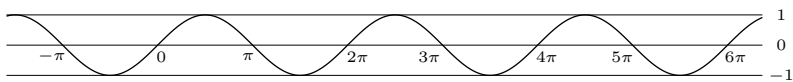
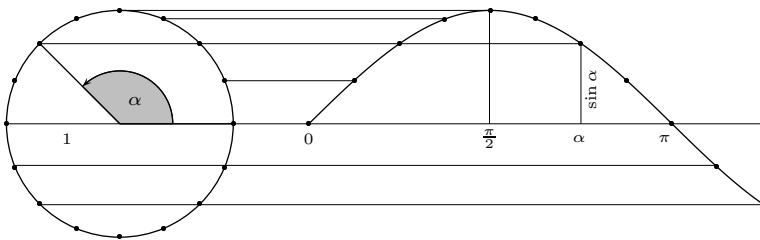
Dimostrazione: Dalla figura in alto a destra a pag. 12 si vede che il triangolo è rettangolo con ipotenusa c se e solo se $b_a = 0$ (oppure, equivalentemente, $a_b = 0$). L'enunciato segue dal teorema precedente.

Teorema del coseno:

$$a^2 = b^2 + c^2 - 2bc \cos \alpha$$

Dimostrazione: $c_b = c \cos \alpha$ in tutti e quattro i casi del precedente teorema (cfr. le definizioni degli angoli sul cerchio a pag. 13).

Il grafico della funzione seno



Come si vede dalla figura e come sarà dimostrato rigorosamente nel corso di Analisi, la funzione seno è iniettiva sull'intervallo chiuso $[-\frac{\pi}{2}, \frac{\pi}{2}]$ e assume su questo intervallo tutti i valori possibili per il seno, cioè tutti i valori tra -1 e 1. Possiamo quindi definire una funzione biettiva $\sin \alpha : [-\frac{\pi}{2}, \frac{\pi}{2}] \rightarrow [-1, 1]$. L'inversa di questa funzione viene denotata con \arcsin . In modo analogo si definiscono l'inversa arccos della funzione biettiva $\cos \alpha : [0, \pi] \rightarrow [-1, 1]$ e l'inversa arctan della funzione biettiva $\tan \alpha : (-\frac{\pi}{2}, \frac{\pi}{2}) \rightarrow (-\infty, \infty)$.

La periodicità di seno e coseno

Dalle definizioni date a pag. 13 segue che

$$\begin{aligned} \cos(\alpha + 360^\circ) &= \cos \alpha \\ \sin(\alpha + 360^\circ) &= \sin \alpha \end{aligned}$$

per ogni numero reale α . Invece di 360° possiamo anche scrivere 2π , quindi

$$\begin{aligned} \cos(\alpha + 2\pi) &= \cos \alpha \\ \sin(\alpha + 2\pi) &= \sin \alpha \end{aligned}$$

per ogni numero reale α . Le funzioni \sin e \cos sono quindi funzioni periodiche con periodo 2π .

Facendo percorrere α l'asse reale e riportando $\sin \alpha$ come ordinata, otteniamo il grafico della funzione seno rappresentato in basso a sinistra.

Altre proprietà di seno e coseno

$$\begin{aligned} \cos(-\alpha) &= \cos \alpha \\ \sin(-\alpha) &= -\sin \alpha \end{aligned}$$

per ogni numero reale α , come si vede dai disegni a pagina 13. Il coseno è quindi una funzione pari, il seno una funzione dispari.

Teorema di addizione:

$$\begin{aligned} \sin(\alpha + \beta) &= \sin \alpha \cos \beta + \cos \alpha \sin \beta \\ \cos(\alpha + \beta) &= \cos \alpha \cos \beta - \sin \alpha \sin \beta \end{aligned}$$

Dimostrazione: Non richiesta. Una dimostrazione geometrica si trova nei libri scolastici, una dimostrazione analitica forse verrà data nei corsi di Analisi.

Esercizio: $\cos \alpha = \sin(\alpha + \frac{\pi}{2})$.

Verificare l'enunciato prima nelle illustrazioni a pag. 13 e utilizzare poi il teorema di addizione per la dimostrazione.

Esercizio: Calcolare $\sin 2\alpha$ e $\cos 2\alpha$.

Teorema: $\sin^2 \alpha + \cos^2 \alpha = 1$.

Ciò segue direttamente dalle definizioni geometriche. Mentre queste proprietà algebriche delle funzioni trigonometriche rimangono valide anche per un argomento α complesso, ciò non è più vero per le disuguaglianze $|\sin \alpha| \leq 1$ e $|\cos \alpha| \leq 1$. Infatti, se dall'analisi complessa anticipiamo le formule

$$\begin{aligned} \cos z &= \frac{e^{iz} + e^{-iz}}{2} \\ \sin z &= \frac{e^{iz} - e^{-iz}}{2i} \end{aligned}$$

valide per ogni numero complesso z , vediamo che ad esempio $\cos ai = \frac{e^{-a} + e^a}{2}$, quindi per a reale e tendente ad infinito (in questo caso e^{-a} tende a 0) $\cos ai$ si comporta come $\frac{e^a}{2}$ e tende quindi fortemente ad infinito.

arcsin, arccos e arctan

Queste funzioni, definite a sinistra, sono determinate dalle seguenti relazioni:

$$\arcsin x = \alpha \iff \sin \alpha = x$$

$$\text{per } -1 \leq x \leq 1 \text{ e } -\frac{\pi}{2} \leq \alpha \leq \frac{\pi}{2}$$

$$\arccos x = \alpha \iff \cos \alpha = x$$

$$\text{per } -1 \leq x \leq 1 \text{ e } 0 \leq \alpha \leq \pi$$

$$\arctan x = \alpha \iff \tan \alpha = x$$

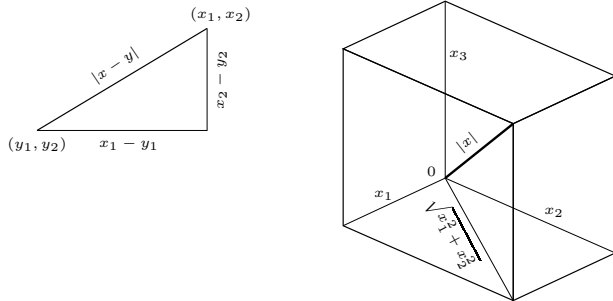
$$\text{per } -\infty < x < \infty \text{ e } -\frac{\pi}{2} < \alpha < \frac{\pi}{2}$$

Distanze in \mathbb{R}^n

La distanza tra due punti $x = (x_1, x_2)$ e $y = (y_1, y_2)$ del piano reale \mathbb{R}^2 si calcola secondo il teorema di Pitagora come

$$|x - y| = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}.$$

La distanza del punto x dall'origine è quindi $|x| = \sqrt{x_1^2 + x_2^2}$ e viceversa la distanza di x e y è proprio la lunghezza del vettore $x - y$.



Formule del tutto analoghe si hanno nello spazio tridimensionale \mathbb{R}^3 . Calcoliamo prima la lunghezza $|x|$ di un vettore $x = (x_1, x_2, x_3)$ utilizzando la figura a destra, dalla quale si vede che

$$|x|^2 = (\sqrt{x_1^2 + x_2^2})^2 + x_3^2 = x_1^2 + x_2^2 + x_3^2,$$

per cui

$$|x| = \sqrt{x_1^2 + x_2^2 + x_3^2}.$$

Se adesso $y = (y_1, y_2, y_3)$ è un altro punto, la distanza tra x e y sarà uguale alla lunghezza di $x - y$, quindi

$$|x - y| = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2}.$$

Per ogni $n \geq 1$ possiamo definire lunghezze e distanze in \mathbb{R}^n nello stesso modo. Per $x = (x_1, \dots, x_n) \in \mathbb{R}^n$ poniamo

$$|x| := \sqrt{x_1^2 + \dots + x_n^2},$$

e se $y = (y_1, \dots, y_n)$ è un altro punto, la distanza tra x e y è la lunghezza di $x - y$, cioè

$$|x - y| = \sqrt{(x_1 - y_1)^2 + \dots + (x_n - y_n)^2}.$$

Il prodotto scalare

Siano come sopra $x = (x_1, \dots, x_n)$ e $y = (y_1, \dots, y_n)$ due punti di \mathbb{R}^n . Calcoliamo la lunghezza $|x + y|$ della somma dei due vettori; questo è anche in statistica il punto di partenza per la definizione del *coefficiente di correlazione* che, nonostante il nome prometta molto di più, non è altro che un mezzo per confrontare x, y e $x + y$!

$$\begin{aligned} |x + y|^2 &= \sum_{k=1}^n (x_k + y_k)^2 = \sum_{k=1}^n x_k^2 + \sum_{k=1}^n y_k^2 + 2 \sum_{k=1}^n x_k y_k \\ &= |x|^2 + |y|^2 + 2 \sum_{k=1}^n x_k y_k \end{aligned}$$

L'espressione $\sum_{k=1}^n x_k y_k$ si chiama il *prodotto scalare* dei vettori x ed y . Esso è di fondamentale importanza per tutta la geometria. Introduciamo le abbreviazioni

$$\langle x, y \rangle := (x, y) := x_1 y_1 + \dots + x_n y_n$$

La seconda è più diffusa della prima, comporta però il pericolo di confusione con la coppia (x, y) che proprio nella statistica multidimensionale appare spesso contemporaneamente.

Sostituendo y con $-y$ otteniamo

$$|x - y|^2 = |x|^2 + |y|^2 - 2\langle x, y \rangle.$$

I due punti x ed y formano insieme all'origine 0 un triangolo (eventualmente degenerato) i cui lati hanno le lunghezze $|x|$, $|y|$ e $|x - y|$. Assumiamo che il triangolo non sia degenerato e sia α l'angolo opposto al lato di lunghezza $|x - y|$. Per il teorema del coseno abbiamo

$$|x - y|^2 = |x|^2 + |y|^2 - 2|x||y| \cos \alpha, \text{ da cui } \langle x, y \rangle = |x||y| \cos \alpha.$$

Ortogonalità

La formula fondamentale

$$\langle x, y \rangle = |x||y| \cos \alpha$$

rimane valida anche se x e y sono uno un multiplo dell'altro, ad esempio $y = tx$ per $t \in \mathbb{R}$, però entrambi $\neq 0$ (ciò implica $t \neq 0$). In questo caso infatti il triangolo determinato da x, y e 0 è degenerato, ma è naturale assegnare all'angolo tra x e y il valore 0 (per cui $\cos \alpha = 1$) se $t > 0$ e invece il valore 180° (cosicché $\cos \alpha = -1$) se $t < 0$.

Inoltre $\langle x, y \rangle = \langle x, tx \rangle = t \langle x, x \rangle = t|x|^2$ e $|x||y| = |x||tx| = |t||x||x| = |t||x|^2$. Dimostrare queste relazioni e concludere da soli, stando attenti ai segni.

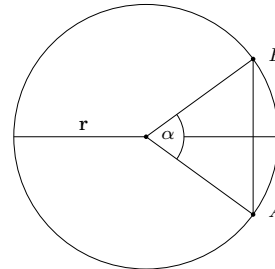
Quindi se i due vettori sono diversi da zero (ciò implica che anche $|x| \neq 0$ e $|y| \neq 0$), allora essi sono ortogonali (cioè $\alpha = 90^\circ$ oppure $\alpha = 270^\circ$) se e solo se $\cos \alpha = 0$, cioè se e solo se $\langle x, y \rangle = 0$.

Siccome infine $\langle x, 0 \rangle = 0$ per ogni x , è naturale includere anche il vettore 0 tra i vettori ortogonali ad x . Raccogliendo tutto possiamo perciò dire:

Due vettori x ed y di \mathbb{R}^n sono ortogonali se e solo se $\langle x, y \rangle = 0$.

Esercizi per gli scritti

14. Dalla figura si vede che la lunghezza c della corda (cioè del segmento di retta) tra A e B è uguale a $2r \sin \frac{\alpha}{2}$, mentre la distanza (cioè l'arco) d sul cerchio tra A e B (nell'ipotesi $0 \leq \alpha < \pi$) è uguale ad $r\alpha$.



La funzione cordale $\frac{c}{r} = 2 \sin \frac{\alpha}{2}$ è probabilmente la più antica funzione trigonometrica e venne tabulata già da Ipparco da Nikaia (Nicea) nel secondo secolo prima di Cristo (*tavola delle corde*). Già i babilonesi possedevano però una rudimentale trigonometria, di cui erano molto orgogliosi.

Calcolare la differenza $d - c$, che corrisponde all'errore che si commette usando c al posto di d per misurare la distanza tra i punti A e B sulla sfera terrestre, che possiede un raggio medio r di 6371 km, per $c = 1$ km, 10 km, 50 km, 100 km, 500 km, 1000 km.

Attenzione: Se $\sin \beta = u$, come si calcola β ?

$$\mathbf{15.} \quad \tan(x + y) = \frac{\tan x + \tan y}{1 - \tan x \tan y}.$$

$$\mathbf{16.} \quad \cos 5x = 16 \cos^5 x - 20 \cos^3 x + 5 \cos x.$$

17. Quanto è lunga la diagonale di un cubo unitario (cioè di lato uguale a 1) m -dimensionale, cioè qual'è la distanza d_m del vettore $(1, \dots, 1)$ m -dimensionale dall'origine?

Il raggio della sfera m -dimensionale inscritta in un tale cubo è invece sempre uguale a $\frac{1}{2}$, quindi la sfera inscritta dista di $d_m - \frac{1}{2}$ dai vertici del cubo e, siccome questa espressione diventa sempre più grande, il cubo m -dimensionale al crescere di m assomiglia sempre di più a un riccio con corpo sferico e aculei sempre più lunghi.

R

R è un linguaggio di programmazione ad altissimo livello orientato soprattutto all'uso in statistica. In verità lo sbilanciamento verso la statistica non deriva dalla natura del linguaggio, ma dalla disponibilità di grandi raccolte di funzioni statistiche e dagli interessi dei ricercatori che lo hanno inventato e lo mantengono. R è gratuito e molto simile a un linguaggio commerciale, S, creato negli anni '80 e anch'esso molto usato. S viene commercializzato come sistema S-Plus. Le differenze non sono grandissime se non sul piano della programmazione, dove R aderisce a una impostazione probabilmente più maneggevole.

R ed S-Plus sono particolarmente popolari nella statistica medica, ma vengono anche usati nella statistica economica o sociale, in geografia, nella matematica finanziaria. L'alto livello del linguaggio permette di creare facilmente librerie di funzioni per nuove applicazioni. Il punto debole è la velocità di esecuzione in calcoli numerici in grandi dimensioni, mentre sono ricchissime le capacità grafiche.

Benché così indirizzato verso la statistica, R non deve essere considerato un pacchetto di statistica. È un vero linguaggio di programmazione, anzi un linguaggio di programmazione molto avanzato, e ciò permette di adattarlo ad ogni compito informatico. Nella stessa statistica questa flessibilità è molto importante proprio oggi, dove continuamente si scoprono nuovi bisogni applicativi, nuove necessità di tradurre metodi matematici, ad esempio nella statistica di complessi dati clinici o geografici, in strumenti informatici.

Le funzioni d'aiuto

R dispone di numerose funzioni d'aiuto. Da un lato ci sono varie guide disponibili sul sito, dall'altra parte si possono invocare gli aiuti mentre si sta lavorando con il programma. Con

```
help.start()
```

appare una pagina web (mantenuta sul vostro PC) attraverso la quale si accede a manuali e informazioni generali. Cliccando sulla voce *Packages* si trovano elenchi dei molti pacchetti di base o aggiuntivi disponibili.

Dopo `help.start()` le informazioni di aiuto appaiono sullo schermo del browser; per disattivare questa modalità si può usare `options(htmlhelp=FALSE)` dal terminale. Infatti spesso si lavora più velocemente rimanendo sul terminale. Ci sono diversi modi per ottenere le informazioni d'aiuto. Il modo più semplice, ma molto efficiente è di anteporre un punto interrogativo al comando su cui si desidera sapere di più; con

```
?help.start
```

vengono fornite i dettagli sull'utilizzo della `help.start`. R distingue tra il nome `f` di una funzione e la sua invocazione con `f()`; naturalmente la parentesi può anche contenere argomenti. Proprio su questi argomenti si consulerà spesso l'aiuto in linea. Leggendo più attentamente il testo della guida si osserva che le funzioni di R hanno spesso argomenti opzionali determinati dal loro nome; ciò è tipico dei linguaggi in qualche modo derivati dal Lisp e verrà ancora trattato con più dettaglio.

Per uscire da un file informativo chiamato con `?` basta premere il tasto `q`. Da

R si esce con `q()` o, equivalentemente, con `quit()`. Per saperne di più si può usare il comando `?q`; le informazioni che ci vengono fornite a questo punto sono più complicate del necessario, come invero accade spesso, d'altra parte il sistema di aiuto in linea di R è veramente molto completo anche se non perfettamente organizzato, perché richiede che si sappia già come si chiamano i comandi e perché i comandi non sono raggruppati bene secondo le funzionalità. Per sapere di più su `help` usare `?help`.

Esiste comunque un'altra funzione che permette di cercare informazioni su comandi di cui non si conosce il nome o su gruppi di comandi. Questa funzione è `help.search`; per capire come bisogna utilizzarla battiamo `?help.search`. Assumiamo adesso che cerchiamo le funzioni trigonometriche. Proviamo prima con

```
help.search("trigo")
```

trovando una breve pagina d'aiuto che ci rimanda al pacchetto `Trig`. Se adesso battiamo `?Trig`, troviamo l'elenco delle funzioni disponibili

```
cos(x)
sin(x)
tan(x)
acos(x)
asin(x)
atan(x)
atan2(y,x)
```

con l'indicazione degli argomenti, seguito da un'esposizione sull'uso.

In questo numero

- 16 R
 - Le funzioni d'aiuto
 - Installazione di R
 - Il libro di Crawley
- 17 Programmare in R
 - Programmi autonomi
 - Nomi in R
 - Assegnamento
- 18 Successioni
 - Angoli espressi in gradi
 - Figure di Lissajous
 - I commenti
 - La matematica del futuro
 - Esercizi 18-22

Installazione di R

Per l'uso e l'installazione di R sotto Windows possiamo dare solo informazioni molto vaghe.

Il sito a cui rivolgersi è, per ogni sistema operativo,

www.r-project.org/

dove, per copiare il programma, si sceglie *CRAN* e poi uno dei siti depositori; quello che funziona meglio è forse *cran.r-mirror.de*. Nella voce *FAQ* esiste invece una guida per Windows.

L'installazione sotto Linux è molto semplice. Si ritira il file *.rpm* adatto per la propria versione di Linux e poi lo si installa, diventando *root*, con

```
rpm -Uvh nome-del-file.rpm
```

A questo punto, tornati utenti normali, basta battere *R* dalla tastiera e il programma parte.

Sotto Windows viene consigliato di lanciare *bin\Rgui.exe* oppure di creare un *shortcut* a questo file.

Il libro di Crawley

One of the objectives of statistical analysis is to distil a long and complicated set of data into a small number of meaningful descriptive statistics. Many of the modern computer statistics packages, however, do exactly the opposite of this. They generate literally pages of output from the most meagre sets of data. This copious output has several major shortcomings: it is open to uncritical acceptance; it can lead to over interpretation of data; and it encourages the bad habit of data trawling (dredging through the output looking for significant results without any prior notion of a testable hypothesis).

S-Plus, on the other hand, tells you nothing unless you explicitly ask for it ... The computing is presented in S-Plus, but all the examples will also work in the freeware program R, which can be downloaded from the web, free of charge, anywhere in the world ... S-Plus encourages good habits of data exploration by providing a superb range of graphical facilities.

La statistica, in fondo, si sviluppa attorno a questa domanda: *What do we mean when we say that a result is significant?*

M. Crawley: Statistical computing. Wiley 2004.

© www.bio.ic.ac.uk/research/mjrcraw/crawley.htm

Programmare in R

Benché si tratti di un linguaggio ad alto livello, gli ideatori di R preferiscono presentare R come linguaggio con cui lavorare in linea e non mediante l'esecuzione di programmi scritti su files. Oggigiorno ciò non è perfettamente comprensibile ed è comunque possibile scrivere programmi e farli eseguire nel modo familiare ai programmatori con la tecnica che adesso descriviamo.

Prima creiamo una nuova cartella (*directory*) in cui vogliamo svolgere un determinato lavoro.

Poi in questa cartella creiamo il file *.Rprofile* che contiene solo queste righe:

```
alfa = function ()
{source("prove")}
```

Le istruzioni contenute in *.Rprofile* vengono, come vedremo nel prossimo numero, eseguite alla partenza di R.

Adesso creiamo un file *prove* in cui scriviamo (per il momento) tutto il resto del programma, ad esempio

```
f = function (x)
{exp(-x*x)}

cat(f(2), '\n')
```

La funzione *f* così definita corrisponde alla funzione matematica $\bigcirc_x e^{-x^2}$, mentre *cat* è la funzione elementare di visualizzazione (un po' complicata anch'essa come tutte le funzioni di input/output di R) che qui visualizza $f(x)$ per $x = 2$.

'\n' è il carattere di invio che, nell'output, fa in modo che dopo la visualizzazione il programma torni su una nuova riga.

Per eseguire il programma battiamo R dalla tastiera e poi, una volta in R, diamo il comando

```
alfa()
```

che, in accordo con la sua definizione, carica il file *prove* ed esegue le istruzioni in esso contenute. Viene visualizzato il risultato:

```
0.01831564
```

cioè e^{-4} , come possiamo verificare aggiungendo la riga

```
cat(exp(-4), '\n')
```

a *prove*. Così possiamo continuare a lavorare, rimanendo in R, ma scrivendo il programma e le sue modifiche in *prove*, usando il terminale solo per ripetere il comando *alfa()* (a questo scopo in ambiente Linux è sufficiente premere il tasto ↑ che utilizza la storia dei comandi dati in precedenza che si trova nel file nascosto *.Rhistory* nella nostra stessa cartella).

Forse non tutti conoscono il valore del numero e ; esso è uguale ad e^1 , quindi lo possiamo visualizzare aggiungendo

```
cat(exp(1), '\n')
```

al nostro programma, cioè al file *prove*.

Vediamo così che $e = 2.718282$, se calcolato sulle prime 6 cifre dopo il punto decimale, perché in verità e è un numero trascendente, cioè non è radice di un polinomio a coefficienti interi, e quindi sicuramente non è razionale.

Per scrivere il programma dobbiamo usare un editor che crea soltanto files in formato testo puro o impostare questa modalità nell'editor che vogliamo usare.

Programmi più grandi verranno scritti su più files, ad esempio *prove*, *funzioni*, *grafica*. In tal caso possiamo o modificare la funzione principale nel file *.Rprofile*,

```
alfa = function ()
{source("prove")
source("funzioni")
source("grafica")}
```

Bisogna stare un po' attenti all'ordine in cui le inclusioni vengono effettuate; infatti un altro punto debole di R, che deriva dalla filosofia di voler costringere l'utente a programmare sul terminale, è che le funzioni possono essere ridefinite, possibilità che compromette notevolmente la trasparenza dei programmi se il programmatore non è abituato ad organizzare bene il proprio lavoro. Esempio:

Cancelliamo tutto il contenuto del file *prove*, scrivendo poi

```
f = function (x)
{x*x}

f = function (x)
{-x}

cat(f(3), '\n')
```

Se eseguiamo il programma con *alfa()* viene visualizzato -3. Ciò è comodo nel lavoro sul terminale, ma piuttosto problematico quando si vogliono creare programmi più consistenti o librerie di funzioni.

Programmi autonomi

Dal punto di vista della filosofia Unix, la tecnica di programmazione che abbiamo usato nell'articolo precedente, ha un neo importante. Infatti il programmatore Unix è abituato che tutti i programmi possono essere combinati tra di loro, cioè che possono essere eseguiti da un altro programma e che si possono scambiare dati. La nostra tecnica non crea però files eseguibili autonomamente perché richiede l'esecuzione dal terminale di R.

Praticamente tutti gli altri linguaggi interpretati permettono, sotto Unix, la creazione di programmi eseguibili semplicemente mettendo nella prima riga del file il comando

```
#!/interprete
```

che fa in modo che il resto del file venga eseguito dall'interprete indicato. Per un programma in R dovremmo quindi scrivere

```
#!/usr/bin/R/
```

Per ragioni misteriose questa possibilità per R non è mai stata realizzata.

Si può ovviare a questo problema scrivendo nella prima riga

```
/usr/bin/R --slave <<<
```

Benché funzioni (almeno in ambiente Unix o Linux), non è molto soddisfacente, anche perché in ogni esecuzione R viene caricato con molte delle sue librerie e quindi l'avvio è lento; i tipici programmi Unix si caricano in genere solo con quanto è necessario per poter partire.

Nomi in R

Nomi (detti anche identificatori) in R consistono in una lettera (A-Za-z) seguita da una o più caratteri che possono essere lettere, cifre o punti. Quindi

```
alfa
a35
fx.588p
```

sono nomi ammissibili, mentre non lo sono 35a, .8a6, a5+3. In R (come in C o in Perl) bisogna distinguere tra minuscole e maiuscole.

Bisogna anche evitare i nomi riservati di R. Purtroppo in R anche alcuni caratteri singoli sono riservati:

```
c
q
t
C
D
F
I
T
```

inoltre ci sono parecchi nomi che consistono di due lettere, ad esempio *df* e *pt*. Questa scelta sicuramente non è ottimale. Ad esempio non possiamo usare *t* per il tempo e nemmeno *F* per le funzioni.

Assegnamento

L'assegnamento di un valore (spesso rappresentato da un'espressione) a una variabile *x* avviene mediante l'istruzione

```
x = valore
```

Esiste anche la forma tradizionale (leggermente più generale)

```
x <- valore
```

sicuramente meno leggibile. Per saperne di più usare ?"<-".

Più istruzioni sulla stessa riga devono essere separate da un punto e virgola, mentre il punto e virgola alla fine di una riga è (a differenza dal C) facoltativo:

```
x=10; y=x*x+2; cat(y, '\n');
cat((x+y)/2, '\n');
```

con output

```
102
56
```

Successioni

Uno dei punti forti di R è che molte funzioni sono definite direttamente per successioni finite di valori. Ciò significa che se definiamo una funzione in R per la funzione matematica $\bigcirc x^2$, la possiamo im-

mediatamente applicare a una successione (x_1, \dots, x_n) per ottenere la successione (x_1^2, \dots, x_n^2) . In C allo stesso scopo bisogna lavorare con un ciclo, ad esempio un `for`, e riflettere sulla struttura di dati che si vogliono utilizzare; in Lisp e Perl si può usare la funzione `map` che, data una funzione f , trasforma la successione nella successione $(f(x_1), \dots, f(x_n))$. In R è tutto molto più semplice e automatico:

```
Quad = function (x) {x*x}
```

```
u=c(1,2,3,4,5)
v=Quad(u)
cat(v, '\n')
```

con output

```
1 4 9 16 25
```

Si osservi qui che la successione `u` è stata creata utilizzando l'operatore `c`, il cui nome viene da *concatenate* e che unisce una sequenza di valori in un unico oggetto. `?c` per dettagli.

Anche le operazioni algebriche vengono eseguite su tutti gli elementi di una successione; possiamo ad esempio moltiplicare una successione con un numero oppure anche due successioni tra di loro. Esempi:

```
u=c(1,2,3,4,5); v=c(2,1,3,5,6);
cat(u+v, '\n');
cat(u*v, '\n');
cat(u+10, '\n');
cat(10*u, '\n');
```

con output

```
3 3 6 9 11
2 2 9 20 30
11 12 13 14 15
10 20 30 40 50
```

Un operatore molto utile per generare successioni di valori equidistanti è la funzione `seq`. Il risultato di

```
seq(a,b)
```

è la successione

```
a, a+1, a+2, ...
```

che è continuata fino a quando non si supera `b`. Il passo di progressione è quindi 1 se non viene impostato come terzo argomento:

```
seq(a,b,p)
```

restituisce la successione

```
a, a+p, a+2p, ...
```

anch'essa continuata fino a quando l'ultimo valore non supera `b`. Esempi:

```
u=seq(0,5); cat(u, '\n');
u=seq(0,2,0.3); cat(u, '\n');
u=seq(3.7,9); cat(u, '\n');
u=seq(3.7,5,0.2); cat(u, '\n');
```

Otteniamo

```
0 1 2 3 4 5
0 0.3 0.6 0.9 1.2 1.5 1.8
3.7 4.7 5.7 6.7 7.7 8.7
3.7 3.9 4.1 4.3 4.5 4.7 4.9
```

Angoli espressi in gradi

Successioni possono essere valori di una funzioni. Creiamo ad esempio una funzione `Geode` per la risoluzione del problema geodetico visto in precedenza. Questa funzione restituisce una successione `u` i cui componenti `u[1]` e `u[2]` sono x e y come a pagina 11. A differenza dal C l'indice delle successioni parte da 1. Per il `c` di pagina 11 usiamo la variabile `base`, perché `c` è una parola riservata.

