Un'immagine, un dipinto, una curva cartesiana,
una spirale, un'iperbole,... ogni forma di bellezza
insomma, non è altro che lo sviluppo armonico
di una delle proprietà del cerchio.
«Togliete i numeri alla bellezza, questa svanirà ».

Scriba & Isidoro di Siviglia

Parte IV

Applicazioni grafiche

Edizione Test - Agosto 2008

Capitolo 11

Preparazione, inserimento e trattamento delle immagini

Premessa d'ordine generale sulla grafica in LaTeX 2ε

Quando TeX fu scritto non esisteva il formato PostScript; né esistevano i formati `.gif`, `.jpg`, o `.eps`: fu la povertà d'interfacce grafiche che indusse DONALD KNUTH ad ideare un output secondo il formato poi chamato DVI.[1] Solo quando poi si trasformò il file DVI in un file PostScript nacquero i packages epsf e psfig scritti per il LaTeX2.09.

Il problema dell'inserimento della grafica fu affrontato dal team di LaTeX3 nel 1994, con la *release* LaTeX 2ε: in quell'occasione furono interamente riscritte le istruzioni, ed il package graphicx sostituì graphics, mostrando rispetto a questo maggiore ricchezza di istruzioni e versatilità.

In questa parte vedremo come opera in LaTeX l'ambiente grafico, inteso con il termine (in via di prima approssimazione) tutto ciò che si risolve in un elemento grafico, sia esso relativo all'inserimento di figure ovvero alla creazione di disegni e grafici; va tenuto presente che a seconda dell'azione comandata, il sistema segue procedure diverse.

Questo primo passo pone una distinzione fondamentale fra grafica interna ed esterna al sistema.

La grafica *interna* è quella che viene creata richiamando nel preambolo particolari applicativi come pstricks ovvero sfruttando picture, l'ambiente standard per la creazione di disegni. In questo caso LaTeX agisce soprattutto sulle coordinate fornite, nel senso che ogni singola figura geometrica (linea, cerchio, poligono,...) viene individuata e posizionata secondo le coordinate del piano cartesiano; packages sofisticati consentono con le stesse operazioni, ma a più ampio spettro, di rinnovare l'ambiente in cui si opera, creando grafici e disegni.

La grafica *esterna* è invece quella relativa all'inserimento di files grafici creati con dispositivi esterni: su queste immagini possono essere effettuati soltanto interventi di ridimensionamento, rotazione, scalatura,... ma non sulle immagini già composte. Inserendo files grafici[2] LaTeX riceve

1. Una delle prime pubblicazioni su questo dispositivo apparve sul TUGboat, Volume 10, 1989, n. 1, il file è disponibile all'indirizzo: `http://www.tug.org/TUGboat/Articles/tb10-1/tb23broeren.pdf`. Un file DVI contiene tutte le informazioni necessarie per una stampa od una viaualizzazione a video del file, ad eccetto dei bitmap, dei tipi di fonts creati con applicativi dedicati e di altro materiale introdotto con speciali comandi. Il *canonico* riferimento per comprendere la struttura di un file DVI è l'esame del sorgente predisposto da DONALD KNUTH, visualizzabile all'indirizzo `ftp://cam.ctan.org/tex-archive/systems/knuth/dist/texware/dvitype.web`.

2. Il trattamento di files grafici è una delle operazioni più delicate in LaTeX, meritevole di particolare approfondimento. Si consiglia pertanto la lettura di testi specifici fondamentali. Innanzi tutto *The LaTeX Graphics Companion* [11, IV], e per quanto riguarda i manuali disponibili in rete, *Using Imported Graphics in LaTeX 2ε* di KEITH RECKDAHL [16, IV], *Graphics in LaTeX 2ε* di URS OSWALD [15, IV], ed in lingua italiana, *Gestione di figure*

le immagini previa la compatibilità con i formati ammessi, operando sulle stesse le trasformazioni impartite con eventuali istruzioni: larghezza, altezza, rotazione, riduzione in scala,...

