

Gestione di Figure e Tabelle con L^AT_EX

Nozioni di Base ed Alcuni Concetti Avanzati

Luca Caucci

caucci at mail.dm.unipi.it

&

Mariano Spadaccini

spadacciniweb at yahoo.it

VERSIONE 1.1.0 – 4 MARZO 2005

Indice

Prefazione	3
Introduzione	4
Perché questo manuale?	4
Struttura del manuale	5
Ringraziamenti	5
1 Concetti di base	6
1.1 Il comando <code>\includegraphics</code>	6
1.2 Esempio di inserimento di una figura	7
1.3 Principali opzioni di <code>\includegraphics</code>	8
1.4 Didascalie e referenziamento di figure e tabelle	10
1.5 Direttive di posizionamento	11
2 Gestione di sottofigure e sottotabelle	13
2.1 Un primo tentativo	13
2.2 L'ambiente <code>minipage</code> per la creazione di sottofigure	14
2.3 Il pacchetto <code>subfigure</code>	17
2.3.1 Creazione di sottofigure	17
2.3.2 Creazione di sottotabelle	19
3 Come ottenere determinate impaginazioni	22
3.1 Forzare l'impaginazione con <code>minipage</code>	22
3.2 La mia figura qui!	23
3.3 Figure in "landscape"	25
3.4 L'ambiente <code>floatfigure</code>	26
3.5 Perché solo immagini o tabelle?	30

4	Personalizzazione delle didascalie	33
4.1	Il “vecchio” pacchetto <code>caption</code>	33
4.1.1	Un primo esempio	33
4.1.2	Ridefinizione locale	35
4.2	Il pacchetto <code>caption</code> , versione 3	36
4.2.1	Esempi	37
5	Alcuni concetti avanzati	41
5.1	Creazione di grafici con <code>xfig</code>	41
5.2	Il pacchetto <code>psfrag</code>	42
5.3	Grafico di funzioni matematiche	44
5.4	Gestire file non-EPS	47
5.5	<code>pdfL^AT_EX</code> e <code>\DeclareGraphicsExtensions</code>	47
5.6	“Elenco delle figure” e “Elenco delle tabelle”	49
5.7	Cornici attorno alle figure con <code>PSTricks</code>	51
A	Alcune utili informazioni	57
A.1	Le unità di misura	57
A.2	Le “scatole”	58
A.3	Un esempio di <code>Makefile</code> per compilare documenti <code>L^AT_EX</code>	59
A.4	Alcuni consigli	61
A.5	Per saperne di più	64
A.6	Manuali su <code>L^AT_EX</code> (e <code>T_EX</code>)	69
	Riferimenti bibliografici	71

Prefazione

Il presente manuale tratta l'inclusione di figure e tabelle in un documento formattato con \LaTeX : presenta diversi metodi—dai più semplici ai più raffinati—per inserire figure e tabelle in documenti \LaTeX e per ottenere determinate impaginazioni e disposizioni degli oggetti.

Questo manuale, risultato della fusione di [Cau04] e [Spa04] scritti indipendentemente dai due autori, si rivolge principalmente ad utenti che abbiano un minimo di dimestichezza con \LaTeX e che sappiano, ad esempio, come includere un pacchetto passandovi alcune opzioni. Nonostante ciò, è stato ritenuto utile fornire alcuni riferimenti a manuali base sull'uso di \LaTeX (e \TeX): una lista di utili informazioni e riferimenti può essere trovata nella parte finale di questo manuale.

Invitiamo i lettori a segnalarci¹ errori (anche ortografici) ed omissioni nonché loro commenti e suggerimenti per rendere questo manuale aggiornato ed il più completo possibile.

Copyright (c) Luca Caucci e Mariano Spadaccini, 2004.
Questo documento può essere riprodotto, distribuito
e/o modificato, in tutto o in parte, secondo i termini
della GNU Free Documentation License, versione 1.1 o
successiva, pubblicata dalla Free Software Foundation;
senza Sezioni Non Modificabili, senza Testi Copertine
e senza Testi di Retro Copertina.

¹Gli indirizzi email degli autori sono indicati nella copertina di questo manuale.

Introduzione

Il pacchetto `graphicx` è il principale pacchetto per l’inserimento di immagini in documenti \LaTeX . Lo scopo di questo manuale è mostrare le funzionalità di base che tale pacchetto mette a disposizione. Inoltre, vedremo come sia possibile, ad esempio, creare sottofigure e sottotabelle e vedremo come personalizzare lo stile delle didascalie. Vedremo anche come sia possibile importare grafici creati mediante il programma `xfig` e come ottenere grafici “più piacevoli” utilizzando il pacchetto `psfrag`.

Lo scopo che ci si prefigge è quello di dare le informazioni base per poter creare figure e tabelle mediante \LaTeX . Sicuramente vi saranno alcuni argomenti che non verranno trattati in maniera approfondita; per essi l’utente interessato potrà far riferimento alla documentazione esaustiva dei vari pacchetti che verranno via via citati.

In questo manuale verranno frequentemente mostrati frammenti di codice d’esempio ed il relativo risultato, nella speranza che ciò possa essere d’aiuto alla comprensione e possa costituire un punto di partenza per la sperimentazione di altre opzioni. Inoltre, benché la maggior parte di quanto discusso in questo manuale resti valido anche per altri sistemi operativi, assumeremo che l’utente utilizzi un sistema operativo GNU/LINUX.

Perché questo manuale?

La necessità di dover includere immagini oppure creare tabelle in un documento PDF o POSTSCRIPT scritto con \LaTeX si presenta molto spesso ed altrettanto spesso capita di dover consultare diverse guide per trovare l’informazione cercata ed ottenere il risultato voluto.

Questo manuale nasce dalla volontà da parte degli autori di raccogliere e condensare in un unico documento i numerosi concetti sparsi in svariati manuali e di presentarli in maniera pratica, il più possibile chiara, organica ed esaustiva. Inoltre, crediamo sia utile fornire il maggior numero di consigli e suggerimenti nonché

una vasta gamma di riferimenti ad altri manuali su \LaTeX : concluderemo fornendo dunque un'ampia lista di consigli pratici nonché di riferimenti ad altri pacchetti (e relativa documentazione) per la manipolazione e creazione di figure, tabelle ed altro.

Struttura del manuale

Questo manuale è organizzato in cinque capitoli più un'appendice. Nel Capitolo 1 vengono date le informazioni di base sul pacchetto `graphicx` e su come creare figure/tabelle e riferimenti ad esse. Il Capitolo 2 mostra diversi modi per creare sottofigure e sottotabelle mentre nel Capitolo 3 vengono illustrate diverse tecniche per alterare la disposizione standard delle figure e tabelle che \LaTeX offre. Il Capitolo 4 descrive come personalizzare le didascalie delle figure e tabelle mentre l'ultimo capitolo—il Capitolo 5—elenca alcuni concetti avanzati come, ad esempio, l'uso del pacchetto `psfrag`, il comando `\DeclareGraphicsExtensions` e come creare gli elenchi delle figure e delle tabelle. Chiude questo manuale l'Appendice A che dà delle informazioni e suggerimento sulla gestione di figure e tabelle ed inoltre elenca una lista di utili manuali su \LaTeX (e \TeX).

Ringraziamenti

Il primo autore (L. C.) sente il dovere di ringraziare il gruppo S_{LAM} —Sapere Libero A Matematica—del Dipartimento di Matematica, Università di Pisa ed, in particolare, Fabio Natali per le sue utili osservazioni.

Il secondo autore (M. S.) desidera ringraziare l'ing. Marco Pratesi per i preziosi suggerimenti ed il dott. Damiano Verzulli sia per il lavoro di revisione sia per il prezioso suggerimento introducendo lo stesso autore all'utilizzo di un ottimo strumento per la creazione di documenti: \LaTeX . Si ringrazia inoltre l'ing. Francesco Di Cintio per l'attento lavoro di revisione.

Capitolo 1

Concetti di base

1.1 Il comando `\includegraphics`

Il comando `\includegraphics` è senza dubbio il comando più importante del pacchetto `graphicx` ed è quello che permette di inserire un'immagine nel documento. Non è questo il luogo più opportuno per darne una descrizione dettagliata, ciononostante, è utile conoscere, seppur a grandi linee, cosa esso faccia.

Il comando `\includegraphics` accetta come parametro obbligatorio il nome di un file; tale file contiene l'immagine che dovrà essere inserita nel documento e, nella maggior parte dei casi, è un file in formato `.eps`. Ad `\includegraphics` può essere passata anche una lista di parametri opzionali, di cui verrà data ampia descrizione più avanti. Ciò che `\includegraphics` si limita a fare è esaminare il file che contiene l'immagine da includere, elaborare le eventuali opzioni, determinare lo spazio da riservare all'immagine da inserire e fare in modo che tutte le informazioni necessarie siano presenti nel file che `LATEX` dovrà generare. È solo nella fase di generazione del file `.ps` oppure `.pdf` che l'immagine verrà fisicamente inserita nel documento finale.

Usualmente, il comando `\includegraphics` viene utilizzato all'interno dell'ambiente `figure` per creare una figura (che nel documento finale sarà posizionata in accordo a determinate opzioni e regole estetiche) con, eventualmente, una didascalia. Può, però, essere utilizzato in altri modi; ad esempio, quando si vuole inserire un'immagine in un punto ben preciso del documento e non si ha bisogno di una didascalia (ad esempio per inserire il logo della propria azienda nella copertina di una relazione).

Nelle sezioni che seguono verranno introdotte le nozioni di base che è necessario sapere per creare figure e saranno descritte le principali opzioni di questo comando.

1.2 Esempio di inserimento di una figura

In questo esempio, vedremo i comandi base per poter inserire una figura in un documento \LaTeX . Il frammento di codice che è stato utilizzato per creare la Figura 1.1 è:

```
----->&-----  
\begin{figure}[htbp]  
  \centering  
  \includegraphics{eps/mission.eps}  
  \caption{File \protect\url{mission.eps}\label{fig:mission}}  
\end{figure}  
----->&-----
```



Figura 1.1: File mission.eps

Usualmente, l'immagine di una figura dovrà essere centrata orizzontalmente nella pagina. Nell'esempio mostrato, ciò viene ottenuto mediante il comando `\centering`.

L'immagine viene quindi inserita nel documento, viene indicata la didascalia e viene creata un'etichetta per poter in seguito indicare un riferimento alla figura. Questo esempio mostra lo schema base per la creazione di figure: esso si rileva

sufficiente nella maggior parte dei casi. Differenti impaginazioni e risultati più evoluti possono essere ottenuti prendendo come punto di partenza queste linee di codice ed utilizzando opportuni pacchetti e/o comandi.

1.3 Principali opzioni di `\includegraphics`

La sintassi generale del comando `\includegraphics` è:

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

in cui “opzioni” sta ad indicare una lista di una o più opzioni separate da virgole e “nomefile.eps” è il nome del file in formato EPS (estensione .eps) che si vuole inserire.

Le principali opzioni che possono essere utilizzate sono:¹

- “**height**”: indica l’altezza dell’immagine; l’immagine verrà ridotta oppure ingrandita affinché risulti dell’altezza voluta. Esempio: **height=10cm**.
- “**width**”: indica la larghezza dell’immagine; l’immagine verrà ridotta oppure ingrandita affinché risulti della larghezza voluta. Esempio: **width=10em**.
- “**scale**”: permette di ingrandire oppure ridurre le dimensioni di un’immagine. Esempi: **scale=1.5**; **scale=.25**.
- “**angle**”: permette di ruotare in senso antiorario l’immagine di un certo angolo espresso in gradi. Esempi: **angle=180** oppure **angle=-35**.
- “**keepaspectratio**”: quando sia la larghezza sia l’altezza dell’immagine vengono specificate; con questa opzione si previene che l’immagine scalata venga distorta.

Facciamo osservare che le eventuali opzioni da applicare all’immagine vengono effettuate da sinistra verso destra e, quindi, scambiare l’ordine di due o più opzioni potrebbe dar luogo a risultati diversi. Ad esempio, la lista di opzioni **height=6cm**, **angle=45** indica che l’immagine deve prima venir scalata in modo che abbia l’altezza indicata e, successivamente, la rotazione viene applicata all’immagine scalata; le opzioni **angle=45**, **height=6cm**, invece, indicano che prima l’immagine deve essere ruotata e, successivamente, l’immagine ruotata deve venir scalata in modo tale che essa risulti di una particolare altezza: il risultato che si ottiene ora non è, in generale, lo stesso che si otteneva nel caso precedente.

¹Per un elenco completo rimandiamo il lettore a [Car99] e [Rec97].


Vediamo un esempio d'uso delle opzioni appena elencate:

```
----->⌘-----  
\begin{figure}[htbp]  
  \centering  
  \includegraphics[height=6cm, width=5cm, angle=30,  
    keepaspectratio]{eps/lena.eps}  
  \caption{Esempio d'uso delle opzioni}  
\end{figure}  
----->⌘-----
```



Figura 1.2: Esempio d'uso delle opzioni

Il comando `\includegraphics` non inizia un nuovo paragrafo e quindi può essere utilizzato “in-line” e senza racchiuderlo nell’ambiente `figure`, per inserire una piccola immagine oppure un simbolo in una riga di testo come nel seguente esempio:

“In Arizona, la  permette di andare da Tucson fino al confine con la riserva indiana dei Navajo”.

Le opzioni maggiormente utilizzate sono riassunte nella Tabella 1.1; per un elenco completo si rimanda il lettore a [Car99] oppure [Rec97].

Opzione	Effetto ottenuto
“width”	Ridimensiona la figura alla larghezza specificata mantenendo inalterate le proporzioni se è assente <code>height</code> .
“height”	Ridimensiona la figura all’altezza specificata mantenendo inalterate le proporzioni se è assente <code>width</code> .
“angle”	Ruota la figura in senso antiorario di un certo angolo espresso in gradi.
“scale”	Scala la figura lasciandone inalterate le proporzioni.

Tabella 1.1: Opzioni per il pacchetto `graphicx`

1.4 Didascalie e referenziamento di figure e tabelle

Per poter creare una didascalia² da associare ad una figura/tabella è necessario utilizzare il comando `\caption` indicando tra parentesi graffe il contenuto della didascalia. Se si inverte l’ordine dei comandi `\caption` ed `\includegraphics`, la figura/tabella sarà posizionata sotto la didascalia.

In uno degli esempi precedenti (creazione della Figura 1.1), si può notare che è stato usato il comando `\label`; esso permette di associare ad una sequenza di caratteri (cioè ad una “etichetta”) il numero della figura che si sta creando. Ad esempio, il comando `\label{fig:mission}` permette di associare all’etichetta “`fig:mission`” il numero della figura creata. Quando è necessario effettuare un riferimento ad una figura/tabella, il comando `\ref` permette di ottenere il numero associato alla figura/tabella. Il codice:

```
----->⌘-----
La \figurename~\ref{fig:mission} è una figura
creata con l’ambiente \verb"figure".
----->⌘-----
```

produce:

La Figura 1.1 è una figura creata con l’ambiente `figure`.

²Nella didascalia della Figura 1.1 è stato utilizzato il comando `\url` fornito dal pacchetto `url` per poter ottenere un carattere opportuno per indicare il nome del file. Più in generale, il comando `\url` permette di inserire indirizzi Internet che possono venir spezzati in corrispondenza di “/” oppure “.” per creare le necessarie andate a capo.

È importante utilizzare il comando `\figurename` (che genera “Figura”) perché se si vuole in seguito indicare tutte le figure con un’altra parola (ad esempio con “Fig.”) è sufficiente ridefinire (mediante `\renewcommand`) il comando `\figurename` e non vi è la necessità di controllare tutto il file sorgente e modificarlo dove necessario. Sono disponibili anche i comandi `\chaptername`, `\appendixname`, `\tablename`, `\partname` e `\pagename` che vanno utilizzati nello stesso modo e per lo stesso motivo di `\figurename`.