Siccome vogliamo poter indicare gli angoli in gradi, definiamo prima tre funzioni che corrispondono alle funzioni trigonometriche `cos`, `sin` e `tan` per angoli espressi in gradi. L'argomento va quindi moltiplicato con $\frac{\pi}{180}$ come visto a pagina 13.

```
# pi/180
constpid180 = 0.017453292519943;

# Funzioni trigonometriche per
# angoli indicati in gradi.
Cos = function (x) {cos(x*constpid180)}
Sin = function (x) {sin(x*constpid180)}
Tan = function (x) {tan(x*constpid180)}
```

```
# Problema di geodesia a pagina 11.
# Gli angoli vengono indicati in gradi.
Geode = function (base,alfa,beta)
{tanalfa=Tan(alfa); tanbeta=Tan(beta);
x=base*tanbeta/(tanalfa+tanbeta);
y=x*tanalfa; c(x,y)}
```

```
u=Geode(38,62,47); x=u[1]; y=u[2];
cat(x,y, '\n');
```

La base del triangolo era di 38 mm, gli angoli di approssimativamente 62 e 47 gradi. L'output è

```
13.79908 25.95230
```

e corrisponde abbastanza bene alle misure reali.

Useremo da ora in avanti per le funzioni sempre nomi che iniziano con una maiuscola, ricordandoci che sono riservati i nomi C, D, F, I, T.

Figure di Lissajous

Illustreremo nel prossimo numero le capacità grafiche di R. Provare intanto questo esempio che dovrebbe far apparire una figura di Lissajous, cioè una curva con una rappresentazione parametrica tipicamente della forma $\bigcirc(\cos mt, \sin nt)$ per $m, n \in \mathbb{R}$.

```
x=seq(0,2*pi,length=1000)
lato=c(-1,1)
# plot imposta la grafica.
plot(lato,lato,
type="n",xlab="",ylab="")
lines(sin(8*x),sin(7*x))
locator(n=2,type="p")
# Cliccare 2 volte.
dev.off() # Uscire dalla grafica.
```

I commenti

Se una riga contiene, al di fuori di una stringa, il carattere `#`, tutto il resto della riga è considerato un commento, compreso il carattere `#` stesso.

Molti altri linguaggi interpretati (Perl, Python, la shell di Unix) usano questo simbolo per i commenti. In C una funzione analoga è svolta dalla sequenza `//`.

La matematica del futuro

The main scientific challenges of the twenty-first century may no longer be divided into the classical disciplines of mathematics, informatics, physics, chemistry, biology, etc. For example, theoretical biology is currently in the phase of formulating laws of nature in terms of mathematical statements; quantum chemistry has already become an important research field in applied mathematics; physics needs more and more input from computer science and mathematics, including logic and informatics; and, outside of the natural sciences, financial mathematics has developed highly reliable tools for economic market and stock analysis. But how will researchers be motivated to do interdisciplinary research in a university environment, given the current system in which academic careers (typically) advance based upon a record of publication in a single field?

www.wpi.ac.at/

And the missing ingredient in facing those problems ... is mathematics.

D. Donoho: High-dimensional data analysis - the curses and blessings of dimensionality. Internet 2000, 32p.

Esercizi per gli scritti

$$18. \begin{pmatrix} 2 & 1 \\ 6 & 2 \end{pmatrix} \begin{pmatrix} 6 \\ 2 \end{pmatrix} = 14$$

$$19. \begin{pmatrix} 2 \\ 1 \end{pmatrix} \begin{pmatrix} 6 & 2 \end{pmatrix} = \begin{pmatrix} 12 & 4 \\ 6 & 2 \end{pmatrix}$$

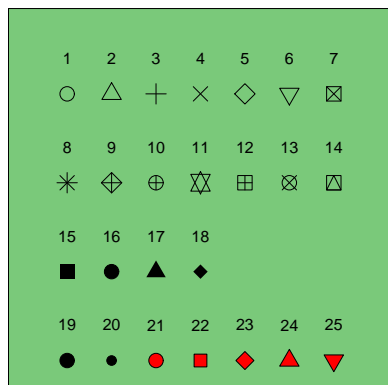
Qual'è la ragione per cui nella seconda riga del prodotto riappare il secondo fattore?

$$20. \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \begin{pmatrix} 5 & 6 & 7 \\ 8 & 9 & 10 \end{pmatrix} = \begin{pmatrix} 21 & 24 & 27 \\ 47 & 54 & 61 \end{pmatrix}$$

$$21. \begin{pmatrix} 0 & 1 & 2 \\ 5 & 2 & 1 \\ 8 & 6 & 3 \end{pmatrix} \begin{pmatrix} 2 & 5 \\ 4 & 1 \\ 6 & 2 \end{pmatrix} = \begin{pmatrix} 16 & 5 \\ 24 & 29 \\ 58 & 52 \end{pmatrix}$$

22. La traccia $\text{tr } A$ di una matrice A è la somma degli elementi nella sua diagonale principale. Una matrice 2×2 $\begin{pmatrix} a & b \\ c & d \end{pmatrix} = A$ può (in parte) essere considerata come elemento (a, b, c, d) di \mathbb{R}^4 . Possiamo quindi formare il prodotto scalare $\|A, B\|$ di due matrici 2×2 in \mathbb{R}^4 . Dimostrare che $\|A, B\| = \text{tr } AB^t$, dove B^t denota la matrice trasposta di B , cioè la matrice che si ottiene da B scambiando righe e colonne.

La grafica di R



R possiede ricche capacità grafiche, però anche molti comandi per la grafica che è difficile imparare con tutte le loro opzioni. Da un lato si può usare l'aiuto in linea (provare ?plot, ?plot.default, ?par, ?curve, ?contour, ?points), dall'altro lato però è utile una certa disciplina nell'impiegare una serie scelta e ben composta di comandi che si padroneggia. Cercheremo anche noi di fare così, ad esempio utilizzando il comando plot in genere solo per predisporre la finestra grafica su

cui successivamente le immagini verranno create con comandi separati, anche se molte figure potrebbero essere ottenute con un unico comando plot complicato.

In particolare useremo sistematicamente comandi par separati dalle istruzioni di disegno. Infatti il comando par permette una perfetta impostazione dei parametri grafici (colore, spessore delle linee, disposizione degli assi, caratteristiche dei testi che appaiono nelle figure, dimensioni varie, possibilità di linee tratteggiate in vari modi), quindi nonostante la molteplicità delle opzioni andrebbe studiato bene, ma il suo utilizzo risulta più trasparente se viene usato in modo esplicito al di fuori degli altri comandi. La figura nell'inserto è spiegata a pagina 22.

plot e lines

Diamo solo le varianti essenziali di questi comandi. In particolare useremo plot solo per predisporre la finestra grafica, non per il disegno stesso. Per far apparire la figura in una finestra sotto Linux in alcune versioni di R bisogna prima chiamare il dispositivo grafico con X11(), ma non sembra più necessario nelle versioni più recenti. Per terminare il lavoro del dispositivo grafico si usa dev.off().

plot richiede (nell'uso che ne facciamo) come primi argomenti l'indicazione dei limiti per le coordinate x ed y , entrambi nella forma $c(a,b)$. Se vogliamo disegnare il coseno, possiamo ad esempio impostare l'intervallo per x con $intx=c(-2\pi, 2\pi)$ e l'intervallo per y con $inty=c(-1, 1)$.

I comandi di base per impostare la finestra grafica sono allora i seguenti:

```
intx=c(-2*pi, 2*pi); inty=c(-1, 1)
plot(intx, inty, type="n",
     xlab="", ylab="",
     asp=1, axes=FALSE, frame.plot=TRUE)

dev.off()
```

Se eseguiamo il programma, vedremo per un istante lampeggiare la finestra grafica che però si chiude subito. Per poterla guardare, inseriamo locator(1) nella penultima riga; il parametro 1 indica che il programma aspetta che clicchiamo una volta sull'immagine, prima di chiuderla. Vediamo allora un quadrato nella finestra che è stato disegnato a causa dell'indicazione frame.plot=TRUE nel comando plot.

axes=FALSE significa che gli assi delle coordinate non vengono mostrati; asp=1 è

spesso utile per imporre che le coordinate x ed y vengano utilizzate nella stessa scala (asp=2 farebbe in modo che le coordinate y appaiono in scala doppia rispetto alle coordinate x); xlab e ylab riguardano le scritte che appaiono ai lati del diagramma; infine type="n" indica che plot viene usato solo per l'impostazione e non per disegnare una figura.

A questo punto, far apparire nella finestra il grafico del coseno, è sufficiente inserire la riga

```
lines(x, cos(x))
```

Infatti, l'operazione a cui corrisponde lines matematicamente può essere descritta in questo modo: Se sono dati due vettori $x = (x_1, \dots, x_m)$ e $y = (y_1, \dots, y_m)$, allora lines(x,y) unisce il punto (x_1, y_1) con (x_2, y_2) , il punto (x_2, y_2) con (x_3, y_3) , ..., e (x_{m-1}, y_{m-1}) con (x_m, y_m) . Quindi se f è una funzione, con lines(x,f(x)) otteniamo un'approssimazione poligonale della funzione f che, in una figura normale, per m sufficientemente grande (ad esempio maggiore di 1000) sembrerà una rappresentazione perfetta della funzione.

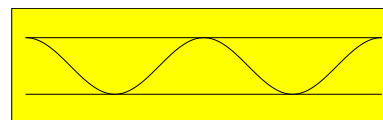
Possiamo utilizzare la stessa funzione lines per aggiungere alla figura (che vediamo nell'inserto a destra) le rette $y = 1$ e $y = -1$. Per fare ciò uniamo con una retta i punti $(-2\pi, 1)$ e $(2\pi, 1)$ e con una seconda retta i punti $(-2\pi, -1)$ e $(2\pi, -1)$:

```
lines(c(-2*pi, 2*pi), c(1, 1))
lines(c(-2*pi, 2*pi), c(-1, -1))
```

In questo numero

- 19 La grafica di R
plot e lines
Il comando postscript
- 20 I numeri binomiali
- 21 La formula di Stirling
Coordinate polari nel piano
Coordinate cilindriche nello spazio
Coordinate polari nello spazio
Rotazioni nel piano
- 22 Numeri complessi in R
Funzioni in R
points
symbols e rect
Esercizi 23-27

Il comando postscript



Per far apparire questa figura sullo schermo usiamo quindi

```
intx=c(-2*pi, 2*pi); inty=c(-1, 1)
par(mai=c(0,0,0,0))
par(lwd=0.5, bg="yellow")
plot(intx, inty, type="n", xlab="", ylab="",
     asp=1, axes=FALSE, frame.plot=TRUE)
x=seq(-2*pi, 2*pi, length=1000)
lines(x, cos(x))
lines(c(-2*pi, 2*pi), c(1, 1))
lines(c(-2*pi, 2*pi), c(-1, -1))
locator(1)
dev.off()
```

Se vogliamo invece conservare l'immagine in un file in formato *PostScript*, possiamo usare il comando postscript come nella seguente sequenza di istruzioni, dove abbiamo anche tolto l'istruzione interattiva locator(1):

```
postscript("../ps/19-coseno.ps",
           width=2.0, height=0.6,
           horizontal=FALSE, onefile=FALSE,
           paper="special");
intx=c(-2*pi, 2*pi); inty=c(-1, 1)
par(mai=c(0,0,0,0))
par(lwd=0.5, bg="yellow")
plot(intx, inty, type="n", xlab="", ylab="",
     asp=1, axes=FALSE, frame.plot=TRUE)
x=seq(-2*pi, 2*pi, length=1000)
lines(x, cos(x))
lines(c(-2*pi, 2*pi), c(1, 1))
lines(c(-2*pi, 2*pi), c(-1, -1))
dev.off()
```

Abbiamo aggiunto, nei due comandi par, le specifiche di alcuni parametri grafici: mai=c(0,0,0,0) per azzerare i margini della figura (la i viene da inches, pollici, l'unità di misura utilizzata da R), bg="yellow" per usare uno sfondo (background) giallo, e lwd=0.5 per dimezzare lo spessore delle linee (linewidth).

Anche in questo caso non bisogna dimenticare dev.off() alla fine.

I numeri binomiali

Situazione 20.1. $n \in \mathbb{N}$, X un insieme finito con n elementi.

Definizione 20.2. Denotiamo con $|X|$ il numero degli elementi di X .

In particolare $|\emptyset| = 0$.

Nota 20.3. Se A e B sono sottoinsiemi di X tali che $X = A \uplus B$ (cioè $X = A \cup B$ e $A \cap B = \emptyset$), allora

$$|X| = |A| + |B|.$$

Definizione 20.4. Per $k \in \mathbb{N}$ denotiamo con $\mathcal{P}(X, k)$ l'insieme di quei sottoinsiemi di X che possiedono esattamente k elementi:

$$\mathcal{P}(X, k) = \{A \subset X \mid |A| = k\}$$

Esempio 20.5. Sia $X = \{1, 2, 3, 4, 5\}$. Quindi $|X| = 5$. Quanti sottoinsiemi con esattamente 2 elementi possiede X ?

Questi sottoinsiemi sono $\{1, 2\}$, $\{1, 3\}$, $\{1, 4\}$, $\{1, 5\}$, $\{2, 3\}$, $\{2, 4\}$, $\{2, 5\}$, $\{3, 4\}$, $\{3, 5\}$, $\{4, 5\}$. X possiede perciò 10 sottoinsiemi con 2 elementi, in altre parole $|\mathcal{P}(X, 2)| = 10$.

Definizione 20.6. Per $k \in \mathbb{N}$ sia $\binom{n}{k}$ il numero dei sottoinsiemi di X che possiedono esattamente k elementi. Quindi

$$\binom{n}{k} = |\mathcal{P}(X, k)|$$

I numeri della forma $\binom{n}{k}$ si chiamano *coefficienti binomiali* o *numeri binomiali*. La definizione naturalmente non dipende da X , ma solo da n .

Osservazione 20.7.

$\binom{n}{0} = 1$ perché \emptyset è l'unico sottoinsieme di X con 0 elementi.

$\binom{n}{n} = 1$ perché X è l'unico sottoinsieme con n elementi.

$\binom{n}{1} = n$ perché i sottoinsiemi con un elemento corrispondono univocamente agli elementi di X , e di questi ce ne sono proprio n .

Proposizione 20.8. $\binom{n}{n-k} = \binom{n}{k}$
per $0 \leq k \leq n$.

Dimostrazione. Ogni sottoinsieme $A \subset X$ determina univocamente il suo complemento $X \setminus A$ e viceversa. Se A ha k elementi, $X \setminus A$ ne ha $n - k$. Abbiamo quindi una biiezione tra $\mathcal{P}(X, k)$ e $\mathcal{P}(X, n - k)$.

Corollario 20.9. $\binom{n}{n-1} = n$.

Osservazione 20.10. $\binom{n}{k} = 0$ per $k > n$.

Dimostrazione. X non può possedere sottoinsiemi con più di n elementi, perché si ha sempre $|A| \leq |X|$ per $A \subset X$.

Nota 20.11. e sia un elemento esterno che non appartiene ad X . Allora $X \cup \{e\}$ possiede $n + 1$ elementi. Sia $k \in \mathbb{N}$ con $1 \leq k \leq n$.

Siccome ogni sottoinsieme di X è anche un sottoinsieme di $X \cup \{e\}$, i sottoinsiemi di $X \cup \{e\}$ con k elementi sono o sottoinsiemi con k elementi di X , oppure consistono di e e di altri $k - 1$ elementi appartenenti ad X .

Vediamo così che $\mathcal{P}(X \cup \{e\}, k)$ è uguale a

$$\mathcal{P}(X, k) \uplus \{A \uplus \{e\} \mid A \in \mathcal{P}(X, k - 1)\}$$

per cui

$$\binom{n+1}{k} = \binom{n}{k} + \binom{n}{k-1}$$

per $1 \leq k \leq n$. Questa formula vale però anche per $k > n$:

Infatti per $k = n + 1$ abbiamo a sinistra $\binom{n+1}{n+1} = 1$ e anche a destra $\binom{n}{n+1} + \binom{n}{n} = 0 + 1 = 1$; per $k \geq n + 2$ abbiamo 0 sia a destra che a sinistra.

Teorema 20.12.

$$\binom{n+1}{k} = \binom{n}{k} + \binom{n}{k-1}$$

per ogni $n \geq 0$ ed ogni $k \geq 1$.

Dimostrazione. Nota 20.11.

Il teorema 20.12 è equivalente al *triangolo di Pascal*:

$$\begin{array}{cccccccc} & & & & 1 & & & \\ & & & 1 & & 1 & & \\ & & 1 & & 2 & & 1 & \\ & 1 & & 3 & & 3 & & 1 \\ 1 & & 1 & & 4 & & 6 & & 4 & & 1 \\ & 1 & & 5 & & 10 & & 10 & & 5 & & 1 \\ & 1 & & 6 & & 15 & & 20 & & 15 & & 6 & & 1 \end{array}$$

Teorema 20.13. Teorema binomiale:

$$(a + b)^n = \sum_{k=0}^n \binom{n}{k} a^{n-k} b^k$$

per ogni $a, b \in \mathbb{R}$. Questa formula è più in generale valida in ogni anello commutativo con unità.

Dimostrazione. Non richiesta, ma facile. Corsi di Geometria o Analisi.

Definizione 20.14. Poniamo

$$n! := 1 \cdot 2 \cdots (n-1)n$$

$$\text{per } n \geq 1$$

$$0! := 1$$

$n!$ si pronuncia *n fattoriali*; si parla anche del *fattoriale di n*.

Dalla definizione segue direttamente l'equazione funzionale

$$(n+1)! = n!(n+1)$$

Teorema 20.15. $\binom{n}{k} = \frac{n!}{k!(n-k)!}$

per $0 \leq k \leq n$.

Dimostrazione. Facile per induzione su n , fissato k . Non richiesta.

Nota 20.16. La formula del teorema 20.15 è molto importante nella teoria, ma, un po' simile a come accade per il determinante, può essere utilizzata solo per numeri molto piccoli.

Infatti per calcolare $\binom{200}{4}$ con il teorema 20.15 dovremmo calcolare $200!$ e $196!$, due numeri giganteschi, e formare $\frac{200!}{4!196!}$. In pratica si usa invece la regola

$$\binom{n}{k} = \frac{n(n-1) \cdots (n-k+1)}{1 \cdot 2 \cdots k}$$

dove sopra stanno tanti numeri quanti sotto, cioè k .

Ad esempio

$$\binom{12}{3} = \frac{12 \cdot 11 \cdot 10}{1 \cdot 2 \cdot 3}$$

$$\binom{20}{5} = \frac{20 \cdot 19 \cdot 18 \cdot 17 \cdot 16}{1 \cdot 2 \cdot 3 \cdot 4 \cdot 5}$$

$$\binom{200}{4} = \frac{200 \cdot 199 \cdot 198 \cdot 197}{1 \cdot 2 \cdot 3 \cdot 4}$$

Non è difficile giustificare questa regola (esercizio 23).

Definizione 20.17. I numeri

$$\binom{n}{k}^+ := \frac{n(n+1) \cdots (n+k-1)}{1 \cdot 2 \cdots k}$$

si chiamano *binomiali superiori*. Anche qui sia sopra che sotto stanno k numeri. Infatti

$$\binom{n}{k}^+ = \binom{n+k-1}{k}$$

Basta leggere il numeratore all'indietro nella definizione. Nonostante ciò, i binomiali superiori hanno interessanti interpretazioni combinatoriche.

Nota 20.18. Per calcolare fattoriali e binomiali molto grandi al calcolatore conviene utilizzare il logaritmo. Siccome

$$0! = 1 \text{ e } n! = (n-1)!n \text{ per } n \geq 1$$

abbiamo

$$\log 0! = 0 \text{ e}$$

$$\log n! = \log(n-1)! + \log n \text{ per } n \geq 1$$

da cui si deduce facilmente un algoritmo.

Quando si usa il logaritmo, si può anche usare il teorema 20.15 per calcolare i numeri binomiali:

$$\log \binom{n}{k} = \log n! - \log k! - \log(n-k)!$$

Quando questi numeri, ad esempio $n!$ per $0 \leq n \leq 1000$, servono spesso, conviene creare una tabella.

In R comunque tutte queste funzioni sono già comprese:

<code>factorial(n)</code>	$n!$
<code>lfactorial(n)</code>	$\log n!$
<code>choose(n,k)</code>	$\binom{n}{k}$
<code>lchoose(n,k)</code>	$\log \binom{n}{k}$

Sorprendentemente con `factorial(3.5)` non otteniamo, come magari ci aspetteremmo, lo stesso risultato come con `factorial(3)`, cioè 6 (perché si potrebbe pensare che semplicemente l'argomento venga arrotondato, essendo il fattoriale definito per argomenti che sono numeri naturali); viene invece restituito il misterioso valore 11.63173.

La ragione è molto profonda; infatti la funzione fattoriale $\bigcirc_n n! : \mathbb{N} \rightarrow \mathbb{N}$ è la restrizione ad \mathbb{N} della *funzione gamma*

$$\Gamma : \mathbb{C} \setminus (-\mathbb{N}) \rightarrow \mathbb{C}$$

una delle più importanti funzioni della matematica e della statistica. $\Gamma(z)$ è quindi definito per ogni numero complesso z che non sia il negativo di un numero naturale, cioè per ogni $z \in \mathbb{C}$ con $z \notin \{0, -1, -2, \dots\}$. Il fattoriale è legato alla funzione gamma dalla relazione

$$n! = \Gamma(n+1)$$

Provare gamma e lgamma in R.

La formula di Stirling

Teorema 21.1. Il valore di $n!$ è sempre compreso tra

$$n^n e^{-n} \sqrt{2\pi n} e^{\frac{1}{12n} - \frac{1}{120n^2}}$$

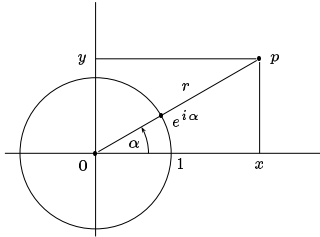
e

$$n^n e^{-n} \sqrt{2\pi n} e^{\frac{1}{12n} + \frac{1}{120n^2}}$$

Dimostrazione. Non richiesta. Richiede i potenti strumenti dell'analisi complessa.

Coordinate polari nel piano

Sia $p = (x, y)$ un punto del piano reale.



Si vede che, se $p \neq (0, 0)$, allora

$$\begin{cases} x = r \cos \alpha \\ y = r \sin \alpha \end{cases} \quad (*)$$

dove $r = \sqrt{x^2 + y^2}$, mentre l'angolo α è univocamente determinato se chiediamo $0 \leq \alpha < 2\pi$.

Nel caso $p = (0, 0)$ la rappresentazione (*) rimane valida con $r = 0$ e qualsiasi α , la biiettività della (*) viene quindi meno nel punto $p = (0, 0)$.

Scriviamo adesso $e^{i\alpha} := (\cos \alpha, \sin \alpha)$ come abbiamo già fatto nel disegno; allora la relazione (*) può anche essere scritta nella forma

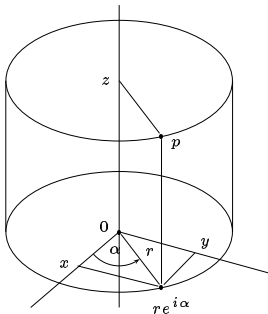
$$p = r e^{i\alpha}$$

Questo prodotto di r con $e^{i\alpha}$ può essere interpretato come prodotto dello scalare reale r con il vettore $e^{i\alpha}$ di \mathbb{R}^2 ed è allo stesso tempo il prodotto dei numeri complessi r ed $e^{i\alpha}$ come vedremo nel prossimo numero.

Coordinate cilindriche nello spazio

Dalla figura si vede che un punto $p = (x, y, z)$ dello spazio può essere rappresentato nella forma

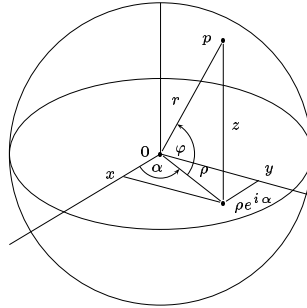
$$\begin{cases} x = r \cos \alpha \\ y = r \sin \alpha \\ z = z \end{cases}$$



La rappresentazione è univoca per quei punti per cui $(x, y) \neq (0, 0)$, quindi per tutti i punti che non si trovano sull'asse z .

Coordinate polari (o sferiche) nello spazio

Un punto $p = (x, y, z)$ dello spazio tridimensionale può anche essere rappresentato come nella figura seguente:



Avendo $\rho = r \cos \varphi$, si vede che

$$\begin{cases} x = r \cos \alpha \cos \varphi \\ y = r \sin \alpha \cos \varphi \\ z = r \sin \varphi \end{cases}$$

con

$$\begin{cases} r \geq 0 \\ 0 \leq \alpha < 2\pi \\ -\frac{\pi}{2} \leq \varphi \leq \frac{\pi}{2} \end{cases}$$

Questa rappresentazione è quella che si usa nelle coordinate geografiche di un punto della terra o della sfera celeste:

α = longitudine, φ = latitudine.

Anche in questo caso la corrispondenza non è biiettiva, perché non solo per $p = (0, 0, 0)$ la rappresentazione è valida per $r = 0$ e valori arbitrari di

α e φ , ma anche per ogni altro punto $\neq (0, 0, 0)$ dell'asse z bisogna porre $\varphi = 90^\circ$ e quindi $\cos \varphi = 0$ e $\sin \varphi = 1$ se $z > 0$ oppure $\varphi = -90^\circ$ e quindi $\cos \varphi = 0$ e $\sin \varphi = -1$ se $z < 0$, e allora ogni α va bene. Quindi su tutta l'asse z le coordinate polari non sono univocamente determinate.

Spesso al posto di φ si usa

$$\theta := 90^\circ - \varphi$$

quindi $\cos \varphi = \sin \theta$, $\sin \varphi = \cos \theta$.

Molte funzioni della matematica e della fisica presentano *simmetrie*. A una funzione $f = f(x, y, z)$ definita su \mathbb{R}^3 (per semplicità, ma spesso bisognerà studiare bene il più adatto dominio di definizione) possiamo associare la funzione $g = g(r, \alpha, \varphi)$ definita da

$$g(r, \alpha, \varphi) := f(r \cos \alpha \cos \varphi, r \sin \alpha \cos \varphi, r \sin \varphi)$$

che in caso di simmetrie può avere una forma analitica molto più semplice della f .

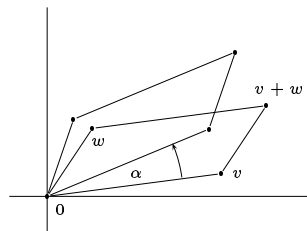
$f(x, y, z) = x^2 + y^2 + z^2$ ad esempio diventa così $g(r, \alpha, \varphi) = r^2$, una funzione di una sola variabile notevolmente più semplice. Altre volte una funzione dipende solo dalla direzione e quindi non da r ; in questo caso g è una funzione di sole due variabili e anche questa è una importante semplificazione. Nello stesso modo si usano le coordinate cilindriche e le coordinate polari piane.

Rotazioni nel piano

Consideriamo l'applicazione f_α da \mathbb{R}^2 in \mathbb{R}^2 che consiste nel ruotare un punto $v = (v_1, v_2)$ per l'angolo fissato α in senso antiorario. È chiaro che $f_\alpha(\lambda v) = \lambda f_\alpha(v)$ per ogni $\lambda \in \mathbb{R}$ e dal disegno si vede che anche

$$f_\alpha(v + w) = f_\alpha(v) + f_\alpha(w)$$

per $v, w \in \mathbb{R}^2$. Una rotazione è quindi un'applicazione lineare.



Sia e_1, e_2 la base canonica di \mathbb{R}^2 . Allora

$$v = v_1 e_1 + v_2 e_2$$

perciò

$$f_\alpha(v) = v_1 f_\alpha(e_1) + v_2 f_\alpha(e_2)$$

Ma

$$\begin{aligned} f_\alpha(e_1) &= \begin{pmatrix} \cos \alpha \\ \sin \alpha \end{pmatrix} \\ f_\alpha(e_2) &= \begin{pmatrix} -\sin \alpha \\ \cos \alpha \end{pmatrix} \end{aligned}$$

$f_\alpha(e_2)$ è il vettore magico di $f_\alpha(e_1)$! Cfr. esercizio 24.

Quindi

$$\begin{aligned} f_\alpha(v) &= v_1 \begin{pmatrix} \cos \alpha \\ \sin \alpha \end{pmatrix} + v_2 \begin{pmatrix} -\sin \alpha \\ \cos \alpha \end{pmatrix} \\ &= \begin{pmatrix} v_1 \cos \alpha - v_2 \sin \alpha \\ v_1 \sin \alpha + v_2 \cos \alpha \end{pmatrix} \end{aligned}$$

Se per una matrice

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in \mathbb{R}_2^2$$

definiamo

$$Av = \begin{pmatrix} av_1 + bv_2 \\ cv_1 + dv_2 \end{pmatrix}$$

vediamo che possiamo prendere

$$A = \begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix}.$$

Notiamo anche che le colonne di A sono proprio $f_\alpha(e_1)$ e $f_\alpha(e_2)$.

Numeri complessi in R

Un *numero complesso* è un punto $z = (x, y)$ del piano reale. Secondo questa definizione, i numeri complessi non sono nuovi come oggetti. Definiremo però nel prossimo numero le due operazioni, addizione e moltiplicazione, per i numeri complessi, di cui abbiamo già parlato a pagina 6.