Un aspetto fondamentale nel trattamento delle immagini, è che queste, al pari delle tabelle, sono trattate come *oggetti flottanti*: [3] il fatto che le figure siano considerate oggetti flottanti vuol dire soprattutto che esse non compaiono (quasi mai) nel punto indicato, perché le immagini devono assolvere alla fondamentale esigenza di non essere spezzate. Le figure sono cioè contenute in appositi ambienti che facendole flottare (scivolare) lungo il testo, le collocano nel punto in cui il sistema ritiene più opporturo in conformità ai propri parametrici estetici e funzionali. La risoluzione di alcuni problemi legati (e derivanti) da questi oggetti si vedrà a pagina 304. Esamineremo anche, a pagina 311, ambienti flottanti, che presiedono a particolari combinazioni di *mescolamento*, più che di inserimento, di immagini e testo. [4]

Quasi un *terzo tipo* di grafica dato dalla commistione dei due elementi è reso posibile da psfrag. che opera da sistema interventi su disegni ed immagini costruiti con altri applicativi.

Un breve capitolo (il 12.mo) sarà dedicato ai box, oggetti di forte attinenza con la grafica, pur conservando legami con le caratteristiche proprie dei caratteri: il legame con la grafica deriva ai box dal fatto che si presentano individuati da *altezza*, *larghezza*, *profondità*, e che possono ruotare come le immagini ed i grafici; il legame con i caratteri deriva dal fatto che i box si prestano a *maneggevolezza* come se si trattasse proprio di un carattere.

In tutti questi casi vige *assoluta* la filosofia WYSIWYM: prima di mettersi al lavoro occorre avere deciso il tipo di file che si desidera ottenere: se PDF o PS. La scelta del tipo di file in output è determinante anche sotto altri aspetti. Se ad esempio s'intende lavorare con strumenti di alta sofisticazione come i pacchetti della serie ps-tricks, che consentono rappresentazioni grafico-matematiche di altissimo livello, si deve sapere che questi pacchetti rispondono soltanto ai comandi di un file PS e non anche PDF, a meno di non usare ulteriori applicativi, e che, ancora, le immagini devono essere in formati diversi a seconda dell'output finale: ancora una volta PS o PDF.

Formati grafici

L'ideale sarebbe di poter disporre di immagini nel formato destinatario senza operare alcuna conversione che inevitabilmente disturba la qualità, ma quasi mai questo è possibile.

Prima di vedere come trasfomare le immagini in un formato diverso dall'originale, idonee ad essere ricevute da LaTeX, è necessario premettere alcune informazioni sui dati delle immagini, ricordando brevemente che i formati grafici conoscono una sostanziale suddivisione: formati cosiddetti vettoriali e formati a matrici di punti.

La seguente discussione, per la sua chiara brevità, avrà un valore semplicemente descrittivo.

Formati vettoriali

In questo formato agisce una *routine* interna che disegna aree di colore sostanzialmente analoghe. In pratica la *routine* individua una serie di punti sui quali fa scorrere le linee che delimitano i contorni, ed il disegno si costruisce così *secondo contorni di aree d'identico colore* e l'operazione di scalatura dell'immagine si presenta abbastanza agevole.

---

e Tabelle con LaTeX, di LUCA CAUCCI e MARIO SPADACCINI [6, IV].

3. Questo vuol dire soltanto eguaglianza nella filosofia di trattamento, per il resto i due oggetti usano contatori diversi e code di allocamento diverse, altrimenti non potrebbero esiste due indici diversi: delle tabelle e delle figure.

4. La rilevanza degli ambienti flottanti e di un sistema d'inclusione dei files articolato, come vedremo in seguito, risiede in questo.Lavorando in ambienti (e sistemi) non specifici quali word, ad esempio, l'immagine viene a far parte integrante della pagina, in una parola — per quanto possa sembrare strano — diviene testo; e quindi un suo eventuale ridimensionamento per adattarla alla pagina sfruttando le applicazioni del processor che si sta usando, comporta una perdita originaria dei dati dell'immagine, e quindi della qualità.

In LaTeX al contrario, l'immagine resta quella originaria, e sono le varie opzioni di `\include` a prendorsi cura del suo posizionamento rispetto alla pagina, secondo quanto desiderato, senza alcun deterioramento dei dati.