È altresì importante utilizzare il carattere `~` (detto “spazio insecabile”) per evitare un antiestetico numero all’inizio di una riga in seguito ad una andata a capo.

1.5 Direttive di posizionamento

L’ambiente `figure` (così come l’ambiente `table` ed altri) permette di creare un oggetto “mobile” all’interno del quale inserire un oggetto grafico. Gli oggetti mobili—in quanto tali—non sempre appaiono nel documento finale nel punto esatto in cui vengono dichiarati nel file sorgente: possono venir sposta in altri punti del documento al fine di ottenere una impaginazione più piacevole. È possibile specificare, mediante un parametro opzionale dell’ambiente `figure`, delle “preferenze di posizionamento”: la lettera “`h`” indica che vorremmo che la figura/tabella fosse posizionata esattamente nel punto in cui l’ambiente corrispondente è utilizzato; “`t`” indica che la figura/tabella vorremmo fosse posizionata in cima alla pagina; “`b`” indica in fondo alla pagina mentre “`p`” indica che la figura/tabella vorremmo fosse messa in una pagina a sé stante.

In fase di compilazione del documento, \LaTeX cerca di determinare l’impaginazione esteticamente migliore in accordo alla lista di direttive di posizionamento specificate. Nel codice d’esempio che ha generato la Figura 1.1, se non fosse possibile inserire la figura nel punto del sorgente \LaTeX in cui l’ambiente `figure` viene utilizzato, \LaTeX cercherà di posizionare la figura in cima alla pagina. Se anche questa soluzione non si rilevasse soddisfacente, \LaTeX cercherà di posizionare la figura in fondo alla pagina e, se anche tale scelta non venisse ritenuta quella ottimale, la figura sarà posizionata in una pagina a sé stante. Facciamo osservare che \LaTeX analizza le direttive di posizionamento secondo l’ordine “`h`” \rightarrow “`t`” \rightarrow “`b`” \rightarrow “`p`”, saltando le direttive non indicate. In altre parole, l’ordine con cui le direttive vengono specificate è ininfluente: conta solo se esse sono state indicate o meno. È sconsigliabile indicare una sola direttiva di posizionamento pretendendo che \LaTeX inserisca la figura esattamente dove si desidera. Infatti, in tali casi, se la figura non può essere posizionata in accordo con l’unica direttiva indicata, tutte le successive

figure del documento non verranno inserite (in quanto il posizionamento di figure deve rispettare l'ordine di creazione delle stesse) e, arrivato alla fine del processo di compilazione, \LaTeX emetterà un messaggio di errore.

È anche possibile chiedere a \LaTeX di “rilassare” alcune restrizioni imposte, mediante la direttiva di posizionamento “!” (es: `!htbp`); in tal caso, si accetta di ottenere un'impaginazione che potrebbe essere leggermente meno piacevole chiedendo a \LaTeX di cercare di ottenere una delle prime impaginazioni indicate. Se non si indica alcuna direttiva di posizionamento, \LaTeX , tacitamente, assumerà che sia stata indicata la sequenza `tbp`.

La Tabella 1.2 riassume il significato dei caratteri per la gestione del posizionamento di oggetti mobili.

Carattere	Sposta l'oggetto...
“h”	“ <i>here</i> ”, nel punto in cui compare il testo
“t”	“ <i>top</i> ”, in cima ad una pagina
“b”	“ <i>bottom</i> ”, in fondo ad una pagina
“p”	“ <i>page of floats</i> ”, in una pagina contenente solo oggetti mobili
“!”	rilassa alcuni parametri estetici riguardanti il posizionamento di oggetti mobili (in tal modo è più probabile ottenere una delle prime impaginazioni desiderate)

Tabella 1.2: Significato dei caratteri di posizionamento

Capitolo 2

Gestione di sottofigure e sottotabelle

A volte può presentarsi la necessità di voler creare sottofigure cioè raggruppare in un'unica figura due o più immagini correlate tra loro. Problema analogo si presenta nella gestione delle tabelle. In questo capitolo saranno presentati alcuni metodi per poter creare sottofigure e sottotabelle.

2.1 Un primo tentativo

Se volessimo disporre più figure orizzontalmente, potremmo utilizzare alcune delle nozioni apprese nel precedente capitolo. Si consideri il seguente esempio:

```
----->&-----  
\begin{figure}[htbp]  
  \centering  
  \includegraphics[width=35mm]{eps/airplane.eps}%    "%" necessario  
  \quad\quad  
  \includegraphics[width=45mm]{eps/lena.eps}  
  \caption{Didascalia comune alle due figure}  
\end{figure}  
----->&-----
```

Questo metodo potrebbe, in alcuni casi, rivelarsi inadeguato perché, ad esempio, non vi è modo di impostare una didascalia per ogni immagine. Per questo e



Figura 2.1: Didascalia comune alle due figure

per altri problemi che si possono facilmente presentare,¹ è necessario analizzare altre soluzioni. \LaTeX , ovviamente, offre diversi metodi tra cui scegliere per risolvere il problema. Tra tutti, impera il metodo implementato dal pacchetto `subfigure`, che ha il vantaggio di riutilizzare quanto appreso nel precedente capitolo. Prima di far ciò, ci proponiamo di mostrare un’alternativa che fa uso dell’ambiente `minipage`. Questi due differenti modi di procedere sono ampiamente descritti anche in altri manuali fra i quali ricordiamo [Rec97] e [Coc95].

Facciamo notare l’uso del carattere “%” per evitare che \LaTeX interpreti i caratteri di andata a capo come degli spazi che potrebbero comportare un non centramento esatto dell’immagine.

2.2 L’ambiente `minipage` per la creazione di sottfigure

In questa sezione è illustrato un metodo alternativo il quale, anche se potrebbe risultare primitivo, è spesso applicato in contesti differenti; questo metodo consiste nell’uso dell’ambiente `minipage`. In questo caso, l’ambiente `minipage` sostituisce l’ambiente `figure` introducendo una differenza concettuale rispetto a quanto descritto sinora: l’ambiente `minipage` non tratta le figure come oggetti “mobili”, bensì costringe a trattarle come oggetti “statici”. Così facendo, si costringe \LaTeX ad inserire l’immagine nel punto esatto in cui viene dichiarata nel file sorgente anche se ciò potrebbe comportare una impaginazione sgradevole.

La sintassi generale dell’ambiente `minipage` è la seguente:

¹Ad esempio, se si affiancano due figure di dimensioni molto diverse, questa metodologia di inclusione risulterebbe esteticamente poco piacevole.

```
-----><-----  
\begin{minipage}[posizione]{larghezza}  
  ...  
  ...  
  ...  
\end{minipage}  
-----><-----
```

in cui il parametro:

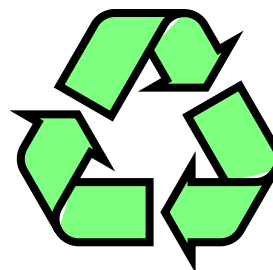
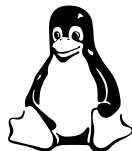
- “posizione” può essere “c”, “t” oppure “b” a seconda che l’allineamento verticale voluto sia “center”, “top” oppure “bottom”;
- “larghezza” indica la larghezza del testo racchiuso all’interno dell’ambiente minipage ed è esprimibile come una qualsiasi lunghezza in L^AT_EX (si veda la Sezione A.1).

Vediamo alcuni esempi di utilizzo dell’ambiente minipage.

Esempio 1

```
-----><-----  
\begin{center}  
  \begin{minipage}[c]{.40\textwidth}  
    \centering  
    \includegraphics[width=.30\textwidth]{eps/penguin.eps}  
  \end{minipage}%  
  \hspace{10mm}%  
  \begin{minipage}[c]{.40\textwidth}  
    \centering  
    \includegraphics[width=.70\textwidth]{eps/recycle.eps}  
  \end{minipage}  
\end{center}  
-----><-----
```

che produce:



Si può notare che le didascalie non sono state indicate perché, così come è strutturato, non è possibile utilizzare il comando `\caption`. In questo esempio non è stato fatto uso dell'ambiente `figure`: come mostrato nell'esempio che segue, se si utilizza l'ambiente `figure`, è possibile indicare le didascalie.

Esempio 2

```
----->&
\begin{figure}[htbp]
  \centering
  \begin{minipage}[c]{.40\textwidth}
    \centering\setlength{\captionmargin}{0pt}%
    \includegraphics[width=.30\textwidth]{eps/penguin.eps}
    \caption{Immagine piccola}
  \end{minipage}%
  \hspace{10mm}%
  \begin{minipage}[c]{.40\textwidth}
    \centering\setlength{\captionmargin}{0pt}%
    \includegraphics[width=.70\textwidth]{eps/recycle.eps}
    \caption{Immagine grande}
  \end{minipage}
  \caption{Didascalia comune alle
    due figure\label{fig:minipage2}}
\end{figure}
----->&
```

Il risultato che si ottiene è mostrato in Figura 2.4.

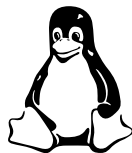


Figura 2.2: Immagine piccola

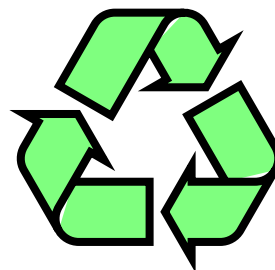


Figura 2.3: Immagine grande

Figura 2.4: Didascalia comune alle due figure

Facciamo notare che nel secondo esempio è stata alterata (modificando il valore di `\captionmargin`) l'usuale larghezza dei margini destri e sinistri per le didascalie

delle sottofigure. Inoltre, è possibile modificare (localmente oppure globalmente) l'allineamento e lo stile delle didascalie delle figure o sottofigure. Ritorniamo su questi punti nel Capitolo 4.

2.3 Il pacchetto `subfigure`

2.3.1 Creazione di sottofigure

Il pacchetto `subfigure` è senza ombra di dubbio la soluzione più comoda e flessibile per la creazione di sottofigure. Infatti, questo pacchetto mette a disposizione il comando `\subfigure` che permette di creare in maniera semplice e veloce sottofigure. Inoltre, a ciascuna sottofigura è possibile associare una didascalia ed una etichetta; quest'ultima, viene definita mediante l'usuale comando `\label`.

La sintassi del comando `\subfigure` è molto semplice e ne diamo, quindi, immediatamente un esempio d'uso:

```
-----><-----
\begin{figure}[htbp]
  \centering%
  \subfigure[\protect\url{airplane.eps}\label{fig:airplane}]%
    {\includegraphics{eps/airplane.eps}}\qqquad\qqquad
  \subfigure[\protect\url{lena.eps}\label{fig:lena}]%
    {\includegraphics{eps/lena.eps}}\qqquad\qqquad
  \subfigure[\protect\url{peppers.eps}\label{fig:peppers}]%
    {\includegraphics{eps/peppers.eps}}
  \caption{Creazione di sottofigure\label{fig:sottofigure}}
\end{figure}
-----><-----
```

Questo esempio, suggerisce che il comando `\subfigure` non faccia altro che creare un'area all'interno di una figura, alla quale può essere opzionalmente associata una didascalia ed una etichetta. Le etichette corrispondenti a sottofigure, vengono referenziate nel modo usuale mediante il comando `\ref`; ad esempio il frammento di codice “`\figurename~\ref{fig:peppers}`” genera “Figura 2.5(c)”. Facciamo notare la che le varie sottofigure di una figura vengono opportunamente sottonumerate cosa che non avviene se si usa l'ambiente `minipage` per creare sottofigure. La documentazione del pacchetto `subfigure` [Coc95] descrive dettagliatamente come personalizzare lo stile della sottonumerazione.



(a) airplane.eps



(b) lena.eps



(c) peppers.eps

Figura 2.5: Creazione di sottofigure

2.3.2 Creazione di sottotabelle

A dispetto del nome, il pacchetto `subfigure` definisce il comando `\subtable` che può essere utilizzato per creare sottotabelle—come quelle mostrate nella Tabella 2.1. Il codice che ha permesso di ottenere tale risultato è:

```
----->
\begin{table}[htbp]
  \centering
  \subtable[Treno \textsl{R/}~21338\label{tab:R21338}]{%
    \begin{tabular}{l|l}
      \multicolumn{1}{c|}{\textbf{Stazione}} & & \\
      \multicolumn{1}{c}{\textbf{Ora}} & & \\
      \hline
      Pisa Centrale & & 16:38 & \\
      Pisa S.~R. & & 16:43 & \\
      Lucca & & 17:00 & \\
      Altopascio & & 17:10 & \\
      Pescia & & 17:19 & \\
      Montecatini C. & & 17:25 & \\
      Montecatini T. & & 17:28 & \\
      Pistoia & & 17:37 & \\
      Prato P.~S. & & 17:47 & \\
      Prato Centrale & & 17:49 & \\
    \end{tabular}
  } \quad \quad
  \subtable[Treno \textsl{R/}~6548\label{tab:R6548}]{%
    \begin{tabular}{l|l}
      \multicolumn{1}{c|}{\textbf{Stazione}} & & \\
      \multicolumn{1}{c}{\textbf{Ora}} & & \\
      \hline
      Prato Centrale & & 17:54 & \\
      Vaiano & & 18:01 & \\
      Vernio & & 18:08 & \\
      San Benedetto & & 18:28 & \\
      Grizzana & & 18:32 & \\
      Monzuno-Vado & & 18:39 & \\
      Pianoro & & 18:46 & \\
      Bologna S.~R. & & 18:54 & \\
      Bologna Centrale & & 19:01 & \\
    \end{tabular}
  }
  \caption{Esempio di creazione di sottotabelle}
\end{table}
----->
```

Stazione	Ora	Stazione	Ora
Pisa Centrale	16:38	Prato Centrale	17:54
Pisa S. R.	16:43	Vaiano	18:01
Lucca	17:00	Vernio	18:08
Altopascio	17:10	San Benedetto	18:28
Pescia	17:19	Grizzana	18:32
Montecatini C.	17:25	Monzuno-Vado	18:39
Montecatini T.	17:28	Pianoro	18:46
Pistoia	17:37	Bologna S. R.	18:54
Prato P. S.	17:47	Bologna Centrale	19:01
Prato Centrale	17:49		

(a) Treno *R* 21338(b) Treno *R* 6548**Tabella 2.1:** Esempio di creazione di sottotabelle

Anche le tabelle sono oggetti mobili e quindi per esse \LaTeX utilizza le stesse regole di posizionamento delle figure. Ciò significa che è possibile far uso delle stesse direttive di posizionamento che si hanno per le figure (vedi Tabella 1.2). Inoltre, la posizione finale dell’oggetto definito mediante l’ambiente `table` viene determinata da \LaTeX allo stesso modo in cui si determina il posizionamento di un oggetto creato con l’ambiente `figure` (si veda la Sezione 1.5).

Se si utilizzano i pacchetti `caption` oppure `ccaption` per poter alterare lo stile delle didascalie delle figure, tale stile sarà applicato anche alle didascalie delle tabelle.

Facciamo notare che è definito il comando `\tablename` (che produce la stringa “Tabella”) e che, per questioni di portabilità, andrebbe utilizzato ogniqualvolta si creano riferimenti a tabelle.

Segnaliamo infine che utilizzando i pacchetti `booktabs` e/o `ctable` è possibile creare tabelle esteticamente più piacevoli di quelle che gli usuali comandi \LaTeX permettano di creare (si confrontino la Tabella 2.2(a) con la Tabella 2.2(b); il simbolo “€” è stato ottenuto mediante il comando `\EUR` fornito dal pacchetto `marvosym`).