Per stavolta usiamo i numeri complessi solo come la forma più comoda per rappresentare punti del piano in R. In matematica il numero complesso $z = (x, y)$ viene anche scritto nella forma $z = x + yi$, ad esempio $(3, 8) = 3 + 8i$. In modo quasi identico vengono rappresentati i numeri complessi in R, mentre però in matematica per $4 + 1i$ si scrive $4 + i$, in R bisogna scrivere $4+1i$. Anche i da solo non è ammesso, dobbiamo scrivere $1i$; R stesso nell'output utilizza $0+1i$. Si può usare 0 per il numero complesso 0 (nell'output $0+0i$). La parte reale si ottiene con $\text{Re}(z)$, la parte immaginaria con $\text{Im}(z)$.

Molte funzioni grafiche in R sono definite per successioni di punti. Siccome in statistica accade spesso che si hanno due serie di misurazioni, ad esempio $x = (x_1, \dots, x_m)$ e $y = (y_1, \dots, y_m)$, molte di queste funzioni possono essere usate nella forma $f(x, y, \dots)$. Una coppia di misurazioni (cioè di vettori) $(x, y) \in \mathbb{R}^m \times \mathbb{R}^m$ è però equivalente a una successione $(x_1, y_1), \dots, (x_m, y_m)$ di punti del piano, cioè a un elemento di $(\mathbb{R}^2)^m$. Questa *dualità* è onnipotente nella statistica multivariata. In R dovremmo allora scrivere $f(c(x_1, y_1), \dots, c(x_m, y_m), \dots)$ che ci costringe a una rappresentazione un po' difficoltosa.

Le stesse funzioni grafiche operano correttamente se diamo i punti come una successione di numeri complessi, possiamo cioè usarle ad esempio nella forma $f(c(3+2i, 5-1i, 3i, 10+6i))$. Non dimenticare l'operatore c .

Funzioni in R

```
Cerchi = function(x,raggi,
  colore=1,coloreri=NA)
{for (p in x) symbols(p,circles=raggi,
  add=T,inches=F,fg=colore,bg=coloreri)}
```

```
Grafica = function(x,y,cornice=FALSE,
  rxy=1,assi=FALSE,marg=rep(0,4))
{par(mai=marg); par(lwd=0.5);
plot(x,y,type="n",xlab="",ylab="",
  asp=rxy,axes=assi,frame.plot=cornice)}
```

```
Postscript = function(file,larg,alt)
{postscript(file,width=larg,height=alt,
  horizontal=FALSE,onefile=FALSE,
  paper="special")}
```

```
Punti = function(x,simb,tip="p",
  colore=1,coloreri=NA,scla=1)
{points(Re(x),Im(x),pch=simb,type=tip,
  col=colore,bg=coloreri,cex=scla)}
```

```
Rettangoli = function(x,larg,alt,
  colore=1,coloreri=NA)
{rect(Re(x),Im(x),Re(x)+larg,Im(x)+alt,
  col=coloreri,border=colore)}
```

Da questi esempi si dovrebbero poter dedurre le regole più importanti per la definizione di funzioni in R. Parametri predefiniti sono assegnati con $=$. Le funzioni `points`, `symbols` e `rect` verranno discusse adesso.

points

Per disegnare punti e simboli in R si possono usare le funzioni `points` e `symbols`. Il primo comando ha il formato

```
points(punti, pch=simbolo,
  type=tipo="p", col=colore,
  bg=coloreri, cex=scla=1)
```

Usiamo qui la seguente convenzione: In primo luogo gli argomenti possibili nelle funzioni di R sono in genere talmente tanti, che i prototipi che indichiamo corrispondono soltanto a un formato scelto per semplicità e funzionalità. R permette inoltre argomenti riconoscibili per nome, che quindi non devono necessariamente seguirsi nell'ordine indicato. Quando nel nostro prototipo appaiono due segni di uguaglianza, ciò significa che l'ultimo valore è il valore predefinito in R. Analizziamo in dettaglio i parametri.

`punti` sono le coordinate dei punti, che possono essere dati sia come elementi x, y di $\mathbb{R}^m \times \mathbb{R}^m$ oppure mediante un vettore di numeri complessi.

`simbolo` è il simbolo con cui il punto viene rappresentato. Questo simbolo può essere una lettera oppure un numero tra 1 e 25. Le forme che corrispondono a questi numeri sono elencate nella prima figura a pagina 19.

`tipo` può omettere oppure scelto tra "l" e "b". Se viene omettere, assume il valore predefinito "p", che sta per *point* e indica che nel punto indicato viene disegnata la figura desiderata. "l" fa in modo che i punti vengano uniti con linee (in questo caso `pch` è superfluo), mentre l'interessante opzione "b" (per *both*) implica che vengano disegnati sia i simboli che le linee congiungenti.

`colore` è il colore in cui il simbolo (o il suo bordo quando è riempibile) viene disegnato. Se manca, viene usato il colore di disegno attualmente impostato.

`coloreri` è il colore di riempimento per i simboli con i numeri da 21 a 25. I colori possono essere indicati per nome (ad esempio "red", "green", ...) un elenco dei nomi disponibili lo si ottiene, almeno sotto Linux, con il comando `colors()` oppure in formato RGB, ad esempio "#ff0000" per il rosso.

`scla` è la scala. È normalmente impostata ad 1, per avere un formato più grande si può aumentare la scala, ad esempio con `scla=2.7`.

La figura a pagina 19 che rappresenta i simboli e i numeri a cui corrispondono è stata ottenuta con questi comandi:

```
Postscript("../ps/19-simboli.ps",2,2)
lato=c(0,2); par(bg="#7ac97a")
Grafica(lato,lato,cornice=TRUE)
for (i in 1:18) {j=i-1; x=0.25*(1+j%7);
  y=1.6-0.5*(j%7); points(x,y,pch=i);
  text(x,y+0.2,i,cex=0.5)}
for (i in 19:25) {j=i-19;
  x=0.25*(1+j%7); y=0.1;
  points(x,y,pch=i,bg="red");
  text(x,y+0.2,i,cex=0.5)}
dev.off()
```

Per ottenere l'output su una finestra grafica è sufficiente inserire `locator(1)` prima di `dev.off()`.

`a:b` è lo stesso come `seq(a,b)`.

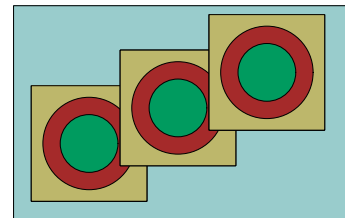
symbols e rect

Nella funzione `Cerchi` dell'ultimo paragrafo abbiamo usato la funzione `symbols` che permette di disegnare simboli più complessi di quelli che si ottengono con `points`. È una funzione che si usa poco però e anche noi la utilizziamo solo per disegnare cerchi il cui raggio (in pollici, come tutte le misure di R) possono essere assegnati. Cerchi si possono anche ottenere con `points`, dove però invece del raggio si può indicare solo la scala relativa impostando il parametro `cex` (o `scla` in `Punti`).

Anche rettangoli si possono disegnare sia con `symbols` che con `points`; siccome esiste però una funzione apposita `rect` in R, possiamo adattarla definendo una nostra funzione `Rettangoli`. Usiamo anche qui il plurale perché tutte queste figure permettono di disegnare serie di figure.

L'immagine è stata ottenuta con

```
Postscript("../ps/22-dischi.ps",1.9,1.2)
Grafica(c(0,1.9),c(0,1.2))
Rettangoli(0,1.9,1.2,coloreri="#99cccc")
punti=c(0.1+0.1i,0.6+0.3i,1.1+0.5i)
lato=0.65; lati=rep(lato,3)
for (p in punti)
{Rettangoli(p,lati,lati,coloreri="#bdb76b")
centro=p+(lato+lato*(0+1i))/2;
Cerchi(centro,lato*0.4,coloreri="brown")
Cerchi(centro,lato*0.25,
  coloreri="#009b5f")}
dev.off()
```



Il semplice ma potente `for` di R viene usato nella forma

```
for (x in successione) istruzione
```

`rep(x,n)` è la successione $c(x, \dots, x)$ con n ripetizioni di x .

Esercizi per gli scritti

23. Giustificare la regola della nota 20.16.
24. Per un vettore $v = (a, b)$ del piano il vettore $w = (-b, a)$ che si ottiene da v per rotazione di 90 gradi in senso antiorario si chiama il *vettore magico* di v . Usare le matrici di rotazione per ottenere w da v .
25. Usando le matrici di rotazione dimostrare che $f_\alpha \circ f_\beta = f_{\alpha+\beta} = f_\beta \circ f_\alpha$.
26. Se w è il vettore magico di v , allora $f_\alpha(w)$ è il vettore magico di $f_\alpha(v)$. Usare l'esercizio 25! Riflettere sul caso $v = e_1$.
27. Scrivere una funzione per disegnare rettangoli in cui il primo argomento corrisponde al vettore dei centri.

I numeri complessi

Abbiamo già detto che un numero complesso è un punto $z = (x, y)$ del piano reale e che secondo questa definizione i numeri complessi non sono nuovi come oggetti. Definiamo però adesso per i numeri complessi due operazioni, addizione e moltiplicazione.

Siano $z = (x, y)$ e $c = (a, b)$. Allora

$$\begin{aligned} c + z &:= (a + x, b + y) \\ cz &:= (ax - by, ay + bx) \end{aligned}$$

L'addizione è l'addizione vettoriale nel piano, la moltiplicazione è invece motivata nel modo seguente.

L'equazione $x^2 = -1$ non ha soluzioni reali (perché $x \in \mathbb{R}$ implica $x^2 \geq 0$ e $1 + x^2 \geq 1 > 0$). Ci chiediamo allora se è possibile aggiungere ai numeri reali altri numeri, numeri immaginari, tra cui un numero i (i appunto perché immaginario) che soddisfa l'equazione

$$i^2 = -1$$

Naturalmente vorremmo che le usuali leggi aritmetiche siano conservate anche con i nuovi numeri. Con la nostra definizione tutto funziona bene:

Chiamiamo il punto $(0, 1)$ del piano i , poniamo cioè

$$i := (0, 1)$$

e identifichiamo il numero reale a con il numero complesso $(a, 0)$ - ciò significa geometricamente che consideriamo la retta reale come sottoinsieme del piano nel solito modo, identificandola con l'asse delle x .

Siano $a, b, x, y \in \mathbb{R}$. Allora:

(1) $(a, 0) \cdot (x, y) = (ax, ay)$, dove a sinistra il prodotto è il nuovo prodotto per i numeri complessi. Infatti, secondo la nostra definizione,

$$\begin{aligned} (a, 0) \cdot (x, y) &= (ax - 0 \cdot y, 0 \cdot x + ay) \\ &= (ax, ay) \end{aligned}$$

Ciò significa che $(a, 0)(x, y)$ è uguale ad $a \cdot (x, y)$, cioè al prodotto dello

scalare reale a con il vettore (x, y) del piano.

$$(2) \quad i^2 = -1$$

Infatti

$$\begin{aligned} i^2 &= (0, 1) \cdot (0, 1) \\ &= (0 \cdot 0 - 1 \cdot 1, 1 \cdot 0 + 0 \cdot 1) \\ &= (-1, 0) = -1 \end{aligned}$$

(3) Più in generale abbiamo

$$\begin{aligned} (a, b) &= (a, 0) + (0, b) \\ &= a + b(0, 1) = a + bi \end{aligned}$$

e quindi anche $(x, y) = x + yi$, e

$$\begin{aligned} (a + bi)(x + yi) &= ax + ayi + bxi + byi^2 \\ &= ax - by + (ay + bx)i \end{aligned}$$

Osserviamo bene quest'ultima formula. Il risultato è in accordo con la nostra definizione per la moltiplicazione di numeri complessi, ma è stato raggiunto eseguendo il calcolo secondo le regole algebriche usuali, a cui abbiamo aggiunto la nuova legge $i^2 = -1$.

(4) E infatti per le operazioni $+$ e \cdot definite all'inizio valgono le stesse leggi algebriche come per i numeri reali (cfr. pagina 6), perché, come si verifica adesso facilmente, l'insieme dei numeri complessi con queste operazioni è un anello commutativo in cui il numero reale $1 = (1, 0)$ è l'elemento neutro per la moltiplicazione. Quest'ultima affermazione segue dal punto (1). Vedremo a pagina 24 che i numeri complessi formano in verità un campo, cioè che ogni numero complesso diverso da zero possiede un inverso per la moltiplicazione.

Il campo dei numeri complessi viene denotato con \mathbb{C} . In pratica $\mathbb{C} = \mathbb{R}^2$, però con le nuove operazioni. Abbiamo già visto che $(a, b) = a + bi$, e siccome $bi = ib$, si può anche scrivere $(a, b) = a + ib$.

In questo numero

- 23 I numeri complessi
La formula di Euler
- 24 Il campo dei numeri complessi
La formula di de Moivre
Parte reale e parte immaginaria
Disuguaglianze fondamentali
Il segno del prodotto scalare
Poligoni con \mathbb{R}
- 25 Radici di un polinomio
Radici n -esime dell'unità
Radici di un numero complesso
Esercizi 28-43

La formula di Euler

Abbiamo introdotto già a pagina 21 la notazione

$$e^{i\alpha} := (\cos \alpha, \sin \alpha)$$

per $\alpha \in \mathbb{R}$. Adesso la possiamo riscrivere nella forma

$$e^{i\alpha} = \cos \alpha + i \sin \alpha$$

(formula di Euler). Questa notazione, per il momento puramente simbolica (perché solo nei corsi di Analisi si potrà dimostrare che si tratta veramente di una potenza), non è solo molto comoda, come vedremo adesso, ma anche estremamente importante nella teoria.

Dal teorema di addizione per le funzioni trigonometriche si deduce immediatamente che

$$e^{i(\alpha+\beta)} = e^{i\alpha} \cdot e^{i\beta}$$

per $\alpha, \beta \in \mathbb{R}$ (esercizio 33). Si vede che nel campo complesso il teorema di addizione assume una forma molto più semplice.

Per $z = x + iy$ possiamo anche più in generale definire

$$e^z := e^x e^{iy}$$

Allora

$$e^{z+w} = e^z \cdot e^w$$

per ogni $z, w \in \mathbb{C}$. La funzione esponenziale è quindi un omomorfismo

$$(\mathbb{C}, +) \longrightarrow (\mathbb{C}, \cdot)$$

Leonhard Euler (1707-1783), matematico svizzero-tedesco, passò gran parte della sua vita a Pietroburgo e a Berlino. È stato uno dei più prolifici matematici di tutti i tempi; la sua opera riempie 88 volumi. Non era solo un geniale analista, ma ha inventato anche la teoria dei grafi, anticipando un campo della matematica discreta oggi molto studiato per le numerose applicazioni (ad esempio in ricerca operativa e in informatica). Ha lavorato su quasi tutti i campi della matematica pura e applicata del suo tempo.

- C. Boyer. Storia della matematica. Mondadori.
- U. Bottazzini. Il flauto di Hilbert. Utet 1990.
- C. Berge. The theory of graphs. Dover 2001.
- R. Diestel. Graph theory. Springer 1997.

Il campo dei numeri complessi

Abbiamo visto a pagina 21 che ogni punto $z = (x, y)$ di \mathbb{R}^2 può essere scritto nella forma

$$z = (r \cos \alpha, r \sin \alpha)$$

dove $r \geq 0$ è univocamente determinato ed $\alpha \in \mathbb{R}$. Abbiamo anche visto che

$$r = |z| = \sqrt{x^2 + y^2}$$

Se $z \neq 0$, anche α è univocamente determinato se chiediamo $0 \leq \alpha < 2\pi$ oppure, come il matematico preferisce dire, univocamente determinato modulo 2π .

Possiamo quindi scrivere ogni numero complesso z nella forma

$$z = r e^{i\alpha}$$

anch'essa già anticipata a pagina 21, con r ed α come sopra.

Sia adesso anche $\beta \in \mathbb{R}$. Allora

$$e^{i\beta} z = r e^{i(\alpha+\beta)}$$

come segue dal teorema di addizione visto a pagina 24. Il prodotto $e^{i\beta} z$ ha quindi la stessa lunghezza di z ed un angolo aumentato di β rispetto a z . Ciò significa che la moltiplicazione con $e^{i\beta}$ è la stessa cosa come una rotazione per l'angolo β in senso antiorario.

Prendiamo adesso un numero complesso c arbitrario. Lo possiamo rappresentare nella forma $c = s e^{i\beta}$ con $s \geq 0$. Per il prodotto cz otteniamo evidentemente $cz = s r e^{i(\alpha+\beta)}$ e quindi vediamo che la moltiplicazione con un numero complesso consiste sempre di una rotazione combinata con un allungamento (o accorciamento se $|c| < 1$).

Definizione 24.1. Sia $z = x + iy \in \mathbb{C}$ con $x, y \in \mathbb{R}$. Allora $\bar{z} := x - iy$ si chiama il numero complesso *congiugato* a z .

Geometricamente \bar{z} si ottiene mediante riflessione di z rispetto all'asse reale. È chiaro che $\bar{\bar{z}} = z$.

Lemma 24.2. Sia $z = x + iy$. Allora

$$z\bar{z} = x^2 + y^2 = |z|^2.$$

Dimostrazione. Immediata. Perché è anche un caso speciale dell'esercizio 30?

Nota 24.3. Ogni numero complesso $z \neq 0$ possiede un inverso rispetto alla moltiplicazione, infatti

$$z \cdot \frac{\bar{z}}{z\bar{z}} = \frac{z\bar{z}}{z\bar{z}} = 1$$

per cui possiamo porre

$$\frac{1}{z} = \frac{\bar{z}}{z\bar{z}}$$

o, equivalentemente,

$$\frac{1}{x + iy} = \frac{x - iy}{x^2 + y^2}$$

$(\mathbb{C}, +, \cdot)$ è quindi un campo. Come in ogni campo anche in \mathbb{C} l'inverso è univocamente determinato.

La formula di de Moivre

Sia $z = r e^{i\alpha} \in \mathbb{C}$. Per $n \geq 1$ allora, secondo le formule viste precedentemente, abbiamo

$$z^n = r^n e^{in\alpha}$$

Abraham de Moivre (1667-1754) era un matematico francese emigrato giovane in Inghilterra. Ha scritto un famoso trattato sul calcolo delle probabilità (*Doctrines of chances*, 1718).

Parte reale e parte immaginaria

Definizione 24.4. $z = x + iy$ (con x, y reali come sempre) sia un numero complesso. Definiamo allora

$$\operatorname{Re} z := x$$

$$\operatorname{Im} z := y$$

Re z si chiama la *parte reale* di z , Im z la *parte immaginaria*.

Osservazione 24.5. Sia $z \in \mathbb{C}$. Allora

$$\operatorname{Re} z = \frac{z + \bar{z}}{2}$$

$$\operatorname{Im} z = \frac{z - \bar{z}}{2i}$$

Osservazione 24.6. Sia $z \in \mathbb{C}$. Allora

$$|\operatorname{Re} z| \leq |z| \quad e \quad |\operatorname{Im} z| \leq |z|.$$

Dimostrazione. Infatti, con $z = x + iy$,

$$|\operatorname{Re} z| = |x| = \sqrt{x^2} \leq \sqrt{x^2 + y^2} = |z|.$$

Nello stesso modo per Im z .

Osservazione 24.7. Per un numero complesso z la parte reale e la parte immaginaria di z non sono altro che le coordinate di z come punto di \mathbb{R}^2 . È quindi chiaro che, se w è un altro numero complesso, si ha $z = w$ se e solo se z e w coincidono sia nelle parti reali che nelle parti immaginarie.

Usare questa osservazione nella dimostrazione degli esercizi 41 e 42.

Disuguaglianze fondamentali

Teorema 24.8. Siano $x = (x_1, \dots, x_n)$ e $y = (y_1, \dots, y_n)$ due punti di \mathbb{R}^n . Allora

$$||x, y|| \leq |x||y|$$

Questa è una delle disuguaglianze più importanti di tutta la matematica e prende il nome di **disuguaglianza di Cauchy-Schwarz**.

Dimostrazione. Possiamo ricondurre questa fondamentale disuguaglianza al caso $n = 2$. Infatti i due vettori stanno su un piano e il prodotto scalare si esprime mediante l'angolo α che essi formano in questo piano (pagina 15):

$$||x, y|| = |x||y| \cos \alpha$$

e siccome $|\cos \alpha| \leq 1$ abbiamo

$$||x, y|| = |x||y| \cos \alpha \leq |x||y|$$

Proposizione 24.9. Siano ancora

$x = (x_1, \dots, x_n)$ e $y = (y_1, \dots, y_n)$ due punti di \mathbb{R}^n . Allora

$$|x + y| \leq |x| + |y|$$

Questa seconda disuguaglianza fondamentale è detta **disuguaglianza triangolare**.

Dimostrazione. Ciò è una facile conseguenza della formula

$$|x + y|^2 = |x|^2 + |y|^2 + 2||x, y||$$

per il prodotto scalare vista a pagina 15 e della disuguaglianza di Cauchy-Schwarz:

$$\begin{aligned} |x + y|^2 &= |x|^2 + |y|^2 + 2||x, y|| \leq \\ &= |x|^2 + |y|^2 + 2|x||y| = \\ &= (|x| + |y|)^2 \end{aligned}$$

per cui anche

$$|x + y| \leq |x| + |y|$$

Il segno del prodotto scalare

Nota 24.10. Nella disuguaglianza di Cauchy-Schwarz anche a sinistra dobbiamo mettere il segno di valore assoluto, perché il prodotto scalare può essere negativo.

Infatti il segno del prodotto scalare ha una importantissima interpretazione geometrica: Siano come finora $x = (x_1, \dots, x_n)$ e $y = (y_1, \dots, y_n)$ due punti di \mathbb{R}^n , entrambi diversi da 0. Come nella dimostrazione della disuguaglianza di Cauchy-Schwarz sia α l'angolo che i due vettori formano in un piano comune (un tale piano esiste sempre ed è univocamente determinato se i due vettori non sono paralleli). Sappiamo che

$$||x, y|| = |x||y| \cos \alpha$$

Per ipotesi $|x| > 0$ e $|y| > 0$. Ciò implica che $||x, y||$ e $\cos \alpha$ hanno lo stesso segno; in particolare

$$||x, y|| \geq 0 \iff \cos \alpha \geq 0$$

e

$$||x, y|| = 0 \iff \cos \alpha = 0$$

Fissiamo adesso x . Allora i vettori $y \in \mathbb{R}^n$ per i quali $\cos \alpha = 0$ sono esattamente i vettori ortogonali ad x . Essi formano l'*iperpiano ortogonale* ad x (una retta ortogonale ad x in \mathbb{R}^2 , un piano ortogonale ad x in \mathbb{R}^3). Come si vede dalle figure a pagina 13 per $y = x$ il coseno di α è uguale ad 1, e se, partendo da $y = x$, avviciniamo y all'iperpiano ortogonale di x , il coseno diventa sempre più piccolo, rimanendo però positivo fino a quando non tocchiamo l'iperpiano ortogonale. Se y passa invece dall'altra parte dell'iperpiano, il coseno di α diventa negativo.

Avendo $||x, y||$ e $\cos \alpha$ però lo stesso segno, otteniamo il seguente enunciato geometrico, importante anche in molte applicazioni, ad esempio nella teoria dell'ottimizzazione:

Teorema 24.11. Siano $x = (x_1, \dots, x_n)$ e $y = (y_1, \dots, y_n)$ due punti di \mathbb{R}^n , entrambi diversi da zero. Allora $||x, y|| > 0$ se e solo se y si trova dalla stessa parte dell'iperpiano ortogonale ad x come x stesso.

Poligoni con R

La semplicità delle seguenti costruzioni è impressionante - ma richiede la nostra matematica! Esaminare bene ogni singola riga del programma e variarlo.

```
Postscript("25-pentagoni.ps",1.9,1.2)
Grafica(c(-1.1,2.3),c(-1.1,1.1))
Rettangoli(-1.1-1.1i,3.4,2.2,
  coloreri="yellow")
par(lwd=5) # Matita spessa.
angoli=pi/4+seq(0,2*pi,2*pi/5)
poligononero=cos(angoli)+1i*sin(angoli)
lines(poligononero)
z45=cos(pi/4)+1i*sin(pi/4)
poligonorosso=z45*poligononero
par(col="red")
lines(poligonorosso)
poligonoverde=0.5*poligonorosso+0.2
par(col="green")
lines(poligonoverde)
par(lwd=2)
centri=1.7+1i*seq(-0.55,0.55,0.25)
Cerchi(centri,0.3,
  coloreri="forest green",colore="red")
dev.off()
```

Si ottiene la figura a pagina 25.

Radici di un polinomio

Teorema 25.1. Siano

$f = a_0 + a_1x + \dots + a_nx^n$
 un polinomio con coefficienti complessi,
 $n \in \mathbb{N} + 1$ e $a_n \neq 0$. Allora esistono numeri
 complessi $\alpha_1, \dots, \alpha_n$, univocamente deter-
 minati, tali che

$$f = (x - \alpha_1) \cdots (x - \alpha_n)$$

Questa uguaglianza è intesa come uguaglianza di polinomi, cioè sviluppando il prodotto a destra si ottiene un polinomio con gli stessi coefficienti di f e quindi proprio f .

È chiaro anche che $f(\alpha_k) = 0$ per ogni k e si può dimostrare che ogni radice di f , cioè ogni α per cui $f(\alpha) = 0$, è uno degli α_k .

Dimostrazione. Non richiesta. Esistono molte dimostrazioni, le più semplici utilizzano ancora l'analisi complessa.

Il teorema 25.1 si chiama il *teorema fondamentale dell'algebra*.

In \mathbb{R} le radici di un polinomio possono essere ottenute con la funzione `polyroot` che prende come argomento il vettore dei coefficienti di f elencati iniziando con il coefficiente costante. Per arrotondare i risultati usiamo la funzione `round`, il cui secondo argomento è il numero di cifre decimali a cui si vuole arrotondare. Esempio:

```
radici=polyroot(c(6,-5,7,-5,1))
print(round(radici,2))
```

con output 0+1i 0-1i 2+0i 3+0i. Infatti

$$6 - 5x + 7x^2 - 5x^3 + x^4 \\ = (x-2)(x-3)(x-i)(x+i).$$

Radici n -esime dell'unità

Sia $n \in \mathbb{N} + 1$ fissato. Consideriamo il numero

$$\varepsilon = e^{\frac{2\pi i}{n}}$$

(che naturalmente dipende da n). Dalla formula di de Moivre segue che ε elevato alla n -esima potenza, è uguale a 1. È anche chiaro che

$$(\varepsilon^k)^n = (\varepsilon^n)^k = 1^k = 1$$

per ogni $k \in \mathbb{Z}$. Consideriamo adesso i numeri $1, \varepsilon, \varepsilon^2, \dots, \varepsilon^{n-1}$.

Dalla formula di Euler vediamo che gli ε^k si trovano tutti sul cerchio unitario, con ε^k ruotato di $\frac{2\pi}{n}$ (cioè di $\frac{360}{n}$ gradi) rispetto ad ε^{k-1} . Essi formano in altre parole (almeno per $n \geq 3$) i vertici del poligono regolare con n vertici iscritto al cerchio unitario con primo vertice uguale ad 1. Ciò implica che gli ε^k per $k = 0, 1, \dots, n-1$ sono tutti distinti tra di loro e costituiscono un insieme di n radici del polinomio $x^n - 1$, mentre per altri k i valori si ripetono. Dal teorema fondamentale dell'algebra segue che ogni radice di $x^n - 1$ è uno degli ε^k e che

$$x^n - 1 \\ = (x-1)(x-\varepsilon)(x-\varepsilon^2) \cdots (x-\varepsilon^{n-1})$$

I numeri $1, \varepsilon, \varepsilon^2, \dots, \varepsilon^{n-1}$ si chiamano le n -esime radici dell'unità.

Radici di un numero complesso

Siano $r \geq 0$ un numero reale non negativo ed $n \in \mathbb{N} + 1$. Nei corsi di Analisi si impara che esiste un unico numero $\rho \geq 0$ tale che $\rho^n = r$. Denotiamo questo numero con $\sqrt[n]{r}$.

Sia adesso $z \neq 0$ un numero complesso diverso da 0. Allora $z = re^{it}$ con $r > 0$ e $t \in \mathbb{R}$. Cerchiamo le radici n -esime di z , cioè le radici del polinomio $x^n = z$. Una radice la troviamo subito; infatti la formula di de Moivre implica che

$$\alpha_1 := \sqrt[n]{r} e^{i \frac{t}{n}}$$

soddisfa l'equazione $\alpha_1^n = z$. Se λ adesso è un numero complesso tale che $\lambda^n = 1$, allora anche

$$(\alpha_1 \lambda)^n = \alpha_1^n \lambda^n = \alpha_1^n = z$$

Però noi conosciamo i λ per cui $\lambda^n = 1$; sono le n -esime radici dell'unità. Quindi ciascuno dei numeri

$$\alpha_1 := \sqrt[n]{r} e^{i \frac{t}{n}}$$

$$\alpha_2 := \varepsilon \alpha_1$$

$$\alpha_3 := \varepsilon^2 \alpha_1$$

$$\dots$$

$$\alpha_n := \varepsilon^{n-1} \alpha_1$$

è una radice di $x^n - z$. D'altra parte la moltiplicazione di un numero complesso w con ε^k corrisponde alla rotazione di w per un angolo di $k \frac{360}{n}$ gradi e, siccome $z \neq 0$ e quindi anche $\alpha_1 \neq 0$, tutti gli α_k sono distinti tra di loro. Dal teorema fondamentale dell'algebra segue che

$$x^n - z = (x - \alpha_1) \cdots (x - \alpha_n)$$

e che ogni numero complesso α che soddisfa l'equazione $\alpha^n = z$ è uno degli α_k .

