

# PCTO L<sup>A</sup>T<sub>E</sub>X

Grazia Messineo — Salvatore Vassallo

Maggio - Giugno 2026

# Indice

|          |                                                                        |           |
|----------|------------------------------------------------------------------------|-----------|
| <b>1</b> | <b>T<sub>E</sub>X e L<sup>A</sup>T<sub>E</sub>X</b>                    | <b>4</b>  |
| 1.1      | Cosa sono . . . . .                                                    | 4         |
| 1.2      | Vantaggi e svantaggi . . . . .                                         | 5         |
| 1.3      | Come funziona L <sup>A</sup> T <sub>E</sub> X? . . . . .               | 6         |
| <b>2</b> | <b>Il testo</b>                                                        | <b>8</b>  |
| 2.1      | La struttura di un file L <sup>A</sup> T <sub>E</sub> X . . . . .      | 8         |
| 2.1.1    | Il preambolo . . . . .                                                 | 8         |
| 2.1.2    | La classe di un documento . . . . .                                    | 8         |
| 2.1.3    | I pacchetti . . . . .                                                  | 9         |
| 2.1.4    | La compilazione . . . . .                                              | 10        |
| 2.2      | Il documento . . . . .                                                 | 11        |
| 2.2.1    | Struttura del testo . . . . .                                          | 12        |
| 2.2.2    | Comandi . . . . .                                                      | 14        |
| 2.2.3    | Le parti di un documento . . . . .                                     | 24        |
| 2.2.4    | Ambienti . . . . .                                                     | 25        |
| 2.2.5    | Le figure . . . . .                                                    | 35        |
| 2.2.6    | Le scatole . . . . .                                                   | 38        |
| 2.2.7    | Il codice . . . . .                                                    | 40        |
| 2.2.8    | La gestione della pagina . . . . .                                     | 41        |
| 2.2.9    | Le macro di L <sup>A</sup> T <sub>E</sub> X . . . . .                  | 42        |
| <b>3</b> | <b>L<sup>A</sup>T<sub>E</sub>X e la matematica</b>                     | <b>44</b> |
| 3.1      | Le modalità matematiche . . . . .                                      | 44        |
| 3.1.1    | L'ambiente <code>math</code> . . . . .                                 | 44        |
| 3.1.2    | L'ambiente <code>displaymath</code> . . . . .                          | 44        |
| 3.2      | I principali operatori matematici . . . . .                            | 45        |
| 3.3      | Le funzioni . . . . .                                                  | 46        |
| 3.4      | Lettere greche . . . . .                                               | 47        |
| 3.5      | Operatori, Relazioni, Frecce, Delimitatori, Puntini ed altri simboli . | 47        |
| 3.6      | Parentesi . . . . .                                                    | 47        |
| 3.7      | Testo all'interno di una formula . . . . .                             | 48        |
| 3.8      | Modifiche della spaziatura . . . . .                                   | 49        |
| 3.9      | Modificare lo stile ed il corpo del testo . . . . .                    | 49        |
| 3.10     | Gli ambienti della Matematica . . . . .                                | 49        |
| 3.10.1   | L'ambiente <code>array</code> . . . . .                                | 50        |
| 3.10.2   | Matrici . . . . .                                                      | 50        |
| 3.10.3   | Casi . . . . .                                                         | 50        |
| 3.10.4   | Gli altri ambienti . . . . .                                           | 51        |

|           |                                                       |           |
|-----------|-------------------------------------------------------|-----------|
| 3.11      | Teoremi, definizioni...                               | 53        |
| 3.11.1    | Stili dei teoremi                                     | 56        |
| 3.11.2    | Dimostrazioni                                         | 57        |
| <b>4</b>  | <b>L<sup>A</sup>T<sub>E</sub>X e le altre scienze</b> | <b>59</b> |
| <b>5</b>  | <b>Il sezionamento</b>                                | <b>59</b> |
| 5.1       | Le parti di un documento                              | 59        |
| 5.1.1     | Altri sezionamenti                                    | 60        |
| <b>6</b>  | <b>Gli oggetti flottanti</b>                          | <b>63</b> |
| <b>7</b>  | <b>I riferimenti incrociati</b>                       | <b>66</b> |
| <b>8</b>  | <b>La bibliografia</b>                                | <b>67</b> |
| 8.1       | L'ambiente thebibliography                            | 68        |
| <b>9</b>  | <b>L'indice</b>                                       | <b>70</b> |
| <b>10</b> | <b>Il risultato</b>                                   | <b>71</b> |
| <b>11</b> | <b>Le presentazioni</b>                               | <b>74</b> |
| 11.1      | beamer                                                | 74        |
| 11.1.1    | Il preambolo                                          | 74        |
| 11.1.2    | Il corpo del documento                                | 76        |
| 11.1.3    | Tabelle, figure, suoni ed animazioni                  | 78        |
| <b>12</b> | <b>La grafica</b>                                     | <b>79</b> |
| 12.1      | PSTricks                                              | 79        |
| 12.1.1    | Concetti introduttivi                                 | 81        |
| 12.1.2    | I colori                                              | 81        |
| 12.1.3    | L'ambiente pspicture                                  | 82        |
| 12.1.4    | La griglia                                            | 82        |
| 12.1.5    | I punti                                               | 83        |
| 12.1.6    | Le linee                                              | 83        |
| 12.1.7    | Stili e riempimenti                                   | 84        |
| 12.1.8    | Le forme geometriche di base                          | 85        |
| 12.1.9    | Oggetti matematici                                    | 86        |
| 12.1.10   | I nodi                                                | 87        |
| 12.1.11   | Altri effetti                                         | 88        |
| 12.2      | TikZ e PGF                                            | 88        |
| 12.2.1    | Rettangoli, cerchi, ellissi, parabole                 | 90        |

|           |                               |           |
|-----------|-------------------------------|-----------|
| 12.2.2    | Grafici di funzioni . . . . . | 92        |
| 12.2.3    | Nodi e testo . . . . .        | 93        |
| 12.2.4    | E inoltre ... . . . .         | 95        |
| <b>13</b> | <b>Per saperne di più</b>     | <b>95</b> |

# 1 $\text{\TeX}$ e $\text{\LaTeX}$

## 1.1 Cosa sono

$\text{\LaTeX}$  è un linguaggio di “markup” usato per la preparazione di testi, basato sul programma di composizione tipografica  $\text{\TeX}$ .

$\text{\TeX}$  è stato scritto dal matematico ed informatico Donald Knuth (si veda ad esempio [https://it.wikipedia.org/wiki/Donald\\_Knuth](https://it.wikipedia.org/wiki/Donald_Knuth)) nel 1978 per scrivere matematica per la seconda edizione del suo lavoro più importante *The Art of Computer Programming*<sup>1</sup>.

$\text{\TeX}$  è oggi il solo programma di composizione tipografica che include strumenti di formattazione del testo matematico, scritto e impaginato esattamente come i matematici lo vogliono.

Il numero di versione di  $\text{\TeX}$  tende a  $\pi$  e attualmente (3 maggio 2024) è 3.141592653.

$\text{\LaTeX}$  è stato scritto dal matematico ed informatico Leslie Lamport (si veda ad esempio [https://it.wikipedia.org/wiki/Leslie\\_Lamport](https://it.wikipedia.org/wiki/Leslie_Lamport)) nel 1985 e contiene i comandi per un utilizzo comodo di  $\text{\TeX}$  insieme a delle impostazioni predefinite per l’impaginazione del testo.

*Osservazione 1.* La pronuncia del nome non è “tex”, ma “tek” (o meglio “tech” con il “ch” come in “Bach”), poiché l’origine del nome  $\text{\TeX}$  deriva dalla radice della parola greca τέχνη (téchnē), che significa “arte, tecnica, tecnologia”. La terminazione del nome è la lettera greca chi maiuscola (X) la cui grafia si confonde con quella della latina “ics” (X).

Knuth ha reso gratuitamente disponibile il programma, quindi per molti anni non c’è stata alcuna forma di pubblicità commerciale che lo facesse conoscere al di fuori dell’ambito tecnico o scientifico.

Chiariamo le differenze tra i due prodotti principali,  $\text{\TeX}$  e  $\text{\LaTeX}$ :

- $\text{\TeX}$  è un programma di composizione tipografica. Implementa un linguaggio di programmazione per la tipografia (basato su macro) che svolge circa 300 operazioni di base.
- $\text{\LaTeX}$  è una interfaccia per  $\text{\TeX}$ , progettata per automatizzare tutte le operazioni necessarie per la preparazione di un documento. Consente agli autori e agli impaginatori di utilizzare la potenza di  $\text{\TeX}$  senza dover studiare a fondo l’intero linguaggio.

---

<sup>1</sup>Circolano voci che Knuth abbia iniziato il progetto  $\text{\TeX}$  perché stanco di vedere i suoi manoscritti maltrattati dall’American Mathematical Society e lo abbia scritto dopo aver visto le prove di stampa del libro

Per usare  $\text{\LaTeX}$  è necessario, ovviamente, un computer, qualsiasi computer con qualsiasi sistema operativo: Microsoft®Windows™, Apple®Mac OS, Linux, Android, FreeBSD, Solaris™, ... e anche senza installarlo usando siti dedicati.

Esistono alcune distribuzioni  $\text{\TeX}$ , ma noi faremo riferimento alla distribuzione  $\text{\TeX}$ Live (<http://tug.org/texlive>) che esiste per tutti i sistemi operativi (per Apple esiste una sua derivazione,  $\text{\MacTeX}$  <http://tug.org/mactex>).

Noi faremo riferimento sempre a  $\text{\TeX}$  e  $\text{\LaTeX}$ , ma in realtà esistono diverse varianti di questi programmi, ad esempio attualmente per la produzione di testi vengono usati soprattutto i programmi  $\text{\pdfLaTeX}$ ,  $\text{\XeLaTeX}$ ,  $\text{\LuaLaTeX}$  che si basano su estensioni di  $\text{\TeX}$  che si chiamano  $\text{\pdfTeX}$ ,  $\text{\XeTeX}$ ,  $\text{\LuaTeX}$ .

Ognuna di queste varianti ha le sue specificità, ma noi non ce ne occuperemo in questa sede.

## 1.2 Vantaggi e svantaggi

I vantaggi nell'uso di  $\text{\LaTeX}$  possono essere così riassunti:

1. il file che si scrive in  $\text{\LaTeX}$  è un normale file di testo, che può essere scritto con un qualsiasi editor (ad esempio Blocco Note, WordPad, ecc.) ed è quindi “eterno”<sup>2</sup>;
2. è un programma multiplatforma;
3. chi scrive non si deve preoccupare della forma, ma solo del contenuto. Alla forma pensa  $\text{\LaTeX}$ ;
4. le funzioni complesse, come la realizzazione di un indice, della bibliografia o dell'indice analitico, sono difficili da realizzare con la maggior parte degli altri editor, mentre basta un solo comando in  $\text{\LaTeX}$ ;
5.  $\text{\LaTeX}$  è particolarmente adatto ai testi che contengono formule matematiche;
6.  $\text{\LaTeX}$  è estendibile (gli si possono aggiungere nuove funzioni);
7. è un linguaggio che usa tag, quindi la trasformazione dei documenti  $\text{\LaTeX}$  in altri formati (ad es. html) è più semplice. L'apprendimento di  $\text{\LaTeX}$  parallelamente ad altri linguaggi con tag risulta più semplice;
8. Migliaia di persone in tutto il mondo usano  $\text{\LaTeX}$  e sono attivi vari gruppi di discussione; molti utenti sono stati in grado di risolvere problemi particolari e hanno messo a disposizione le soluzioni sotto forma di pacchetti aggiuntivi;

---

<sup>2</sup>Noi presumiamo che un qualsiasi computer o device analogo sia sempre in grado di leggere un file di testo

9. il motore  $\text{T}_{\text{E}}\text{X}$  sottostante a  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  ha un linguaggio di programmazione Turing-completo;
10. è gratis.

Gli svantaggi, invece, sono:

1. non si vede il documento così come verrà stampato mentre lo si scrive.  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  non è *WYSIWYG* (*What You See Is What You Get*), contrariamente ad altri editor. Per visualizzare il documento bisogna compilarlo con  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ , poi utilizzare un programma di visualizzazione (In realtà questo è un vantaggio. Infatti un file compilato  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  avrà sempre lo stesso aspetto su qualsiasi schermo, con qualsiasi stampante, con qualsiasi computer/sistema operativo. Un file scritto con altri editor appare uguale tra stampa e video, ma diverso su due PC diversi, o anche sullo stesso PC con 2 stampanti diverse collegate);
2. Richiede di intercalare il testo con comandi di composizione che rispecchiano la struttura logica del testo;
3. non è un ambiente integrato con foglio di lavoro, posta elettronica, calendario degli appuntamenti e agenda degli indirizzi;
4. iniziare a lavorare in  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  è più difficile che con altri editor “visuali”;
5. non è intuitiva la gestione di oggetti quali le figure o di attributi quali i caratteri;
6. non permette di gestire pagine più grandi di 33 metri quadrati.

### 1.3 Come funziona $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ?

**Una premessa “filosofica”**  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  prevede che un documento sia strutturato in *unità logiche*: molti comandi di  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  descrivono quello che si chiama la struttura logica del documento, cioè il documento stesso, i capitoli, le sezioni, le sottosezioni, le tabelle, ecc..  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  è concepito in modo che l’utente sia concentrato prevalentemente sul testo e non sulla struttura logica, della quale si preoccupa il programma, cioè invita l’utente a concentrarsi sul contenuto anziché sulla forma.

Ad esempio ci sono comandi per definire i capitoli (in un libro), le sezioni, le sottosezioni, ecc. ed è poi  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  che decide sull’aspetto grafico del testo: il font e le dimensioni da usare per i titoli dei capitoli, delle sezioni, delle sottosezioni e la loro numerazione, come dividere in sillabe le parole, dove spezzare le righe in modo da avere la giustificazione del margine destro. . . . L’utente deve intervenire sulla forma solo in casi rari.

**Lavorare con L<sup>A</sup>T<sub>E</sub>X** L<sup>A</sup>T<sub>E</sub>X produce il suo output da un file di testo (sorgente) solitamente con estensione `tex` mediante una *compilazione* con un programma (`pdflatex`, `xelatex`, `lualatex`): l'output è (di solito) un file `pdf` che può essere stampato o visualizzato a schermo.

Quando si compila un documento, L<sup>A</sup>T<sub>E</sub>X genera sempre oltre al file di output altri due file:

- \* un file `.log`, che contiene una copia dei messaggi che appaiono durante la compilazione (utile, per esempio, quando vengono rilevati errori di compilazione);
- \* un file `.aux`, che contiene gli eventuali riferimenti incrociati (bibliografia, note, ecc.).

A seconda della complessità del documento, possono essere creati anche altri file:

- \* il file `.toc`, che contiene l'indice;
- \* il file `.lot`, che contiene la lista delle tabelle;
- \* il file `.lof`, che contiene la lista delle figure;
- \* il file `.idx`, che contiene l'indice analitico non formattato.

Per lavorare in L<sup>A</sup>T<sub>E</sub>X sul proprio pc sono necessari, come detto, una distribuzione T<sub>E</sub>X (per noi T<sub>E</sub>XLive) e un editor di testo, meglio se specializzato come, per esempio, T<sub>E</sub>Xstudio, disponibile per ogni piattaforma.

È possibile però (ed è quello che faremo) utilizzare L<sup>A</sup>T<sub>E</sub>X completamente online su alcune piattaforme gratuite e più che sufficienti per un uso non professionale:

- *Cocalc* (<https://cocalc.com/>) che non è solo un editor per T<sub>E</sub>X, ma una vera e propria piattaforma, utilizzabile in modo collaborativo, con programmi di calcolo, di statistica e la possibilità di utilizzare un ambiente completo Linux
- *Overleaf* (<https://it.overleaf.com/>) che permette l'utilizzo di un editor “user friendly” e di una distribuzione T<sub>E</sub>X pressoché completa.

Noi useremo *Overleaf*. Si comincia registrandosi con una mail o con la propria utenza Google. A questo punto si può creare un progetto cioè il nostro primo documento L<sup>A</sup>T<sub>E</sub>X. Scegliamo un progetto bianco.



## 2 Il testo

### 2.1 La struttura di un file $\text{\LaTeX}$

Un file  $\text{\LaTeX}$  è sempre formato da due parti: il *preambolo* e il *documento* vero e proprio

#### 2.1.1 Il preambolo

Tutti i file di input che devono essere elaborati da  $\text{\LaTeX}$  cominciano con il comando:

```
\documentclass[opzioni]{classe del documento}
```

Questo comando è indispensabile e deve sempre comparire nella prima linea di tutti i documenti  $\text{\LaTeX}$ <sup>3</sup>.

Definisce la *classe* del documento e le *opzioni* da applicare (se si devono inserire più opzioni, occorre separarle con una virgola). Il preambolo serve inoltre a caricare le *estensioni* (*package*), che servono ad aggiungere nuove funzionalità a  $\text{\LaTeX}$ . Esse vengono caricate tramite il comando:

```
\usepackage[opzioni]{nome del package}
```

#### 2.1.2 La classe di un documento

La classe, che è il parametro obbligatorio del comando *documentclass*, definisce il tipo del documento. Le quattro classi principali sono:

- \* **article** è la classe utilizzata per i documenti più corti (articoli per riviste scientifiche...);
- \* **report** è utilizzata per i documenti più lunghi che devono essere divisi in capitoli (mentre la classe *article* usa, di default, sezioni e sottosezioni, non capitoli);
- \* **book** è utilizzata per i libri (ci sono poche differenze rispetto alla classe **report** questa classe, se non è diversamente specificato, produce documenti con pagine destre speculari rispetto alle sinistre, per es. per quanto riguarda i margini, e con diversi titoletti di pagina);
- \* **letter** permette di scrivere delle lettere.

---

<sup>3</sup>In realtà nella prima riga possono apparire altre cose come delle *direttive* che sono precedute da “% !”, o dei comandi di controllo.

Esistono poi molte altre classi definite da pacchetti, come, ad esempio `beamer` usata per le presentazioni, `exam` usata per comporre esami, `memoir`, `Komascript` ed altre ancora che sostituiscono le classi principali di  $\text{\LaTeX}$ .

**Le opzioni di classe** Le opzioni di classe, che sono i parametri opzionali del comando `documentclass`, servono a modificare l’impaginazione del documento. Le due principali sono:

- ★ `10pt`, `11pt`, `12pt` porta la grandezza dei caratteri, rispettivamente, a dieci, undici o dodici punti (la grandezza normale, senza opzioni, è di dieci punti).
- ★ `twocolumn` formatta il documento in modo che il testo sia su due colonne;

### 2.1.3 I pacchetti

I pacchetti servono a  $\text{\LaTeX}$  per svolgere alcune operazioni che da “solo” non farebbe e, soprattutto, per facilitare le operazioni di formattazione del testo. La sintassi è la seguente:

$$\backslash\text{usepackage}[opzioni]\{nome\ del\ pacchetto\}$$

Le *opzioni* sono facoltative o quasi, invece il *nome del pacchetto* no. I *pacchetti* essenziali per scrivere in *italiano* e alcune altre lingue europee sono i seguenti:

```
\usepackage[italian]{babel}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
```

Il pacchetto `babel` con l’opzione `italian` permette a  $\text{\LaTeX}$  la divisione delle parole a fine riga e di scrivere in italiano: “Capitolo”, “Indice”, ecc. al posto di “Chapter”, “Index”, ecc. `babel` non traduce automaticamente, invece, espressioni tipo *Teorema*, *Corollario*, ecc., che devono quindi essere tradotte “a mano”.

I pacchetti `inputenc` e `fontenc` sono tecnici e riguardano l’input e l’output dei caratteri

Le funzioni del pacchetto `babel` non si limitano alla traduzione e alla sillabazione. Per scoprire tutte le altre funzioni del pacchetto, si può consultare l’ampia documentazione ad esso relativa<sup>4</sup>.

Alcuni altri pacchetti che possono risultare utili nella scrittura di documenti  $\text{\LaTeX}$  sono:

---

<sup>4</sup>Tutti i pacchetti  $\text{\LaTeX}$  sono corredati di manuali che ne spiegano il funzionamento, i comandi e le opzioni. Tale documentazione viene scaricata insieme al pacchetto ed installata, di solito, nella cartella “doc”. Usando il terminale o la riga di comando la documentazione appare digitando `texdoc "nomepacchetto"`.

| Pacchetto                                                                      | Funzioni                                                                                                                                                         |
|--------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>geometry</code> :                                                        | è un pacchetto utile per ridefinire i margini delle pagine (alto - basso -sinistro - destro)                                                                     |
| <code>amsmath</code> :                                                         | fornisce diversi strumenti che permettono di gestire meglio la struttura dei documenti che contengono formule matematiche (ad esempio integrali, matrici, ecc.). |
| <code>amsthm</code> :                                                          | permette di gestire teoremi, corollari, dimostrazioni, ecc.                                                                                                      |
| <code>amssymb</code> :                                                         | fornisce diversi simboli matematici                                                                                                                              |
| <code>enumerate</code> ed<br><code>enumitem</code> :                           | consentono di creare elenchi puntati e numerati, o liste descrittive diverse da quelle standard, in modo semplice                                                |
| <code>fancyhdr</code> :                                                        | consente di gestire intestazioni e piè di pagina (numerazione delle pagine, ecc.)                                                                                |
| <code>xcolor</code> :                                                          | è un pacchetto per la gestione (di base e avanzata) dei colori                                                                                                   |
| <code>beamer</code> :                                                          | è un pacchetto che consente di realizzare presentazioni (da proiettare) e/o lucidi (da stampare e proiettare)                                                    |
| <code>hyperref</code> :                                                        | serve a gestire i collegamenti ipertestuali nei file pdf creati da $\text{\LaTeX}$                                                                               |
| <code>graphicx</code> :                                                        | permette di utilizzare grafici all'interno di $\text{\LaTeX}$                                                                                                    |
| <code>pstricks</code> , <code>tikz</code> e<br>tutti i pacchetti<br>collegati: | permettono la realizzazione di grafici e disegni.                                                                                                                |

Tabella 1: Principali package di  $\text{\LaTeX}$

#### 2.1.4 La compilazione

Una volta scritto il documento  $\text{\TeX}$  è necessario *compilarlo* con uno dei programmi di composizione: `pdf $\text{\LaTeX}$` , `X $\text{\LaTeX}$` , `Lua $\text{\LaTeX}$` , ...

Noi facciamo riferimento a `pdf $\text{\LaTeX}$` , ma per gli altri programmi la procedura è simile.

Come si compila? Usando un editor per  $\text{\LaTeX}$  di solito c'è un'icona o una voce di menu per lanciare la compilazione, su *Overleaf* c'è un pulsante.

Se la compilazione va a buon fine viene prodotto un file **pdf** che, spesso viene visualizzato automaticamente. Con molti editor c'è la possibilità di sincronizzare il file **tex** e il file **pdf** cioè ci si può spostare sul file **pdf** al punto corrispondente del file **tex** e viceversa.

E se la compilazione non va a buon fine? In questo caso viene prodotto un messaggio d'errore che, se non è visibile direttamente a schermo, può essere letto sul file **log** che il programma produce. I messaggi d'errore di  $\text{\TeX}$  sono a volte criptici, ma spesso chiariscono il problema e indicano la riga del file **tex** dove la compilazione si è fermata (l'errore potrebbe non essere esattamente in quella riga). Gli errori più comuni sono banali errori di battitura: se io scrivo `\textbg` anziché `\textbf`,  $\text{\TeX}$  non è in grado di interpretare il comando e si interrompe, spesso un ambiente viene aperto, ma non viene chiuso o, analogamente, viene aperta una parentesi e poi non viene chiusa.

Il consiglio è quello di compilare spesso il documento così che eventuali errori si possono trovare solo in una piccola porzione di testo.

## 2.2 Il documento

Il documento vero e proprio è contenuto tra i comandi:

```
\begin{document}
```

e

```
\end{document}
```

Tutto ciò che è scritto dopo il secondo comando viene ignorato.

Il testo in un documento  $\text{\LaTeX}$  può essere completamente scritto utilizzando solo i caratteri con codici **ASCII** tra 32 e 126, cioè quelli mostrati in Figura 1. Tutti gli altri caratteri si possono ottenere mediante comandi. Questa è l'impostazione originaria di  $\text{\TeX}$ . Attualmente è possibile inserire qualsiasi carattere da tastiera e passare agevolmente da una lingua a un'altra, ma è molto spesso più comodo, se non si devono usare contemporaneamente due lingue, utilizzare i comandi per inserire lettere diverse da quelle della tastiera italiana.