Item		
Animal	Descr.	Price (€)
Gnat	per gram	13.65
	each	0.01
Gnu	stuffed	92.50
Emu	stuffed	33.33
Armadillo	frozen	8.99

(a) Tabella creata con `booktabs`

Item		
Animal	Descr.	Price (€)
Gnat	per gram	13.65
	each	0.01
Gnu	stuffed	92.50
Emu	stuffed	33.33
Armadillo	frozen	8.99

(b) Tabella creata senza `booktabs`

Tabella 2.2: Confronto tra una tabella creata con il pacchetto `booktabs` ed una creata con gli usuali comandi `LATEX`

Capitolo 3

Come ottenere determinate impaginazioni

3.1 Forzare l'impaginazione con minipage

Quando si vogliono inserire una o più figure di larghezza limitata, è preferibile disporre la didascalia di fianco alla figura o alle figure per ottenere un miglior impatto visivo. Vediamo immediatamente un esempio chiarificatore:

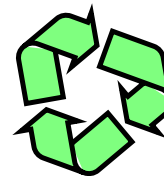
```
----->&-----  
\begin{figure}[!htbp]  
  \centering  
  \mbox{%  
    \begin{minipage}{.20\textwidth}  
      \includegraphics[angle=80, width=\textwidth]{eps/penguin.eps}  
    \end{minipage}%  
    \quad  
    \begin{minipage}[c]{.50\textwidth}  
      \caption{Questa \ 'e la didascalia per l'immagine di sinistra}  
      \caption{Questa \ 'e la didascalia per l'immagine di destra}  
    \end{minipage}%  
    \quad  
    \begin{minipage}{.20\textwidth}  
      \includegraphics[angle=-80, width=\textwidth]{eps/recycle.eps}  
    \end{minipage}  
  }  
\end{figure}  
----->&-----
```

In questo modo è possibile disporre le didascalie delle due figure centralmente una sopra all'altra mentre le figure vengono disposte lateralmente. Per forzare sia le didascalie che le figure a stare tutte sulla stessa linea impedendo eventuali “andate a capo”, è sufficiente racchiudere il tutto con il comando `\mbox`.



Figura 3.1: Questa è la didascalia per l'immagine di sinistra

Figura 3.2: Questa è la didascalia per l'immagine di destra



3.2 La mia figura qui!

A volte potrebbe capitare di voler che una certa figura (o tabella) venga posizionata in un punto ben preciso di un documento. Anche se in generale sarebbe preferibile lasciar scegliere a \LaTeX il punto più opportuno in cui disporre una figura oppure una tabella, vi sono diversi modi per forzare una data impaginazione.

Ad esempio, una figura può essere inclusa utilizzando una dichiarazione simile alla seguente:

```
-----><-----
\begin{figure}[!h]
  \centering
  \includegraphics[scale=.3]{eps/penguin.eps}
  \caption{Semplice inclusione}
\end{figure}
-----><-----
```

in cui le opzioni di posizionamento “!h” cercano di forzare una determinata impaginazione chiedendo a \LaTeX di non considerare alcuni parametri estetici, i quali potrebbero impedire all'oggetto di essere posizionato nel punto il cui l'autore del documento vorrebbe venga posizionato. Ciò, però, non è sufficiente per fare in modo che \LaTeX disponga l'oggetto esattamente nel punto in cui lo si è dichiarato.¹

¹ \LaTeX , essendo molto flessibile, ha vari gradi di “forzatura”.

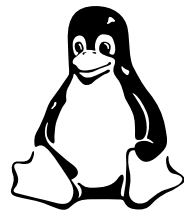
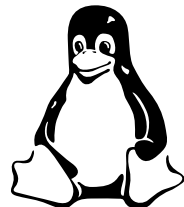


Figura 3.3: Semplice inclusione

Una soluzione più restrittiva—ed esteticamente discutibile, poiché \LaTeX ignorerà tutti i parametri estetici—è quella di *non* trasformare le figure in oggetti mobili, cosicché \LaTeX le disporrà nel punto esatto del documento in cui vengono dichiarate. Il codice seguente mostra un esempio di questa soluzione:

```
-----><-----  
\begin{center}  
  \includegraphics[scale=.3]{eps/penguin.eps}  
\end{center}  
-----><-----
```

che produce semplicemente:



In questo modo si forza l’inserimento dell’immagine subito dopo il testo che la precede nel file sorgente e, pertanto, potrebbe verificarsi che la figura esca dal margine inferiore del foglio. Inoltre, l’esempio appena riportato è di poca utilità poiché la figura è impoverita a causa della mancanza della usuale didascalia a cui siamo abituati.

La soluzione al problema di forzare il posizionamento di una figura con didascalia in un ben preciso punto del documento, viene fornita dal pacchetto `float` il quale mette a disposizione una versione modificata dell’ambiente `figure` con la quale è possibile ottenere il risultato voluto. Il pacchetto `float` permette di specificare la direttiva di posizionamento “H” che viene utilizzata per forzare il posizionamento

di una figura nel punto del file sorgente in cui essa viene dichiarata. Se si indica la direttiva di posizionamento “H”, non è possibile indicarne altre per la stessa figura.

Ad esempio, il codice:

```
----->⌘-----  
\begin{figure}[H]  
  \centering  
  \includegraphics[scale=.4]{eps/penguin.eps}  
  \caption{Risultato ottenuto con il pacchetto  
    \textsf{float} e con la direttiva di posizionamento  
    “\protect\verb"H””\label{fig:here}}  
\end{figure}  
----->⌘-----
```

permette di ottenere il risultato mostrato in Figura 3.4.

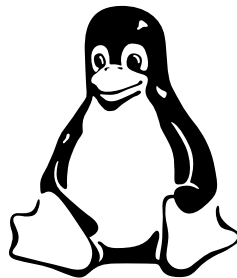


Figura 3.4: Risultato ottenuto con il pacchetto `float` e con la direttiva di posizionamento “H”

3.3 Figure in “landscape”

L’ambiente `sidewaysfigure` (fornito dal pacchetto `rotating`) permette di inserire figure in “landscape”, cioè ruotate di 90° oppure di 270° (a seconda del tipo di documento e del fatto che il numero della pagina corrente sia pari oppure dispari) in maniera tale che anche figure di larghezza elevata possano essere inserite in un documento \LaTeX .

L’ambiente `sidewaysfigure` sostituisce in tutto e per tutto l’ambiente `figure` con l’unica differenza che non è possibile indicare direttive di posizionamento: l’immagine viene *sempre* inserita in una pagina a sé stante. Come esempio, si consideri il seguente frammento di codice \LaTeX :

```

----->
\begin{sidewaysfigure}
  \centering
  \subfigure[\protect\url{airplane.eps}]%
  {\includegraphics{eps/airplane.eps}}\quad\quad
  \subfigure[\protect\url{lena.eps}]%
  {\includegraphics{eps/lena.eps}}\quad\quad
  \subfigure[\protect\url{peppers.eps}]%
  {\includegraphics{eps/peppers.eps}}
  \caption{Creazione di figure in ‘‘landscape’’. La didascalia
    di questa figura \e \emph{volutamente} molto lunga
    per mostrare come l’ampiezza delle didascalie sia
    correttamente calcolata in funzione della dimensione
    orizzontale dell’area di stampa. Confrontando questa
    figura con la \figurename~\ref{fig:sottofigure} si
    nota la diversa disposizione delle sottofigure dovuta
    alle diverse dimensioni dell’area di stampa
    \label{fig:esempioSideways}
  }
\end{sidewaysfigure}
----->

```

Come si può osservare dalla Figura 3.5, la larghezza della didascalia è determinata in funzione della larghezza dell’area di stampa (ovvero in funzione dell’altezza del foglio). Inoltre, benché sia stato utilizzato praticamente lo stesso codice L^AT_EX che ha permesso di creare la Figura 2.5, la disposizione delle sottofigure è diversa: all’interno dell’ambiente `figure` le sottofigure si comportano allo stesso modo in cui si comportano le parole durante la scrittura di un paragrafo e quindi, laddove vi sia la necessità, una o più figure posso essere “mandate a capo”.

Analogamente a `sidewaysfigure`, l’ambiente `sidewaystable` permette di inserire in “landscape” tabelle molto larghe che, anche in questo caso, verranno posizionate in una pagina a sé stante.

3.4 L’ambiente `floatfigure`

Il pacchetto `floatflt` mette a disposizione l’ambiente `floatfigure` con il quale è possibile inserire del testo accanto ad una figura. Ciò può rilevarsi utile quando si hanno figure piccole e si vorrebbe rendere più “compatto” possibile il documento evitando di avere ampi spazi bianchi a destra e a sinistra di una figura.

La sintassi dell’ambiente `floatfigure` è la seguente:



(a) airplane.eps



(b) lena.eps



(c) peppers.eps

Figura 3.5: Creazione di figure in “landscape”. La didascalia di questa figura è *volutamente* molto lunga per mostrare come l’ampiezza delle didascalie sia correttamente calcolata in funzione della dimensione orizzontale dell’area di stampa. Confrontando questa figura con la Figura 2.5 si nota la diversa disposizione delle sottofigure dovuta alle diverse dimensioni dell’area di stampa

```

----->⌘-----
\begin{floatfigure}[posizione]{larghezza}
  ...
  ...
  ...
\end{floatfigure}
----->⌘-----

```

in cui il parametro:

- “posizione” può essere “r”, “l”, “p” oppure “v” in funzione di dove si vuole posizionare la figura rispetto al testo: “r” sta per “right”; “l” sta per “left”; “p” indica che la figura deve essere posizionata a destra se il numero della pagina è dispari, a sinistra se il numero della pagina è pari; “v” indica che il posizionamento della figura deve essere quello indicato come opzione quando il pacchetto floatflt è stato caricato;
- “larghezza” indica la larghezza dell’area da riservare alla figura ed è esprimibile come una qualsiasi lunghezza in L^AT_EX (si veda la Sezione A.1).

La Figura 3.6 mostra un esempio di figura creata con il pacchetto floatflt.

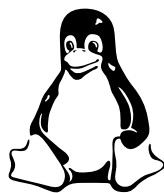


Figura 3.6: Risultato ottenuto con il pacchetto floatflt

Il pacchetto definisce anche il comando `floatingfigure` che permette di creare tabelle con del testo che continua affianco. Il codice che ha permesso di ottenere il risultato di Figura 3.6 è:

```

----->⌘-----
...
La \figurename~\ref{fig:floatfltexample} mostra un esempio
di figura creata con il pacchetto \textsf{floatflt}.
\begin{floatingfigure}[p]{.50\textwidth}
  \centering
  \includegraphics[width=.15\textwidth]{eps/penguin.eps}

```

Sfortunatamente, questo pacchetto presenta molti problemi e limitazioni per cui non sempre si comporta come ci si aspetta. Una descrizione dei problemi che possono insorgere usando il pacchetto floatflt è riportata nella documentazione del pacchetto [Dah98]; comunque, nella maggior parte dei casi, floatflt si comporta egregiamente. Il

```

\caption{Risultato ottenuto con il pacchetto
\textsf{floatflt}\label{fig:floatfltexample}}
\end{floatingfigure}
Sfortunatamente, questo pacchetto presenta molti problemi
e limitazioni per cui non sempre si comporta come ci si
aspetta.
...
-----><-----

```

Il consiglio che ci sentiamo di dare è di provare ad usare il pacchetto `floatflt` e, se non dovesse caricare l'immagine e/o disporla come si desidera, si può ricorrere al più affidabile ambiente `minipage` del quale è già stato presentato un modo di impiego nella Sezione 2.2. Mostriamo ora in dettaglio come sia possibile utilizzare l'ambiente `minipage` per ottenere una impaginazione analoga a quella che si ottiene con il pacchetto `floatflt`.

Vediamo ora un esempio che utilizza l'ambiente `minipage`:

Esempio 1

```

-----><-----
\emph{Testo sopra l'immagine.}\
\begin{minipage}{.30\textwidth}
\centering
\includegraphics[angle=80, width=.70\textwidth]{eps/recycle.eps}
\end{minipage}%
\begin{minipage}{.70\textwidth}
\emph{Testo a destra dell'immagine.}
\end{minipage}\
\emph{Testo sotto l'immagine.}
-----><-----

```

Testo sopra l'immagine.



Testo a destra dell'immagine.

Testo sotto l'immagine.

L'esempio può essere completato attraverso l'inserimento manuale della didascalia:

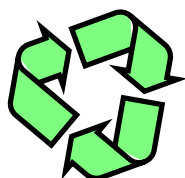
Esempio 2

```

----->
\emph{Testo sopra l'immagine.}\\
\begin{minipage}{.30\textwidth}
  \centering
  \includegraphics[angle=80, width=.70\textwidth]
    {eps/recycle.eps}\\
  \emph{\small Didascalia}\bigskip
\end{minipage}
\begin{minipage}{.70\textwidth}
  \emph{Testo a destra dell'immagine.}
\end{minipage}\\
\emph{Testo sotto l'immagine.}
----->

```

Testo sopra l'immagine.



Testo a destra dell'immagine.

Didascalia

Testo sotto l'immagine.

Ci preme sottolineare la non praticità dell'uso dell'ambiente `minipage` per ottenere questo tipo di impaginazioni. In particolare, non è possibile utilizzare il comando `\caption` per la creazione delle didascalie ed inoltre la quantità di testo da disporre accanto all'immagine non può essere scelta in maniera automatica ma va determinata manualmente volta per volta.

3.5 Perché solo immagini o tabelle?

Negli esempi mostrati precedentemente, gli ambienti `figure` e `table` sono stati utilizzati per inserire immagini e tabelle, rispettivamente. Ciò non impedisce di utilizzare tali ambienti con cose diverse da immagini o tabelle, ... come dimostra la Figura 3.7.

Il codice che ha permesso di ottenere il risultato di Figura 3.7 è:²

```
----->
\begin{figure}[htbp]
  \centering\rotatebox{150}{\scalebox{0.75}{\fbox{%
  \begin{minipage}{.70\linewidth}
    La soluzione del problema di Cauchy:
    \begin{equation*}
      \left\{
        \begin{array}{l}
          y'(x) = a(x) y(x) + b(x) \quad \backslash\backslash
          y(x_0) = y_0 \quad \backslash\backslash
        \end{array}
      \right.
    \end{equation*}
    \end{minipage}
  }}
  \caption{Esempio di figura senza
    immagine!\label{fig:esempioAltro}}%
\end{figure}
----->
```

²Non commenteremo tutte le istruzioni che hanno permesso ottenere il risultato mostrato; per ulteriori informazioni, si rimanda il lettore alla documentazione del pacchetto `graphicx`.

e

$$y(x) = e^{\int_x^{x_0} a(t) dt} \left(y_0 + \int_x^{x_0} b(s) e^{-\int_s^{x_0} a(t) dt} ds \right)$$

$$\left. \begin{aligned} y'(x) &= a(x)y(x) + b(x) \\ y(x_0) &= y_0 \end{aligned} \right\}$$

La soluzione del problema di Cauchy:

Figura 3.7: Esempio di figura senza immagine!

Capitolo 4

Personalizzazione delle didascalie

Benché la personalizzazione delle didascalie sia, in generale, rara e dunque poco interessante, tratteremo, per completezza, tale argomento seppur in maniera molto sintetica. Per una trattazione esaustiva riamandiamo il lettore al documento [Som04].

