

TEX Tutorial

Paolo Baldi
(con piccole modifiche di G. Benfatto)

Marzo 2009

TEX

TEX è un software per la preparazione di documenti, elaborato da Donald Knuth alla fine degli anni '70. Esso è caratterizzato da

- una elevata qualità tipografica del risultato
- una grande capacità di produrre formule, anche complesse, con grande precisione ed in maniera relativamente semplice e rapida

Non si tratta però di un sistema in cui si vede immediatamente quello che si sta scrivendo, a differenza di altri software sul mercato. Altre proprietà di TEX diventeranno chiare man mano che impareremo ad usarlo.

Un sistema TEX è composto da

- un editor (il vostro preferito);
- il compilatore TEX vero e proprio;
- un driver di stampa o di visualizzazione

Per produrre il documento che vi interessa occorrerà

- scrivere con l'editor il testo sorgente (il testo vero e proprio ed i comandi di formattazione); diciamo che il file che avete prodotto si chiama `toto.tex`;
- invocare il compilatore TEX che produrrà, a partire da `toto.tex` un file `toto.dvi`, dove `dvi` sta per *device independent*;
- invocare il driver appropriato per visualizzare il documento, oppure produrre un file postscript (`toto.ps`), oppure pdf (`toto.pdf`), oppure inviarlo direttamente alla stampante.

Questo schema può apparire complicato, ma in realtà tutte queste operazioni *oggi* si effettuano in maniera semplice, dato che ci sono dei sistemi integrati in cui le tre operazioni descritte sopra si effettuano semplicemente clicchettando qua e là, come vedremo.

Il vantaggio di avere decomposto l'operazione di preparazione del documento nelle tre tappe descritte prima sta anche nel fatto che in questo modo è facile portare il sistema da un sistema operativo all'altro. Oggigiorno T_EX è disponibile su quasi tutti i sistemi operativi (Windows, Mac, Linux, tutte le distribuzioni Unix, ...). Esistono ancora dei PC nei quali T_EX gira sotto dos. ... Basta infatti ricompilare il compilatore T_EX, che attualmente è scritto in C e riscrivere i driver.

Ciò significa in particolare che in qualunque parte del mondo, se avete con voi un file .tex che avete scritto sul computer di casa, sarà facile trovare sul posto un sistema T_EX con il quale potrete visualizzare o stampare il vostro documento.

Per volontà dell'autore, D.Knuth, il programma T_EX è *gratuito* ed il suo copyright è proprietà dell'American Mathematical Society.

Dove lo trovo?

L'installazione di T_EX sul proprio PC dipende dal sistema operativo. Sotto Windows in questo momento la distribuzione di riferimento è *miktex*. Per installarla si fa così:

- connettersi con il sito

<http://www.miktex.org>

- scaricare dalla colonna di sinistra la versione “MIKTeX 2.8” ed installarla, seguendo le istruzioni. Vi verrà chiesto se installare una versione ridotta o completa; se pensate di lavorare senza un collegamento ad Internet, è meglio scegliere quella completa, a meno che non abbiate poco spazio sul disco rigido. Infatti, la versione ridotta contiene il minimo indispensabile e richiede talora il download dei file mancanti, la prima volta che servono.

- DOPO avere installato T_EX, potete installare Winedt, che è l'editore meglio integrato con i vari programmi necessari per compilare e visualizzare un testo scritto in TeX. Winedt si trova all'indirizzo

<http://www.winedt.com>

Si tratta di un software shareware, utilizzabile gratuitamente solo per un mese. In ogni caso, il pacchetto “MIKTeX 2.8” contiene anche l'editore TeXworks, che è anch'esso integrato con il TeX, senza contare che qualunque editor ASCII (anche il Blocco Note di windows) va bene. Deve comunque essere un editor solo ASCII, meglio evitare Word e Wordstar.

Molto materiale e documentazione su T_EX si trova al sito del Tex Users Group:

www.tug.org

compresi molti manuali free in formato pdf. Buona fortuna...

LA STRUTTURA DI UN FILE `.tex`

Esistono in realtà vari dialetti di $\text{T}_{\text{E}}\text{X}$. Infatti il programma di base fornisce solo un ristretto numero di comandi di formattazione. Per potersi servire realmente di $\text{T}_{\text{E}}\text{X}$ occorre combinare questi comandi primitivi per produrre i comandi che in realtà si usano per la realizzazione di un testo. I “dialetti” principali sono:

```
plain
eplain
amstex
latex
```

plain era il dialetto prodotto inizialmente da D. Knuth, molto bello e semplice, non contempla però le possibilità di numerazione automatica, creazione degli indici etc. che vedremo. **amstex** è stato ormai assorbito da **latex**, e **eplain** (il mio preferito...) è ormai pochissimo usato (ingiustamente). Ci occuperemo quindi solo di $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$, che è la versione di $\text{T}_{\text{E}}\text{X}$ di gran lunga più usata. Vedremo che si tratta di un sistema potente e ricco di possibilità, solo un po’ rigido: se vi va bene di fare come vuole lui le cose sono molto semplici, altrimenti occorre faticare un po’. $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ si chiama così perché scritto da un signore che si chiamava Lamport.

Vediamo di produrre il nostro primo documento in $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$. La struttura di un documento $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ è la seguente

```
\documentclass[12pt]{book}
\usepackage{amsmath,amssymb}
%altri comandi
\begin{document}
%il vostro testo
```

`\end{document}`

Osservate che

- Il file comincia sempre con un comando `\documentclass` che definisce la classe del documento. Qui è indicata la classe `book`, ma altre classi sono possibili (`article`, `report`,...). Ogni classe produce un testo con delle formattazioni particolari, adatte all'impiego che se ne intende fare. Vedremo man mano che procediamo le differenze tra una classe e l'altra. In linea generale useremo la classe `book`.

- Tra `\documentclass` e `\begin{document}` c'è quello che si chiama il *preambolo*, in cui si trovano tutti i comandi e le formattazioni, spesso personalizzate, necessarie alla produzione del testo. Per ora il preambolo contiene solo il comando

`\usepackage{amsmath,amssymb}`, che indica che usiamo i package `amsmath` e `amssymb`, che contengono delle definizioni che rendono più facile la produzione di formule matematiche. Man mano che andremo avanti vedremo come aggiungere altro materiale al preambolo. Di solito un utilizzatore usa sempre lo stesso preambolo per tutti i suoi documenti e la prima operazione che fa quando inizia un documento nuovo è di fare un copia-incolla del preambolo da un documento vecchio.

- Il testo vero e proprio comincia con `\begin{document}` termina con `\end{document}`.

- Avrete notato che tutti i comandi T_EX iniziano con `\`. Altre due regole utili da sapere: il nome del comando può contenere solo lettere (niente numeri, niente `@`, `?`,...). Inoltre T_EX ignora tutti gli spazi bianchi dopo un comando.

I TRE MODI

In un documento $\text{T}_\text{E}\text{X}$ il testo può trovarsi in tre modi diversi

- il modo testo
- il modo matematico, per le formule all'interno del testo
- il modo formula, per le formule centrate

Consideriamo ad esempio il testo seguente

Dato il polinomio $P(x) = x^2 - 2x + \frac{3}{4}$, allora

$$\frac{P(x)}{x - \frac{1}{2}} = x - \frac{3}{2}.$$

Cosa si può dire delle radici di P ?

Esso è stato ottenuto con i comandi (tra `\begin{document}` e `\end{document}`)

```
Dato      il
polinomio $P(x)=x^2-2x+\frac{3}{4}$, allora
$$\frac {P(x)}{x-\frac{1}{2}}=
x-\frac{3}{2}  .$$
```

Cosa si pu\`o dire delle radici di P ?

Se confrontiamo il testo prodotto con la sorgente osserviamo che

- Nel modo matematico le formule sono racchiuse tra `$...$`, in quello formula tra `$$...$$`.
- Nel modo testo lasciare uno spazio o più spazi non fa differenza; andare a capo è come lasciare uno spazio: per andare a capo occorre

lasciare una riga vuota (e lasciare una riga vuota o più righe vuote fa lo stesso effetto). Le lettere accentate si ottengono facendole precedere da `\` più l'accento che si vuole: `\'e` → è, `\'e` → é, `\^e` → ê, `\"o` → ö `\~o` → ò. Questa regola vale anche per porre accenti sulle consonanti (non che capiti spesso).

- Nel modo matematico o nel modo formula gli spazi bianchi lasciati nella sorgente non vengono presi in considerazione (vedremo poi come si fa a imporre la presenza di uno spazio quando ce n'è bisogno). Le frazioni si fanno con il costrutto

`\frac{numeratore}{denominatore}`

Da notare che `\frac 32` produce $\frac{3}{2}$ nel modo matematico e $\frac{3}{2}$ nel modo formula. Inoltre se numeratore o denominatore sono composti da più di un carattere, devono essere racchiusi da `{}`. La regola per indici e esponenti è semplice: `x^2` → x^2 , `x_2` → x_2 . Se l'indice o l'esponente sono composti da più di un carattere, devono essere racchiusi da `{}`: `x^32` → x^32 , `x^{32}` → x^{32}

Esercizio 1. Realizzate, compilate e visualizzate il vostro primo testo inserendo il testo precedente tra `\begin{document}` e `\end{document}` nella struttura a pagina 3 (anche facendo copia-incolla). Con MikTeX e lo editor Winedt per compilare e visualizzare si fa così: prima si seleziona la finestra associata al file da compilare (ce ne possono essere molte aperte in Winedt), quindi si clicca in alto al centro \LaTeX . Per vedere il risultato (se non ci sono stati errori) si seleziona infine il simbolo che rappresenta una lente d'ingrandimento con la scritta [DVI](#).

Esercizio 2. Produrre: conciossiacosaché, Ferencváros, Nuñez, È così, però, Föllmer, la compattificazione di Čech.

Esercizio 3. Produrre:

$$\frac{17}{4}, \quad f^{(n)}(x), \quad Y_{lm}(x), \quad H_2O, \quad \Gamma_{ij}^h$$

Nel menu in alto di Winedt (cliccare sul simbolo di sommatoria se non lo avete già aperto) sono segnati i principali comandi per produrre i simboli matematici e gli accenti. Attenzione, gli accenti in modo matematico e in modo testo si fanno in maniera diversa. Provate a cliccare sui simboli di integrale (menù Math) o della lettera *Gamma* (menù Greek).

Esercizio 4. Produrre: $\sqrt{2}$, $\frac{\pi}{2}$, $\frac{a}{b}$, $\frac{ab}{cd}$, $\frac{1}{\sqrt{3}}$.

Esercizio 5. Produrre:

$$\frac{1}{\sqrt{2\pi}} i x e^{i\theta x} e^{-x^2/2} dx$$

Nei prossimi paragrafi ci concentreremo soprattutto sulla realizzazione delle formule. Vedremo poi come gestire il modo testo.

MATRICI

(le tabelle le vedremo più tardi) Le matrici usuali (delimitate da parentesi tonde) si fanno inserendo gli elementi tra

`\begin{pmatrix}... \end{pmatrix}`.

Gli elementi di una riga si separano con `&`, la fine di una riga si codifica con `\\`. Esempio:

`$$`

`\begin{pmatrix}`

`a&\frac{1}{2}&e^{-2} \\`

`1-a&\frac{1}{3}&\frac{\pi}{2} \\`

`a^2&1-\frac{\pi}{2}&1`

`\end{pmatrix}`

`$$`

produce

$$\begin{pmatrix} a & \frac{1}{2} & e^{-2} \\ 1-a & \frac{1}{3} & \frac{\pi}{2} \\ a^2 & 1-\frac{\pi}{2} & 1 \end{pmatrix}$$

`\begin{matrix}... \end{matrix}` avrebbe prodotto la stessa matrice senza parentesi.

Qualche volta occorre mettere in evidenza i numeri delle righe e delle colonne:

\$\$

```
\bordermatrix{\&1&2&3\cr
1&a&\frac{1}{2}e^{-2}\cr
2&1-a&\frac{1}{3}\frac{\pi}{2}\cr
3&a^2&1-\frac{\pi}{2}&1 }
```

\$\$

$$\begin{matrix} & 1 & 2 & 3 \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \left(\begin{array}{ccc} a & \frac{1}{2} & e^{-2} \\ 1-a & \frac{1}{3} & \frac{\pi}{2} \\ a^2 & 1-\frac{\pi}{2} & 1 \end{array} \right) \end{matrix}$$

Esercizio 6. Produrre:

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$$

Esercizio 7. Produrre:

$$\begin{pmatrix} 12 & 17 & 22 \\ 7 & 4 & 15 \\ 0 & 11 & 8 \end{pmatrix}$$

Esercizio 8. Come fareste per produrre i coefficienti di Clebsch-Gordan:

$$\begin{pmatrix} l_1 & l_2 & l_3 \\ m_1 & m_2 & m_3 \end{pmatrix} ?$$

SPAZI

Dato che più spazi bianchi vengono ignorati da $\text{T}_{\text{E}}\text{X}$, quando li si vuole occorre dare il comando adatto.

Intanto per le dimensioni si possono usare le unità di misura usuali.

Le più importanti sono

cm (centimetro)

mm (millimetro)

in (inch=pollice=2.54cm)

em (la lunghezza della lettera m nella font corrente)

ex (l'altezza della lettera X nella font corrente)

pt (punto=0.035146 cm) è una unità di misura tradizionale in tipografia. Per avere una idea, l'interlinea in un testo normale è di 10pt oppure 12pt. Se andate a vedere il comando `\documentclass` all'inizio del vostro file trovate `\documentclass[12pt]{book}`, che impone appunto al vostro testo di avere un'interlinea (distanza tra la base di una riga e quella della successiva) uguale a 12pt. Se volete aumentare l'interlinea, ricordate che la sua dimensione è conservata nella variabile `\baselineskip`. Se scrivete nella sorgente `\baselineskip=24pt` vedrete che passate a doppia spaziatura. Limitiamoci per ora a vedere come gestire gli spazi nel modo matematico, in maniera da potere spaziare i vari elementi che compongono una formula.

Ci limiteremo per ora agli spazi orizzontali. Il comando base è : `\hspace{#pt}` che lascia uno spazio di lunghezza `#pt`. Questo spazio comunque viene ignorato se cade all'inizio o alla fine di una riga o formula. Se si vuole imporre uno spazio anche in questo caso il comando è `\hspace*{#pt}`. Importante: `#` può anche essere un numero negativo, utile per eliminare spazi spuri o per sovrapporre un carattere con quello che lo precede. Esistono comunque degli spazi predefiniti di uso più comodo:

`\quad`: uno spazio di 2em

`\quad`: uno spazio di 1em

`\enspace`: uno spazio di 0.5em

`\;` `\>` `\,`: piccoli spazi, in ordine decrescente di dimensione (la larghezza dello spazio dipende dallo stile corrente).

`\!` piccolo spazio *negativo*.

In una formula è tipograficamente opportuno inserire uno spazio `\,` alla fine di una radice quadrata o dopo una frazione. Ad esempio nella formula

$$\frac{1}{\sqrt{2\pi}}ixe^{i\theta x}e^{-x^2/2}$$

la frazione appare troppo appiccicata al resto. Con lo spazietto in più diventa

$$\frac{1}{\sqrt{2\pi}}ixe^{i\theta x}e^{-x^2/2}$$

È anche consigliabile di inserire uno spazio `\,` per separare l'integrando dal differenziale:

$$\int_0^1 f(x) dx \quad \text{piuttosto che} \quad \int_0^1 f(x)dx$$

Esercizio 9. Come fare per scrivere

$$\frac{1}{\sqrt{2\pi}\sigma}, \quad \int_0^1 f(x) dx, \quad \sin x \frac{e^x - e^{-x}}{2}, \quad \mathbb{R}$$

\mathbb{R} è stato usato per molto tempo per indicare i numeri reali. Ora si possono usare le font AMS: \mathbb{R} . Sono particolarmente brutte, ma non ci sono ricette semplici per fare $\mathbb{C}, \mathbb{Z}, \mathbb{Q}$. Per fare queste lettere bisogna indicare, ad esempio, $\mathbb{\mathbb{R}}$. Se ve lo dimenticate, sta nel menu Math di Winedit, in alto a destra.

GRANDI OPERATORI

Sono quelli che compaiono nel settore in alto a sinistra nella barra "Math" di Winedit.

Esempio (osservare come si fa a fare apparire gli indici sopra e sotto):

\$\$

`\sum_{n=1}^{\infty}\frac{1}{n^4}=\frac{\pi^4}{90}`

\$\$

$$\sum_{n=1}^{\infty} \frac{1}{n^4} = \frac{\pi^4}{90}$$

I grandi operatori non sono soltanto dei caratteri: hanno delle particolarità interessanti:

- Hanno associati degli spazi prima e poi.
- Cambiano di dimensione a seconda che siamo in modo formula o in modo matematico (provate a scrivere la stessa formula in modo matematico e vedete la differenza)
- Indici ed esponenti si possono scrivere sopra e sotto (come nella formula qui sopra) oppure accanto. I comandi `\limits` e `\nolimits` permettono di passare da uno stile all'altro. Provate a vedere la differenza scrivendo

\$\$

`\sum\nolimits_{n=1}^{\infty}\frac{1}{n^4}=\frac{\pi^4}{90}`

`\frac{\pi^4}{90}`

\$\$

I grandi operatori mettono per default gli indici sopra e sotto (per questo non c'era bisogno di precisare `\limits` nel primo esempio della pagina precedente). Fa eccezione il simbolo di integrale `\int`, in cui per default gli indici sono posti accanto. Provate a vedere la differenza con una formula in cui compare `\int` nei due modi:

Esercizio 10. Produrre:

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} ix e^{i\theta x} e^{-x^2/2} dx$$

e

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} ix e^{i\theta x} e^{-x^2/2} dx$$

(che è più brutto)

DELIMITATORI

Si tratta delle parentesi e delle frecce (vedi il menu di `winedit`). Anche questi non sono semplici caratteri. `TEX` associa ad essi degli spazi opportunamente calcolati prima e poi.