Troviamo con `R` le quarte radici di $1 + 3i$:

```
radici=polyroot(c(-1-3i,0,0,1))
print(round(radici,2))
```

L'output è

$$1.27+0.41i \quad -0.41+1.27i \\ -1.27-0.41i \quad 0.41-1.27i$$

Fare un disegno con riga e compasso e verificare il risultato. Cfr. esercizio 43.

Esercizi per gli scritti

Per vettori $v_1, \dots, v_m \in \mathbb{R}^n$ si denota talvolta con (v_1, \dots, v_m) non solo la m -pla, ma anche la matrice le cui colonne sono i vettori v_1, \dots, v_m nell'ordine indicato. In particolare possiamo formare la matrice (z, w) per due numeri complessi z e w , considerati come punti di \mathbb{R}^2 .

Siano $z = x + yi$ e $w = u + vi \in \mathbb{C}$ numeri complessi con $x, y, u, v \in \mathbb{R}$.

28. Il vettore magico di z è iz .

29. Calcolare $\|z, w\|$ e $\det(z, w)$ in termini di x, y, u, v .

30. $z\bar{w} = \|z, w\| - \det(z, w)i$.

Quindi

$$\|z, w\| = \operatorname{Re} z\bar{w} \\ \det(z, w) = -\operatorname{Im} z\bar{w}$$

31. Usiamo il simbolo \perp per indicare l'ortogonalità tra due vettori.

$$z \perp w \iff \operatorname{Re} z\bar{w} = 0 \iff z\bar{w} \in \mathbb{R}i.$$

L'insieme $\mathbb{R}i$ si chiama l'asse immaginario, gli elementi di $\mathbb{R}i$, cioè i numeri complessi con parte reale uguale a zero, si chiamano puramente immaginari. z e w sono quindi ortogonali se e solo se $z\bar{w}$ è puramente immaginario.

32. $\overline{z\bar{w}} = \bar{z}w$

$$|zw| = |z||w|.$$

Nella seconda parte non ripetere i conti!

33. $e^{i(\alpha+\beta)} = e^{i\alpha} \cdot e^{i\beta}$

per ogni $\alpha, \beta \in \mathbb{R}$.

34. $e^{z+w} = e^z \cdot e^w$.

35. Calcolare $\frac{1}{i}$.

36. Calcolare $\frac{1}{2+7i}$.

37. Calcolare $\frac{3+i}{4-2i}$.

38. Calcolare i^5 .

39. Calcolare $3+5i+6i^2+8i^3-4i^4+10i^5$.

40. Calcolare $\frac{3-i+6i^2+4i^7}{5-6i^5+i^8}$.

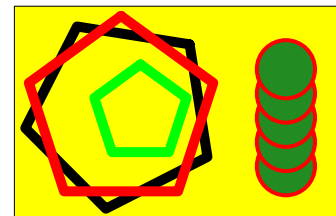
41. $\cos 3x = 4 \cos^3 x - 3 \cos x$.

Dimostrare questa formula per $x \in \mathbb{R}$ (seno e coseno possono essere definiti anche su \mathbb{C} e l'equazione rimane valida, ma ciò non può essere dimostrato qui) usando la formula di de Moivre e confrontando le parti reali e immaginarie.

42. $\cos 5x = 16 \cos^5 x - 20 \cos^3 x + 5 \cos x$.

Questo era l'esercizio 16. Dimostrarlo adesso con il metodo dell'esercizio 41. È solo un piccolo esempio di quanto il passaggio per i numeri complessi possa semplificare anche questioni che apparentemente riguardano soltanto i numeri reali.

43. Siano z un numero complesso diverso da 0 e $\alpha^4 = z$. Allora le altre radici di $x^4 - z$ sono $i\alpha, -\alpha, -i\alpha$.



Grafici di funzioni

Il grafico di una funzione f (di \mathbb{R} o definita da noi) si ottiene, come a pagina 19 il grafico del coseno, con la riga

```
lines(x,f(x))
```

adattando i parametri di ambiente (dimensioni degli intervalli per l'ascissa e per l'ordinata, colori, linee ausiliarie) secondo le necessità. Ricordiamo che il parametro x deve essere un vettore di valori reali che matematicamente siano ad esempio (x_1, \dots, x_k) ; `lines` congiunge i punti $(x_1, f(x_1)), \dots, (x_k, f(x_k))$ con segmenti di rette. Se i punti di x sono sufficientemente vicini, avremo l'impressione di una curva continua. In genere, se la funzione non oscilla troppo, è sufficiente una risoluzione di 0.01, definendo $x=\text{seq}(a,b,by=0.01)$.

Mentre la notazione per vettori

e matrici in R non è particolarmente felice, per figure piane possiamo usare con grande vantaggio la rappresentazione complessa. Per un grafico di funzioni possiamo scrivere

```
lines(x+f(x)*1i)
```

La comodità della forma complessa sta soprattutto nel fatto che possiamo così facilmente eseguire operazioni geometriche, le quali nel piano, come vedremo in questo numero, sono tutte esprimibili mediante operazioni algebriche con numeri complessi. Per spostare il grafico di 1 verso l'alto è sufficiente

```
lines(1i+x+f(x)*1i)
```

per girarlo di 90° è sufficiente

```
lines(1i*(x+f(x)*1i))
```

e così via.

Curve parametrizzate nel piano

Una curva parametrizzata in \mathbb{R}^n è un'applicazione $\varphi : I \rightarrow \mathbb{R}^n$, dove I è un intervallo (aperto o chiuso, finito o infinito, ma più spesso finito e chiuso) di \mathbb{R} . Normalmente si chiede che la φ sia almeno continua o anche che sia due volte differenziabile con derivate continue. Lo studio delle curve continue rientra nel campo della topologia e riguarda in primo luogo proprietà di deformabilità di oggetti geometrici (ci sono applicazioni in robotica, ad esempio la questione, se i bracci di un robot possano o no passare in modo continuo da una posizione a un'altra, può essere formulata e trattata con gli strumenti della topologia), lo studio delle curve e superficie differenziabili fa parte della geometria differenziale.

Quando n è uguale a 2, si parla di curve piane. In questo caso, per ogni $t \in I$ abbiamo un punto $\varphi(t)$ del piano, le cui coordinate $x(t)$ ed $y(t)$ dipendono da t e sono, appunto, legate alla φ dalla relazione $\varphi(t) = (x(t), y(t))$. In questo modo sono definite due funzioni $x : I \rightarrow \mathbb{R}^2$ e $y : I \rightarrow \mathbb{R}^2$ che a loro volta determinano la φ . Spesso si scrive allora

$$x = x(t)$$

$$y = y(t)$$

Notiamo subito che il grafico di una funzione reale $f : I \rightarrow \mathbb{R}^2$ definita su un intervallo è un caso speciale di curva parametrizzata che può essere rappresentato nella forma

$$\begin{aligned} x &= t \\ y &= f(t) \end{aligned}$$

R si presta particolarmente bene per la rappresentazione di curve piane parametrizzate, perché, una volta definito l'intervallo che si vuole usare e che deve essere rappresentato come successione finita t di punti, è sufficiente l'istruzione

```
lines(x(t),y(t))
```

nel programma per ottenere la curva. Abbiamo osservato a pagina 17 che il nome t in R è riservato (viene utilizzata per formare la trasposta di una matrice), ma lo possiamo utilizzare per le variabili locali all'interno di una funzione (naturalmente solo se non abbiamo contemporaneamente bisogno della trasposta).

Anche per le curve parametrizzate useremo la notazione complessa, quindi matematicamente

$$\varphi(t) = x(t) + y(t)i$$

e nel disegno della curva in R useremo l'istruzione

```
lines(x(t)+y(t)*1i)
```

ma anche, se la φ è rappresentata direttamente in R come funzione `phi` a valori complessi, istruzioni della forma

```
lines(phi(t))
```

In questo numero

- 26 Grafici di funzioni
Curve parametrizzate nel piano
Octobrina elegans
- 27 Octobrinidae I
Octobrinidae II
- 28 abline
Funzioni iperboliche
Parabrinidae
Rotazioni
- 29 expression ed eval
Testi matematici
Ellissi
Iperboli
- 30 Rette e segmenti
Equazione di una retta
Proiezione su una retta
Il sito CRAN di Ferrara
Esercizi 44-53

Octobrina elegans

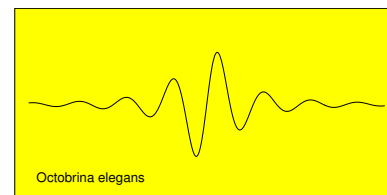
Rappresentiamo il grafico della funzione

$$\bigcirc_x \frac{2 \sin 4x}{1+x^2}$$

a cui in R diamo il nome `Oct`:

```
Oct = function (x)
{2*sin(4*x)/(1+x*x)}
```

nell'intervallo $[-6, 6]$



mediante le istruzioni

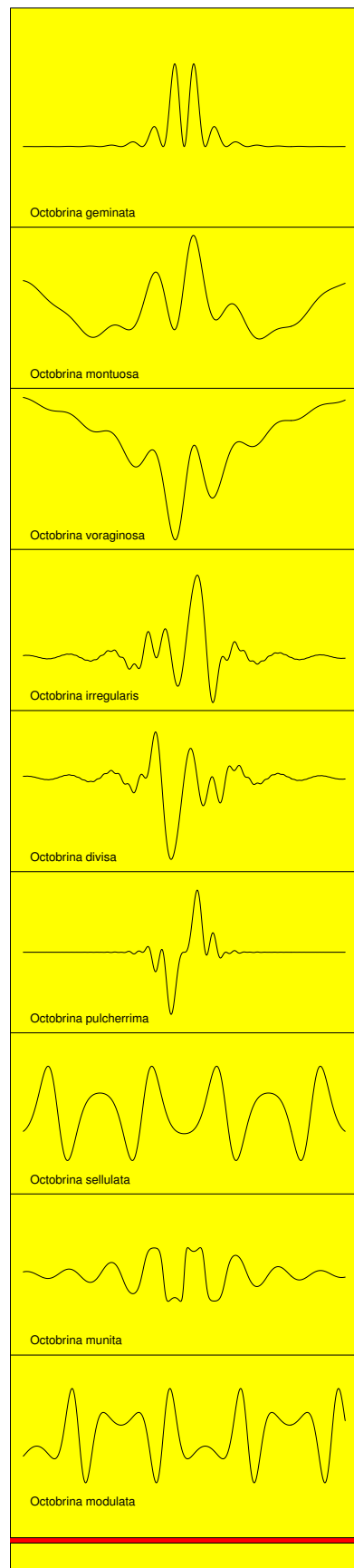
```
Postscript("26-octobrina.ps",2,1)
latox=c(-6,6); latoy=c(-3,3);
par(bg="yellow",cex=0.4)
Grafica(latox,latoy,cornice=T)
t=seq(-6,6,by=0.01)
lines(t+Oct(t)*1i)
text(-6-2.5i,"Octobrina elegans",pos=4)
dev.off();
```

Per vedere la figura sullo schermo bisogna eliminare la prima riga e inserire `locator(1)` prima di `dev.off()`.

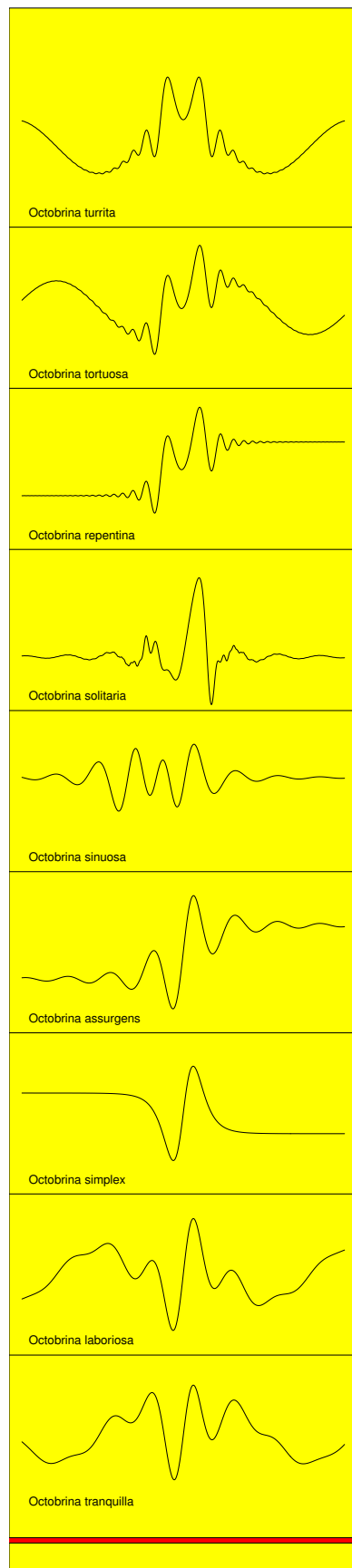
Il programma contiene una sola nuova istruzione, `text`, che viene usata per aggiungere testi a una figura. Il primo parametro indica il punto a cui si riferisce la posizione del testo; se il parametro `pos` è uguale a 4, il testo viene inserito alla destra del punto di riferimento. Quando questo parametro manca, il testo viene centrato nel punto. Vedere `?text` per le altre possibilità.

Per ottenere una scritta piccola abbiamo modificato il parametro grafico `cex` (nella terza riga); `cex` è un'abbreviazione per *character expansion*; l'abbiamo già incontrata a pagina 22.

Octobrinidae I



Octobrinidae II



Il grafico dell'octobrina è così elegante perché presenta una simmetria non perfetta. Se a prima vista la curva può sembrare simmetrica, guardando più attentamente vediamo invece che le due metà si distinguono nel segno; infatti la funzione è una funzione dispari - una funzione f si chiama *dispari*, se

$$f(-x) = -f(x)$$

perché il seno è una funzione dispari e il quadrato una funzione pari; d'altra parte il fattore smorzante $\frac{1}{1+x^2}$ riduce l'ampiezza della curva quando $|x|$ diventa più grande e attenua così la differenza tra le due parti.

Combinando la funzione ad altre funzioni elementari abbiamo ottenuto i grafici su questa pagina che corrispondono alle seguenti funzioni di t , con $f = \frac{2 \sin 4t}{1+t^2}$:

O. geminata	$f^2(t)$
O. montuosa	$f(t) + \cos t$
O. voraginosa	$f(t) + \log(1+t^2)$
O. irregularis	$f(t) + f(t^2)$
O. divisa	$f(t) - f(t^2)$
O. pulcherrima	$f(t)f(t^2)$
O. sellulata	$f(\cos t)$
O. munita	$\frac{2f(t)}{1+f^2(t)}$
O. modulata	$f(\sin t + \cos t)$
O. turrita	$f(t^2) + \cos t$
O. tortuosa	$f(t^2) + \sin t$
O. repentina	$f(t^2) + \tanh t$
O. solitaria	$0.7(f(t) + f(t^2) + f(t^3))$
O. sinuosa	$f(t) + f(t+1) + f(t+2)$
O. assurgens	$f(t) + \tanh t$
O. simplex	$f(\tanh t)$
O. laboriosa	$f(t) + (\tanh t)(\cos t)$
O. tranquilla	$f(t) + (\tanh t)(\sin t)$

Il quadro delle figure nella seconda colonna è stato ottenuto con il seguente programma in R:

```
# 272-octobrinidae

Postscript("272-octobrinidae.ps",2,9)
latox=c(-6,6); latoy=c(0,54)
par(cex=0.4,bg="yellow")
Grafica(latox,latoy,cornice=T)
t=seq(-6,6,by=0.01)
for (k in 1:8) abline(h=6*k)
Rettangoli(-7-1i,14,0.2,coloreri="red")
lines(51i+OctobrinaTurrita(t))
text(-6+48.5i,"Octobrina turrita",pos=4)
lines(45i+OctobrinaTortuosa(t))
text(-6+42.5i,"Octobrina tortuosa",pos=4)
lines(39i+OctobrinaRepentina(t))
text(-6+36.5i,"Octobrina repentina",pos=4)
lines(32i+OctobrinaSolitaria(t))
text(-6+30.5i,"Octobrina solitaria",pos=4)
lines(27.5i+OctobrinaSinuosa(t))
text(-6+24.5i,"Octobrina sinuosa",pos=4)
lines(21i+OctobrinaAssurgens(t))
text(-6+18.5i,"Octobrina assurgens",pos=4)
lines(15i+OctobrinaSimplex(t))
text(-6+12.5i,"Octobrina simplex",pos=4)
lines(9i+OctobrinaLaboriosa(t))
text(-6+6.5i,"Octobrina laboriosa",pos=4)
lines(3i+OctobrinaTranquilla(t))
text(-6+0.5i,"Octobrina tranquilla",pos=4)
dev.off()
```

abline

Nel programma a pagina 27 per la suddivisione del quadro abbiamo usato la funzione `abline` di R. Essa può essere utilizzata in vari modi; con

```
abline(h=y)
```

si ottiene una retta orizzontale all'altezza y , con

```
abline(v=x)
```

una retta verticale con ascissa x . Si possono anche disegnare altre rette, ma tranne per questi casi semplici è forse preferibile usare `lines`. Comoda è la possibilità di disegnare più linee con un'unico comando `abline`:

```
abline(h=0,v=c(0,1,3))
```

Funzioni iperboliche

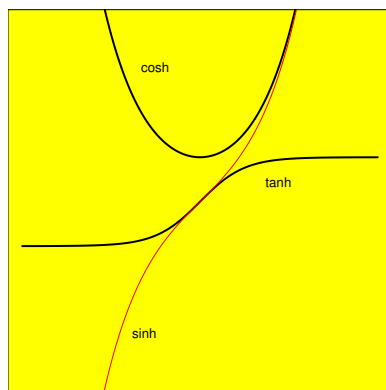
Le funzioni iperboliche \cosh , \sinh e \tanh sono definite da

$$\cosh x := \frac{e^x + e^{-x}}{2}$$

$$\sinh x := \frac{e^x - e^{-x}}{2}$$

$$\tanh x := \frac{\sinh x}{\cosh x} = \frac{e^{2x} - 1}{e^{2x} + 1}$$

Tutte e tre hanno importanti applicazioni tecniche; la tangente iperbolica \tanh è anche nota sotto il nome di *funzione sigmoide* e viene spesso utilizzata per modellare la crescita di popolazioni (*crescita logistica*) o la formazione di impulsi, ad esempio nelle *reti neurali*.

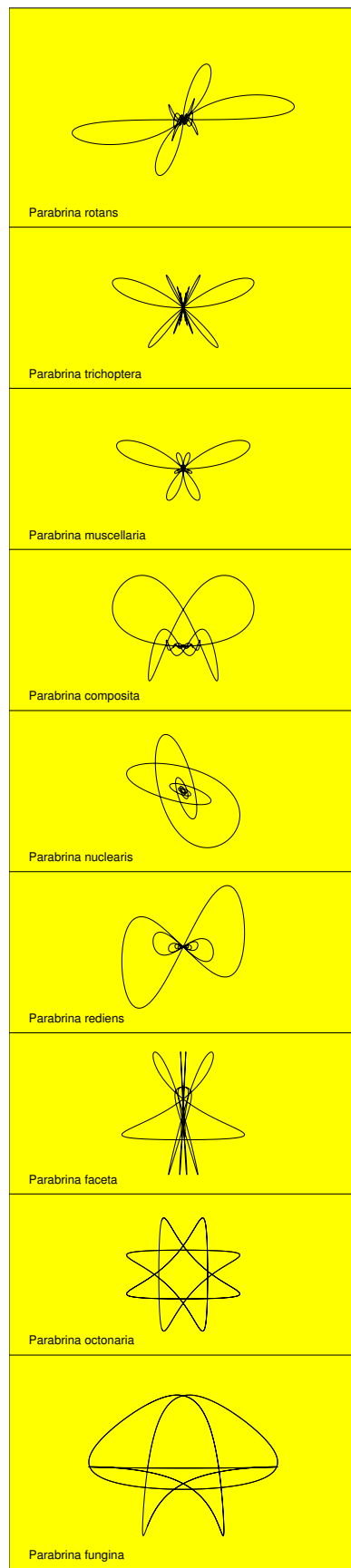


La figura è stata ottenuta con

```
# 28-iperboliche
```

```
Postscript("28-iperboliche.ps",2,2)
latox=c(-4,4); latoy=c(-4,4)
par(cex=0.4,bg="yellow")
Grafica(latox,latoy,cornice=T)
x=seq(-4,4,by=0.01)
par(lwd=1)
lines(x,cosh(x))
text(-1.5+3i,"cosh",pos=4)
lines(x,tanh(x))
text(1.3+0.4i,"tanh",pos=4)
par(col="red",lwd=0.5)
lines(x,sinh(x))
par(col=1)
text(-1.7-3i,"sinh",pos=4)
dev.off()
```

Parabrinidae



Le curve della famiglia delle *Parabrinidae* si ottengono utilizzando la funzione

$$f = \bigcirc_t \frac{2 \sin 4t}{1+t^2}$$

dell'octobrina in entrambe le coordinate della parametrizzazione:

$$\begin{aligned} \text{P. rotans} \quad x &= (\cos t)f(t) \\ y &= (\sin t^2)f(t) \end{aligned}$$

$$\begin{aligned} \text{P. trichoptera} \quad x &= f(t) \\ y &= tf(t) \end{aligned}$$

$$\begin{aligned} \text{P. muscellaria} \quad x &= (\cos t)f(t) \\ y &= (\sin t)f(t) \end{aligned}$$

$$\begin{aligned} \text{P. composita} \quad x &= f(t) \\ y &= f(t^2) \end{aligned}$$

$$\begin{aligned} \text{P. nuclearis} \quad x &= f(t) \\ y &= f(t-1) \end{aligned}$$

$$\begin{aligned} \text{P. rediens} \quad x &= f(t) \\ y &= f(2t) \end{aligned}$$

$$\begin{aligned} \text{P. faceta} \quad x &= f(t) \\ y &= f(\cos t) \end{aligned}$$

$$\begin{aligned} \text{P. octonaria} \quad x &= f(\cos t) \\ y &= f(\sin t) \end{aligned}$$

$$\begin{aligned} \text{P. fungina} \quad x &= f(\cos t) \\ y &= f(1 + \sin t) \end{aligned}$$

Di queste funzioni solo le ultime due sono periodiche, perciò, nonostante le apparenze, le prime sette curve non sono chiuse e ogni allargamento dell'intervallo di definizione ne cambierebbe l'aspetto, anche se i cambiamenti diventano impercettibili per l'influsso smorzante del fattore $\frac{1}{1+x^2}$ contenuto in f .

Rotazioni

Abbiamo visto a pagina 24 che una rotazione per un angolo α attorno all'origine nel piano corrisponde alla moltiplicazione con $e^{i\alpha}$ nell'ambito dei numeri complessi. Se vogliamo esprimere l'angolo in gradi, dobbiamo moltiplicare l'argomento con $\frac{\pi}{180}$ come visto a pagina 13.

Vogliamo inoltre anche rappresentare rotazioni di un punto z attorno a un centro c . A questo scopo dobbiamo prima ruotare il vettore $z - c$ attorno all'origine e poi riattaccarlo in c . Otteniamo così la seguente funzione in R:

```
# Rotazione di z per alfa (in gradi)
# attorno al centro.
Rot = function (z,alfa,centro=0i)
{centro+exp(1i*alfa*CPid180)*(z-centro)}
```

Come sempre nelle funzioni di R, anche questa può essere applicata usando un vettore invece di un singolo punto come primo argomento z . A pagina 29 definiremo una funzione *Ellisse* che crea un vettore consistente dei punti di un'ellisse. Con

```
Rot(Ellisse(t,3,2),alfa,centro)
```

otteniamo l'ellisse ruotata dell'angolo α attorno al centro.

expression ed eval

R permette di creare *espressioni formali* con la funzione `expression`. Mentre quindi dopo

```
formula=expression(x+3)
```

`formula` è un'espressione formale il cui valore non viene però calcolato (infatti questa espressione è lecita anche quando `x` non è ancora stato definito), con

```
valore=eval(formula)
```

ne possiamo calcolare il valore. Esempio:

```
formula=expression(x+3)
x=10
u=eval(formula)

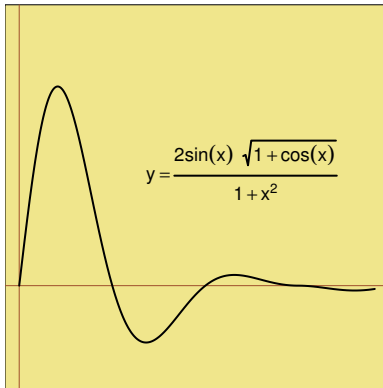
print(formula)
print(u)
```

con output

```
expression(x + 3)
[1] 13
```

Testi matematici

Nelle istruzioni `text`, `legend`, `mtext` e `axis`, che permettono l'aggiunta di testi alle figure, le espressioni matematiche (nel senso dell'articolo precedente) vengono elaborate secondo regole simili a quelle del *Latex* per ottenere le formule matematiche corrispondenti. I dettagli sono forniti da `?plotmath` e `demo(plotmath)`. Esempio:



La figura è stata ottenuta con

```
# 29-plotmath

Postscript("29-plotmath.ps",2,2)
latox=c(0,4); latoy=c(-1,3)
par(cex=0.6,bg="#f0e68c")
Grafica(latox,latoy,cornice=T)
x=seq(0,4,by=0.01)
par(col="sienna")
abline(h=0,v=0)
par(lwd=1,col=1)
lines(x,
      2*sin(3*x)*sqrt(1+cos(x))/(1+x*x))
text(2.5+1.3i,expression(y ==
      frac(2*sin(x)~sqrt(1+cos(x)),1+x^2)))
dev.off()
```

Per ragioni tipografiche abbiamo suddiviso alcune istruzioni su due righe. Si possono anche rappresentare integrali, simboli logici e insiemistici, indici ecc.

Ellissi

Siano $a, b, t \in \mathbb{R}$ con $a, b > 0$ ed $x = a \cos t$, $y = b \sin t$. Allora

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$$

quindi il punto (x, y) si trova su un'ellisse con gli assi a e b . Non è difficile dimostrare (con le conoscenze sulle funzioni trigonometriche e le loro inverse che si imparano nei corsi di Analisi) che viceversa ogni punto dell'ellisse può essere rappresentato in questa forma. In altre parole,

$$\begin{aligned} x &= a \cos t \\ y &= b \sin t \end{aligned}$$

con $0 \leq t \leq 2\pi$ è una parametrizzazione dell'ellisse in forma normale. Per $a = b$ otteniamo un cerchio di raggio a . Possiamo quindi usare la seguente funzione in R:

```
Ellisse = function (t,a,b)
{a*cos(t)+1i*b*sin(t)}
```

Questa funzione restituisce un vettore di punti che, se il vettore dei parametri t ha una risoluzione sufficientemente fine, quando lo disegniamo con `lines` apparirà come una curva liscia continua.

Più concretamente useremo i punti definiti da

```
t=seq(0,2*pi,by=0.01)
```

per disegnare un'ellisse intera, ma possiamo anche limitarci a una parte dell'intervallo con

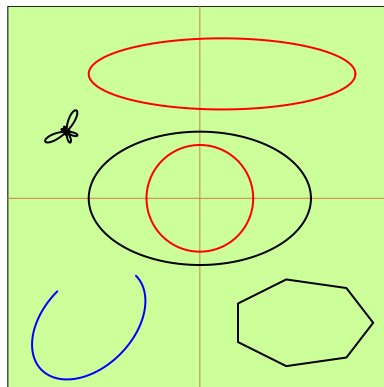
```
t=seq(t1,t2,by=0.01)
```

per ottenere un arco oppure suddividere l'intervallo in n parti, per ottenere un poligono (regolare per $a = b$) ad n vertici:

```
t=seq(0,2*pi,length=7)
```

per un esagono (perché l'ultimo punto lo aggiungiamo per chiudere la curva).

Per spostare il centro è sufficiente usare l'addizione in \mathbb{R}^2 (usando i numeri complessi di \mathbb{R}), per ruotare la figura possiamo utilizzare la funzione `Rot` che abbiamo introdotto a pagina 28. Naturalmente traslazioni e rotazioni possono essere applicate anche alle altre figure che abbiamo definito.