Il pacchetto `caption2` è recentemente divenuto obsoleto in quanto sostituito dalla versione 3 del pacchetto `caption`, trattato nella Sezione 4.2. La nuova versione del pacchetto `caption`, rilasciata il 16 luglio 2004, mette a disposizione nuove opzioni e caratteristiche e sostituisce le precedenti versioni, perciò si potrebbe ignorare la Sezione 4.1, e concentrare la propria attenzione esclusivamente sulla Sezione 4.2. Purtroppo, essendo il rilascio del nuovo pacchetto molto recente, può essere utile riferirsi anche ad una precedente versione del medesimo pacchetto, sia perché la distribuzione \TeX che si utilizza potrebbe non essere stata aggiornata con la versione più recente disponibile, sia perché la versione 3 del pacchetto `caption` è comunque in grado di interpretare tutte le opzioni ed i comandi riconosciuti dalle precedenti versioni.

4.1 Il “vecchio” pacchetto `caption`

Per includere tale pacchetto è necessario inserire nel preambolo:

```
\usepackage[opzioni]{caption}
```

in cui le principali opzioni supportate sono riportate nella Tabella 4.1.

4.1.1 Un primo esempio

Il seguente esempio:

Opzione	Selezione ...	
	... <i>l'allineamento della didascalia.</i>	
“ <i>Style</i> ” {	normal center centerlast flushleft flushright hang indent	vedi Figura 4.1 vedi Figura 4.2 vedi Figura 4.3 definibile solo globalmente definibile solo globalmente vedi Figura 4.4 definibile solo globalmente
	... <i>la dimensione del carattere per l'etichetta ed il testo.</i>	
“ <i>Font size</i> ” {	scriptsize footnotesize small normalsize large Large	molto piccolo abbastanza piccolo piccolo normale grande molto grande
	... <i>lo stile del carattere dell'etichetta.</i>	
“ <i>Font Family</i> ” {	up it sl sc rm sg tt md bf	upright (tondo) <i>italic (corsivo)</i> <i>slanted (inclinato)</i> SMALL CAPS SHAPE (MAIUSCOLETTO) roman sans serif typewriter medium (normale) bold font (grassetto)

Tabella 4.1: Lista delle opzioni più comuni per il “vecchio” pacchetto caption



Figura 4.1: Normal Caption Style. Normal Caption Style.



Figura 4.2: Center Caption Style. Center Caption Style.



Figura 4.3: Centerlast Caption Style. Centerlast Caption Style.



Figura 4.4: Hang Caption Style. Hang Caption Style.

```
\usepackage[bf, small]{caption}
```

combina le opzioni “bf” e “small” illustrate nella Tabella 4.1 e tali opzioni vengono applicate a tutte le didascalie presenti nel documento, escluse quelle ridefinite localmente (si veda Sezione 4.1.2).

4.1.2 Ridefinizione locale

È sconsigliabile ricorrere alla ridefinizione locale delle didascalie poiché la non omogeneità del documento (in particolare delle impostazioni delle didascalie) è poco gradevole. Comunque, per completezza di trattazione, di seguito è riportato un semplice esempio:

```
----->-----
\begin{figure}[htbp]
  \centering
  \makeatletter
  \let\as@caption\as@isucaption
  \makeatother
  \includegraphics[width=.4\textwidth]{eps/penguin.eps}
```

```
\caption{Normal Caption Style.}
\end{figure}
```

----->&

Questo esempio può essere facilmente compreso ricorrendo alla documentazione ufficiale del manuale [Som04].

4.2 Il pacchetto `caption`, versione 3

Per includere il pacchetto `caption` versione 3, è necessario inserire nel preambolo:

```
\usepackage[opzioni]{caption}[2004/07/16]
```

in cui “opzioni” è una stringa di parametri opzionali i quali, se dichiarati, operano su tutte le didascalie del documento. Di seguito è illustrato un esempio tratto da [Som04] che mostra la potenzialità espressiva delle opzioni riconosciute dal pacchetto `caption` per la personalizzazione delle didascalie:

```
\usepackage[margin=10pt, font=small, labelfont=bf]{caption}
```

Benché sia possibile personalizzare localmente¹ una didascalia, ciò è generalmente sconsigliato in quanto la non omogeneità del documento risulta spesso poco gradevole. Comunque, di seguito è riportato per completezza un semplice esempio:

```
----->&-----
\begin{figure}[htbp]
  \centering
  \includegraphics[width=.4\textwidth]{eps/penguin.eps}
  \captionsetup{font=medium}
  \caption{Font alterato}
\end{figure}
```

----->&-----

in cui la sola figura in esame ha una didascalia personalizzata; è opportuno notare che il comando `\captionsetup` ha effetto esclusivamente nell’ambiente definito, quindi, è possibile utilizzarlo in alternativa alla definizione globale definita precedentemente. Ciò significa che i seguenti frammenti di codice:

¹cioè alterare lo stile di una o più didascalie e fare in modo che alle restanti sia applicato lo stile scelto in fase di caricamento del pacchetto.

```
\usepackage[margin=10pt, font=small, labelfont=bf]{caption}
```

e

```
\usepackage{caption}
\captionsetup[margin=10pt, font=small, labelfont=bf]
```

producono lo stesso risultato.

4.2.1 Esempi

Nella Tabella 4.2 sono elencate le opzioni più comuni per il pacchetto `caption`, versione 3.

Opzione	Selezione ...
Formatting	
$ \text{"format"} \left\{ \begin{array}{l} \text{default} \\ \text{hang} \end{array} \right. $	<p>... <i>la formattazione della didascalia.</i></p> <p>opzione di default indenta la didascalia</p>
$ \text{"labelformat"} \left\{ \begin{array}{l} \text{empty} \\ \text{simple} \\ \text{parens} \end{array} \right. $	<p>... <i>la formattazione dell'etichetta.</i></p> <p>nessuna etichetta opzione di default numero della didascalia tra parentesi</p>
$ \text{"labelsep"} \left\{ \begin{array}{l} \text{none} \\ \text{colon} \\ \text{period} \\ \text{space} \\ \text{quad} \\ \text{newline} \end{array} \right. $	<p>... <i>il separatore.</i></p> <p>nessun separatore separatore di default un punto e uno spazio un singolo spazio inserisce un <code>\quad</code> inserisce un <code>\newline</code></p>

... continua ...

... continua ...

Giustificazione

		... <i>l'allineamento della didascalia.</i>	
"Justification"	{	justified	vedi Figura 4.1
		centering	vedi Figura 4.2
		centerlast	vedi Figura 4.3
		centerfirst	solo la prima linea è centrata
		raggedright	allineamento a sinistra
		RaggedRight	simile alla precedente
		raggedleft	allineamento a destra

Font, labelfont, textfont

		... <i>la dimensione del carattere per l'etichetta ed il testo.</i>	
"Font size"	{	scriptsize	molto piccolo
		footnotesize	abbastanza piccolo
		small	piccolo
		normalsize	normale
		large	grande
		Large	molto grande
		... <i>lo stile del carattere dell'etichetta.</i>	
"Font Family"	{	up	upright (tondo)
		it	<i>italic (corsivo)</i>
		sl	<i>slanted (inclinato)</i>
		sc	SMALL CAPS SHAPE (MAIUSCOLETTO)
		rm	roman
		sg	sans serif
		tt	typewriter
		md	medium (normale)
bf	bold font (grassetto)		

Tabella 4.2: Lista delle opzioni più comuni per il pacchetto caption, versione 3

Presentiamo di seguito alcuni esempi della sintassi da utilizzare mostrando per ciascuno di essi i risultati ottenuti.

Esempio I

```
\captionsetup{format=hang, indentation=-2cm}
```

produce una didascalia del tipo:

Figura 4.5: Per entrambi i formati `default` e `hang` è possibile specificare il parametro `indentation` con il quale è possibile modificare la dimensione del rientro delle linee della didascalia successive alla prima

Esempio II

```
\captionsetup{labelformat=parens, labelsep=period}
```

produce una didascalia del tipo:

Figura (4.6). E' possibile impostare `labelformat` a `empty`, `simple` (valore di default) oppure `parens`, ed è possibile impostare `labelsep` a `none`, `colon`, `period`, `space`, `quad` oppure `newline`

Esempio III

```
\captionsetup{labelsep=newline, singlelinecheck=false}
```

produce una didascalia del tipo:

Figura 4.7
E' possibile impostare `singlelinecheck` a `true` (valore di default) oppure `false`

Esempio IV

```
\captionsetup{labelsep=newline, singlelinecheck=true,
              justification=centerlast}
```


produce una didascalia del tipo:

Figura 4.8

E' possibile impostare `justification` a `justified`, `centering`, `centerlast`, `centerfirst`, `raggedright`, `RaggedRight` oppure `raggedleft`

Esempio V

```
\captionsetup{font={small,it}, labelfont=bf}
```

produce una didascalia del tipo:

Figura 4.9: Le opzioni `font` e `labelfont` permettono di alterare l'aspetto dei caratteri utilizzati per la didascalia e per l'etichetta della didascalia, rispettivamente

Esempio VI

```
\captionsetup{margin=15pt}
```

produce una didascalia del tipo:

Figura 4.10: Con `margin` è possibile alterare l'ampiezza dello spazio inserito orizzontalmente tra la didascalia ed i margini dell'area di stampa

Esempio VII

```
\captionsetup{width=.6\textwidth}
```

produce una didascalia del tipo:

Figura 4.11: Con `width` è possibile impostare da larghezza complessiva della didascalia

Capitolo 5

Alcuni concetti avanzati

5.1 Creazione di grafici con xfig

xfig è un programma molto semplice ma potente che permette di creare velocemente grafici. La Figura 5.1 mostra una tipica schermata di xfig. xfig permette

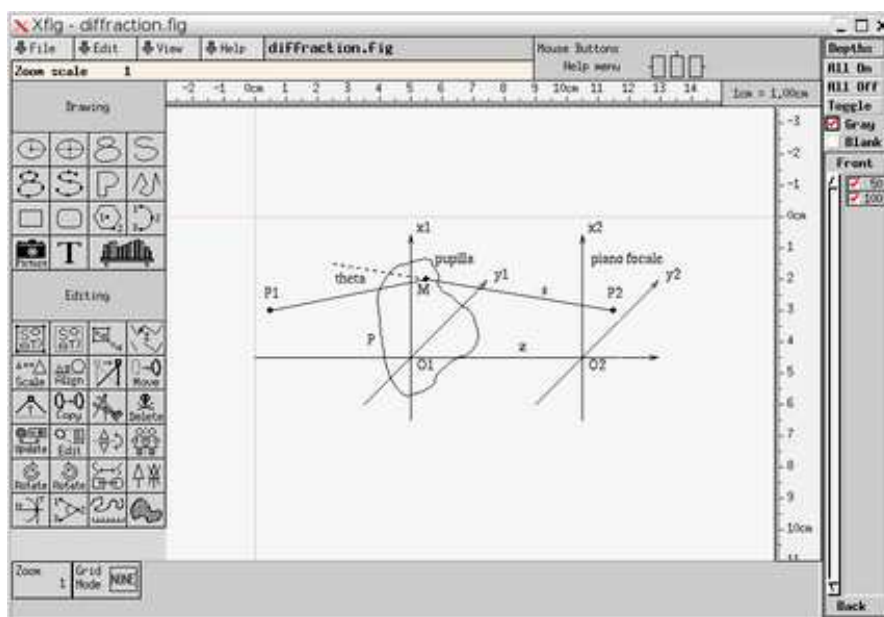


Figura 5.1: Esempio d'uso del programma xfig

di salvare un grafico nel formato proprietario .fig, che, per essere inserito in un

documento \LaTeX , deve essere convertito in un file con estensione `.eps`. Vi sono due modi:

- esportare mediante `xfig` il file in formato EPS;
- convertire il file `.fig` in `.eps` mediante il programma `fig2dev`.

In particolare, se si vuole optare per la seconda soluzione, la sintassi del comando `fig2dev` è:

```
fig2dev -L eps nomefile.fig nomefile.eps
```

Questa soluzione è particolarmente comoda nel caso di documenti molto complessi e che contengono molti grafici creati con `xfig`: infatti, utilizzando il programma `make` e scrivendo un opportuno `Makefile`, è possibile automatizzare la creazione di ogni file `.eps` indicando un'opportuna regola e specificando le dipendenze tra i file. Per indicazioni più dettagliate, si rimanda alla Sezione A.3 ed alla documentazione del programma `make`.

5.2 Il pacchetto `psfrag`

Benché `xfig` permetta di inserire testo all'interno di grafici, molto spesso tale soluzione non è soddisfacente: si vorrebbe aver a disposizione tutta la potenza di \LaTeX per poter inserire formule matematiche e, allo stesso tempo, si vorrebbe sfruttare la praticità di `xfig` per la creazione di grafici.

Il pacchetto `psfrag` permette di sopperire alle limitazioni tipografiche di `xfig`. Infatti, con `psfrag` è possibile sostituire del testo scritto in un grafico creato con `xfig` con un frammento di testo (o formula matematica) scritto con \LaTeX . Il pacchetto `psfrag` mette a disposizione il comando `\psfrag` la cui sintassi (semplificata) è:

```
\psfrag{vecchio}{nuovo}
```

dove “vecchio” indica il testo che si vuole rimpiazzare con l'espressione \LaTeX “nuovo”. Le opzioni del comando `\psfrag` permettono di impostare l'allineamento del testo, un fattore di scala (per ottenere ingrandimenti/riduzioni) ed un angolo espresso in gradi che permette di ruotare il frammento di testo da sostituire.

Per chiarire meglio le idee, si consideri la Figura 5.2 nella quale è riportato un grafico creato con `xfig`; la Figura 5.3 mostra il risultato ottenuto rimpiazzando il testo originale con del testo \LaTeX . Il codice che ha permesso di ottenere il risultato di Figura 5.3 è:

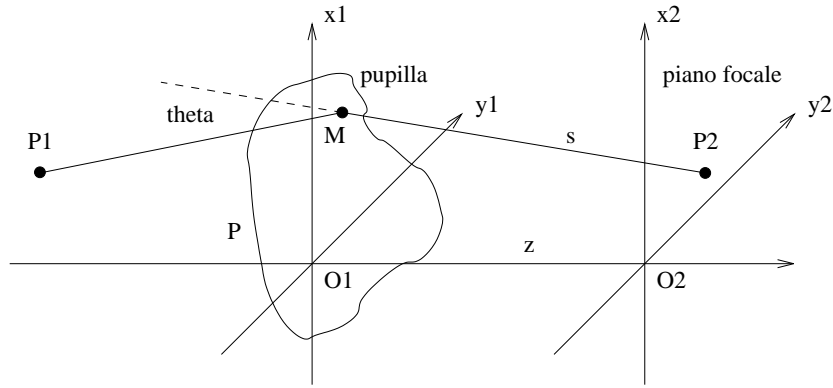


Figura 5.2: Grafico “nudo”, creato con xfig

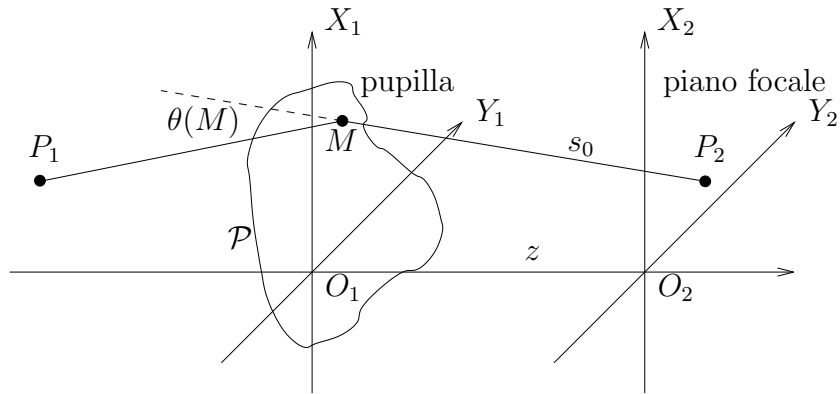


Figura 5.3: Risultato ottenuto con il pacchetto psfrag

```

----->
\begin{figure}[htbp]
  \centering
  {
    \psfrag{x1}{ $X_1$ }\psfrag{x2}{ $X_2$ }
    \psfrag{y1}{ $Y_1$ }\psfrag{y2}{ $Y_2$ }
    \psfrag{01}{ $0_1$ }\psfrag{02}{ $0_2$ }
    \psfrag{P1}{ $P_1$ }\psfrag{P2}{ $P_2$ }
    \psfrag{s}{ $s_0$ }\psfrag{z}{ $z$ }
    \psfrag{theta}{ $\theta_{r, s}(M)$ }
    \psfrag{P}{ $\mathcal{P}$ }
    \psfrag{piano focale}{piano focale}
    \psfrag{pupilla}{pupilla}
    \psfrag{M}{ $M$ }
    \includegraphics[scale=.80]{fig/diffraction.eps}
  }
  \caption{Risultato ottenuto con il pacchetto
    \textsf{psfrag}\label{fig:diffractionXFIG}}
\end{figure}
----->

```

Facciamo osservare l'uso delle parentesi graffe per poter “rendere locali” le sostituzioni ed evitare che esse vengano inavvertitamente applicate anche alle eventuali altre immagine inserite nella stessa figura. Inoltre, il pacchetto `psfrag` può essere utilizzato solo con immagini EPS “vettoriali”: non ha alcun effetto su immagini EPS “bitmap” ottenute, ad esempio, convertendo un'immagine JPEG in EPS. Mettiamo inoltre in guardia il lettore del fatto che alcuni visualizzatori di file `.dvi` (come, ad esempio, il programma `xdvi`) *non* supportano il pacchetto `psfrag`; in tali casi è dunque necessario creare il file POSTSCRIPT/PDF per visionare correttamente le sostituzioni ottenute con `psfrag`.