Inoltre essi sono *scalabili*: ad esempio

`\uparrow` ↑

`\big\uparrow` ↑

`\Big\uparrow` ↑

`\bigg\uparrow` ↑

`\Bigg\uparrow` ↑

Si possono anche adattare automaticamente alle dimensioni di quello che segue (utile soprattutto per le parentesi): ad esempio

`$$`

`\left(\int_0^{2\pi}\sin^2 x, dx\right)^{1/2}`

`$$`

produce

$$\left(\int_0^{2\pi}\sin^2 x dx\right)^{1/2}$$

Il costrutto `\left ... \right` è molto comodo, spesso però produce delle parentesi troppo grandi:

$$\left(\sum_{i=1}^N \frac{1}{n} \right)^2$$

Meglio in questo caso `\bigg ... \bigg` che dà

$$\left(\sum_{i=1}^N \frac{1}{n} \right)^2$$

Se si usa `\left ... \right` è indispensabile che ci siano sia il `\left` che il `\right`. Se si vuole mettere una parentesi sola, a sinistra ad esempio, si può scrivere `\right.` (puntolino) alla fine.

Esercizio 11. Provate a realizzare

$$\left[\begin{array}{ccc} a_{11} & \dots & \dots \\ a_{21} & \dots & \dots \\ \dots & \dots & \dots \\ a_{n1} & \dots & \dots \end{array} \right.$$

FRAZIONI E SIMILI

Abbiamo visto che le frazioni si fanno con `\frac`. Ad esempio `\frac{n+1}{n}` produce

$$\frac{n+1}{n}$$

Si sarebbe anche potuto scrivere `{n+1 \over n}`. Due varianti: coefficiente binomiale: `{sopra \choose sotto}`

$$\binom{n+1}{n}$$

e senza linea di frazione `{sopra \atop sotto}`:

$$\begin{array}{c} n+1 \\ n \end{array}$$

DIMENSIONE DEI CARATTERI IN MODO MATEMATICO

\TeX , quando è in modo matematico, usa tre dimensioni per lo stesso simbolo

1) normale: n . Usato anche per numeratore e denominatore in modo formula.

2) script: n . Usato per indici e esponenti: a_n, e^n . Anche numeratore e denominatore in modo matematico vanno in dimensione script. E anche frazioni che si trovino al numeratore o denominatore di altre frazioni.

3) scriptscript: n . Usato per gli indici di indici: a_{k_n} ; ma anche per frazioni che si trovano a numeratore o a denominatore in modo matematico o ad esponente. . .

Per un paragone diretto: $n \ n \ n$

Esempio

$$\frac{n + \frac{1}{n}}{n^2}$$

Tutto ciò è automatico. È possibile però forzare la dimensione dei caratteri con i comandi `\displaystyle`, `\textstyle`, `\scriptstyle` e `\scriptscriptstyle`. Ad esempio

La successione di termine generale $\frac{1}{n}$ converge a 0 grazie a. . .

può diventare

La successione di termine generale $\frac{1}{n}$ converge a 0 grazie a. . .

che si ottiene scrivendo `\displaystyle\frac 1n$`. Ma l'effetto non è proprio bello. Più utile talvolta è poter scrivere

$$\frac{1}{2}(1+x^2) \quad \text{piuttosto che} \quad \frac{1}{2}(1+x^2)$$

Qui si scrive `\textstyle\frac 12}`.

Oppure se trovate che in

$$e^{\frac{n}{n+1}}$$

il numeratore e il denominatore all'esponente siano troppo microscopici, potete usare `\textstyle` e ottenere

$$e^{\frac{n}{n+1}} \quad \rightarrow \quad e^{\frac{n}{n+1}}$$

I package AMS definiscono già i comandi `\dfrac` e `\tfrac` che forzano la realizzazione della frazione rispettivamente in modo display ed in modo testo. La frazione di destra qui sopra si può quindi ottenere sia con

`e^{\textstyle {n\over n+1}}`

che con

`e^{\textstyle\frac n{n+1}}`

oppure

`e^{\tfrac n{n+1}}`

In linea di massima comunque conviene evitare di modificare i default tipografici di T_EX.

Equazioni numerate

Le formule centrate tra $...$ non sono numerate. Per dare un numero alle equazioni si usa il costrutto

```
\begin{equation}\label{lab1}
-- testo dell'equazione --
\end{equation}
```

Da notare che

- non c'è bisogno di $...$.
- Non c'è bisogno di indicare il numero dell'equazione. \LaTeX provvederà a numerarle automaticamente in progressione. L'espressione **lab1** permette di richiamare (cross-referencing) il numero dell'equazione con il comando `\ref`. Se, più in là nel testo si vorrà dire “come indicato nell'equazione...”, basterà scrivere

come indicato nell'equazione `\ref{lab1}...`

La stringa di caratteri del label, **lab1** in questo caso, viene scelta di volta in volta da chi scrive, magari cercando qualcosa di facile da ricordare quando ci sarà da fare il richiamo.

ALLINEAMENTI

Ci sono equazioni che si estendono su più righe. In altri casi si vogliono disporre più equazioni una sotto l'altra, magari in modo che siano allineate. Come si può fare?

Intanto bisogna evitare di disporre più costrutti $$$\dots$$$ uno sull'altro. In questo modo rimane troppo spazio tra un'equazione e l'altra e poi è difficile fare allineamenti.

\LaTeX prevede diverse possibilità per risolvere questo problema. Il carattere $\&$ viene usato in \LaTeX per gestire gli allineamenti.

Con `eqnarray`:

```
\begin{eqnarray}
% \nonumber to remove numbering (before each equation)
  x+y&=&3 \label{p1} \\
\nonumber x-y^{2+\alpha}&=&1
\end{eqnarray}
Grazie alla (\ref{p1})\dots
```

$$\begin{array}{rcl} x + y & = & 3 \\ x - y^{2+\alpha} & = & 1 \end{array} \tag{1}$$

Grazie alla (1)...

Da notare il comando `\label`, che permette di richiamare il numero della prima equazione. Questo environment lascia però troppo spazio intorno al segno di uguale.

Si possono anche centrare le equazioni, invece di allineare i segni di =:

```
\begin{eqnarray}
& x+y=3 & \\\
& x-y^{2+\alpha}=1 & \\
\end{eqnarray}
```

$$x + y = 3 \tag{2}$$

$$x - y^{2+\alpha} = 1 \tag{3}$$

Oppure allineare le equazioni a sinistra, invece di allineare i segni di =:

```
\begin{eqnarray}
&& x+y=3 & \\\
&& x-y^{2+\alpha}=1 & \\
\end{eqnarray}
```

$$x + y = 3 \tag{4}$$

$$x - y^{2+\alpha} = 1 \tag{5}$$

L'environment `align` (occorre avere il package `amsmath` indicato nel preambolo) fa la stessa cosa che `eqnarray`, ma non ci sono spazi parassiti (e quindi è molto meglio)

```
\begin{align}
%\nonumber to remove numbering (before each equation)
  x+y&=3 \\\
  x-y^{2+\alpha}&=1
\end{align}
```

$$x + y = 3 \tag{6}$$

$$x - y^{2+\alpha} = 1 \tag{7}$$

Con `array` si possono costruire delle matrici, e quindi anche fare degli allineamenti.

```
$$
\begin{array}{l}
x+y&=3 \\\
  x-y^{2+\alpha}&=1
\end{array}
$$
```

$$x + y = 3$$

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

Da notare che per usare `array` occorre passare prima in modo matematico mettendo i `$$` (con gli altri environment non era necessario). E poi che le righe non vengono automaticamente numerate.

Nel costrutto precedente `{ll}` sta a indicare che ogni riga è costituita da due colonne da allineare entrambe a sinistra. `l` sta per “left”. Altre possibilità sono `c` (“center”) e `r` (“right”). Avremmo scritto `{lcr}` se ogni riga fosse stata composta da quattro colonne, la prima da allineare a sinistra, la seconda centrata e le ultime due allineate a destra.

`array` è molto utile quando occorre dare un numero unico ad una equazione che si sviluppa su più di una riga (cosa che capita abbastanza spesso)

```
\begin{equation}\label{eq1}
\begin{array}{c}
...\\
...
\end{array}
\end{equation}
```

$$\begin{aligned}
 \phi'(\theta) &= E[iX e^{i\theta X}] = \\
 &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} i x e^{i\theta x} e^{-x^2/2} dx = \\
 &= -\frac{1}{\sqrt{2\pi}} i e^{i\theta x} e^{-x^2/2} \Big|_{-\infty}^{+\infty} + \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} i \cdot i\theta e^{i\theta x} e^{-x^2/2} dx = \\
 &= -\theta \phi(\theta)
 \end{aligned} \tag{8}$$

Grazie alla (8)...

Non è venuta un gran che. Infatti gli elementi di `array` vengono scritti in modo testo. Si può però forzare il passaggio al modo display (basta aggiungere, all’inizio di ogni riga, il comando `\displaystyle`):

$$\begin{aligned}
\phi'(\theta) &= E[iX e^{i\theta X}] = \\
&= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} ix e^{i\theta x} e^{-x^2/2} dx = \\
&= -\frac{1}{\sqrt{2\pi}} i e^{i\theta x} e^{-x^2/2} \Big|_{-\infty}^{+\infty} + \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} i \cdot i\theta e^{i\theta x} e^{-x^2/2} dx = \\
&= -\theta \phi(\theta)
\end{aligned} \tag{9}$$

Se non è necessario numerarle, le equazioni si possono centrare con il costrutto `\displaylines`:

```

$$
\displaylines{
\phi'(\theta)=E[iX e^{i\theta X}]
={1\over \sqrt{2\pi}}\int_{-\infty}^{+\infty}
ix e^{i\theta x}e^{-{x^2/ 2}}\,dx=\cr
=-{1\over \sqrt{2\pi}} ie^{i\theta x}e^{-{x^2/ 2}}
\Big|_{-\infty}^{+\infty}+
{1\over \sqrt{2\pi}}\int_{-\infty}^{+\infty}
i\cdot i\theta e^{i\theta x}e^{-{x^2/ 2}}\,dx=
-\theta\phi(\theta)\cr
}
$$

```

La costruzione è

```

$$\displaylines{
le vostre equazioni, separate tra di loro
da \cr oppure\
}$$

```

$$\begin{aligned}\phi'(\theta) &= E[iXe^{i\theta X}] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} ixe^{i\theta x} e^{-x^2/2} dx = \\ &= -\frac{1}{\sqrt{2\pi}} i e^{i\theta x} e^{-x^2/2} \Big|_{-\infty}^{+\infty} + \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} i \cdot i\theta e^{i\theta x} e^{-x^2/2} dx = -\theta\phi(\theta)\end{aligned}$$

Utile anche è saper scrivere cose come

$$|x| = \begin{cases} x & \text{se } x \geq 0 \\ -x & \text{se } x < 0 \end{cases}$$

Il costrutto è

```
|x|=\begin{cases}x&\mbox{se } x\ge0\\
-x &\mbox{se } x<0
\end{cases}
```

dove si vede anche che, per inserire del testo in una formula occorre usare `\mbox`. Da notare che se occorrono degli spazi, questi devono essere inclusi nel testo all'interno di `\mbox{}`, dato che in modo matematico `TEX` ignora gli spazi. Vedremo più tardi il costrutto `\mbox` (mathematical box)

Alcuni dettagli per completare gli elementi di base della composizione delle formule.

I puntini d'interpunzione... si codificano `\dots` e non `...` che darebbe `...` che è più brutto. Ad esempio `$1,2,\dots,n$` dà `1,2,...,n`. In modo matematico ci sono anche i puntini a mezza altezza (`\cdots`): `1\cdots n = n!` e in diagonale, `\ddots`, utili per una matrice: Ci sono anche i tre puntini in verticale (`\vdots`) e in diagonale `\cdot\cdot` (`\ddots`). Non sono invece previsti i puntini in diagonale nell'altro senso: `\cdot\cdot` (ma ce li si può sempre fabbricare... Vedi più tardi quando impareremo a programmare in `TEX`)

Ricordiamo che gli accenti sulle lettere si mettono in modo diverso a seconda che siamo in modo matematico o in modo testo. Ad esempio: `\tilde a` dà \tilde{a} . Per mettere una sbarra sopra un carattere ci sono due possibilità `\bar a` produce \bar{a} mentre `\overline a` produce \overline{a} (una sbarretta un po' più lunga). Questo secondo costrutto può mettere una sbarra anche su un gruppo più lungo:

`\overline{zw}=\overline{z}\, \overline{w}`

ad esempio produce $\overline{zw} = \overline{z}\overline{w}$.

Informazioni sulle diverse possibilità di accenti in modo matematico si trovano nel menu “Math” di winedit. Per gli accenti in modo testo occorre invece fare riferimento al menu “International”.

Prima di provare tutte queste possibilità negli esercizi, vediamo come si può produrre una formula come la seguente:

$$Z_n \xrightarrow[n \rightarrow \infty]{L^p} Z$$

Un modo semplice fa uso del comando `\mathop`: `\mathop{\dots}` attribuisce a \dots gli attributi di un grande operatore. Dunque `\mathop{\to}` scrive una freccia \rightarrow , ma attribuendole le proprietà dei grandi operatori. In particolare esponenti e indici vengono per default posizionati sopra e sotto il simbolo \rightarrow . La formula precedente si può dunque ottenere mediante

`\dots\mathop{\to}^{\{L^p\}}_{\{n\to\infty\}}\dots`

ESERCIZI

Esercizio 12.

$$\lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n = ?$$

Esercizio 13.

$$\lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n = ?$$

Esercizio 14.

$$\begin{aligned} x_1 + x_3 &= 5 \\ x_1 + 2x_2 &= 7 \\ 5x_1 - x_2 + x_3 &= 11 \end{aligned} \tag{10}$$

Esercizio 15.

$$\left(\sum_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n}} \frac{1}{i!j!} \right) \quad \left(\sum_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n}} \frac{1}{i!j!} \right) \quad \left(\sum_{\substack{1 \leq i \leq m \\ 1 \leq j \leq n}} \frac{1}{i!j!} \right)$$

Esercizio 16. I numeri di Eulero

$$\langle n \rangle_k$$

Esercizio 17.

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & \text{se } x > 0 \\ 0 & \text{altrimenti} \end{cases} \tag{11}$$

Esercizio 18. Una matrice con le parentesi quadre:

$$\begin{bmatrix} 1 & a & a^2 & a^3 \\ a & a^2 & a^3 & 1 \\ a^2 & a^3 & 1 & a \\ a^3 & 1 & a & a^2 \end{bmatrix}$$

Esercizio 19. $||x| - |y|| \quad 2^{2^{2^{2^2}}}$

Esercizio 20.

$$\pi = 2 \times \frac{2}{\sqrt{2}} \times \frac{2}{\sqrt{2 + \sqrt{2}}} \times \frac{2}{\sqrt{2 + \sqrt{2 + \sqrt{2}}}} \times \dots$$

Esercizio 21.

$$\sum_{\substack{1 \leq i \leq n \\ I \subset J \subset K}} A_I x_{i,J}^2$$

Esercizio 22.

$$\overline{z + z'} = \bar{z} + \bar{z'} \quad \overline{\sigma^2} \neq \bar{\sigma}^2$$

Esercizio 23.

$$\prod_{k \geq 0} \frac{1}{1 - q^k z} = \sum_{n \geq 0} \frac{z^n}{\prod_{1 \leq k \leq n} \frac{1}{1 - q^k z}}$$

Esercizio 24.

$$\Gamma(z) = \sum_{n=0}^{\infty} \frac{(-1)^n}{n!(n+z+1)} + \int_1^{+\infty} t^{z-1} e^{-t} dt$$

$$\Gamma(z) = \int_0^{+\infty} t^{z-1} e^{-t} dt$$

$$\Gamma(z+1) = z \Gamma(z)$$

Esercizio 25.

$$\begin{pmatrix} x_1 & * & \dots & * \\ y_1 & x_2 & \dots & * \\ 0 & \dots & \dots & \vdots \\ 0 & 0 & y_{n-1} & x_n \end{pmatrix}$$

Esercizio 26.

$$J = \begin{pmatrix} \begin{pmatrix} \lambda & 1 \\ 0 & \lambda \end{pmatrix} & 0 \\ 0 & \begin{pmatrix} \mu & 1 & 0 \\ 0 & \mu & 1 \\ 0 & 0 & \mu \end{pmatrix} \end{pmatrix}$$

Esercizio 27.

$$\Lambda^*(x) = x \log \frac{x}{p} + (1-x) \log \frac{1-x}{1-p} \quad \text{se } 0 \leq x \leq 1.$$

Esercizio 28. (Una formula numerata su due righe, la prima un po' spostata a sinistra, la seconda un po' spostata a destra)

$$\begin{aligned} \phi'(\theta) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} ixe^{i\theta x} e^{-x^2/2} dx = \\ &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} i \cdot i\theta e^{i\theta x} e^{-x^2/2} dx = -\theta\phi(\theta) \end{aligned} \tag{12}$$

Esercizio 29.

$$y = \begin{cases} 1 & \text{se } 0 \leq x \leq 1 \\ \frac{1}{2} & \text{se } 1 \leq x \leq 2 \end{cases}$$

Esercizio 30.

$$E\left(\int_0^t X_s dB_s\right) = 0 \tag{13}$$

$$E\left[\left(\int_0^t X_s dB_s\right)^2\right] = E\left(\int_0^t X_s^2 ds\right) \tag{14}$$

e

$$E\left(\int_0^t X_s dB_s\right) = 0$$

$$E\left[\left(\int_0^t X_s dB_s\right)^2\right] = E\left(\int_0^t X_s^2 ds\right)$$

Esercizio 31.

$$S_1 + a_1 = 0 \tag{15}$$

$$S_2 + S_1 a_1 + 2a_2 = 0 \tag{16}$$

...

$$S_n + S_{n-1}a_1 + S_{n-2}a_2 + \dots + S_1 a_{n-1} + na_n = 0 \tag{17}$$

Esercizio 32.

$$\frac{1}{a + \frac{1}{a + \frac{1}{a + \dots}}}$$

Esercizio 33.

$$f(x) \stackrel{\text{def}}{=} \frac{1}{1+x^2}.$$

Esercizio 34.

$$X_n \xrightarrow[n \rightarrow \infty]{L^2} X_\infty$$

SCATOLE

TEX è fatto di scatole.