! " # \$ % & ' ( ) \* + , - . /  
 0 1 2 3 4 5 6 7 8 9 : ; < = > ?  
 @ A B C D E F G H I J K L M N O  
 P Q R S T U V W X Y Z [ \ ] ^ \_  
 ` a b c d e f g h i j k l m n o  
 p q r s t u v w x y z { | } ~

Figura 1: I caratteri ASCII da 32 a 128

### 2.2.1 Struttura del testo

**Caratteri speciali e primi comandi** Alcuni caratteri hanno un significato speciale per  $\text{\LaTeX}$ . Ad esempio `\` indica un comando, `~` uno spazio insecabile e `%` un commento. Una delle fonti di errore più frequenti è scrivere nel testo un carattere speciale senza l'opportuno comando. Nella Tabella 2 sono raccolti tali caratteri, il loro significato ed i comandi che permettono di farli apparire nel testo (anziché usarli come comandi):

| Carattere          | Effetto                 | Comando                       |
|--------------------|-------------------------|-------------------------------|
| <code>~</code>     | spazio insecabile       | <code>\textasciitilde</code>  |
| <code>&amp;</code> | tabulazione             | <code>\&amp;</code>           |
| <code>#</code>     | argomento di funzioni   | <code>\#</code>               |
| <code>_</code>     | indice nelle formule    | <code>\_</code>               |
| <code>\$</code>    | modo matematico         | <code>\\$</code>              |
| <code>\</code>     | comando $\text{\LaTeX}$ | <code>\textbackslash</code>   |
| <code>%</code>     | commento                | <code>\%</code>               |
| <code>{</code>     | inizio gruppo           | <code>\{</code>               |
| <code>^</code>     | esponente nelle formule | <code>\textasciicircum</code> |
| <code>}</code>     | fine gruppo             | <code>\}</code>               |

Tabella 2: Caratteri speciali

Come detto il carattere `(\)` introduce un comando. La barra può essere seguita

da uno o più caratteri o da una stringa di caratteri e uno o più argomenti (racchiusi tra parentesi graffe) come si vede nella Tabella 3.

| Comando                         | Effetto (descrizione)                                                     |
|---------------------------------|---------------------------------------------------------------------------|
| <code>\today</code>             | 21 maggio 2026 (data odierna)                                             |
| <code>\LaTeX</code>             | L <sup>A</sup> T <sub>E</sub> X (logo di L <sup>A</sup> T <sub>E</sub> X) |
| <code>\TeX</code>               | T <sub>E</sub> X (logo di T <sub>E</sub> X)                               |
| <code>\ldots</code>             | ... (puntini di sospensione)                                              |
| <code>\_</code>                 | (uno spazio)                                                              |
| <code>\textit{argomento}</code> | <i>Scrivo l'argomento in italico</i>                                      |
| <code>\textbf{argomento}</code> | <b>Scrivo l'argomento in "grassetto"</b>                                  |
| <code>\Large{argomento}</code>  | Scrivo più grande                                                         |
| <code>\hspace{2cm}</code>       | (Uno spazio orizzontale di 2cm)                                           |

Tabella 3: Alcuni esempi di comando in L<sup>A</sup>T<sub>E</sub>X

## Altri caratteri particolari

- *virgolette*: nella tastiera italiana non esistono, occorre quindi inserirle con dei comandi. Le *virgolette semplici* si ottengono con la coppia ‘ (accento aperto = ascii 096, che non esiste sulla tastiera italiana) e ’ (apostrofo). Si può anche usare `\lq` (left quote) per la virgoletta aperta e `\rq` (right quote) per quella chiusa. Le *virgolette doppie* si ottengono invece con i comandi `""` (doppie virgolette 2 volte) per aprirle e `''` (2 apostrofi) per chiuderle (se si usa `babel`); oppure `\lq\lq` e `\rq\rq` (che stanno per left quotation e right quotation).

Molti editor hanno facilitazioni per la scrittura di virgolette, il più comune è quello di usare le normali virgolette doppie (sopra il “2”) e singole.

- *Accenti*: L<sup>A</sup>T<sub>E</sub>X consente l’uso degli *accenti* e dei *caratteri speciali* usati in molte lingue. Ad esempio:

```
H\^otel, na\"i ve, {\L}odz\\
sm\o rrebr\o d, iSe\~norita!,\\
Sch\"onbrunner Schlo\ss{} Stra\ss e.
```

producono:

Hôtel, naïve, Łodz

smørrebrød, ¡Señorita!,  
 Schönbrunner Schloß Straße.

Molti editor hanno a disposizione barre per l’inserimento facilitato di questi caratteri.

- *Puntini di sospensione*: esistono dei comandi appositi, che per il testo sono

`\ldots ...`                      `\dots ...`

- *Trattini*: in  $\text{\LaTeX}$  si utilizzano tre tipi di trattini: quello singolo - (trattino in un’unica parola, ad es. e-mail), quello doppio -- (range di numeri, ad es. 1–4) e quello triplo --- (usato per gli incisi).
- *Altri caratteri*: ci sono altri caratteri che “sembrano” normali, ma non lo sono. Ad esempio i segni di minore e maggiore daranno errore se usati in modalità testo: bisogna usare `\textless` e `\textgreater` per ottenere  $<$  e  $>$ .

## 2.2.2 Comandi

In  $\text{\LaTeX}$  qualsiasi operazione si voglia fare sul testo viene svolta mediante un *comando*

---

Questo testo è `\textit{italico}`, mentre questo è `\textbf{grassetto}` è questo è `\texttt{a larghezza fissa}`.

Posso anche scrivere `\small piccolo`, o `\footnotesize più piccolo` o `\tiny piccolissimo`.

Ma si può anche `\large ingrandire` anche `\huge di più`.

---

Questo testo è *italico*, mentre questo è **grassetto** è questo è a larghezza fissa.

Posso anche scrivere piccolo, o più piccolo o piccolissimo

Ma si può anche ingrandire anche di più.

Ci sono tre tipi di comandi, per quanto riguarda la forma:

- un singolo carattere
  - spazio \_
  - tilde ~
  - circonflesso ^
  - linea bassa \_
- barra rovescia e un carattere non alfabetico (simboli di controllo) come `\$`

- barra rovescia e una successione di caratteri alfabetici (parole di controllo) come `\textbf`

*Maiuscolo e minuscolo sono **diversi** nelle parole di controllo*

## Lo spazio

- più spazi contano come uno solo
- un fine riga conta come uno spazio
- una riga vuota conta come fine capoverso
- sa a fine riga scrivo `%`, allora non conta come spazio
- gli spazi dopo i comandi del terzo tipo sono *ignorati!* infatti scriviamo `\LaTeX\` oppure `\LaTeX{ }` e non `\LaTeX_`

**Tilde** La tilde `~` indica a  $\text{\LaTeX}$  di lasciare uno spazio che non può essere usato per andare a capo

Dante Alighieri scrisse ```La~Divina Commedia''`

Dante Alighieri scrisse “La Divina Commedia”

**Linea bassa e circonflesso** Si usano in matematica

$$\text{\$x^3+x^2-1\$} \rightarrow x^3 + x^2 - 1$$

$$\text{\$\log_{2} 8=3\$} \rightarrow \log_2 8 = 3$$

**Simboli di controllo** Sono della forma “barra rovescia e un carattere non alfabetico” cioè un carattere che non è né una lettera né un numero

`\!` `\'` `\%` `\(` `\)` `\=` `\^` `\'` `\*` `\``  
`\[` `\]` `\|` `\@` `\` `\,` `\;` `\:` `\-` `\_`  
`\_` `\&` `\/` `\$` `\{` `\}` `\.`

Alcuni li abbiamo già visti come `\'` o `\$`, altri li vedremo più avanti  
 Generalmente servono a modificare il carattere successivo o a scrivere caratteri non stampabili o a produrre spazi.



**Comandi** I comandi possono essere

- dichiarazioni
- azioni

Un esempio di dichiarazione è `\Large` che ingrandisce il testo da quel punto in poi

`Ingrandiamo \Large i caratteri` → `Ingrandiamo i caratteri`

Un esempio di azione è `\textbf` che fa scrivere una parte di testo in grassetto

`Testo in \textbf{grassetto}` → `Testo in grassetto`

**Argomenti di un comando** Spesso i comandi hanno un argomento, cioè un “qualcosa” (una porzione di testo, una lunghezza, ...) a cui viene applicato il comando.

Nell'esempio di prima

- `\textbf` è il comando
- `grassetto` è l'argomento

Possono esserci più argomenti

`\setlength{\fboxrule}{1pt}`

- `\setlength` è il comando
- `\fboxrule` è il primo argomento
- `1pt` è il secondo argomento

`\setlength{\fboxrule}{1pt} \fbox{a}` → `\fbox{a}`  
`\setlength{\fboxrule}{3pt} \fbox{a}` → `\fbox{a}`

**Argomenti opzionali di un comando** Alcuni comandi hanno anche *argomenti opzionali* che vanno tra parentesi quadre

Ne abbiamo già visti:

`\usepackage[italian]{babel}`

- `\usepackage` è il comando
- `italian` è l'argomento opzionale
- `babel` è l'argomento obbligatorio

*Esempio 1.* Un `\scalebox{2}{\fbox{esempio}}` → Un

**esempio**

Un `\scalebox{2}[.5]{\fbox{esempio}}` → Un

**esempio**

Un altro `\framebox{esempio}` → Un altro

**esempio**

Un altro `\framebox[3cm]{esempio}` → Un altro

**esempio**

Un altro `\framebox[3cm][r]{esempio}` → Un altro

**esempio**

Un altro `\framebox[3cm][l]{esempio}` → Un altro

**esempio**

**Paragrafi e spaziatura** Un documento  $\text{\LaTeX}$  è formato da *paragrafi* separati l'uno dall'altro tramite una o più linee bianche. Un paragrafo è composto di *parole* separate da segni di punteggiatura o spazi. È possibile inserire quanti spazi si vogliono tra due parole (anche se ne basta uno), dato che  $\text{\LaTeX}$  li considera tutti come uno spazio unico. Un solo ritorno a capo è considerato come uno spazio. Un doppio ritorno a capo, invece, è interpretato come segno di un nuovo paragrafo. In questo caso, la prima linea è automaticamente fatta rientrare verso destra rispetto alle linee successive. Se si vuole evitare il rientro, è sufficiente utilizzare il comando `\noindent` all'inizio di ogni paragrafo, o, in alternativa, aggiungere il comando `\setlength{\parindent}{0pt}` nel preambolo<sup>5</sup>.

Esistono inoltre diversi comandi per gestire in modo puntuale la spaziatura tra parole (che in genere  $\text{\LaTeX}$  gestisce in modo automatico). Ad esempio, se si vuole impedire che due parole posizionate alla fine di una linea vengano separate, bisogna usare il carattere `~`. Questo carattere rappresenta uno *spazio insecabile*, che viene considerato come uno spazio (e quindi apparirà come tale in stampa), ma  $\text{\LaTeX}$  non separerà mai, alla fine della linea, le due parole. Ad esempio, la scrittura `Prof.~Rossi` farà in modo che il titolo e il cognome non siano mai separati.

<sup>5</sup>Questa seconda opzione è preferibile se non si vuole il rientro nell'intero documento.

**Interruzioni**  $\text{\LaTeX}$  fornisce diversi tipi di comandi per le interruzioni di pagina:

`\newpage`: inizia una nuova pagina;

`\newline` o `\\`: inizia una nuova riga senza rientro;

`\\[ampiezza]` va a capo lasciando uno spazio dell'ampiezza scelta;

*linea bianca* o `\par`: nuovo paragrafo con rientro;

`\hspace{larghezza}` e `\vspace{larghezza}`: rispettivamente spazio orizzontale e verticale;

è importante notare che uno o più spazi o tabulazioni sono considerati come un solo spazio; una sola interruzione di riga è considerata come uno spazio; lo spazio a inizio riga non viene considerato.

Nei comandi `\hspace`, `\vspace` e `\\[ampiezza]` la misura può essere espressa in:

- **mm** millimetri;
- **cm** centimetri;
- **pt** punti;

e in diverse altre unità di misura assolute e relative. Inoltre, deve essere racchiusa tra parentesi graffe (ad esempio: `\vspace{2cm}`)

**Sillabazione**  $\text{\LaTeX}$  divide le parole in sillabe automaticamente. Se l'algoritmo di sillabazione non trova i punti di divisione corretti, si può porre rimedio al problema usando i seguenti comandi per "insegnare" a  $\text{\TeX}$  come deve essere sillabata quella particolare parola. Il comando

`\hyphenation{elenco di parole}`

fa sì che le parole elencate nell'argomento siano divise solo in corrispondenza dei punti marcati da un trattino("-"). L'argomento del comando deve contenere parole costituite solo da normali lettere o per meglio dire segni che sono considerati come normali lettere nel contesto attivo. Se nel preambolo è stato caricato il pacchetto `babel` con più lingue, la sillabazione verrà fatta secondo le regole della lingua attiva in quel momento. La lingua da attivare in un certo momento può essere scelta mediante il comando

`\uselanguage{nome della lingua}`

Più parole possono essere mantenute insieme sulla stessa riga con il comando

`\mbox{testo}`

che fa in modo che il suo argomento sia mantenuto insieme in qualunque caso.

**Note**  $\text{\LaTeX}$  permette di scrivere note

- a piè di pagina
- marginali

Per scrivere note al termine del capitolo o del libro bisogna usare il pacchetto `endnotes`.

Per scrivere una nota a piè di pagina si usa il comando `\footnote`

---

```
\footnote[richiamo]{testo della nota}
```

---

Il richiamo di default (che quindi non va scritto) è un numero all'apice. Può essere anche un asterisco o altro.

Nella nota non possono essere inseriti tutti i comandi.

Le note sono testi speciali, perché vengono composte e trattenute in memoria fino a quando non viene composta la pagina nella quale esse devono comparire;  $\text{\LaTeX}$  fa il possibile affinché le note non siano spezzate fra pagine successive e che ognuna sia collocata al piede della pagina nella quale compare il richiamo.

All'occorrenza si può fare uso di un comando separato per inserire il richiamo e per scrivere il testo:

---

```
\footnotemark[richiamo]  
\footnotetext[richiamo]{testo della nota}
```

---

Questa costruzione è necessaria per inserire una nota in una tabella.

Se i margini del documento sono abbastanza ampi, possono esserci le *note a margine*. Il comando per produrle è `\marginpar` e segue la sintassi:

---

```
\marginpar[nota di sinistra]{nota di destra}
```

---

**Commenti**  $\text{\LaTeX}$  permette di includere dei *commenti* nel testo, che non verranno stampati. Possono servire come punti di riferimento per l'autore, ricordargli che non ha finito di scrivere un paragrafo, per ricordarsi a cosa servono certi pacchetti o comandi, ecc. Un commento comincia con `%` e termina alla fine della linea.

**L'allineamento del testo**  $\text{\LaTeX}$ , per default, allinea il testo giustificando i margini. Se si vuole ottenere un diverso tipo di allineamento, si possono usare i comandi della Tabella 4: oppure è possibile usarne degli altri, di cui si parlerà più avanti, trattando degli ambienti.

|                          |                             |
|--------------------------|-----------------------------|
| <code>\flushleft</code>  | allinea il testo a sinistra |
| <code>\flushright</code> | allinea il testo a destra   |
| <code>\center</code>     | allinea il testo al centro  |

Tabella 4: Tipi di allineamento

**Caratteri** Le lettere ed i simboli utilizzati per scrivere i documenti (detti *type*) sono caratterizzati da un loro *stile* (*style*) e da una *grandezza* (*size*). In  $\text{\LaTeX}$ , uno *stile* può essere indicato dagli attributi **family**, **series** e **shape**, illustrati nella tabella 5. In  $\text{\LaTeX}$ , esiste un comando particolare, `\emph`, che permette

| STILE  | Comando                         | Dichiarazione            | Nome                 | Effetto      |
|--------|---------------------------------|--------------------------|----------------------|--------------|
| FAMILY | <code>\textmd{testo}</code>     | <code>\mdseries</code>   | Serie media          | testo        |
|        | <code>\textbf{testo}</code>     | <code>\bfseries</code>   | Serie nera           | <b>testo</b> |
| FAMILY | <code>\textrm{testo}</code>     | <code>\rmfamily</code>   | Tondo con grazie     | testo        |
|        | <code>\textsf{testo}</code>     | <code>\sffamily</code>   | Lineare senza grazie | <b>testo</b> |
|        | <code>\texttt{testo}</code>     | <code>\ttfamily</code>   | Monospaziato         | <b>testo</b> |
| SHAPE  | <code>\textup{testo}</code>     | <code>\upshape</code>    | Forma diritta        | testo        |
|        | <code>\textit{testo}</code>     | <code>\itshape</code>    | Forma corsiva        | <i>testo</i> |
|        | <code>\textsl{testo}</code>     | <code>\slshape</code>    | Forma inclinata      | <i>testo</i> |
|        | <code>\textsc{testo}</code>     | <code>\scshape</code>    | Maiuscoletto         | TESTO        |
|        | <code>\textnormal{testo}</code> | <code>\normalfont</code> | Font di default      | testo        |

Tabella 5: Stili di caratteri

di distinguere il testo racchiuso tra parentesi graffe dal testo circostante. Questo significa che se il resto del testo è scritto in modo “normale”, il comando `\emph` farà sì che il testo tra parentesi sia corsivo; viceversa, se il testo attorno è in corsivo, il testo in parentesi sarà normale. Ad esempio, scrivere:

L'attributo `\emph{family}`...

produrrà come risultato:

L'attributo *family*...

mentre:

`\textit{L'attributo \emph{family}...}`

produrrà come risultato:

*L'attributo family...*

Tradizionalmente, la dimensione del carattere viene misurata in punti. Per default,  $\text{\TeX}$  produce caratteri di dimensione 10 pt. Esistono alcune *dichiarazioni* previste in  $\text{\LaTeX}$  per cambiare la dimensione. Sono riassunte nella tabella 6: È an-

|            |                                        |
|------------|----------------------------------------|
| dimensione | <code>\tiny {dimensione}</code>        |
| dimensione | <code>\scriptsize{dimensione}</code>   |
| dimensione | <code>\footnotesize{dimensione}</code> |
| dimensione | <code>\small{dimensione}</code>        |
| dimensione | <code>\normalsize{dimensione}</code>   |
| dimensione | <code>\large{dimensione}</code>        |
| dimensione | <code>\Large{dimensione}</code>        |
| dimensione | <code>\LARGE{dimensione}</code>        |
| dimensione | <code>\huge{dimensione}</code>         |
| dimensione | <code>\Huge{dimensione}</code>         |

Tabella 6: Dimensioni del carattere

che possibile mediante pacchetti opportuni avere grandezze intermedie o maggiori o minori di quelle indicate.

**Cambiare carattere** Quando si scrive in  $\text{\LaTeX}$  c'è una famiglia di caratteri usata di default *Computer Modern* nelle sue tipologie *Serif* (con grazie), *Sans Serif* (senza grazie) e *Typewriter* (monospaziata)

Si può scegliere un pacchetto che modifichi la famiglia come, ad esempio, `lmodern`, `tgbonum`, `newpertext` o `newpertext`. Un pacchetto per la scelta del carattere generalmente stabilisce tutte e tre le tipologie: ad esempio `newttext` ha come per il Serif il carattere Times, per il Sans Serif il carattere Helvetica e infine Courier come carattere Typewriter. Se possibile viene anche stabilito il carattere per la matematica, mentre a volte bisogna caricare due diversi pacchetti (ad esempio `newpertext` e `newpixmap`). Si sconsiglia vivamente di lanciarsi in esperimenti

di accoppiamenti di caratteri: stabilire quali caratteri “stanno bene” insieme non è alla portata di tutti.

Se si vuole scegliere un carattere solo per una porzione di documento la sintassi da usare è

---

```
{\fontfamily{xxx}\selectfont
  Questo testo ha un carattere diverso
}
```

---

dove `xxx` è il codice del font. Ad esempio

---

```
{\fontfamily{put}\selectfont
  Questo testo usa il carattere Utopia
}
```

---

### Questo testo usa il carattere Utopia

Se si usano Lua $\text{\LaTeX}$  o Xe $\text{\LaTeX}$  si hanno a disposizione *tutti* i font OpenType presenti sul sistema.

La gestione e la scelta dei font è però diversa e non verrà trattata qui.

**Intestazione e piè di pagina**  $\text{\LaTeX}$  accetta quattro combinazioni predefinite di *intestazione* e *piè di pagina* — i cosiddetti *stili di pagina*. Il parametro *stile* del comando

```
\pagestyle{stile}
```

definisce quale stile sarà utilizzato. Questi sono gli stili di pagina predefiniti.

**plain** stampa i numeri di pagina nella parte inferiore della stessa, al centro del piè di pagina. Questo è lo stile di pagina predefinito.

**headings** stampa il titolo del capitolo corrente e il numero di pagina sulla intestazione di ciascuna pagina, mentre il piè di pagina rimane vuoto.

**myheadings** il piè di pagina rimane vuoto, sulla intestazione di ciascuna pagina viene stampato il numero di pagina e altre informazioni indicate dall’utente

**empty** imposta la intestazione e il piè di pagina in modo che non vi sia stampato nulla.

è possibile cambiare lo stile di pagina della pagina corrente con il comando

```
\thispagestyle{stile}
```

Esistono anche alcuni pacchetti che consentono una gestione personalizzata di queste caratteristiche, ad esempio il pacchetto **fancyhdr**, che consente di cambiare in modo veloce la formattazione delle intestazioni e dei piè di pagina.

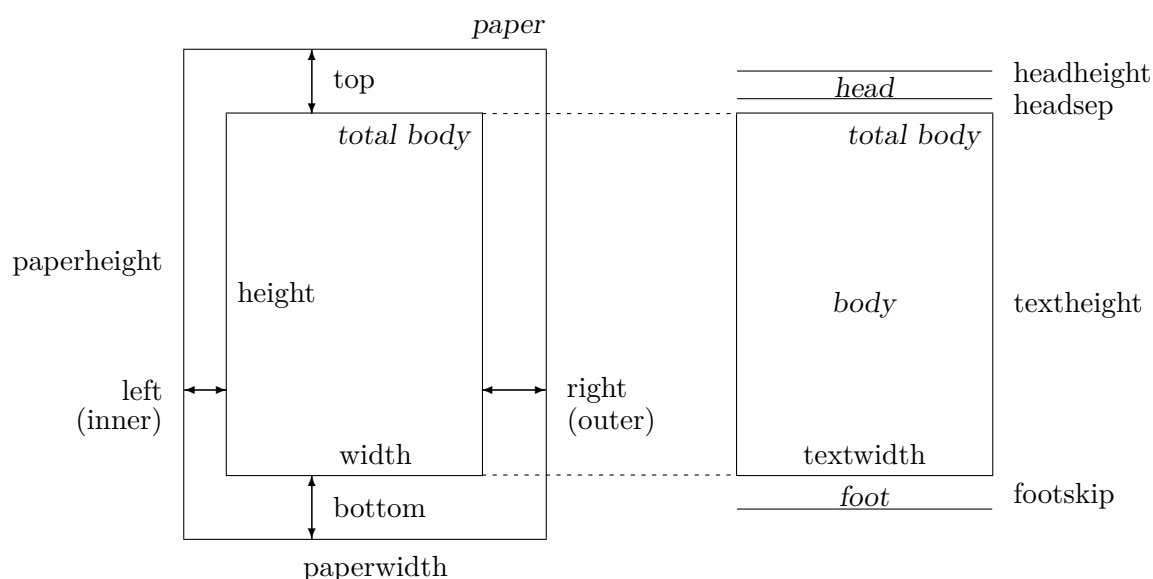


Figura 2: Nomi delle dimensioni del package `geometry`

**Il layout di pagina** Lo strumento più semplice ed al tempo stesso più efficace per cambiare il layout di pagina è il package `geometry`.

La figura 2 mostra le dimensioni usate dal pacchetto in questione. Per default, la grandezza `width` è posta uguale a `textwidth` e, analogamente, la grandezza `height` uguale a `textheight`. Con `left`, `right`, `top` e `bottom` indichiamo i margini della pagina (inner e outer sono sinonimi di left e right). Se viene specificata l'opzione `twoside`, “left” è il margine interno, “right” il margine esterno. È possibile cambiare le dimensioni dei margini attraverso il comando:

```
\geometry{margin=misura}
```

```
\geometry{hmargin=misura}
```

```
\geometry{vmargin=misura}
```

dove:

**margin** indica tutti i margini della pagina

**hmargin** indica i margini orizzontali (destra e sinistra)

**vmargin** indica i margini verticali (alto e basso)



### 2.2.3 Le parti di un documento

**Title** La sezione “Titolo” di un documento è formata in genere dal nome del documento, dai nomi degli autori e talvolta anche dalla data. Per produrre il titolo, si utilizzano i comandi:

```
\title{nome del documento}  
\author{nomi degli autori}  
\date{data}
```

nel preambolo e:

```
\maketitle
```

dopo il comando `\begin{document}`.

Se non si vuole che compaia uno dei campi del titolo (ad esempio la data), occorre lasciare vuote le parentesi graffe. Il comando `\maketitle` è indispensabile per stampare titolo, autore e data.

Alcuni pacchetti permettono di inserire altri campi nel titolo (ad es. i ringraziamenti, l’indirizzo di posta elettronica, ecc.). Di default, tutto ciò che viene stampato da questi comandi viene centrato sulla linea nella quale appare. Se il testo è troppo lungo per essere collocato in un’unica linea, viene automaticamente separato su più linee. È in ogni caso possibile decidere in quale punto il testo si deve interrompere usando il comando `\`. I nomi degli autori compaiono su un’unica linea se sono separati dal comando `\and`; compaiono invece uno sotto l’altro se sono separati da `\`. Il “titolo” viene stampato su una pagina separata se si usano le classi `book` e `report`; nella prima pagina del documento per la classe `article` (questo comportamento standard può essere modificato dalle opzioni `titlepage` o `notitlepage`).

**Abstract** Per le classi `article` e `report`, è possibile scrivere un abstract, in formato speciale, mediante il comando:

```
\begin{abstract}  
Testo dell’abstract  
\end{abstract}
```

Nella classe `report` appare sulla pagina del titolo (separata dal resto del documento) e nella classe `article` appare sotto il titolo sulla prima pagina. Il comando non è disponibile per la classe `book`.

**Suddivisione del documento** L<sup>A</sup>T<sub>E</sub>X fornisce la seguente gerarchia di *comandi* di “suddivisione” per le classi `book` e `report`:

```
\part (non disponibile per la classe article)  
\chapter (non disponibile per la classe article)
```

`\section`  
`\subsection`  
`\subsubsection`  
`\paragraph`  
`\subparagraph`

La sintassi di tutti questi comandi è `\chapter[opzioni]{titolo}`, dove “opzioni” rappresenta il titolo che appare nella testatina della pagina e che viene stampato nell’indice<sup>6</sup>. Un asterisco (\*) dopo `chapter` fa sì che i capitoli non siano numerati e che non compaiano nell’indice.

#### 2.2.4 Ambienti

Lo strumento fondamentale di  $\text{\LaTeX}$  è l’*ambiente*, con il quale si gestisce ogni tipo di formattazione. Esistono:

- ✓ **Ambienti standard:** sono ambienti definiti in tutte le classi di documenti di  $\text{\LaTeX}$  ;
- ✓ **Ambienti introdotti da specifici pacchetti:** sono quelli definiti nei package specifici che l’utente decide di utilizzare (ad esempio l’ambiente `slide` di `beamer`);
- ✓ **Ambienti definiti dall’utente:** sono quelli che chi scrive introduce, definendoli lui stesso in base alle sue esigenze.