5.3 Grafico di funzioni matematiche

Presentiamo ora due possibili modi per poter inserire all'interno di un documento \LaTeX il grafico di una funzione della quale è nota l'espressione matematica esplicita. In tal caso, è, infatti, molto più comodo e veloce utilizzare un programma esterno che disegni per noi esattamente il grafico della funzione data e che permetta di ottenere il file in formato EPS che andrà poi incluso nel documento che si sta scrivendo.

Un primo modo per ottenere il grafico di una funzione matematica è quello

di utilizzare l'ambiente di calcolo scientifico *Mathematica*.¹ Con *Mathematica* è possibile realizzare una vasta gamma di tipi di grafici ed è possibile esportare direttamente il grafico creato in formato EPS. Inoltre, le primitive e gli algoritmi di *Mathematica* sono molto potenti e ricchi di opzioni e permettono di ottenere grafici molto precisi. A titolo di esempio, riportiamo il codice *Mathematica* che permette di ottenere il grafico della funzione:

$$f: \mathbb{R} \rightarrow \mathbb{R}, \quad f(x) = e^{\sqrt{|x|} - \sin(x^3)}$$

con $x \in [-5/2, 5/2]$:

```
----->⌘-----
figure = Plot[Exp[Sqrt[Abs[x]] - Sin[x^3]], {x, -5 / 2, 5 / 2},
  MaxBend -> 1 / 10, PlotDivision -> 50,
  PlotPoints -> 50, PlotStyle -> RGBColor[1, 0, 0]
];
Export["function_1.eps", figure, "EPS"];
----->⌘-----
```

Il codice indicato mostrerà a video il grafico della funzione $f(x)$ per l'intervallo indicato e, tale grafico, sarà anche salvato nel file `function_1.eps`. La Figura 5.4 mostra ciò che si ottiene includendo tale file in un documento \LaTeX .

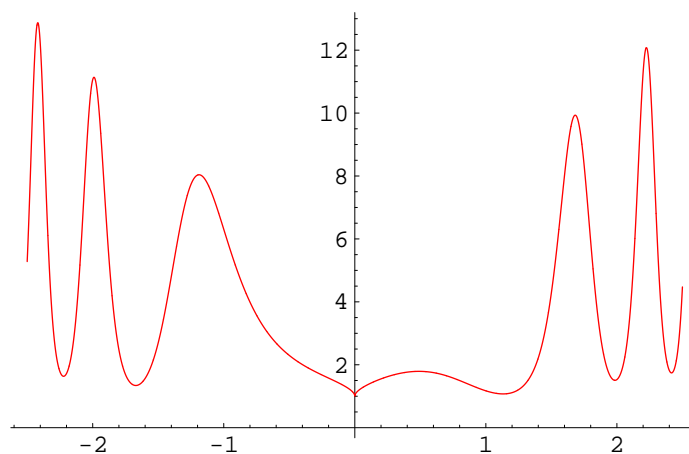


Figura 5.4: Grafico di una funzione creato con *Mathematica* e salvata direttamente in formato EPS

¹Pagina web: <http://www.wolfram.com/>.

In ambiente GNU/LINUX, una possibile alternativa all'uso di *Mathematica* è quella di utilizzare il programma gratuito `gnuplot`² disponibile nella maggior parte delle distribuzioni GNU/LINUX. `gnuplot` è un programma molto semplice che consente di esportare un grafico anche in formato EPS. Inoltre, permette di creare tutti i tipi di grafici di uso più frequente. Anche in questo caso, riportiamo un esempio di codice `gnuplot`; la funzione presa in considerazione è la stessa dell'esempio precedente e viene graficata per il medesimo intervallo dell'asse reale.

```
----->⌘-----  
set xrange [-2.50:2.50]  
set samples 10000  
set output "function_2.eps"  
set terminal postscript eps color  
plot exp(sqrt(abs(x)) - sin(x**3))  
----->⌘-----
```

Il codice indicato esporta direttamente il grafico nel file `function_2.eps` senza visualizzare nulla a video. La Figura 5.5 mostra il grafico così ottenuto.

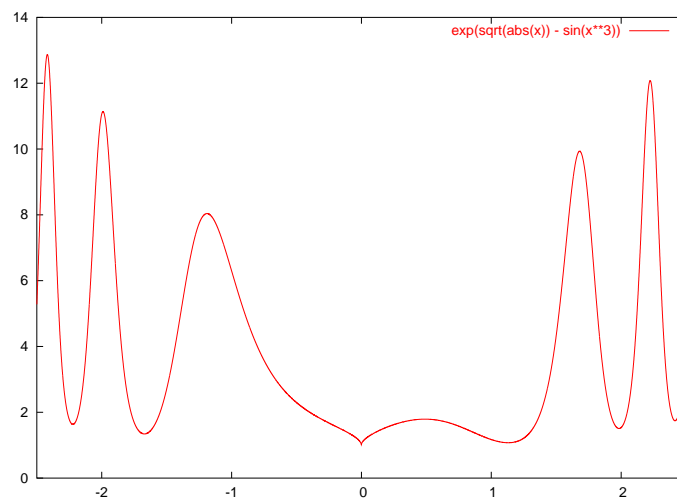


Figura 5.5: Grafico di una funzione creato con `gnuplot` e salvata direttamente in formato EPS

²Pagina web: <http://www.gnuplot.info/>.

5.4 Gestire file non-EPS

Le immagini che \LaTeX permette di inserire direttamente in un documento devono essere necessariamente in formato `.eps`. Sotto GNU/LINUX, è possibile utilizzare il comando `convert` per convertire un'immagine in formato non-EPS in un'immagine in formato EPS. La sintassi è molto semplice:

```
convert immaginenonEPS.ext immagine.eps
```

Se si lavora con altri sistemi operativi, non esiste una soluzione generale. Tuttavia, nella maggior parte dei casi, ogni programma di fotoritocco permette di salvare un'immagine in svariati formati, tra i quali anche EPS.

In generale, per immagini o grafici vettoriali (cioè non bitmap) che stiamo creando personalmente, è consigliabile utilizzare un programma che permetta di salvare *direttamente* il grafico creato in formato EPS evitando così di dover ricorrere a conversioni di formato che potrebbero ridurre la qualità dell'immagine abbassandone la risoluzione. Confrontando la Figura 5.6(a) con la Figura 5.6(b) si nota quale sorta di degrado e perdita di dettaglio³ può risultare in seguito ad una conversione di un'immagine bitmap in EPS.

Per eventuali approfondimenti riguardanti gestione e conversione di immagini in formati diversi e loro inserimento in documenti \LaTeX è possibile consultare la Sezione 9: “Preparazione e gestione delle immagini” di [PLA03].

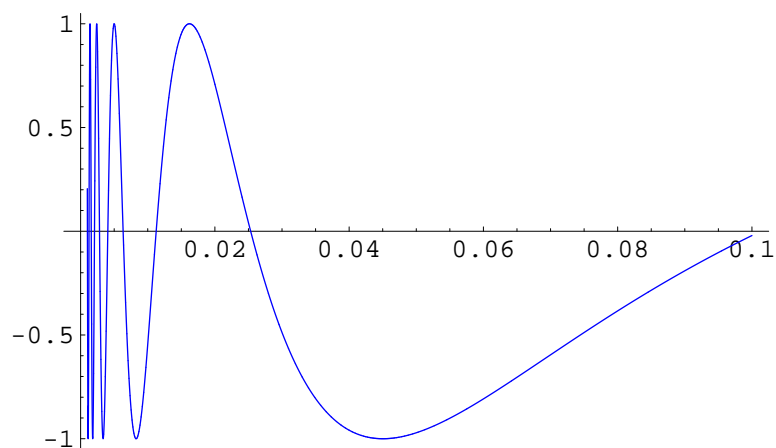
5.5 pdf \LaTeX e `\DeclareGraphicsExtensions`

Se si utilizza pdf \LaTeX per compilare un documento e creare direttamente il file in formato PDF, è possibile aggirare la restrizione sul formato delle immagini descritta nella sezione precedente e fare in modo che altri formati grafici possano essere usati senza ricorrere a conversioni.

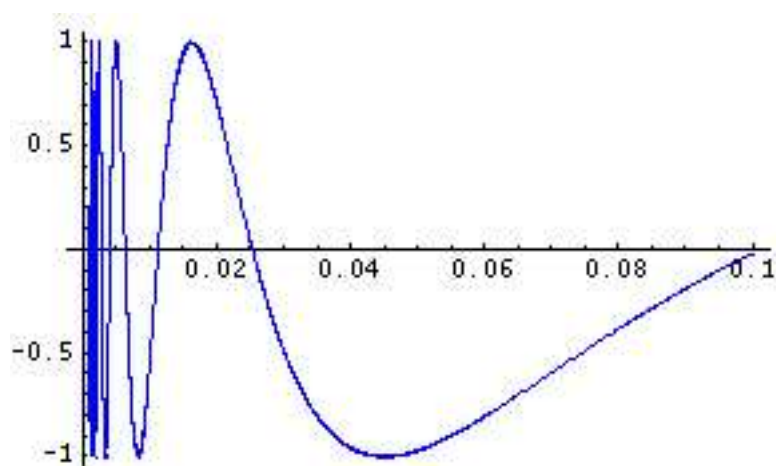
In tali circostanze, è possibile aggiungere nel preambolo un comando in cui si dichiara la lista delle estensioni che le immagini possono avere, cosicché, successivamente non si dovrà indicare esplicitamente l'estensione dell'immagine ma pdf \LaTeX farà una ricerca seguendo l'ordine con cui le loro estensioni sono state elencate. Il comando con il quale è possibile indicare l'elenco delle possibili estensioni è `\DeclareGraphicsExtensions`.

L'esempio seguente sarà chiarificatore:

³La funzione matematica considerata è stata scelta in modo da mettere in particolare evidenza i problemi che possono insorgere.



(a) Immagine salvata direttamente in formato EPS



(b) Immagine salvata in formato TIFF e poi convertita in formato EPS

Figura 5.6: Esempio di degradazione introdotta esportando un grafico in un formato grafico bitmap e convertendolo poi in un file EPS: graficando la funzione $f: \mathbb{R} \setminus \{0\} \rightarrow [-1, 1]$ definita da $f(x) = \sin(1/\sqrt{|x|})$ per $x \in [1/1000, 1/10]$ si può notare la perdita di dettagli che si ha, in particolare, per $x \rightarrow 0$

```

----->⌘-----
\documentclass[pdflatex, 11pt, a4paper]{article}
\usepackage{graphicx}
\DeclareGraphicsExtensions{.png, .pdf}
\begin{document}
  ...
  ...
  ...
\end{document}
----->⌘-----

```

Se, ad esempio, successivamente a tale dichiarazione si vuole inserire l'immagine contenuta nel file `filename.pdf`, si potrà scrivere al posto di `filename.pdf` semplicemente `filename`: pdf \LaTeX cercherà `filename.png` e, qualora non esistesse, passerà alla successiva estensione dichiarata, quindi `filename.pdf`.

5.6 “Elenco delle figure” e “Elenco delle tabelle”

In documenti molto lunghi (libri, tesi, ...) è molto comune inserire dopo l'indice anche gli elenchi delle figure e delle tabelle. \LaTeX permette di gestire automaticamente la creazione/aggiornamento di tali parti del documento e, mediante poche linee di codice \LaTeX , è possibile creare l'indice e gli elenchi delle figure e tabelle. Riportiamo la sequenza di comandi \LaTeX che permette di generare l'indice, l'elenco delle figure e l'elenco delle tabelle:

```

----->⌘-----
\addcontentsline{toc}{chapter}{\numberline{}}\contentsname}
\tableofcontents
\addcontentsline{toc}{chapter}{\numberline{}}\listfigurename}
\listoffigures
\addcontentsline{toc}{chapter}{\numberline{}}\listtablename}
\listoftables
----->⌘-----

```

I comandi `\tableofcontents`, `\listoffigures` e `\listoftables` generano rispettivamente, l'indice, l'elenco delle figure e l'elenco delle tabelle. Il comando `\addcontentsline` permette di inserire una nuova voce nell'indice. Nell'esempio mostrato, il riferimento all'indice, quello all'elenco delle figure e quello all'elenco delle tabelle verranno inseriti nell'indice in questo ordine.

La sintassi generale del comando `\addcontentsline` è:

```
\addcontentsline{elenco}{livello}{voce}
```

dove “elenco” indica in quale lista si vuole inserire una certa voce: “toc” (“Table Of Contents”) indica l’indice; “lof” (“List Of Figures”) indica l’elenco delle figure mentre “lot” (“List Of Tables”) indica l’elenco delle tabelle. Il parametro “livello” può essere `chapter`, `section` oppure `subsection` ed influenza la spaziatura orizzontale tra il margine sinistro dell’area di stampa e la voce che si vuole inserire. Infatti, osservando l’indice di un documento L^AT_EX, si noter  che le voci sono gerarchicamente “indentate”: le voci relative ai capitoli sono messe in maggior evidenza facendo rientrare verso destra le voci relative alle sezioni le quali, a loro volta, vengono evidenziate facendo ulteriormente rientrare le voci relative alle sottosezioni.

Il parametro “voce”   la porzione di testo che si vuole inserire come voce. Nell’esempio riportato sopra,   stato utilizzato il comando `\numberline` che permette di inserire il numero di un capitolo, di una sezione oppure di una sottosezione in rettangolo di larghezza prefissata; se tale numero non   definito (come nel caso mostrato), con `\numberline{}` si ottiene l’inserimento di un’opportuna quantit  di spazio in modo tale che le prime lettere di tutte le voci di parti di documento allo stesso livello, siano verticalmente allineate. Facciamo notare l’impiego dei comandi `\contentsname`, `\listfigurename` e `\listtablename` per questioni di portabilit .