In ambito vettoriale i formati più diffusi sono il PostScript, formato `.ps`, e l'Encapsulated PostScript formato `.eps`. Nel corso del presente capitolo *sarà sempre a questi ultimi formati che ci riferiremo* per il trattamento delle immagini per un output in PS.

In questo formato ridurre le dimensioni in pixel di un'immagine si presenta per il sistema come un'operazione agevole, e per l'utente di una praticità ed efficienza pressoché uniche. Infatti scalare dimensionalmente un'immagine non vuol dire altro in questi formati che ridurre lo spessore delle linee di cui è composta, senza perdita di qualità e, tutto sommato, neanche di dati.

Fra questi formati si va diffondendo sempre più quello caratterizzato dal suffisso `.svg`, dalle iniziali delle parole *Scalable Vector Graphics*.

Formati a matrici di pixel

Gli altri formati hanno varie estensioni a seconda dell'applicativo con cui sono stati concepiti: `.bmp`, `.tiff`, `.gif`, `.png`, `.wmf`,... e tendono a moltiplicarsi a ragione della grande diffusione delle macchine fotografiche digitali che utlizzano spesso formati proprietari, e mal si prestano alla scalatura delle immagini. Inoltre c'è da considerare che alcuni di questi sono *ab origine* compressi, e quando si va a lavorare sulle immagini molte informazioni si perdono.

11.1 Inserimento di files grafici

Il presente capitolo sarà dedicato in maniera quasi esclusiva all'inserimento di files grafici. Per files grafici s'intenono sia le immagini *tout-court*, sia i disegni realizati con altri applicativi e quindi importati nel documento. Tali files verranno spesso individuati con il nome di "immagini". I grafici creati con *routines* compatibili con il sistema saranno trattati al capitolo 13.mo.

In questa prima parte della sezione ci si occuperà della preparazione delle immagini per renderle compatibili con il sistema (se di formato diverso da quello ammesso dall'impostazione del documento). e di alcune caratteristiche proprie delle immagini in formato `.eps`. Successivamente si tratterà l'inclusione dei files grafici con `\includegraphics`.

Preparazione delle immagini

Per un'ottimale organizzazione del lavoro è bene che i files da includere siano allocati in apposite cartelle, una cartella per ogni capitolo o parte del documento. A richiamare poi i singoli files nelle rispettive allocazioni si provvede fornendo le istruzioni che vedremo a pagina 289.

Se l'ideale sarebbe dunque poter lavorare sempre su immagini in formati vettoriali, questo è di rado possibile, in quanto le immagini dei nostri archivi provengono dalle più disparate fonti, e sono dei più differenti formati.

Poiché un file LaTeX ammette output PDF e PS nell'inserire le immagini la prima cosa da considerare è stabilire in quale formato si vuole ottenere un file, se PS o PDF appunto, dal momento che questi formati accettano — rispettivamente — files con suffissi `.eps` e `.ps` o con suffissi `.png` e `.pdf`.

Questo discorso è valido se, partendo dal sorgente, s'intende ottenere *direttamente* un file PDF senza cioè ulteriori passaggi che non tengano conto dei comandi della serie `ps2pdf1.x` già visti sotto la sezione 3.7 a pagina 86.

La prima operazione da fare è allora quella di rendere accessibili a LaTeX immagini di formati diversi da quelli sopra descritti usando un apposito software di conversione. Poiché — ripeto – si desidera un file `.ps`, tutte le nostre immagini avranno il suffisso `.eps` o `.ps`.

Conversione di immagini

Per la conversione *brutale* delle immagini un software molto utile è convert, la cui sintassi è: `convert`

`convert file.* file.eps`. La conversione delle immagini è *nuda e cruda*, senza alcun ritocco, da un formato all'altro.

epstopdf

Alla parte II a pagina 88, avevo ricordato epstopdf accennando alla conversione delle immagini da inserire in un file LaTeX. È questo un'applicativo certamente utile e ben funzionante, che — tra l'altro — ha il suo corrispettivo in pdftops che converte dal formato PDF al formato PS. Considerato il tipo di istruzioni da impartire (`epstopdf immagine.eps`), entrambi sono naturalmente applicativi che funzionano da linea di comando, come pure è fig2dev che trasforma i files in `.eps` tramite l'istruzione: `fig2dev -L miofile.fig miofile.eps`

wmf2eps

Sempre da consolle opera il pacchetto wmf2eps di WOLFANG SCHULTER che trasporta i files dei windows metafile (`.wmf`) nel formato `.eps`.