Un generico ambiente si invoca con i comandi:

```
\begin{ambiente}[opzioni]
testo
\end{ambiente}
```

dove *ambiente* è il nome dell’ambiente. Gli ambienti possono essere chiamati diverse volte l’uno all’interno dell’altro, purché l’ordine di chiamata venga rispettato, ovvero che l’ultimo ambiente introdotto sia il primo ad essere chiuso, e così via. Inoltre, per quasi tutti gli ambienti, è possibile specificare delle opzioni, che vanno racchiuse tra parentesi quadre.

---

<sup>6</sup>Ovviamente, la parola `chapter` del comando deve di volta in volta essere sostituita dal nome della parte che si vuole introdurre. La sintassi resta invariata.

| Ambiente                 | Descrizione                                                                                                                                                                                               | Comando                                                                                     |
|--------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|
| <code>center</code>      | consente di <i>centrare</i> intere porzioni di testo                                                                                                                                                      | <code>\begin{center} ...</code><br><code>\end{center}</code>                                |
| <code>flushleft</code>   | consente di <i>allineare a sinistra</i> intere porzioni di testo                                                                                                                                          | <code>\begin{flushleft} ...</code><br><code>\end{flushleft}</code>                          |
| <code>flushright</code>  | consente di <i>allineare a destra</i> intere porzioni di testo                                                                                                                                            | <code>\begin{flushright} ...</code><br><code>\end{flushright}</code>                        |
| <code>itemize</code>     | crea elenchi puntati (il punto elenco predefinito è il pallino pieno; è possibile specificare il simbolo del punto elenco usando il pacchetto <code>enumitem</code> )                                     | <code>\begin{itemize}</code><br><code>\item... \end{itemize}</code>                         |
| <code>enumerate</code>   | crea elenchi numerati (la numerazione predefinita è in numeri arabi; usando i pacchetti <code>enumerate</code> o <code>enumitem</code> è possibile specificare che si desiderano numeri romani o lettere) | <code>\begin{enumerate}</code><br><code>\item...</code><br><code>\end{enumerate}</code>     |
| <code>description</code> | crea liste descrittive                                                                                                                                                                                    | <code>\begin{description}</code><br><code>\item...</code><br><code>\end{description}</code> |
| <code>tabular</code>     | può essere usato per comporre tabelle, con o senza linee orizzontali e verticali che separano righe e colonne.                                                                                            | <code>\begin{tabular}{formato}</code><br><code>... \end{tabular}</code> <sup>a</sup>        |

Tabella 7: I principali ambienti

<sup>a</sup>`LATEX` determina automaticamente l'ampiezza delle colonne sulla base del testo più largo contenuto in ciascuna riga. Si usa una `l` per specificare una colonna con testo allineato a sinistra, `r` per testo allineato a destra, `c` per testo centrato e `|` per una linea verticale. All'interno di un ambiente `tabular`, `&` separa gli elementi di una riga che stanno su colonne diverse, `\\` inizia una nuova riga e `\hline` inserisce una linea orizzontale. Esistono alcuni pacchetti che consentono di manipolare in modo più efficace le tabelle. Si ricordano `array` che consente di estendere con un unico comando un formato ad un'intera colonna; `longtable` che consente di gestire tabelle molto lunghe che devono essere divise su più pagine; `colortable`, che consente di avere righe, colonne o singole celle colorate.

**Gli ambienti di posizionamento del testo** Ci sono tre ambienti per posizionare il testo `center`, `flushright`, `flushleft`

Ci sono anche tre dichiarazioni che raggiungono lo stesso scopo: `\centering`, `\raggedright` e `\raggedleft`.

È opportuno usare sempre gli ambienti piuttosto che le dichiarazioni, tranne la dichiarazione `\centering` all'interno degli ambienti `figure` e `table`.

```
Testo normale
\begin{flushleft}
Questo testo è allineato a sinistra e non è giustificato quindi
l'aspetto non è bellissimo, ma è sicuramente più leggibile
di quello allineato a destra che è molto più difficile da leggere.
\end{flushleft}
Testo normale
\begin{flushright}
Questo testo è allineato a destra e non è giustificato.
Non solo l'aspetto non è bellissimo, ma è più difficile da leggere.
\end{flushright}
Testo normale
\begin{center}
Testo centrato
\end{center}
Testo normale
```

Testo normale

Questo testo è allineato a sinistra e non è giustificato quindi l'aspetto non è bellissimo, ma è sicuramente più leggibile di quello allineato a destra che è molto più difficile da leggere.

Testo normale

Questo testo è allineato a destra e non è giustificato. Non solo l'aspetto non è bellissimo, ma è più difficile da leggere.

Testo normale

Testo centrato

Testo normale

**Citare testo** Per citare del testo abbiamo tre ambienti:

- `quote`
- `quotation`
- `verse`

In tutti e tre gli ambienti il testo viene evidenziato mettendolo tra margini ristretti e modificando il carattere

Nell'ambiente `quote` possiamo inserire un solo capoverso

---

```
Il famosissimo \textit{incipit} dei \textit{Promessi Sposi} di Alessandro Manzoni
\begin{quote}
    Quel ramo del lago di Como, che volge a mezzogiorno, tra due catene non
interrotte di monti, tutto a seni e a golfi, \dots
\end{quote}
descrive in modo mirabile il ramo lecchese del Lago di Como (o Lario). Gli altri
rami sono quello di Como e quello di Gera.
```

---

Il famosissimo *incipit* dei *Promessi Sposi* di Alessandro Manzoni

Quel ramo del lago di Como, che volge a mezzogiorno, tra due catene  
non interrotte di monti, tutto a seni e a golfi, ...

descrive in modo mirabile il ramo lecchese del Lago di Como (o Lario). Gli altri  
rami sono quello di Como e quello di Gera.

Nell'ambiente `quotation` si possono inserire più capoversi

---

```
Il Vangelo di Giovanni è notevolmente diverso dagli altri tre vangeli, detti
sinottici. Già il suo inizio
\begin{quotation}
    In principio era il Verbo, il Verbo era presso Dio e il Verbo era Dio.

    Egli era in principio presso Dio: tutto è stato fatto per mezzo di lui,
    e senza di lui niente è stato fatto di tutto ciò che esiste.
\end{quotation}
indica che l'autore vuole approfondire la questione dell'identità del Cristo,
inserendo ampie digressioni teologiche.
```

---

Il Vangelo di Giovanni è notevolmente diverso dagli altri tre vangeli, detti sinottici.  
Già il suo inizio

In principio era il Verbo, il Verbo era presso Dio e il Verbo era Dio.

Egli era in principio presso Dio: tutto è stato fatto per mezzo di lui, e  
senza di lui niente è stato fatto di tutto ciò che esiste.

indica che l'autore vuole approfondire la questione dell'identità del Cristo.

L'ambiente `verse` serve per citare/scrivere poesie: alla fine di ogni verso bisogna indicare di andare a capo con il comando `\\` e ogni strofa è trattata come un capoverso, cioè deve essere lasciata una riga bianca fra una strofa e la successiva.

---

```
\begin{verse}
  Nel mezzo del cammin di nostra vita\\
  mi ritrovai per una selva oscura,\\
  ch  la diritta via era smarrita.

  Ahi quanto a dir qual era   cosa dura\\
  esta selva selvaggia e aspra e forte\\
  che nel pensier rinova la paura!
\end{verse}
```

---

|                                                                                                                                                                                                                                                  |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Nel mezzo del cammin di nostra vita<br/>mi ritrovai per una selva oscura,<br/>ch  la diritta via era smarrita.</p> <p>Ahi quanto a dir qual era   cosa dura<br/>esta selva selvaggia e aspra e forte<br/>che nel pensier rinova la paura!</p> |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

**Gli elenchi** In  $\text{\LaTeX}$  abbiamo tre tipologie di elenco

- elenco puntato (ambiente `itemize`)
- elenco numerato (ambiente `enumerate`)
- elenco descrittivo (ambiente `description`)

Ogni elemento della lista   introdotto dal comando `\item` che pu  avere un’opzione: se viene indicata l’opzione invece del contrassegno “normale” viene usato quello indicato nell’opzione. Nel caso dell’ambiente `description` l’opzione   “logicamente” necessaria poich  il senso dell’ambiente   dare una descrizione del termine usato nell’opzione.

---

```

Il mio primo elenco puntato
\begin{itemize}
\item primo
\item secondo
\item terzo
\end{itemize}
Il mio primo elenco numerato
\begin{enumerate}
\item Uno
\item Due
\item Tre
\end{enumerate}
Il mio primo elenco descrittivo
\begin{description}
\item[primo item] primo punto
\item[secondo item] secondo punto
\item[terzo item] terzo punto
\end{description}

```

---

Il mio primo elenco puntato

- primo
- secondo
- terzo

Il mio primo elenco numerato

1. Uno
2. Due
3. Tre

Il mio primo elenco descrittivo

**primo item** primo punto

**secondo item** secondo punto

**terzo item** terzo punto

Gli elenchi possono essere annidati tra di loro.

---

```

Elenchi puntati
\begin{itemize}
\item primo
\begin{itemize}
\item primo del secondo livello
\item secondo del secondo livello
\end{itemize}
\item secondo
\begin{itemize}
\item primo del terzo livello
\item secondo del terzo livello
\end{itemize}
\end{itemize}

```

```

Elenchi numerati
\begin{enumerate}
\item primo
\begin{enumerate}
\item primo del secondo livello
\item secondo del secondo livello
\end{enumerate}
\item secondo
\begin{enumerate}
\item primo del terzo livello
\item secondo del terzo livello
\end{enumerate}
\end{enumerate}

```

---

### Elenchi puntati

- primo
  - primo del secondo livello
  - secondo del secondo livello
    - \* primo del terzo livello
    - \* secondo del terzo livello
  - terzo del secondo livello
- secondo
- terzo

### Elenchi numerati

1. primo
  - (a) primo del secondo livello
  - (b) secondo del secondo livello
    - i. primo del terzo livello
    - ii. secondo del terzo livello
  - (c) terzo del secondo livello
2. secondo
3. terzo

Come si vede a seconda del livello il contrassegno è diverso. È possibile modificare il contrassegno a qualsiasi livello con comandi  $\LaTeX$ , ma il modo più comodo è utilizzare uno dei pacchetti destinati a questo scopo: il più semplice è il pacchetto `enumerate` il più completo è `enumitem`.

Ad esempio con `enumitem` scrivendo

---

```

\begin{enumerate}[label=\textbf{Esercizio \arabic*}]
\item 2+2
\item 3+3
\end{enumerate}

```

---

otteniamo

**Esercizio 1** 2+2

**Esercizio 2** 3+3



**Le tabelle** L'ambiente `tabular` è l'ambiente fondamentale per scrivere tabelle in modalità testo, il suo analogo in modalità matematica è l'ambiente `array`. Ci sono moltissimi pacchetti destinati alla scrittura delle tabelle, ma noi ci limiteremo alla descrizione dell'ambiente `tabular`.

Le tabelle (come le figure di cui ci occuperemo in seguito) possono essere “in testo” o “fuori testo”: gli oggetti “in testo” appartengono al flusso del discorso e non possono esserne scorporati senza comprometterne la comprensione, poiché hanno solitamente dimensioni maggiori del testo normale possono dare problemi di impaginazione. Per risolvere questo problema si usano spesso tabelle e figure “fuori testo” (o “mobili” o “flottanti”) lasciando svolgere a L<sup>A</sup>T<sub>E</sub>X il compito di posizionarli: gli oggetti fuori testo non appartengono al flusso del discorso e possono essere spostati altrove dal punto in cui stanno nel sorgente e devono avere *obbligatoriamente* un'etichetta, un numero progressivo per gli eventuali riferimenti incrociati e una didascalia che ne descriva il contenuto.

Ci occuperemo degli oggetti “mobili” nella Sezione 6, per ora vediamo come scrivere una tabella.

Le regole generali per comporre in modo corretto una tabella sono:

- non usare *mai* filetti verticali;
- evitare filetti doppi;
- se sono presenti misure, scrivere sempre le unità di misura nell'intestazione della colonna e non nel corpo della tabella;
- non usare *mai* le virgolette per ripetere il contenuto di una cella;
- se ci sono numeri decimali, far precedere *sempre* il separatore decimale da almeno una cifra.

La sintassi generale è la seguente

```
\begin{tabular}[allineamento]{colonne}  
... & ... & ... \\  
... & ... & ... \\  
\end{tabular}
```

Il parametro opzionale allineamento può essere `t`, `b` o `c` (di default) e indica come il testo va allineato rispetto a quello circostante, rispettivamente in alto, in basso, centrato.

Il parametro obbligatorio contiene i descrittori delle colonne (uno per colonna) che sono `l` per una colonna allineata a sinistra, `c` per una colonna centrata, `r` per una colonna allineata a destra, `p{larghezza}` per una colonna composta da

un breve capoverso; tra i descrittori di colonna vanno i separatori tra cui citiamo soltanto | per un filetto verticale (da evitare!).

Si può scrivere `*{numero}{descrittore}` per ripetere `numero` di volte un descrittore, anziché ripetere più volte lo stesso separatore.

La composizione della tabella avviene riga per riga, gli elementi delle celle sono separati da `&` e il termine della riga si indica con `\\`. Una linea orizzontale viene inserita con il comando `\hline`: può essere eseguito solo come primo elemento di una tabella, oppure dopo il comando di fine riga `\\`, oppure dopo un altro comando `\hline` (da evitare!).

Il comando `\multicolumn{numero}{posizione}{cella}` serve per comporre un'unica cella con la posizione indicata, che si espande per `numero` colonne.

Il comando `\cline{col1 -col2}` serve per tracciare una linea orizzontale tra le colonne `col1` e `col2`, non si possono raddoppiare le linee, ma si possono avere due linee su due intervalli di colonne diverse.

Vediamo qualche esempio:

---

```
\begin{tabular}{lcr}
\hline
Squadra & Città & Maglia \\
\hline
Inter & Milano & Nerazzurra \\
Juventus & Torino & Bianconera \\
\\
Fiorentina & Firenze & Viola \\
\hline
\end{tabular}
```

---

| Squadra    | Città   | Maglia     |
|------------|---------|------------|
| Inter      | Milano  | Nerazzurra |
| Juventus   | Torino  | Bianconera |
| Fiorentina | Firenze | Viola      |

---

```
\begin{tabular}{l*{4}{c}}
\hline
Lettera & $\alpha$ & $\beta$ & $\gamma$ & $\delta$ \\
Nome & alfa & beta & gamma & delta \\
\hline
\end{tabular}
```

---

| Lettera | $\alpha$ | $\beta$ | $\gamma$ | $\delta$ |
|---------|----------|---------|----------|----------|
| Nome    | alfa     | beta    | gamma    | delta    |

---

```
\begin{tabular}{lp{0.8\textwidth}}
\hline
\textbf{Monarchia}&
```

La monarchia, letteralmente "Governo di uno" è una forma di governo in cui la carica di capo di stato è ricoperta da un re o da una regina oppure un qualsiasi altro sovrano del nobiliare. \\

**Democrazia** &

Democrazia etimologicamente significa ``governo del popolo'', ovvero forma di governo in cui la sovranità è esercitata, direttamente o indirettamente, dal popolo, che generalmente è identificato come l'insieme dei cittadini che ricorrono, in generale, a strumenti di consultazione popolare. \\

\\hline

\\end{tabular}

---

|                   |                                                                                                                                                                                                                                                                                                   |
|-------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Monarchia</b>  | La monarchia, letteralmente “Governo di uno” è una forma di governo in cui la carica di capo di stato è ricoperta da un re o da una regina oppure un qualsiasi altro sovrano del nobiliare.                                                                                                       |
| <b>Democrazia</b> | La democrazia etimologicamente significa “governo del popolo”, ovvero forma di governo in cui la sovranità è esercitata, direttamente o indirettamente, dal popolo, che generalmente è identificato come l’insieme dei cittadini che ricorrono in generale a strumenti di consultazione popolare. |

---



---

```

\begin{tabular}{llc}
\multicolumn{3}{c}{Menù} \\
\hline
\multicolumn{2}{c}{Piatto} \\
& Prezzo \\
\cline{1-2}
Tipo & Descrizione & \\
\hline
Pasta & al pomodoro & 6,50 \\
& al pesto & 8,00 \\
Bistecca & di manzo & 14,00 \\
Crostata & & 6,00 \\
\hline
\end{tabular}

```

---

| Menù     |             |        |
|----------|-------------|--------|
| Piatto   |             | Prezzo |
| Tipo     | Descrizione |        |
| Pasta    | al pomodoro | 6,50   |
|          | al pesto    | 8,00   |
| Bistecca | di manzo    | 14,00  |
| Crostata |             | 6,00   |

Vedremo più avanti nella Sezione 3 che anche altre strutture matematiche che prevedono un allineamento in riga e colonna hanno una sintassi simile.

Tra i pacchetti più usati per ottenere più funzionalità con le tabelle ricordiamo:

- **booktabs**: migliora l’aspetto grafico;

- **array**: estende gli ambienti **array** e **tabular** permettendo di inserire comandi validi per tutte le colonne e definire nuovi tipi di colonna;
- **tabularx**: permette di avere colonne di tipo **p** la cui larghezza è calcolata da  $\text{\LaTeX}$ ;
- **multirow**: per unire più celle su una stessa colonna
- **colortbl**: per tabelle colorate;
- **longtable**: per tabelle molto lunghe che si estendono su più pagine

Tutti gli editor dedicati, ed anche *Overleaf* e *Cocalc*, possiedono un'interfaccia per inserire comodamente le tabelle più comuni.

### 2.2.5 Le figure

Per poter lavorare con le immagini, è necessario caricare il pacchetto **graphicx** che è in grado di interpretare l'immagine e inserirla nel documento.

Una volta caricati i pacchetti, possiamo includere grafici o figure in un documento in due modi:

1. importando un file grafico esterno, in un formato accettato da  $\text{\LaTeX}$  ;
2. “disegnando” direttamente all'interno del documento utilizzando uno dei pacchetti dedicati: **tikz**, **pstricks**, **pgfplots**, **xypic**, **mfpic**, ecc.
3. usando un pacchetto che funzioni da interfaccia verso un programma esterno come **asymptote** o **gnuplot**.

Qui tratteremo solo il primo caso.

Prima di parlare dell'inclusione di figure in  $\text{\LaTeX}$  è utile ricordare che esistono due classi di formati grafici:

- le *immagini vettoriali*: sono descritte da forme e possono essere scalate e deformate senza perdere di definizione; i formati più usati sono **pdf**, **ps** (e il suo parente **eps**), **svg**;
- le *immagini bitmap*: sono matrici di pixel colorati, di solito perdono in definizione se ingrandite o rimpicciolite; i formati più noti sono **jpeg**, **png**, **bmp** e spesso sono compresse.

$\text{\LaTeX}$  è in grado di gestire, usando **pdf $\text{\LaTeX}$**  file in formato **jpg**, **png**, **pdf**, **eps**<sup>7</sup>, il pacchetto **bmpsize** permette inoltre di inserire file bitmap di altri formati come

---

<sup>7</sup>I file in formato **eps** vengono automaticamente convertiti in formato **pdf**.

**bmp** e **gif**.

È quindi necessario, se il file grafico non è supportato da L<sup>A</sup>T<sub>E</sub>X convertirlo in un formato che possa essere interpretato.

**Inclusione di un file esterno** L'immagine viene inclusa in un documento utilizzando il comando:

```
\includegraphics[opzioni]{nomefile}
```

Tra le opzioni del comando `\includegraphics` (tra parentesi quadre), possiamo specificare:

**scale=fattore di scala** consente di scalare proporzionalmente il grafico;

**width=unità** consente di specificare la larghezza del grafico;

**height=unità** consente di specificare l'altezza del grafico;

**keepaspectratio=true/false** evita di distorcere le immagini se vengono specificati sia l'altezza che la larghezza, che diventano ora un limite da non superare;

**angle=angolo di rotazione** consente di specificare l'angolo di rotazione della figura, in gradi ed in senso antiorario.

...

Il file da includere viene cercato nella stessa directory in cui è contenuto anche il file in cui è incluso. Se il file di immagine non è in quella directory (ad esempio, perché si preferisce avere tutte le immagini in una directory apposita, diciamo "immagini") si può indicare il percorso in due modi:

- ✓ direttamente nel comando `\includegraphics`, specificando per intero il percorso del file, ad esempio

```
\includegraphics{immagini/lion3.eps}
```

- ✓ nel preambolo, con il comando `\graphicspath{percorso}`

Vediamo alcuni esempi:

---

```
\includegraphics{lion3}
```

---



---

```
\includegraphics[scale=0.3]{lion3}
```

---



---

```
\includegraphics[width=3cm]{lion3}
```

---



---

```
\includegraphics[height=2cm]{lion3}
```

---



---

```
\includegraphics[height=2cm,width=4cm]{lion3}
```

---



---

```
\includegraphics[angle=30]{lion3}
```

---



Attenzione all'ordine delle opzioni

---

```
\includegraphics[angle=30, width=4cm]{lion3}
```

---



---

```
\includegraphics[width=4cm,angle=30]{lion3}
```

---



*Osservazione 2.* Negli esempi precedenti le lunghezze sono state specificate in centimetri, ma è buona norma dare le lunghezze come dimensioni relative, cioè essere larghe una frazione di `\textwidth` e alte una frazione di `\textheight`, per evitare problemi cambiando classe di documento o numero di colonne.

Utilizzando il pacchetto `subfig` è possibile avere più figure affiancate, il pacchetto `wrapfig` permette di avere figure “immerse” nel testo.

Per generare le immagini da inserire ricordiamo il programma *Geogebra* che permette di generare facilmente e in modo grafico sia figure geometriche che grafici di funzioni di una e due variabili ed esportarli in formati compatibili con  $\text{\LaTeX}$ .

Un altro programma che è in grado di generare immagini vettoriali è *Inkscape*.

### 2.2.6 Le scatole

Una *scatola* è un oggetto che può contenere del testo (anche su più righe) o una figura, ma il suo contenuto, una volta inscatolato, viene trattato dal programma come se fosse un oggetto unico e non lo spezzerà mai alla fine di una riga o di una pagina.

I comandi principali per inscatolare del testo sono

---

```
\mbox{testo}  
\makebox[larghezza][posizione]{testo}  
\fbox{testo}  
\framebox[larghezza][posizione]{testo}
```

---

Questi comandi posizioneranno sempre il testo su una sola riga.

I comandi che cominciano con “f” inseriscono una cornice attorno al testo; quelli che cominciano con la lettera “m” no.

Se la **larghezza** (facoltativa) viene indicata, il testo viene inserito dentro una scatola della larghezza specificata, Se viene specificata la **larghezza** allora si può specificare anche la **posizione**: “l” per allineare il testo a sinistra, “r” per allineare il testo a destra, “s” per distribuire il testo, allargando o restringendo lo spazio tra le parole, altrimenti il testo è centrato.

---

```

\framebox[50mm][l]{testo a sinistra}\\
\framebox[50mm][r]{testo a destra}\\
\framebox[50mm]{testo centrato}\\
\framebox[50mm][s]{testo spaziato}\\

```

---

|                                     |
|-------------------------------------|
| testo a sinistra                    |
| testo a destra                      |
| testo centrato                      |
| testo                      spaziato |

Per inserire del testo dentro scatole che contengono più righe, uno o più capoversi, si può usare il comando `parbox` o l'ambiente `minipage`:

---

```

\parbox[allineamento][altezza][pos.interna]{larghezza}{testo}

```

---

oppure

---

```

\begin{minipage}[allineamento][altezza][pos.interna]{larghezza}
  testo
\end{minipage}

```

---

Tralasciamo i parametri opzionali: il parametro `larghezza` rappresenta la larghezza della scatola.

Il testo in `verbatim` e le box scritte sopra sono in un ambiente `minipage`.

---

```

\fbbox{
  \begin{minipage}{.75\linewidth}
Questo testo è in una \texttt{minipage}\\.
Posso andare a capo.\\
Si possono anche scrivere delle note\footnote{Questa è una nota}.
  \end{minipage}
}

```

---

Questo testo è in una `minipage`.  
 Posso andare a capo.  
 Si possono anche scrivere delle note<sup>a</sup>.

---

<sup>a</sup>Questa è una nota

---

```

\fbbox{
\begin{minipage}{.25\linewidth}
  Questo testo è in una \texttt{minipage} molto più stretta.\\
  Posso andare a capo.\\
  Si possono anche scrivere delle note\footnote{Questa è una nota}.
\end{minipage}
}

```

---



Questo testo è in  
una `minipage` molto  
più stretta.  
Posso andare a  
capo.  
Si possono anche  
scrivere delle note<sup>a</sup>.

---

<sup>a</sup>Questa è una nota

### 2.2.7 Il codice

Come posso fare a far scrivere a  $\text{\LaTeX}$  (come ho fatto sopra) `\textheight` senza ottenere un errore?

E, più in generale, come faccio a inserire del codice di un programma e, in particolare, del codice  $\text{\LaTeX}$ ?

Ci sono molti pacchetti destinati a questo e in particolare citiamo `listings` e `fancyvrb` e la sua estensione `fvextra`. Tutti questi pacchetti sono costruiti su un solo concetto: la possibilità di inserire del testo senza che  $\text{\LaTeX}$  lo interpreti, cioè, come si dice comunemente con una parola latina *verbatim*. I pacchetti non fanno altro che estendere queste capacità e formattare eventualmente il codice in modo da evidenziarne la sintassi.

Noi parleremo solo dell’inserimento del testo *verbatim*.

Ci sono due modalità: una per l’inserimento in linea di poche parole, l’altra per l’inserimento di lunghe parti di testo.

Per l’inserimento in linea si scrive

---

```
\verb simbolo testo da riprodurre simbolo  
oppure  
\verb* simbolo testo da riprodurre simbolo
```

---

dove “simbolo” è un solo carattere e funziona da delimitatore del testo da riprodurre (si usano spesso il “+” o “|”); la versione senza asterisco riproduce gli spazi come spazi; la versione con l’asterisco riproduce gli spazi con il segno speciale  $\sqcup$ .

Quindi se scriviamo `\verb+\textcolor{red}{casa}+` ci apparirà il comando  $\text{\LaTeX}$  e non la parola “casa” in rosso. Normalmente il testo è riprodotto in *typewriter*.