Ogniqualevolta si utilizza il comando `\caption` per creare una didascalia di una figura oppure di una tabella, una nuova voce (relativa alla figura/tabella appena creata) viene aggiunta all’elenco delle figure/tabelle. Normalmente, il testo che viene riportato nella voce   costituito dalla didascalia stessa la quale pu  essere lunga anche diverse linee. Per evitare che voci troppo lunghe appaiano nell’elenco delle figure/tabelle⁴   possibile passare a `\caption` un parametro opzionale contenente una versione “corta” della didascalia; la sintassi completa del comando `\caption`  :

```
\caption[corta]{lunga}
```

in cui “corta”   il frammento di testo che sar  utilizzato nell’elenco delle figure/tabelle e “lunga”   la didascalia vera e propria della figura/tabella.

⁴Voci troppo lunghe potrebbero risultare “dispersive” in quanto forniscono pi  informazioni di quelle necessarie in un elenco delle figure/tabelle.

5.7 Cornici attorno alle figure con PSTricks

Benché \LaTeX metta a disposizione il comando `\fbox` per creare cornici attorno ad oggetti, questo comando risulta inadeguato quando si vogliono ottenere risultati più evoluti. Per questo motivo, sono stati sviluppati diversi pacchetti tra i quali citiamo `fancybox` e l'insieme di pacchetti grafici noti sotto il nome "PSTricks". In particolare, `PSTricks`⁵ permette di ottenere risultati molto raffinati, di cui diamo qui un breve accenno.

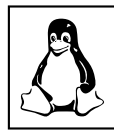
Per utilizzare i comandi base di `PSTricks` è sufficiente includere il pacchetto `pstricks`. Sono disponibili altri pacchetti di uso meno frequente che possono essere caricati singolarmente oppure "in blocco" includendo semplicemente il pacchetto `pst-all`. È consigliabile includere solo quei pacchetti di `PSTricks` strettamente necessari risparmiando così della memoria.

I comandi `PSTricks` per la creazione di cornici sono:

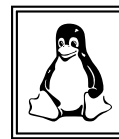
- "`\psframebox`": disegna una cornice rettangolare doppia attorno all'oggetto che viene passato come parametro (vedi Figura 5.7(a));
- "`\psdblframebox`": disegna una cornice rettangolare semplice attorno all'oggetto che viene passato come parametro (vedi Figura 5.7(b));
- "`\psshadowbox`": disegna una cornice rettangolare ombreggiata attorno all'oggetto che viene passato come parametro (vedi Figura 5.7(c));
- "`\pscircularbox`": disegna una cornice circolare attorno all'oggetto che viene passato come parametro (vedi Figura 5.7(d));
- "`\psovalbox`": disegna una cornice ovale attorno all'oggetto che viene passato come parametro (vedi Figura 5.7(e)).

Inoltre, i comandi `PSTricks` elencati, accettano una o più opzioni con le quali è possibile alterare le caratteristiche e lo stile della cornice creata. Proseguiamo ora descrivendo alcune opzioni che possono risultare utili quando si devono creare cornici attorno ad oggetti.

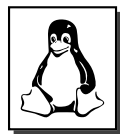
⁵Pagina web: <http://www.pstricks.de/>.



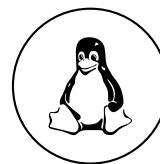
(a) Cornice ottenuta con il comando `\psframebox`



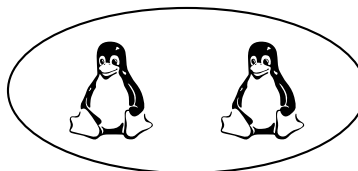
(b) Cornice ottenuta con il comando `\psdblframebox`



(c) Cornice ottenuta con il comando `\psshadowbox`


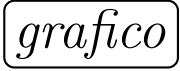






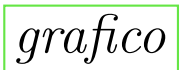
(d) Cornice ottenuta con il comando `\pscirclebox`

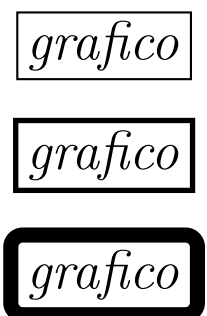



(e) Cornice ottenuta con il comando `\psovalbox`

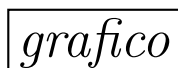
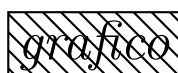
Figura 5.7: Esempi di cornici realizzate con PSTricks

	<p>L'opzione “<code>framearc</code>” permette di impostare il raggio dell'arco di circonferenza utilizzato per gli angoli “arrotondati” della cornice. Il valore impostato mediante <code>framearc</code> deve essere un numero r compreso tra 0 ed 1 (inclusi) ed il valore del raggio dell'arco di circonferenza viene scelto pari a $r/2$ moltiplicato per il minimo tra la larghezza e l'altezza complessiva della cornice.</p> <p>Esempio:</p> <pre>\newcommand{\myimage}{\LARGE\textit{grafico}} \psframebox{\myimage} \psframebox[framearc=0.30]{\myimage} \psdblframebox[framearc=0.30]{\myimage} \psframebox[framearc=0.70]{\myimage}</pre>
	
	
	

	<p>L'opzione “<code>linecolor</code>” permette di scegliere il colore con cui disegnare la cornice. Tale colore può essere uno di quelli predefiniti di PSTricks oppure può anche essere definito dall'utente mediante, ad esempio, i comandi PSTricks <code>\newgray</code>, <code>\newrgbcolor</code>, <code>\newhsbcolor</code> oppure <code>\newcmykcolor</code>. Il colore di default è il nero.</p> <p>Esempio:</p> <pre>\newcommand{\myimage}{\LARGE\textit{grafico}} \newhsbcolor{mycolor}{0.30 0.70 0.90} \psframebox{\myimage} \psdblframebox[linecolor=blue]{\myimage} \psframebox[linecolor=mycolor]{\myimage}</pre>
	
	

	<p>L'opzione “linewidth” permette di scegliere lo spessore della linea con cui disegnare la cornice. Il valore scelto può venir espresso mediante una qualsivoglia unità di misura riconosciuta da L^AT_EX. Il valore di default è pari a 0.80 punti corrispondente a circa 0.28 millimetri.</p> <p>Esempio:</p> <pre> \newcommand{\myimage}{\LARGE\textit{grafico}} \psframebox{\myimage} \psframebox[linewidth=2pt]{\myimage} \psframebox[linewidth=2mm, framearc=0.30]{\myimage} </pre>
---	--

	<p>L'opzione “linestyle” permette di impostare lo stile con cui disegnare le linee della cornice. Possibili stili sono solid (linea continua), dashed (linea tratteggiata), dotted (linea realizzata con punti) e none (nessuno stile). Il valore di default è solid. Se si utilizza lo stile dashed è possibile alterare il tipo di tratteggio mediante l'opzione “dash” in cui il primo valore indica la lunghezza di ogni tratto mentre il secondo indica la distanza tra un tratto ed il successivo I valori di default sono 5 e 3 punti, rispettivamente. Se, invece, si utilizza lo stile dotted è possibile indicare la distanza tra i punti mediante l'opzione “dotsep” il cui valore di default è pari a 3 punti.</p> <p>Esempio:</p> <pre> \newcommand{\myimage}{\LARGE\textit{grafico}} \psframebox{\myimage} \psframebox[linestyle=dashed]{\myimage} \psdblframebox[linestyle=dashed]{\myimage} \psframebox[linestyle=dashed, dash=2pt 6pt]{\myimage} \psframebox[linestyle=dotted]{\myimage} \psframebox[linestyle=dotted, dotsep=1pt]{\myimage} </pre>
--	---

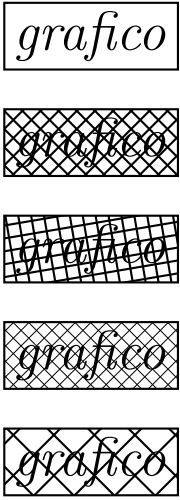



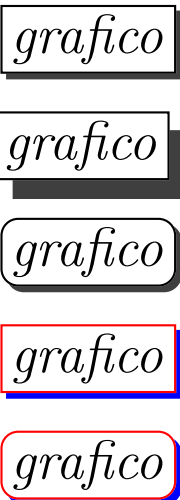



L'opzione “fillstyle” permette di impostare lo stile con cui riempire l'area racchiusa dalla cornice. Lo stile `vlines` corrisponde ad un motivo costituito da linee (eventualmente inclinate). Il parametro `hatchangle` permette di impostare l'angolo di inclinazione delle linee (il valore di default è 45 gradi), `hatchwidth` permette di variare lo spessore delle linee (il valore di default è $\frac{4}{5}$ di punto), mentre `hatchsep` permette di alterare la distanza tra le linee (il valore di default è 4 punti). È anche possibile utilizzare lo stile `hlines`: con `vlines` ad un angolo di inclinazione nullo corrispondono linee verticali mentre con `hlines` si ottengono linee orizzontali.

Esempio:

```
\newcommand{\myimage}{\LARGE\textit{grafico}}
\psframebox{\myimage}
\psframebox[fillstyle=vlines]{\myimage}
\psframebox[fillstyle=vlines,
             hatchangle=280]{\myimage}
\psframebox[fillstyle=vlines,
             hatchwidth=.1pt]{\myimage}
\psframebox[fillstyle=vlines,
             hatchsep=10pt]{\myimage}
```


	<p>Lo stile <code>crosshatch</code> corrisponde ad un motivo costituito da linee incrociate. Il parametro <code>hatchangle</code> permette di impostare l'angolo di inclinazione delle linee (il valore di default è 45 gradi), <code>hatchwidth</code> permette di variare lo spessore delle linee (il valore di default è $\frac{4}{5}$ di punto), mentre <code>hatchsep</code> permette di alterare la distanza tra le linee (il valore di default è 4 punti).</p> <p>Esempio:</p> <pre> \newcommand{\myimage}{\LARGE\textit{grafico}} \psframebox{\myimage} \psframebox[fillstyle=crosshatch]{\myimage} \psframebox[fillstyle=crosshatch, hatchangle=280]{\myimage} \psframebox[fillstyle=crosshatch, hatchwidth=.1pt]{\myimage} \psframebox[fillstyle=crosshatch, hatchsep=10pt]{\myimage} </pre>
---	---

	<p>Una cornice ombreggiata—ottenuta mediante il comando <code>\psshadowbox</code>—può essere personalizzata in diversi modi. È possibile alterare lo spessore dell'ombreggiatura mediante il parametro <code>shadowsize</code> (il valore di default è 3 punti) ed è possibile cambiare il colore dell'ombra (il colore di default è <code>darkgray</code>). È anche possibile utilizzare bordi arrotondati, cambiare il colore con cui viene disegnata la cornice e via dicendo.</p> <p>Esempio:</p> <pre> \newcommand{\myimage}{\LARGE\textit{grafico}} \psshadowbox{\myimage} \psshadowbox[shadowsize=10pt]{\myimage} \psshadowbox[framearc=.5]{\myimage} \psshadowbox[linecolor=red, shadowcolor=blue]{\myimage} \psshadowbox[framearc=.5, linecolor=red, shadowcolor=blue]{\myimage} </pre>
---	--

Appendice A

Alcune utili informazioni

A.1 Le unità di misura

Come mostrato negli esempi descritti nelle sezioni precedenti, è possibile stabilire dimensioni assolute ricorrendo all'utilizzo di diverse unità di misura; le unità di misura riconosciute da \LaTeX sono mostrate nella Tabella A.1; le equivalenze tra alcune di esse sono elencate nella Tabella A.2.

Abbreviazione	Unità di misura	Equivale a ...
"mm"	millimetro	└┐
"cm"	centimetro	└┐ ┌
"in"	"inch" (pollice)	└┐ ┌
"pt"	"point" (punto)	└┐
"pc"	"pica"	└┐ ┌
"bp"	"big point"	└┐
"dd"	"didot point"	└┐
"cc"	"cicero"	└┐ ┌
"sp"	"scaled point"	└┐
"em"	circa la larghezza di "M"	└┐ ┌
"ex"	circa l'altezza di "x"	└┐

Tabella A.1: Unità di misura utilizzabili in \LaTeX

\LaTeX è in grado di gestire lunghezze che vanno da 1 scaled point fino a 2^{30} scaled point (equivalente a circa 5.7583 metri). Le unità di misura `em` ed `ex` non sono "fisse" ma variano a seconda della dimensione del font corrente.

Lunghezza	Equivale a ...
1in	= 25.4 millimetri
1pt	= 1/72.27 di pollice \simeq 1/3 di millimetro
1pc	= 12 punti \simeq 4.22 millimetri
1bp	= 1/72 di pollice
1dd	= $1238/1157$ di punto \simeq 0.38 millimetri \simeq 1/3 di millimetro
1cc	= 12 didot point \simeq 4.51 millimetri
1sp	= 1/65536 di punto \simeq 53.63 Å (1 Å = 10^{-10} metri) \simeq 18 volte il diametro di una molecola d'acqua

Tabella A.2: Equivalenze tra unità di misura

A.2 Le “scatole”

Semplificando il concetto, L^AT_EX genera le pagine di un documenti manipolando quelle che vengono definite “scatole”; ad esempio, sono scatole:

- ogni lettera;
- ogni parola;
- l'ambiente `figure`;
- l'ambiente `minipage`.

Una parola è formata da più lettere (ognuna di esse è una scatola): le lettere di una stessa parola sono “incollate” per formare una scatola più grande, che corrisponde alla parola stessa. A sua volta, una parola (che è vista come una scatola) viene giustapposta alle altre parole per formare una scatola ancora più grande che è la linea. Questo procedimento continua sino a formare la scatola più grande: la pagina. Ovviamente il processo è molto più complesso di quello descritto poiché:


- se l'allineamento è giustificato, le parole saranno assemblate in maniera tale da occupare tutto lo spazio disponibile per la linea, cioè ogni parola sarà distanziata da una quantità di spazio la cui larghezza sarà decisa al momento della compilazione, affinché sia rispettata la giustificazione;

- alcune scatole (come quelle relative alle parole) possono essere “spezzate” (in corrispondenza di un’andata a capo) al fine di ottenere una impaginazione migliore;
- ogni pagina (la scatola più grande) è formata da diverse scatole, le quali si distinguono in “statiche” (come, ad esempio, quelle create con i comandi `\chapter`, `\section`, `\subsection`, `\subsubsection`, `\paragraph` e gli ambienti `tabular`, `minipage`, ... collocate rispettando la successione nel sorgente) e “dinamiche” (come, ad esempio, quelle corrispondenti agli ambienti `figure`, `table`, ... per le quali è possibile consigliare la collocazione tramite i parametri opzionali “h” e/o “t” e/o “b” e/o “p”, cercando, eventualmente, di forzarla tramite il parametro “!”).

Questo processo si compie all’oscuro dell’autore del documento \LaTeX , il quale può benissimo ignorarlo. Però, qualche volta potrebbe interessarci. Considerano quanto appena scritto, è facile comprendere che, generalmente, l’inserimento di una figura non si effettua dichiarando, ad esempio, il comando:

```
\includegraphics[scale=.10]{eps/penguin.eps}
```

tra il testo da visualizzare, in quanto produrrebbe un risultato analogo a quello

mostrato qui di seguito: . Pertanto, in generale, si preferisce inserire la figura da visualizzare all’interno di un’altra scatola, solitamente utilizzando l’ambiente dinamico `figure`.

A.3 Un esempio di Makefile per compilare documenti \LaTeX

Precedentemente si accennava al fatto che scrivendo un opportuno `Makefile` era possibile automatizzare la creazione di un file `POSTSCRIPT/PDF`: in questa sezione mostriamo un esempio di `Makefile` che può tornar utile in molti casi.