Nel programma ad ogni figura corrispondono pochissime istruzioni:

```
# 29-ellissi
```

```
Postscript("29-ellissi.ps",2,2)
lato=c(-4,4); par(bg="#ccff99")
Grafica(lato,lato,cornice=T)
par(col="peru"); abline(h=0,v=0)
par(lwd=1,col=1); t=seq(-6,6,by=0.01);
lines(-3+1.5i
      +Rot(0.3*ParabrinaMuscellaria(t),45))
t=seq(0,2*pi,by=0.01)
lines(Ellisse(t,2.5,1.5))
par(col="red")
lines(0.5+2.8i+Ellisse(t,3,0.8))
lines(Ellisse(t,1.2,1.2))
par(col="blue")
t=seq(pi/2,2*pi,b=0.01)
lines(-2.5-2.8i
      +Rot(Ellisse(t,1.5,1),45))
par(col=1)
t=seq(0,2*pi,length=8)
lines(2.3-2.8i+Ellisse(t,1.6,1))
dev.off()
```

Iperboli

Siano $a, b, t \in \mathbb{R}$ con $a, b > 0$ ed $x = a \cosh t$, $y = \sinh t$. Allora

$$\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$$

quindi il punto (x, y) si trova su un'iperbole di asse reale a e asse immaginaria b . Si può dimostrare che viceversa ogni punto del ramo destro dell'iperbole può essere rappresentato in questa forma. In altre parole,

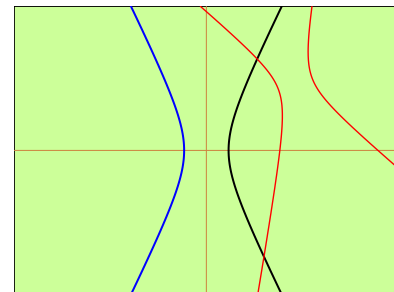
$$\begin{aligned} x &= a \cosh t \\ y &= b \sinh t \end{aligned}$$

con $-\infty < t < \infty$ è una parametrizzazione del ramo destro dell'iperbole in forma normale. Per ottenere il ramo sinistro si può ad esempio sostituire a con $-a$. Per $a = b$ otteniamo un'iperbole equilatera con equazione

$$x^2 - y^2 = a^2$$

Possiamo quindi usare la seguente funzione in R:

```
# Ramo destro di un'iperbole.
# Ramo sinistro con -a.
Iperbole = function (t,a,b)
{a*cosh(t)+1i*b*sinh(t)}
```



A parte i soliti comandi di impostazione il programma contiene le righe

```
par(lwd=1,col=1); t=seq(-9,9,by=0.01);
lines(Iperbole(t,0.5,1))
par(col="blue")
lines(Iperbole(t,-0.5,1))
par(col="red",lwd=0.7)
lines(2+1.5i
      +Rot(Iperbole(t,0.4,0.7),20))
lines(2+1.5i
      +Rot(Iperbole(t,-0.4,0.7),20))
```

Rette e segmenti

Una retta R in \mathbb{R}^n può essere rappresentata nella forma

$$R = \{p_0 + tw \mid t \in \mathbb{R}\} = p_0 + \mathbb{R}w$$

con $p_0, w \in \mathbb{R}^n$ e $w \neq 0$, equivalente alla rappresentazione parametrica

$$x_1 = p_{01} + tw_1$$

...

$$x_n = p_{0n} + tw_n$$

se $w = (w_1, \dots, w_n)$ e $p_0 = (p_{01}, \dots, p_{0n})$. In \mathbb{R}^2 possiamo scrivere

$$R = z_0 + \mathbb{R}w$$

e

$$x = x_0 + tw$$

$$y = y_0 + tw$$

se $w = (u, v)$ e $z_0 = (x_0, y_0)$, oppure, in forma complessa,

$$z = z_0 + tw$$

La retta che congiunge due punti p_0 e p_1 distinti in \mathbb{R}^n è l'insieme $p_0 + \mathbb{R}(p_1 - p_0)$; la retta che congiunge due punti distinti z_0 e z_1 del piano è quindi $z_0 + \mathbb{R}(z_1 - z_0)$. Se i due punti non sono distinti, questo insieme non è una retta, ma contiene soltanto l'unico punto dato.

Se come parametri usiamo solo i valori $t \in [0, 1]$ invece della retta otteniamo il segmento di retta che congiunge i due punti.

Equazione di una retta

Siano $z_0 = (x_0, y_0) \in \mathbb{R}^2$ e $w = (u, v) \in \mathbb{R}^2$ con $w \neq 0$. Allora il vettore $iw = (-v, u)$ è ortogonale alla retta $R := z_0 + \mathbb{R}w$ e un punto z appartiene alla retta se e solo se $z - z_0$ è ortogonale ad iw . Sia $iw = (a, b)$ (cioè $a = -v$ e $b = u$). Allora z appartiene ad R se e solo se x ed y soddisfano l'equazione

$$a(x - x_0) + b(y - y_0) = 0$$

che può essere scritta anche nella forma

$$ax + by = ax_0 + by_0$$

Siccome $w = (b, -a) \neq 0$, anche $(a, b) \neq 0$.

Sia viceversa data un'equazione della forma

$$ax + by = c$$

con $a, b \in \mathbb{R}$ non entrambi uguali a 0. Allora possiamo facilmente trovare $x_0, y_0 \in \mathbb{R}$ tali che $ax_0 + by_0 = c$ e vediamo che l'equazione descrive la retta $z_0 + \mathbb{R}w$ con $z_0 := (x_0, y_0)$ e $w := (b, -a)$.

Per rappresentare rette in R usiamo una funzione che genera i punti della retta tra z_1 e z_2 corrispondenti ai parametri t :

```
Retta = function (t,z1,z2)
{z1+t*(z2-z1)}
```

Anche in questo caso i punti generati possono essere soggetti a traslazioni o rotazioni usando l'addizione tra numeri complessi e la funzione Rot.

Esempio:

```
# Segmento.
t=seq(0,1,by=0.01)
z1=1i; z2=3-1i
lines(Retta(t,z1,z2))

# Retta rotata attorno
# all'origine.
z3=2-1i; z4=4+1i
t=seq(-10,10,by=0.01)
lines(Rot(Retta(t,z3,z4),60))
```

Proiezione su una retta

Siano dati una retta $R = p_0 + \mathbb{R}w$ in \mathbb{R}^n (con $w \neq 0$) e un punto $p \in \mathbb{R}^n$. Vogliamo calcolare la proiezione ortogonale m di p su R . Il punto m deve essere in primo luogo un punto della retta e quindi della forma

$$m = p_0 + tw$$

inoltre il vettore $p - m$ deve essere ortogonale a w , cioè $\|p - m, w\| = 0$, ossia

$$\begin{aligned} \|p, w\| &= \|m, w\| = \|p_0 + tw, w\| \\ &= \|p_0, w\| + t\|w, w\| \end{aligned}$$

Siccome $w \neq 0$, ciò è equivalente a

$$t = \frac{\|p, w\| - \|p_0, w\|}{\|w, w\|}$$

e quindi abbiamo la formula fondamentale

$$t = \frac{\|p - p_0, w\|}{\|w, w\|}$$

Da essa otteniamo la proiezione con

$$m = p_0 + \frac{\|p - p_0, w\|}{\|w, w\|} w$$

Quando $|w| = 1$ (ciò si può sempre ottenere sostituendo w con $\frac{w}{|w|}$), abbiamo la rappresentazione

$$m = p_0 + \|p - p_0, w\| w$$

Dalla derivazione si vede anche che t e con esso m sono univocamente determinati. La distanza di p dalla retta è uguale a $|p - m|$. Tutto ciò è valido in \mathbb{R}^n per ogni n .

È geometricamente chiaro e facile da dimostrare che $m = p$ se p è un punto della retta (esercizio 48).

Definiamo in R una funzione per il calcolo del prodotto scalare di due punti del piano,

```
# Prodotto scalare di due
# punti del piano.
Scalare = function (z,w)
{Re(z*Conj(w))}
```

in cui abbiamo usato la relazione

$$\|z, w\| = \text{Re } z\bar{w}$$

dell'esercizio 30 e la funzione Conj che in R corrisponde alla formazione del complesso coniugato, ottenendo così la funzione per il calcolo della proiezione di un punto su una retta nel piano:

```
# Proiezione di z sulla retta z0+tw.
Proi = function (z,z0,w)
{t=Scalare(z-z0,w)/Scalare(w,w)
z0+t*w}
```

Il sito CRAN di Ferrara

A Ferrara opera un gruppo di biologi che lavora con R (soprattutto nel campo dei *microarrays*) e gestisce anche il deposito italiano del CRAN. Gli indirizzi sono

microarrays.unife.it/CRAN
biotecnologie.unife.it/

Esercizi per gli scritti

Quando è indicato solo una figura, il compito consiste nell'ottenere quella figura con un programma in R.

44. Octobrina montuosa e Octobrina voragiosa si distinguono apparentemente soprattutto nella parte centrale. È vero che per $|x|$ grande hanno un andamento simile?

45. Trovare l'equazione della retta che congiunge i punti $(-2, -1)$ e $(3, 2)$.

46. Trovare, con il metodo indicato su questa pagina, l'equazione della retta che congiunge i punti $(4, -1)$ e $(4, 10)$.

47. Rappresentare la retta con equazione $3x + 10y = 7$ in forma parametrica.

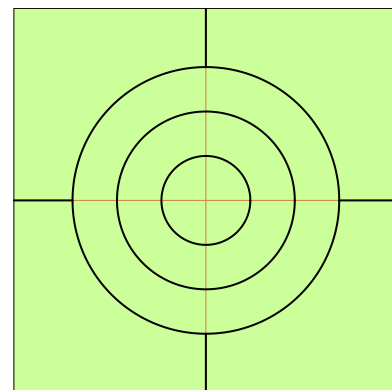
48. R sia una retta in \mathbb{R}^n e p un punto che appartiene a questa retta. Allora la proiezione ortogonale di p su R coincide con p .

49. Calcolare la proiezione ortogonale di $(2, 5)$ sulla retta $(1, 3) + \mathbb{R}w$ con $w = (4, 1)$.

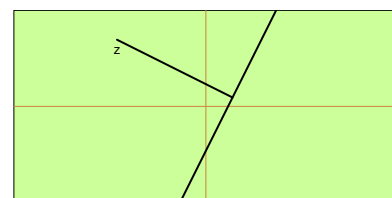
50. Calcolare in \mathbb{R}^3 la proiezione ortogonale di $(1, 0, 3)$ sulla retta $(1, 5, 2) + \mathbb{R}w$ con $w = (4, 2, 1)$.

51. Calcolare la proiezione ortogonale di $(4, 1)$ sulla retta che congiunge i punti $(1, 6)$ e $(2, 3)$.

52. Usare due for.



53. La retta è $y = -1 + 2x$ e z è $-2 + 1.5i$.



Curve di livello

Curve di livello di una funzione in due variabili f non sono altro che gli insiemi della forma $f = k$, dove k è una costante. In particolare per k si ottengono gli zeri di f ; spesso però è interessante guardare come variano queste curve con il variare di k e soprattutto in quel caso, cioè quando si disegnano le curve ($f = k$) (con la notazione della def. 7.1) contemporaneamente per più valori di k , in modo simile come in un atlante si disegnano curve di punti con la stessa altezza o con la stessa quantità di precipitazioni o con la stessa temperatura media ecc., si parla di curve di livello.

In R a questo scopo si può usare la funzione `contour` che, nella versione più semplice, si usa come nella funzione `Livelli` che creiamo per la nostra libreria:

```
Livelli = function (x,y,f,
  valori=NULL,livelli=0,
  scritte=1,colore=1,spessore=0.4)
{if (identical(valori,NULL))
  valori=outer(x,y,f);
  contour(x,y,valori,levels=livelli,
  col=colore,lwd=spessore,
  labcex=0.6*scritte,
  drawlabels=(scritte>0),add=T)}
```

Il significato dei parametri `colore` e `spessore` (della matita) è chiaro. Il parametro `scritte` riguarda i parametri `drawlabels` e `labcex` della funzione originale, di cui il primo, se posto uguale a `TRUE`, implica che sulle linee di livello vengono indicati i valori che corrispondono a quei livelli, mentre `labcex` indica la grandezza delle scritte, di cui abbiamo bisogno anche noi perché il valore preimpostato per la visualizzazione sullo schermo risulta troppo grande per le immagini che otteniamo con `Postscript`; per queste ultime sceglieremo quindi `scritte=0.5`. Si noti che nella nostra funzione `drawlabels` è definito attraverso l'espressione booleana `(scritte>0)`; per ottenere una figura senza scritte sulle curve di livello useremo `scritte=0`.

Nella funzione `contour` abbiamo posto l'argomento `add` uguale a `T`, perché in questo modo possiamo eseguire più volte il comando `contour` (o `Livelli`) sulla stessa

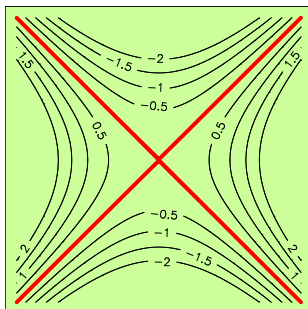
grafica senza cancellare il contenuto precedente; `labcex` riguarda la grandezza delle scritte con cui vengono indicati i valori dei livelli.

`livelli` (nell'originale `levels`) è un singolo numero o un vettore di numeri che rappresenta l'insieme dei livelli che vogliamo disegnare; con `livelli=0` (scelta prestabilita in `Livelli`) viene mostrato l'insieme degli zeri di f , con `livelli=c(0,1,2)` gli insiemi ($f = 0$), ($f = 1$) e ($f = 2$).

La nostra funzione `Livelli` si basa sulla funzione `contour` di R di cui dobbiamo ancora spiegare i primi tre parametri. x e y sono vettori di numeri che corrispondono matematicamente a due successioni finite (x_1, \dots, x_m) e (y_1, \dots, y_n) . Il terzo argomento (il nostro `valori`) deve essere una matrice

$$\begin{pmatrix} v_{11} & \dots & v_{1m} \\ \dots & \dots & \dots \\ v_{n1} & \dots & v_{nm} \end{pmatrix}$$

di n righe ed m colonne; il coefficiente v_{ij} è il valore nel punto (x_j, y_i) della funzione di cui vogliamo disegnare i livelli. Questa matrice è il risultato di un comando `outer(x,y,f)`. La funzione `Livelli` è stata predisposta per essere utilizzata in due modi: possiamo o indicare la funzione f (e allora `outer` viene chiamato all'interno di `Livelli`), oppure usare come parametro `valori` una matrice generata con un `outer` precedente; la seconda variante verrà usata quando vogliamo utilizzare la stessa matrice per più esecuzioni di `Livelli`, ad esempio se ogni volta vogliamo colorare le curve di livello in modo diverso.



In questo numero

- 31 Curve di livello
 $x^2 - y^2 = k$
- 32 outer
ifelse
Una collana
NA
- 33 if... else
identical
Operazioni insiemistiche
dimnames
 $y^2 = x^3 + x^2$
 $y^2 = x^3$
 $y^2 = x^3 - x^2$
- 34 $y^2 = x^3 + x$
 $y^2 = x^3 - x$
Il foglio di Cartesio
La chiocciola di Pascal
La lemniscata
Una curva trascendente
Esercizi 54-58

$x^2 - y^2 = k$

Come sappiamo per ogni $k > 0$ questa equazione rappresenta un'iperbole equilatera con asse uguale all'asse delle x . Per $k < 0$ invece abbiamo un'equazione dello stesso tipo con x ed y invertite, che quindi corrisponde a un'iperbole con asse uguale all'asse delle y .

Per $k = 0$ infine l'equazione diventa

$$x^2 - y^2 = (x - y)(x + y) = 0$$

e corrisponde alle due rette $y = x$ ed $y = -x$.

Otteniamo la figura su questa pagina che mostra le curve di livello della funzione definita da $f(x, y) = x^2 - y^2$ con questo programma in R:

```
# 31-iperboli

Postscript("31-iperboli.ps",1.6,1.6)
lato=c(-2,2); par(bg="#ccff99")
Grafica(lato,lato,cornice=T)
t=seq(-2,2,by=0.01)
f = function (x,y) x*x-y*y
valori=outer(t,t,f)
Livelli(t,t,valori=valori,scritte=0.5,
  livelli=setdiff(seq(-2,2,by=0.5),0))
Livelli(t,t,valori=valori,spessore=2,
  colore="red",scritte=0)
dev.off()
```

L'insieme dei punti su cui avviene l'analisi dei livelli è dato dal quadrato $[-2, 2] \times [-2, 2]$ suddiviso in ogni coordinata con una risoluzione di 0.01, come espresso dall'istruzione

```
t=seq(-2,2,by=0.01)
```

Usiamo la funzione `Livelli` due volte. Infatti la prima volta disegniamo in nero e con le scritte tutti i livelli diversi da 0, la seconda volta solo le due rette $y = \pm x$, in rosso e senza scritte.

Nella prima istruzione i livelli scelti corrispondono a

$$k \in \{-2, -1.5, -1, -0.5, 0.5, 1, 1.5, 2\}$$

Nel programma questo insieme è definito come differenza insiemistica (pagina 33)

```
setdiff(seq(-2,2,by=0.5),0)
```

outer

Questa funzione crea da due successioni finite (x_1, \dots, x_m) e (y_1, \dots, y_n) e una funzione f la matrice

$$\begin{pmatrix} f(x_1, y_1) & \dots & f(x_m, y_1) \\ \vdots & \ddots & \vdots \\ f(x_1, y_n) & \dots & f(x_m, y_n) \end{pmatrix}$$

In R a ciò corrisponde l'espressione `outer(x,y,f)`.

`outer` può essere usata in molti modi, non solo per impostare la matrice dei valori per un comando `contour` o `livelli`, ma ad esempio anche per creare la tabella di moltiplicazione per un'operazione binaria. Consideriamo l'insieme

$$\mathbb{Z}/6 := \{0, 1, 2, 3, 4, 5\}$$

con l'addizione modulo 6; ciò significa che introduciamo per gli elementi di questo insieme un'operazione \oplus definita da

$$a \oplus b := (a + b) \bmod 6$$

dove $n \bmod 6$ è il resto di n modulo 6.

Per vedere tutti i risultati che otteniamo in questo modo definiamo la tabella di moltiplicazione (in questo caso la tabella delle somme modulo 6) con le seguenti istruzioni in R:

```
smod6 = function (x,y) (x+y)%6
x=seq(0,5)
tabella=outer(x,x,smod6)
dimnames(tabella)=list(0:5,0:5)
print(tabella)
```

con output

```
0 1 2 3 4 5
0 0 1 2 3 4 5
1 1 2 3 4 5 0
2 2 3 4 5 0 1
3 3 4 5 0 1 2
4 4 5 0 1 2 3
5 5 0 1 2 3 4
```

Nello stesso modo possiamo definire un prodotto modulo 6, ponendo

$$a \odot b := (ab) \bmod 6$$

In R allora usiamo la funzione

```
pmod6 = function (x,y) (x*y)%6
```

al posto di `smod`. Otteniamo per questa operazione la tabella di moltiplicazione

```
0 1 2 3 4 5
0 0 0 0 0 0
1 0 1 2 3 4 5
2 0 2 4 0 2 4
3 0 3 0 3 0 3
4 0 4 2 0 4 2
5 0 5 4 3 2 1
```

Nel corso di Algebra si dimostra che $(\mathbb{Z}/6, \oplus)$ è un gruppo abeliano, $(\mathbb{Z}/6, \odot)$ è un semigrupp commutativo con unità e che le due operazioni sono legate dalla legge distributiva

$$(a \oplus b) \odot c = (a \odot c) \oplus (b \odot c)$$

La tripla $(\mathbb{Z}/6, \oplus, \odot)$ è quindi un anello commutativo con unità. Se guardiamo bene la seconda tabella, vediamo però che

$$2 \odot 3 = 0$$

benché 2 e 3 siano diversi da 0. Questo anello possiede perciò divisori di zero.

ifelse

x sia un vettore di valori booleani o di oggetti convertibili a valori booleani, a e b due vettori qualsiasi, non necessariamente della stessa lunghezza di x . Allora

```
ifelse(x,a,b)
```

è il vettore che si ottiene sostituendo in x ogni valore vero con un valore di a e ogni valore falso con un valore di b , percorrendo (in maniera ciclica, quando necessario) i vettori a e b .

`ifelse` è effettivamente una funzione che può generare delle configurazioni piuttosto complicate e matematicamente interessanti. Assumiamo che vogliamo generare una successione della forma

$$(a_1, b_2, a_3, b_4, a_5, b_6, a_7, b_8, a_9, b_{10}, a_{11}, b_{12})$$

Allora possiamo procedere come in questo esempio:

```
x=1:12
a=c(100,NA,200,NA,300,NA);
b=c(NA,1,NA,2)
u=ifelse(x%%2,a,b)
cat(u,'\n')
```

con output

```
100 1 200 2 300 1 100 2 200 1 300 2
```

Infatti, siccome i vettori a e b vengono ripetuti ciclicamente, abbiamo questa situazione:

x	a	b
1	100	NA
2	NA	1
3	200	NA
4	NA	2
5	300	NA
6	NA	1
7	100	NA
8	NA	2
9	200	NA
10	NA	1
11	300	NA
12	NA	2

Secondo il programma, per un valore dispari nella colonna x viene scelto il valore da a , per un valore pari un valore da b . Traducendo questa idea in una realizzazione geometrica, otteniamo la collana nella figura.

a e b possono anche consistere di valori singoli, come in

```
x=c(2,5,3,1,2,3,4,7)
u=ifelse(x%%2,"dispari","pari")
for (v in u) cat(v,'\n')
```

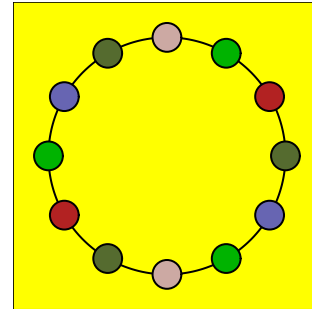
con output

```
pari
dispari
dispari
dispari
pari
dispari
pari
dispari
```

`ifelse` (nella sua forma non vettoriale) è una funzione molto importante anche in informatica teorica e nella teoria dei circuiti digitali, dove appare nella descrizione di *funzioni booleane* (cioè funzioni $\{0,1\}^n \rightarrow \{0,1\}$) mediante *diagrammi binari di decisione*. Invece di `ifelse(x,a,b)` si scrive allora spesso $[x, a, b]$. Questo operatore ternario esiste anche in C dove viene scritto nella forma $x ? a : b$.

Una collana

Abbiamo osservato che `ifelse` è una funzione da cui si possono far derivare configurazioni molto interessanti. Con la stessa idea vista prima otteniamo questa collana:



Nel seguente programma in R appare il termine `u[k]` con cui si denota il k -esimo elemento del vettore u . A differenza dal C in R si inizia a contare da 1.

```
# 32-configurazione
```

```
Postscript("32-configurazione.ps",1.6,1.6)
lato=c(-1.2,1.2); par(bg="yellow")
Grafica(lato,lato, cornice=T)
t=seq(0,2*pi,by=0.01)
cerchio=cos(t)+1i*sin(t)
par(lwd=1,col=1)
lines(cerchio)
a=c("#b22222",NA,"#cca9a3",NA,
"#6765b2",NA)
b=c(NA,"#00b300",NA,"#556b2f")
v=1:12; u=ifelse(v%%2,a,b)
for (k in v) {alfa=k*pi/6;
points(exp(alfa*1i),pch=21,
bg=u[k],cex=2)}
dev.off()
```

La funzione `points` è stata discussa a pagina 22; per disegnare cerchi pieni colorati si può usare anche il comando `polygon`, come vedremo fra poco.

NA

In statistica accade molto spesso che serie di misurazioni contengano dati non validi o non disponibili. A questo corrisponde il valore `NA` (un'abbreviazione per *not available*); l'abbiamo già incontrato a pagina 22. La funzione `sqrt` che calcola la radice quadrata reale di un numero reale x non è definita per $x < 0$, anche se nel campo complesso x possiede le due radici $\sqrt{-x}i$ e $-\sqrt{-x}i$ (ad esempio le radici quadrate di -3 sono $\sqrt{3}i$ e $-\sqrt{3}i$); ciò è un caso speciale di quanto visto a pagina 25. Se è dato un vettore x di numeri reali, non necessariamente tutti positivi, di cui, se sono positivi, vogliamo calcolare le radici quadrate, possiamo usare `ifelse` per convertire tutti i valori negativi in `NA`:

```
x=c(-1,2,0,-3,-4,5)
x=ifelse(x>0,x,NA)
u=sqrt(x)
for (v in u) cat(v,'\n')
```

con output

```
NA
1.414214
0
1.732051
NA
2.236068
```

if ... else

All'interno di una funzione o di un blocco tra parentesi graffe l'if di R viene usato nelle forme

```
if (condizione) istruzione
```

oppure

```
if (condizione) istruzione1
else istruzione2
```

Quando il comando complessivamente occupa più righe e non si trova all'interno di un blocco o di una funzione, bisogna racchiuderlo tra parentesi graffe.

Abbiamo usato la prima forma (un if senza else) nella definizione della funzione Livelli a pagina 31.

identical

In R il test per uguaglianza di oggetti dovrebbe avvenire mediante la funzione identical. Gli operatori == e != sono definiti in modo un po' diverso da come uno se lo aspetterebbe, infatti vengono usati soprattutto per il confronto elemento per elemento di vettori come in

```
a=c(1,2,3,4,5,6)
b=c(1,0,8,4,3,6)
u=(a==b)
print(u)
```

con output

```
TRUE FALSE FALSE TRUE FALSE TRUE
```

e

```
a=c(1,2,3,4,5,6)
b=c(1,0,8,4,3,6)
u=(a!=b)
print(u)
```

con output

```
FALSE TRUE TRUE FALSE TRUE FALSE
```

Infatti nello stesso modo vettoriale funzionano anche gli operatori per il confronto numerico:

```
a=c(1,2,3,4,5,6)
b=c(1,0,8,4,3,6)
u=(a>b)
print(u)
```

con output

```
FALSE TRUE FALSE FALSE TRUE FALSE
```

Operazioni insiemistiche

R contiene alcune semplici, ma potenti funzioni insiemistiche:

```
is.element(x,A) ... (x ∈ A)
union(A,B) ... A ∪ B
intersect(A,B) ... A ∩ B
setdiff(A,B) ... A \ B
setequal(A,B) ... (A = B)
```

Insiemi vengono rappresentati da vettori, astruendo dall'ordine degli elementi e da apparizioni molteplici dello stesso valore.

Abbiamo usato la funzione setdiff a pagina 31 per togliere il valore 0 dall'insieme dei livelli da disegnare nella prima esecuzione di Livelli.

dimnames

Nelle tabelle di moltiplicazione a pagina 32 abbiamo usato la funzione dimnames. Essa è utile per la visualizzazione di tabelle; altrimenti nella riga dei titoli e nella colonna dei nomi delle righe apparirebbero solo gli indici. Infatti con

```
a=matrix(c(2248,20000,1587,2580,
411,8900),nrow=3)
print(a)
```

otteniamo l'output

```
[,1] [,2]
[1,] 2248 2580
[2,] 20000 411
[3,] 1587 8900
```

che diventa più leggibile se aggiungiamo i significati dei numeri:

```
a=matrix(c(2248,20000,1587,2580,
411,8900),nrow=3)
dimnames(a)=list(c("Arabia saud.",
"Mongolia","Svezia"),
c("sup/kmq","ab/1000"))
print(a)
```

con output

	sup/kmq	ab/1000
Arabia saud.	2248	2580
Mongolia	20000	411
Svezia	1587	8900

Si noti che nella lista vengono prima indicati i nomi delle righe, poi quelli delle colonne.

$y^2 = x^3 + x^2$

Con lo stesso programma che a pagina 31 abbiamo usato per disegnare le iperboli, sostituendo solo la riga riguardante la funzione con

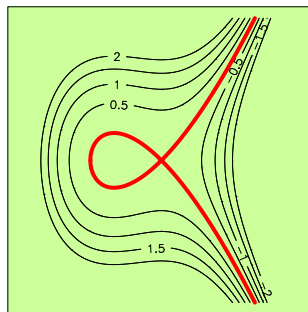
```
f = function(x,y) y*y-x*x*x-x*x
```

possiamo disegnare la famiglia di curve le cui equazioni hanno la forma

$$y^2 = x^3 + x^2 + k$$

In questo caso la curva singolare, in rosso, è la curva

$$y^2 = x^3 + x^2$$

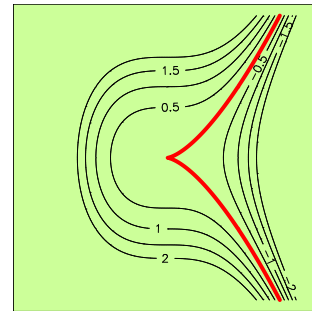


I livelli indicati sono i livelli della funzione $f := \bigcirc_{(x,y)} y^2 - x^3 - x^2$. Si vede che la f assume valori positivi alla destra della curva singolare e valori negativi alla sua sinistra.

Provare da soli, eventualmente ingrandendo la figura, a scoprire il segno di $f(x,y)$ all'interno del nodo.

$y^2 = x^3$

Questa curva è la cuspidale o parabola di Neill:

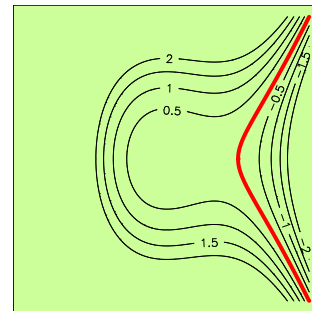


$y^2 = x^3 - x^2$

Purtroppo talvolta il software per la rappresentazione di curve può ingannare, e ciò non vale solo per R, ma anche per Maple, Mathematica ecc. In questo caso per esempio è chiaro che l'origine è una soluzione dell'equazione

$$y^2 = x^3 - x^2$$

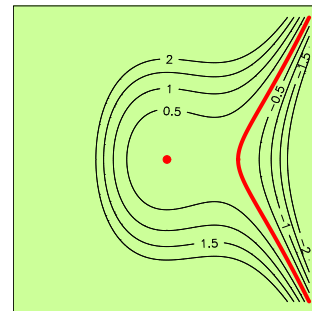
eppure non appare nella figura (dovrebbe essere un punto rosso nell'origine). Quindi non ci si può fidare ciecamente dei disegni che si ottengono con questi programmi.