Una scatola è definita da tre dimensioni:

l'*altezza* al di sopra della linea di base,

la *profondità* al di sotto della linea di base

la *larghezza*



Una o più di queste dimensioni può essere uguale a zero, nel qual caso la scatola può risultare invisibile quando viene stampata, ma questo non vuol dire che la scatola non produca un effetto.

Ad esempio qualche volta bisogna cambiare artificialmente la dimensione di una scatola, come nel caso della formula

$$\sqrt{d} + \sqrt{g} + \sqrt{a}$$

la cui rappresentazione non è gradevole, con le $\sqrt{\quad}$ che vanno su e giù. La quantità `\mathstrut` è una riga verticale della stessa altezza e profondità di “(”, ma di larghezza 0, quindi invisibile. Se si aggiunge all’argomento di `\sqrt{\quad}` una `\mathstrut`, si ottiene

$$\sqrt{d} + \sqrt{g} + \sqrt{a}$$

che va meglio. Un effetto simile si ottiene con i comandi

``, `\hphantom{testo}`, `\vphantom{testo}`

che producono delle scatole invisibili aventi rispettivamente le dimensioni, le dimensioni orizzontali, le dimensioni verticali della scatola formata dal testo. Un altro comando utile è `\smash`: `\smash{\dots}` produce una scatola contenente il testo `\dots`, ma che TEX considera avere altezza e profondità uguali a 0.

Infine segnaliamo il comando

```
\vrule height#1pt depth#2pt width#3pt,
```

che produce una riga (rule) verticale di altezza sulla linea di base uguale a **#1pt**, di profondità **#1pt** e di spessore **#3pt**. Se **#3 = 0**, allora la riga è invisibile, ma la scatola che viene prodotta ha necessariamente altezza e profondità date dalle altre due dimensioni.

Del tutto simile

```
\hrule height#1pt depth#2pt width#3pt.
```

Una scatola si può costruire con il comando:

```
\makebox[larg] [pos] {testo}
```

Questo comando serve per costruire una scatola di larghezza **larg** e contenente il testo voluto. Il parametro **pos** può essere: **l** (allineato a sinistra), **r** (allineato a destra), **c** (centrato) oppure **s** (allargato al massimo ad occupare tutto lo spazio).

Quando si lavora con le scatole, possono venire utili i comandi `\hspace{\fill}`, `\hspace*{\fill}`, `\vspace{\fill}`, `\vspace*{\fill}` che inseriscono spazi elastici che si espandono al massimo all'interno della scatola.

Una idea simile si può utilizzare per costruire delle frecce oppure delle righe di lunghezza voluta. `\rightarrowfill` produce una freccia verso destra di dimensione pari a tutta la scatola. Vi potete immaginare cosa fa il comando `\leftarrowfill`. `\hrulefill` produce una riga orizzontale.

Provate a produrre:

Esercizio 35.

Esercizio 36. $f : E \longrightarrow \mathbb{C}$

Esercizio 37.

$$\begin{array}{ccc} \mathbb{R} & \xrightarrow{\pi} & \mathbb{R}/\mathbb{Z} \\ g \downarrow & & \downarrow \hat{g} \\ \mathbb{R} & \xrightarrow{\hat{\pi}} & \mathbb{R}/\mathbb{Z} \end{array}$$

Esercizio 38.

$$\sqrt{d} + \sqrt{a_j} \quad \sqrt{a} + \sqrt{b} + \sqrt{x_j^2}$$

Esercizio 39.

$$A = \begin{pmatrix} \begin{pmatrix} J_1 \end{pmatrix} & 0 \\ 0 & \begin{pmatrix} J_2 \end{pmatrix} \end{pmatrix}$$

Esercizio 40.

$$(x_1 \ \dots \ x_m) \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix} = (y_1 \ \dots \ y_m)$$

Esercizio 41.

$$\sigma_1^2 := \sum_{\ell_1 \ell_2} C_{\ell_1} a^2\left(\frac{4\ell_1}{N}\right) C_{\ell_2} a^2\left(\frac{4\ell_2}{N}\right) K_N\left(\frac{2\pi}{N}(\ell_1 + \ell_2)\right), \quad (18)$$

$$\sigma_2^2 := \sum_{\ell_1 \ell_2 \ell_3 \ell_4} C_{\ell_1} a^2\left(\frac{4\ell_1}{N}\right) C_{\ell_2} a^2\left(\frac{4\ell_2}{N}\right) C_{\ell_3} a^2\left(\frac{4\ell_3}{N}\right) C_{\ell_4} a^2\left(\frac{4\ell_4}{N}\right) \times \quad (19)$$

$$\times K_N\left(\frac{2\pi}{N}(\ell_1 + \ell_2 + \ell_3 + \ell_4)\right).$$

(qui il problema principale è come disporre due equazioni numerate, allineate a destra e con la seconda che si estende su due righe)

Il comando `\mbox{testo}` equivale a `\makebox{testo}`; la differenza è che non è possibile inserire i comandi opzionali []. Questo comando serve soprattutto a inserire del testo in una formula matematica:

$$f(x) \geq g(x) \quad \text{per ogni } x \text{ irrazionale}$$

Sottolineiamo che se si è in modo matematico o in modo formula, all'interno della `\mbox` si è in modo testo. In particolare se si vogliono mettere delle formule nella `\mbox` occorrerà mettere i `$ $`. Esistono i comandi `\framebox[larg][pos]{testo}` e `\fbox` che fanno la stessa cosa, ma producono una scatola con una cornice.

Non se ne può più!

Esercizio 42.

Si definisce il nucleo di Fejér K_n mediante

$$K_n(t) := \sum_{k=-n}^n \left(1 - \frac{|k|}{n+1}\right) e^{2\pi i k t} = \frac{1}{n+1} \sum_{k=1}^{n+1} \sum_{r=1}^{n+1} e^{2\pi i k t} e^{-2\pi i r t}$$

Il modo testo: capitoli, paragrafi...

Ci siamo finora essenzialmente occupati della costruzione delle formule. Vediamo ora come si producono gli altri elementi che compongono un documento, quindi il testo che sta intorno alle formule.

Un documento si compone di solito di capitoli, paragrafi, ... Naturalmente questo dipende dal tipo di documento che si sta preparando: un articolo in genere non è suddiviso in capitoli, mentre questo accade per testi come libri o tesi.

Dato che per ora ci occupiamo soprattutto della classe di documenti **book**, quest'ultima è la situazione che ci interessa (vedremo le caratteristiche della classe **article** più tardi).

Per la formattazione di capitoli e paragrafi \LaTeX prevede i comandi `\chapter{}`, `\section{}` e `\subsection{}` (i sottoparagrafi).
Ade esempio

```
\chapter{The Peter-Weyl Theorem}
\section{Group representations}\label{sec-gr}
Let  $G$  be a topological group\dots
```

produce il risultato visibile nella prossima pagina

Chapter 1

The Peter-Weyl Theorem

1.1 Group representations

Let G be a topological group...

Da notare che

- Non c'è bisogno di scrivere la parola “Chapter”, questa viene scritta direttamente da \LaTeX . Vedremo poi come convincere \LaTeX a scrivere “Capitolo” se stiamo scrivendo un testo in italiano o “Chapître” se in francese.
- Può essere successo che \LaTeX abbia lasciato una pagina bianca. Questo perché in tipografia i capitoli cominciano sempre alle pagine dispari (quelle di destra). Nel caso viene quindi lasciata una pagina bianca.
- Il numero del paragrafo e del capitolo sono stati scritti direttamente da \LaTeX . Mettendo un label si può richiamare il numero del paragrafo.

Ad esempio `As stated in section \ref{sec-gr}\dots` produrrebbe ora: As stated in section 1.1...

ENUNCIATI, CONTATORI E NUMERAZIONI

Gli enunciati (teoremi, definizioni, ...) in T_EX sono degli environment. Ad ognuno di essi devono essere associate necessariamente due cose: un contatore per la loro numerazione ed un nome. Il nome è quello che viene stampato: **Teorema**, **Proposizione**, **Lemma**,... Ma anche **Congettura**, **Idea sbagliata**,... Potete scegliere quello che volete. Il contatore è una variabile che T_EX crea e che riporta la numerazione. Gli enunciati sono sempre numerati. Gli enunciati vengono definiti mediante il comando `\newtheorem`.

Ad esempio

```
\newtheorem{teorema}{Teorema}[section]
```

oppure

```
\newtheorem{lemma}[teorema]{Lemme}
```

che vanno inseriti nel preambolo. Una volta date queste definizioni, per produrre un enunciato occorrerà scrivere

```
\begin{teorema}\label{teo1}  
...(enunciato qui)...  
\end{teorema}
```

Il comando `\label{teo1}` associa alla stringa `teo1` il numero del Teorema, che potrà essere richiamato scrivendo `\ref{teo1}`, come si faceva per la numerazione delle equazioni.

Ad esempio:

```
\begin{teorema}\label{teo1} Date delle misure
di probabilit\'a  $\mu, \mu_n,$ 
 $n=1,2,\dots$  sullo spazio metrico localmente
compatto separabile
 $E$ , allora  $\mu_n \rightarrow \mu$  strettamente se e solo se
 $\mu_n(f) \rightarrow$ 
 $\mu(f)$  per ogni funzione continua a supporto
compatto.
\end{teorema}
```

Come dice il Teorema [\ref{teo1}](#)...

Produce:

Teorema 1.1.1 *Date delle misure di probabilità $\mu, \mu_n, n = 1, 2, \dots$ sullo spazio metrico localmente compatto separabile E , allora $\mu_n \rightarrow \mu$ strettamente se e solo se $\mu_n(f) \rightarrow \mu(f)$ per ogni funzione continua a supporto compatto.*

Come dice il Teorema 1.1.1...

I due comandi

```
\newtheorem{teorema}{Theorem}[section]
```

e

```
\newtheorem{lemma}[teorema]{Lemme}
```

differiscono per l'ordine degli argomenti tra [] e { } e fanno due cose diverse.

Il primo crea un nuovo environment di tipo theorem (un enunciato, cioè). L'argomento della prima parentesi è il nome dell'environment, cioè voi dovrete scrivere

```
\begin{teorema}... \end{teorema},
```

il secondo (Theorem) il nome che verrà stampato, il terzo indica il contatore di riferimento (che in questo esempio è il contatore delle section, cioè dei paragrafi). Il comando crea anche un contatore che porta il nome teorema, che servirà a numerare progressivamente questi enunciati. Questo contatore verrà rimesso a 1 ogni volta che il contatore di riferimento (quello delle section in questo esempio) cambierà di valore. Questo vuole dire che la numerazione dei teoremi sarà del tipo Theorem 2.1, Theorem 2.2,... fino a che non si passerà alla sezione 3, e allora ricomincerà da 1: Theorem 3.1, Theorem 3.2,... Nel nostro caso, dato che usiamo la classe **book** anzi la numerazione sarà su 3 cifre, con la prima uguale al numero del capitolo.

Il secondo crea un nuovo enunciato, chiamato lemma. Nel testo stampato apparirà il nome Lemme (che è il terzo argomento), mentre il secondo argomento, tra parentesi quadre, indica che questo tipo di enunciati segue la stessa numerazione dei teoremi. Se invece avessimo preferito che i lemmi avessero una numerazione indipendente dai teoremi avremmo dovuto dare il comando

`\newtheorem{lemma}{Lemme}[section]`

Tutto questo lavoro di numerazione viene fatto da $\text{T}_{\text{E}}\text{X}$ con l'uso di *contatori*, che seguono l'evoluzione del numero dei capitoli, delle section (i paragrafi), delle equazioni. . .

È il caso di vedere come funzionano i contatori e come si possono creare, e personalizzare. I comandi relativi sono (supponiamo di lavorare con un contatore chiamato `cnt`):

`\newcounter{cnt}[contatore di riferimento]` → definisce il nuovo contatore `cnt`. Quando il contatore di riferimento viene incrementato di 1, il nuovo contatore `cnt` viene azzerato (come per il contatore `section` di prima). Il contatore di riferimento è opzionale.

Il comando `\newtheorem{cnt}{Theorem}[section]` provvede anche a creare un nuovo contatore chiamato `cnt`, con contatore di riferimento `section`.

Alcuni comandi per la manipolazione dei contatori (utili a capire il loro funzionamento, ma che capita abbastanza di rado di utilizzare)

`\setcounter{cnt}{7}` → attribuisce il valore 7 al contatore `cnt`.

`\addtocounter{cnt}{1}` → aggiunge 1 al valore del contatore `cnt`

`\value{cnt}` → fornisce il valore di `cnt` (ma non lo stampa).

`\setcounter{teorema}{\value{cnt}}` quindi assegna al contatore `teorema` il valore del contatore `cnt`.

Per stampare sul testo il valore di un contatore ci sono i comandi

`\arabic{cnt}` che produce i numeri in cifre arabe
`\roman{cnt}` che produce i numeri in cifre romane minuscole
`\Roman{cnt}` che produce i numeri in cifre romane maiuscole
e altri meno importanti.

Quando un contatore, chiamato `cnt` ad esempio, viene creato, viene sempre definito anche il comando `\thecnt`. Il comando `\thecnt` viene usato da L^AT_EX per produrre sul testo il valore del contatore. Con le definizioni di poco fa era stato definito un contatore `teorema`. Se scrivete nel vostro testo L^AT_EX `\theteorema`, nel testo prodotto apparirà stampato il numero a cui è arrivata la numerazione dei teoremi fino a quel momento.

Ridefinendo questo comando è possibile personalizzare il layout.

Ad esempio con

```
\renewcommand{\theteorema}
{\Roman{section}.\arabic{teorema}}
```

si otterrà che negli enunciati e nei cross-referencing compaia il numero del capitolo in numeri romani e quello del teorema in numeri arabi.

Esercizio 43. Produrre

Définition 1.1.2 *On définit le noyau de Fejér K_n par la formule:*

$$K_n(t) = \sum_{k=-n}^n \left(1 - \frac{|k|}{n+1}\right) e^{2\pi ikt} = \frac{1}{n+1} \sum_{k=1}^{n+1} \sum_{r=1}^{n+1} e^{2\pi ikt} e^{-2\pi irt}$$

Esercizio 44. Produrre

Teorema 1.3 *The representation \hat{U} is irreducible.*

As it is stated in Theorem 1.3,

Esercizio 45. Come fare per ottenere

Teorema e Definizione 1.4 *Sia $\mathcal{B} \subset \mathcal{A}$ una sotto- σ -algebra e X una v.a. reale integrabile (risp. positiva). Allora esiste una v.a. reale integrabile (risp. positiva) Y , unica a meno di una equivalenza, tale che*

$$\int_A Y dP = \int_A X dP, \quad \text{per ogni } A \in \mathcal{B}$$

Bisogna comunque fare attenzione: il comando `\thecnt` viene usato *sia* quando il teorema viene scritto, *sia* quando il teorema viene richiamato come referenza. Quindi se vi inventate un modo fantasioso per scrivere il numero di un teorema, poi lo stesso modo fantasioso ve lo ritroverete tutte le volte che quel numero viene richiamato come cross-reference.

[??]

È opportuno capire come fa \LaTeX a produrre i cross-referencing. Per queste operazioni \LaTeX si serve di un file `.aux`. In questo file vengono scritte tutte le informazioni che legano i `\label` con i numeri che ad essi corrispondono. Ad esempio se il teorema a cui avete attribuito il `\label{teo-fondamentale}` è il Teorema 3.4, nel file `.aux` ci sono le informazioni che permettono a \LaTeX di produrre il numero corretto 3.4 tutte le volte che incontra il comando `\ref{teo-fondamentale}`.

Il file `.aux` viene letto da \LaTeX all'inizio della compilazione e viene poi riscritto durante la compilazione. Dunque se avete fatto dei cambiamenti nel file e ricompilate, il risultato sarà un articolo (o un libro) in cui le numerazioni sono aggiornate, ma i cross-referencing saranno basati sul file `.aux vecchio`. Quindi, se avete aggiunto, ad esempio, un teorema le numerazioni sono cambiate e i cross-referencing porteranno i numeri vecchi (non aggiornati). Perché tutto sia a posto occorrerà compilare una seconda volta.

Su questo occorre fare attenzione. È molto facile dimenticarsi di questo aspetto della doppia compilazione (correzioni dell'ultimo minuto...)