Semplificando il concetto, un `Makefile` non è altro che un file di testo contenente una lista di “regole” che indicano i “passi” da eseguire per poter creare un certo file. Una regola esprime un concetto del tipo: “per creare il file x è necessario creare prima il file y_1 compilando z_1 con il compilatore w_1 ed il file y_2 convertendo nel formato opportuno il file z_2 con il programma w_2 ”. Inoltre, le regole indicano anche le “dipendenze” tra i vari file di un progetto.

Consideriamo il seguente esempio di `Makefile` per la compilazione di un documento \LaTeX e la generazione del file `POSTSCRIPT` oppure `PDF`:

```
----->
# Makefile per compilare documenti con LaTeX.

MAIN          = paper
MAIN_TEX      = $(MAIN).tex
MAIN_DVI      = $(MAIN).dvi
MAIN_PS       = $(MAIN).ps
MAIN_PDF      = $(MAIN).pdf
SHELL         = /bin/sh

EPS           = eps/lena.eps
FIG           = fig/AZ-77.fig
EPS_FIG       = fig/AZ-77.eps

ALL_TEX       = chap1.tex chap2.tex chap3.tex $(MAIN_TEX)
ALL_FILES     = $(ALL_TEX) $(FIG) $(EPS) $(EPS_FIG)
CLEAN_FILES   = *~ *.dvi *.log *.aux *.toc *.lof *.lot \
               *.bbl *.blg *.idx *.ind *.ilg fig/*.bak
DISTCLEAN_FILES = $(MAIN_PS) $(MAIN_PDF) fig/*.eps
TODAY        = `date +%Y%m%d.%H%M%S`

.PHONY:       all dvi ps pdf compress clean distclean

all: $(MAIN_PS)
dvi: $(MAIN_DVI)
ps: $(MAIN_PS)
pdf: $(MAIN_PDF)

compress: distclean
        cd ..; tar cvfzps $(MAIN).$(TODAY).tar.gz $(MAIN)/

clean:
        rm -f $(CLEAN_FILES)

distclean: clean
        rm -f $(DISTCLEAN_FILES)

$(MAIN_DVI): $(ALL_FILES)
        texi2dvi -b -V $(MAIN_TEX)

$(MAIN_PS): $(MAIN_DVI)
        dvips -ta4 -e0 -Z -R -Ppdf -o $(MAIN_PS) $(MAIN_DVI)
```

```

$(MAIN_PDF): $(MAIN_PS)
    ps2pdf -sPAPERSIZE=a4 -dEmbedAllFonts=true          \
    -dMaxSubsetPct=100 -dCompatibilityLevel=1.2       \
    -dSubsetFonts=true $(MAIN_PS) $(MAIN_PDF)
    rm -f $(MAIN_PS)

fig/%.eps: fig/%.fig
    fig2dev -L eps $? $@
-----><-----

```

Il documento dell'esempio si compone di alcuni sorgenti \LaTeX ed alcune immagini. In particolare, una di queste immagini è un diagramma creato con `xfig` e pertanto, per poter essere utilizzata, deve essere prima convertita in formato EPS. Una opportuna regola di questo `Makefile` permette di creare automaticamente il file EPS tutte le volte che ve sia necessità. Inoltre, un file creato in seguito all'esecuzione di una regola viene automaticamente ricreato quando un file da cui esso dipende viene modificato.

Nel `Makefile` dell'esempio, la regola “di default” (cioè quella che viene eseguita invocando il comando `make` senza parametri) è quella per la creazione del file POSTSCRIPT. Sono definite anche altre regole; ad esempio con `make pdf` si ottiene la creazione del file PDF. Le regole `clean` e `distclean` permettono di rimuovere i file temporanei ed eventualmente anche i documenti POSTSCRIPT/PDF creati da \LaTeX , mentre la regola `compress` permette di archiviare in un file compresso con estensione `.tar.gz` tutti i file del documento \LaTeX in questione. Affinché tale regola possa essere utilizzata è necessario che tutti i file del documento si trovino in una directory il cui nome coincide con il nome del file `.tex` in cui è stato utilizzato il comando `\documentclass` (il nome di tale file dovrà venir associato alla variabile “MAIN” definita nel `Makefile`). Inoltre, il nome del file compresso creato conterrà la data e l'ora di creazione formattata in modo tale che una lista dei nomi di file compressi ordinata alfabeticamente rispetti l'ordine cronologico di creazione permettendo così una rapida individuazione dell'archivio compresso più recente.

A.4 Alcuni consigli

In questa sezione sono forniti alcuni consigli che potrebbero tornar utili quando si vuole creare una figura (o una tabella) in un documento \LaTeX .

- Utilizzare il pacchetto `placeins` con l'opzione `section` per forzare lo “svuotamento” della coda delle figure e tabelle prima dell'inizio della sezione successiva. Infatti, se per motivi estetici non è possibile inserire una figura/tabella

appena la si incontra nel corso della compilazione del file sorgente, \LaTeX inserisce temporaneamente tale figura/tabella in una coda di oggetti non ancora elaborati e nel prosieguo del processo di compilazione, controlla se uno di tali oggetti può essere disposto in un punto successivo del documento. Forzando lo svuotamento della coda di oggetti mobili prima dell'inizio di ogni sezione, si evita che una figura o tabella venga posta "troppo avanti" nel documento rendendo dunque scomoda la lettura. Tuttavia, nel caso di documenti con molte immagini e tabelle, forzare lo svuotamento della coda delle figure e tabelle prima dell'inizio della sezione successiva, potrebbe causare che alcune pagine vengano riempite solo parzialmente con del testo dando quindi luogo ad impaginazioni sgradevoli. In tali casi, è sconsigliabile l'uso di questo pacchetto.

- Il pacchetto `varioref` mette a disposizione una variante "arricchita" del comando `\ref` la quale, oltre ad indicare il riferimento, fornisce informazioni sulla pagina in cui l'oggetto referenziato appare. Tali informazioni aggiuntive permettono di ottenere in maniera automatica riferimenti del tipo "la Figura 3.7 a pagina 38" oppure "la Tabella 2.1 nella pagina precedente".
- Evitare, laddove possibile, l'uso di lunghezze e dimensioni espresse mediante unità di misura assolute: per questioni di portabilità, è preferibile utilizzare quantità espresse in unità di misura relative.
- È noto che \LaTeX potrebbe aver bisogno di più esecuzioni per completare la creazione di tutti i riferimenti presenti in un documento. Per questo motivo, in alcune distribuzioni GNU/LINUX è presente lo shell script `texi2dvi` che permette di ottimizzare questo processo eseguendo \LaTeX (e, se necessario, anche \BibTeX) il numero strettamente necessario di volte.
- In molte distribuzioni GNU/LINUX è presente il comando `import` che può essere utilizzato per "catturare" il contenuto di finestre (come nel caso della Figura 5.1) oppure di porzioni di schermo. Grande è la tentazione di usare questo comando ogniquale volta vi sia la necessità di includere in un documento un grafico riportato in un articolo pubblicato online; tuttavia, tale modo di procedere, andrebbe evitato. È preferibile ricreare tale grafico mediante, ad esempio, il programma `xfig` ed, eventualmente, ricorrere all'uso del pacchetto `psfrag` per potervi inserire delle formule matematiche. Procedere in questo modo ha diversi vantaggi: meno problemi di copyright nei confronti dell'autore/i dell'articolo; grafici vettoriali e quindi di alta qualità; consente di utilizzare notazioni diverse da quelle utilizzate nel grafico originale.

- Molte delle funzionalità sinora esposte utilizzano direttamente alcuni dei comandi del linguaggio POSTSCRIPT che potrebbero non essere supportati da certi visualizzatori di file `.dvi`. Ad esempio, `xdvi` non è in grado di interpretare i comandi del pacchetto `psfrag`. Analogamente, `pdfLATEX` non è in grado di interpretare alcuni comandi POSTSCRIPT. In questi casi, il modo più semplice per risolvere il problema è quello di creare il file POSTSCRIPT (estensione `.ps`) ed utilizzare un interprete POSTSCRIPT per la visualizzazione del documento. Se lo si desidera, tale file `.ps` può essere successivamente convertito in PDF mediante il comando GNU/LINUX `ps2pdf`.
- I file EPS creati con alcuni programmi (es: FreeHand 8 per WINDOWS) potrebbero contenere informazioni ausiliarie che solo il programma che ha generato tale file è in grado di utilizzare. Con il comando GNU/LINUX `eps2eps` è possibile ottenere una versione “distillata” del file EPS, funzionalmente equivalente alla precedente ed ottimizzata.
- Con alcune versioni di sistemi operativi *Microsoft*TM, potrebbe essere non semplice creare immagini EPS adatte ad essere inserite in documenti L^AT_EX. Infatti, in alcuni casi, il driver non è in grado di generare codice EPS ma soltanto codice POSTSCRIPT. Un file POSTSCRIPT è un file che contiene una collezione di comandi POSTSCRIPT necessari per stampare una o più pagine mediante un dispositivo in grado di comprendere il linguaggio POSTSCRIPT. Il linguaggio POSTSCRIPT include, dunque, sia istruzioni che permettono di disegnare un oggetto grafico sia istruzioni che permettono di stabilire le proprietà “fisiche” del documento complessivo (come, ad esempio, suddivisione in pagine, formato di stampa, ...). D’altro canto, un file `.eps` può e deve contenere solo i comandi necessari per creare un certo oggetto grafico e per indicare le sue dimensioni: un file `.eps` non può contenere, ad esempio, comandi per la gestione delle pagine. I file `.eps` sono adatti ad essere inclusi in altri documenti e non ad essere stampati direttamente. Un modo per risolvere questi problemi qualora il driver di stampa non sia in grado di generare file `.eps` validi, potrebbe essere quello di ricorrere ad altri programmi per poter convertire il file in un file che contenga solo comandi ammessi in un file `.eps`. Ad esempio, è possibile utilizzare il programma `GSview` (sito web: <http://www.cs.wisc.edu/~ghost/>) per convertire il file creato in un file EPS valido. Alternativamente, è possibile utilizzare il comando `ps2epsi` per ottenere un risultato analogo. Questo modo di procedere dà un buon risultato anche quando il driver di stampa sia in grado di generare codice EPS ma le dimensioni del grafico (indicate nel file `.eps` dal commento “`%%BoundingBox`”)

non sono quelle giuste. Infatti, anche se il grafico è più piccolo di una pagina, alcuni driver di stampa indicano come dimensione del grafico la dimensione dell'intera pagina. Sia `GSview` che `ps2epsi` sono in grado di calcolare la dimensione corretta del grafico e di inserire tale informazione nel file `.eps` da creare.

- Quando si creano figure oppure tabelle, utilizzare liste di direttive di posizionamento con almeno due preferenze. Più preferenze si indicano, maggiori saranno le impaginazioni possibili tra le quali \LaTeX dovrà scegliere quella tipograficamente più gradevole.

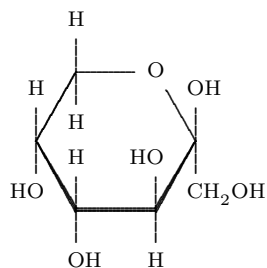
A.5 Per saperne di più

In questa sezione viene fornita una lista a manuali ed informazioni per la gestione di figure e tabelle, indicando i link alla documentazione dei pacchetti precedentemente citati ed anche ad altri pacchetti che potrebbero rivelarsi utili nella gestione di figure e/o tabelle. Forniremo inoltre alcuni riferimenti alla documentazione di programmi esterni che possono essere utilizzati per creare figure.

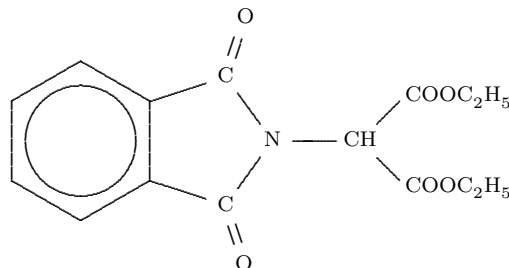
- <http://tex.loria.fr/ctan-doc/macros/latex/packages/graphics/grfguide.ps>
Manuale che raccoglie le informazioni di base sui pacchetti per la gestione di oggetti grafici mediante \LaTeX .
- <http://tex.loria.fr/ctan-doc/macros/latex/packages/graphics/color.dvi>
Per la gestione di colori.
- <http://tex.loria.fr/ctan-doc/macros/latex/packages/graphics/graphicx.dvi>
Manuale che raccoglie tutte le informazioni sull'uso del pacchetto `graphicx`.
- <http://www.pd.infn.it/TeX/doc/latex/caption/caption.dvi>
Permette di personalizzare le didascalie di figure e tabelle.
- <http://odur.let.rug.nl/alfa/tex/latex/styles/placeins.txt>
Permette di forzare il posizionamento di tutte le figure/tabelle create in una sezione o in un capitolo prima dell'inizio della sezione o capitolo successivi.
- <http://odur.let.rug.nl/alfa/tex/latex/tools/longtable.ps>
Permette di creare tabelle "lunghe", che possono venir spezzate e fatte proseguire nella pagina successiva.

- <http://odur.let.rug.nl/alfa/tex/latex/tools/tabularx.ps>
Manuale del pacchetto `tabularx` che mette a disposizione l'ambiente `tabularx`, simile all'ambiente `tabular`, ma molto più potente. Questo pacchetto consente di creare tabelle aventi una ben precisa larghezza indicando quali colonne possono essere allargate/ristrette per poter ottenere l'ampiezza complessiva desiderata.
- <http://odur.let.rug.nl/alfa/tex/latex/styles/float.ps>
Permette di creare nuovi oggetti mobili il cui posizionamento segue le stesse regole per il posizionamento di figure e tabelle.
- <http://odur.let.rug.nl/alfa/tex/latex/floatflt/floatflt.ps>
Permette di posizionare una figura oppure una tabella in una porzione dell'area di stampa e consente al testo di proseguire accanto alla figura/tabella (testo che "gira" attorno alla figura/tabella).
- <http://odur.let.rug.nl/alfa/tex/latex/hyperref/manual.pdf>
Qualora venga utilizzato pdf \LaTeX per creare direttamente un file `.pdf`, questo pacchetto rende i riferimenti a capitoli, sezioni, figure, tabelle, equazioni e riferimenti bibliografici come "link ipertestuali" all'interno del documento stesso. Inoltre, gli indirizzi Internet inseriti mediante il pacchetto `url` vengono interpretati come link a pagine web (ovvero: cliccando sopra, un browser web sarà lanciato e fatto puntare all'indirizzo indicato).
- <http://odur.let.rug.nl/alfa/tex/latex/styles/subfigure.ps>
Pacchetto per la creazione e gestione di sottofigure e sottotabelle.
- <http://odur.let.rug.nl/alfa/tex/latex/supertab/supertabular.ps>
Permette di creare tabelle "lunghe", che possono venir spezzate e fatte continuare nella pagina successiva. Frammenti di testo del tipo "continua nella pagina seguente" e "segue dalla pagina precedente" vengono inseriti automaticamente laddove necessario.
- <http://odur.let.rug.nl/alfa/tex/latex/psfrag/pfgguide.ps>
Manuale d'uso del pacchetto `psfrag`.
- <http://www.pstricks.de/>, <http://www.tug.org/applications/PSTricks/>
Serie di pacchetti con i quali è possibile creare diagrammi di vario tipo, grafici, circuiti elettrici, diagrammi di strutture di molecole, Potrebbe essere considerato come un'alternativa all'uso di `xfig` per la creazione di alcuni tipi di diagrammi.