xfig

Al contrario xfig [5] lavora in ambiente grafico, ed anche se non salva i files in `.eps`, tuttavia con l'opzione `save as` i files si rendono accessibili ad una successiva importazione.

paint

Fra i programmi di trattamento di grafico vanno ancora ricordati paint e soprattutto il già citato gimp, un versatilissimo programma di trattamento d'immagini da qualche anno disponibile anche per la piattaforma windows.

gimp

display

Ricordo da ultimo l'utilissimo e versatilissimo ImageMagick, (che si lancia digitando `display`) disponibile in tutte le piattaforme UNIX e LINUX, e che consente una trasposizione delle immagini in un gran numero di formati, con in aggiunta un buon trattamento delle stesse: in caso questo devosse essere necessariamnete sofisticato, è buona norma allora ricorrere a gimp.

I BoundingBox

I files in formato `.eps` ed `.ps`, sono caratterizzati da un particolare elemento, i `BoundingBox`, generalmente individuati dalla sigla `BB`. Questi rappresentano gli elementi che permettono di calcolare lo spazio riservato alla figura, il perimetro della stessa che idealmente dovrebbe coincidere con l'immagine.

I `BoundingBox` sono rappresentati da quei dati che compaiono alle prime righe dell'immagine di un file `.eps` o `.ps`, e poiche questi files non sono altro che files ASCII, sono leggibili (operazione riservata ai *semidei*) se si aprono con un qualsiasi editor testuale o con un applicativo dedicato come gview. Le prime righe, quelle qui che interessano (accessibili ai mortali), appaiono in questa forma prima delle informazioni binarie dell'immagine:

```
%!PS-Adobe-3.0 EPSF-3.0
%%Creator: (ImageMagick)
%%Title: (/home/heinrich/latex/appunti_2008/eps/barca.eps)
%%CreationDate: (Tue Mar 27 00:38:41 2007)
%%BoundingBox: 0 0 440 444                  <-----------
%%HiResBoundingBox: 0 0 100.685 77
%%DocumentData: Clean7Bit
%%LanguageLevel: 1
%%Pages: 1
%%EndComments
```

Come si nota nel file vengono riportate anche tutte le informazioni che il sistema riesce a catturare sulla genesi dell'immagine. Noi ci soffermiamo su quelle riportate alla linea indicata con una freccia a sinistra. I quattro numeri vanno letti a coppia, e specificano ascisse e ordinate, a partire dall'angolo in basso a sinistra a quello in alto a destra.

I singoli elementi che sono così individuati: `%%BoundingBox:` *llx lly urx ury*, esprimono rispettivamente per *llx lly* le coordinate per l'angolo basso a sinistra e per l'angolo superiore

5. xfig è un applicativo disponibile anch'esso in ambienti UNIX e LINUX permette di creare grafici di notevole complessità ed importarli con facilità in LaTeX.

destro, e per *urx ury* le distanze dall'orlo sinistro del foglio sino al primo tratto d'inchiostro a destra e la distanza dall'orlo inferiore del foglio alla marcatura più alta. Mi rendo conto che il discorso sembra espresso in termini poveri, ma le cose stanno effettivamente in questi termini, ed anche nei testi più *aulici*: [10, II, pag. 354], sono riportate espressioni simili.

Tuttavia sono convinto che esprimendoci in termini geometrici - matematici, il discorso apparirà più chiaro. Possiamo allora dire che le singole cifre (prese per ciascun gruppo) rappresentano:

- le coordinate x dell'angolo basso a sinistra;
- le coordinate y dell'angolo basso a sinistra;
- le coordinate x dell'angolo superiore destro;
- le coordinate y dell'angolo superiore destro.

Secondariamente va detto cosa esprimono numericamente le cifre dei `BB`.