L’ambiente è invece

---

```
\begin{verbatim}  
testo da riprodurre  
di solito  
su più righe
```

---

```
\end{verbatim}
```

oppure

```
\begin{verbatim*}  
...  
\end{verbatim*}
```

---

## 2.2.8 La gestione della pagina

In  $\text{\LaTeX}$  è ovviamente possibile gestire sia la “geometria” della pagina (margini, spazio per la testatina e il piè di pagina, ...) sia lo stile della pagina, cioè cosa far apparire sulla testatine e nel piè di pagina.

Ci sono quattro stili di base

- `empty`
- `plain`
- `headings`
- `myheadings`

Lo stile viene scelto indicando nel preambolo `\pagestyle{stile}`; è possibile modificare lo stile per una pagina con il comando `\thispagestyle{stile}`.

Lo stile `empty` lascia testatina e piè di pagina vuoti.

Lo stile `plain` (di default nelle classi `article` e `report`) lascia vuota la testatina e scrive a piè di pagina il numero di pagina in centro.

Lo stile `headings` (di default nella classe `book`) lascia vuoto il piè di pagina, la testatina della pagina sinistra contiene il numero di pagina a sinistra (esterno della pagina) e il titolo del capitolo corrente a destra (interno) mentre la testatina della pagina destra ha il numero di pagina a destra (esterno) e il titolo del paragrafo a sinistra (interno).

Lo stile `myheadings` è simile a `headings` o e va usato quando non si vuole che le testatine dipendano dai titoli del capitolo o del paragrafo correnti; si dovrà specificarne il contenuto a ogni nuovo capitolo (o paragrafo, se la classe è `article`), con il comando `\markboth` per comporre entrambe e `\markright` per quella di destra.

La gestione degli stili di pagina per adeguarli alle necessità viene molto facilitata dal pacchetto `fancyhdr`.

La geometria della pagina si gestisce invece in modo semplice con il pacchetto `geometry` che ha moltissime funzionalità: anche in questo caso il suggerimento è quello di non intervenire se non strettamente necessario. Prendete un libro che vi appare “ben stampato”, misurate i margini e valutate che proporzione sono

della pagina: vedrete che sono molto più ampi di quello che un utente solitamente imposta.

### 2.2.9 Le macro di $\text{\LaTeX}$

$\text{\TeX}$  è più un linguaggio di programmazione che un programma di scrittura: è quindi naturale “creare” all’interno del documento delle macro, delle scorciatoie, degli ambienti “utili” per il proprio lavoro.

Ad esempio se vogliamo mandare il nostro file `tex` a un nostro amico negli Stati Uniti che ha una tastiera senza lettere accentate, dobbiamo usare i comandi per scrivere “à”, “è”, ecc.

Se scrivere “é” non è difficile `\'e`, scrivere “è” è più lungo perché il simbolo ``` non è immediatamente disponibile sulla tastiera.

Allora può convenire creare un comando così che possiamo inserire più semplicemente quelle lettere accentate.

Scriveremo allora *nel preamble*

---

```
\newcommand{\aacc}{à}
\newcommand{\eacc}{è}
\newcommand{\iacc}{ì}
\newcommand{\oacc}{ò}
\newcommand{\uacc}{ù}
```

---

ed ora scrivere “à” è immediato...

Lasciamo al lettore la scoperta sia del motivo per cui non abbiamo usato i comandi `\a` e lo “strano” comando per definire la “ì”

Un altro esempio: si vedrà che per scrivere  $\mathbb{R}$ ,  $\mathbb{Z}$ , ecc. si usa il comando `\mathbb R`, ecc. È normale avere delle “scorciatoie” per questi comandi: una delle più usate è

```
\newcommand{\set}[1]{\mathbb {#1}}
```

Cosa vuol dire? La sintassi è:

```
\newcommand{nome comando}[numero argomenti]{definizione}
```

dove **nome comando** è il nome del nuovo comando; **numero argomenti** è un numero, il numero di argomenti della funzione; **definizione** è il testo (o sequenza di comandi) che deve essere sostituito tutte le volte che si scrive `\nuovo comando`; un parametro della forma **#n** sarà sostituito dall’ennesimo argomento della funzione.

Allora, il comando di prima: `\newcommand{\set}[1]{\mathbb{#1}}` vuol dire “ogni volta che scrivo `\set R` traducilo come `\mathbb R`” In questo caso, il comando ha un solo argomento (R nell’esempio).

Vediamo adesso un esempio di comando con due argomenti:

`\newcommand{\serie}[2]{\sum_{\#1=\#2}^{+\infty}}`.

Vediamone un'applicazione:

`\[\serie{n}{0}{q^n}`

produce:

$$\sum_{n=0}^{+\infty} q^n$$

Con questo comando, ogni volta che scriviamo `\serie{argomento1}{argomento2}`,  $\text{\LaTeX}$  scriverà la sommatoria, nella parte inferiore inserirà il primo argomento (che sarà l'indice della sommatoria); il segno di uguale ed il secondo argomento (che sarà il valore iniziale dell'indice). Nella parte superiore verrà sempre inserito il simbolo di infinito.

Definire nuovi comandi è non solo utile come scorciatoia, ma anche perché permette di poter modificare rapidamente l'apparenza di un documento. Supponiamo di dover evidenziare alcune parole del documento e decidiamo di farlo con `\emph{parola}`. Se in un secondo momento decidiamo di evidenziare le stesse parole in grassetto o usando un colore diverso dobbiamo cercare in tutto il documento le parole evidenziate e cambiare stile. Se invece definiamo un comando

`\newcommand{\ev}[1]{\emph{\#1}}`

e evidenziamo le parole con `\ev{parola}`, quando cambiamo idea basta modificare la macro in

`\newcommand{\ev}[1]{\textbf{\#1}}`.

Con una sintassi simile può essere ridefinita una funzione definita internamente da  $\text{\LaTeX}$  o definita precedentemente da noi. La sintassi sarà

`\renewcommand{nomecomando}[argomenti]{definizione}`

ed è del tutto analoga a quella del comando `newcommand`.

In modo simile a quello visto per la creazione di nuovi comandi si possono creare nuovi ambienti o ridefinire ambienti già esistenti: la sintassi è

`\newenvironment{nome}[argomenti]`

`{definizione inizio} {definizione fine}`

dove `nome` è il nome dell'ambiente, `argomenti` è il numero dei parametri presenti nella definizione, `definizione inizio` è il testo che verrà sostituito ogni volta che scriveremo `\begin{nome}` e `definizione fine` è il testo che verrà sostituito ogni volta che scriveremo `\end{nome}`.

## 3 $\text{\LaTeX}$ e la matematica

### 3.1 Le modalità matematiche

$\text{\LaTeX}$  dispone di due modalità matematiche, che consentono di inserire formule nel documento:

- *l'ambiente  $\text{math}$* : è utilizzato per le formule che devono comparire all'interno del testo;
- *l'ambiente  $\text{displaymath}$* : è utilizzato per le formule che si desidera far stampare su una linea a sè stante.

Molti editor hanno barre degli strumenti che facilitano l'inserimento dei vari simboli.

#### 3.1.1 L'ambiente $\text{math}$

Questo ambiente consente, come si è detto, di inserire formule in linea con il resto del testo. Esistono diversi comandi che consentono di creare l'ambiente in questione:

- $\backslash(\dots\backslash)$
- $\$...\$$

Ad esempio, si può equivalentemente scrivere:

$\backslash(2x+y=3\backslash)$

$\$2x+y=3\$$

ed il risultato sarà sempre:  $2x + y = 3$ .

I puristi di  $\text{\LaTeX}$  consigliano però il primo metodo per una migliore gestione degli spazi.

#### 3.1.2 L'ambiente $\text{displaymath}$

è il modo matematico che permette di inserire formule su una linea a parte. Esistono anche qui diversi comandi per creare l'ambiente:

- $\backslash[\dots\backslash]$
- $\$...\$$

Ad esempio, se si scrive:

`\[2x+y=3\]`

`$$2x+y=3$$`

il risultato sarà sempre:

$$2x + y = 3$$

In questo caso è **fortemente** consigliato usare la prima scrittura, sempre per una migliore gestione degli spazi.

Gli spazi non hanno alcun effetto quando si scrive in modalità matematica, quindi

`$2x+y=3$`

darà lo stesso risultato di

`$\displaystyle 2x+y=3$`

Nell'ambiente `displaymath` occorre solo fare attenzione a non lasciare linee bianche, altrimenti durante la compilazione si ottengono messaggi di errore.

La principale differenza tra l'ambiente `math` e l'ambiente `displaymath`, oltre alla posizione del testo già citata, è relativa alla dimensione dei principali operatori matematici. Infatti, tali operatori sono di dimensione più piccola quando sono inseriti in linea con il testo (`math`), mentre sono più grandi quando sono inseriti su una linea separata e centrati (`displaymath`). In ambiente `math` si può utilizzare il comando `\displaystyle`, che dà ai simboli di integrale e sommatoria, così come ad altri simboli a grandezza variabile la grandezza che assumerebbero in ambiente `displaymath` (in questo modo, però, l'interlinea del testo risulta modificata)<sup>8</sup>.

Inoltre alcuni pedici ed apici sono posizionati in modo diverso

$$\lim_{x \rightarrow 0} f(x) = 0 \qquad \int_0^1 f(x) dx = 2$$

$$\lim_{x \rightarrow 0} f(x) = 0 \qquad \int_0^1 f(x) dx = 2$$

## 3.2 I principali operatori matematici

Ora descriveremo i principali operatori matematici, illustrando i comandi `LATEX` che consentono di ottenerli.

---

<sup>8</sup>Esiste un analogo comando, `\textstyle`, che consente di scrivere gli operatori in display come se fossero in linea con il testo.

| Operatore           | Comando                                      | Esempio                           | Risultato          |
|---------------------|----------------------------------------------|-----------------------------------|--------------------|
| Apici <sup>a</sup>  | <code>^{\}</code>                            | <code>&amp;\$x^{2}\$</code>       | $x^2$              |
| Pedici <sup>b</sup> | <code>_{\}</code>                            | <code>\$x_{1}\$</code>            | $x_1$              |
| Frazioni            | <code>\frac{num}{den}</code>                 | <code>\$\frac{1}{2}\$</code>      | $\frac{1}{2}$      |
| Radici              | <code>\sqrt[indice]{argomento}</code>        | <code>\$\sqrt[5]{x}\$</code>      | $\sqrt[5]{x}$      |
| Sommatorie          | <code>\sum_{ind. inf.}^{ind. sup.}arg</code> | <code>\$\sum_{n=0}^5 q^n\$</code> | $\sum_{n=0}^5 q^n$ |
| Integrali           | <code>\int_{inf.}^{sup.}f(x)dx</code>        | <code>\$\int_0^3 3x^4 dx\$</code> | $\int_0^3 3x^4 dx$ |

Tabella 8: Gli operatori matematici in  $\text{\LaTeX}$ .

<sup>a</sup>In modalità testo il comando è `\ap`

<sup>b</sup>In modalità testo il comando è `\ped`

### 3.3 Le funzioni

In  $\text{\LaTeX}$  esistono dei comandi appositi che consentono di scrivere le funzioni elementari nel modo corretto (carattere e spaziatura). Tali comandi sono elencati nella Tabella 9.

|                      |                    |                   |                   |                      |                      |                   |                    |
|----------------------|--------------------|-------------------|-------------------|----------------------|----------------------|-------------------|--------------------|
| <code>\arccos</code> | <code>\cos</code>  | <code>\csc</code> | <code>\exp</code> | <code>\ker</code>    | <code>\limsup</code> | <code>\min</code> | <code>\sinh</code> |
| <code>\arcsin</code> | <code>\cosh</code> | <code>\deg</code> | <code>\gcd</code> | <code>\lg</code>     | <code>\ln</code>     | <code>\Pr</code>  | <code>\sup</code>  |
| <code>\arctan</code> | <code>\cot</code>  | <code>\det</code> | <code>\hom</code> | <code>\lim</code>    | <code>\log</code>    | <code>\sec</code> | <code>\tan</code>  |
| <code>\arg</code>    | <code>\coth</code> | <code>\dim</code> | <code>\inf</code> | <code>\liminf</code> | <code>\max</code>    | <code>\sin</code> | <code>\tanh</code> |

Tabella 9: Funzioni

Per avere le funzioni “in italiano” è necessario definirle: il modo migliore è usare un comando del pacchetto `amsmath`

|                                                              |
|--------------------------------------------------------------|
| <code>\DeclareMathOperator{comando}{testo che appare}</code> |
| ad esempio                                                   |
| <code>\DeclareMathOperator{\sen}{sen}</code>                 |

### 3.4 Lettere greche

I comandi che permettono di ottenere le lettere greche sono elencati nella Tabella 10.

| Minuscole     |                          |             |                        |             |                        |           |                      |
|---------------|--------------------------|-------------|------------------------|-------------|------------------------|-----------|----------------------|
| $\alpha$      | <code>\alpha</code>      | $\theta$    | <code>\theta</code>    | $\pi$       | <code>\pi</code>       | $\phi$    | <code>\phi</code>    |
| $\beta$       | <code>\beta</code>       | $\vartheta$ | <code>\vartheta</code> | $\varpi$    | <code>\varpi</code>    | $\varphi$ | <code>\varphi</code> |
| $\gamma$      | <code>\gamma</code>      | $\iota$     | <code>\iota</code>     | $\rho$      | <code>\rho</code>      | $\chi$    | <code>\chi</code>    |
| $\delta$      | <code>\delta</code>      | $\kappa$    | <code>\kappa</code>    | $\varrho$   | <code>\varrho</code>   | $\psi$    | <code>\psi</code>    |
| $\epsilon$    | <code>\epsilon</code>    | $\lambda$   | <code>\lambda</code>   | $\sigma$    | <code>\sigma</code>    | $\omega$  | <code>\omega</code>  |
| $\varepsilon$ | <code>\varepsilon</code> | $\mu$       | <code>\mu</code>       | $\varsigma$ | <code>\varsigma</code> |           |                      |
| $\zeta$       | <code>\zeta</code>       | $\nu$       | <code>\nu</code>       | $\tau$      | <code>\tau</code>      |           |                      |
| $\eta$        | <code>\eta</code>        | $\xi$       | <code>\xi</code>       | $\upsilon$  | <code>\upsilon</code>  |           |                      |
| Maiuscole     |                          |             |                        |             |                        |           |                      |
| $\Gamma$      | <code>\Gamma</code>      | $\Lambda$   | <code>\Lambda</code>   | $\Sigma$    | <code>\Sigma</code>    | $\Psi$    | <code>\Psi</code>    |
| $\Delta$      | <code>\Delta</code>      | $\Xi$       | <code>\Xi</code>       | $\Upsilon$  | <code>\Upsilon</code>  | $\Omega$  | <code>\Omega</code>  |
| $\Theta$      | <code>\Theta</code>      | $\Pi$       | <code>\Pi</code>       | $\Phi$      | <code>\Phi</code>      |           |                      |

Tabella 10: Lettere greche

### 3.5 Operatori, Relazioni, Frecce, Delimitatori, Puntini ed altri simboli

Nella Tabella 11 vediamo alcuni esempi di altri simboli matematici che possono essere scritti in  $\text{\LaTeX}$ .

### 3.6 Parentesi

Oltre alle normali parentesi  $( )$ ,  $[ ]$ ,  $\{ \}$  che hanno dimensione fissata, è possibile usare altre parentesi “espandibili” o, usando il pacchetto `amsmath`, parentesi più grandi, ma sempre di grandezza fissata.

La sintassi per le parentesi “espandibili” è `\left( \right)` e analogamente per le altre. Le parentesi graffe possono essere indicate anche con la coppia `\left\lbrack \right\rbrack`. Rientrano in questa tipologia anche i seguenti simboli

`\left\langle \right\rangle` `\left| \right|` `\left\| \right\|` `\left\lfloor \right\rfloor` `\left\lceil \right\rceil`

che producono

$\langle \rangle$   $\|$   $\|$   $\lfloor \rfloor$   $\lceil \rceil$





### 3.8 Modifiche della spaziatura

In alcuni casi può essere utile, anche nel testo matematico, modificare la spaziatura per allontanare o avvicinare alcuni simboli.

|                 |                |                     |                                |
|-----------------|----------------|---------------------|--------------------------------|
| <code>\,</code> | spazio piccolo | <code>\!</code>     | spazio negativo piccolo        |
| <code>\:</code> | spazio medio   | <code>\quad</code>  | spazio piccolo tra due formule |
| <code>\;</code> | spazio grande  | <code>\qquad</code> | spazio grande tra due formule  |

Tabella 12: Comandi di spaziatura

I comandi `quad` e `qquad` sono parte del pacchetto `amsmath`.

### 3.9 Modificare lo stile ed il corpo del testo

I comandi: `\mathit`; `\mathrm`; `\mathbf` e `\mathsf` permettono di cambiare lo stile dei caratteri in modalità matematica. Occorre però ricordare che *non tutti i caratteri matematici sono disponibili con tutti gli stili*.

Il comando `\mathcal` genera delle lettere maiuscole corsive. Ad esempio: `$_{\mathcal{C}}$` fa stampare:  $\mathcal{C}$ .

I simboli per gli insiemi numerici  $\mathbb{R}$ ,  $\mathbb{Z}$ ,  $\mathbb{N}$ , ecc. si ottengono con il comando `$_{\mathbb{R}}$`, ecc. (`mathbb`= blackboard bold).

Ci sono quattro stili di composizione della matematica

---

|                                 |
|---------------------------------|
| <code>\displaystyle</code>      |
| <code>\textstyle</code>         |
| <code>\scriptstyle</code>       |
| <code>\scriptscriptstyle</code> |

---

e questi stili permettono (se necessario, ma è meglio evitarlo!) di modificare la grandezza dei caratteri.

In particolare, a grandi linee, lo stile `\textstyle` differisce dal `\displaystyle` perché è più raccolto in verticale, mentre `\scriptstyle` è un carattere più piccolo.

### 3.10 Gli ambienti della Matematica

L<sup>A</sup>T<sub>E</sub>X ha diversi ambienti specifici per la matematica. Tutti possono essere inseriti nei documenti attraverso gli appositi comandi.

Ora esamineremo i più usati di questi ambienti:

### 3.10.1 L'ambiente `array`

Le matrici (o tutto ciò che può essere scritto come una tabella) si ottengono con l'ambiente `array` (il cui funzionamento è analogo a quello dell'ambiente `tabular` per il testo). Ad esempio, una matrice potrà essere scritta in questo modo:

```
\[{\mathcal A} = \left(
\begin{array}{ccc}
a_{11} & a_{12} & a_{13} \\
a_{21} & a_{22} & a_{23} \\
a_{31} & a_{32} & a_{33}
\end{array}
\right)\]
```

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

Il parametro racchiuso tra parentesi graffe dell'ambiente `array` indica il numero di colonne della tabella ed il loro allineamento, `&` è il separatore di colonna e `\\` quello di riga, esattamente come nell'ambiente `tabular`.

### 3.10.2 Matrici

Il pacchetto `amsmath` mette a disposizione altri ambienti per la costruzione di matrici oltre all'ambiente `array`

- `\begin{pmatrix}...\end{pmatrix}` → (matrice)
- `\begin{bmatrix}...\end{bmatrix}` → [matrice]
- `\begin{Bmatrix}...\end{Bmatrix}` → {matrice}
- `\begin{vmatrix}...\end{vmatrix}` → |matrice|
- `\begin{Vmatrix}...\end{Vmatrix}` → ||matrice||
- `\begin{matrix}...\end{matrix}` → matrice

La spaziatura è inferiore a quella di `array`. Si possono avere fino a 10 colonne centrate: se le colonne devono essere allineate a destra o a sinistra, bisogna usare `array`.

### 3.10.3 Casi

Il pacchetto `amsmath` fornisce anche un ambiente per le definizioni fatte per *casi* che funziona benissimo anche per i sistemi di equazioni: l'ambiente `cases`. La sintassi dell'ambiente `cases` è analoga a quella dell'ambiente `array`, ma senza allineamento e con 2 sole colonne: le colonne sono separate da `&` e ogni riga termina con `\\`. La

parentesi e l'allineamento sono automatici. Ad esempio:

```
\[ f(x)=\begin{cases}
x^2 & \text{ per } x < 1 \\
3 & \text{ per } 1 \leq x < 3 \\
\ln x & \text{ per } x \geq 3
\end{cases} \]
```

$$f(x) = \begin{cases} x^2 & \text{ per } x < 1 \\ 3 & \text{ per } 1 \leq x < 3 \\ \ln x & \text{ per } x \geq 3 \end{cases}$$

### 3.10.4 Gli altri ambienti

Il pacchetto `amsmath` fornisce nuove strutture per le equazioni in display, oltre a quelle fornite da  $\text{\LaTeX}$ .

- `equation equation*`
- `gather gather*`
- `split`
- `multline multline*`
- `align align*`
- `flalign flalign*`

**Le equazioni numerate e non: gli ambienti `equation` e `equation*`** L'ambiente `equation` produce formule con lo stesso stile dell'ambiente `displaymath`, ma in più numera l'equazione. La versione senza asterisco permette la numerazione automatica. Ad esempio:

```
\begin{equation}
x^2+y^2=4
\end{equation}
```

$$x^2 + y^2 = 4 \quad (3.1)$$

L'aggiunta di un asterisco (\*) dopo `equation` fa sì che le equazioni non vengano numerate.

**L'ambiente `gather`** L'ambiente `gather` serve per scrivere più equazioni, ognuna centrata in una riga a sé, numerata ed etichettabile separatamente. Le equazioni non sono allineate l'una con l'altra. è possibile non numerare le equazioni:

- *singolarmente*: una singola equazione che finisce con `\notag` non è numerata;
- *globalmente*: `gather*` (come `equation*`) non numera *tutte* le equazioni contenute nell'ambiente.

Ad esempio:

```
\begin{gather}
```

```
a_1=b_1+c_1\\
```

```
a_2=b_2+c_2-d_2+e_2
```

```
\end{gather}
```

$$a_1 = b_1 + c_1 \quad (3.2)$$

$$a_2 = b_2 + c_2 - d_2 + e_2 \quad (3.3)$$

```
\begin{gather}
```

```
f(x)=\ln x \notag\\
```

```
g(x,y)=\sqrt{x^2+y^2}
```

```
\end{gather}
```

$$f(x) = \ln x$$

$$g(x,y) = \sqrt{x^2 + y^2} \quad (3.4)$$

**L'ambiente `split`** L'ambiente `split` serve per dividere su più righe una singola equazione. Le righe vengono allineate tra loro, con riferimento ad un punto preciso indicato da chi scrive. Infatti, il comando `&` marca il punto di una riga da incolonnare coi punti corrispondenti delle altre righe. L'ambiente `split` non può essere utilizzato da solo, ma unicamente all'interno degli ambienti `equation` o `gather` (con o senza asterisco). Il numero attribuito a ciascuna equazione è centrato verticalmente. Ad esempio:

```
\begin{equation}
```

```
\begin{split}
```

```
a&=b+c-d\\
```

```
& \quad +e-f\\
```

```
&=g+h\\
```

```
&=i
```

```
\end{split}
```

```
\end{equation}
```

$$\begin{aligned} a &= b + c - d \\ &\quad + e - f \\ &= g + h \\ &= i \end{aligned} \quad (5)$$

**L'ambiente `multline`** L'ambiente `multline` serve per una singola equazione da dividere su più righe, senza necessità di un particolare allineamento fra le varie righe. Ogni equazione è numerata singolarmente. In stampa la prima riga comincia a sinistra, l'ultima finisce a destra, e le altre vengono centrate. Il numero di equazione viene stampato al margine destro dell'ultima riga. Ad esempio:

```
\begin{multline}
```

```
a+b+c\\
```

```
+d+e+f\\
```

```
+i+j+k\\
```

```
+l+m+n
```

```
\end{multline}
```

$$\begin{aligned} a + b + c \\ + d + e + f \\ + i + j + k \\ + l + m + n \end{aligned} \quad (6)$$

**L'ambiente `align`** L'ambiente `align` permette di scrivere più equazioni, ciascuna su una riga, numerate singolarmente, e da allineare fra loro. Come per altri

ambienti, il comando `&` indica la posizione di una riga da incolonnare con le altre. Di solito si allinea prima dell'uguale: `&=`. Ad esempio:

```
\begin{align}
a_1&=b_1+c_1\\
a_2&=b_2+c_2-d_2+e_2
\end{align}
```

$$a_1 = b_1 + c_1 \quad (7)$$

$$a_2 = b_2 + c_2 - d_2 + e_2 \quad (8)$$

### 3.11 Teoremi, definizioni...

Nelle pubblicazioni di carattere matematico, teoremi e dimostrazioni sono tra gli elementi più comuni, ma si usano anche molti altri ambienti simili, come lemmi, proposizioni, assiomi, corollari, congetture, definizioni, ecc. Questi elementi sono trattati in  $\text{\LaTeX}$  come ambienti. Le diverse classi di documento non forniscono ambienti predefiniti per questi elementi perché la loro varietà è così ampia da rendere impossibile per una classe di documento fornire ogni elemento che sarebbe richiesto.

Esiste invece il comando `newtheorem` che facilita l'impostazione degli elementi richiesti per un particolare documento.

Il comando `newtheorem` è un comando di base di  $\text{\LaTeX}$ . Esistono alcuni pacchetti (ad esempio `amsmath` e `ntheorem`) che lo ridefiniscono per consentirne un uso più ampio. Qui faremo riferimento al comando `newtheorem` così come definito nel pacchetto `amsthm`.

La sintassi di base è

```
\newtheorem{nome}{dicitura}
```

Come si può vedere, tale comando ha due argomenti obbligatori: il primo è il nome dell'ambiente che l'autore desidera usare per questo elemento; il secondo è la dicitura che compare.

L'ambiente così definito viene usato nel documento con la solita sintassi

```
\begin{nome}
  enunciato
\end{nome}
```

Per esempio:

```
\newtheorem{teor}{Teorema}
\begin{teor}
  I numeri primi sono infiniti.
\end{teor}
```

produce:

**Teorema 1.** *I numeri primi sono infiniti.*

Se si usa `newtheorem*` al posto di `newtheorem` nell'esempio sopra, non sarà generata la numerazione automatica per quel tipo di ambiente.