Non siamo riusciti ad ottenere l'origine come punto della curva mediante il programma; dobbiamo quindi, appellandoci ai risultati della matematica, aggiungerlo a mano:

```
...
f = function(x,y) y*y-x*x*x+x*x
...
points(0,0,pch=20,cex=0.8,col="red")
dev.off()
```

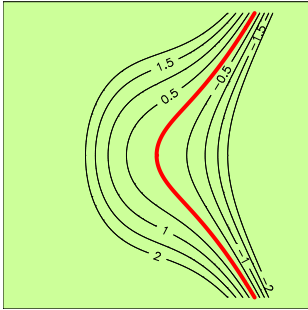
ottenendo così



In questo caso potrebbe anche trattarsi di un difetto della funzione contour (o di lines, se contour usa lines) che non sembra in grado di disegnare un punto isolato.

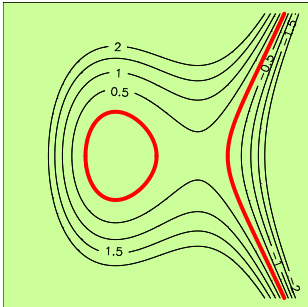
$$y^2 = x^3 + x$$

Anche la curva $y^2 = x^3 + x$ la possiamo ottenere cambiando semplicemente la funzione nel nostro programma.



$$y^2 = x^3 - x$$

Vediamo ancora che una semplice modificazione dell'equazione implica un cambiamento sostanziale nella forma della curva. Alcune differenze scompaiono comunque se si considerano queste curve come curve definite sul campo dei numeri complessi. Come una retta complessa corrisponde a un piano reale, così una curva complessa corrisponde a una superficie reale. Completate nel senso della geometria proiettiva le cubiche finora viste diventano *superficie di Riemann*, la cui teoria è ancora oggi un campo attivo della ricerca.

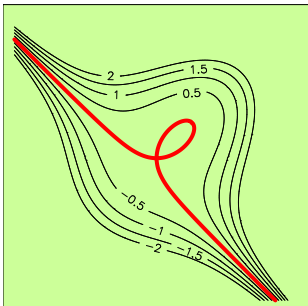


Nelle applicazioni, ad esempio in statistica, ancora più importanti sono però proprio le forme reali, la cui classificazione è molto difficile.

Il foglio di Cartesio

Questa curva ha l'equazione

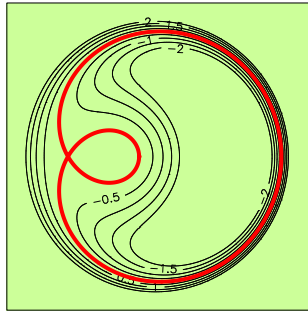
$$x^3 + y^3 - xy = 0$$



La chiocciola di Pascal

La chiocciola di Pascal ha l'equazione

$$(x^2 + y^2 - 2x)^2 = x^2 + y^2$$



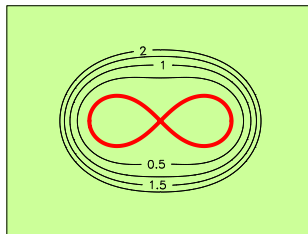
Il programma è stato leggermente modificato:

```
# 34-pascal
Postscript("34-pascal.ps",1.6,1.6)
latox=c(-0.7,3.3); latoy=c(-2,2);
par(bg="#ccff99")
Grafica(latox,latoy,cornice=T)
x=seq(-0.7,3.3,by=0.01)
y=seq(-2,2,by=0.01)
f = function (x,y)
{u=x^2+y^2-2*x; u^2-x^2-y^2}
valori=outer(x,y,f)
Livelli(x,y,valori=valori,scritte=0.5,
  livelli=setdiff(seq(-2,2,by=0.5),0))
Livelli(x,y,valori=valori,spessore=2,
  colore="red",scritte=0)
dev.off()
```

La lemniscata

La lemniscata di Bernoulli è una curva ad otto con equazione

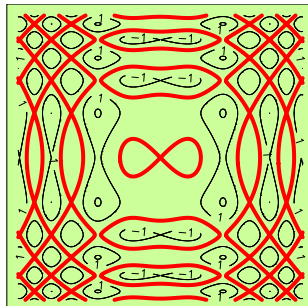
$$(x^2 + y^2)^2 = x^2 - y^2$$



Una curva trascendente

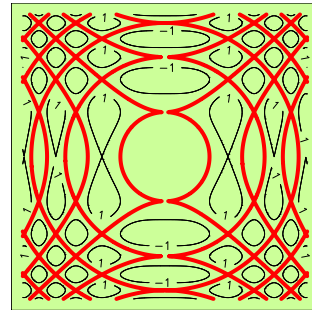
Consideriamo la funzione

$$f(x, y) = \sin y^2 - \cos x^2 + e^{-x^2}$$



Il termine e^{-x^2} converge velocemente a zero quando x diventa più grande; la figura esprime quindi fortemente la periodicità di seno e coseno.

Se togliamo però questo termine, la curva diventa più regolare, ma forse meno bella:



Esercizi per gli scritti

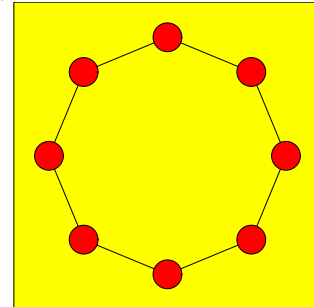
54. Tabella di addizione e di moltiplicazione modulo 12. Ci sono divisori di zero in $(\mathbb{Z}/12, \oplus, \odot)$?

55. Tabella di addizione e di moltiplicazione modulo 5. Ci sono divisori di zero in $(\mathbb{Z}/5, \oplus, \odot)$?

56. Usare ifelse per ottenere l'output

1 2 3 a 5 6 7 b 9 10 11 c

57.

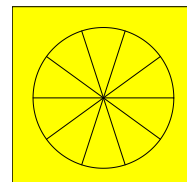


Sono indicate le prime 4 lettere di ogni riga del programma:

```
lato
Graf
t=se
poli
line
t=se (*)
poli
poin
loca
dev.
```

La riga (*) serve per non ridisegnare il primo dei cerchi piccoli. Come si fa?

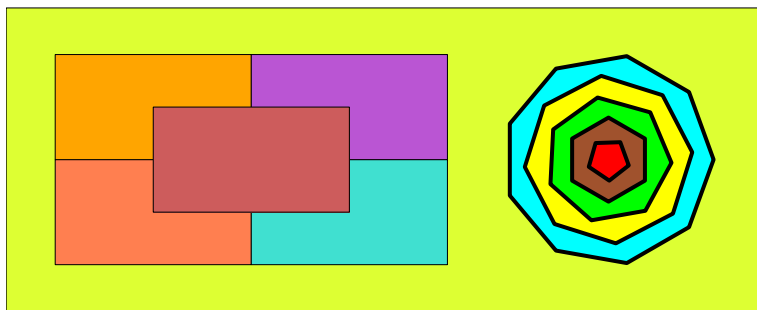
58.



La funzione *polygon*

La funzione *polygon* è molto simile a *lines*; anch'essa unisce una successione di punti, chiudendola però, mentre per ottenere un poligono chiuso con *lines* dobbiamo aggiungere il punto di partenza alla successione. *polygon* permette inoltre di colorare l'interno del poligono (oppure di tratteggiarlo in vari modi). Useremo soprattutto questa possibilità per disegnare poligoni colorati, e quindi anche rettangoli, cerchi ed ellissi (che otteniamo scegliendo successioni di punti molto ravvicinati). Come *lines* anche *polygon* permette l'uso di linee interrotte impostando il parametro *lty*; vedere ?par.

I parametri per definire i colori sono *border* per il bordo e *col* per il colore di riempimento.



Il file *Figure*

Abbiamo visto, quando abbiamo definito le funzioni *Ellisse*, *Iperbole* e *Retta* (pagine 29-30), che in un approccio matematico alla grafica una strategia molto efficiente e versatile è quella di creare le figure di base come insiemi di punti e di separare da esse le operazioni di disegno. Useremo da ora in avanti questo modo di procedere, eliminando quindi le funzioni *Cerchi* e *Rettangoli* definite a pagina 22 dalla nostra libreria; inseriremo le figure di base in un file *Figure* della libreria.

Definiamo adesso due funzioni per la creazione di rettangoli e di poligoni, funzioni che riassumono semplicemente alcune costruzioni già viste.

```
# c(bsin,bdes,ades,asin) di
# un rettangolo di larghezza dx
# e altezza dy.
Rettangolo = function (dx,dy,
  ades=NULL, asin=NULL,
  bdes=NULL, bsin=NULL, centro=0)
{if (!is.null(asin)) {ades=asin+dx;
bsin=asin-li*dy; bdes=bsin+dx} else
if (!is.null(ades)) {asin=ades-dx;
bsin=asin-li*dy; bdes=bsin+dx} else
if (!is.null(bsin)) {asin=bsin+li*dy;
ades=asin+dx; bdes=bsin+dx} else
if (!is.null(bdes)) {bsin=bdes-dx;
asin=bsin+li*dy; ades=asin+dx} else
{asin=Re(centro)-
  dx/2+li*(Im(centro)+dy/2);
ades=asin+dx;
bsin=asin-li*dy; bdes=bsin+dx}
c(bsin,bdes,ades,asin)}
```

In questa funzione i parametri *dx* e *dy* indicano la larghezza e l'altezza del rettangolo; si può inoltre impostare, a scelta, il centro, il punto in alto a destra o a sinistra, il punto in basso a destra o a sinistra. Il risultato della funzione è il vettore che consiste dei quattro vertici del rettangolo

nell'ordine dato dall'ultima riga del listato. Fare un disegno e verificare l'algoritmo.

Abbiamo usato la condizione *is.null(x)* al posto di *identical(x,NULL)* utilizzato a pagina 31.

Più semplice è la funzione per i poligoni:

```
# Vertici di un n-gono regolare
# di raggio r.
Poligono = function (n,r)
{t=seq(0,2*pi,length=n+1)
Cerchio(t,r)}
```

dove abbiamo introdotto la funzione

```
Cerchio = function (t,r)
{Ellisse(t,r,r)}
```

Otteniamo la figura in alto con

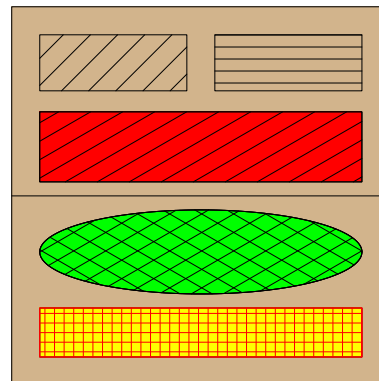
```
Postscript("35-poligoni.ps",10,4)
latox=c(0,10); latoy=c(0,4);
par(bg="#dfff33")
Grafica(latox,latoy,cornice=T)
centro=2.5+2i;
r=Rettangolo(ades=centro,2,1.5)
polygon(r,col="coral")
r=Rettangolo(bdes=centro,2,1.5)
polygon(r,col="orange")
r=Rettangolo(asin=centro,2,1.5)
polygon(r,col="turquoise")
r=Rettangolo(bsin=centro,2,1.5)
polygon(r,col="#ba55d3")
r=Rettangolo(centro=centro,2,1.5)
polygon(r,col="#cd5c5c")
centro=6.5+2i; j=1; r=1.5; alfa=0;
colori=c("cyan","yellow","green",
"sienna","red")
for (n in 9:5) {p=Poligono(n,r);
polygon(Rot(p,alfa)+centro,
  lwd=2,col=colori[j])
r=r-0.3; j=j+1; alfa=alfa+50}
dev.off()
```

In questo numero

- 35 La funzione *polygon*
Il file *Figure*
Tratteggi
- 36 Riflessione in un punto
Riflessione in una retta
Perturbazione dei coefficienti
Una modifica in *Postscript*
- 37 I cammini delle radici
Il teorema di Rouché
Continuità delle radici
- 38 Creazione di matrici
Parallele
Come nasce una forma
Esercizi 59-66

Tratteggi

Per tratteggiare l'interno di un poligono dobbiamo usare i parametri *density* e *angle* di *polygon*. Una densità maggiore corrisponde a un numero maggiore di linee; *angle* imposta l'angolo di inclinazione ed è inizialmente uguale a 45° . Nel poligono tratteggiato il parametro *col* diventa il colore delle linee di tratteggio. Per tratteggi multipli o per colorare allo stesso tempo l'interno del poligono si possono impiegare più istruzioni *polygon* sullo stesso insieme di punti.



Otteniamo la figura con

```
Postscript("35-tratteggi.ps",5,5)
lato=c(0,5); par(bg="tan");
Grafica(lato,lato,cornice=T)

abline(h=2.5)

r=Rettangolo(asin=0.2+4.8i,2,1,0.8)
polygon(r,density=10)
r=r+2.5; polygon(r,density=16,angle=0)

r=Rettangolo(bsin=0.2+2.7i,4,6,1)
polygon(r,col="red")
polygon(r,density=12,angle=30)

t=seq(0,2*pi,by=0.01)
e=Ellisse(t,2,3,0.6)+2.5+1.7i;
polygon(e,col="green")
polygon(e,density=12,angle=30)
polygon(e,density=12,angle=150)

r=Rettangolo(bsin=0.2+0.2i,4,6,0.7)
polygon(r,col="yellow")
polygon(r,density=20,angle=0,col="red")
polygon(r,density=20,angle=90,col="red")
dev.off()
```


Riflessione in un punto

Siano p ed m due punti in \mathbb{R}^n . Diciamo che un punto p' è la riflessione di p in m , se m è il punto di dimezzamento del segmento tra p e p' , cioè se

$$m = \frac{p + p'}{2}$$

Ciò implica che p' è univocamente determinato con

$$p' = 2m - p$$

Possiamo quindi definire una funzione in R che realizza questa operazione per punti del piano:

```
# Riflessione di z nel punto m.
Riflpunto = function (z,m)
{2*m-z}
```

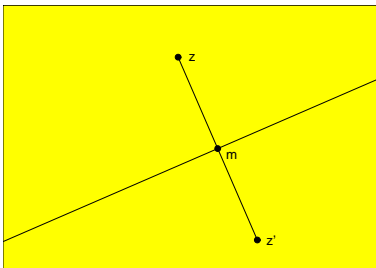
Nel piano in verità la riflessione in m coincide con la rotazione per 180 grado attorno ad m , ma la formula $p' = 2m - p$ è più semplice e valida in ogni \mathbb{R}^n .

Riflessione in una retta

Siano p un punto ed R una retta in \mathbb{R}^n . Diciamo che un punto p' è la riflessione di p in R , se p' è la riflessione di p nella proiezione m di p sulla retta (cfr. pagina 30).

Nel piano possiamo facilmente combinare le funzioni `Proi` e `Riflpunto` per realizzare questa operazione in R:

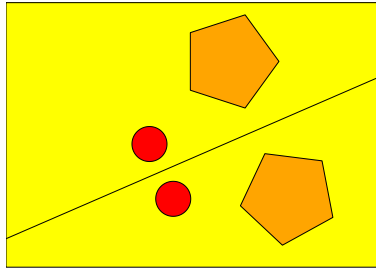
```
# Riflessione di z nella retta z0+tw.
Riflretta = function (z,z0,w)
{Riflpunto(z,Proi(z,z0,w))}
```



Possiamo utilizzare questa funzione anche a un insieme di punti, cioè a una figura, ad esempio ai punti di un cerchio o di un poligono:

```
Postscript("36-poligonoriflesso.ps",
5,3.5)
latox=c(0,5); latoy=c(0,3.5)
par(bg="yellow")
Grafica(latox,latoy,cornice=T)
w=3+1.3i; z0=0.8+0.7i;
t=seq(-2,3,by=0.01)
r=Retta(t,z0,z0+w); lines(r)
t=seq(0,2*pi,by=0.01)
e=Cerchio(t,0.25)+1.85+1.62i
polygon(e,col="red")
er=Riflretta(e,z0,w)
polygon(er,col="red")
p=Poligono(5,0.7)+3+2.8i
polygon(p,col="orange")
pr=Riflretta(p,z0,w)
polygon(pr,col="orange")
dev.off()
```

Otteniamo l'immagine in alto nella colonna accanto.



Perturbazione dei coefficienti

Abbiamo usato la funzione `polyroot` già a pagina 25. Il suo unico argomento è il vettore dei coefficienti (reali o complessi) di un polinomio, elencati iniziando con il coefficiente costante. Se i coefficienti delle potenze più alte sono uguali a zero, vengono scartate. Il risultato è un vettore che rappresenta le radici complesse del polinomio con le loro molteplicità. Abbiamo anche usato la funzione `round` per arrotondare i numeri che otteniamo.

Siccome il risultato di `polyroot` è un vettore di numeri complessi, è facile rappresentare le radici di un polinomio con i nostri strumenti; possiamo ad esempio disegnare un piccolo cerchio in ogni radice.

Vogliamo adesso studiare il comportamento delle radici quando i coefficienti del polinomio vengono modificati.

Consideriamo prima il polinomio

$$f = x^3 + x^2 + 1$$

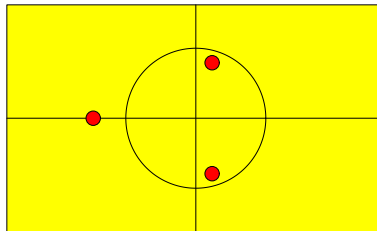
Otteniamo le sue radici, se sul terminale di R battiamo

```
f=c(1,0,1,1)
rad=polyroot(f)
round(rad,2)
```

con output

```
0.23+0.79i 0.23-0.79i -1.47+0.00i
```

Ciò significa che la terza radice è reale, mentre le altre due sono l'una la coniugata complessa dell'altra. Nella rappresentazione grafica otteniamo



Consideriamo adesso il polinomio

$$g = x^3 + x^2 + x + 1$$

in cui appare anche x ; infatti

$$g = f + x$$

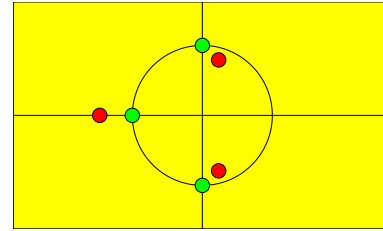
Se calcoliamo nello stesso modo le radici di g , otteniamo la risposta

```
0+1i -1+0i 0-1i
```

Infatti

$$g = (x^2 + 1)(x + 1) = (x - i)(x + i)(x + 1)$$

Dalla rappresentazione grafica, in cui le radici di g corrispondono ai cerchietti verdi, vediamo che esse non sono troppo lontane da quelle di f :



La figura è stata ottenuta con

```
Postscript("36-polyroot2.ps",5,3)
latox=c(-2.5,2.5); latoy=c(-1.5,1.5)
par(bg="yellow")
Grafica(latox,latoy,cornice=T)
abline(h=0,v=0);
t=seq(0,2*pi,by=0.01)
polygon(Cerchio(t,1))
f=c(1,0,1,1)
rad=polyroot(f)
for (z in rad)
  points(z,pch=21,bg="red")
g=c(1,1,1,1)
rad=polyroot(g)
for (z in rad)
  points(z,pch=21,bg="green")
dev.off()
```

Consideriamo il polinomio

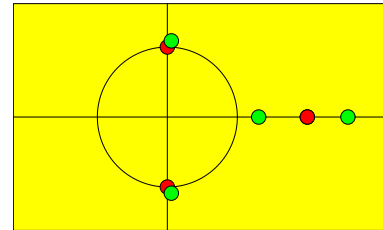
$$f = x^4 - 4x^3 + 5x^2 - 4x + 4$$

$$= (x - 2)^2(x - i)(x + i)$$

Esso possiede una radice doppia, che quindi nella rappresentazione grafica corrisponderà a un unico cerchietto rosso nel punto $(2, 0)$. Se poi modifichiamo il coefficiente di x con -5 , otteniamo il polinomio

$$g = x^4 - 4x^3 + 5x^2 - 5x + 4$$

che, come si vede dalla figura, ha due radici molto vicine alle radici semplici di f e due radici che in un certo senso provengono dalla radice doppia di f anche se sono visibilmente distanti da essa.



Una modifica in Postscript

Da questo numero abbiamo modificato la funzione `Postscript`, dividendo i parametri che indicano larghezza e altezza per 2.54. Ciò ci permette di misurare le figure su carta direttamente in centimetri.

```
Postscript = function (file,larg,alt)
{postscript(file,width=larg/2.54,
height=alt/2.54, horizontal=FALSE,
onefile=FALSE, paper="special")}
```

I cammini delle radici

Consideriamo il polinomio

$$\begin{aligned} f &= x^4(x-2)^2(x+2-i) \\ &= x^7 + (-2-i)x^6 \\ &\quad + (-4+4i)x^5 + (8-4i)x^4 \end{aligned}$$

e per ogni $t \in [0, 1]$ il polinomio

$$f_t = f + tg_t$$

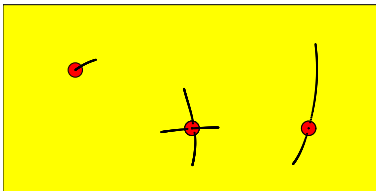
con

$$\begin{aligned} g_t &= -ix^6 + x^5 + ix^4 \\ &\quad + x^3 + 2x^2 + 0.5x - t \end{aligned}$$

Stavolta le radici di g_t le disegniamo usando in `points` il simbolo `pch=20` con un `cex=0.1`:

```
Postscript("37-ft.ps",5,2.5)
latox=c(-3,3); latoy=c(-1,2)
par(bg="yellow")
Grafica(latox,latoy,cornice=T)
f=c(0,0,0,0,8-4i,-4+4i,-2-1i,1)
rad=polyroot(f)
for (z in rad)
  points(z,pch=21,bg="red")
I=seq(0,1,by=0.01)
for (t in I)
  {ft=f+t*c(-t,0.5,2,1,1i,1,-1i,0)
  rad=polyroot(ft)
  for (z in rad)
    points(z,pch=20,cex=0.1)}
dev.off()
```

Otteniamo

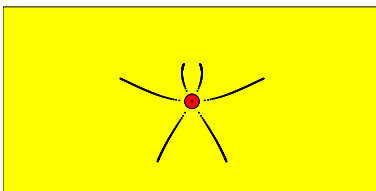


Questi grafici possono anche ingannare; in questo caso ogni ramo si accresce solo in una direzione dalla radice di partenza; se usiamo l'intervallo $[-1, 1]$ avremo invece due (mezzi) rami per ogni nuova radice come nel seguente esempio in cui, con le stesse notazioni,

$$f = x^3$$

e

$$g_t = 0.5x^2 + ix + 1$$



In questo esempio si vede particolarmente bene un fenomeno già osservato nell'ultima figura a pagina 36: Sembra che le nuove radici non partano direttamente dalla radice tripla, ma a una certa distanza da essa, continuando poi su un cammino più continuo. Infatti tipicamente le radici multiple risentono più fortemente di una perturbazione dei coefficienti del polinomio di quanto accade per le radici semplici.

Il teorema di Rouché

Teorema 37.1. *g ed h siano due polinomi a coefficienti complessi e C una circonferenza (cioè il bordo di un cerchio) nel piano tale che*

$$|g(z)| > |h(z)| \text{ per ogni } z \in C.$$

La disuguaglianza è quindi richiesta solo sul bordo del cerchio, non al suo interno. Allora i polinomi g e g + h possiedono all'interno del cerchio lo stesso numero di radici, se queste vengono contate con le loro molteplicità.

Dimostrazione. Questo è un teorema molto importante dell'analisi complessa. Non confondere C con \mathbb{C} .

Il teorema di Rouché si usa nel modo seguente. Assumiamo che vogliamo studiare gli zeri di un polinomio f. Allora decomponiamo f nella forma $f = g + h$, dove g è un polinomio sui cui zeri, almeno all'interno di un cerchio, abbiamo sufficienti informazioni. Consideriamo in altre parole f come perturbazione di g mediante h. Se riusciamo a dimostrare che sul bordo del cerchio si ha sempre $|g(z)| > |h(z)|$, allora possiamo concludere che f e g all'interno del cerchio possiedono lo stesso numero di radici, tenuto conto della loro molteplicità. Quindi se g nel cerchio possiede una sola radice tripla, non possiamo dire che anche f possiede una radice tripla, possiamo però dire che f nel cerchio possiede esattamente tre radici se queste vengono contate con le loro molteplicità.

Il teorema di Rouché permette una veloce dimostrazione del teorema fondamentale dell'algebra (teorema 25.1); naturalmente dovremmo prima, in un corso di analisi complessa, dimostrare il teorema di Rouché.

Sia quindi dato un polinomio

$$f = a_n x^n + a_{n-1} x^{n-1} + \dots + a_0$$

a coefficienti complessi, con $a_n \neq 0$ e non costante, cioè con $n \geq 1$.

Per applicare il teorema di Rouché consideriamo f come perturbazione di

$$g := a_n x^n$$

mediante

$$h := a_{n-1} x^{n-1} + \dots + a_0$$

Per ogni $z \in \mathbb{C}$ si ha $|g(z)| = |a_n| |z|^n$ e si vede facilmente che per $|z|$ sufficientemente grande questa n-esima potenza domina largamente le potenze inferiori e quindi $|h(z)|$, per quanto grandi siano i coefficienti di queste potenze inferiori rispetto al coefficiente a_n di x^n .

Perciò esiste certamente un $R > 0$ tale che per ogni z che appartiene alla circonferenza di raggio R (cioè tale che $|z| = R$) si ha $|g(z)| > |h(z)|$.

Il teorema di Rouché implica allora che f, il nostro polinomio dato, all'interno del cerchio $|z| < R$ possiede lo stesso numero di radici come g. Ma $g = a_n x^n$ in questo cerchio possiede la radice $z = 0$ di molteplicità n. E quindi anche f possiede all'interno dello stesso cerchio n radici e quindi almeno una radice perché per ipotesi $n \geq 1$.

Continuità delle radici

Definizione 37.2. Per un polinomio h a coefficienti complessi sia $|h|_{\text{coeff}}$ il massimo dei valori assoluti dei suoi coefficienti.

Osservazione 37.3. h sia un polinomio a coefficienti complessi di grado $\leq n$. Allora

$$|h(z)| \leq |h|_{\text{coeff}} \sum_{k=0}^n |z|^k$$

per ogni $z \in \mathbb{C}$.

Osservazione 37.4. Siano $z, \alpha \in \mathbb{C}$ e $k \in \mathbb{N}$. Allora

$$|z|^k \leq (|z - \alpha| + |\alpha|)^k$$

Assumiamo di sapere che $|\alpha| \leq A$. Allora, se $|z - \alpha| = r$, si ha

$$|z|^k \leq (r + A)^k$$

Corollario 37.5. Siano $z, \alpha \in \mathbb{C}$ e $k \in \mathbb{N}$.

Assumiamo di sapere che $|\alpha| \leq A$. h sia un polinomio a coefficienti complessi di grado $\leq n$. Allora, se $|z - \alpha| = r$, si ha

$$|h(z)| \leq |h|_{\text{coeff}} \sum_{k=0}^n (r + A)^k$$

Teorema 37.6. $g = a_0 + a_1 x + \dots + a_n x^n$ con $a_n \neq 0$ ed $n \geq 1$ sia un polinomio con coefficienti complessi. $\alpha_1, \dots, \alpha_s$ siano le radici distinte di g; esse possiedono le molteplicità m_1, \dots, m_s . Possiamo allora scegliere $r > 0$ in modo tale che, per ogni $j = 1, \dots, s$, α_j sia l'unica radice di g nel disco aperto $|z - \alpha_j| < r$ e inoltre g non si annulli sulla circonferenza $|z - \alpha_j| = r$ e quindi nemmeno sull'unione K di queste circonferenze. Se vogliamo, possiamo scegliere r anche molto piccolo.

Fissato in tal modo r, per un teorema elementare e fondamentale sulle funzioni continue definite su un insieme compatto (nel nostro caso K), esiste un numero $\mu > 0$ tale che

$$|g(z)| \geq \mu$$

per ogni $z \in K$. Sia inoltre

$$A := \max(|\alpha_1|, \dots, |\alpha_s|)$$

il massimo dei valori assoluti delle radici e sia

$$\delta := \frac{\mu}{\sum_{k=0}^n (r + A)^k}$$

Allora per ogni polinomio h di grado $\leq n$ con

$$|h|_{\text{coeff}} < \delta$$

il polinomio $f := g + h$ possiede per ciascun j esattamente m_j radici all'interno del cerchio $|z - \alpha_j| < r$, se queste vengono contate con le loro molteplicità, f possiede cioè lo stesso numero di radici in questo cerchio come g.

Dimostrazione. Abbiamo preparato quasi tutto per poter applicare il teorema di Rouché. Fissiamo j. Sia $|z - \alpha_j| = r$. Dobbiamo dimostrare che $|h(z)| < |g(z)|$.

Ma per il corollario 37.5 abbiamo

$$\begin{aligned} |h(z)| &\leq |h|_{\text{coeff}} \sum_{k=0}^n (r + A)^k \\ &< \frac{\mu}{\sum_{k=0}^n (r + A)^k} \sum_{k=0}^n (r + A)^k \\ &= \mu \leq |g(z)| \end{aligned}$$

Creazione di matrici

Matrici possono essere definite con il comando `matrix` che abbiamo già incontrato a pagina 33. I componenti di una matrice possono essere numeri reali o complessi, stringhe oppure avere il valore `NA`.