MODO TESTO: LE FONT

Vediamo come gestire i vari tipi di carattere. Una stessa font può possedere vari stili:

grassetto `\textbf{}` oppure `{\bf ...}`:

abcdef...ABC

inclinato (slanted) `\textsl{}` oppure `{\sl ...}`:

abcdef...ABC

italico `\textit{}` oppure `{\it ...}`:

abcdef...ABC

grassetto inclinato (combinando `\textsl{}` e `\textbf{}`):

abcdef...ABC

grassetto italico (combinando `\textit{}` e `\textbf{}`):

abcdef...ABC

Oltre a questi che sono i vari stili della font roman, c'è la font senza i trattini (si chiamano le “grazie”, “serif” in inglese): `{\sf ...}`:

abcdef...ABC

e quella che usa come minuscole delle piccole maiuscole: csc (capitals small capitals) `\textsc{}` oppure `{\sc ...}`:

ABCDEF...ABC

Un altro tipo di font di uso frequente è quella calligrafica:

\mathcal{F} e \mathcal{A} producono \mathcal{F} e \mathcal{A}

Se usate il package `rsfs` (bisogna aggiungere `mathrsfs` alla lista dei package nel preambolo), \mathscr{F} e \mathscr{A} producono \mathscr{F} e \mathscr{A} .

Nelle font calligrafiche sono disponibili solo le lettere maiuscole e sono utilizzabili solo in modo matematico o formula.

Altri comandi riguardano le dimensioni:

`{\tiny ...}`: abcdef...ABC

`{\footnotesize ...}`: abcdef...ABC

`{\small ...}`: abcdef...ABC

`{\normalsize ...}`: abcdef...ABC

`{\large ...}`: abcdef...ABC

`{\Large ...}`: abcdef...ABC

`{\LARGE ...}`: abcdef...ABC

`{\huge ...}`: abcdef...ABC

`{\Huge ...}`: abcdef...ABC

Questo testo è in `{\Large ...}`.

Enumerazioni e liste

Ci sono varie possibilità in questo senso. Ad esempio l'environment `itemize`

I vantaggi di T_EX su altri sistemi sono

- Una grande efficienza nella realizzazione di un output tipografico di qualità, grazie a degli algoritmi di posizionamento del materiale particolarmente raffinati.
- La capacità di tenere conto di dettagli tipografici raffinati, come il *kerning*.
- La portabilità su virtualmente ogni sistema operativo.
- La rapidità di realizzazione, che permette di ottenere formule complesse con una digitazione relativamente semplice.

La sintassi è molto semplice:

```
\begin{itemize}
\item ...
\item
\end{itemize}
```

L'environment **enumerate**, che usa numeri progressivi invece dei pallini:

1. Una grande efficienza nella realizzazione di un output tipografico di qualità, grazie a degli algoritmi di posizionamento del materiale particolarmente raffinati.
2. La capacità di tenere conto di dettagli tipografici raffinati, come il *kerning*.
3. La portabilità su virtualmente ogni sistema operativo.
4. La rapidità di realizzazione, che permette di ottenere formule complesse con una digitazione relativamente semplice.

(sintassi identica, cambia solo il nome dell'environment)

Infine l'environment **description**, utile per produrre dizionari e simili.

c terza lettera dell'alfabeto, consonante, che pronunciasi *ci*...

ca' apocope di casa.

càbala *sf* cabbala; *com* arte che, per mezzo di numeri, lettere e segni, pretende aver commercio con esseri soprannaturali...

cabalétta arietta musicale, vivace e cadenzata, che per lo più serve di chiusa a un duetto o a un assolo.

La sintassi è

```
\begin{description}
\item [c\'abala] ...
\item [#] ...
\end{description}
```

Questi environment, a mio gusto lasciano troppo spazio bianco.

Sottigliezze tipografiche (e non)

- Trattini. Ce ne sono di tre tipi:
corti: – per nomi doppi: Galton-Watson, Porta Furba - Quadra-
ro...
medi: -- per indicare intervalli: "...alle pagine 35–37..."
lunghi: --- per parentesi del discorso "...era stanco — o comun-
que così diceva — ...".
- puntini di sospensione: abbiamo già visto che non si fanno scri-
vendo tre puntini ..., ma con `\dots`: ...Questo funziona sia in
modo testo che in modo matematico: **C'era una volta\dots**
dà C'era una volta... Ricordare che dopo i puntini non si mette
mai il punto fermo: `\dots.` produrrebbe che è brutto.
- Le virgolette aprenti si codificano ‘ ‘ (doppio accento grave),
mentre quelle chiudenti sono ’ ’ (doppio accento acuto o apostrofo):
“Il coraggio, chi non ce l’ha non se lo può dare”
- Ricordare che in italiano gli accenti sulle lettere a, i, o, u sono tutti
gravi: à, ì, ò, ù. Sulla lettera e: è (verbo essere) è grave, come cioè.
Sono invece *acuti* gli accenti perché, poiché, né, affinché acciocché,
conciossiacosaché,...
- L’interlinea (distanza tra le le linee di base di due righe consecu-
tive) si chiama `\baselineskip`. Come tutte le lunghezze questa
quantità può essere modificata. `\baselineskip12pt` porta l’in-
terlinea a 12punti. `\baselineskip=2\baselineskip` moltiplica

per due l'interlinea. Questo comando è locale: se viene dato all'interno di un gruppo (`{ }`), esso vale solo lì. Se viene dato all'interno di un paragrafo, si applica a tutto il paragrafo dall'inizio. Questo paragrafo è realizzato con `\baselineskip24pt`.

- Ci sono tre comandi per lasciare delle righe bianche o comunque per spaziare degli elementi di testo tra di loro:

`\smallskip`

`\medskip`

`\bigskip`

Essi lasciano uno spazio bianco pari rispettivamente a $\frac{1}{4}$, $\frac{1}{2}$ e 1 dell'interlinea (`=\baselineskip`). Con questi comandi bisogna sempre lasciare una riga bianca dopo il comando. Volendo una distanza su misura, si può scrivere `\vspace{dim}`, dove `dim` indica una lunghezza qualunque, anche negativa.

Queste spaziature comunque vengono ignorate se cadono all'inizio di una pagina. Se si vuole imporre uno spazio anche in questo caso il comando è `\vspace*{dim}`.

- Quando s'inizia un paragrafo nuovo, `TEX` lo fa iniziare un po' rientrato ("indented"). Se si vuole iniziare un paragrafo senza la l'indentazione basta fare precedere il paragrafo dal comando `\noindent`.

La lunghezza dell'indentazione si controlla con il parametro `\parindent`. Si tratta di una dimensione. `\parindent=0pt` annulla l'indentazione. Questo testo è scritto con `\parindent=0pt`. V'immaginate cosa produce `\parindent=-4pt`?

- Quando si scrive in una lingua diversa dall'inglese, che è la lingua di default di T_EX, la troncatura delle parole quando si va a capo non è più corretta. Questo si risolve con il package `babel`. Con il comando

```
\usepackage[italian]{babel}
```

le andate a capo si fanno secondo le regole della lingua italiana. Inoltre vengono tradotte in italiano un certo numero di parole che compaiono nei testi: `abstract`→riassunto, `chapter`→capitolo, `index`→indice, E si tiene conto di alcune consuetudini tipografiche italiane.

È anche possibile utilizzare in un testo più di una lingua: il comando

```
\usepackage[english,czech,italian]{babel}
```

indica che verranno usate, eventualmente, le lingue inglese, ceca e italiana. L'ultima delle lingue indicate (l'italiano in questo caso) è quella di default. Per passare al ceco basterà inserire il comando `\selectlanguage{czech}` (che però funziona solo dopo `\begin{document}`)

Qualche volta anche così le andate a capo sono scorrette. Si può provvedere con l'inserimento di `\-` all'interno della parola: scrivendo `de\ -fi\ -ni\ -zio\ -ne` T_EX cercherà di andare a capo in uno dei punti in cui avete messo `\-` (e non stamperà gli `\-`).

Un'altra possibilità consiste nell'indicare esplicitamente le andate a capo delle parole più "critiche" nel preambolo. Ad esempio

```
\hyphenation{ cre-scen-te de-cre-scen-te sem-pli-ci  
pa-ra-me-tro geo-me-tri-ca geo-me-tri-che in-sie-me}
```

- Per centrare un testo (un titolo, o più righe di testo):

`\begin{center}...\end{center}`

Allo stesso modo `\begin{flushright}...\end{flushright}` e `\begin{flushleft}...\end{flushleft}` iniziano dei paragrafi allineati a destra ed a sinistra rispettivamente (e senza giustificazione). Se si vuole inserire una porzione di testo senza iniziare un nuovo paragrafo (e senza lasciare una riga bianca all'inizio ci sono i comandi `\centering`, `\raggedright` e `\raggedleft`. Bisognerà rinchiuderli tra `{...}` in maniera che abbiano effetto solo all'interno di un gruppo. Questo paragrafo è in `{\raggedright ...}`.

Attenzione: perché questi comandi abbiano effetto occorre che il paragrafo venga terminato con un segno di fine paragrafo (`\par` oppure riga vuota).

Ad esempio

```
{\raggedright bla bla...  
...  
...bla bla}
```

non produrrà l'effetto voluto. Occorrerà scrivere

```
{\raggedright bla bla...  
...  
...bla bla  
}
```

- I caratteri `%`, `#`, `{}`, `$`, `\`, `_`, `^` sono di controllo. Abbiamo già visto a cosa servono `$ $`, `{ }`, sappiamo che `_` `^` servono a fare indici e esponenti e tutti i comandi L^AT_EX cominciano con `\`.

% invece serve per aggiungere dei commenti nel testo: L^AT_EX ignora il testo che segue % fino alla fine della riga. Se dobbiamo scrivere un testo in cui compaiono questi caratteri, basterà scrivere `\%, \#, \{, \}, \$, _`. Per scrivere `\` si potrà, in modo matematico battere `\backslash`: `\` oppure `\setminus`: `\` (non so se ci sia una differenza). Anche il carattere `~` è speciale: in modo testo indica uno spazio che non si può spezzare. Se, ad esempio, scrivete

`...come si vede dalla Figura 1...`,

si rischia che T_EX vada a capo tra **Figura** e **1**, che sarebbe antipatico. Scrivendo `Figura~1`, questa possibilità viene inibita. Per stampare una `~` si può scrivere `\~\` (un accento tilde su uno spazio bianco), che però è brutto: `~`, oppure passare in modo matematico e fare `\sim`: `\sim`. Ad esempio questo file si può trovare all'indirizzo:

www.mat.uniroma2.it/~processi/tex1.pdf.

- I comandi `\begin{verbatim}... \end{verbatim}` permettono di entrare nel modo `verbatim` nel quale cioè il testo viene riprodotto tale quale. All'interno di questo environment in particolare caratteri come `%, #, {, }` vengono riprodotti tali quali. E così pure spazi, e andate a capo. Utile ad esempio per inserire nel testo delle porzioni di codice di programmi.

Per inserire delle parole in `verbatim` all'interno di una riga si usa invece il costrutto `\verb#...#` dove `#` è un qualunque carattere (che però non deve comparire nel testo ...).

PERSONALIZZARE WINEDT

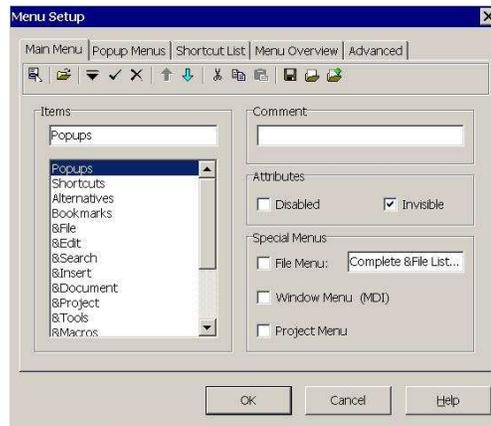
L'editor Winedt può essere predisposto in modo da rendere più facile la preparazione del file sorgente

- Per prima cosa si possono definire alcuni tasti in maniera da semplificare alcune operazioni. Ad esempio inseriamo una definizione per il tasto F9 nel modo seguente.

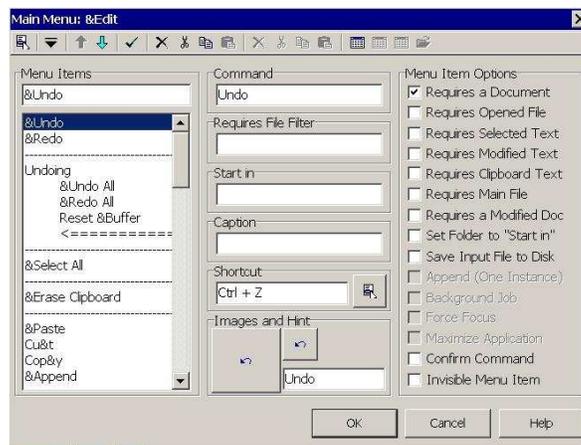
PRIMA copiare sul clipboard la riga

```
Ins("{");CMD("Char Left");
```

poi aprire il pop-up opzioni → Menu setup. Compare la finestra:



selezionare nella lista di sinistra "edit" Compare la finestra:



Posizionare il puntatore in fondo alla lista di sinistra fare clic col tasto *destra* del mouse. Scegliere insert→macro. Posizionare il puntatore sulla riga “Menu item” (dove c’è scritto “undo” nella figura precedente). Scrivere il nome della macro (a vostra scelta, “graffe”, ad esempio). Posizionare il puntatore sulla riga “macro” (accanto) e incollare la riga che avevate messo nel clipboard. Infine scendere nella riga “shortcut”, cliccare e premere il tasto F9. Uffah. Uscire dal menu. Provate ora a schiacciare il tasto F9 e vedete un po’ quello che fa. Questa personalizzazione è particolarmente utile se avete una tastiera italiana che vi obbligherebbe a delle circonvoluzioni per ottenere le parentesi graffe.

Può essere utile definire anche i tasti F6, F7 e F8, ad esempio con i tre comandi seguenti

```
Ins("‘ ")
Ins("$$");NewLine;NewLine;Ins("$$");CMD("Line Up");
Ins("$$");CMD("Char Left");
```

Queste personalizzazioni, dopo averci fatto un po’ la mano, rendono molto più veloce la scrittura.

- Winedt è automaticamente configurato con il correttore ortografico in inglese: le parole segnate in rosso sono quelle che il correttore ortografico indica come scorrette. I PC di quest’aula sono probabilmente già configurati in italiano. Se volete configurare in italiano il vostro PC personale, l’operazione si fa così:

Procuratevi il dizionario italiano di winedt, che si chiama it.dic. Potete trovarlo nel PC davanti a voi, probabilmente nella directory “C:\Programmi\WinEdt Team\WinEdt54\Dict\it.dic”.

oppure, su internet, nello stesso sito da cui avete scaricato winedit. Copiate il file it.dic nel vostro PC personale. Diciamo in

“C:\Programmi\WinEdt Team\WinEdt54\Dict\it.dic”.

Aprirete il pop-up options→Dictionary manager. Nella finestra del vostro PC “italiano” non comparirà. Per aggiungerlo fate clic sulla lista di sinistra col tasto destro del mouse e scegliete “insert”. Scrivete poi sulla riga “Dictionaries” il nome del dizionario (“italiano”, per esempio) e poi sulla riga “definition” l’indirizzo del file del dizionario, cioè

“C:\Programmi\WinEdt Team\WinEdt54\Dict\it.dic”,

o l’indirizzo dove lo avete messo.

Ora il dizionario è installato. Per usarlo correttamente dovete però renderlo attivo. Questo si fa cliccando col mouse destro su “italiano”, scegliendo “load” e poi scrocettando la casella “enable”. Dovete però anche disinserire il dizionario inglese, se no il correttore automatico vi dà come corrette anche le parole che sono scorrette in italiano ma corrette in inglese. Questo si fa cliccando col mouse destro su “english(common)”, scegliendo “close” e poi togliendo la crocetta alla casella “enable”.

Ogni volta che si vuole cambiare di dizionario bisognerà ripetere la stessa procedura: caricare (load) il dizionario, abilitarlo (crocettare “enable) e poi disabilitare il dizionario precedente.

HEADLINE E FOOTLINE

La pagina di T_EX è composta dalla sovrapposizione di tre scatole: la parte alta della pagina (headline) la pagina contenente il testo vero e proprio e la parte bassa. Come default nella classe **article** la parte alta è vuota ed in quella bassa al centro c'è il numero della pagina. Nella classe **book** i numeri stanno in alto, quelli pari a sinistra, quelli dispari a destra (tranne che nelle pagine in cui comincia un capitolo).

Intanto si possono¹ aggiungere delle note a pié di pagina con il comando `\footnote{...}`, il cui effetto si può vedere qui sotto. `\footnoterule` cambia la lunghezza della riga che separa la pagina principale dal testo della/e nota/e. Ci sono altri comandi² che permettono di gestire un diverso posizionamento delle note (ad esempio per metterle tutte in un apposita sezione alla fine del documento).

Attenzione, il comando `\footnote` dà dei risultati imprevedibili se usato all'interno di un comando come `\section`, oppure i campi `\author` che vedremo nella classe **article**. In questi casi occorre scrivere `\protect\footnote{...}`:

Per cambiare i contenuti di headline e footline c'è il comando `\pagestyle{opzioni}` dove **opzioni** può prendere i valori

- **empty** headline e footline sono vuote (a parte le note)
- **plain** headline vuoto e footline col solo numero della pagina (e le note).
- **headings**: dipende dalla class del documento.

¹Non funziona mica sempre!