Se ad esempio si avesse un solo lemma il cui nome dovrebbe essere *Lemma di Euclide* invece di *Lemma+* il numero, allora la dichiarazione

```
\newtheorem*{EU}{Lemma di Euclide}
```

permetterebbe di scrivere

```
\begin{EU} I numeri primi sono infiniti. \end{EU}
```

e di ottenere

**Lemma di Euclide.** *I numeri primi sono infiniti.*

Notiamo che è possibile “dare un nome” al teorema anche mantenendo la numerazione automatica, utilizzando la seguente sintassi:

```
\begin{teor}[nome teorema]
  testo
\end{teor}
```

Ad esempio

```
\begin{teor}[di Euclide]
  I numeri primi sono infiniti.
\end{teor}
```

produce:

**Teorema 1** (di Euclide). *I numeri primi sono infiniti.*

Osserviamo che il comando `newtheorem` viene di solito inserito nel preambolo, ma può essere usato anche nel corpo del documento, purché, ovviamente prima di utilizzare l'ambiente definito dal comando stesso.

Come si è detto, è possibile creare numerosi ambienti diversi, uno per i teoremi, uno per le definizioni, uno per i lemmi, uno per le ipotesi, ecc. Di default ognuno

di essi ha un contatore distinto. Tutti questi ambienti funzionano esattamente come *teorema*.

Se si vuole che due ambienti condividano la stessa sequenza di numerazione occorre usare la sintassi `\newtheorem{nome1}[nome2]{dicitura}` che numera gli ambienti “nome1” e “nome2” con lo stesso contatore. Ad esempio, i comandi

```
\newtheorem{thm}{Theorem}
\newtheorem{prop}[thm]{Proposition}
```

faranno sì che Theorem e Proposition condividano la stessa sequenza di numerazione.

Inoltre, è possibile subordinare il contatore ad un’unità di sezionamento. Con il comando

```
\newtheorem{thm}{Teorema}[chapter]
```

il contatore di `thm` sarà della forma Teorema 7.1, con 7=numero del capitolo e 1=numero del teorema all’interno del capitolo.

Ad esempio:

```
\newtheorem{thm}{Teorema}[chapter]
\newtheorem{ex}{Esempio}
...
\chapter{Capitolo 2}
\begin{thm}
Questo è un teorema
\end{thm}
\begin{ex}
Questo è un esempio
\end{ex}
\chapter{Capitolo 3}
\begin{thm}
Questo è un altro profondo
teorema
\end{thm}
\begin{ex}
Questo è un esempio
chiarificatore
\end{ex}
```

## Capitolo 2

**Teorema 2.1.** *Questo è un teorema*

**Esempio 1.** *Questo è un esempio*

## Capitolo 3

**Teorema 3.1.** *Questo è un altro profondo teorema*

**Esempio 2.** *Questo è un esempio chiarificatore*



### 3.11.1 Stili dei teoremi

Usando il pacchetto `amsthm`, si possono usare tre stili predefiniti di presentazione dei teoremi (è possibile definirne altri ancora ed alcuni pacchetti permettono una maggiore libertà):

- ✓ `plain`
- ✓ `definition`
- ✓ `remark`

che forniscono una differente enfasi visiva a seconda del livello di importanza. In generale lo stile `plain` produce testo in corsivo, mentre gli altri due forniscono testo in tondo. Per dare a un enunciato un certo stile basta premettere alla definizione dell'enunciato il comando `theoremstyle`; se non si dà alcun comando `theoremstyle`, lo stile usato sarà `plain`.

Ad esempio:

```
\theoremstyle{plain}% default
\newtheorem{thm}{Teorema}
\newtheorem*{cor}{Corollario}

\theoremstyle{definition}
\newtheorem{defn}{Definizione}
\newtheorem{conj}{Congettura}

\theoremstyle{remark}
\newtheorem{ex}{Esempio}
\begin{defn}
Un numero si dice primo se possiede solo
2 divisori distinti.
\end{defn}
\begin{thm}[Euclide]
Non esiste il massimo dei numeri primi.
\end{thm}
\begin{cor}
Esistono infiniti numeri primi.
\end{cor}
\begin{conj}[Goldbach]
Ogni numero pari è somma di 2 numeri primi
\end{conj}
\begin{ex}
\[[32=19+13\]
\end{ex}
```

**Definizione 3.** Un numero si dice primo se possiede solo 2 divisori distinti.

**Teorema 4** (Euclide). *Non esiste il massimo dei numeri primi.*

**Corollario.** *Esistono infiniti numeri primi.*

*Congettura 1* (Goldbach). Ogni numero pari è somma di 2 numeri primi

*Esempio 2.*

$$32 = 19 + 13$$

### 3.11.2 Dimostrazioni

Il pacchetto `amsmath` ha anche l'ambiente `proof` per le dimostrazioni:

---

```
\begin{thm}[Sorpresa] Si ha che
  \[0=1\]
\end{thm}
\begin{proof}
Da
\[e^{2n\pi i}=1\]
segue che
\[e^{1+2n\pi i}=e\]
\[ (e^{1+2n\pi i})^{1+2n\pi i}=e^{(1+2n\pi i)^2}=e\]
per cui
\[e^{1+4n\pi i-4n^2\pi^2}=e\]
e infine
\[e^{4n\pi i-4n^2\pi^2}=1\]
Mandando  $n\rightarrow +\infty$  si ha la tesi.
\end{proof}
```

---

**Teorema 5** (Sorpresa). *Si ha che*

$$0 = 1$$

*Dimostrazione.* Da

$$e^{2n\pi i} = 1$$

segue che

$$e^{1+2n\pi i} = e$$

$$(e^{1+2n\pi i})^{1+2n\pi i} = e^{(1+2n\pi i)^2} = e$$

per cui

$$e^{1+4n\pi i-4n^2\pi^2} = e$$

e infine

$$e^{4n\pi i-4n^2\pi^2} = 1$$

Mandando  $n \rightarrow +\infty$  si ha la tesi. ■

L'ambiente `proof` produce l'intestazione *Proof* (o *Dimostrazione* se si usa `babel` con opzione `italian`) con interpunzione e spaziatura appropriate. C'è un parametro opzionale per il nome della dimostrazione:

```
\begin{proof}[Dimostrazione del teorema 1]
...
\end{proof}
```

Un simbolo  $\square$ , viene automaticamente aggiunto alla fine dell'ambiente `proof`; per cambiarlo si deve utilizzare

```
\renewcommand{\qedsymbol}{nuovo simbolo}
```

Ad esempio, per usare il simbolo ■, occorre usare la sintassi

```
\renewcommand{\qedsymbol}{\blacksquare}
```

Nel nostro articolo, ad esempio, abbiamo introdotto la *definizione* in questo modo:

```

\begin{defn}
La quantità  $x_1 - x_0 = \Delta x$  si dice
\emph{incremento della variabile}  $x$ .

La quantità  $y_1 - y_0 = \Delta y$  o, meglio,
 $f(x_0 + \Delta x) - f(x_0) = \Delta f$  si dice
\emph{incremento della funzione}  $f$  relativo
all'incremento  $\Delta x$ .

La quantità

$$\frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} = \frac{\Delta f}{\Delta x}$$

si dice \emph{rapporto incrementale della funzione}
 $f(x)$  relativo all'intervallo  $[x_0, x_0 + \Delta x]$ .
\end{defn}

```

Questo ambiente era stato introdotto nel preambolo in questo modo:

```

\theoremstyle{definition}
\newtheorem{defn}[thm]{Definizione}

```

Il risultato è visibile nella Figura 3

**Definizione 1.1.** La quantità  $x_1 - x_0 = \Delta x$  si dice *incremento della variabile  $x$* .  
 La quantità  $y_1 - y_0 = \Delta y$  o, meglio,  $f(x_0 + \Delta x) - f(x_0) = \Delta f$  si dice *incremento della funzione  $f$*  relativo all'incremento  $\Delta x$ .  
 La quantità

$$\frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} = \frac{\Delta f}{\Delta x}$$

si dice *rapporto incrementale della funzione  $f(x)$*  relativo all'intervallo  $[x_0, x_0 + \Delta x]$ .

Figura 3: Una definizione dell'articolo di esempio

Analogamente, abbiamo usato l'ambiente `thm` per i teoremi, in questo modo:

```

\begin{thm}
[Teorema di Fermat]\label{thm:Teorema Fermat}
Sia  $f:(a,b) \rightarrow \mathbb{R}$  derivabile in  $x_0 \in (a,b)$ .
Se  $x_0$  è un punto di estremo locale allora  $f'(x_0) = 0$ .
\end{thm}

```

Anche questo ambiente era stato definito nel preambolo:

```

\newtheorem{thm}{Teorema}[section]

```

Il risultato è visibile nella Figura 4

Sono stati usati, inoltre, gli ambienti `cor` per i corollari, `lem` per i lemmi e `prop` per le proposizioni.

**Teorema 2.1** (Teorema di Fermat). *Sia  $f : (a, b) \rightarrow \mathbb{R}$  derivabile in  $x_0 \in (a, b)$ . Se  $x_0$  è un punto di estremo locale allora  $f'(x_0) = 0$ .*

Figura 4: Un teorema dell'articolo di esempio

## 4 L<sup>A</sup>T<sub>E</sub>X e le altre scienze

Abbiamo dedicato un'intera sezione alla matematica perché L<sup>A</sup>T<sub>E</sub>X è nato per scrivere matematica, ciò non toglie che venga utilizzato per scrivere in altre scienze e in particolare in fisica.

Per tutte le scienze esistono pacchetti che forniscono modelli specifici per scrivere sulle riviste di settore.

Di sicuro è necessario citare il pacchetto `siunitx` che, tra tante altre cose, permette di scrivere in modo corretto, sia come forma che come spaziatura, le unità di misura, lasciando all'utente ampio spazio per usare il codice preferito.

```
\unit{kg.m.s^{-1}} \\
\unit{\kilogram\metre\per\second} \\
\unit[per-mode = symbol]
{\kilogram\metre\per\second} \\
```

kg m s<sup>-1</sup>  
 kg m s<sup>-1</sup>  
 kg m/s

Ci sono pacchetti specifici per la fisica delle alte energie, per l'ottica (in particolare per grafici), per i diagrammi di Feynman, per i diagrammi elettrici, per la termodinamica.

Per quanto riguarda la chimica ci sono pacchetti che permettono tra l'altro di scrivere e disegnare facilmente le formule chimiche, gli schemi di reazioni, ecc.

Per la biologia ci sono pacchetti per gli alberi genealogici.

## 5 Il sezionamento

### 5.1 Le parti di un documento

Come abbiamo detto un documento scritto con la classe `article` può essere suddiviso in sezioni, sottosezioni, sottosottosezioni, paragrafi e sottoparagrafi, introdotti rispettivamente con i comandi:

```
\section
\subsection
\subsubsection
\paragraph
```

`\subparagraph`

mentre nelle classi `report` e `book` è possibile la suddivisione in parti e capitoli (che sono di livello superiore rispetto a `section`) con i comandi:

`\part`

`\chapter`

La sintassi è, per tutti, del tipo `\section[opzioni]{titolo}`, dove “opzioni” rappresenta il titolo che appare nella testatina della pagina e che viene stampato nell’indice.

È possibile anche usare la sintassi `\section*{titolo}` che produrrà il titolo, senza numerare la sezione né introdurla nell’indice.

L’aspetto (carattere usato, grandezza e stile del carattere, posizione) del titolo di un capitolo, una sezione, ecc. come l’aspetto del titolo che appare sulla testatina della pagina, può essere modificato sia con comandi  $\text{\LaTeX}$ , sia, più comodamente, usando il pacchetto `titlesec` che permette anche di modificare l’aspetto dell’indice.

### 5.1.1 Altri sezionamenti

Nella classe `book` sono presenti dei “supersezionamenti” (le parti in italico sono necessarie)

- Materiale iniziale (`\frontmatter`): *Frontespizio*, Colophon, Dedicata, Sommario, *Indice generale*, Elenco delle figure, Elenco delle tabelle, Altri elenchi, Prefazione, Ringraziamenti, Introduzione non numerata. Non sono numerate, la numerazione delle pagine è con numeri romani minuscoli
- Materiale principale (`\mainmatter`): Introduzione numerata, *Capitoli*, una o più appendici numerate. Numera le sezioni e le pagine con numeri arabi (la numerazione della pagine riprende da 1)
- Materiale finale (`\backmatter`): una o più appendici non numerate, Glossario, *Bibliografia*, Indice analitico. Non sono numerate, la numerazione delle pagine prosegue quella principale

**Title** La sezione “Titolo” di un documento è formata in genere dal nome del documento, dai nomi degli autori e talvolta anche dalla data. Per produrre il titolo, si utilizzano i comandi:

`\title{nome del documento}`

`\author{nomi degli autori}`

`\date{data}`

inseriti nel preambolo<sup>9</sup> e:

```
\maketitle
```

inserito dopo il comando `\begin{document}`.

Per L<sup>A</sup>T<sub>E</sub>X il campo `title` è obbligatorio (se è presente il comando `\maketitle`), il campo `author` è opzionale, se non compare il campo `date` viene inserita la data attuale. Se non si vuole che compaia uno dei campi del titolo si lasciano vuote le parentesi graffe.

Il comando `\maketitle` è indispensabile per stampare titolo, autore e data. Tutti i campi vengono stampati centrati sulla linea nella quale appaiono. Se il testo è troppo lungo per essere collocato in un'unica linea, viene automaticamente separato su più linee. È possibile decidere in quale punto il testo si deve interrompere usando il comando `\\`. I nomi degli autori compaiono su un'unica linea se sono separati dal comando `\and`; compaiono invece uno sotto l'altro se sono separati da `\\`.

Il comando `\\` viene usato anche per inserire indirizzo, e-mail, ecc. dell'autore. Inoltre, con il comando `\thanks{...}` può essere creata una nota a piè di pagina, nella quale possono essere inseriti ringraziamenti, classificazione dell'articolo, parole chiave, ecc.

Il “titolo” viene stampato su una pagina separata se si usano le classi `book` e `report`; nella prima pagina del documento per la classe `article` (questo comportamento standard può essere modificato dalle opzioni `titlepage` o `notitlepage`).

Nell'articolo che intendiamo scrivere come esempio, i comandi che permettono di ottenere il titolo sono:

---

```
\title{Alcuni importanti teoremi del Calcolo differenziale}  
\author{Prof. Pico de' Paperis\\Università di Paperopoli}  
\date{25 Dicembre 2006}  
  
\begin{document}  
  
\maketitle
```

---

Il risultato è visibile nella figura 5.

**Abstract** Per le classi `article` e `report`, è possibile scrivere un abstract, in formato speciale, mediante il comando:

```
\begin{abstract}  
Testo dell'abstract  
\end{abstract}
```

---

<sup>9</sup>I comandi che producono il titolo possono essere anche inseriti dopo `\begin{document}`, ma in genere si preferisce includerli nel preambolo perché è il luogo ove si fanno tutte le dichiarazioni propedeutiche alla scrittura del documento vero e proprio.

# Alcuni importanti teoremi del Calcolo differenziale

Prof. Pico de' Paperis  
Università di Paperopoli

25 Dicembre 2023

Figura 5: Il titolo dell'articolo di esempio

È possibile scrivere un abstract anche con la classe **report**: in tal caso apparirà sulla pagina del titolo (separata dal resto del documento) mentre nella classe **article** l'abstract appare sotto il titolo sulla prima pagina.

Nel nostro articolo, abbiamo scritto un breve sommario con i comandi:

```
\begin{abstract}  
Vengono presentati alcuni interessanti  
teoremi del calcolo differenziale  
per funzioni di una variabile.  
\end{abstract}
```

Il risultato è un sommario in formato speciale, come si può vedere nella Figura 6. Il pacchetto **babel** traduce automaticamente l'inglese "abstract" in "somma-

## Sommario

Vengono presentati alcuni interessanti teoremi del calcolo differenziale per funzioni di una variabile.

Figura 6: Il sommario dell'articolo di esempio

rio": se si vuole una diversa traduzione della parola, si può usare il comando:

```
\def\abstractname{nuovo nome}
```

## 6 Gli oggetti flottanti

Alcuni oggetti, quali figure e tabelle, richiedono un trattamento particolare, perché non possono essere spezzate e stampate su più pagine.  $\text{\LaTeX}$  rende mobili (o flottanti<sup>10</sup>) le figure o le tabelle che non stanno sulla pagina corrente e le stampa eventualmente su una pagina successiva, riempiendo quella corrente con il testo successivo. Questo tipo di trattamento di tali oggetti è solitamente causa di frustrazione per l'utente che, abituato dai programmi ordinari di videoscrittura, vorrebbe vedere posizionata una tabella o una figura *esattamente* dove *lui* vuole e non dove il programma decide di posizionare l'oggetto. Prima di vedere come  $\text{\LaTeX}$  tratta gli oggetti flottanti e quali comandi si devono usare, è utile cercare di capire che questo tipo di trattamento è il più *razionale* per una corretta impaginazione del testo. Se l'utente decide dove mettere una tabella o una figura, poiché tale oggetto non può essere spezzato su più pagine, può succedere (e succede molto spesso) che la pagina contenente l'oggetto o quella precedente non sia “piena” di testo, ma presenti una parte bianca che è un effetto tipografico da evitare (tranne ovviamente che alla fine di un capitolo): una pagina tipograficamente ben formattata dovrebbe apparire da una certa distanza come un rettangolo uniformemente grigio.

$\text{\LaTeX}$  offre due ambienti di base per gli *oggetti mobili*, uno per le tabelle (*table*) e uno per le figure (*figure*).

Questi ambienti sono introdotti rispettivamente dai comandi

```
\begin{figure}[posizionamento]
```

```
...
```

```
\caption{testo della didascalia} \label{etichetta della figura}
```

```
\end{figure}
```

per le figure e

```
\begin{table}[posizionamento]
```

```
...
```

```
\caption{testo della didascalia} \label{etichetta della tabella}
```

```
\end{table}
```

per le tabelle.

All'interno dell'ambiente **figure** si dovrà creare la figura vera e propria (ad esempio con il comando `\includegraphics`), mentre all'interno dell'ambiente **table** deve essere nidificato un ambiente **tabular** o **array**, che crea la vera e propria tabella (**figure** e **table** sono solo “involucri” che consentono di creare riferimenti, inserire didascalie, ecc.).

---

<sup>10</sup>Useremo la dicitura classica, benché scorretta “flottante” anziché la traduzione letterale (“galleggiante”) proprio perché è ormai invalsa nell'uso.



Tutto il materiale presente in un ambiente `figure` o `table` viene trattato come un oggetto mobile, che verrà posizionato nel testo da  $\text{\LaTeX}$  seguendo le indicazioni contenute nel parametro opzionale `posizionamento`. Questo parametro indica dove gli oggetti possono essere spostati, secondo lo schema della tabella 13. È possibile indicare più di un parametro di posizionamento (ad esempio `[ht]`), seguendo l'ordine di preferenza. Se il posizionamento non viene specificato, le classi

| Carattere      | Permesso di spostare l'oggetto...                                                                                       |
|----------------|-------------------------------------------------------------------------------------------------------------------------|
| <code>h</code> | qui ( <i>here</i> ), nel punto esatto in cui compare nel testo. Si usa principalmente per piccoli oggetti mobili.       |
| <code>t</code> | in cima ( <i>top</i> ) a una pagina                                                                                     |
| <code>b</code> | in fondo ( <i>bottom</i> ) a una pagina                                                                                 |
| <code>p</code> | in una pagina speciale ( <i>page</i> ) contenente solo oggetti mobili.                                                  |
| <code>!</code> | senza considerare molti dei parametri interni <sup>a</sup> che potrebbero impedire a questo oggetto di essere spostato. |

<sup>a</sup>Come il numero massimo di oggetti mobili che è possibile avere in una pagina.

Tabella 13: Caratteri di trasferimento.

standard assumono che sia `[tbp]`.

Per forzare  $\text{\LaTeX}$  a porre un oggetto flottante dove lo mettiamo si può usare il pacchetto `float` che fornisce (tra le altre cose) l'opzione di posizionamento `H` (l'oggetto deve essere inserito esattamente nel punto indicato).

Se non si ha necessità di usare didascalie o riferimenti alla figura o alla tabella, il modo più semplice per posizionare figure e tabelle dove vogliamo è non usare gli ambienti `figure` o `table`, ma semplicemente `tabular` o `includegraphics`.

$\text{\LaTeX}$  posiziona ogni oggetto mobile che incontra secondo le indicazioni dell'autore, se un oggetto non può essere posizionato nella pagina corrente, viene accumulato nella coda delle *figure* o delle *tabelle*<sup>11</sup>. Quando viene iniziata una nuova pagina,  $\text{\LaTeX}$  controlla innanzitutto se è possibile creare una pagina speciale di soli oggetti mobili con quelli presenti nelle code. Se questo non è possibile, il primo oggetto in ciascuna coda è trattato come se fosse appena comparso nel testo:  $\text{\LaTeX}$  prova ancora a posizionarlo secondo il posizionamento che gli è stato attribuito (tranne per “`h`”, che non è più possibile). Tutti i nuovi oggetti mobili che compaiono nel testo vengono posti nelle rispettive code.  $\text{\LaTeX}$  mantiene strettamente l'ordine di apparizione originale per ogni tipo di oggetto mobile. Ecco perché

<sup>11</sup>Queste code sono di tipo FIFO (*First In First Out*, primo arrivato primo ad uscire).

una figura che non può essere posizionata spinge tutte le figure successive alla fine del documento. Per imporre a  $\text{\LaTeX}$  di stampare immediatamente tutti gli oggetti mobili rimasti nelle code si usano i comandi `\clearpage` o `\cleardoublepage`: con il primo si inizia una nuova pagina, con il secondo una nuova pagina dispari.

Con il comando `\caption{testo della caption}` è possibile specificare una didascalia per l'oggetto mobile.  $\text{\LaTeX}$  aggiunge la stringa “Figura” o “Tabella” e un numero progressivo. Il pacchetto `caption` permette di modificare in modo semplice l'apparenza delle didascalie.

Con `\label` e `\ref`, si possono creare riferimenti a oggetti mobili nel testo (al riguardo si veda la sezione 7).

Il pacchetto `subcaption` permette di trattare in modo semplice figure o tabelle “multiple” cioè composte di più “sottofigure” o “sottotabelle”.

Il pacchetto `float` permette di aggiungere nuovi ambienti (come, ad esempio, “grafico” o “algoritmo”) alla classe degli oggetti flottanti.

Molti editor hanno delle funzioni per facilitare l'inserimento di tabelle o anche importarle da fogli elettronici.

Nel nostro articolo abbiamo utilizzato due tipi di oggetti flottanti, una figura ed una tabella, con opzioni diverse.

Analizziamo la figura, che è stata inserita nell'articolo di esempio usando i seguenti comandi:

```
\begin{figure}[ht]
\begin{center}
\includegraphics[scale=.8]{derivata3}
\end{center}
\caption{Interpretazione geometrica della derivata}
\label{fig:def-derivata}
\end{figure}
```

Le opzioni `[ht]` indicano che l'oggetto può indifferentemente essere posizionato nel punto esatto in cui compare nel testo o, in alternativa, in cima alla pagina. Questo consente un'impaginazione adeguata dell'oggetto e del testo che precede e segue: non ci sono inutili spazi bianchi (si veda la figura dell'articolo in ultima pagina).

Analizziamo ora la tabella, che è stata inserita nell'articolo di esempio usando i seguenti comandi:

```

\begin{table}[H]
\begin{array}{|l|c|l|c|}
\hline
\hspace{2cm} f & f' & \hspace{1cm} f & f' \\
\hline
\hfill & \hfill & \hfill & \hfill \\
c \; ; \text{(costante)} & 0 & x & 1 \\
\hfill & \hfill & \hfill & \hfill \\
\frac{1}{x} & -\frac{1}{x^2} & x^2 & 2x \\
\hfill & \hfill & \hfill & \hfill \\
\sqrt{x} & \frac{1}{2\sqrt{x}} & x^n & n x^{n-1} \\
\hfill & \hfill & \hfill & \hfill \\
\sin x & \cos x & \cos x & -\sin x \\
\hfill & \hfill & \hfill & \hfill \\
\tg x & 1+\tg^2 x & \frac{1}{\cos^2 x} & e^x & e^x \\
\hfill & \hfill & \hfill & \hfill \\
\ln x & \frac{1}{x} & \arctg x & \frac{1}{1+x^2} \\
\hfill & \hfill & \hfill & \hfill \\
\hline
\end{array}
\caption{Derivate di alcune funzioni elementari.}\label{tab:derivate}
\end{table}

```

Il parametro opzionale [H], messo a disposizione dal package `float`, dice a  $\text{\LaTeX}$  che la tabella va inserita *esattamente* nel punto in cui compare nel testo. Questo consente all'autore di controllare perfettamente il posizionamento dell'oggetto, ma a prezzo di un'impaginazione non ottimale: si noti lo spazio bianco presente in fondo alla seconda pagina dell'articolo (si veda la figura in ultima pagina).

## 7 I riferimenti incrociati

A volte in un documento è necessario fare riferimento a oggetti, figure, pagine. Spesso infatti leggiamo frasi del tipo “Come abbiamo visto nel Capitolo 3” oppure “Si veda la Figura 7 a pagina 45” o ancora “Dal Teorema 4 si deduce che”.

Gli ambienti “etichettabili” ovvero che consentono di avere un riferimento che può essere richiamato in altre parti del documento sono:

- `table` e `figure` (oggetti flottanti)
- `equation`<sup>12</sup> ed enunciati (teoremi, proposizioni, ecc.)
- sezioni, capitoli, ecc.
- item di elenchi numerati, note, ecc.

<sup>12</sup>Si ricordi che negli ambienti che contengono più equazioni, è possibile etichettare ciascuna di esse separatamente.

L'oggetto a cui si deve fare riferimento avrà un'etichetta `\label{nome etichetta}` che può essere poi richiamata con il comando `\ref{nome etichetta}`. L'etichetta (`label`) deve essere posizionata dopo il comando `\caption` negli oggetti flottanti; dopo il comando di sezionamento per le parti del documento (capitoli, sezioni, ecc.); dopo il comando di apertura dell'ambiente per teoremi, definizioni, equazioni, ecc. Si può fare riferimento anche alla pagina in cui si trova l'oggetto con il comando `\pageref{nome etichetta}`

Esistono, come sempre, pacchetti (come `smartref`, `refstyle`, `varioref` ed altri) che consentono di personalizzare i riferimenti e scrivere, ad esempio, “la tabella della pagina seguente”.