`matrix` ha come argomento obbligatorio un vettore di dati; inoltre si possono indicare con `nrow` il numero delle righe o con `ncol` il numero delle colonne. La matrice viene riempita *colonna per colonna*, se non si indica `byrow=T`. Se `a` è una matrice, con `c(a)` si ottiene il vettore dei suoi elementi, elencati colonna per colonna. Esempi:

```
dati=c(1,2,3,4,5,6,7,8,9,10,11,12)
a=matrix(dati,nrow=3)
print(a); print(c(a))
```

con output (senza le dimensioni)

```
1  4  7 10
2  5  8 11
3  6  9 12
```

```
1 2 3 4 5 6 7 8 9 10 11 12
```

e invece, con gli stessi dati,

```
a=matrix(dati,nrow=3,byrow=T)
print(a); print(c(a))
```

con output

```
1  2  3  4
5  6  7  8
9 10 11 12
```

```
1 5 9 2 6 10 3 7 11 4 8 12
```

Impareremo nel prossimo numero le funzioni di R per l'algebra lineare, in particolare come risolvere un sistema lineare $Ax = b$; adesso le useremo per un po' di grafica.

Parallele

L'istruzione `lines(p)` unisce i punti di un vettore di numeri complessi `p` con segmenti di rette. Se però incontra un valore `NA`, interrompe il tracciato e lo riprende con il prossimo valore definito. Usiamo questa caratteristica per creare schemi di rette parallele.

La funzione `Parallele` che adesso definiamo ha come argomento la larghezza e l'altezza dello schema e il numero di righe parallele; l'argomento opzionale `bordo` viene messo uguale a `T` se vogliamo anche disegnare i bordi superiore e inferiore oppure uguale ad `"s"` o `"i"` se vogliamo solo visualizzare il bordo superiore o inferiore.

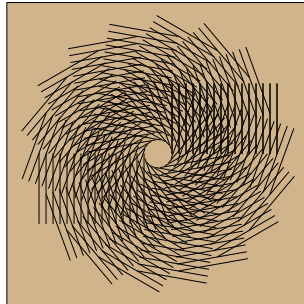
Il risultato della funzione è un vettore di numeri complessi interrotto da valori `NA`.

```
Parallele = function (larg,alt,n,
  bordo=F)
{z1=seq(0,larg,length=n)
 z2=z1+1i*alt; na=rep(NA,n)
 p=c(matrix(c(z1,z2,na),nrow=3,byrow=T))
 bsin=0; asin=1i*alt;
 bdes=larg; ades=larg+asin;
 if (identical(bordo,"s")) c(p,asin,ades)
 else if (identical(bordo,"i"))
   c(p,bsin,bdes)
 else if (bordo)
   c(p,asin,ades,NA,bsin,bdes)
 else p}
```

Come nasce una forma

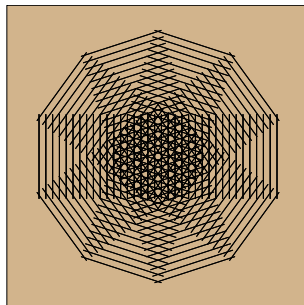
La natura è molto efficiente a creare una straordinaria varietà di forme con elementi semplicissimi. La ricchezza non sta quindi negli ingredienti, ma nella fantasia delle combinazioni. Uno degli strumenti più efficaci è la ripetizione associata a piccole modifiche negli elementi.

La funzione `Parallele` genera un insieme di punti che possiamo sottoporre a traslazioni, rotazioni o altre operazioni geometriche e disegnare con `lines`, come si vede nell'esercizio 65 e nelle seguenti figure:



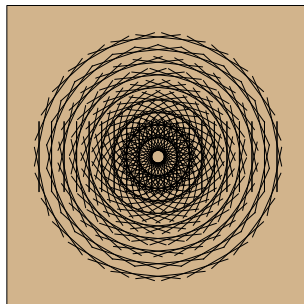
```
Postscript("38-parall1.ps",4,4)
lato=c(-2,2); par(bg="tan")
Grafica(lato,lato,cornice=T)
p=Parallele(1.5,1,16)+0.2
for (alfa in seq(0,360,by=20))
{lines(Rot(p,alfa))}
dev.off()
```

Quanto è semplice!



```
p=Parallele(3.4,1.2,36)-1.7-0.6i
for (alfa in seq(0,360,by=36))
{lines(Rot(p,alfa))}
```

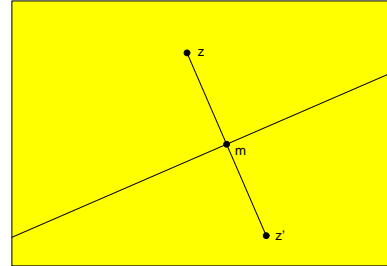
Lo schema è stato centrato.



```
p=Parallele(3.4,1,20)-1.7-0.5i
for (alfa in seq(0,360,by=24))
{lines(Rot(p,alfa))}
```

Esercizi per gli scritti

59.



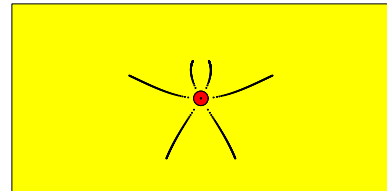
60. f sia un polinomio a coefficienti reali ed $\alpha \in \mathbb{C}$ tale che $f(\alpha) = 0$. Allora anche $f(\overline{\alpha}) = 0$.

Non è difficile, è sufficiente scrivere f esplicitamente con i suoi coefficienti.

Dall'enunciato di questo esercizio deriva facilmente (ma non fa parte del compito) che le radici non reali di un polinomio a coefficienti reali appaiono sempre a coppie coniugate e che un polinomio a coefficienti reali di gradi dispari possiede sempre una radice reale.

61. Le radici del polinomio $f = (x-1)(x-i)$ non sono coniugate tra di loro. Perché non è una contraddizione all'esercizio precedente?

62. Spiegazione a pagina 37.



63. h sia un polinomio a coefficienti complessi di grado $\leq n$. Allora

$$|h(z)| \leq |h|_{\text{coeff}} \sum_{k=0}^n |z|^k$$

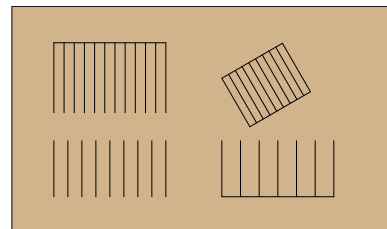
per ogni $z \in \mathbb{C}$.

È quindi richiesta la dimostrazione dell'osservazione 37.2 (molto facile).

64. Spiegare la funzione `Parallele` nella prima colonna su questa pagina.

Ciò significa che nel compito può essere dato il listato della funzione e bisogna spiegare, anche e soprattutto con appositi disegni, il significato delle singole istruzioni.

65. Prendere le misure con la riga.



66. Riflessione del punto $(4, 6)$ nella retta che congiunge i punti $(1, 3)$ e $(13, 5)$.

Matrici in R

Conosciamo già il comando `matrix` per la creazione di matrici (pagina 38) e vogliamo trattare in questo numero le più importanti operazioni con matrici fornite da R. È spesso utile (in statistica, ma anche in matematica pura, cfr. esercizio 22) considerare una matrice $A \in \mathbb{R}_m^n$ (n righe ed m colonne) come un unico lungo vettore di \mathbb{R}^{nm} (elencando gli elementi colonna per colonna come fa R e come sarebbe in effetti più logico sia in algebra lineare che in statistica) oppure di \mathbb{R}_{nm} (elencando gli elementi riga per riga come talvolta è più intuitivo, ma solo perché siamo abituati a leggere riga per riga). A questo punto bisogna distinguere tra le operazioni definite per vettori da quelle definite per matrici.

Abbiamo anche, in un altro senso, due operazioni di moltiplicazione per matrici: da un lato il prodotto elemento per elemento (prodotto di Hadamard), dall'altro la moltiplicazione matriciale. A e B siano due matrici con le appropriate dimensioni (cioè stesso numero di righe e stesso numero di colonne per il prodotto di Hadamard, numero di colonne di A uguale al numero di righe di B per il prodotto matriciale). Allora i coefficienti P_{ij} del prodotto di Hadamard $P := A \bullet B$ sono definiti da

$$P_{ij} = A_{ij} B_{ij}$$

i coefficienti C_{ij} del prodotto matriciale $C := AB$ da

$$C_{ij} = \sum_k A_{ik} B_{kj}.$$

Possiamo naturalmente anche formare la somma di due matrici della stessa forma e moltiplicare una matrice con un numero reale o complesso:

$$(A + B)_{ij} = A_{ij} + B_{ij}$$

$$(tA)_{ij} = tA_{ij}$$

In R le operazioni matriciali corrispondono ai seguenti comandi:

A+B ... addizione
t*A ... moltiplicazione di una matrice con un numero
A*B ... prodotto di Hadamard
A%*%B ... prodotto matriciale
A^2 ... non A^2 , ma $A \bullet A$!
t(A) ... trasposta A^t di A
crossprod(A,B) ... $A^t B$

Sembra che possiamo senza problema usare `t` per numeri anche in uno stesso contesto, non dobbiamo comunque ridefinire `t` come funzione se vogliamo usare anche la trasposta.

$A \bullet B$ è anche definito quando A e B hanno la stessa lunghezza (`length`) totale, ma l'operazione non ha molto senso se le due matrici non hanno la stessa forma.

L'espressione `A%*%b` denota anche il prodotto di una matrice A con un vettore b, in accordo con la notazione matematica; confonde invece probabilmente che si possa usare l'operatore `%*%` anche per il prodotto scalare di vettori.

Esempi per le operazioni matriciali

Definiamo prima una nostra funzione per la creazione di una matrice:

```
Matrix = function
  (dati, righe=NULL, col=NULL)
{if (!is.null(righe))
  matrix(dati,nrow=righe,byrow=T)
else if (!is.null(col))
  matrix(dati,ncol=col)
else NULL}
```

Una matrice è direttamente convertibile in un vettore mediante l'operatore `c`, come abbiamo visto a pagina 38, ma anche in ogni contesto in cui R si aspetta un vettore. Quindi è corretta la sequenza

```
A=Matrix(1:12,col=3)
print(A)
B=Matrix(A,col=4)
print(B)
```

con output semplificato

```
1  5  9
2  6 10
3  7 11
4  8 12

1  4  7 10
2  5  8 11
3  6  9 12
```

In questo numero

- 39 Matrici in R
Esempi per le operazioni matriciali
Piccoli operatori
- 40 Indici vettoriali
Il crivello di Eratostene
for, while e repeat
`v[v%%p0]`
- 41 Assegnazione vettoriale
Indici matriciali
L'opzione drop
`rbind` e `cbind`
- 42 Sistemi lineari con R
`length` e `dim`
`det` (determinante) e traccia
Matrici diagonali
`abs` (valore assoluto) e `sign`
Autovalori
- 43 Matrici simmetriche
`eigen` (autovalori e autovettori)
Matrici reali
I cerchi di Gershgorin
Esercizi 67-76

Piccoli operatori

`prod(v)` è il prodotto degli elementi del vettore `v`; anche questa funzione può essere applicata a matrici, infatti `prod(A)` è uguale a `prod(c(A))`. Esempi:

```
print(prod(1:6))
# output: 720

print(prod(2,6,1,3,4,3))
# output: 432
```

`sum(v)` e `mean(v)` sono la somma e la media del vettore `v`. Esempi:

```
print(sum(1:6))
# output: 21

print(mean(1:6))
# output: 3.5
```

Queste funzioni vengono naturalmente usate molto spesso in statistica.

`cumsum(v)` e `diff(v)` sono i vettori delle somme cumulative e delle differenze del vettore `v`.

```
a=c(3,2,5,8,7,6,10,9)
print(a)
# output: 3 2 5 8 7 6 10 9

s=cumsum(a)
print(s)
# output: 3 5 10 18 25 31 41 50

b=diff(s)
print(b)
# output: 2 5 8 7 6 10 9

print(diff(c(0,s)))
# output: 3 2 5 8 7 6 10 9
```

L'operatore che forma le differenze è inverso all'operatore di sommazione!

Indici vettoriali

R possiede dei meccanismi molto generali e sofisticati per l'uso degli indici in vettori e matrici. Consideriamo il vettore

```
v=11:18
```

Indicando un singolo indice o un vettore di indici ne possiamo estrarre singoli elementi o parti:

```
w=v[2]
print(w)

w=v[c(2,3,8)]
print(w)

w=v[5:9]
print(w)
```

con output

```
12
12 13 18
15 16 17 18 NA
```

Mediante l'uso di indici negativi possiamo escludere alcuni elementi:

```
w=v[-2]
print(w)

w=v[-c(2,3,8)]
print(w)
```

ottenendo

```
11 13 14 15 16 17 18
11 14 15 16 17
```

Una caratteristica avanzata di R è che come indici si possono anche usare vettori di valori booleani, cioè vettori i cui componenti sono o T o F. Se in questo caso il vettore booleano ha una lunghezza minore di quella del vettore da cui vogliamo estrarre, i valori booleani vengono ciclicamente ripetuti. Esempi:

```
v=c(1:8)
filtro=c(F,F,T,T,F,T,F,F)
u=v[filtro]
print(u)
```

con output 3 4 6. Infatti vengono riprodotti in u gli elementi di v che corrispondono a posizioni in cui il valore del vettore booleano è uguale a T.

Con `filtro=c(T,F)` otteniamo ogni secondo elemento di v, con `filtro=c(F,T)` ogni secondo elemento di v, saltando il primo:

```
v=c(1:8)
filtro=c(T,F)
u=v[filtro]
print(u)
filtro=c(F,T)
u=v[filtro]
print(u)
```

con output

```
1 3 5 7
2 4 6 8
```

Possiamo adesso riscrivere in R un famoso algoritmo che permette di trovare i numeri primi $\leq n$ per un dato numero naturale n.

Il crivello di Eratostene

Definizione 40.1. Siano $a, b \in \mathbb{Z}$.

Diciamo che a divide b e scriviamo $a|b$, se esiste $k \in \mathbb{Z}$ tale che $b = ka$.

Quindi $a|b$ se e solo se b è un multiplo di a , cioè se e solo se $b \in \mathbb{Z}a$. È chiaro che allora anche $\mathbb{Z}b \subset \mathbb{Z}a$ e viceversa, quindi $a|b$ se e solo se $\mathbb{Z}b \subset \mathbb{Z}a$.

Questa riduzione della relazione di divisibilità a una relazione insiemistica è molto importante e conduce alla *teoria degli ideali*, un ramo centrale dell'algebra commutativa.

Definizione 40.2. Un numero naturale p si chiama *primo*, se $p \geq 2$ e se, per $d \in \mathbb{N}$ con $d|p$ si ha $d \in \{1, p\}$.

Osservazione 40.3. $n \in \mathbb{N} + 2$ sia un numero naturale ≥ 2 fissato. m sia un altro numero naturale con $2 \leq m \leq n$ che non sia primo. Allora esiste $a \in \mathbb{N}$ con $2 \leq a \leq \sqrt{n}$ tale che $a|m$.

In altre parole, se sappiamo che $m \leq n$, per verificare che m sia primo, dobbiamo solo dimostrare che m non possiede divisori tra 2 e \sqrt{n} .

Nota 40.4. Sia $n \in \mathbb{N} + 2$ un numero naturale ≥ 2 fissato. Possiamo creare una lista di tutti i primi con il seguente procedimento che prende il nome di *crivello di Eratostene*; lo modifichiamo leggermente ai fini del programma in R con cui verrà realizzato.

- (1) Definiamo due vettori variabili u e v e poniamo inizialmente

```
u = ()
v = (2, 3, ..., n)
```

- (2) Calcoliamo la radice $r = \sqrt{n}$.
- (3) Poniamo $p = v_1$.
- (4) Se $p > r$, andiamo alla fine (F).
- (5) Aggiungiamo p a u , ridefinendo $u = (u, p)$ intendendo con ciò che p viene aggiunto come ultimo elemento ad u .
- (6) Eliminiamo da v tutti gli elementi divisibili per p (compreso p stesso).
- (7) Torniamo al punto (3).
- (F) A questo punto la concatenazione (u, v) di u e v rappresenta la successione ordinata dei primi $\leq n$.

Dimostrazione. Infatti il numero p al punto (3) è il più piccolo numero che non è stato eliminato da v nel passaggio precedente; esso non è quindi divisibile da nessun numero tra 2 e $p - 1$ ed è perciò primo, cosicché lo aggiungiamo ad u . Poi eliminiamo da v tutti i multipli di p . Dall'osservazione 40.3 segue che ci possiamo fermare quando $p > \sqrt{n}$.

Possiamo trascrivere questo algoritmo direttamente in una funzione in R che conserviamo nel file *Primi* della nostra libreria:

```
Eratostene = function(n)
{v=2:n; u=(); r=sqrt(n)
repeat
{p=v[1]; if(p>r) break;
u=c(u,p)
v=v[v%%p>0]}
c(u,v)}
```

Per provare la funzione usiamo

```
u=Eratostene(200)
u=c(u,0,0)
A=Matrice(u,righe=6)
print(A)
```

ottenendo l'output

```
 2  3  5  7 11 13 17 19
23 29 31 37 41 43 47 53
59 61 67 71 73 79 83 89
97 101 103 107 109 113 127 131
137 139 149 151 157 163 167 173
179 181 191 193 197 199  0  0
```

Abbiamo creato una matrice per una migliore visualizzazione; per poter riempire la matrice abbiamo anche aggiunto due zeri che naturalmente non derivano dall'algoritmo.

for, while e repeat

R possiede tre istruzioni per l'esecuzione di cicli: `for` (che abbiamo già usato alle pagine 22, 32, 35-37), `while` e `repeat` (che abbiamo usato in Eratostene); esse vengono utilizzate con questa sintassi:

```
for (x in v) istruzioni

while (condizione) istruzioni

repeat istruzioni
```

dove `istruzioni` è un'istruzione singola oppure una successione di istruzioni che allora deve essere racchiusa tra parentesi graffe.

Per uscire da un ciclo si usa `break`, mentre `next` interrompe il passaggio corrente di un ciclo e porta all'immediata esecuzione del passaggio successivo, cosicché

```
for (x in 1:20)
{if (x%%2>0) next
print(x)}
```

stampa sullo schermo i numeri pari compresi tra 1 e 20.

v[v%%p>0]

Dobbiamo ancora analizzare questa misteriosa espressione che abbiamo usato nella funzione Eratostene. Intuitivamente è chiaro che dovrebbe rappresentare quegli elementi del vettore v il cui resto modulo p è maggiore di zero. Ma come si inquadra nella sintassi che abbiamo visto finora?

La spiegazione è che in $v[v%%p>0]$ viene prima calcolato il vettore interno che funge da filtro, cioè il vettore booleano $v%%p>0$ che, secondo il modo vettoriale delle operazioni di R, contiene il valore T in ogni posizione dove in v si trova un numero non divisibile per p , e nelle altre posizioni contiene il valore F.

All'inizio dell'algoritmo, quando $v=2:n$ e $p=2$, $v%%p>0$ è uguale a $c(F,T,F,T,F,T,\dots)$, nel passaggio successivo, in cui $v=c(3,5,7,9,11,13,15,17,\dots)$, $v%%p>0$ è uguale a $c(F,T,T,F,T,T,F,T,\dots)$, e così via.

In modo simile (anche questo esempio è importante nella teoria dei numeri), se $v=0:n$, allora $v[v%%4==1]$ consisterà di tutti i numeri naturali tra 0 e n con resto 1 modulo 4.

Assegnazione vettoriale

Le espressioni della forma `v[a]` possono essere anche usate per assegnare valori alla parte del vettore `v` descritta dall'espressione. Esempi:

```
v=1:8

v[c(3,5,2)]=0
print(v)
# output: 1 0 0 4 0 6 7 8

v[1:4]=c(2,3)
print(v)
# output: 2 3 2 3 0 6 7 8
# Quindi ripetizione di 2,3!

v[-c(7,8)]=c(4,5,1)
print(v)
# output: 4 5 1 4 5 1 7 8

v[-c(6,7,8)]=c(2,0,9)
print(v)
# output: 2 0 9 2 0 1 7 8
# Ma avvertenza!

v[c(T,F)]=c(5,3)
print(v)
# output: 5 0 3 2 5 1 3 8
```

Nel penultimo esempio R ci avverte che la lunghezza della parte dove vogliamo modificare i valori (in questo caso 5) non è un multiplo del vettore che contiene i nuovi valori (in questo caso 3).

Sono operazioni piuttosto potenti.

Indici matriciali

Il coefficiente nella *i*-esima riga e *j*-esima colonna di una matrice `A` viene denotato con `A[i,j]`.

Anche qui possiamo però usare espressioni più complesse, cosicché ad esempio con `A[c(1,3),c(2,4,6)]` otteniamo la matrice `2x3` che consiste dei coefficienti nella prima e terza riga e nella seconda, quarta e sesta colonna di `A`.

Se `u` e `v` sono vettori di indici, allora `A[u,]` è la matrice che consiste delle righe di `A` con indici da `u`, mentre `A[,v]` consiste delle colonne con indici da `v`. `A[-u,]` è invece la matrice da cui abbiamo tolto le righe con indici da `u`. Anche vettori booleani di indici possono essere usati per matrici. Esempi:

```
dati=c(11:16,21:26,31:36,41:46)

A=Matrice(dati,righe=4)
print(A)
```

con output

```
11 12 13 14 15 16
21 22 23 24 25 26
31 32 33 34 35 36
41 42 43 44 45 46
```

e, continuando con la stessa matrice `A`,

```
B=A[c(1,2),]
print(B)
```

con output

```
11 12 13 14 15 16
21 22 23 24 25 26
```

```
B=A[,c(3,5)]
print(B)
```

con output

```
13 15
23 25
33 35
43 45
```

```
B=A[c(T,F),]
print(B)
```

con output

```
11 12 13 14 15 16
31 32 33 34 35 36
```

```
B=A[c(T,F),c(T,F)]
print(B)
```

con output

```
11 13 15
31 33 35
```

```
B=A[, -c(5,6)]
print(B)
```

con output

```
11 12 13 14
21 22 23 24
31 32 33 34
41 42 43 44
```

Se usiamo un filtro, otteniamo il vettore dei coefficienti della matrice che soddisfano il criterio espresso dal filtro, elencati colonna per colonna:

```
filtro=A%%4==1
print(filtro)
```

```
B=A[filtro]
print(B)
```

con output

```
FALSE FALSE TRUE FALSE FALSE FALSE
TRUE FALSE FALSE FALSE TRUE FALSE
FALSE FALSE TRUE FALSE FALSE FALSE
TRUE FALSE FALSE FALSE TRUE FALSE

21 41 13 33 25 45
```

Anche indici matriciali possono essere usati per assegnare valori a parti di una matrice:

```
A[1,]=c(91:96)
print(A)
A[,2]=c(81:84)
print(A)
```

con output

```
91 92 93 94 95 96
21 22 23 24 25 26
31 32 33 34 35 36
41 42 43 44 45 46
```

```
91 81 93 94 95 96
21 82 23 24 25 26
31 83 33 34 35 36
41 84 43 44 45 46
```

L'opzione drop

Quando scegliamo solo una riga o una colonna di una matrice, queste vengono automaticamente convertite in un vettore. Talvolta però si vorrebbe poter considerare quella riga o colonna come matrice; per fare ciò bisogna aggiungere l'opzione `drop=F` all'indice. Esempio:

```
dati=c(11:14,21:24,31:34)

A=Matrice(dati,righe=3)
print(A)
cat('-----\n')

v=A[1,]
print(v)
cat('-----\n')

v=A[1,,drop=F]
print(v)
```

con output, stavolta completo, così come appare sullo schermo,

```
      [,1] [,2] [,3] [,4]
[1,] 11  12  13  14
[2,] 21  22  23  24
[3,] 31  32  33  34
-----
[1] 11 12 13 14
-----
      [,1] [,2] [,3] [,4]
[1,] 11  12  13  14
```

rbind e cbind

Possiamo unire più righe con `rbind` e più colonne con `cbind`; le righe o colonne possono essere date singolarmente o anche come matrici. Esempi:

```
dati=c(11:13,21:23,31:33)
altridati=c(14:15,24:25,34:35)

A=Matrice(dati,righe=3)
print(A)
B=Matrice(altridati,righe=3)
print(B)
C=cbind(A,B)
print(C)
```

con output

```
11 12 13
21 22 23
31 32 33

14 15
24 25
34 35

11 12 13 14 15
21 22 23 24 25
31 32 33 34 35
```

e, continuando con la stessa matrice `C`,

```
D=rbind(C,81:85,91:95)
print(D)
```

con output

```
11 12 13 14 15
21 22 23 24 25
31 32 33 34 35
81 82 83 84 85
91 92 93 94 95
```

Sistemi lineari con R

R permette, per matrici che devono essere quadratiche e invertibili, la risoluzione di un sistema lineare

$$Ax = b$$

mediante l'istruzione

```
x=solve(A,b)
```

Proviamo prima con il sistema che abbiamo risolto a pagina 8:

```
dati=c(3,-2,1,6)
A=Matrice(dati,righe=2)
v=c(8,5)
x=solve(A,v)
print(x)
```

L'output è 2.90 0.35, in accordo con la soluzione già trovata, infatti

$$\frac{58}{20} = 2.9, \quad \frac{7}{20} = 0.35$$

Proviamo con il sistema dell'esempio 9.1:

```
dati=c(3,2,-1,4,-9,2,1,1,-14)
A=Matrice(dati,righe=3)
v=c(10,6,2)
x=solve(A,v)
print(round(x,4))
```

La risposta è

```
2.9305 0.6611 0.1137
```

in accordo con quanto già trovato.

Se vogliamo le soluzioni come numeri razionali, nell'ultima riga del programma possiamo usare

```
print(fractions(x))
```

La funzione `fractions` richiede però la libreria `MASS`, che dobbiamo caricare con

```
library(MASS)
```

Otteniamo allora l'output

```
1392/475 314/475 54/475
```

L'inversa di una matrice non singolare A è data da

```
solve(A)
```

Per verificare se la matrice è invertibile si può calcolare il determinante con `det(A)`.

La libreria `MASS` contiene la funzione `ginv` che calcola l'inversa generalizzata nel senso di Penrose di matrici non necessariamente quadratiche.

length e dim

`length(v)` è la lunghezza di un vettore o di una matrice.

Nel caso di una matrice questa viene considerata come vettore, la lunghezza è quindi uguale ad nm , se la matrice possiede n righe ed m colonne.

`dim(A)` è allora il vettore (n, m) .

det (determinante) e traccia

`det(A)` è il determinante della matrice A .

Per la traccia sembra che non ci sia una apposita funzione che però possiamo facilmente definire:

```
Traccia = function (A)
{if (length(A)>1) sum(diag(A))
else c(A)}
```

Matrici diagonali

La funzione `diag` può essere usata in più modi:

```
diag(6)
```

genera una matrice identica 6×6 ,

```
diag(v)
```

genera, per un vettore v , una matrice diagonale che contiene i coefficienti di v nella diagonale principale, mentre infine

```
diag(A)
```

è il vettore che consiste degli elementi della diagonale principale di A . Esempi:

```
A=diag(4)
print(A)
```

con output

```
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
```

```
A=diag(c(2,3,5))
print(A)
```

con output

```
2 0 0
0 3 0
0 0 5
```

```
dati=c(11:13,21:23,31:33)
A=Matrice(dati,righe=3)
print(A)
print(diag(A))
```

con output

```
11 12 13
21 22 23
31 32 33
```

```
11 22 33
```

abs (valore assoluto) e sign

`abs(x)` è il valore assoluto del numero reale o complesso x .

Secondo la filosofia di R, se v è un vettore, con `abs(v)` si ottiene il vettore dei valori assoluti degli elementi del vettore; lo stesso vale per una matrice dalla quale otteniamo quindi la matrice dei valori assoluti dei suoi coefficienti.

`sign(x)` è il segno di un numero reale (uguale a zero se x è uguale a zero).

Autovalori

Situazione 42.1. K sia un campo e $A \in K_n^n$ una matrice quadratica.

Denotiamo con δ la matrice identica.

Teorema 42.2. Consideriamo l'applicazione

$$\varphi = \bigcirc_x Ax : K^n \longrightarrow K^n$$

Allora i seguenti enunciati sono equivalenti:

- (1) φ non è biiettiva.
- (2) φ non è suriettiva.
- (3) φ non è iniettiva.
- (4) $\det A = 0$.

È chiaro che la condizione (3) si verifica se e solo se esiste un vettore $x \neq 0$ in K^n tale che $Ax = 0$.

Dimostrazione. Corsi di Geometria.

Definizione 42.3. Un vettore $x \in K^n$ si chiama un *autovettore* di A , se $x \neq 0$ e se esiste un elemento $\lambda \in K$ tale che $Ax = \lambda x$.

Ciò accade se e solo se $(A - \lambda\delta)x = 0$.

Definizione 42.4. Un elemento $\lambda \in K$ si chiama un *autovalore* di A , se esiste un vettore $x \in K^n$ con $x \neq 0$ e $Ax = \lambda x$.

In tal caso diciamo anche che x è un autovettore di A per l'autovalore λ e che λ è un autovalore di A per l'autovettore x .

Osservazione 42.5. Sia $\lambda \in K$. Allora λ è un autovalore di A se e solo se

$$\det(A - \lambda\delta) = 0.$$

Dimostrazione. Teorema 42.2.

Nota 42.6. Sviluppando il determinante, non è difficile dimostrare che

$$\det(\lambda\delta - A) = (-1)^n \det(A - \lambda\delta)$$

è un polinomio di grado n in λ (qui si usa spesso la stessa lettera λ sia per un elemento concreto di K che per un'indeterminata) che si chiama il *polinomio caratteristico* di A .