²Ma che complicazione!

article: lo headline contiene il numero della pagina ed il titolo della section corrente; footline con le sole note.

book: lo headline contiene il numero della pagina ed il titolo del capitolo (pagine pari) o della section (pagine dispari). Fanno però eccezione le pagine iniziali dei capitoli, in cui lo headline è vuoto. Footline sempre con le sole note.

Esiste anche il comando `\thispagestyle{opzioni}`, del tutto simile come risultato, ma che ha effetto solo sulla pagina corrente. Questa pagina è stata ottenuta con `\thispagestyle{headings}`.

Attenzione, gli effetti di questi stili possono cambiare a seconda della documentclass (**book**, **article**,...) che avete scelto.

Esiste anche un'altra opzione **myheadings**, che permette all'autore di personalizzare il layout della pagina. Ma, se si vuole disegnare un proprio layout conviene utilizzare il package **fancyhdr** (che va dunque incluso nel preambolo).

Se si dà ora il comando `\pagestyle{fancy}`, si possono definire, in maniera abbastanza facile gli oggetti da mettere in alto e in basso.

Per maggiori dettagli si può dare un'occhiata alla documentazione del pacchetto.

BIBLIOGRAFIA

Ci sono varie possibilità di organizzare la bibliografia che di solito si trova alla fine dell'articolo o della tesi.

Nella pagina successiva si vede qual'è il risultato. Occorre codificare

```
\begin{thebibliography}{abc}
\bibitem{bib:BS}Baldi P., Sanz M., Une remarque
sur la th'eorie des grandes
d'eviations. S'eminare de
Probabilit'es XXV, 345--348, Lecture
Notes in Math. 1485, Springer, Berlin, 1991.
%
\bibitem{bib:deAc}de Acosta A., Small deviations
in the functional central limit
theorem with applications to functional laws of
the iterated
logarithm. {\it Ann. Probab.}, {\bf 11}, 78--101
(1983).
...
\end{thebibliography}
```

Se ora compare una pagina bianca, questo accade perché, nel modo **book**, la bibliografia, come i capitoli, inizia sempre in una pagina propria, che deve essere una pagina dispari. Con la classe **article** invece la Bibliografia inizia subito senza lasciare pagine bianche.

Bibliografia

- [1] Baldi P., Sanz M., Une remarque sur la théorie des grandes déviations. Séminaire de Probabilités XXV, 345–348, Lecture Notes in Math. 1485, Springer, Berlin, 1991.
- [2] de Acosta A., Small deviations in the functional central limit theorem with applications to functional laws of the iterated logarithm. *Ann. Probab.*, **11**, 78–101 (1983).

Da notare che la parola “Bibliografia” è stata scelta da \LaTeX sulla base del fatto che, in questo momento, la lingua prescelta è l’italiano. Se non avessimo inserito `babel` avremmo trovato “Bibliography”.

Questo environment permette di citare in maniera corretta i riferimenti bibliografici. Ad esempio `...vedi \cite{bib:deAc}...` produce “...vedi [2]....”

Il parametro `abc` indica il numero massimo di caratteri utilizzabili per la citazione (3: abc è una parola di 3 lettere)

L'environment accetta un parametro opzionale: se avessimo scritto

```
\begin{thebibliography}{abcd}
\bibitem[BS91]{bib:BS}Baldi P., Sanz M., Une
remarque sur la th\'eorie des grandes
d\'eviations. S\'eminaire de Probabilit\'es XXV,
345--348, Lecture
Notes in Math. 1485, Springer, Berlin, 1991.
%
\bibitem[dA83]{bib:deAc}de Acosta A., Small
deviations in the functional central limit
theorem with applications to functional laws
of the iterated
logarithm. {\it Ann. Probab.}, {\bf 11},
78--101 (1983).
%
...
\end{thebibliography}
```

avremmo ottenuto una bibliografia in cui i riferimenti sono indicati dagli acronimi [BS91], [dA83],... sia nell'elenco che nei riferimenti quando gli elementi vengono richiamati nel testo con il comando `\cite`.

Per la gestione dei riferimenti bibliografici esiste però uno strumento molto più potente che è **bibtex**, che è un programma esterno. Vediamo come si usa e quali sono i suoi vantaggi.

Nel testo sorgente basterà scrivere soltanto

```
\bibliography{bibbase}  
\bibliographystyle{siam}
```

Occorrerà avere preparato a parte un file, che nell'esempio considerato si chiamerà **bibbase.bib**, che conterrà i riferimenti bibliografici in un formato che poi vedremo.

Bisognerà poi, dopo aver compilato il documento, lanciare il programma esterno **bibtex** (che è già installato, vedi i comandi in alto di winedit) e poi ricompilare *due volte*. Questo programma andrà a cercare nel file **bibbase.bib** i riferimenti bibliografici che vengono citati nel documento. Dopo la seconda compilazione la bibliografia comparirà (come per miracolo...) ed anche i riferimenti.

Ad esempio, se ora si fanno le citazioni

```
...vedi\cite{BI:81} e \cite{BL:94}...
```

si ottiene "... [1] e [2]...".

Il file **bibbase.bib** in questione è un file in cui vengono messi insieme tutti i riferimenti bibliografici di cui l'autore si serve. È un file che diventa sempre più grosso e da cui **bibtex** pescherà solo i riferimenti che vengono effettivamente citati nel documento..

In esso i riferimenti compaiono sotto forma di schede del tipo seguente

```
@book{BL:94,  
AUTHOR = "N. Bouleau and D. Lepingle",  
TITLE = "Numerical methods for stochastic  
processes",  
PAGES = "",  
PUBLISHER = "J. Wiley and son",  
ADDRESS = "New York",  
YEAR = "1994"}
```

```
@article{BI:81,  
AUTHOR = "M. Bismut",  
TITLE = "Martingales, the Malliavin Calculus  
and hypoellipticity  
under general Hormander conditions",  
JOURNAL = "Z. Wahrscheinlichkeitstheorie  
verw. Gebiete",  
VOLUME = "56",  
PAGES = "469--505",  
YEAR = "1981"}
```

Il comando `\bibliographystyle{siam}` indica lo stile della bibliografia. Infatti ci sono molti modi di fare i riferimenti (con dei numeri, con un'abbreviazione,...) `siam` qui indica che si usa lo stile della Society for Industrial and Applied Mathematics. Ci sono varie decine di stili. Uno dei vantaggi di bibtex è che basta sostituire `siam` con un altro stile per cambiare il layout senza fatica.

Bibliografia

- [1] M. BISMUT, *Martingales, the Malliavin calculus and hypoellipticity under general Hormander conditions*, Z. Wahrscheinlichkeitstheorie verw. Gebiete, 56 (1981), pp. 469–505.
- [2] N. BOULEAU AND D. LEPINGLE, *Numerical methods for stochastic processes*, J. Wiley and son, New York, 1994.

Questo è il risultato

La creazione del file contenente la base dati bibliografica (bibbase.bib, nell'esempio di prima) è molto facile perché, ad esempio questi si possono trovare su MathSciNet.

Esercizio 46. 1) Creare un file .bib mettendo insieme tre riferimenti bibliografici su MathSciNet.
2) Inserire nel vostro file di lavoro le citazioni e dare i comandi per la creazione della bibliografia e dei riferimenti.

TABELLE

In modo testo può capitare di avere necessità di fare delle tabelle e quindi di dover allineare del materiale. L'environment apposito è `\tabular`. Ad esempio

Esame	voto	cfu
AL1	22	8
AM1	26	8
IN1	30	6
Inglese	si	4
totale		26

Il codice per ottenere la tabella precedente è immediatamente comprensibile:

```
\begin{tabular}{|l|c|r|}  
  \hline  
  Esame & voto & cfu \\  
  \hline  
  AL1 & 22 & 8 \\  
  \hline  
  AM1 & 26 & 8 \\  
  \hline  
  IN1 & 30 & 6 \\  
  \hline  
  Inglese & s\{'{\i} & 4 \\  
  \hline\hline  
  totale & & 26 \\  
  \hline  
\end{tabular}
```

Il significato dei comandi è semplice, ma ci sono dei dettagli da precisare. La sintassi dell'environment è

```
\begin{tabular}[pos. esterna]{pos. interna}  
\end{tabular}
```

Le specifiche di posizione interna sono, per cominciare, le solite (`c`, `l`, `r`), che indicano che il contenuto delle celle deve essere posizionato al centro, a sinistra o a destra. Ce ne sono però altre che possono servire. `p{100pt}` ad esempio specifica che la dimensione della casella debba essere di 100pt. Se il testo che ci va dentro è più lungo, allora verrà scritto su più di una riga.

Esercizio 47. Produrre

Esame	voto	cfu	data
AL1	22	8	05/02/2005
AM1	26	8	10/06/2005
IN1	30	6	25/02/2005
Inglese	sì	4	04/05/2005
totale		26	

Esame	voto	cfu
AL1	22	8
AM1	26	8
IN1	30	6
Inglese	sì	4
totale		26

Esame	voto	cfu
AL1	22	8
AM1	26	8
IN1	30	6
Inglese	ancora da convalidare	4
totale		26

Un'altra possibilità come specifica di posizionamento interno è `@{...}`, che permette di utilizzare come separatore delle colonne quanto specificato in `...`. Questo comando elimina anche gli spazi ai bordi delle colonne. Quindi specificare `@{\vrule}` produrrebbe

Esame	voto	cfu
AL1	22	8
AM1	26	8
IN1	30	6
Inglese	si	4
totale		26

Evidentemente `@{}` produce una tabella senza separazioni e senza spazi intercolonna.

Altri due comandi utili per creare tabelle contenenti sotto tabelle: `\multicolumn{num}{pos}{}`: crea una casella che verrà posizionata su una estensione di `num` caselle della tabella principale.

`\cline{#1-#2}`: traccia una linea orizzontale in corrispondenza delle colonne `#1` e `#2`.

Per capire come funziona, la cosa migliore è fare delle prove. Ad esempio la tabella

Telecom Italia		
mese	Prezzo	
	min	max

si produce con

```
\begin{tabular}{|c||r|c|}
\hline
\multicolumn{3}{|c|}{Telecom Italia}\
\hline
\hline
```

```

&\multicolumn{2}{c|}{Prezzo}\\
\cline{2-3}
mese&min&max\\
\hline
\end{tabular}

```

Esercizio 48. Completate la tabella precedente per ottenere

Telecom Italia		
mese	Prezzo	
	min	max
gen 2005	2.238	2.274
feb 2005	2.268	2.312
mar 2005	2.290	2.356

Tecnicamente parlando una tabella è una scatola. Due scatole possono essere posizionate l'una accanto all'altra. Per farlo occorre prima mettere le due tabelle in una `\mbox`, magari separate da un po' di spazio. In questo caso occorre anche specificare il parametro di posizionamento esterno che può essere uno dei valori `t`, `b`, `c` (cioè top, bottom oppure center). Due scatole con l'opzione `t`, ad esempio verranno allineate sulla riga superiore .

Esercizio 49. Realizzate

Esame	voto	cfu
AL1	22	8
AM1	26	8
IN1	30	6
Inglese	sì	4
totale		26

Esame	voto	cfu
AM2	23	7
GE1	27	8
IN2	21	7
PS1	29	5
LC1	30	4
totale		31

e poi

Esame	voto	cfu
AL1	22	8
AM1	26	8
IN1	30	6
Inglese	sì	4
totale		26

Esame	voto	cfu
AM2	23	7
GE1	27	8
IN2	21	7
PS1	29	5
LC1	30	4
totale		31

Esercizio 50. Realizzate

Funzioni	Primitive
$\frac{1}{1+x^2}$	$\arctan x + C$
$\frac{1}{\sqrt{x^2+h}}$	$\log(x + \sqrt{x^2+h}) + C$
e^{-x^2}	non esiste una primitiva esprimibile in maniera elementare

FLOATS (OGGETTI FLOTTANTI)

Qualche volta si vuole introdurre nel testo un oggetto “ingombrante” (una tabella, una figura,...), che non può essere spezzato in due. Se quest’oggetto si trova posizionato alla fine della pagina (e non ci sta nella pagina corrente) esso verrà spostato nella pagina successiva e la pagina corrente sarà mezza vuota. Un “float” è una scatola che T_EX si autorizza a inserire nel testo non proprio esattamente dove viene posizionata nella sorgente. Un esempio di float è

```
\begin{table}[!hbtpt]
materiale vario (una tabella ad esempio)
\end{table}
```

T_EX posizionerà la tabella, magari non proprio dove l’avete scritta voi, ma magari nella pagina successiva, o in fondo alla pagina, in un modo che lui stimerà essere quello che produce l’impaginazione più soddisfacente.

Il significato dei parametri **!hHbtpt** è il seguente:

h dà l’indicazione di fare apparire, di preferenza, la tabella là dove si trova (h=here).

t dà l’indicazione di fare apparire la tabella all’inizio (top) della pagina, quella attuale o la successiva.

b dà l’indicazione di fare apparire la tabella alla fine (bottom) di una pagina.

p dà l’indicazione di fare apparire la tabella in una pagina apposita riservata ai float.

! dà l’indicazione di privilegiare la vostra scelta, rispetto ad esigenze d’impaginazione.

Ad esempio `[h]` darà l'indicazione di inserire la tabella là dove si trova nel testo. Ma `TEX` si autorizza a spostarla se ritiene che ciò produca una pagina “brutta” (troppo vuota ad esempio). `[!h]` cercherà di forzare un po' la mano al programma, perché ponga la tabella là dov'è, anche se la pagina riuscirà un po' brutta (ma `TEX` imporrà la sua scelta se gli sembrerà che l'impaginazione che risulta è inaccettabile). `[hbt]` indicherà il vostro ordine di preferenza.

Vediamo un esempio per maggiori dettagli.

Come si vede dalla Tabella 1.1...

Che è stato ottenuto con

```
\begin{table} [!b]
\centering
\begin{tabular}{|c|p{180pt}|}
\end{tabular}
\caption{Da imparare a memoria\dots?\label{tab1}}
\end{table}
```

Come si vede dalla Tabella `\ref{tab1}\dots`

Da notare:

- Poiché abbiamo indicato `!b`, \TeX ha posto la tabella in fondo alla pagina.
- Il comando `\centering` pone la tabella al centro della riga. `\flushright` e `\flushleft` avrebbero rispettivamente posto la tabella allineata con i margini destro e sinistro.
- I caratteri della tabella sono più piccoli.
- Il comando `\caption{...}` che aggiunge la didascalia. Se `\caption` è assente, la tabella non viene numerata. Si può indicare un `\label` per richiamare il numero della tabella in cross-referencing, ma questo `\label` dovrà essere messo *dentro* la cap-

Funzioni	Primitive
$\frac{1}{1+x^2}$	$\arctan x + C$
$\frac{1}{\sqrt{x^2+h}}$	$\log(x + \sqrt{x^2+h}) + C$
e^{-x^2}	non esiste una primitiva esprimibile in maniera elementare

Tabella 1.1: Da imparare a memoria...?

tion. Da notare che $\text{T}_{\text{E}}\text{X}$ ha scritto “Tabella” invece di “Table” perché ho incluso il package `babel` con l’opzione `italian`. Se avessi indicato l’opzione `czech` avrebbe scritto “Tabulka”...

LE FIGURE (AHI, AHI...)

I float servono anche (e forse soprattutto) a inserire delle immagini o dei grafici. L'environment si chiama

```
\begin{figure}...\end{figure}
```

ed ha una sintassi del tutto simile a quella di

```
\begin{table}...\end{table}
```

(solo che il comando `\caption` ora scriverà “Figura 1” invece di “Tabella 1”)

Per approfittare pienamente di questo environment occorre imparare ad inserire immagini e figure (e imparare a produrle).

Un esempio è il seguente

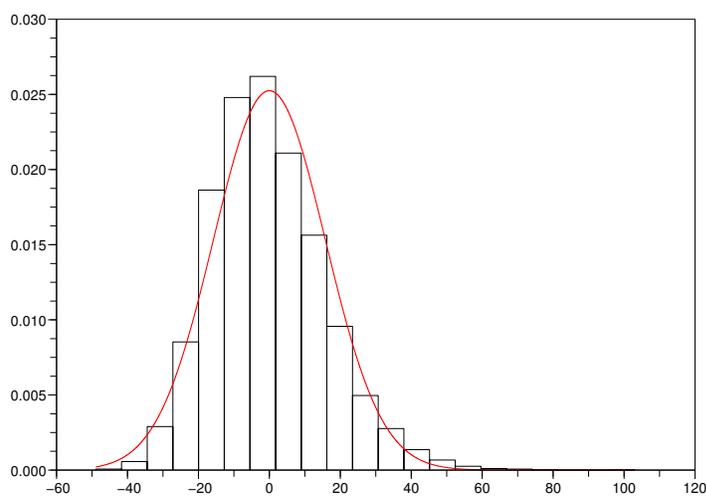


Figura 1.1: Histogram of kurtosis 8-11-14

Prima di vedere come si fa, due parole di avvertimento: i comandi di inserimento delle immagini *non* sono comandi T_EX: T_EX non processa le immagini, si limita a lasciare lo spazio necessario. Sono comandi T_EX che si limita a far passare al driver di stampa. Questo implica che si tratta di comandi meno portabili. Se cambiate di PC, il programma T_EX sarà sicuramente lo stesso, mentre il driver potrebbe essere diverso. Soprattutto se passate da un sistema all'altro (mac, linux. . .).

Occorre anche ricordare che qualche volta i previewer *non* riescono a far apparire sullo schermo l'immagine. In questo caso non bisogna disperare, probabilmente l'immagine c'è ed apparirà alla stampa. Per controllare impareremo a produrre un file postscript (o pdf), che potremo vedere sullo schermo.