Un file che contiene riferimenti incrociati va compilato almeno 2 volte.

In molti editor dedicati Inoltre, se si scrive `\label{}` oppure `\ref{}` (ovvero se si scrive il comando lasciando vuote le parentesi graffe, che contengono il parametro obbligatorio), compaiono in una finestra le etichette già usate, nel primo caso per evitare di ripetere lo stesso nome, nel secondo per scegliere a quale oggetto fare riferimento (viene inserito con il tasto “Invio”). Inoltre, un doppio click sul nome, porta all'oggetto etichettato (e dunque è possibile vedere se il riferimento scelto è corretto).

Si consiglia di attribuire ai vari oggetti etichette evocative anche del *tipo* di oggetto etichettato: ad esempio, `fig:nome` per le figure, `tab:nome` per le tabelle, ecc. (si veda il listato dell'articolo di esempio).

Esistono poi il comando `\pageref`, con cui si può richiamare il numero di pagina in cui si è posta un'etichetta e il comando `\eqref` del pacchetto `amsmath`, con cui si può richiamare un'equazione (sempre etichettata).

## 8 La bibliografia

In  $\text{\LaTeX}$ , la bibliografia può essere creata in più modi:

- ✓ utilizzando l'ambiente `thebibliography`;
- ✓ utilizzando il programma `BibTeX`;
- ✓ utilizzando il programma `biber`.

Il primo metodo è molto veloce e permette di avere la bibliografia all'interno dello stesso documento sul quale si lavora; il secondo e il terzo prevedono la creazione di un data base di riferimenti bibliografici esterno al documento (che deve essere quindi trasmesso insieme al documento principale se questo deve essere compilato da un'altra persona), che può essere utilizzato in documenti diversi e permette un aggiornamento più agile della bibliografia di tutti i documenti collegati.

Noi ci occuperemo solo del primo metodo.

## 8.1 L'ambiente `thebibliography`

L'ambiente `thebibliography` funziona più o meno come `enumerate` e viene aperto e chiuso, rispettivamente, dai comandi:

```
\begin{thebibliography}{x}  
\bibitem[etichetta]{sigla} autore dell'opera, titolo (in corsivo),  
editore, anno di pubblicazione  
...  
\end{thebibliography}
```

Il parametro opzionale `x` racchiuso nella seconda parentesi graffa del comando di apertura può essere un numero (9 oppure 99 oppure...) o una parola e indica la larghezza massima delle etichette da applicare ai riferimenti bibliografici: “9” per numeri di una cifra, “99” per numeri di due cifre, “Michelangelo Buonarroti” per etichette personalizzate.

Il comando `\bibitem[etichetta]{sigla} nome articolo` viene utilizzato per elencare le voci da inserire nella bibliografia. Il parametro opzionale `etichetta` indica come verrà indicato nell'elenco il riferimento bibliografico (se manca viene usato di default [numero]). Il parametro “sigla” permette di assegnare al documento citato un nome, che consentirà di richiamarlo all'interno del documento mediante il comando: `\cite{sigla}` oppure `\cite{sigla1, sigla2}`. Questo comando stamperà tra parentesi quadre il numero assegnato all'articolo o l'etichetta personalizzata ad esso attribuita (in genere, nome dell'autore ed anno).

Esiste anche il comando

```
\nocite{sigla1, sigla45}
```

che permette di inserire in bibliografia dei testi che non sono citati nel documento.

Questo è un esempio di bibliografia realizzata con l'ambiente `thebibliography`:

---

```
\begin{thebibliography}{9}  
\bibitem{lamport-latex} Leslie Lamport. \emph{\LaTeX: a document  
preparation system}. Addison-Wesley, 1994.  
\bibitem{companion} Michel Goossens, Frank Mittelbach, and Alexander  
Samarin. \emph{The \LaTeX\ Companion}. Addison-Wesley, 1994.  
\end{thebibliography}
```

---

il risultato è:

# Bibliografia

- [1] Leslie Lamport. *TEX: a document preparation system*. Addison-Wesley, 1994.
- [2] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The TEX Companion*. Addison-Wesley, 1994.

Nel testo, avremo:

Si veda~\cite{lamport-latex} per maggiori dettagli.

ovvero:

Si veda [1] per maggiori dettagli.

Questo è invece un esempio di bibliografia con etichette personalizzate:

```
\begin{thebibliography}
\bibitem[Lamport 94]{lamport-latex} Leslie Lamport. \emph{\LaTeX: a document
preparation system}. Addison-Wesley, 1994.
\bibitem[Goossens et al. 94]{companion} Michel Goossens, Frank Mittelbach,
and Alexander Samarin. \emph{The \LaTeX\ Companion}. Addison-Wesley, 1994.
\end{thebibliography}
```

che produce:

# Bibliografia

- |                      |                                                                                                            |
|----------------------|------------------------------------------------------------------------------------------------------------|
| [Lamport 94]         | Leslie Lamport. <i>TEX: a document preparation system</i> . Addison-Wesley, 1994.                          |
| [Goossens et al. 94] | Michel Goossens, Frank Mittelbach, and Alexander Samarin. <i>The TEX Companion</i> . Addison-Wesley, 1994. |

In questo caso, nel testo apparirà:

Si veda~\cite{lamport-latex} per maggiori dettagli.

ovvero:

Si veda [Lamport 94] per maggiori dettagli.

È anche possibile indicare capitoli, pagine, ecc. nel riferimento passando un parametro opzionale (tra parentesi quadre) al comando `\cite`. Ad esempio:

Si veda `\cite[pag. 187]{lamport-latex}` per maggiori dettagli.

produce

Si veda [Lamport 94, pag. 187] per maggiori dettagli.

Il “nome” della bibliografia è gestito da `babel` (se caricato) ed è

- *Bibliografia* (Bibliography) per le classi `book` e `report`
- *Riferimenti bibliografici* (References) per la classe `article`

Il nome può essere cambiato con il comando:

```
\def\refname{nuovo nome}
```

Esistono molti pacchetti che permettono una gestione più flessibile della bibliografia.

È molto usato il pacchetto `natbib` che permette di scegliere facilmente sia la “forma” della citazione (tra parentesi tonde, quadre, graffe, come nota) sia lo “stile” (numerico, autore-anno, ecc.)

Nel nostro articolo, la bibliografia è stata inserita in questo modo:

```
\begin{thebibliography}{99}  
\bibitem{R}{Ricci G., \emph{Analisi matematica: vol. I},  
Milano Libreria Editrice, 1962}  
\end{thebibliography}
```

## Riferimenti bibliografici

- [1] Ricci G., *Analisi matematica: vol. I*. Milano Libreria Editrice, 1962

Figura 7: La bibliografia dell’articolo di esempio

## 9 L’indice

L’*indice* viene creato con il comando `\tableofcontents`. Dopo questo comando non inizia automaticamente una nuova pagina, ma questo deve essere forzato col comando `\newpage`. È possibile evitare l’aggiunta di capitoli, sezioni, ecc. all’indice

usando la forma con asterisco del comando `chapter`, `section`, ecc. È possibile aggiungere un campo all'indice con il comando `\addcontentsline{toc}{livello}{campo}` che aggiunge il campo al livello indicato (sezione, capitolo). Ad esempio, poiché la bibliografia non è inserita nell'indice, si può scrivere

```
\addcontentsline{toc}{chapter}{Bibliografia}
```

per inserire la bibliografia nell'indice come un capitolo.

Accanto all'indice, con sintassi identica, esistono la *lista delle figure* (lof) inserita con il comando `\listoffigures` e la *lista delle tabelle* (lot) inserita con il comando `\listoftables`.

Può anche essere creata una *lista dei simboli*, ma occorre inserire nel preambolo il package `listofsymbols` oppure il package `symbolindex`, in quanto l'opzione in questione non è prevista nelle classi standard.

## 10 Il risultato

Prima di presentare, nella prossima pagina, tutto il breve articolo prodotto, vogliamo sottolineare alcuni aspetti di formattazione che erano stati presentati nella prima parte o che non sono stati spiegati nelle pagine precedenti perché troppo tecnici.

- i) Alcune frazioni nel testo (ad esempio le due che precedono la figura) sono state prodotte con il comando `\dfrac` di `amsmath` che dà il formato display: risultano perciò più grandi dell'usuale.
- ii) Il limite alla quartultima riga di pagina 1 è in modalità testo e quindi la posizione di  $x \rightarrow x_0$  è a sinistra e non sotto il simbolo di limite.
- iii) Alcune formule matematiche sono in modalità display (e non numerate) mentre altre sono numerate: queste ultime sono create con il comando `\equation` (o simile).
- iv) L'ultimo corollario ha le formule riquadrate e con una numerazione diversa. Il riquadro può essere ottenuto con il comando `\boxed{ equazione }` del pacchetto `amsmath`. La numerazione "subordinata" si ottiene con l'ambiente `subequations` (sempre del pacchetto `amsmath`) che fa sì che tutte le equazioni racchiuse in quell'ambiente ricevano una numerazione subordinata: un comando `\label` posto subito dopo `\begin{subequations}` produrrà referenze al numero dell'equazione (nel nostro caso 2), mentre il comando `\label` posto in un'equazione darà le referenze nel modo usuale. Nel nostro caso l'ambiente `subequations` racchiude un ambiente `gather` che produce gruppi di equazioni non allineate (ognuna è centrata separatamente).



|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> \documentclass[12pt]{article}  %Pacchetti usati nel documento \usepackage[italian]{babel} %Serve per la traduzione in italiano di alcune parole e per la sillabazione \usepackage[utf8]{inputenc} \usepackage[T1]{fontenc}%Servono per il riconoscimento e la stampa dei caratteri accentati inseriti direttamente da tastiera \usepackage{graphicx}%Consente l'inclusione delle immagini nel documento \usepackage{amsmath}%Pacchetto che definisce nuovi ambienti matematici \usepackage{amsthm}%Pacchetto per i teoremi \usepackage{amsfonts}%Pacchetto che introduce nuovi caratteri matematici \usepackage{enumerate}%Pacchetto per la personalizzazione degli elenchi numerati \usepackage{float}%Pacchetto per la gestione degli oggetti flottanti \usepackage[margin=2cm]{geometry}%Pacchetto per la gestione del layout di pagina  %Qui vengono definiti gli ambienti di tipo teorema che saranno usati nel testo \newtheorem{thm}{Teorema}[section] \newtheorem{cor}[thm]{Corollario} \newtheorem{lem}[thm]{Lemma} \newtheorem{prop}[thm]{Proposizione} \theoremstyle{definition} \newtheorem{defn}[thm]{Definizione} \theoremstyle{remark} \newtheorem{rem}[thm]{Osservazione}  %Qui vengono definiti i nuovi comandi che vengono usati nel testo \newcommand{\D}{\displaystyle} \newcommand{\T}{\textstyle} \newcommand{\dlim}{\displaystyle \lim} \newcommand{\vps}{\varepsilon}  %Qui vengono definiti i nomi degli operatori matematici che saranno usati nel testo \DeclareMathOperator{\sen}{sen} \DeclareMathOperator{\vtg}{tg} \DeclareMathOperator{\arcsen}{arcsen} \DeclareMathOperator{\arctg}{arctg}  %Topmatter: contiene il titolo, l'autore e la data dell'articolo \title{Alcuni importanti teoremi del Calcolo differenziale} \author{Prof. Pico de' Paperis\Università di Paperopoli} \date{25 Dicembre 2023}  %Qui inizia il documento vero e proprio \begin{document} \maketitle %Stampa il titolo, l'autore e la data nel documento  \begin{abstract}%Serve per scrivere il sommario dell'articolo in formato speciale Vengono presentati alcuni interessanti teoremi del calcolo differenziale per funzioni di una variabile. \end{abstract}  \section{Derivata di una funzione.} Vediamo ora la definizione di derivata. Partiremo dall'interpretazione geometrica, ma vedremo subito altre interpretazioni. Consideriamo la figura \ref{fig:def-derivata}: il punto <math>P</math> ha coordinate <math>(x_0, y_0)</math>, il </pre>                                                                                                                                                                                                                                    | <pre> punto <math>Q</math> ha coordinate <math>(x_1, y_1)</math> (con <math>f(x_0)=y_0</math> e <math>f(x_1)=y_1</math>), la differenza <math>x_1-x_0</math> sarà indicata con <math>\Delta x</math> e la differenza <math>y_1-y_0</math> con <math>\Delta y</math>.  \begin{figure}[ht]%Inclusione di una figura nell'ambiente flottante figure. \begin{center} \includegraphics[scale=.8]{derivata3} \end{center} \caption{Interpretazione geometrica della derivata}\label{fig:def-derivata} %Vengono inserite la didascalia e l'etichetta che consentirà di richiamare la figura nel testo \end{figure}  Il coefficiente angolare della retta secante <math>PQ</math> è <math>\frac{\Delta y}{\Delta x} = \frac{f(x_1)-f(x_0)}{x_1-x_0}</math>. Poiché abbiamo detto che la retta tangente sarà la posizione limite della retta secante quando il punto <math>Q</math> si avvicina a <math>P</math> abbiamo che il coefficiente angolare <math>m</math> della retta tangente nel punto <math>P</math> è dato da <math>m = \lim_{x_1 \rightarrow x_0} \frac{f(x_1)-f(x_0)}{x_1-x_0}</math>.  Questa espressione non è però molto comoda. Indichiamo <math>\Delta x</math> con <math>x_1-x_0</math> e <math>\Delta y</math> avremo allora <math>y_0=f(x_0)</math> e <math>y_1=f(x_0+\Delta x)</math> e il limite precedente diventa: <math>\lim_{\Delta x \rightarrow 0} \frac{f(x_0+\Delta x)-f(x_0)}{\Delta x}</math>.  \begin{defn}%Uso dell'ambiente per le definizioni La quantità <math>x_1-x_0=\Delta x</math> si dice \emph{incremento della variabile} <math>\Delta x</math>.  La quantità <math>y_1-y_0=\Delta y</math> o, meglio, <math>f(x_0+\Delta x)-f(x_0)=\Delta f</math> si dice \emph{incremento della funzione} <math>\Delta f</math> relativo all'incremento <math>\Delta x</math>.  La quantità <math>\frac{f(x_0+\Delta x)-f(x_0)}{\Delta x} = \frac{\Delta f}{\Delta x}</math> si dice \emph{rapporto incrementale della funzione} <math>f'(x)</math> relativo all'intervallo <math>[x_0, x_0+\Delta x]</math>. \index{rapporto incrementale} \end{defn}  Allora il coefficiente angolare della retta tangente è il limite del rapporto incrementale, quando l'incremento tende a 0.  \begin{defn} Sia <math>f:(a,b) \rightarrow \mathbb{R}</math>; <math>f'</math> si dice \emph{derivabile} in un punto <math>x_0</math> se il limite: <math display="block">\lim_{\Delta x \rightarrow 0} \frac{f(x_0+\Delta x)-f(x_0)}{\Delta x}</math> esiste finito. Tale limite si dice \emph{derivata prima} (o \emph{derivata}) di <math>f</math> nel punto <math>x_0</math> e si indica con uno dei seguenti simboli: <math display="block">f'(x_0); \quad \frac{df}{dx}(x_0); \quad \frac{d}{dx}f(x_0); \quad \dot{f}(x_0).</math> Quindi \begin{equation}\label{eq:derivata} \lim_{\Delta x \rightarrow 0} \frac{f(x_0+\Delta x)-f(x_0)}{\Delta x} = f'(x_0) \end{equation} \end{defn}  \dotfill  Abbiamo il seguente risultato: \begin{prop}%Uso dell'ambiente per le proposizioni Sia <math>f:(a,b) \rightarrow \mathbb{R}</math> derivabile in <math>x_0</math> in <math>(a,b)</math>. Allora <math>f'</math> è continua in <math>x_0</math>.  In particolare se <math>f'</math> è derivabile in tutto <math>(a,b)</math> allora è continua in <math>(a,b)</math>.  \end{prop} \begin{proof} Poiché <math>f'</math> è derivabile vale la \eqref{eq:derivata} e quindi <math>f'(x_0+\Delta x) =</math> </pre> |
| <pre> f(x_0)=f'(x_0)\Delta x+\eps(x) \Delta x\$ con <math>\eps(x)</math> infinitesimo per <math>\Delta x \rightarrow 0</math>. Passando al limite per <math>\Delta x \rightarrow 0</math> si ottiene allora <math>\lim_{\Delta x \rightarrow 0} (f(x_0+\Delta x)-f(x_0)) = 0</math> e quindi la funzione <math>f'</math> è continua. \end{proof}  \dotfill[2mm]  Ed ecco una tabella con le derivate di alcune funzioni elementari: \begin{table}[H] \begin{array}{ l l l } \hline \hspace{2cm} f &amp; f' &amp; \hspace{1cm} f &amp; f' \\ \hline \fill &amp; \fill &amp; \fill &amp; \fill \\ c \text{ (costante)} &amp; 0 &amp; x &amp; 1 \\ \fill &amp; \fill &amp; \fill &amp; \fill \\ \frac{1}{x^2} &amp; -\frac{2}{x^3} &amp; x^2 &amp; 2x \\ \fill &amp; \fill &amp; \fill &amp; \fill \\ \sqrt{x} &amp; \frac{1}{2\sqrt{x}} &amp; x^n &amp; n x^{n-1} \\ \fill &amp; \fill &amp; \fill &amp; \fill \\ \cos x &amp; -\sin x &amp; \sin x &amp; \cos x \\ \fill &amp; \fill &amp; \fill &amp; \fill \\ e^x &amp; e^x &amp; \ln x &amp; \frac{1}{x} \\ \fill &amp; \fill &amp; \fill &amp; \fill \\ \hline \end{array} \caption{Derivate di alcune funzioni elementari.}\label{tab:derivate} \end{table}  \dotfill \section{Teoremi fondamentali del calcolo differenziale.} \subsection{Punti stazionari. I Teoremi di Rolle e Lagrange.} \begin{thm} [Teorema di Fermat]\label{thm:Teorema Fermat} Sia <math>f:(a,b) \rightarrow \mathbb{R}</math> derivabile in <math>x_0</math> in <math>(a,b)</math>. Se <math>x_0</math> è un punto di estremo locale allora <math>f'(x_0)=0</math>. \end{thm}  \begin{rem} Il teorema precedente esprime il fatto che un punto di estremo locale in cui una funzione è derivabile è un punto stazionario. Bisogna prestare attenzione a due fatti: \begin{enumerate} [a)] \item Non tutti i punti stazionari sono punti di estremo locale: la derivata della funzione <math>f(x)=x^3</math> si annulla in <math>x=0</math>, ma tale punto non è di estremo (la funzione è crescente in tutto <math>\mathbb{R}</math>). \item Non tutti i punti di estremo sono stazionari: infatti possiamo avere punti di estremo anche agli estremi dell'intervallo di definizione della funzione o in punti in cui la funzione non è derivabile. \end{enumerate} \end{rem}  Dimostriamo allora il seguente: \begin{lem} [Teorema di Rolle]\label{thm:teorema Rolle} Sia <math>f:[a,b] \rightarrow \mathbb{R}</math> una funzione continua nell'intervallo chiuso <math>[a,b]</math> e derivabile nell'intervallo aperto <math>(a,b)</math>. Se <math>f(a)=f(b)</math> esiste un punto <math>\xi</math> in <math>(a,b)</math> tale che <math>f'(\xi)=0</math>. \end{lem} </pre> | <pre> Veniamo ora al teorema di Lagrange: \begin{thm} [Teorema di Lagrange]\label{thm:teorema Lagrange} Sia <math>f:[a,b] \rightarrow \mathbb{R}</math> una funzione continua nell'intervallo chiuso <math>[a,b]</math> e derivabile nell'intervallo aperto <math>(a,b)</math>. Allora \text{esiste un punto <math>\xi</math> in <math>(a,b)</math> tale che} <math>f'(\xi) = \frac{f(b)-f(a)}{b-a}</math>. \end{thm}  \begin{proof} Si veda \cite{R}. \end{proof}  \begin{cor}[Test di monotonia]\label{thm:test monotonia} Sia <math>f:(a,b) \rightarrow \mathbb{R}</math> derivabile. \begin{subequations} \begin{gather} \boxed{f'(x) &gt; 0} \text{, for all } x \text{ in } (a,b) \rightarrow f \text{ \textit{crescente}} \\ \boxed{f'(x) \geq 0} \text{, for all } x \text{ in } (a,b) \rightarrow f \text{ \textit{non decrescente}} \\ \boxed{f'(x) &lt; 0} \text{, for all } x \text{ in } (a,b) \rightarrow f \text{ \textit{decrescente}} \\ \boxed{f'(x) \leq 0} \text{, for all } x \text{ in } (a,b) \rightarrow f \text{ \textit{non crescente}} \end{gather} \end{subequations} \end{cor}  \begin{thebibliography}{99} \bibitem{R}Ricci G., \emph{Analisi matematica: vol. I}. Milano Libreria Editrice, 1962 \end{thebibliography} \end{document} </pre>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

# Alcuni importanti teoremi del Calcolo differenziale

Prof. Pico de' Paperis  
Università di Paperopoli

25 Dicembre 2023

## Sommario

Vengono presentati alcuni interessanti teoremi del calcolo differenziale per funzioni di una variabile.

## 1 Derivata di una funzione.

Vediamo ora la definizione di derivata. Partiremo dall'interpretazione geometrica, ma vedremo subito altre interpretazioni. Consideriamo la figura 1: il punto  $P$  ha coordinate  $(x_0, y_0)$ , il punto  $Q$  ha coordinate  $(x_1, y_1)$  (con  $f(x_0) = y_0$  e  $f(x_1) = y_1$ ), la differenza  $x_1 - x_0$  sarà indicata con  $\Delta x$  e la differenza  $y_1 - y_0$  con  $\Delta y$ .

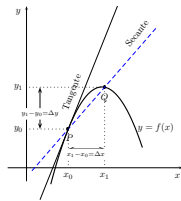


Figura 1: Interpretazione geometrica della derivata

Il coefficiente angolare della retta secante  $PQ$  è  $\frac{y_1 - y_0}{x_1 - x_0} = \frac{\Delta y}{\Delta x}$ . Poiché abbiamo detto che la retta tangente sarà la posizione limite della retta secante quando il punto  $Q$  si avvicina a  $P$  abbiamo che il coefficiente angolare  $m$  della retta tangente nel punto  $P$  è dato da  $m = \lim_{x_1 \rightarrow x_0} \frac{y_1 - y_0}{x_1 - x_0}$ .

Questa espressione non è però molto comoda. Indichiamo  $x_1$  con  $x_1 = x_0 + \Delta x$ : avremo allora  $y_0 = f(x_0)$  e  $y_1 = f(x_1) = f(x_0 + \Delta x)$  e il limite precedente diventa:

$$\lim_{\Delta x \rightarrow 0} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x}.$$

1

**Definizione 1.1.** La quantità  $x_1 - x_0 = \Delta x$  si dice *incremento della variabile*  $x$ .

La quantità  $y_1 - y_0 = \Delta y$  o, meglio,  $f(x_0 + \Delta x) - f(x_0) = \Delta f$  si dice *incremento della funzione*  $f$  relativo all'incremento  $\Delta x$ .

La quantità

$$\frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} = \frac{\Delta f}{\Delta x}$$

si dice *rapporto incrementale della funzione*  $f(x)$  relativo all'intervallo  $[x_0, x_0 + \Delta x]$ .

Allora il coefficiente angolare della retta tangente è il limite del rapporto incrementale, quando l'incremento tende a 0.

**Definizione 1.2.** Sia  $f : (a, b) \rightarrow \mathbb{R}$ ;  $f$  si dice *derivabile* in  $x_0$  se il limite:

$$\lim_{\Delta x \rightarrow 0} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x}$$

esiste finito. Tale limite si dice *derivata prima* (o *derivata*) di  $f$  nel punto  $x_0$  e si indica con uno dei seguenti simboli:

$$f'(x_0); \quad \frac{df}{dx} x_0; \quad Df(x_0); \quad f(x_0).$$

Quindi

$$\lim_{\Delta x \rightarrow 0} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} = f'(x_0) \quad (1)$$

Abbiamo il seguente risultato:

**Proposizione 1.3.** Sia  $f : (a, b) \rightarrow \mathbb{R}$  derivabile in  $x_0 \in (a, b)$ . Allora  $f$  è continua in  $x_0$ .

In particolare se  $f$  è derivabile in tutto  $(a, b)$  allora è continua in  $(a, b)$ .

**Dimostrazione.** Poiché  $f$  è derivabile vale la (1) e quindi  $f(x_0 + \Delta x) - f(x_0) = f'(x_0)\Delta x + \varepsilon(x)\Delta x$  con  $\varepsilon(x)$  infinitesimo per  $\Delta x \rightarrow 0$ . Passando al limite per  $\Delta x \rightarrow 0$  si ottiene allora  $\lim_{\Delta x \rightarrow 0} f(x_0 + \Delta x) - f(x_0) = 0$  e quindi la funzione  $f$  è continua.  $\square$

Ed ecco una tabella con le derivate di alcune funzioni elementari:

2

| $f$                      | $f'$                                             | $f$        | $f'$              |
|--------------------------|--------------------------------------------------|------------|-------------------|
| $c$ (costante)           | 0                                                | $x$        | 1                 |
| $\frac{1}{x}$            | $-\frac{1}{x^2}$                                 | $x^2$      | $2x$              |
| $\sqrt{x} \quad (x > 0)$ | $\frac{1}{2\sqrt{x}}$                            | $x^n$      | $nx^{n-1}$        |
| $\sin x$                 | $\cos x$                                         | $\cos x$   | $-\sin x$         |
| $\operatorname{tg} x$    | $1 + \operatorname{tg}^2 x = \frac{1}{\cos^2 x}$ | $e^x$      | $e^x$             |
| $\ln x$                  | $\frac{1}{x}$                                    | $\arctg x$ | $\frac{1}{1+x^2}$ |

Tabella 1: Derivate di alcune funzioni elementari.