- http://www.math.upenn.edu/tex_docs/context/ppchtex/mp-ch-en.pdf
 Documentazione del pacchetto PCH_{TEX} per la creazione di diagrammi di strutture di molecole (vedi Figura A.1). Permette di creare in maniera semplice e precisa diagrammi anche molto complessi (si veda anche: http://pdf.berflonet.nl/latex_packages_ppchtex_up-002-s.pdf).



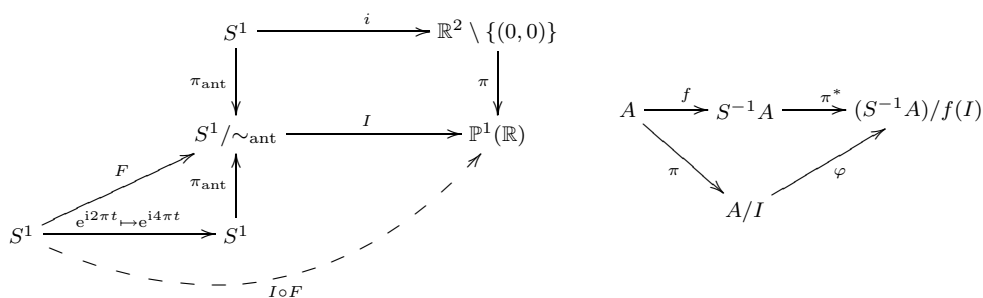
(a) β -D-fructopiranosio



(b) Immido-malonato

Figura A.1: Esempi di diagrammi di strutture di molecole

- http://www.cs.luc.edu/~rig/home/tex/treetex/tree_doc.ps
 Documentazione di un pacchetto per la creazione di alberi binari.
- <http://odur.let.rug.nl/alfa/tex/latex/ms/eso-pic.ps>
 Pacchetto per inserire un'immagine come sfondo ad ogni pagina (ad esempio: logo dell'università nello sfondo di ogni lucido).
- <http://www.ctan.org/tex-archive/macros/generic/diagrams/xypic/>
 Pacchetto molto evoluto per la creazione di diagrammi commutativi. Esempi:



- <http://tex.loria.fr/packages/diagrams-manual.ps.gz>
 Un altro pacchetto per la creazione di diagrammi commutativi.

- <http://odur.let.rug.nl/alfa/tex/latex/booktabs/booktabs.ps>
Permette di creare tabelle esteticamente più piacevoli di quelle normalmente ottenibile con i comandi standard di \LaTeX (si confronti la Tabella 2.2(a) con la Tabella 2.2(b)).
- <http://www.ctan.org/tex-archive/macros/latex/contrib/ctable/ctable.pdf>
Pacchetto simile al precedente, per la creazione di tabelle. Permette inoltre di creare in maniera semplice note a “pie’ di tabella”.
- <http://www.ctan.org/tex-archive/macros/musictex/>
Pacchetto per scrivere musica con \LaTeX .
- <http://www.ctan.org/tex-archive/fonts/skak/>
Pacchetto per realizzare scacchiere per il gioco degli scacchi.
- <http://www.cs.uni.edu/Help/gnuplot/>
Semplice guida introduttiva per imparare a creare grafici di funzioni 2D e 3D mediante gnuplot.
- <http://documents.wolfram.com/v5/>
Documentazione completa del programma di calcolo scientifico *Mathematica*.
- <http://mathcs.holycross.edu/~ahwang/current/ePiX.html>
Programma gratuito, disponibile per l’ambiente GNU/LINUX, che consente di creare il grafico di una funzione mediante uno script la cui sintassi assomiglia a quella di un programma in linguaggio C. Permette anche di inserire espressioni \LaTeX nei grafici creati. La Figura A.2 mostra un esempio di grafico ottenuto con questo programma. ePiX permette di ottenere grafici molto precisi in quanto tutte le componenti di un grafico possono venir calcolate o costruite analiticamente, a partire dalle loro proprietà geometriche. Inoltre, sfruttando la potenza del linguaggio di programmazione C, è possibile parametrizzare alcuni degli oggetti geometrici da disegnare e fare in modo che le caratteristiche degli altri vengano calcolate automaticamente.
- <http://cm.bell-labs.com/who/hobby/MetaPost.html>
Pagina web di METAPOST, potente linguaggio di descrizione di grafici. Il linguaggio METAPOST permette di creare file in formato POSTSCRIPT che possono essere inseriti in un documento \LaTeX mediante l’usuale comando `\includegraphics`. La Figura A.3 mostra alcuni esempi di grafici ottenuti con questo linguaggio.

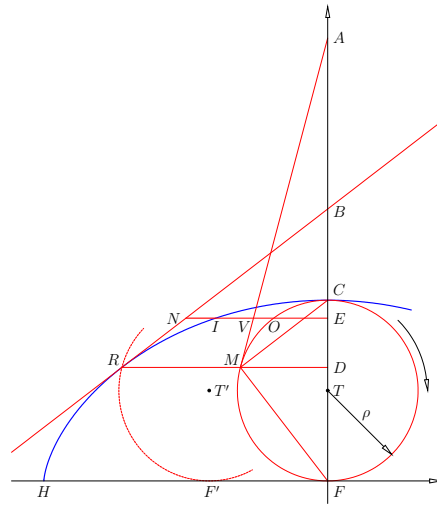


Figura A.2: Esempio di grafico creato con ePiX

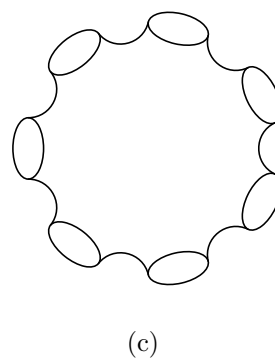
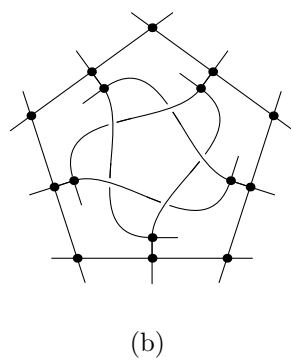
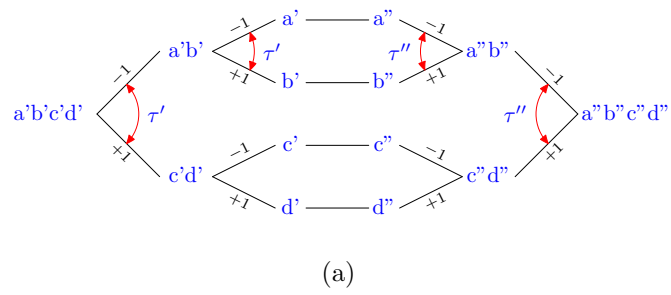


Figura A.3: Esempi di grafici creati con METAPOST

- <http://profs.sci.univr.it/~gregorio/egtex.html>
 Pagina realizzata dal prof. Enrico Gregorio (Università di Verona) contenente numerosi esempi di cosa si può fare con \LaTeX . Contiene anche una lista di “errori” (così definiti dall’autore stesso della pagina) ovvero: “cosa non si deve assolutamente fare durante la scrittura di un documento \LaTeX ”.

A.6 Manuali su \LaTeX (e \TeX)

In questa sezione viene fornita una lista di riferimenti a manuali e informazioni di vario tipo. Benché non si tratti di argomenti correlati strettamente alla gestione di figure e tabelle, ne diamo comunque una breve descrizione ritenendo che tali informazioni possano risultare preziose in molte altre circostanze. Si avverte il lettore che alcuni dei pacchetti citati non fanno generalmente parte di un’installazione “standard” di \LaTeX ; in tal caso sarà necessario provvedere ad un’installazione manuale dei pacchetti di cui si necessita.

- <http://www.tex.ac.uk/tex-archive/info/companion-rev/ch8.ps>
 Contiene un aggiornamento del Capitolo 8 di [GMS94] con numerosi informazioni ed esempi su come usare i pacchetti della American Mathematical Society (AMS) per gestire formule matematiche.
- http://www.math.upenn.edu/tex_docs/latex/amsmath/amslldoc.dvi
 Guida all’uso dei pacchetti della AMS.
- <http://www.tex.ac.uk/ctan/fonts/amsmath/doc/amslldoc.ps>
 Contiene una descrizione dettagliata di come utilizzare i font prodotti dalla AMS.
- <http://www.tex.ac.uk/tex-archive/info/epslatex.ps>
 Contiene molte informazioni ed esempio su come inserire figure in documento \LaTeX . Parte di questo manuale è stato preparato attingendo da questo manuale.
- <http://www.tex.ac.uk/tex-archive/info/gentle/gentle.ps.gz>
 Buon manuale su \TeX .
- <http://www.ctan.org/tex-archive/systems/knuth/tex/>
 Contiene l’intero codice sorgente del libro “*The \TeX book*” scritto dal creatore di \TeX . Tale codice è stato volutamente reso incompilabile dall’autore stesso mediante l’inserimento di alcuni errori.

- <http://www.tex.ac.uk/tex-archive/info/lshort/english/lshort.ps>
Ottimo manuale su L^AT_EX.
- http://ftp.pluto.it/pub/pluto/ildp/misc/impara_latex/impara_latex.ps.gz
Manuale in italiano che permette di imparare a scrivere documenti con L^AT_EX partendo dai concetti base.
- <http://www.ctan.org/tex-archive/info/JMPL.ps.gz>
Manuale in lingua francese molto interessante in quanto ricco di esempi (e corrispondente codice L^AT_EX). Riporta alcune cose molto evolute che possono essere fatte con L^AT_EX; ulteriori informazioni andrebbero poi reperite consultando la documentazione dei vari pacchetti utilizzati.
- <http://www.tex.ac.uk/tex-archive/info/symbols/comprehensive/>
Lista di tutti i simboli (e relativi comandi) che possono essere ottenuti con L^AT_EX.¹
- <http://www.ctan.org/tex-archive/language/>
Informazioni su come scrivere documenti L^AT_EX in diversi linguaggi (arabo, aramaico, cirillico, cinese, giapponese, coreano, greco, ...).
- <http://www.ctan.org/tex-archive/fonts/hieroglyph/>
Per la creazione di geroglifici egiziani con L^AT_EX (si veda anche <http://guit.sssup.it/downloads/hierotex1.pdf>).
- <http://www.ctan.org/tex-archive/fonts/tipa/>
Per poter riprodurre simboli fonetici.
- <http://odur.let.rug.nl/alfa/tex/bibtex/base/btxdoc.ps>
Guida all'uso di BIB_TE_X.
- <http://odur.let.rug.nl/alfa/tex/latex/seminar/sem-user.ps>
Per realizzare lucidi.
- <http://tex.loria.fr/english/packages.html>
Contiene link ai manuali di molti dei pacchetti disponibili per L^AT_EX.
- <http://www.tex.ac.uk/cgi-bin/texfaq2html>
La principale lista di FAQ ("Frequently Asked Questions") su L^AT_EX. È possibile utilizzare un motore di ricerca locale per effettuare ricerche nell'archivio di tutte le FAQ.

¹Al momento della preparazione di questa guida, il documento citato ne elenca ben 2826.

- <http://www.ce.cmu.edu/~kijoo/latex2pdf.pdf>
A volte, la conversione di un file POSTSCRIPT in PDF mediante il comando GNU/LINUX `pdf2ps`, potrebbe, dar luogo ad un file `.pdf` di scarsa qualità grafica e/o con font “bitmap” ed in bassa risoluzione. Il link citato sopra, illustra un possibile metodo per poter arginare questo tipo di problema.
- http://www.accademiadellacrusca.it/faq/faq_risp.php?id=4016&ctg_id=93
A volte, anche la propria lingua madre può dar problemi, in modo particolare se è una lingua difficile come l’italiano. Probabilmente, gli errori più frequenti riguardano l’uso degli accenti. La pagina web citata—una tra le più autorevoli—permette di chiarire ogni dubbio. Gli accenti utilizzati nella lingua italiana sono l’accento grave (̀) e l’accento acuto (´). Se si utilizza una tastiera italiana per scrivere i propri documenti, è possibile includere il pacchetto `inputenc` con l’opzione `latin1`. In questo modo, è possibile utilizzare direttamente le lettere accentate. Non tutti i sistemi operativi sono però in grado di gestire e visualizzare correttamente i caratteri accentati e ciò impedisce di giovare del pacchetto `inputenc`. Ciò resta vero anche quando si devono condividere sorgenti \LaTeX con altri utenti i quali potrebbero avere il problema della visualizzazione delle lettere accentate. Se, per uno dei motivi precedenti non è possibile utilizzare le lettere accentate presenti sulle tastiere italiane, potrebbe essere non facile—soprattutto per quegli utenti che per primi si avvicinano al mondo di \LaTeX —ricordare quale sia l’accento da utilizzare su una certa lettera. Un modo semplice per ricordarsi qual’è l’accento giusto è il seguente:

sulle vocali “a”, “i”, “o” ed “u” l’accento è sempre a “casetta” e ciò resta quasi sempre vero anche per la vocale “e”.

Consideriamo, ad esempio, la parola “Forlì” (Figura A.4). Potremmo immaginarci che l’accento sull’ultima lettera della parola sia una porzione del tetto di una casa vista frontalmente. Questo pezzo di tetto, idealmente, si appoggia sulle estremità superiori destre delle ultime due lettere della parola “Forlì” e, quindi, trovandosi nella parte destra della “casa” deve essere orientato in un certo modo piuttosto che nel verso opposto. Questa semplice analogia permette dunque di ricordare che l’accento è solitamente quello grave. I casi in cui l’accento acuto va usato sono molto pochi e, con un po’ di pratica, diviene ben presto facile ricordarli.

Forlì

Figura A.4: Spiegazione della regola: *sulle vocali “a”, “i”, “o” ed “u”
l’accento è sempre a “casetta”*

Bibliografia

- [Car99] D. P. CARLISLE, *Package in the ‘graphics’ bundle*, gennaio 1999, URL: <http://www.ctan.org/tex-archive/macros/latex/required/graphics/grfguide.ps>
- [Cau04] LUCA CAUCCI, *Inserimento di Figure con \LaTeX : Utilizzo del Pacchetto “graphicx” e di Altri Pacchetti per Ottenere Determinati Risultati ed Impaginazioni*, luglio 2004, URL: http://www.guit.sssup.it/downloads/fig_tips.ps
- [Coc95] STEVEN DOUGLAS COCHRAN, *The subfigure package*, marzo 1995, URL: <http://odur.let.rug.nl/alfa/tex/latex/styles/subfigure.ps>
- [Dah98] MATS DAHLGREN, *Welcome to the floatflt package!*, giugno 1998, URL: <http://www.math.psu.edu/doc/teTeX/latex/floatflt/floatflt.dvi>
- [GMS94] MICHEL GOOSSENS, FRANK MITTELBAACH e ALEXANDER SAMARIN, *The \LaTeX Companion*, Addison-Wesley Publishing Company, Reading, MA, USA, 1994
- [PLA03] MARCO PRATESI, MARCO LATINI e MICHELE ANTONECCHIA, *\LaTeX -PDF-Howto*, febbraio 2003, URL: <http://www.telug.it/marco/LaTeX-PDF-HOWTO/LaTeX-PDF-HOWTO.html>
- [Rec97] KEITH RECKDAHL, *Using Imported Graphics in $\LaTeX 2_{\epsilon}$* , dicembre 1997, URL: <http://www.ctan.org/tex-archive/info/epslatex.ps>
- [Som04] ALEX SOMMERFELDT, *Typesetting captions with the caption package*, luglio 2004, URL: <http://www.ctan.org/tex-archive/macros/latex/contrib/caption/caption.pdf>

-
- [Spa04] MARIANO SPADACCINI, *LaTeX-Figures-HowTo*, giugno 2004, URL:
<http://utenti.lycos.it/spadacciniweb/LaTeX-Figures-HowTo.pdf>