I numeri non rappresentano qualcuna delle consuete unità usate in LaTeX, bensì unità PostScript, quelle che DONALD KNUTH nel suo libro chiama *big points*: ogni punto è 1/72 di pollice, ed i punti PostScript sono più grandi, anche se leggermente, di un punto TeX, e per questo Knuth li ha chiamati *big points*.

Partiamo da numeri immaginari e supponiamo di avere dei `BB` di questa entità: `90 90 410 410`. La stessa grandezza dei numeri ci fornisce già un'indicazione su quello che rappresentano. In questo caso `90 90` rappresenteranno le coordinate dell'angolo in basso a sinistra, e `410 410` le coordinate dell'angolo in alto a destra.

Se i BoundingBox presentano dimensioni troppo generose rispetto al formato dell'immagine si correggono o agendo sul programma di grafica oppure aprendo il file `.eps` con un editor testuale e modificando opportunamente la serie di numeri con nuovi parametri.

Questo lavoro, ad esempio, si è reso necessario quasi sempre nei presenti *Appunti*, non tanto introducendo immagini con il suffisso `.eps`, quanto introducendo (specie parte III e V) come esempi files generati con la routine `dvips`, ed inclusi nel testo come se fossero immagini, con il solo suffisso `.ps`. Quasi sempre questi files, nonostante un'impostazione accurata del layout, occupavano una porzione spropositata della pagina al di là delle loro effettive imensioni, e la didascalia dell'immagine veniva talvolta a cadere parecchi centimetri dopo l'immagine reale, perché esisteva uno spazio bianco intorno all'immagine, considerato parte della stessa.

In occasione di alcuni files grafici che presentavano i `BB` troppo generosi, ho avuto persino un blocco di stampa nell'output diretto a stampante di un file PostScript.

Il motivo per cui spesso è necessario agire sui parametri dei BoundingBox è proprio questo, la necessità di *scontornare* l'immagine dagli spazi bianchi (e vuoti) che non hanno alcuna attinenza specifica con l'immagine e di alleggerirla nel peso.

Questo costringe ad un lungo lavoro se i BoundingBox risultano esagerati per numerose immagini, per cui conviene (inserendo immagini) impostare correttamente sin da principio i parametri di grafica, partendo magari da immagini di un identico formato dimensionale.

Se poi l'autore del testo è lo stesso fotografo, non dimentichi allora una cosa fondamentale: *scattare*, come si diceva una volta, tutte le immagini, per quanto è possibile, alla stessa distanza e con lo stesso obiettivo, altrimenti i ridimensionamenti saranno continui, e nel lavoro si avrà ora un'immagine scalata del 50%, una dell'80%, e così via dicendo, con resa grafica dimensionale assai diversa da una figura all'altra.

Un particolare effetto di questa operazione sui `BB` si vedrà fra breve a pagina 291.

Nel sorgente riportato alla pagina precedente, in sesta riga, è presente un'istruzione similare nella scrittura e nell'output: `%%HiResBoundingBox: 0 0 100.685 77`. Si nota che una delle coppie di numeri contiene un decimale. Anche questa come la precedente è un'opzione di `\includegraphics`, opzioni che vedremo a breve alla tabella 11.1.

Il riferimento a questa tipologia di `BoundingBox`, anziché all'altra, forza il sistema a sfruttare le impostazioni dei `BoundingBox` con precisione decimale.

Opzione	Valore
`draft`	Le figure sono lette, *verificate*, ma non inserite finché nella scrittura è presente l'opzione: `\usepackage[draft]{graphicx}`. Compaiono cornici con il nome del file grafico
`final`	È l'opposto di draft, ne annulla gli effetti
`hiderotate`	Non fa vedere gli oggetti ruotati
`hidescale`	Non fa vedere il testo ridimensionato
`hiresbb`	Cerca le indicazioni di dimensionamento degli oggetti nelle linee dei `%%HiResBoundingBox` anziché nelle linee dei `%%BoundingBox`.