**Dimostrazione.** Si veda [1].  $\square$

**Corollario 2.5** (Test di monotonia). Sia  $f : (a, b) \rightarrow \mathbb{R}$  derivabile.

$$f'(x) > 0 \quad \forall x \in (a, b) \Rightarrow f \text{ crescente} \quad (2a)$$

$$f'(x) \geq 0 \quad \forall x \in (a, b) \Rightarrow f \text{ non decrescente} \quad (2b)$$

$$f'(x) < 0 \quad \forall x \in (a, b) \Rightarrow f \text{ decrescente} \quad (2c)$$

$$f'(x) \leq 0 \quad \forall x \in (a, b) \Rightarrow f \text{ non crescente} \quad (2d)$$

## Riferimenti bibliografici

[1] Ricci G., *Analisi matematica: vol. I*. Milano Libreria Editrice, 1962

4

## 2 Teoremi fondamentali del calcolo differenziale.

### 2.1 Punti stazionari. I Teoremi di Rolle e Lagrange.

**Teorema 2.1** (Teorema di Fermat). Sia  $f : (a, b) \rightarrow \mathbb{R}$  derivabile in  $x_0 \in (a, b)$ . Se  $x_0$  è un punto di estremo locale allora  $f'(x_0) = 0$ .

**Osservazione 2.2.** Il teorema precedente esprime il fatto che un punto di estremo locale in cui una funzione è derivabile è un punto stazionario. Bisogna prestare attenzione a due fatti:

- Non tutti i punti stazionari sono punti di estremo locale: la derivata della funzione  $f(x) = x^3$  si annulla in  $x = 0$ , ma tale punto non è di estremo (la funzione è crescente in tutto  $\mathbb{R}$ ).
- Non tutti i punti di estremo sono stazionari: infatti possiamo avere punti di estremo anche agli estremi dell'intervallo di definizione della funzione o in punti in cui la funzione non è derivabile.

Dimostriamo allora il seguente:

**Lemma 2.3** (Teorema di Rolle). Sia  $f : [a, b] \rightarrow \mathbb{R}$  una funzione continua nell'intervallo chiuso  $[a, b]$  e derivabile nell'intervallo aperto  $(a, b)$ . Se  $f(a) = f(b)$  esiste un punto  $c \in (a, b)$  tale che  $f'(c) = 0$ .

Veniamo ora al teorema di Lagrange:

**Teorema 2.4** (Teorema di Lagrange). Sia  $f : [a, b] \rightarrow \mathbb{R}$  una funzione continua nell'intervallo chiuso  $[a, b]$  e derivabile nell'intervallo aperto  $(a, b)$ . Allora

$$\text{esiste un punto } c \in (a, b) \text{ tale che } f'(c) = \frac{f(b) - f(a)}{b - a}.$$

3

# 11 Le presentazioni

## 11.1 beamer

Il pacchetto **beamer** è un ottimo strumento per effettuare presentazioni sia di tipo didattico che scientifico, utilizzando un proiettore ed ha a disposizione un notevole numero di “effetti” video <sup>13</sup> come transizioni e box.

L’output di beamer è un file **pdf** e pertanto è portabile su tutte le piattaforme per cui sia disponibile un visualizzatore Adobe Reader<sup>14</sup>.

I file **pdf** ottenuti sono, in certo senso, interattivi, per cui ogni slide può contenere anche i “pulsanti” per muoversi nella sequenza di slide che costituiscono la presentazione ed è possibile inoltre avere una barra contenente una sorta di indice generale della presentazione sempre visibile che non solo danno conto dello stato di avanzamento della presentazione, ma costituiscono hyperlink utili per navigare rapidamente attraverso la presentazione.

Poiché beamer è una classe il suo uso per un utente **L<sup>A</sup>T<sub>E</sub>X** non si discosta molto da quello delle altre classi e, come sempre in **L<sup>A</sup>T<sub>E</sub>X**, permette di concentrarsi sul contenuto più che sull’aspetto esteriore del documento.

### 11.1.1 Il preambolo

Un preambolo minimo per una presentazione fatta in beamer può essere di questo tipo

```
\documentclass{beamer}
\mode<presentation>
{\usetheme{Boadilla} \usecolortheme{albatross}}
\usefonttheme{structuresmallcapsserif}
\usepackage[italian]{babel}
% o altra lingua
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}

\title[Titolo Abbreviato] {Titolo completo}
\date{15 agosto 2024}
\author {Autore 1 \inst{1} \and Autore 2\inst{2}}
\institute[Università Cattolica di Milano]
```

---

<sup>13</sup>L’autore del pacchetto, nel manuale (più di 200 pagine), dà tra le molte raccomandazioni per ottenere una presentazione efficace, quella di non eccedere con sfondi vistosi ed effetti video, che tendono a distrarre l’uditorio.

<sup>14</sup>Non con tutti i visualizzatori di file **pdf** è possibile vedere gli effetti che vengono creati come ad esempio la trasparenza. Inoltre è indispensabile impostare la visualizzazione a pieno schermo.

```

\inst{1}
Dipartimento di Matematica\\
Università Cattolica di Milano
\and
\inst{2}%
Dipartimento di Economia\\
Università Cattolica di Milano}

\pgfdeclareimage[height=0.5cm]{logo}{nome del file del logo}
\logo{\pgfuseimage{logo}}

```

Nel preambolo è già possibile vedere alcune caratteristiche di beamer: i “modi” e i “temi”.

Con uno stesso file beamer è infatti possibile ottenere diversi documenti: la presentazione, un file contenente le slides pronte per la stampa (quindi senza effetti di transizione, forse meno colorate, spesso con più slides per pagina), un file contenente una versione a stampa della presentazione sotto forma di articolo. Il file beamer può contenere quindi istruzioni e testo specifici per ognuna o per più o tutte queste modalità: nell’esempio le istruzioni

```
\usetheme{Boadilla} e \usecolortheme{albatross}
```

sono argomento del comando

```
\mode<presentation>
```

e quindi si riferiscono solo alla modalità presentazione, mentre l’istruzione

```
\usefonttheme{structuresmallcapsserif}
```

si riferisce a tutte le modalità possibili.

Quasi tutti gli aspetti globali della presentazione sono invece controllati da un “tema”: ci sono temi globali che definiscono tutti gli elementi della presentazione (colori, caratteri, la presenza o meno di una barra di navigazione) e temi che modificano solo alcuni aspetti: l’istruzione `\usetheme{Boadilla}` è del primo tipo, l’istruzione `\usecolortheme{albatross}`, del secondo.

La scelta di un tema dipende ovviamente, oltre che dal gusto personale, dall’ambito in cui verrà tenuta la presentazione, dalla luce dell’ambiente, e così via. beamer ha a disposizione di default moltissimi temi globali, modificabili con opzioni diverse o variando temi locali; comunque una semplice ricerca su Google con parole chiave “beamer theme” fornisce molte migliaia di siti.

È utile osservare che beamer, come molti altri pacchetti e programmi per presentazioni, usa di default caratteri sans-serif.

Si possono inserire titolo (ed eventualmente sottotitolo), data ed autore con le solite istruzioni: si noti l’istruzione `\inst` che serve a dichiarare l’affiliazione dell’autore ad una istituzione o azienda.

L'istruzione `\logo` serve ad inserire il logo della propria istituzione o della conferenza

### 11.1.2 Il corpo del documento

La classe `beamer` somiglia alla classe `book`, non ci sono i capitoli, ma le parti (`part`), le sezioni e le sottosezioni (`section` e `subsection`). Questa suddivisione è quella che viene usata nell'indice e (se esiste) nella barra di navigazione.

L'analogo della pagina è il *frame* il cui contenuto può essere visualizzato in un'unica soluzione o in tempi successivi, eventualmente con effetti. Il frame stesso può essere introdotto da un *effetto di transizione* che può essere parzialmente modificato. Il frame può essere introdotto o mediante un ambiente (`\begin{frame}` e `\end{frame}`) o mediante un comando (`\frame{testo del frame}`) che esiste per compatibilità con versioni più vecchie di beamer.

```
\begin{frame}
\transglitter
testo molto interessante
\end{frame}
```

Mediante il comando `\titlepage` viene creato all'interno di un frame una pagina con titolo, autore, ecc.

Invece, all'interno di un frame il comando `\frametitle{titolo del frame}` crea un titolo evidenziato per quel frame (il modo in cui viene evidenziato dipende dal tema scelto).

Generalmente il contenuto di un frame deve risultare in un'unica pagina a video, l'opzione `[allowframebreaks]` permette al contenuto di espandersi su più pagine.

Come detto il contenuto di un frame può essere visualizzato a passi: ognuno di questi successivi passi viene detto *slide* e il contenuto e il tempo di visualizzazione di ogni slide può essere modificato in molti modi.

Il comando più semplice per una visualizzazione passo-passo è `\pause`: beamer mostra il contenuto del frame fino a tale comando e aspetta un input dall'utente (clic del mouse, pressione di un tasto) per avanzare.

Un altro comando, molto più flessibile, è `\onslide<...>{testo}` che mostra il testo contenuto tra parentesi graffe nelle slides specificate tra `<>`: è possibile indicare una sola slide `<3>`, più slides `<3,5,9>`, un intervallo `<2-4>`, tutte le slides fino a (a partire da) una specificata `<-4>` (`<2->`).

All'interno di beamer molti comandi  $\text{\LaTeX}$  e ambienti (come `theorem`) accettano come opzione `<...>`: ad esempio

```

\begin{frame}
  \textbf{Linea in grassetto su tutte e tre le slides.}
  \textbf<2>{Linea in grassetto solo sulla seconda slide.}
  \textbf<3>{Linea in grassetto solo sulla terza slide.}
\end{frame}

```

Ci sono altri comandi simili a `\onslide` come, ad esempio, `\only`, `\uncover`, `\visible`, `\alt`. È opportuno notare che beamer può trattare il testo non visibile in due modi diversi: o come testo invisibile (che occupa spazio e può eventualmente avere effetti di trasparenza) o come testo che viene inserito solo nelle slides specificate e viene “gettato via” nelle altre: queste diverse modalità possono essere facilmente causa di frustrazione qualora non vengano ben comprese. Ad esempio il comando `\only` è del secondo tipo mentre il comando `\onslide` di default è del primo tipo, ma può assumere dei modificatori per ottenere risultati diversi.

Un caso particolarmente importante di visualizzazione passo-passo si ha con gli elenchi: in tale caso, oltre a definire le slides su cui un item deve apparire dandone la specificazione direttamente con il comando `\item<...>`, si può definire direttamente l’avanzamento passo-passo nell’ambiente `itemize` (o `enumerate`) mediante l’opzione `<+>`:

```

\begin{itemize}[<+>]
  \item Questo appare dalla prima slide.
  \item questo appare dalla seconda slide .
  \item<1-> Anche questo appare dalla prima slide essendo
    specificato nell'item
  \item questo appare dalla terza slide
\end{itemize}

```

La classe beamer ha a disposizione comandi ed ambienti per evidenziare parti di testo.

I due comandi principali sono `\alert<...>{testo}` e `\structure<...>{testo}` che modificano grandezza e/o colore del testo così da evidenziarlo rispetto a quello circostante. Entrambi questi comandi possono essere usati globalmente all’interno di un ambiente `itemize` o `enumerate` come si vede dal seguente esempio:

```

\begin{itemize}[<+>| alert@+>]
  \item 1
  \item 2
  \item 3
\end{itemize}

```

In questo esempio quando un item appare, viene evidenziato in rosso.

```

\begin{itemize}
  \item<2->\alert<2> 1
  \item<2->\alert<3> that has
  \item<2->\alert<4> beginning
\end{itemize}

```

In questo esempio tutti gli item appaiono sulla seconda slide e vengono evidenziati uno dopo l'altro.

Un altro modo per sottolineare alcuni concetti della presentazione è usare dei box colorati. Alcuni ambienti come ad esempio `theorem`, `definition`, `example` costruiscono attorno al testo una “scatola” colorata (i primi due con il colore assegnato agli oggetti “structure”, l'altro in verde), ma è possibile creare box diversi sia con il colore “structure” sia con il colore “alert” (generalmente rosso) mediante i comandi:

```

\begin{block}
  {Titolo}
  testo
\end{block}
\begin{alertblock}
  {Titolo}
  testo
\end{alertblock}

```

È possibile inoltre creare box con i colori che si desiderano con gli ambienti `beamercolorbox` e `beamerboxesrounded`.

```

\setbeamercolor{postit}{fg=black,bg=yellow}
\begin{beamercolorbox}[sep=1em,wd=5cm]{postit}
  Appiccicami da qualche parte!
\end{beamercolorbox}

\setbeamercolor{uppercol}{fg=white,bg=ugreen}
\setbeamercolor{lowercol}{fg=black,bg=lgreen}
\begin{beamerboxesrounded}[upper=uppercol,lower=lowercol,shadow=true]{Theorem}
  $A = B$.
\end{beamerboxesrounded}}

```

### 11.1.3 Tabelle, figure, suoni ed animazioni

Tutti gli effetti prima visti possono essere applicati anche a figure e tabelle, per cui è possibile far apparire una riga alla volta di una tabella o solo alcune parti di una figura.

Una possibilità molto interessante offerta da beamer è quella di ingrandire parti di una figura (effetto zoom) con il comando `\framezoom`.

È inoltre possibile inserire all'interno della presentazione file multimediali sia audio che video (i file stessi non sono inseriti nel file `pdf` che contiene solo un collegamento: quindi è necessario avere i file multimediali insieme al file della presentazione): a seconda della configurazione del computer e del visualizzatore sarà possibile mostrare il file all'interno della presentazione o con un visualizzatore esterno.

Il comando `\animate<...>` fornisce una visualizzazione automatica passo-passo. Tale visualizzazione può essere modificata con il comando `\animatevalue<...>` per il cui uso si rimanda al manuale di beamer. Un altro metodo per ottenere un'animazione è utilizzare uno dei comandi `\hypersetup{pdfpageduration=t}` o `\transduration<...>{t}` dove `t` è la durata in secondi e, nel secondo, `<...>` indica il gruppo di slide a cui ci si riferisce.

## 12 La grafica

Per poter lavorare con le immagini, come abbiamo già visto nel Paragrafo 2.2.5, è innanzitutto necessario occorre caricare il pacchetto `graphicx` che permette di utilizzare file grafici all'interno di  $\text{\LaTeX}$ . Se poi si vuole creare le immagini all'interno del file è utile caricare il pacchetto `xcolor` che consente una gestione avanzata dei colori, non solo in ambiente grafico.

Come abbiamo detto è possibile generare figure con  $\text{\LaTeX}$ , sia all'interno del testo che in file separati che possono poi essere inseriti. Ci sono diversi pacchetti per questo scopo e noi ci occuperemo di due di questi che sono i più potenti e i più usati: `PSTricks` e `tikz`.

Iniziamo col dire che si tratta di pacchetti molto grossi con manuali estremamente lunghi (`PSTricks` più di 300 pagine, `tikz` addirittura 1300) e che, inoltre, esistono moltissimi pacchetti complementari (oltre un centinaio) per compiti specifici, ognuno con il proprio manuale. Di conseguenza noi accenneremo soltanto al loro utilizzo.

### 12.1 PSTricks

`PSTricks` è una collezione di macro di interfaccia tra  $\text{\TeX}$  (e  $\text{\LaTeX}$ ) e il linguaggio PostScript: quindi l'immagine che viene creata è di tipo PostScript e *non* viene interpretata nativamente da `pdf $\text{\LaTeX}$` . Per questo motivo è necessario o utilizzare un pacchetto apposito (`pst-pdf`) o generare separatamente i file grafici, convertirli



in pdf e poi importarli, oppure ancora utilizzare Lua $\text{\LaTeX}$  o Xe $\text{\LaTeX}$  che effettuano automaticamente la conversione<sup>15</sup>.

La maggiore difficoltà di compilazione è compensata dal fatto che il linguaggio PostScript è il linguaggio nativo delle stampanti e quindi è praticamente “onnipotente” nella resa grafica.

*PSTricks* consente di disegnare tutti i tipi di oggetti, grafici 2D e 3D, a colori e non, e molto altro.

Usando *PSTricks*, un qualsiasi disegno viene “tracciato” attraverso una serie di comandi.

Ad esempio, se si vuole disegnare una circonferenza con centro nell’origine degli assi e raggio 3, si userà il comando

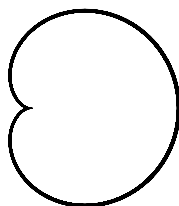
```
\pscircle(0,0){3}
```

Ancora, il grafico di una cardioide di equazione, in coordinate polari,  $\rho(\theta) = 1 + \cos(\theta)$  può essere disegnato attraverso il comando

---

```
\parametricplot[algebraic,plotpoints=1500]{0}{360}{0.5+cos(t)+0.5*cos(2*t)|sin(t)+0.5*sin(2*t)}
```

---



che rappresenta le equazioni cartesiane parametriche della cardioide  $(0.5 + \cos(t) + 0.5 * \cos(2 * t))$  è l’espressione dell’ascissa del punto corrente sulla curva e  $\sin(t) + 0.5 * \sin(2 * t)$  è l’espressione dell’ordinata).

Oltre al pacchetto base, *PSTricks*, esistono numerosissimi altri pacchetti che ne estendono le funzionalità.

Attualmente, il pacchetto più importante è, a nostro avviso, *pstricks-add*, che estende le potenzialità di *PSTricks* e ne facilita l’uso in molti ambiti. Ad esempio, consente di sovrapporre colori e immagini, permette tracciare in modo più immediato i grafici delle funzioni e di scriverne l’espressione in notazione algebrica (quella a cui siamo abituati) anziché in notazione inversa polacca (quella standard di *PSTricks*), e molto altro ancora.

Per un elenco completo dei pacchetti disponibili e delle loro funzionalità si può far riferimento al sito internet: [www.PSTricks.de](http://www.PSTricks.de)

---

<sup>15</sup>C’è anche un’altra possibilità utilizzando il compilatore  $\text{\LaTeX}$ , convertendo il file dvi ottenuto in Postscript e infine convertendo il file ps in pdf.

### 12.1.1 Concetti introduttivi

Quasi tutti i comandi di PSTricks utilizzano la seguente simbologia per i delimitatori:

- ✓ *le parentesi graffe*: racchiudono gli argomenti obbligatori;
- ✓ *le parentesi quadre*: racchiudono gli argomenti facoltativi;
- ✓ *le parentesi tonde*: racchiudono le coordinate dei punti, separate tra loro da virgole;
- ✓ *il simbolo di uguale (=)*: viene usato per indicare che un certo parametro deve assumere un dato valore; i valori attribuiti ai diversi parametri sono separati da virgole.

Dato che le virgole sono usate come separatori delle opzioni del comando, si devono indicare le cifre decimali con il punto.

I valori di default della maggior parte dei parametri grafici possono essere cambiati

- ✓ *globalmente, da un certo punto in poi*, con il comando:

---

```
\psset{parametro1 = valore1, parametro2 = valore2, ...}
```

---

Ad esempio, se vogliamo cambiare lo spessore di una linea, useremo il comando:

---

```
\psset{linewidth=0.2cm}
```

---

che consentirà di disegnare (con un comando apposito) una linea di spessore 0.2 cm.

Nello stesso modo potranno essere cambiati anche i valori di tutti gli altri parametri grafici.

- ✓ *localmente*, come opzioni di un comando (tra parentesi quadre).

### 12.1.2 I colori

In PSTricks, ci sono diversi modi per specificare i colori.

I colori di base (red, yellow, blue, ecc.) vengono riconosciuti semplicemente indicandoli con il loro nome inglese.

Poiché PSTricks carica automaticamente il pacchetto `xcolor`, possono essere usati indicandoli con il loro nome tutti i modelli di colori definiti da `xcolor`

### 12.1.3 L'ambiente `pspicture`

Gli oggetti grafici creati con PSTricks devono essere contenuti all'interno dell'ambiente *pspicture*.

Il comando che consente di aprire e chiudere l'ambiente è:

---

```
\begin{pspicture}(a, b)(c, d)
...
\end{pspicture}
```

---

Il comando crea una “scatola” in un piano cartesiano immaginario con vertice inferiore sinistro in  $(a, b)$  e vertice superiore destro in  $(c, d)$ : gli oggetti verranno sistemati nella scatola partendo da tali coordinate.

Il comando appena descritto permette agli oggetti grafici di fuoriuscire dalla scatola.

Se si vuole invece che ciò che esce dalla scatola non venga disegnato si deve usare la versione asteriscata.

---

```
\begin{pspicture*}(a, b)(c, d)
...
\end{pspicture*}
```

---

### 12.1.4 La griglia

Uno strumento estremamente utile, quando si lavora in PSTricks, è la *griglia*, che viene creata con il comando `\psgrid`.

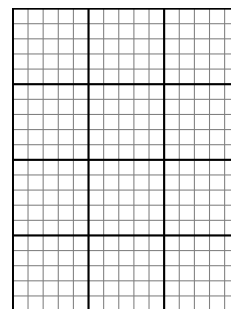
Questo comando crea un rettangolo con un reticolato, che parte dall'angolo inferiore definito dal primo parametro di `\pspicture` e finisce nell'angolo superiore, quello definito dalla seconda coordinata.

Ad esempio:

---

```
\begin{pspicture}(0,0)(3,4)
\psgrid
\end{pspicture}
```

---



La griglia può essere eliminata, nella versione finale del documento, commentandola in ognuno dei grafici, oppure globalmente con il comando `\let\psgrid\relax`.

### 12.1.5 I punti

Il comando `\psdots[opzioni](x1,y1)(x2,y2)\dots(xn,yn)` disegna un punto per ognuna delle coordinate specificate.

Le principali opzioni sono:

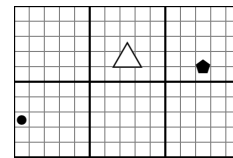
- ✓ *dotstyle=stile*: consente di modificare l'aspetto dei punti (triangoli, cerchi, quadrati, ecc.);
- ✓ *dotsize=misura*: consente di specificare la grandezza assoluta dei punti;

Ad esempio:

---

```
\begin{pspicture}(0,0)(3,2)
  \psgrid
  \psdots (0.1,0.5)
  \psdots[dotstyle=triangle,dotsize=10pt](1.5,1.3)
  \psdots[dotstyle=pentagon*,dotsize=5pt](2.5,1.2)
\end{pspicture}
```

---



### 12.1.6 Le linee

- per i segmenti si usa il comando:

---

```
\psline[opzioni] (coordinate punto iniziale)(coordinate punto finale)
```

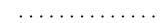
---

Con lo stesso comando è anche possibile creare delle frecce: alla linea si aggiungeranno tra parentesi graffe una linea, preceduta da un `<` e/o seguita da un `>`, per indicare la direzione della freccia.

---

```
\begin{pspicture}(0,0)(2,1)
  \psline[linewidth=2pt]{<->}(0,2)(2,2)
  \psline[linestyle=dashed,linecolor=blue,linewidth=2pt]%(0,1)(2,1)
  \psline[linestyle=dotted] (0,0)(2,0)
  \psline (0,-1)(2,-1)
\end{pspicture}
```

---



- per le spezzate, si usa il medesimo comando, solo che la linea viene fatta passare per più di due punti:

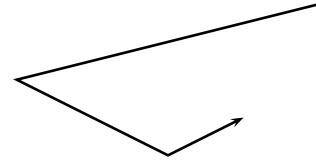
---

```

\begin{pspicture}(0,0)(4,2)
  \psline[linewidth=1pt]{->}(4,2)(0,1)(2,0)(3,.5)
\end{pspicture}

```

---



- per i poligoni, si usa il comando `\pspolygon[opzioni](x0,y0)(x1,y1)(x2,y2) \dots (xn,yn)`, che consente di tracciare delle linee (anche intrecciate) che, a differenza di quelle tracciate con il comando `\psline`, vengono chiuse automaticamente.

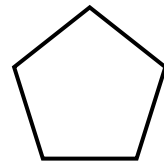
---

```

\begin{pspicture}(-0.3,-0.3)(2.3,2.3)
  \pspolygon[linewidth=1.5pt](0,1.21)(0.38,0)(1.62,0)(2,1.21)(1,2)
\end{pspicture}

```

---



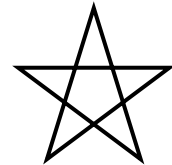

---

```

\begin{pspicture}(-0.3,-0.3)(2.3,2.3)
  \pspolygon[linewidth=1.5pt](0,1.21)(2,1.21)(0.38,0)(1,2)(1.62,0)
\end{pspicture}

```

---



### 12.1.7 Stili e riempimenti

È possibile, come abbiamo visto nel paragrafo 12.1.6, cambiare lo stile, lo spessore e il colore delle linee:

1. **L'opzione *linestyle*:** permette di cambiare lo stile delle linee (che di default sono continue). Alcuni degli stili disponibili, sono la linea tratteggiata (*linestyle=dashed*), la linea a puntini (*linestyle=dotted*) e la linea "invisibile" (*linestyle=none*), la cui importanza sarà illustrata più avanti.
2. **L'opzione *linecolor*:** consente di cambiare il colore di una linea (che di default è nero).
3. **L'opzione *linewidth*:** consente di gestire lo spessore della linea che stiamo disegnando, che di default è di 0.8 punti.

Vediamo ora le opzioni per modificare il riempimento di una figura:

1. **L'opzione *fillcolor*:** serve per determinare il colore con cui viene riempito l'interno di una figura.  
Il comando funziona solo se viene anche specificato uno stile di riempimento, mediante l'opzione *fillstyle* che vediamo qui di seguito (il valore di default di questo secondo comando, *none*, fa sì che l'interno della figura non venga riempito).

2. **L'opzione *fillstyle*:** determina lo stile con il quale un oggetto viene riempito.

Ad esempio:

---

```
\begin{pspicture}(3,3)
  \psframe[fillstyle=solid,fillcolor=red](0,1)(3,2)
\end{pspicture}
```

---



Gli stili di riempimento accettati sono:

- ✓ *none*: è lo stile di default e determina il non-riempimento della figura;
- ✓ *solid*: l'oggetto viene riempito in modo uniforme nel colore determinato da `fillcolor` (di default bianco);
- ✓ *vlines*, *vlines\**, *hlines*, *hlines\**, *crosshatch*, *crosshatch\**: riempie la figura di linee rispettivamente verticali, orizzontali o incrociate;
- ✓ *transparent*: consente di riempire l'interno della figura di colore "trasparente" (è possibile scegliere il grado di trasparenza)<sup>16</sup>.