Una altro dettaglio importante: quando si produce un file ps o pdf, il grafico o l'immagine sono contenute nel file prodotto. Grafico o immagine però *non* sono contenute nel file tex. Se inviate il file tex a un corrispondente (o lo portate dal PC di casa a quello dell'università) *attenzione* a non dimenticare di inviare o prendere *anche* i file delle immagini (una classica dimenticanza. . .)

Ciò detto, il package più pratico è `graphicx`, che, per cominciare, deve essere incluso nel vostro preambolo. Dopo di che le cose sono piuttosto semplici. La figura precedente è stata prodotta con

```
\begin{figure}[h]
\centering
\includegraphics[height=8cm]{ku8.eps}
\caption{Histogram of kurtosis 8-11-14}
\label{hist:ku-10}}
\end{figure}
```

Come si vede il comando principale è

```
\includegraphics[opzioni]{nome file}.
```

Il nome del file qui era `ku8.eps`, cioè era un file nel formato encapsulated postscript. Si possono inserire altri tipi di file (jpg, bmp, ...), ma i risultati sono talvolta inattesi. I file encapsulated postscript contengono delle informazioni sulle loro dimensioni che altri formati non hanno. Per includere una immagine, converrà prima trasformarla nel formato eps con un software adeguato. Ad esempio Questo si può fare, in assenza di un software specifico, nel



Figura 1.2: Lena

modo seguente.

- Si aggiunge al sistema una stampante postscript con l'indicazione che la stampa *vada fatta su un file*. Una stampante ps qualunque

va bene, ma a colori, se l'immagine è a colori. Ad esempio la Lexmark Optra Color 1200 PS.

- Poi si stampa l'immagine, chiamiamola lena.jpg ad esempio, con questa stampante. Diamo al file su cui si stampa il nome lena.ps (l'estensione ps non è comunque indispensabile).

- Si apre il file lena.ps con **ghostview**. Nel menu *file* di ghostview si sceglie *ps to eps*. Seguendo le indicazioni, si salva il file col nome lena.eps. Questo file può essere incluso come indicato prima.

Alternativamente, in internet è facile procurarsi il programma jpeg2ps.exe che traduce direttamente le immagini dal formato .jpg a .eps.

Vediamo ora alcune opzioni del comando `\includegraphics`.

- `\height 8cm` impone che la figura abbia un'altezza di 8cm. `\width 4cm` avrebbe avuto un effetto prevedibile, imponendo la larghezza. In questi casi l'immagine viene riscalata in maniera da non deformarla. Imporre sia una altezza che una larghezza invece produce una deformazione dell'immagine originaria. Ad esempio

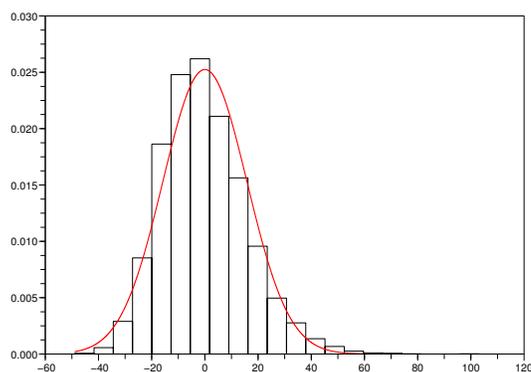


Figura 1.3: height=6cm

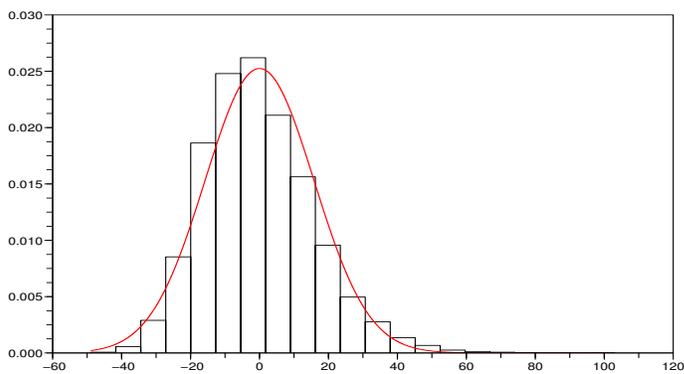


Figura 1.4: height=6cm,width=11cm

Altre opzioni utili

- **scale #1** ingrandisce l'immagine originale di un fattore **#1**. Ad esempio **scale 1.25** moltiplica le dimensioni dell'immagine per 1.25.
- **angle #1** ruotare (in senso antiorario) l'immagine di un angolo di **#1** (in gradi).

Finora abbiamo supposto di *disporre già* dell'immagine da inserire. È chiaro però che quest'ultima dovrà essere prodotta in qualche modo. Se non l'abbiamo scovata da qualche parte (su internet...?) o un amico gentile ce l'ha fornita, sarà necessario produrla con un software adatto.

Vediamo ora come si può produrre l'immagine seguente usando **scilab**. Le tappe che seguiremo si possono ripetere *mutatis mutandis* con qualunque altro software che sia il vostro preferito.

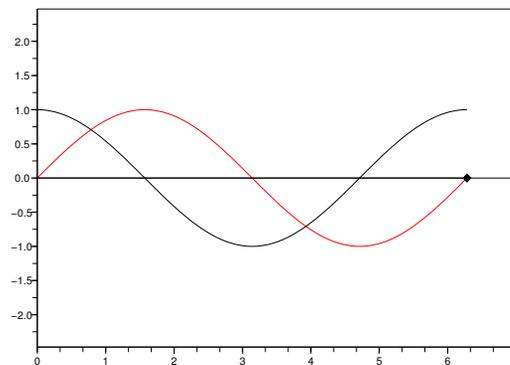


Figura 1.5: seno e coseno...

Occorrerà aprire scilab, poi dare i comandi che fanno produrre il grafico in questione. Ad esempio

```
-->tt=[0:.01:1]*2*%pi;yy1=sin(tt);yy2=cos(tt);
```

```
-->xbasc();plot2d(tt,yy1,frameflag=4);
```

```
-->plot2d(tt,yy2,frameflag=4);
```

```
-->plot2d(2*%pi,0,style=-4);plot2d([0 7],[0 0])
```

Fatto questo basta portarsi nella finestra grafica di scilab, aprire il

pop-up menu file e cliccare *export*. Scegliere *postscript* e *portrait*. Se volete produrre un file da stampare sceglierete l'opzione *Black and white*, se invece il vostro file verrà visionato su uno schermo potete scegliere *color*. Dopo di che dovete solo indicare il nome del file in cui il grafico verrà salvaguardato.

Esercizio 51. Produrre un grafico come il seguente

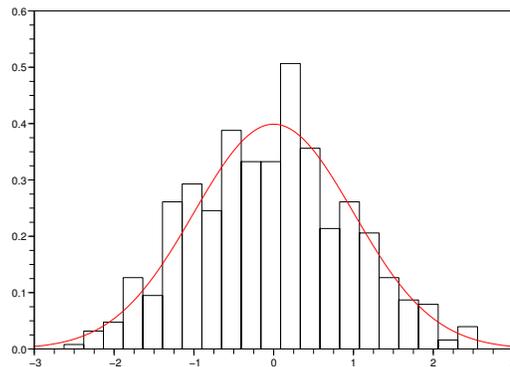


Figura 1.6: Istogramma di 512 numeri casuali con distribuzione $N(0,1)$. Questo generatore aleatorio fa pena...

Soluzione:

```
-->xx=rand(512,1,"n");xbasc();histplot(21,xx);  
  
-->tt=[-3:.01:3];yy=exp(-tt.^2/2)/sqrt(2*%pi);  
  
-->plot2d(tt,yy,style=5)
```

Esistono naturalmente moltissimi package che definiscono altri tipi di float ed altre funzioni di inclusione grafica. Ne vedremo solo due: il primo è il pacchetto `wrapfig` che permette di fare la cosa seguente.



Vediamo ora un altro esempio. L'immagine accanto mostra un esempio in cui l'uso di un processo di segmentazione migliora considerevolmente i risultati. In particolare i contorni risultano molto più netti, mentre senza la modellizzazione di un processo di contorni inevitabilmente porta ad un addolcimento delle transizioni ed a contorni inevitabilmente più sfumati.

La sintassi è molto semplice:

```
\begin{wrapfigure}[4]{1}[10pt]{3cm}
\includegraphics[height=2cm]{len.eps}
\end{wrapfigure}
Vediamo ora...
```

dove

- `[4]` indica che bisogna lasciare uno spazio per la figura pari a 4 righe.
- `[1]` indica che la figura deve stare a sinistra. Altri valori possibili di questo parametro sono `[r]`, `[i]`, `[o]`, che indicano rispettivamente che la figura deve stare a destra, al centro o all'esterno.
- `[10pt]` indica che la figura può sporgere nel margine della pagina per 10pt.
- `{3cm}` indica che bisogna lasciare spazio per una larghezza di 3 cm per la figura.

Il secondo package che può essere utile è **subfigure**, che permette, ad esempio, di appaiare più figure, ciascuna con la sua didascalia (più eventualmente una didascalia per l'insieme). Un esempio è

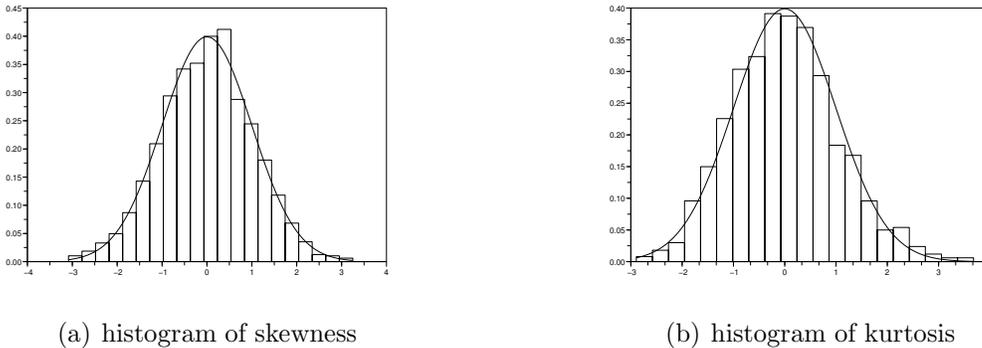


Figura 1.7: Skewness e kurtosi a confronto

```

\begin{figure}[htbp]
\centering%
\subfigure[histogram of skewness\label
{fig:sk10b.eps}]%
{\includegraphics[height=4cm]{sk10b.eps}}
\quad\quad
\subfigure[histogram of kurtosis
\label{fig:ku10b.eps}]%
{\includegraphics[height=4cm]{ku10b.eps}}
\caption{Skewness e kurtosi a confronto
\label{fig:sottofigure}}
\end{figure}

```

Esercizio 52. Come si fa a produrre la formula della pagina seguente?

$$\sum_{k=0}^n \binom{n}{k}^2 = \binom{2n}{n}$$

Come si fa vedere?

INDICI...

In un documento possono comparire molti tipi di indici.

1. l'indice generale (table of contents)
2. l'indice delle figure
3. l'indice delle tabelle
4. l'indice analitico

I primi tre sono molto facili da produrre: per l'indice generale basta inserire nel documento il comando `\tableofcontents`. Questo comando deve essere messo là dove si vuole che l'indice compaia (di solito all'inizio o alla fine del documento). Per gli altri due la regola è la stessa, i comandi sono rispettivamente `\listoffigures` e `\listoftables`. Questi indici vengono prodotti automaticamente da L^AT_EX, che raccoglie automaticamente le informazioni dai comandi `\section`, `\section` e altri.

Nell'indice generale compariranno automaticamente i capitoli e i paragrafi con i numeri di pagina in cui si trovano nel documento. Ricordatevi che se aggiungete del materiale (un nuovo paragrafo per esempio) dopo la prima compilazione le numerazioni saranno quelle del testo vecchio e quindi sbagliate. Come già detto in precedenza quando si aggiunge materiale che cambia le numerazioni progressive, occorre sempre compilare due volte.

Tutto ciò è molto semplice, a meno che naturalmente non si voglia una presentazione personalizzata. Eviteremo di entrare nel dettaglio di queste personalizzazioni e dei package che si possono usare a questo scopo. Osserviamo solo che se si vuole aggiungere all'indice

un elemento in più di quelli che già compaiono per default, si usa il comando

```
\contentsline{tipo}{testo}{n.pagina}
```

Ad esempio `\contentsline{chapter}{Prefazione}{2}` aggiungerà all'indice una riga con scritto "Prefazione", il numero di pagina 2 nello stile in cui vengono indicati nell'indice i capitoli.

Più complessa è la generazione dell'indice analitico, anche se all'atto pratico si tratta di un'operazione molto rapida. Per esperienza, la generazione dell'indice analitico, anche di grossi documenti è un'operazione di poche ore, anche se è meglio generarlo man mano che la preparazione del documento progredisce. Tenuto conto della grande aiuto che un indice analitico fornisce all'utilizzatore, il mio consiglio è di non esitare a produrlo. Il relatore o controrelatore che trova facilmente nella tesi le cose che cerca viene messo di solito di buon umore. . .

Distinguiamo due fasi

1. come raccogliere il materiale che andrà nell'indice;
2. come produrre l'indice.

Per la prima questione il comando è `\index{...}`. Ad esempio, inserire nel testo il comando `\index{Banach-Steinhaus, theorem}` farà comparire nell'indice analitico l'indicazione “Banach-Steinhaus, theorem” con accanto il numero di pagina in corrispondenza della quale il comando `\index{...}` è stato inserito.

Questo comando può essere usato in maniera più raffinata. Ad esempio i comandi `\index{Theorem!closed graph}` e `\index{Theorem!Banach-Steinhaus}` faranno apparire delle voci come in un capitoletto intitolato “Theorem” con la lista dei nomi dei teoremi. È possibile anche raggruppare gli elementi fino a tre livelli: ad esempio `\index{animali!mammiferi!felini}`.

Una volta inseriti i comandi `\index{...}`, per generare l'indice analitico occorre

1. aggiungere `\usepackage{makeidx}` e il comando `\makeindex` nel preambolo

2. aggiungere il comando `\printindex` nel punto del testo in cui si vuole che l'indice analitico compaia (di solito alla fine del documento).

Ma non è finita. Occorre compilare il documento e poi eseguire il programma *makeindex*. Si tratta di un programma esterno che, come per *bibtex* può essere chiamato da una finestra dos oppure da *winedit* (vedi in alto verso destra nel menu di *winedit*). Ciò fatto occorre poi compilare \LaTeX due volte. E l'indice analitico comparirà miracolosamente...

La classe “articolo”

Nella classe articolo L^AT_EX prevede dei comandi appositi per l’inserimento del nome degli autori, del titolo, dello abstract. . .

Naturalmente all’inizio del preambolo si dovrà indicare

```
\documentclass[10pt]{article}
```

([10pt] o [12pt] è opzionale).

Un esempio si trova nella pagina seguente. Nella seguente ancora si trova un’indicazione dei comandi che occorre inserire per ottenere il risultato indicato.

EXPLICIT COMPUTATION OF SECOND ORDER MOMENTS OF IMPORTANCE SAMPLING ESTIMATORS FOR FRACTIONAL BROWNIAN MOTION

PAOLO BALDI and BARBARA PACCHIAROTTI
Università di Roma Tor Vergata

Abstract

We study a family of importance sampling estimators of the probability of level crossing when the crossing level is large or the intensity of the noise is small. We develop a method which gives explicitly the asymptotics of the second order moment. Some of the results apply to fractional Brownian Motion, some of them are more general. The main tools are some refined versions of classical large deviations results.

AMS 1991 subject classifications. Primary 60F10; secondary 60J60, 60G15, 65C50.

Key words and phrases. Importance Sampling; Large Deviations; Ruin Probabilities; Gaussian Processes.

Acknowledgement The authors wish to thank Th.Jeulin for some illuminating remarks.

1 Introduction

In this paper we address the problem of computing the asymptotics of the second-order moment of a class of importance sampling estimators arising naturally when dealing with gaussian processes. We consider two questions. Let $X = (X_t)_{t \geq 0}$ be a fractional Brownian motion. A classical problem in risk theory is the investigation of the ruin probability

$$P\left(\sup_{t>0}(X_t - \varphi_t) > B\right). \quad (1.1)$$

Under suitable assumptions on the function φ (typically $\varphi_t \rightarrow +\infty$ as $t \rightarrow +\infty$) this probability is very small and its computation by a naive simulation requires a large number of iterations in order to get a reasonable precision. A natural technique is then *importance sampling*, that is the simulation of the process under a new probability Q , for which the event considered is not rare and to compensate by taking into account the density of P with respect to Q .

A closely related question is the computation, as $\varepsilon \rightarrow 0$, of the probability

$$p_\varepsilon \stackrel{def}{=} P\left(\sup_{0<t \leq 1}(\varepsilon X_t - \varphi_t) > 1\right). \quad (1.2)$$

More precisely we consider, for the computation of the level crossing probability (1.2), the class of admissible importance sampling estimators of the form

$$1_{\{\tau_\varepsilon \leq 1\}} Z_{\tau_\varepsilon} \quad (1.3)$$

where the process X is simulated according to a distribution Q_ε , obtained by translating P with a path $\frac{1}{\varepsilon}\gamma$, γ belonging to the Reproducing Kernel Hilbert Space of X . Here τ_ε is the time at which level 1 is

\title{\sc Explicit Computation of Second Order Moments of
Importance Sampling Estimators
for fractional Brownian motion}

\maketitle

\author{ {\sc Paolo Baldi}

\ \ and \ \ {\sc Barbara Pacchiarotti}}\ \

{\it Universit\ 'a di Roma Tor Vergata}

\begin{abstract}

We study a family of...