Consideriamo il caso $n = 2$. Allora

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

e

$$\lambda\delta - A = \begin{pmatrix} \lambda - a & -b \\ -c & \lambda - d \end{pmatrix}$$

quindi

$$\begin{aligned} \det(\lambda\delta - A) &= \begin{vmatrix} \lambda - a & -b \\ -c & \lambda - d \end{vmatrix} \\ &= (\lambda - a)(\lambda - d) - bc \\ &= \lambda^2 - \lambda(a + d) + ad - bc \\ &= \lambda^2 - \lambda \cdot \text{tr } A + \det A \end{aligned}$$

Per una matrice triangolare

$$A = \begin{pmatrix} a & b \\ 0 & d \end{pmatrix}$$

il polinomio caratteristico diventa

$$\begin{vmatrix} \lambda - a & -b \\ 0 & \lambda - d \end{vmatrix} = (\lambda - a)(\lambda - d)$$

e quindi i due autovalori in questo caso sono gli elementi a e d nella diagonale principale.

Matrici simmetriche

Consideriamo una matrice reale simmetrica

$$A = \begin{pmatrix} a & b \\ b & d \end{pmatrix}$$

Per la formula vista a pagina 42

$$\det(A - \lambda \delta) = \lambda^2 - \lambda(a + d) + ad - b^2$$

Gli autovalori di A (nel campo complesso) sono quindi dati da

$$\lambda = \frac{a + d \pm \sqrt{D}}{2}$$

con

$$\begin{aligned} D &= (a + d)^2 - 4(ad - b^2) \\ &= a^2 + 2ad + d^2 - 4ad + 4b^2 \\ &= a^2 - 2ad + d^2 + 4b^2 \\ &= (a - d)^2 + 4b^2 \end{aligned}$$

Perciò $D \geq 0$ e vediamo che gli autovalori di una matrice reale *simmetrica* 2×2 sono reali.

Ciò vale anche per le dimensioni superiori e può essere dimostrato, in modo molto più semplice, utilizzando il prodotto scalare e l'ortogonalità.

eigen (autovalori e autovettori)

In R gli autovalori di una matrice reale o complessa si trovano con la funzione `eigen`. Essa calcola, se non si pone l'opzione

```
only.values=T
```

anche un sistema di autovettori; ciò può rallentare il calcolo per matrici molto grandi. Il risultato è una *lista* in R, un concetto che non riusciamo più a trattare, e le componenti si ottengono con la sintassi `$values` e `$vectors`. Creiamo due funzioni che inseriamo nel file *Matrici* della nostra libreria:

```
Autovalori = function (A)
{eigen(A, only.values=T)$values}

Autovettori = function (A)
{eigen(A)$vectors}
```

Calcoliamo gli autovalori della matrice

$$\begin{pmatrix} 6 & -2 \\ 2 & 6 \end{pmatrix}$$

con

```
A=Matrice(c(6,2,-2,6),col=2)
lambda=Autovalori(A)
print(round(lambda,2))
```

trovando $6+2i$ $6-2i$ e nello stesso modo quelli di

$$\begin{pmatrix} 6 & 2 \\ 2 & 6 \end{pmatrix}$$

che, secondo quanto visto prima, dovrebbero essere reali. Infatti troviamo la risposta

```
8 4.
```

Matrici reali

Una matrice reale $A \in \mathbb{R}_n^n$ non possiede necessariamente autovalori reali; d'altra parte A può essere anche considerata come una matrice in \mathbb{C}_n^n e il polinomio caratteristico, assumendo naturalmente $n \geq 1$, in \mathbb{C} possiede sempre radici. Quindi A possiede in ogni caso autovalori e autovettori complessi e vediamo di nuovo che lo studio di un problema formulato in ambito reale ci porta in modo naturale nel regno dell'analisi complessa.

I cerchi di Gershgorin

Per una matrice reale o complessa 2×2

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

definiamo i cerchi

$$C_1 := \{z \in \mathbb{C} \mid |z - a| \leq |b|\}$$

$$C_2 := \{z \in \mathbb{C} \mid |z - d| \leq |c|\}$$

similmente per una matrice 3×3

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

poniamo

$$C_1 := \{z \in \mathbb{C} \mid |z - a_{11}| \leq |a_{12}| + |a_{13}|\}$$

$$C_2 := \{z \in \mathbb{C} \mid |z - a_{22}| \leq |a_{21}| + |a_{23}|\}$$

$$C_3 := \{z \in \mathbb{C} \mid |z - a_{33}| \leq |a_{31}| + |a_{32}|\}$$

Questi cerchi, che si definiscono in modo analogo per le dimensioni superiori, si chiamano i *cerchi di Gershgorin* di A .

I centri dei cerchi di Gershgorin sono gli elementi nella diagonale principale della matrice; se la matrice è reale o se almeno gli elementi nella diagonale sono tutti reali, i centri dei cerchi di Gershgorin si troveranno quindi sull'asse reale.

Teorema 1 di Gershgorin. *Gli autovalori di A sono tutti contenuti nell'unione dei suoi cerchi di Gershgorin.*

Dimostrazione. Corsi di Analisi numerica. Abbastanza facile.

Definiamo una funzione (che farà parte del file *Matrici* nella libreria) che restituisce il centro e il raggio dell' i -esimo cerchio di Gershgorin della matrice A :

```
Gershgorin = function (A,i)
{centro=A[i,i]
raggio=sum(abs(A[i,]))-abs(centro)
c(centro,raggio)}
```

Calcoliamo i cerchi di Gershgorin della matrice dell'esercizio 75:

```
dati=c(1,4,1,3,2,-1,4,0,
3,2,5,-2,4,-3,1,8)
A=Matrice(dati,righe=4)
```

```
for (i in 1:4)
print(Gershgorin(A,i))
```

Esercizi per gli scritti

67. Una matrice di rotazione

$$\begin{pmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{pmatrix}$$

possiede autovalori reali se e solo se $\cos \alpha \in \{1, -1\}$. Come è fatta allora la matrice e come opera?

68. Gli autovalori di

$$\begin{pmatrix} 6 & -2 \\ 2 & 6 \end{pmatrix}$$

sono $6 + 2i$ e $6 - 2i$. Calcolare in questi esercizi gli autovalori a mano.

69. Gli autovalori di

$$\begin{pmatrix} 6 & 2 \\ 2 & 6 \end{pmatrix}$$

sono 8 e 4.

70. Gli autovalori di una matrice reale o complessa della forma

$$\begin{pmatrix} a & b \\ b & a \end{pmatrix}$$

sono $a + b$ e $a - b$.

71. Gli autovalori di

$$\begin{pmatrix} 15 & 4 \\ 4 & 0 \end{pmatrix}$$

sono -1 e 16 . Verificare con R.

72. Gli autovalori di

$$\begin{pmatrix} 21 & 4 \\ 4 & 6 \end{pmatrix}$$

sono 22 e 5 . Come si trovano queste belle matrici? Guardare a pagina 12.

73. Sia λ un autovalore di

$$\begin{pmatrix} 0 & -2i \\ 3 + 4i & 0 \end{pmatrix}$$

Allora $|\lambda| \leq 5$.

74. Sia λ un autovalore di

$$\begin{pmatrix} 0 & 2 & 1 \\ 3 & 0 & 2 - i \\ 1 - i & 2 + i & 0 \end{pmatrix}$$

Allora $|\lambda| \leq 3 + \sqrt{5}$.

75. Disegnare su sfondo giallo i cerchi di Gershgorin (in rosso) della matrice

$$\begin{pmatrix} 1 & 4 & 1 & 3 \\ 2 & -1 & 4 & 0 \\ 3 & 2 & 5 & -2 \\ 4 & -3 & 1 & 8 \end{pmatrix}$$

Usare `points` con `pch=20` per i centri. La funzione `Gershgorin` viene chiamata dall'interno del programma.

76. Definire con `dati=...` e `A=...` come alla fine della colonna precedente la matrice dell'esercizio 75. Creare, con un unico comando `B=ifelse(...)` la matrice che si ottiene da A sostituendo tutti i valori negativi con 0. Verificare con `print(B)`.

Derivazione simbolica

Come vediamo nell'articolo su questa pagina, la formazione della derivata ha molti aspetti puramente formali e algebrici. Per questa ragione anche R è in grado di effettuare la differenziazione di espressioni formali e fornisce a questo scopo la funzione D che si usa

come in questo esempio:

```
formula=expression(x^4-x)
derivata=D(formula,"x")
print(derivata)
```

con risultato

$4 * x^3 - 1$

La derivata

$f : U \rightarrow \mathbb{R}$ sia una funzione a valori reali definita su un aperto U di \mathbb{R} (quasi sempre U sarà un intervallo aperto, anche infinito, di \mathbb{R} oppure un'unione di un numero finito di intervalli aperti). Sia $x_0 \in U$. Se il limite

$$f'(x_0) := \lim_{h \rightarrow 0} \frac{f(x_0 + h) - f(x_0)}{h}$$

esiste, allora si dice che la funzione f è differenziabile in x_0 ed $f'(x_0)$ si chiama la derivata di f in x_0 .

Segue abbastanza direttamente dalla definizione che, se anche g è una funzione definita in U e se esistono le derivate $f'(x_0)$ e $g'(x_0)$, allora anche $(f + g)'(x_0)$ esiste e si ha

$$(f + g)'(x_0) = f'(x_0) + g'(x_0)$$

Si dimostra inoltre che esiste anche $(fg)'(x_0)$ e che

$$(fg)'(x_0) = f'(x_0)g(x_0) + f(x_0)g'(x_0)$$

Qui fg è la funzione $\bigcirc_x f(x)g(x)$, da non confondere con la composizione delle due funzioni. Se D è l'insieme dei punti in cui f e g sono entrambe differenziabili, possiamo considerare f' e g' come funzioni definite su D e scrivere più brevemente

$$(f + g)' = f' + g'$$

$$(fg)' = f'g + fg'$$

Per una funzione costante $f(x_0 + h)$ e $f(x_0)$ coincidono, e vediamo che la derivata di una funzione costante è uguale a zero.

Consideriamo la funzione identica. In questo caso il limite da studiare è

$$\lim_{h \rightarrow 0} \frac{(x_0 + h) - x_0}{h} = \lim_{h \rightarrow 0} \frac{h}{h} = 1$$

perciò la funzione identica id ha derivata 1 in ogni punto.

Assumiamo adesso che la f sia differenziabile in x_0 e consideriamo le funzioni f^2 , f^3 e f^4 . Per la regola del prodotto abbiamo

$$(f^2)' = f'f + ff' = 2ff'$$

$$(f^3)' = f'f^2 + f(f^2)'$$

$$= f'f^2 + f \cdot 2ff' = 3f'f^2$$

$$(f^4)' = f'f^3 + f(f^3)'$$

$$= f'f^3 + f \cdot 3f'f^2 = 4f'f^3$$

Intravediamo la formula generale

$$(f^n)' = n f' f^{n-1}$$

che infatti si dimostra facilmente per ogni $n \in \mathbb{N}$ con induzione. Se poniamo f uguale all'identità otteniamo la derivata di un monomio

$$(\text{id}^n)' = n \text{id}^{n-1}$$

che, in modo meno preciso ma più intuitivo, talvolta è scritta nella forma

$$(x^n)' = n x^{n-1}$$

Per un numero reale a abbiamo

$$(af)' = af'$$

perché la derivata della funzione costante $\bigcirc_x a$ è uguale a zero, come abbiamo visto.

Ciò mostra, insieme alla regola per la somma, che la formazione della derivata è un operatore lineare e ci permette di calcolare la derivata di una funzione polinomiale. Sia infatti

$$f = \bigcirc_x a_0 + a_1 x + a_2 x^2 + \dots + a_n x^n$$

Usando la formula per la derivata di un monomio e la linearità otteniamo

$$f' = \bigcirc_x a_1 + 2a_2 x + \dots + na_n x^{n-1}$$

Consideriamo adesso il quoziente $\varphi := \frac{f}{g}$ di due funzioni differenziabili, dove restringiamo U in modo tale che non contenga zeri di g . Siccome $\varphi g = f$, dalla regola del prodotto abbiamo $\varphi'g + \varphi g' = f'$ e quindi

$$\left(\frac{f}{g}\right)' = \frac{f' - \varphi g'}{g} = \frac{f' - \frac{f}{g}g'}{g}$$

$$= \frac{f'g - fg'}{g^2}$$

Nei corsi di Analisi si dimostra (e la dimostrazione richiede un po' di attenzione!) la regola della catena, molto importante, per la composizione di due funzioni differenziabili:

$$(g \circ f)'(x_0) = f'(x_0) \cdot g'(f(x_0))$$

che in modo astratto può essere scritta anche così (nel dominio dove tutto è definito):

$$(g \circ f)' = f' \cdot (g' \circ f)$$

In questo numero

- 44 Derivazione simbolica
La derivata
La funzione esponenziale
- 45 Abbreviazioni
Esempi per l'uso di D
La serie di Taylor
Options e print(x,n)
- 46 any e all
Ordinamento
La libreria

La funzione esponenziale

Esiste una funzione che è la derivata di se stessa? Nei corsi di Analisi si impara che una tale funzione, cioè una funzione φ con $\varphi' = \varphi$ esiste veramente e che è unica se chiediamo che $\varphi(0) = 1$. Questa funzione si chiama la funzione *esponenziale* e viene denotata con \exp . Essa può essere definita per ogni numero reale - in verità può essere estesa su tutto \mathbb{C} , è quindi una funzione

$$\exp : \mathbb{R} \rightarrow \mathbb{R}$$

oppure, rivelando allora la sua vera natura e la profonda parentela con le funzioni trigonometriche, una funzione

$$\exp : \mathbb{C} \rightarrow \mathbb{C}$$

\exp è quindi una soluzione dell'equazione differenziale

$$\varphi' = \varphi$$

e se con D denotiamo l'operatore (di cui sappiamo che è lineare) di differenziazione (che è definito su un qualche spazio vettoriale di funzioni di cui la funzione esponenziale fa parte), vediamo che \exp è soluzione dell'equazione lineare

$$D\varphi = \varphi$$

e quindi un autovettore per l'autovalore 1 (cioè un punto fisso) dell'operatore lineare D e non ci sorprende che, come si verifica facilmente, l'insieme delle soluzioni forma un sottospazio vettoriale in quello spazio di funzioni. Si dimostra anzi che le soluzioni formano una retta e che quindi le funzioni che coincidono con la propria derivata sono esattamente i multipli scalari $a \exp$ della funzione esponenziale. Per $a = 0$ otteniamo la funzione costante zero che effettivamente anch'essa è la derivata di se stessa.

Sia $\varphi = a \exp$. Allora $\varphi(0) = a$ e vediamo che ogni soluzione della nostra equazione differenziale può essere scritta nella forma

$$\varphi = \varphi(0) \cdot \exp$$

Si dimostra nei corsi di Analisi che $\exp(x)$ coincide con e^x , dove il numero e (detto anche base del logaritmo naturale) è definito da

$$e = \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n = 2.718281828459 \dots$$

In R la funzione esponenziale corrisponde alla funzione `exp`; il numero e non è predefinito ma può essere ottenuto calcolando `exp(1)`. Ciò costa tempo però in esecuzioni ripetute, quindi inseriamo il valore nel file *Costanti*:

$$\text{CME} = 2.718281828$$

Abbreviazioni

Per evitare di dover battere la parola complicata *expression* mettiamo la seguente riga nel file *Abbreviazioni* della nostra libreria:

```
Formula = expression
```

Nello stesso modo possiamo rinominare altre funzioni:

```
Radici = polyroot
Tratto = lines
Trattoc = polygon
```

Esempi per l'uso di D

Verifichiamo le formule

```
f(x) = x^7 ==> f'(x) = 7*x^6
f(x) = (x^2+1)/(x^3+2*x+6) ==> f'(x) =
= (2*x)/(x^3+2*x+6) - ((x^2+1)*(3*x^2+2))/(x^3+2*x+6)^2
cos' = -sin, sin' = cos, exp' = exp
f(x) = e^x ==> f'(x) = 2*x*e^x^2
f(x) = cos(x)*sin(x) ==>
f'(x) = cos^2(x) - sin^2(x)
f(x) = sqrt(x) ==> f'(x) = 1/(2*sqrt(x))
tan' = 1/cos^2, cosh' = sinh
```

con

```
Dx=function (F) D(F,"x")

F=Formula(x^7)
print(Dx(F))
# output: 7*x^6

F=Formula((x^2+1)/(x^3+2*x+6))
print(Dx(F))
# output: 2*x/(x^3+2*x+6)
# -(x^2+1)*(3*x^2+2)/(x^3+2*x+6)^2

F=Formula(cos(x))
print(Dx(F))
# output: -sin(x)

F=Formula(sin(x))
print(Dx(F))
# output: cos(x)

F=Formula(exp(x))
print(Dx(F))
# output: exp(x)

F=Formula(exp(x^2))
print(Dx(F))
# output: exp(x^2)*(2*x)

F=Formula(cos(x)*sin(x))
print(Dx(F))
# output: cos(x)*cos(x)-sin(x)*sin(x)

F=Formula(sqrt(x))
print(Dx(F))
# output: 0.5 * x^-0.5

F=Formula(tan(x))
print(Dx(F))
# output: 1/cos(x)^2

F=Formula(tanh(x))
print(Dx(F))
# output: 1/cosh(x)^2

F=Formula(cosh(x))
print(Dx(F))
# output: sinh(x)
```

La serie di Taylor

Molte funzioni reali (ad esempio i polinomi, le funzioni trigonometriche e iperboliche, la funzione esponenziale, i quozienti di polinomi) possono, per ogni punto x_0 del loro dominio di definizione ed ogni x sufficientemente vicino ad x_0 (e spesso per ogni $x \in \mathbb{R}$) essere rappresentate mediante la loro serie di Taylor:

$$f(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(x_0)}{k!} (x - x_0)^k$$

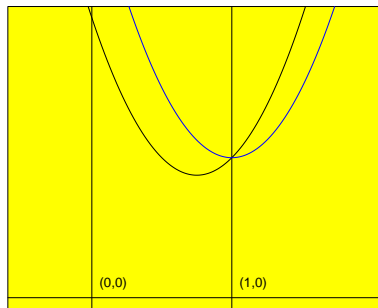
dove $f^{(k)}(x_0)$ è la k -esima derivata di f nel punto x_0 . Questa rappresentazione fornisce molte informazioni sul comportamento della funzione vicino al punto x_0 .

Anche quando f è una funzione polinomiale la serie di Taylor in x_0 è interessante, perché ci dice come la f si esprime in x_0 . In questo caso naturalmente la serie di Taylor è formalmente un polinomio in $x - x_0$ dello stesso grado di quello di partenza. Abbiamo ad esempio

$$2 - 3x + 2x^2 = 1 + (x - 1) + 2(x - 1)^2$$

$$3 - 4x + 2x^2 = 1 + 2(x - 1)^2$$

e vediamo che le due funzioni coincidono in $x_0 = 1$, ma vicino ad $x_0 = 1$ il secondo polinomio si comporta come la funzione $y = 1 + 2x^2$ in $x = 0$, il primo invece piuttosto come la retta $y = 1 + x$.



Creiamo adesso una funzione *Taylor* che restituisce i primi $n+1$ coefficienti di una funzione (rappresentata mediante un'espressione formale) in un punto x_0 . Per calcolare il valore della k -esima derivata in x_0 usiamo *eval* che conosciamo dalla pagina 29.

```
Taylor = function (F,x0,n)
{ x=x0; coeff=c(eval(F)); fatt=1
  for (k in 1:n)
  { F=D(F,"x"); fatt=fatt*k
    coeff=c(coeff,eval(F)/fatt)
  }
  coeff }
```

La usiamo per calcolare i coefficienti del primo dei due polinomi visti sopra; questo polinomio ha grado 2 e quindi possiamo impostare n uguale a 2:

```
F=Formula(2-3*x+2*x^2)
taylor=Taylor(F,1,2)
print(taylor)
```

Otteniamo, correttamente, 1 1 2.

Proviamo adesso con lo sviluppo di Taylor della funzione esponenziale in $x_0 = 0$:

```
F=Formula(exp(x))
taylor=Taylor(F,0,8)
print(taylor)
```

Vediamo dei numeri che a prima vista non sembrano dire molto; ma se visualizziamo invece i coefficienti come frazioni di interi con

```
print(fractions(taylor,
  max.denominator=100000))
```

dopo aver caricato, se necessario, la libreria *MASS* con *library(MASS)* (cfr. pagina 42), troviamo

```
1 1 1/2 1/6 1/24 1/120 1/720 1/5040 1/40320
```

Questi sono i reciproci dei fattoriali! Infatti la serie di Taylor della funzione esponenziale nell'origine è

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!}$$

Abbiamo aumentato il limite superiore del denominatore in *fractions*. Proviamo con il coseno, sostituendo *cos* ad *exp*, e otteniamo

```
1 0 -1/2 0 1/24 0 -1/720 0 1/40320
```

Infatti

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \dots$$

Sostituiamo *cos* con *cosh* per calcolare lo sviluppo di Taylor del coseno iperbolico e ritroviamo gli stessi coefficienti, ma senza segno. Quindi

$$\cosh x = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \frac{x^6}{6!} + \frac{x^8}{8!} + \dots$$

Proviamo infine con $f(x) = \frac{1}{1-x}$:

```
F=Formula(1/(1-x))
taylor=Taylor(F,0,8)
print(taylor)
# output: 1 1 1 1 1 1 1 1 1
```

Infatti $\frac{1}{1-x} = 1 + x + x^2 + x^3 + x^4 + \dots$,

ma stavolta la serie di Taylor converge solo per $|x| < 1$, benché il quoziente $\frac{1}{1-x}$ sia definito per ogni $x \neq 1$.

Options e print(x,n)

Il numero delle cifre significative che vengono visualizzate in un numero reale è preimpostato a 7 e può essere modificato mediante il comando

```
options(digits=12)
```

per avere ad esempio 12 cifre significative. Il valore massimo è probabilmente 22, ma sembra che la precisione arrivi a circa 16 cifre dietro il punto decimale. *options* funziona quindi in modo simile a *par*. Per vedere anche gli altri parametri provare *options()*.

Se si vuole stampare un valore solo una volta con un certo numero di cifre, si può utilizzare il secondo parametro di *print*.

Esempio:

```
e=exp(1); print(e)
# output: 2.718282
```

```
print(e,14)
# output: 2.7182818284590
```

```
options(digits=10)
print(e)
# output: 2.718281828
```

any e all

Se v è un vettore di valori booleani (cioè uguali a TRUE o FALSE), $\text{any}(v)$ è vero se e solo se almeno uno dei componenti di v è vero, mentre $\text{all}(v)$ è vero se e solo se tutti i componenti di v sono veri. Esempi:

```
v=c(T,T,F,T)
print(all(v))
# output: FALSE
print(any(v))
# output: TRUE

v=c()
print(all(v))
# output: TRUE
print(any(v))
# output: FALSE

u=c(-1,3,5,0,4)
print(u>0)
# output: FALSE TRUE TRUE FALSE TRUE
print(all(u>0))
# output: FALSE
print(any(u>0))
# output: TRUE

u=c(0.7,0.3,2,1)
v=u>u^2; print(v)
# output: TRUE TRUE FALSE FALSE
print(all(v))
# output: FALSE
print(any(v))
# output: TRUE
```

Ordinamento

Trattiamo qui le seguenti funzioni:

```
rev sort rank order
```

rev non è una vera funzione di ordinamento, ma restituisce semplicemente gli elementi di un vettore in ordine invertito:

```
u=c(1:3,8,0,3:6,2); print(u)
# output: 1 2 3 8 0 3 4 5 6 2
i=rev(u); print(i)
# output: 2 6 5 4 3 0 8 3 2 1
```

La funzione di ordinamento sort ordina un vettore di numeri reali, in ordine crescente se non è indicata l'opzione decreasing=T . Usiamo lo stesso u :

```
u=c(1:3,8,0,3:6,2)
crescente=sort(u)
print(crescente)
# output: 0 1 2 2 3 3 4 5 6 8
decrecente=sort(u,decreasing=T)
print(decrecente)
# output: 8 6 5 4 3 3 2 2 1 0
```

Con rank si ottiene la posizione di ogni elemento nella successione ordinata, con order invece successivamente la posizione dell'elemento più piccolo nella successione originale, poi quella del secondo e così via:

```
# u: 1 2 3 8 0 3 4 5 6 2
# sort(u): 0 1 2 2 3 3 4 5 6 8
print(rank(u))
# output con 2 invece di 2.0 ecc.:
# 2 3.5 5.5 10 1 5.5 7 8 9 3.5
print(order(u))
# output: 5 1 2 10 3 6 7 8 9 4
```

La libreria

Elenchiamo qui i singoli files della nostra libreria in ordine alfabetico, tralasciando le *Octobrinidae* e *Parabrinidae*.

Abbreviazioni

```
Formula = expression
Radici = polyroot
Tratto = lines
Trattoc = polygon
```

Analisi

```
Taylor = function (F,x0,n)
{x=x0; coeff=c(eval(F)); fatt=1
for (k in 1:n)
{F=D(F,"x"); fatt=fatt*k
coeff=c(coeff,eval(F)/fatt)}
coeff}
```

Costanti

```
CME = 2.718281828 # Altrimenti
# ogni volta ricalcolo di exp(1)
CMPid180 = 0.01745329;
CM180dPi = 57.29577951;
```

Figure

```
Cerchio = function (t,r)
{Ellisse(t,r,r)}
```

```
Ellisse = function (t,a,b)
{a*cos(t)+1i*b*sin(t)}
```

```
# Ramo destro di un'iperbole.
# Ramo sinistro con -a.
Iperbole = function (t,a,b)
{a*cosh(t)+1i*b*sinh(t)}
```

```
# Pagina 38.
Parallele = function (larg,alt,n,
bordo=F)
{z1=seq(0,large,length=n)
z2=z1+1i*alt; na=rep(NA,n)
p=c(matrix(c(z1,z2,na),nrow=3,byrow=T))
bsin=0; asin=1i*alt;
bdes=larg; ades=larg+asin;
if (identical(bordo,"s")) c(p,asin,ades)
else if (identical(bordo,"i"))
c(p,bsin,bdes)
else if (bordo)
c(p,asin,ades,NA,bsin,bdes)
else p}
```

```
# Vertici di un n-gono regolare
# di raggio r.
Poligono = function (n,r)
{t=seq(0,2*pi,length=n+1)
Cerchio(t,r)}
```

```
# Punti della retta tra z1 e z2
# corrispondenti ai parametri t.
Retta = function (t,z1,z2)
{z1+t*(z2-z1)}
```

Geometria

```
# Proiezione di z sulla retta z0+tw.
Proi = function (z,z0,w)
{t=Scalare(z-z0,w)/Scalare(w,w)
z0+t*w}
```

```
# Riflessione di z nel punto m.
Riflpunto = function (z,m)
{2*m-z}
```

```
# Riflessione di z nella retta z0+tw.
Riflretta = function (z,z0,w)
{Riflpunto(z,Proi(z,z0,w))}
```

```
# Rotazione di z per alfa (in gradi)
# attorno al centro.
Rot = function (z,alfa,centro=0i)
{centro+exp(1i*alfa*CMPI180)*(z-centro)}
```

```
# Prodotto scalare di punti del piano.
Scalare = function (z,w) {Re(z*Conj(w))}
```

Grafica

```
Grafica = function (x,y,cornice=F,
rxy=1,assi=FALSE,marg=0)
{par(mai=rep(marg,4), oim=c(0,0,0,0), lwd=0.5);
plot(x,y,type="n",xlab="",ylab="",
asp=rxy,axes=assi,frame.plot=cornice)}
```

```
Livelli = function (x,y,f,
valori=NULL,livelli=0,
scritte=1,colore=1,spessore=0.4)
{if (identical(valori,NULL))
valori=outer(x,y,f);
contour(x,y,valori,levels=livelli,
col=colore,lwd=spessore,
labcex=0.6*scritte,
drawlabels=(scritte>0),add=T)}
```

```
Punti = function (x,simb,tip="p",
colore=1,coloreri=NA,scala=1)
{points(Re(x),Im(x),pch=simb,type=tip,
col=colore,bg=coloreri,cex=scala)}
```

Grafica-strumenti

```
Postscript = function (file,large,alt)
{postscript(file,width=large/2.54,
height=alt/2.54, horizontal=FALSE,
onefile=FALSE, paper="special")}
```

Matrici

```
Autovalori = function (A)
{eigen(A,only.values=T)$values}
```

```
Autovettori = function (A)
{eigen(A)$vectors}
```

```
Gershgorin = function (A,i)
{centro=A[i,i]
raggio=sum(abs(A[i,]))-abs(centro)
c(centro,raggio)}
```

```
Matrice = function
(dati, righe=NULL, col=NULL)
{if (!is.null(righe))
matrix(dati,nrow=righe,byrow=T)
else if (!is.null(col))
matrix(dati,ncol=col)
else NULL}
```

```
Traccia = function (A)
{if (length(A)>1) sum(diag(A))
else c(A)}
```

Primi

```
# Pagina 40.
Eratostene = function (n)
{v=2:n; u=c(); r=sqrt(n)
repeat
{p=v[1]; if(p>r) break;
u=c(u,p)
v=v[v%%p>0]}
c(u,v)}
```