Tabella 11.1: Opzioni dei packages graphics e graphicx

11.2 Inserimento di files

I packages che sovrintendono alle istruzioni di grafica sono: epsfig,[6] graphics, e graphicx. Altri, come color e xcolor pur essendo sempre attinenti alla grafica, devon essere considerati dei completamenti di questi packages. graphicx rappresenta un'estensione ed un'evoluzione di graphics che deve considerarsi ormai obsoleto, e qui si prenderà in considerazione solo questo.

graphicx; istruzione `\includegraphics` ed opzioni

graphicx dispone di una serie di comandi ed opzioni che miscelati ad altri sempre dell'ambiente grafico, consentono una notevole manipolazione delle immagini grazie alle combinazioni che si rendono possibili facendo interagire comandi ed opzioni. I comandi possibili, i cui effetti sono mostrati alla tabella 11.2 assieme ad alcune opzioni, sono:

- `\includegraphics[opzione]{....}`
- `\rotatebox[opzione]{angle}{....}`[7]
- `\scalebox{h-scale}[v-scale]{....}`
- `\resizebox{width}{height}{....}`

Rilevanza in questo package assumono le opzioni mostrate in tabella 11.1. Altre opzioni (`scale`, `rotate`, `angle`,...), sono trattate ai seguenti paragrafi e mostrate in tabella 11.3.

L'istruzione principale del package nell'inserimento dei files grafici è `\includegraphics`. Il comando può essere utilizzato come istruzione a se stante o all'interno dell'ambiente figure seguito, fra parentesi graffe, dal nome dell'immagine nella forma: `\includegraphics[opzioni]{file.eps}`.

Dicendo che l'istruzione può comparire anche in maniera a se stante, s'intende dire che l'istruzione può disporre che un'immagine appaia anche su di una linea di testo.

La scrittura `\includegraphics[height=4mm]{immagine.eps}` renderà  facendo apparire l'immagine sulla linea di scrittura. Altre istruzioni, tipo `\scalebox`, consentono effetti come quello appresso mostrato, relativo sempre alla scrittura di un testo trattato graficamente sulla linea:

```
\scalebox{-1}{\textsf{Testo rovesciato}}
\scalebox{1}[-1]{\textsf{Testo rovesciato}}
```

rendono:

Un altro esempio di questa scrittura (istruzione `\reflectbox`) è mostrata a pagina 297.

6. epsfig fu riscritto da SEBASTIAN RAHTZ assemblando fra loro due preesistenti *routines*: PSFIG di TREVOR DARREL, e EPSF di TOM ROKICKI, per assicurare la portabilità dell'interfaccia di PSFIG sulle *routines* di EPSF.
L'istruzione va a cercare le informazioni sull'immagine nei `BoundingBox`, calcola i valori di scala, ed inserisce l'immagine-oggetto in un box di quella dimensione nel documento.

7. `\rotatebox` è unistruzione propria dei box che sarà esaminata al capitolo 12, a pagina 322.

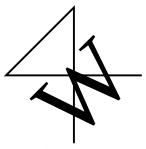
Istruzione	Effetto generato
`\includegraphics[width=0.5in,scale=3,angle=45]{w.eps}`	
`\rotatebox{65}{\includegraphics[totalheight=0.5in]{w.eps}}`	
`\scalebox{.60}{\includegraphics{w.eps}}`	
`\resizebox{0.5in}{!}{\includegraphics{w.eps}}`	

Tabella 11.2: Comandi di \includegraphics ed alcune opzioni

L'immagine può anche non possedere alcuna specifica estensione, essendo comunque in un dato formato. Alla dichiarazione `\includegraphics{immagine}` dovrà allora precedere nel preambolo l'espressione `\DeclareGraphicsExtensions{.eps, .ps}`. `\DeclareGraphicsExtensions`

A fini puramente di organizzazione del lavoro, quando si ha a che fare con molteplici files e molteplici immagini, è utile raggrupparle ed organizzarle per directory, riservando a ciascuna un determinato tipo di immagini. Nel preambolo va fornita la seguente istruzione: `\graphicspath{{eps/}}`, supponendo ovviamente di avere una cartella denominata eps. `\graphicspath`

L'inclusione dell'immagine comparirirà nel documento secondo questa sequenza:

```
\DeclareGraphicsExtensions{.eps, .ps}      \graphicspath{{eps/}}
\begin{document} \includegraphics[scale=.8]{eps/miaimmagine.eps} \end{document}
```

Opzioni di \includegraphics

Le opzioni principali di includegraphics sono quelle di cui alla tabella 11.3. I diversi tipi d'immagini ottenuti con mutamento di opzioni ed applicazione di questi parametri sono mostrati alla tabella 11.4 alla pagina seguente. [8] Il sorgente di quelle immagini, essendo un'applicazione di subfigure è mostrato nella sezione dedicata, a pagina 301).