\end{abstract}

\bigskip

{\sl AMS 1991 subject classifications\rm.

Primary 60F10; secondary 60J60,

60G15, 65C50.}

\medskip

\sl Key words and phrases. Importance Sampling; Large
Deviations; Ruin Probabilities; Gaussian Processes.\rm

\bigskip

\noindent{\bf Acknowledgement} The authors wish to
thank Th.Jeulin for some illuminating remarks.

\section{Introduction}\label{intro}

In this paper we address...

Macro

L'utilizzatore può definire dei nuovi comandi (delle macro) mediante il comando `\newcommand`. Ad esempio

```
\newcommand{\ag}{\`}
```

definisce un nuovo comando `\ag` che semplicemente sostituisce l'accento grave (utile per chi non ha una tastiera americana). Altri esempi

```
\newcommand{\succ}{\{x_1,\dots,x_n\}}
```

Si può anche pensare di usare una macro per evitare di dover scrivere una frase lunga. Per esempio

```
\newcommand{\PR}{\$(\Omega,{\mathcal F},P)\$}
```

oppure

```
\newcommand{\TF}{trasformata di Fourier}
```

quest'ultima però però ha qualche inconveniente. Quale?

Quando si definiscono nuovi comandi occorre ricordare che

- tutti i comandi $\text{T}_{\text{E}}\text{X}$ cominciano con lo antislash `\`.
- $\text{T}_{\text{E}}\text{X}$ ignora gli spazi alla fine di un comando.
- Un comando $\text{T}_{\text{E}}\text{X}$ è composto da *sole lettere*. Quindi niente numeri e niente simboli come `@`, `#`, `&`,...

La possibilità di definire comandi nuovi diventa più interessante quando si tratta di comandi più complessi. Intanto la macro può avere degli argomenti: provate a definire

```
\newcommand{\succe}[2]{\{\{#1\}_1,\dots,\{#1\}_{#2}\}}
```

e poi a scrivere $\$ \backslash \text{succes}\{x\}\{n\} \$$ e $\$ \backslash \text{succes}\{y\}\{m\} \$$.

La sintassi completa è

```
\newcommand{\nomemacro}[numero di argomenti]
{la definizione (che dipende da #1,...,#n)}
```

Gli strumenti che abbiamo visto (scatole, loro manipolazioni, spaziature positive e negative...) permettono anche di combinare simboli tra di loro per produrne di nuovi. Un comando utile in questo senso è `\overset` (occorre `amsmath`). `\overset{simb1}{simb2}` produce un simbolo formato dai due simboli `simb1` e `simb2`, il primo posto al di sopra del secondo. Di solito però bisogna poi lavorarci un po': questo metodo produce degli spazi spuri.

Esercizio 53. In probabilità si usa indicare con \perp l'indipendenza di due variabili aleatorie. Sapreste costruire questo simbolo?

Esercizio 54. La parte interna dell'insieme B si denota ponendo un circoletto al di sopra del nome, B , dell'insieme. Ciò si potrebbe ottenere mediante `\overset{\circ}{B}`. Ma il risultato è $\overset{\circ}{B}$, dove il tondino volteggia un po' troppo in alto (ed è un po' spostato a sinistra). Sapreste aggiustare le cose e produrre invece $\overset{\circ}{B}$, $\overset{\circ}{A}$?
Definite poi un comando, chiamato `\interior`, con un argomento, tale che, ad esempio, `\interior{C}` produca $\overset{\circ}{C}$.

Esercizio 55. Sapreste produrre \circ ?

Esercizio 56. Sapreste produrre

Ⓜ ① ② ?

Esercizio 57. Definire una macro che

- vada a capo e lasci una riga bianca

• senza indentazione scriva in grassetto **Esercizio** seguito da un numero progressivo.

(sarà opportuno andare a rivedere i comandi per la gestione dei contatori 44.

Esercizio 58. Sapreste definire un comando che manda a capo e poi inizia un paragrafo con un pallino (`\bullet`)? E un'altro che fa la stessa cosa ma invece del pallino mette un numero progressivo in grassetto?

È bene però ricordare che, se si desidera produrre una macro complessa, che realizzi delle cose particolari, piuttosto che mettersi a programmare una macro, conviene andare a vedere se esiste già una macro che fa la cosa che si vuole.

A questo scopo è utile consultare il package browser (start→Miktex 2.7→Browse packages).

Ad esempio se si devono disegnare degli Young tableaux, cercando con le capacità di ricerca del package browser si trova che esiste il package **youngtab** che ha questa capacità. Oppure se si devono realizzare dei diagrammi commutativi, si trova che esistono vari packages specializzati a questo scopo. Per scoprire come funzionano e quali comandi bisogna dare per farli funzionare, è opportuno sapere che le documentazioni dei packages si trovano in program files→ Miktex 2.7→miktex→doc (buona fortuna. . .).

L^AT_EX per lo web

Per preparare un testo che debba poi essere letto su uno schermo ci sono due opzioni utili: il colore e i collegamenti ipertestuali. Entrambi queste caratteristiche si gestiscono facilmente, grazie ad alcuni package appropriati.

Per il colore una possibilità consiste nell'aggiungere il package `color` alla lista del preambolo (altri sono possibili). Dopo di che si devono definire i colori che si vogliono usare mediante uno dei comandi

```
\definecolor{nome colore}{modo}{componenti}
```

Il modo è uno tra `rgb` (red-green-blue) oppure `cmymk` (cyan-magenta-yellow-black). Sono due modi diversi di definizione del colore.

Ad esempio

```
\definecolor{blue}{rgb}{0,0,1}
```

Una volta definito il colore il comando `\color{blue}` produce il testo che segue in blu (bisognerà usare delle parentesi graffe se si vuole in blu solo una parte del testo). Invece

```
\definecolor{rossomattone}{cmymk}{0,0.51,1,0}
```

Produce un colore così. Si possono anche definire dei livelli di grigio:

```
\definecolor{pocogrigio}{gray}{0.35}
```

produce col comando `\color{pocogrigio}` un testo in grigio. La gradazione va da 1 (bianco) a 0 (nero). Possono essere utili i comandi `\colorbox` e `\fcolorbox`: digitando

`\colorbox[cmyk]{0,0,0.6,0}{Attenzione quarantena!}` si
ottiene Attenzione quarantena!
mentre con `\fcolorbox` si sarebbe ottenuta la stessa scatola con
un bordo.

Infine il package **hyperref** semplicemente aggiunge dei collegamenti ipertestuali a tutti gli elementi numerati. Per usare di queste capacità occorre aggiungere al preambolo questo package. Si devono poi produrre i file .dvi e .pdf. In quest'ultimo nel testo, in corrispondenza di cose come “come dice l'equazione (3.2)” il numero dell'equazione è associato a un collegamento ipertestuale che porta al punto in cui l'equazione (3.2) effettivamente si trova. Idem per le numerazioni di teoremi, e paragrafi. Ci sono delle opzioni: ad esempio con

```
\hypersetup{backref,linkcolor=blue}
```

viene specificato che gli elementi ipertestuali debbano essere indicati in blu (invece che in rosso, che sarebbe il default) e che si vogliono collegamenti ipertestuali anche per le citazioni bibliografiche.

PICTEX

Pictex è un altro package, che serve per la produzione di grafici o altre figure geometriche. Il vantaggio è che non usa il metodo della inclusione di figure (che, come abbiamo visto, presenta qualche inconveniente) e che permette un controllo dell'apparenza finale abbastanza diretto. Quindi niente file esterni da includere. Vediamo subito un esempio: Questa figura è stata realizzata con i

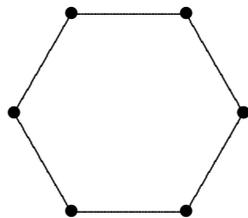


Figura 1.8: Esagono...

comandi:

```
\begin{figure}[h]
\hbox to\hsize\bgroup\hss
\beginpicture
\setcoordinatesystem units <.6truein,.6truein>
\setplotarea x from -1.2 to 1.2, y from -1 to 1.45
\plot 1 0 .5 .866 -.5 .866 -1 0 -.5 -.866
.5 -.866 1 0 /
\multiput {$\bullet$} at 1 0 .5 .866
-.5 .866 -1 0 -.5 -.866 .5 -.866 1 0 /
\endpicture
\hss\egroup
\caption{Esagono\dots}
\end{figure}
```

che ora vediamo di analizzare. Ricordiamo comunque che è necessario aggiungere `pictex` alla lista dei package del preambolo.

I comandi `\beginpicture` e `\endpicture` indicano l'inizio e la fine dei comandi di ambiente `Pictex`.

```
\setcoordinatesystem units <.6truein,.6truein>
```

indica le unità di misura degli assi x e y . Modificando questi parametri il grafico può essere ingrandito o rimpicciolito o anche deformato (se le due unità di misura vengono cambiate in maniera diversa).

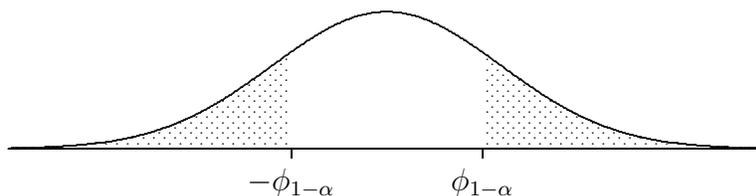
```
\setplotarea x from -1.2 to 1.2, y from -1 to 1.45
```

Indica la porzione del piano in cui il grafico verrà disegnato. se vengono indicati degli assi coordinati, questi vengono disegnati automaticamente per tutta l'ampiezza dell'area indicata in `\setplotarea`

Il comando `\plot .../` congiunge con un segmento i punti di coordinate ... (/ indica la fine della lista dei punti).

`\multiput {\bullet} at .../` pone il simbolo `{\bullet}` nei punti di coordinate

Un altro esempio:



```

\begin{figure}
\hbox to\hsize\bgroup\hss
\beginpicture
\setcoordinatesystem units <.6truein,1.8truein>
\setplotarea x from -3.3 to 3.3, y from 0 to .42
\axis bottom ticks short withvalues {\$-\phi_{1-\alpha}\$}
{\phi_{1-\alpha}\$} / at -0.83333 0.83333 / /
\plot
...
/
\setquadratic
\setshadesymbol <z,z,z,z> ({\fiverm .})
\setshadegrid span <2pt>
\vshade
-3.00000 0 0.004432
-2.75000 0 0.00909
-2.50000 0 0.01753
-2.00000 0 0.053991
-1.75000 0 0.086277
-1.50000 0 0.129518
-1.25000 0 0.182649
-1.00000 0 0.241971
-0.83333 0 0.281912

```

```

% -0.66667 0 0.319448
% -0.50000 0 0.352065
/
\vshade
0.83333 0 0.281912
1.00000 0 0.241971
1.25000 0 0.182649
1.50000 0 0.129518
1.75000 0 0.086277
2.00000 0 0.053991
2.50000 0 0.01753
2.75000 0 0.00909
3.00000 0 0.004432
% -0.66667 0 0.319448
% -0.50000 0 0.352065
/
\endpicture
\hss\egroup
\end{figure}

```

Da notare qui

- il comando `\axis` per il disegno degli assi coordinati (qui uno solo)
- l'opzione `\setquadratic`: in questo modo `pictex` effettua una interpolazione quadratica dei punti che gli vengono dati da plottare. Tranne che per definire delle aree da ombreggiare meglio evitare perché rallenta l'esecuzione e poi si tratta sempre di un'approssimazione difficile da valutare. Ad ogni modo richiede un numero dispari di punti.

Altri comandi utili per la realizzazione di un grafico: lo stile si gestisce mediante

```
\setdots [<2pt>]  
\setdashes [<2pt>]  
\setsolid
```

Il primo comando traccia le curve a puntini, il secondo a trattini, il terzo serve per tornare a tratto continuo. Il parametro opzionale `<2pt>` sta a indicare la distanza tra i puntini oppure tra i trattini (la scelta di 2 è puramente a titolo di esempio. Se non si precisa l'opzione il programma sceglie `<5pt>` (che è forse un po' troppo).

Esercizio 59. Realizzare i grafici

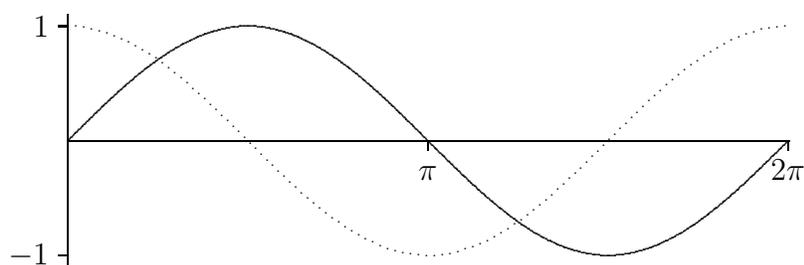


Figura 1.9: I grafici delle funzioni seno (tratto continuo) e coseno (a puntini).

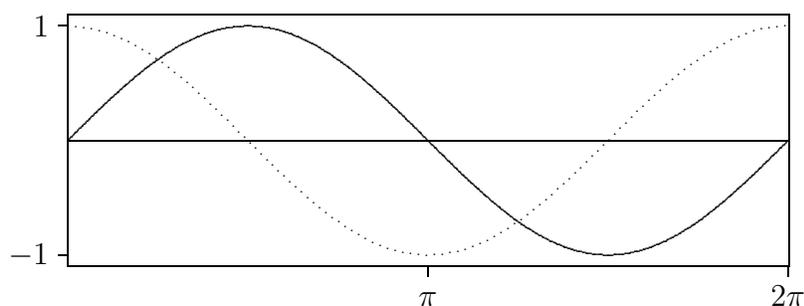


Figura 1.10: I grafici delle funzioni seno (tratto continuo) e coseno (a puntini).

Vi sono anche comandi per la realizzazione di istogrammi. Provate a vedere che figura produce il comando

```
\begin{figure}[h]
\hbox to\hsize\bgroup\hss
\beginpicture
\setcoordinatesystem units <.12cm,.092cm>
\setplotarea x from 0 to 80, y from 0 to 40
\axis left shiftedto x=-3 ticks short withvalues
{$0$} {$10$} {$20$} {$30$} {$40$} /
at 0 10 20 30 40 / /
\axis bottom ticks short withvalues
{$0$} {$10$} {$20$} {$40$} {$80$} /
at 0 10 20 40 80 / /
\sethistograms
\plot 0 0 10 15 20 35 40 15 80 5 /
\endpicture
\hss\egroup
\caption{Numero di sigarette fumate al giorno.}
\end{figure}
\end{figure}
```

Esercizio 60. Combinando L^AT_EX e scilab generate a caso 512 numeri casuali con distribuzione $\chi^2(3)$ e fatene l'istogramma.

A questo scopo ricordate che:

- il comando per generare a caso in scilab dei numeri con distribuzione qualunque è **grand** (date uno sguardo alla sintassi di questo comando). Più precisamente per generare dei numeri con distribuzione χ^2 è **Y=grand(m,n,'chi',dg)**, che produce una matrice

$m \times n$ di numeri indipendenti con distribuzione χ^2 e **gd** gradi di libertà.

• Per procurarsi i dati per fare l'istogramma il comando è **dsearch**: se mettete i valori $0, 1, \dots, 8$ nel vettore **val**, e indichiamo con **xx** il vettore contenente i nostri 512 numeri simulati, allora con il comando

```
[ind, occ, out] = dsearch(xx, val)
```

si ottiene, nel vettore **occ** il numero di punti di **xx** che cadono negli intervalli $[0, 1],]1, 2], \dots,]7, 8]$. Andate a vedere lo help del comando se volete sapere il significato di **ind** e di **out**. È previsto un comando per disegnare un arco di cerchio in **\pictex**. Ad esempio

```
\circulararc 30 degrees from 2.5 2 center at 1 0
```

produce il disegno di un arco di cerchio di 30° , con centro in $(1, 0)$ ed a partire dal punto $(2.5, 2)$.

Esercizio 61. Riprendere l'esempio dell'esagono ed aggiungere al grafico un cerchio circoscritto.

Esercizio 62. Disegnare il grafico della gaussiana:

$$f(x) = \frac{1}{2\pi} e^{-x^2/2}$$

sovrapponendovi quella di una densità di Laplace

$$g(x) = \frac{1}{2} e^{-|x|}$$

a trattini

PSTRICKS

pstricks è un package di realizzazione grafica che usa, in associazione con $\text{T}_{\text{E}}\text{X}$, le potenzialità grafiche del linguaggio postscript. Per servirsene occorre aggiungerlo alla lista dei packages: se si vogliono utilizzare tutte le sue potenzialità bisogna aggiungere nel preambolo il comando `\usepackage{pst-all}`.

Grafici e immagini costruite con **pstricks** si trovano definite all'interno del file sorgente, e non sono quindi importati dall'esterno. Talora però i previewer, come **yap** che viene usato in distribuzioni vecchie di **MiKTeX**, non sono in grado di far vedere il risultato finale. Per vederlo correttamente occorrerà convertirlo prima in **postscript** (vedi il comando in alto verso destra del menu di **winedit**) e poi vederlo con **gsview**.

Questo problema non si presenta con la versione 2.8 di **MiKTeX**. Tuttavia, c'è un altro problema, indipendente dal previewer; non si può compilare la sorgente con **PDFLaTeX** per produrre un file **pdf**. Ciò rende impossibile usare l'editore integrato in **MiKTeX 2.8**, **TeXworks**, che usa **PDFLaTeX** per visualizzare il testo.