Ad esempio:

---

```
\begin{pspicture}(3,3)
  \psframe[fillstyle=vlines,hatchcolor=red,hatchangle=0](0,1)(3,2)
\end{pspicture}
```

---



### 12.1.8 Le forme geometriche di base

In PSTricks si creano facilmente le forme geometriche di base: vediamo alcuni esempi.

**Rettangoli** Sono creati dal comando `\psframe`.

Ad esempio:

---

```
\begin{pspicture}(0.5,1.8)(3.5,3.3)
  \psframe[linewidth=2pt](0.7,2)(3.3,3)
\end{pspicture}
```

---



**Cerchi** Si ottengono con il comando `\pscircle[opzioni](x0,y0){raggio}`, nel quale  $(x_0, y_0)$  sono le coordinate del centro e *raggio* è (ovviamente) la misura del raggio.

Ad esempio:

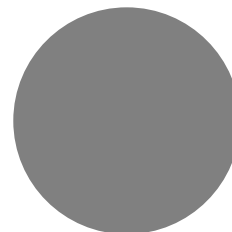
---

<sup>16</sup>Per i dettagli dell'opzione, si veda la documentazione di `pstricks-add`.

---

```
\begin{pspicture}(1,-1)(5,3)
  \psset{linewidth=1pt,linecolor=gray}
  \pscircle*(3,1){1.5}
\end{pspicture}
```

---



**Parabole** Si ottengono mediante il comando `\parabola[opzioni](x0,y0)(x1,y1)`, che disegna una parabola con asse parallelo all'asse delle ordinate, passante per il punto di coordinate  $(x_0, y_0)$  e avente vertice nel punto di coordinate  $(x_1, y_1)$ . Ad esempio:

---

```
\begin{pspicture}(3,3)
  \parabola[linestyle=dashed](0,1)(1,2.5)
\end{pspicture}
```

---



Esistono comandi anche per ellissi, archi, settori, ...  
Si noti che la versione con asterisco riempie l'oggetto del colore della linea.

### 12.1.9 Oggetti matematici

**Le funzioni** Per disegnare i grafici delle funzioni di una variabile si useranno molte delle macro predefinite in `pstricks-add`, che supporremo quindi caricato di default.

`PSTricks` accetta, di base, solo funzioni scritte in *RPN* (*Reverse Polish Notation*). Uno dei vantaggi di `pstricks-add` è quello di poter usare, solo per funzioni di una variabile, la notazione algebrica usuale, mediante il comando `\psset{algebraic=true}`.

**Gli assi** La macro che consente di creare gli assi è:  
`\psaxes[opzioni]{frecce}(x0,y0)(x1,y1)(x2,y2)`, dove  $(x_1, y_1)$  e  $(x_2, y_2)$  sono gli angoli opposti del rettangolo che delimita gli assi;  $(x_0, y_0)$  sono le coordinate del punto di intersezione degli assi (l'origine).

È possibile modificare l'unità di misura di uno o entrambi gli assi e anche usare la scala logaritmica.

**Funzioni 2D** Il comando che ci consente di disegnare i grafici delle funzioni è:

---

```
\psplot{ascissa iniziale}{ascissa finale}{funzione}
```

---

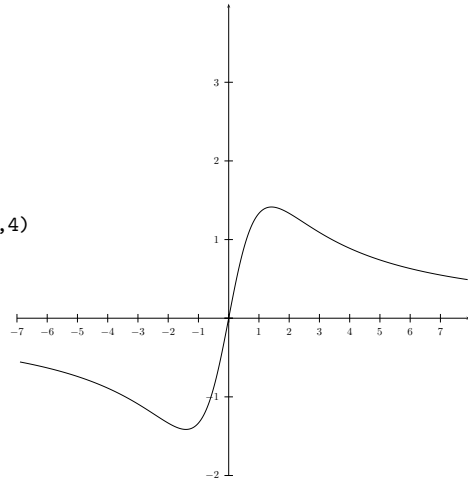
Vediamo un esempio: vogliamo disegnare il grafico della funzione:

$$y = \frac{4x}{x^2 + 1}$$

---

```
\psset{yunit=2.6cm,xunit=1cm}
\begin{pspicture}(-7,-2)(8,4)
\psaxes[labels=all,ticks=all,Dx=1,Dy=1]{->}(0,0)(-7,-2)(8,4)
\psset{algebraic=true,plotpoints=400}
\psplot{-6.9}{7.9}{(4*x)/(2+x^2)}
\end{pspicture}
```

---



È possibile tracciare i grafici di funzioni in coordinate polari, di curve parametriche (abbiamo visto l'esempio della cardioide), di funzioni di 2 variabili.

### 12.1.10 I nodi

Un *nodo* è una forma (di solito un cerchio o un rettangolo) che può contenere un testo al proprio interno: le funzioni sono di aggiungere testo al disegno e di identificare un punto.

I nodi possono essere facilmente creati in PSTricks caricando il pacchetto **pst-node**. I principali comandi per creare i nodi sono:

- ✓ `\rnode[punto di riferimento]{nome}{testo}`: crea un nodo etichettato con *nome*, per il quale appare quanto scritto in *testo*; i connettori puntano al punto di riferimento indicato nella prima parentesi quadra;
- ✓ `\pnode(x,y){nome}`: crea un nodo nel punto di coordinate  $(x,y)$  senza occupare spazio;
- ✓ `\cnode[opzioni](x,y){raggio}{nome}`: crea un nodo circolare, di raggio fissato dal valore nella prima parentesi graffa, nel punto di coordinate  $(x,y)$  (la versione asteriscata crea un cerchio pieno, quella senza asterisco vuoto).

I nodi possono essere collegati tra loro da linee di vario tipo.

Il comando principale,

`\ncline[opzioni]{frecce}{nodo1}{nodo2}`, permette di unire due nodi con una

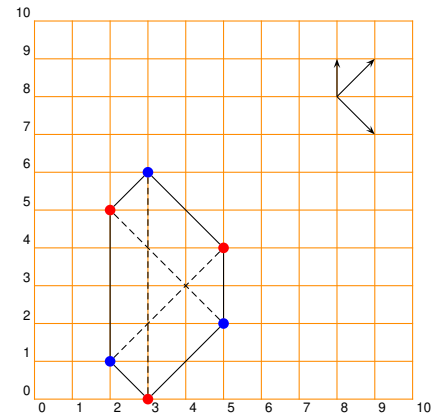


linea (il cui aspetto è settato nelle opzioni), semplicemente richiamando i nodi da unire con le etichette prima attribuite.

Vediamo un esempio:

---

```
\begin{pspicture}(0,0)(10,10)
\psgrid[subgriddiv=1,gridcolor=DarkOrange]
\cnode*[linecolor=red](3,0){A}
\cnode*[linecolor=red](5,4){B}
\cnode*[linecolor=red](2,5){C}
\cnode*[linecolor=blue](2,1){D}
\cnode*[linecolor=blue](5,2){E}
\cnode*[linecolor=blue](3,6){F}
\ncline{A}{D} \ncline{C}{D}
\ncline{A}{E} \ncline{C}{F}
\ncline{E}{B} \ncline{B}{F}
\ncline[linestyle=dashed]{B}{D}
\ncline[linestyle=dashed]{C}{E}
\ncline[linestyle=dashed]{A}{F}
\psline[arrowsize=5pt]{->}(8,8)(8,9)
\psline[arrowsize=5pt]{->}(8,8)(9,9)
\psline[arrowsize=5pt]{->}(8,8)(9,7)
\end{pspicture}
```




---

$\text{\Huge}\{\left[\left\{0,4,5\right\}\overset{2}{=}\left\{1,2,6\right\}\right]\}$   $\{0,4,5\} \stackrel{2}{=} \{1,2,6\}$

---

### 12.1.11 Altri effetti

Con PSTricks è possibile lavorare sul testo, trasformando scritte (e anche espressioni matematiche) in diversi modi.

Esistono inoltre numerosi altri pacchetti per la grafica 3d, per i circuiti, per i frattali, ecc., per un elenco completo dei quali si rimanda al già citato sito <https://tug.org/PSTricks>.

## 12.2 TikZ e PGF

Il pacchetto PGF e la sua interfaccia (non grafica) TikZ permettono di costruire figure mediante comandi simili a quelli di altri programmi o pacchetti, e di avere come output sia un file PostScript, sia un file PDF, senza l'intervento di programmi esterni, ma con l'usuale compilazione con  $\text{\LaTeX}$  o  $\text{\PDF\LaTeX}$ . Il fatto di essere compatibile con  $\text{\PDF\LaTeX}$  ha però l'inconveniente di una minore potenza del pacchetto rispetto a PSTricks in quanto il linguaggio PDF è meno potente (soprattutto da un punto di vista matematico) del linguaggio PostScript.

Per usare il pacchetto tikz occorre caricarlo con il comando `\usepackage{tikz}` oppure caricare il pacchetto pgf. Si possono poi, per particolari scopi, caricare altre librerie specializzate di pgf ad esempio per disegnare diversi tipi di frecce o per usare sfondi, ecc.

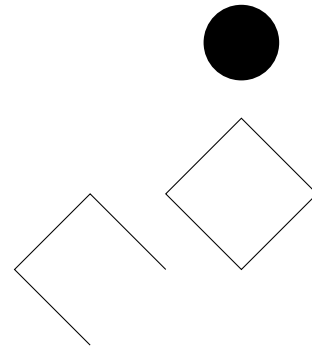
Si tenga comunque presente che tutte le operazioni grafiche vengono in realtà eseguite dalla libreria pgf e che TikZ è solo un'interfaccia che possiede comandi

semplificati per l'uso della libreria (i cui comandi sono alquanto involuti). I comandi che servono per descrivere il grafico vengono inseriti nell'ambiente `tikzpicture`. Il comando fondamentale è `\path` che descrive un oggetto (segmento, cerchio, rettangolo, ecc.). Quando a questo comando, in un qualsiasi punto, viene data l'opzione `[draw]`, l'oggetto viene disegnato; con l'opzione `[fill]`, l'oggetto viene riempito; con l'opzione `[fill,draw]`, l'oggetto viene disegnato e riempito. È possibile scrivere direttamente `\draw`, `\fill`, `\filldraw`, anziché dare opzioni al comando `\path`.

---

```
\begin{tikzpicture}
\path(1,0)--(0,1)--(-1,0)--(0,-1) [draw];
\draw(3,1)--(2,2)--(1,1)--(2,0)--cycle;
\fill (2,3) circle (5mm);
\end{tikzpicture}
```

---



Si noti che i comandi sono terminati dal segno di punto e virgola, i punti sono indicati dalle loro coordinate cartesiane (è però possibile cambiare sistema di coordinate) e l'unità di misura, se non diversamente specificato, è il centimetro.

Oltre a quelle menzionate esistono altre opzioni, ovvero possibilità di disegno.

Il comando, in generale, è `\path <specificazioni>`, dove le specificazioni seguono una sintassi ben definita.

Le operazioni più semplici da eseguire sono il movimento e il disegno di un segmento (o di una spezzata):

---

```
\begin{tikzpicture}
\draw (0,0)--(1,0) (0,-1)--(1,-1)--(-.5,-.5);
\end{tikzpicture}
```

---



Il comando `\draw (x1,y1)--(x2,y2)` disegna un segmento (una spezzata se si indicano più di due punti), con  $(x1,y1)$   $(x2,y2)$  indichiamo che la “penna” deve spostarsi dal punto  $(x1,y1)$  al punto  $(x2,y2)$  senza tracciare nulla.

Due punti possono essere uniti non solo con un segmento, ma anche con una coppia di segmenti uno orizzontale e uno verticale (o viceversa) o mediante linee “bufe”, ad esempio a zigzag, a molla, ecc., (l'autore chiama tali tipi di linea *snakes*); per disegnare tali linee è necessario nel preambolo caricare la libreria apposita con il comando `\usepackage{pgflibrarysnakes}`.

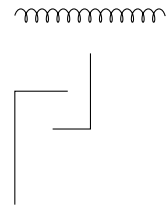
---

```

\begin{tikzpicture}
\draw (0.5,0)-|(1,1) (0,-1)|- (.7,.5);
\draw [snake=coil,segment length=4pt] (0,1.5)--(2,1.5);
\end{tikzpicture}

```

---



Aggiungendo `--cycle` al termine dei punti di una spezzata, si ottiene una poligonale: il segmento tracciato chiude la poligonale meglio rispetto a quanto viene fatto indicando manualmente il punto finale (si noti l'opzione globale per avere una linea di larghezza maggiore):

---

```

\begin{tikzpicture} [line width=10pt]
\draw [color=blue] (0,0)--(1,1)--(1,0)--(0,0);
\draw (2,0)--(3,1)--(3,0)-- cycle;
\end{tikzpicture}

```

---



Nell'esempio precedente si può anche vedere come è possibile dare a un oggetto un colore diverso da quello di default. In generale le opzioni vengono date come in  $\text{\LaTeX}$  tra parentesi quadre.

In  $\text{\TikZ}$  è facile disegnare in un ambiente 3d: in effetti è sufficiente usare tre coordinate per farlo o specificare che si lavora in un sistema di coordinate  $xyz$ . Di default le coordinate  $x$  e  $y$  sono come quelle del piano, mentre il punto  $(0,0,1)$  viene rappresentato assonometricamente con il punto di coordinate  $(-\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}})$ . Nel seguente esempio il primo e il terzo punto sono definiti implicitamente e il secondo esplicitamente (nell'esempio ci sono anche comandi per inserire testo che vedremo più avanti).

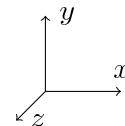
---

```

\begin{tikzpicture}[->]
\draw (0,0,0)--(1,0,0) node[above=2pt]{$x$};
\draw (0,0,0)--(xyz cs:y=1) node[right=2pt]{$y$};
\draw (0,0,0)--(0,0,1) node[right=2pt]{$z$};
\end{tikzpicture}

```

---



### 12.2.1 Rettangoli, cerchi, ellissi, parabole

Rettangoli, cerchi, ellissi, parabole hanno comandi specifici per essere disegnati. La sintassi è:

- Per disegnare un rettangolo con vertici opposti in  $(x_0, y_0)$  e  $(x_1, y_1)$

---

```

\path (x0,y0) rectangle (x1,y1);

```

---

- Per disegnare un cerchio

---

```
\path (x0,y0) circle (raggio);
```

---

dove, se specificato,  $(x0, y0)$  è il centro mentre se non è specificato il centro è il punto corrente. Il punto corrente alla fine del comando è ancora il centro.

- Per disegnare un ellisse

---

```
\path (x0,y0) ellipse (a and b);
```

---

dove  $(x0, y0)$  è il centro,  $a$  il semiasse nella direzione dell'asse  $x$  e  $b$  il semiasse nella direzione dell'asse  $y$ .

- Per disegnare una parabola abbiamo i seguenti comandi

---

```
\path (x0,y0) parabola (x1,y1);
\path (x0,y0) parabola bend (x2,y2) (x1,y1);
```

---

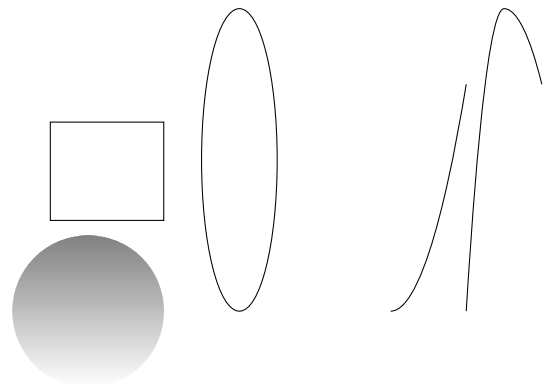
Entrambi i comandi tracciano un arco di parabola passante per il punto corrente  $(x0, y0)$  (che come sempre può non essere indicato) e il punto  $(x1, y1)$ , ma nel secondo caso si impone che il vertice sia in  $(x2, y2)$  mentre nel primo il vertice è nel punto corrente.

Nel seguente esempio si noti il comando `\shade` che riempie l'oggetto con un gradiente di colore.

---

```
\begin{tikzpicture}
\draw (-.5,1.2) rectangle (1,2.5);
\shade (0,0) circle (1 cm);
\draw (2,2) ellipse (.5 and 2);
\draw (4,0) parabola (5,3);
\draw (5,0) parabola bend (5.5,4)
(6,3);
\end{tikzpicture}
```

---



### 12.2.2 Grafici di funzioni

Questo è uno degli aspetti in cui maggiormente si avverte la differenza tra **pgf** e **PSTricks** a causa della minore potenza matematica del linguaggio PDF.

Cominciamo con il comando

---

```
\draw plot[opzioni] coordinates{(x0,y0) (x1,y1) ... (xn,yn)};
```

---

che disegna una curva che unisce i punti indicati: la curva è una spezzata, ma con l'opzione **smooth** viene “lisciata”.

È anche possibile leggere le coordinate da un file esterno, per cui se si ha una tabulazione di molti punti si può tracciare il grafico in questo modo.

Il comando per disegnare il grafico di una funzione è

---

```
\draw plot[opzioni]{espressione con coordinate}
```

---

dove l'espressione con coordinate contiene una macro speciale (la variabile, che di default è **\x**); l'espressione viene valutata per diversi valori della variabile e la tabella di coordinate risultante viene disegnata.

Tra le opzioni segnaliamo **smooth** che fa sì che i punti del grafico siano connessi usando una curva liscia, mentre di default i punti sono connessi da segmenti (il meccanismo usato per “lisciare” la curva è ben lungi dall'essere perfetto a detta dello stesso autore).

Le opzioni sono, oltre alle solite su spessore, colore, ecc.,

- **variable= macro** (la variabile),
- **samples= numero di punti** (il numero di punti in cui l'espressione viene valutata, di default 25),
- **domain= inizio:fine** (il dominio in cui i punti vengono presi, di default -5:5),
- **samples at={lista di numeri}** (fornisce esplicitamente in quali punti deve essere valutata l'espressione, tale opzione annulla eventuali dichiarazioni di **samples** o **\domain**).

La simbologia matematica è quella usuale, le funzioni seno e coseno sono calcolate in gradi, per calcolarle in radianti si deve usare esplicitamente l'unità **r**,  $\pi$  si indica con **pi** mentre “e” deve essere calcolato con **exp(1)**; se si usano parentesi (solo tonde!) è necessario racchiudere l'espressione tra parentesi graffe.

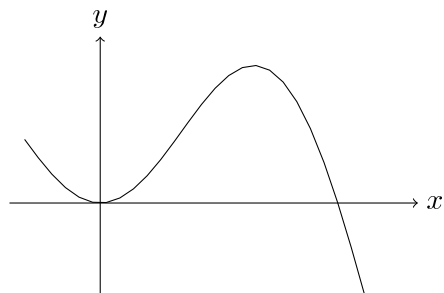
---

```

\begin{tikzpicture}[domain=-1:3.5]
\draw[->](-1.2,0)--(4.2,0)node[right]{$x$};
\draw[->](0,-1.2)--(0,4.2)node[above]{$y$};
\draw plot (\x,{\x*sin(\x r)});
\end{tikzpicture}

```

---



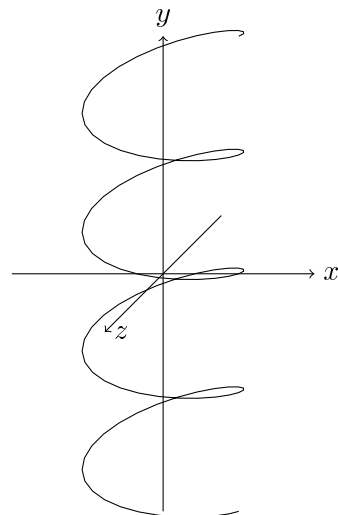

---

```

\begin{tikzpicture}
\draw[->](-2,0,0)--(2,0,0)node[right]{$x$};
\draw[->](0,-pi,0)--(0,pi,0)node[above]{$y$};
\draw[->](0,0,-2)--(0,0,2)node[right]{$z$};
\draw plot [domain=-pi:pi,variable=\t,samples=100]
({cos(4*\t r)},\t,{sin(4*\t r)});
\end{tikzpicture}

```

---



**PGFplots** Per disegnare ogni tipo di grafico è più opportuno utilizzare il pacchetto `pgfplots` costruito sopra la libreria `pgf` e quindi pienamente compatibile con `TikZ`.

### 12.2.3 Nodi e testo

Come si è visto in alcuni esempi precedenti è possibile inserire del testo mediante la sintassi `node[opzioni]{testo}`.

In realtà il comando `node` è un po' più generale e permette di dare a un punto un nome (opzionale) per referenze future e inserire del testo in una figura (cerchio o rettangolo). Le sintassi possibili sono:

---

```

\path ... node[opzioni](nome){testo}
node[opzioni](nome) at (coordinate){testo}

```

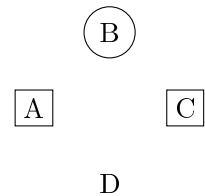
---

La seconda opzione è possibile solo al di fuori dell'ambiente `path` e permette di inserire il nodo alle coordinate specificate; in un ambiente `path` il nodo sarà inserito al punto corrente.

---

```
\begin{tikzpicture}
\draw (0,0)node[rectangle,draw]{A}(1,1)node[circle,draw]{B}
(2,0)node[draw]{C} (1,-1)node{D};
\end{tikzpicture}
```

---



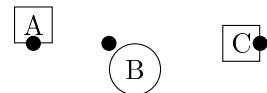
Si noti nel precedente esempio che se non vengono date specificazioni sulla forma il testo è inserito in un rettangolo e, se non viene data l'opzione `draw` il nodo contiene solo il testo.

Tra le opzioni è possibile dare anche opzioni per il testo (grandezza, colore, stile, giustificazione, trasparenza<sup>17</sup> ecc.) e, soprattutto, è possibile dire dove deve essere posto il nodo rispetto alle coordinate del punto: infatti di default il nodo ha il suo centro nel punto in cui è posto, ma spesso è preferibile che sia posizionato in modo diverso. Ciò può essere ottenuto con le opzioni `above`, `below`, `left` e `right` e le loro combinazioni a due a due; a ogni specificazione può essere aggiunta una misura come `above=3mm`, se la misura non viene data il nodo viene traslato così che il bordo sia sulle coordinate specificate.

---

```
\begin{tikzpicture}
\fill (0,1) circle (1mm);
\draw (0,1)node[rectangle,draw,above]{A};
\fill (1,1) circle (1mm);
\draw (1,1)node[circle,draw,below right=1mm]{B};
\fill (3,1) circle (1mm);
\draw (3,1)node[draw,left]{C};
\end{tikzpicture}
```

---

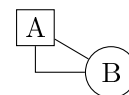


Se a un nodo viene dato un nome è possibile successivamente usare quel nome per indicare il punto.

---

```
\begin{tikzpicture}
\draw (0,1)node(A)[draw,above]{A}(1,1)
node(B)[circle,draw,below]{B};
\draw (A)--(B);
\draw (B)-|(A);
\end{tikzpicture}
```

---



Caricando la libreria `shapes` con il comando `\usetikzlibrary{shapes}` è possibile usare altre forme per i nodi come stelle o poligoni regolari o ellissi; è inoltre

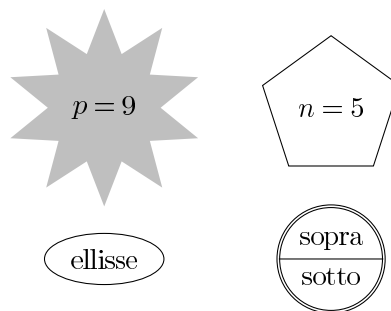
<sup>17</sup>Si ricordi che essendo l'output un file PDF è possibile avere effetti di trasparenza.

possibile suddividere la regione interna di un nodo in parti in modo da avere del testo in ognuna di esse.

---

```
\begin{tikzpicture}
\draw (0,0) node [star,star points=10,fill=lightgray] {$p=9$}
(3,0) node [regular polygon,regular polygon sides=5,draw] {$n=5$}
(0,-2) node [ellipse,draw] {ellisse}
(3,-2) node [circle split,draw,double]{sopra\nodepart{lower}{sotto}};
\end{tikzpicture}
```

---



#### 12.2.4 E inoltre ...

Usando `pgf` è possibile ottenere molto altro: al proposito, oltre alla lettura del manuale, è utile visitare i siti <https://texample.net/> e <https://tikz.net/> che contengono molte immagini ottenute con `TikZ` e `pgf`. Purtroppo per ottenere alcuni risultati grafici è necessario usare codice dalla libreria `pgf` e non la sintassi più semplice offerta da `TikZ`.

## 13 Per saperne di più

Ci sono migliaia di guide per l'utilizzo di  $\text{\LaTeX}$  a qualsiasi livello e (quasi) ogni pacchetto ha la sua documentazione. Su *Overleaf* sono presenti guide ed esempi per i principali pacchetti.

Consigliamo di leggere e avere sempre a disposizione le guide presenti sul sito del GUIT (Gruppo Italiano Utilizzatori  $\text{\TeX}$ ) e, in particolare:

- **L'Arte di scrivere con  $\text{\LaTeX}$**  di Lorenzo Pantieri e Tommaso Gordini, un'introduzione a  $\text{\LaTeX}$  rivolta sia ai principianti sia a utenti più avanzati
- **$\text{\LaTeX}$  per l'impaziente** di Lorenzo Pantieri, è una versione ridotta dell'Arte
- **Introduzione all'arte della composizione tipografica con  $\text{\LaTeX}$** , di Claudio Beccari, è una guida più avanzata a  $\text{\LaTeX}$  e alla tipografia digitale.



- [L<sup>A</sup>T<sub>E</sub>X](#)pedia, di Lorenzo Pantieri, presenta L<sup>A</sup>T<sub>E</sub>X in modo più approfondito

Sempre sullo stesso sito sono a disposizione guide tematiche e molti numeri della rivista ArsTeXnica con altre guide ed esempi.

Un sito interessante sia per L<sup>A</sup>T<sub>E</sub>X che per avere suggerimenti sulla scrittura scientifica, ma non solo, è <https://minerva.ic.cnr.it/course/view.php?id=1091&lang=it>

Per risolvere problemi, capire come si può fare qualcosa e tanto altro ci sono due forum a disposizione: il forum del GUIT <https://www.guitex.org/home/it/forum/index> e <https://tex.stackexchange.com/> (in inglese).