Nell'inserimento delle opzioni, ed in caso di opzioni plurime, assume rilevanza l'ordine d'inserimento delle stesse, in quanto LaTeX legge la prima istruzione e la esegue, legge la seconda e la esegue, e così via dicendo. Trattiamo l'immagine apparsa prima sulla linea in diverse forme, con le opzioni inserite in sequenza diversa:

8. Per un approfondimento consulta *Using Imported Graphics in LaTeX 2ε* di KEITH RECKDAHL [16, IV], dove sono presenti anche altre opzioni per l'istruzione. A quel documento si rinvia.

Istruzione	Effetto generato
`height`	Altezza della figura in una delle unità di misura di LaTeX
`totalheight`	Altezza "totale" della figura in una delle unità di misura di LaTeX
`width`	Larghezza della figura in una delle unità di misura di LaTeX
`scale`	Fattore scalare della figura in una delle unità di misura di LaTeX
`angle`	Angolo di rotazione espresso in gradi ed in senso antiorario
`origin`	Rotazione della figura: `origin=c` ruota la figura attorno al centro
`bb`	Specifica i parametri `BoundingBox`

Tabella 11.3: Principali opzioni di \includegraphics

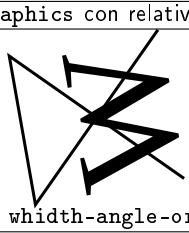
Opzioni di \includegraphics con relative istruzioni			
 `scale`	 `height`	 `whidth-angle-origin`	 `whidth`
 `totalheight`	 `whidth-angle`	 `bb=100 100 90 140`	

Tabella 11.4: Esempi sulle principali opzioni di includegraphics

I sorgenti per le tre immagini sono, a partire da quella di sinistra. i seguenti:

```
\includegraphics[height=10mm,angle=45,scale=.8]{logo.eps}
\includegraphics[angle=75,height=12mm,scale=.8]{logo.eps}
\includegraphics[scale=.8,height=9mm,angle=45]{logo.eps}
```

Qualche parola sulle unità di misura utilizzate come valori nelle opzioni.

È ormai risaputo che LaTeX accetta una quantità di unità di misura come millimetri, centrimetri, pollici, pica,... Ma nel definire un valore opzionale è sempre buona norma evitare la forma `\includegraphics[width=70mm]{immagine.eps}`, ed optare invece per una dichiarazione del tipo `\includegraphics[width=.7\linewidth]{immagine.eps}`.

Quale la differenza fra queste due dichiarazioni? Se un giorno ci si dedicesse a mutare le dimensioni della pagina, nel caso della prima dichiarazione la larghezza dell'immagine resterebbe

Figura 11.1: Variante asterisco di `\includegraphics`

ancorata ad un valore assoluto (70mm), mentre legando le dimensioni dell'immagine a `\linewidth` le sue dimensioni coinciderebbero sempre con l'indicata percentuale di larghezza della pagina.

Per questo motivo l'istruzione è da preferire sempre ad altre consimili del tipo `\textwidth` o `\columnwidth`. Ovviamente come unità di misura si possono utilizzare anche i punti, e se ne vedrà un'applicazione a pagina 295. Il fatto che spesso compaiono negli esempi misure in cm ed in pollici è dovuto al fatto che intendevo mostrare le reali dimensioni dell'immagine con riferimento ad un'unità di misura assoluta.