Se si vuole produrre un file **pdf**, occorrerà prima produrre un file **ps** e poi convertirlo nel formato **pdf**. Se si usa Windows, si deve aprire una finestra DOS (programma “prompt dei comandi” nella cartella “Accessori” dell'elenco dei programmi, nel menu “Avvio”), quindi spostarsi nella cartella che contiene il file sorgente e dare i comandi seguenti, dove “source” indica il nome della sorgente:

```
latex source.tex
dvips source.dvi
ps2pdf source.ps
```

Vediamo alcuni comandi alla rinfusa, per cominciare. Osservate che spesso i comandi `pstricks` sono della forma `\ps...`. Svilupperemo solo una minima parte delle potenzialità di `pstricks`. Per maggiori informazioni consultare la documentazione che si trova nella cartella `C:\Programmi\MiKTeX 2.8\doc\generic\pstricks` in particolare il file `pst-user.pdf`. Può anche essere utile, dopo che si è compreso il funzionamento di `pstricks`, stamparsi la guida rapida `pst-quickref.pdf`, reperibile nella stessa cartella.

Scatole incorniciate

```
\psframebox[fillcolor=gray,fillstyle=solid]
{materiale vario}
```

crea una scatola contenente il `materiale vario` e con uno sfondo grigio. Il colore si controlla con il parametro `fillcolor`. Attenzione perché per default il parametro `fillcolor` è posto uguale a “none”, e dunque la scatola non viene colorata.

Altri parametri per `\psframebox` sono

`framearc=num`; ad esempio `framearc=.2` definisce un parametro di arrotondamento degli angoli della scatola.

`linewidth=dim`; ad esempio `linewidth=2pt` indica lo spessore del bordo della scatola (`=0pt` per default).

Esercizio 63. Provate a modificare la scatola di sopra aggiungendo le opzioni `framearc=.1,linewidth=1pt`

Esercizio 64. Realizzare

Quando lo studente dice `f è continua e quindi derivabile` il professore vede rosso...

Esercizio 65. Realizzare

Capitolo 1

Comandi simili a `\psframebox` sono `\pscirclebox` (scatole di forma circolare) `\psovalbox` (di forma ovale). Provate a ripetere l'Esercizio 64 usando `\psshadowbox` invece di `\psframebox`...

Colori

I colori usuali sono predefiniti (in inglese): `red`, `green`, `yellow`,... , ma se ne possono definire di nuovi.

Per le sfumature di grigio (le stampanti laser a colori sono ancora rare):

```
\newgray{lightlightgray}{.85}%
```

Un parametro uguale a 0 vuole dire *nero*, uguale a 1 *bianco*. Ad esempio questo è l'effetto

Esercizio 66. Realizzare e confrontare il colore definito sopra con il colore `lightgray`, che è predefinito.

materiale vario

e

materiale vario

Per definire dei nuovi colori “veri” ci sono due modi **rgb** (rosso+verde+blu) e **cmyk** (ciano+magenta+giallo+nero). Veramente sarebbe complicato entrare nella spiegazione delle differenze tra

questi due metodi, che sono molto diversi (uno è a sottrazione di colore, l'altro a sovrapposizione). Ad ogni modo è abbastanza complicato inventarsi un nuovo colore. Con il secondo metodo il comando è, ad esempio,

```
\newcmykcolor{rossoarancio}{0 0.77 0.87 0}
```

si ottiene , un po' diverso dal . Per questi esperimenti è però meglio disporre di una tavola di colori, con le ricette.

Esercizio 67. Provate gli effetti dei colori Navyblue e Rawsiena (vedi il file `cmyk.pdf`).

Oggetti grafici

Come si vede dagli esempi precedenti, le scatole incorniciate hanno le dimensioni del testo in esso contenuto e sono posizionate nel punto in cui appare l'istruzione. Ciò non succede invece per gli oggetti grafici, di cui stiamo per descrivere alcuni esempi; in tal caso, infatti, ogni comando produce una scatola di dimensioni nulle e non provoca uno spostamento del *punto corrente*, quello in cui appare l'istruzione. Ne consegue che la successiva istruzione grafica farà riferimento allo stesso punto corrente. La posizione degli oggetti grafici rispetto al punto corrente è determinata dai parametri di posizione, nel modo che vedremo.

Il vantaggio di questa procedura è che più oggetti grafici possono essere posizionati sulla pagina immaginando che nel punto corrente ci sia l'origine del sistema di coordinate. Naturalmente, ad evitare

che il disegno si sovrapponga al testo (se lo si vuole evitare) bisogna creare opportunamente lo spazio che lo contenga, usando i comandi usuali del TeX.

Supponiamo, per esempio, di volere disegnare due cerchi di raggio 2 (nell'unità di misura predefinita, eguale a 1 cm), i cui centri distano 4 cm in direzione orizzontale, al centro della pagina, sotto l'ultima riga di testo. Il comando per disegnare un cerchio di raggio r con centro nel punto (x, y) è `\pscircle[opzioni](x,y){r}`. Dovremo quindi scrivere un insieme di istruzioni tipo:

```
\vspace{2cm}
\begin{center}
\pscircle[linewidth=2pt,fillcolor=yellow,
fillstyle=solid](-2,1){1}
\pscircle[linewidth=2pt,fillcolor=yellow,
fillstyle=solid](2,1){1}
\end{center}
```



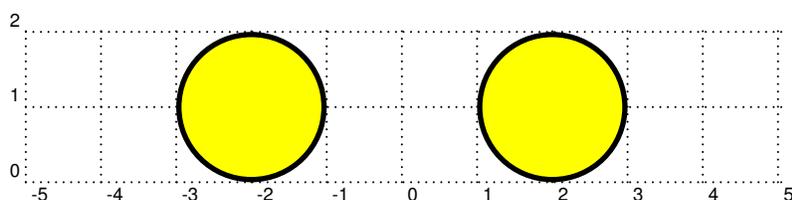
Infatti, le regole con cui il TeX definisce il punto corrente fanno sì che esso (quindi l'origine delle coordinate) venga posizionato, in questo esempio, un pò più di 2 cm sotto l'ultima riga di testo, al centro della pagina. Esiste comunque un metodo molto semplice di conoscere quale è esattamente il punto corrente, anzi un'intera griglia di punti nello spazio delle coordinate, usando il comando

```
\psgrid[subgriddiv=1,griddots=10,gridlabels=7pt]
(x1,y1)(x2,y2)
```

che crea una griglia con due vertici nei punti $(x1, y1)$ (punto in basso a sinistra) e $(x2, y2)$ (punto in alto a destra); questo comando ha dei parametri opzionali, che sono descritti nel manuale, ma il cui significato è chiarito dall'esempio seguente. Aggiungiamo alle istruzioni precedenti, dopo il comando `\begin{center}`, il comando

```
\psgrid[subgriddiv=1,griddots=10,gridlabels=7pt]
(-5,0)(5,2)
```

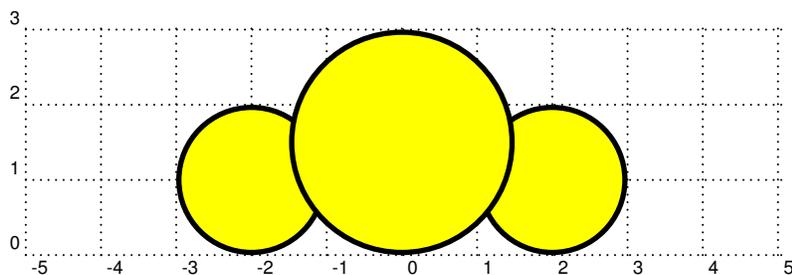
Otteniamo il disegno seguente.



La griglia è ovviamente uno strumento utile nella fase di costruzione delle figure, anche se essa non ne deve fare parte; la si può infatti inserire come guida all'inizio e cancellarla dopo avere ottenuto il risultato desiderato.

Esercizio 68. Cercate di realizzare

Inizio esempio



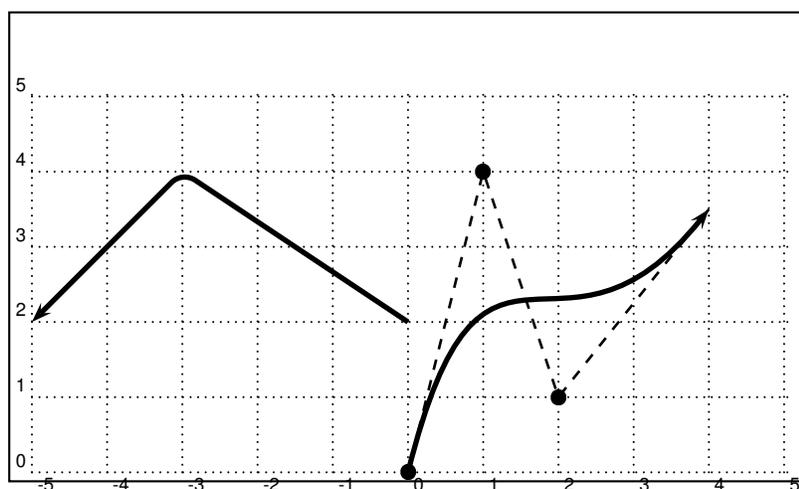
Fine esempio

Il disegno può essere inserito anche in una scatola di dimensioni prefissate, utilizzando l'ambiente `minipage`. Si consideri l'esempio seguente, che utilizza due altri comandi grafici, `\psbezier` e `\psline`. Il primo disegna una curva di Bezier associata a 4 punti, anch'essi disegnati a causa dell'opzione `showpoints=true`; il secondo disegna una linea spezzata con gli spigoli arrotondati. Ambedue i comandi disegnano una freccia alla fine della curva.

```

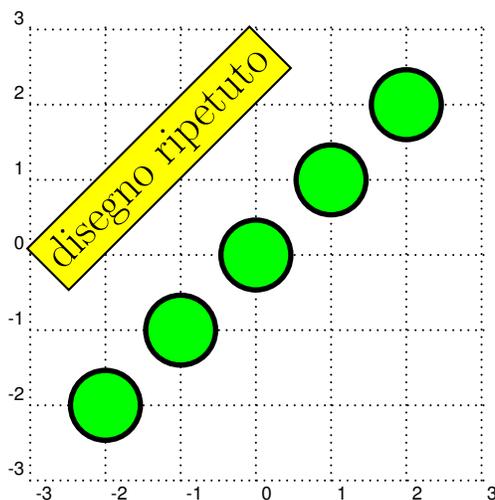
\psframebox{
\begin{minipage}[b][6cm]{10cm}
\begin{center}
\psgrid[subgriddiv=1,griddots=10,gridlabels=7pt]
(-5,0)(5,5)
\psbezier[linewidth=2pt,showpoints=true]
{->}(0,0)(1,4)(2,1)(4,3.5)
\psline[linewidth=2pt,linearc=.25]
{->}(0,2)(-3,4)(-5,2)
\end{center}
\end{minipage}

```



La grandezza della scatola che contiene la figura ed il sistema di coordinate cui fanno riferimento gli oggetti grafici possono essere fissati in modo più diretto usando l'ambiente `pspicture`. Si consideri l'esempio seguente, che introduce anche alcuni altri comandi grafici:

```
\begin{center}
\begin{pspicture}(-3,-3)(3,3)
\def\cerchio{\pscircle[linewidth=2pt,
fillcolor=green,fillstyle=solid](0,0){.5}}
\psgrid[subgriddiv=1,griddots=10,gridlabels=7pt]
(-3,-3)(3,3)
\multips(-2,-2)(1,1){5}{\cerchio}
\rput[b]{45}(-1,1){\psframebox[fillcolor=yellow,
fillstyle=solid]{disegno ripetuto}}
\end{pspicture}
\end{center}
```



In questo esempio si precisa che il materiale grafico è riferito alle coordinate $(-3, 3)$ (vertice del rettangolo in basso a sinistra e $(3, 3)$ (vertice in alto a destra). Il comando

`\multips(x0,y0)(dx,dy){n}{...}`

ripete n volte gli oggetti grafici descritti in $\{...\}$, posizionando il primo gruppo con l'origine nel punto (x_0, y_0) e gli altri traslati di (dx, dy) rispetto al precedente. Il comando

`\rput[pos]{ang}(x,y){testo}`

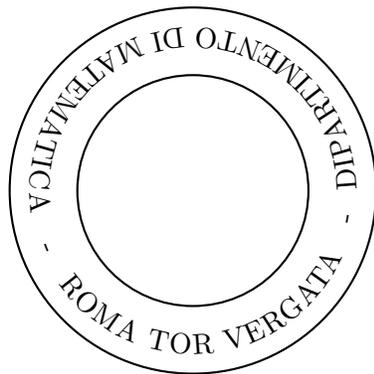
posiziona la scatola contenente **testo**, ruotata di un angolo **ang**, in modo che un suo punto, determinato dal parametro **pos**, si trovi nel punto (x, y) . Nell'esempio, il valore **b** di **pos** implica che il punto $(x, y) = (-1, 1)$ corrisponde al centro della linea di fondo della scatola.

Ci sono molti altri comandi che definiscono delle curve, Per esempio, c'è il comando

`\pstextpath{curva}{testo}`

che scrive **testo** lungo il percorso della curva

Esercizio 69. Combinare i comandi `\pscircle` `\pstextpath` per realizzare



I nodi

Per costruire delle figure geometriche è possibile assegnare a degli oggetti grafici quello che si chiama un *nodo*. Questo permette poi di disegnare segmenti (o altre curve) che collegano questi nodi in maniera semplice. Vediamo un esempio. Il comando

```
\rput(x,y){\circlenode*[opzioni] {nomenodo}{testo}}
```

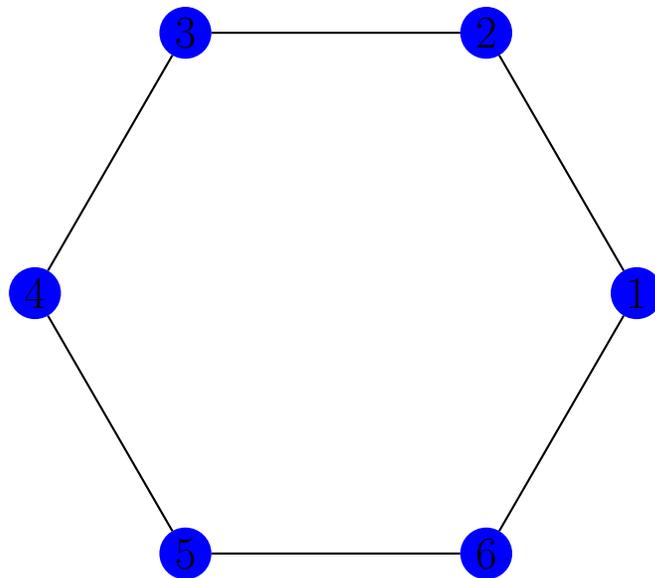
definisce un nodo di forma circolare, gli dà un nome, lo pone alle coordinate (x, y) . Ci sono molti altri tipi di nodi possibili.

L'oggetto grafico a cui il nodo fa riferimento sarà un cerchietto contenente *testo*. Una volta definiti dei nodi, li si può connettere mediante delle curve più o meno complesse. La più semplice è naturalmente

```
\ncline[opzioni]{...}{nomenodo1}{nomenodo2}
```

che collega i due nodi con un segmento. Qui ... indica il tipo di segmento: <-> ad esempio aggiungerebbe delle frecce.

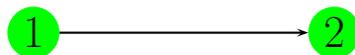
Ad esempio



si fa con

```
\begin{center}
\begin{pspicture}(0,0)(8,-8)
\rput(8,-4){\circlenode*[fillcolor=blue,
fillstyle=solid]{A1}{1}}
\rput(6.,-0.5358984){\circlenode*[fillcolor=blue,
fillstyle=solid]{A2}{2}}
\rput(2.,-0.5358984){\circlenode*[fillcolor=blue,
fillstyle=solid]{A3}{3}}
\rput(0,-4){\circlenode*[fillcolor=blue,
fillstyle=solid]{A4}{4}}
\rput(2.,-7.4641016){\circlenode*[fillcolor=blue,
fillstyle=solid]{A5}{5}}
\rput(6.,-7.4641016){\circlenode*[fillcolor=blue,
fillstyle=solid]{A6}{6}}
\ncline{A1}{A2} \ncline{A2}{A3}
\ncline{A3}{A4} \ncline{A4}{A5} \ncline{A5}{A6}
\ncline{A6}{A1}
\end{pspicture}
\end{center}
```

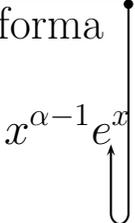
Esercizio 70. Realizzare



Esercizio 71. Realizzare l'esagono della pagina precedente con in più un settimo vertice non numerato al centro e connesso agli altri 6 da un segmento che finisce in una freccia.

Un'altra possibilità dell'uso dei nodi: collegare due elementi concettualmente vicini all'interno di una pagina:

Di uso frequente è la funzione esponenziale. Ad esempio nelle densità Gamma, che sono della forma

$$\frac{\lambda^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-x}$$


che viene ottenuto mediante

Di uso frequente è la funzione

```
\rnode{A}{\psframebox{esponenziale}}.
```

Ad esempio nelle densità Gamma, che sono della forma

```
$$
```

```
\frac{\lambda^\alpha}{\Gamma(\alpha)},
```

```
x^{\alpha-1}\rnode{B}{e^{-x}}
```

```
$$
```

```
\ncangles[angleA=-90, armA=3cm, armB=.5cm, angleB=-90, linearc=.15]**->{A}{B}
```