Oltre alle opzioni listate nella tabella 11.4 ne esistono altre cui qui mi limiterò ad accennare rinviando per una più completa descrizione dei comandi ai manuali già citati. Esse sono:

- `keepaspectratio`. Mantiene inalterato il rapporto dell'immagine (altezza e larghezza) quando un'eventuale specifica congiunta di `width` ed `height` potrebbe condurre a sproporzioni in altezza rispetto alla larghezza.
- `viewport`. Nella forma `\includegraphics[viewport=0 0 180 180, clip]{immagine.eps}`, l'istruzione apre una sorta di finestra attraverso cui guardare nl'immagine. le quattro coppie di numeri specificano le coordinate orizzontali (prima e terza coppia) e le coordinate verticali (seconda e quarta coppia). [9]
- `trim` Questa funzione è simile alla precedente, però i valori indicati tagliano questa volta *strisce* di immagine in questo ordine: da sinistra, dal basso, da destra, dall'alto. La scrittura è `\includegraphics[height=.3\linewidth,trim=60 40 30 20, clip]{immagine.eps}`.
- `clip`. Una volta che si siano effettuati tagli di strisce d'immagini con trim o viewport, l'opzione clip elimina quanto sporge fuori dai rettangoli.

Variante asterisco di `\includegraphics`

`\includegraphics` conosce la variante asterisco. La differenza di `\includegraphics*`, rispetto all'istruzione basilare, consiste nel fatto che l'istruzione asteriscata taglia l'immagine secondo i `BoundingBox` specificati.

In figura 11.1 è rappresentata la stessa immagine che in origine aveva i `BoundingBoxB` *tarati* a `0 0 163 123`, modificati in `20 10 198 123` per adattare la figura alla pagina. Queste modifiche sono comunque irrilevanti nel discorso che stiamo facendo. Rilevante è invece il diverso comportamento che il sistema segue nell'inserimento delle immagini a seconda che si usi o meno la versione asteriscata.

9. Il significato dell'ulteriore istruzione che compare dopo l'ultimo numero è spiegato all'ultima riga dell'elenco.

Nel primo caso `\includegraphics[20,10][198,123]{file.eps}` (a sinistra) l'immagine è inserita secondo la sua costruzione originaria senza alcun intervento, nel secondo caso (a destra) l'istruzione asteriscata `\includegraphics*[20,10][198,123]{file.eps}` produce l'effetto di ridimensionare l'immagine secondo le coordinate specificate nei `BB`, come è visibile non solo per le ridotte dimensioni dell'immagini, ma anche per la diversa superficie disegnata dalla cornice rispetto all'immagine ed un leggero spostamento verso l'alto.

In sostanza l'uso di questa istruzione produce un effetto analogo a quello dell'opzione `clip` vista alla pagina precedente.

11.3 Posizionamento dei files grafici: ambienti figure e minipage

Per posizionamento dei files s'intende l'*allocazione fisica di questi sulla singola pagina in cui dovranno comparire*, e tale posizionamento riguarda sia la centratura orizzantale quanto quella verticale, nonché alcune particolari configurazioni tipiche per lo più di packages specifici.

Per l'allineamento orizzontale, cioè le operazioni che collocano il file al centro, destra o sinistra della pagina, nonché agli estremi di questa,... LaTeX si serve dei singoli ambienti di centratura che fra breve andremo a vedere: alcuni derivano dal LaTeX standard e li abbiamo già esaminati alla parte II, altri vanno caricati con appositi packages. Particolari tipi di posizionamento, come le figure incorniciate nel testo, saranno trattate ai paragrafi successivi.

Diverso è il posizionamento della figura nella pagina, il punto in cui comparirà l'immagine. L'allineamento verticale si ottiene ricorrendo fornendo delle debite opzioni l'ambiente figure. Grazie a queste (comuni a quelle delle tabelle: `[!htbp]`) già visti a pagina 194, cui si rinvia per la descrizione, è possibile gestire l'allineamento verticale compatibilmente con l'esistenza di altre figure: *vedi* comunque in proposito quanto detto nella precedente nota [3] circa il numero limite di files grafici nella pagina.

Si consiglia di usare l'opzione [!h] (h = here) solo quando l'immagine è unica o lontana da altre. Qualora tale allocazione non riuscisse per via di altre immagini o di quantità di testo, il sistema procederà al collocamento dell'immagine all'inizio od alla fine della pagina. Se ancora il tentativo fallisse l'immagine sarà collocata nella prima area ritenuta disponibile alla pagina successiva. Se l'immagine è troppo grande rispetto alla pagina viene collocata a fine capitolo. Se non è indicata alcuna opzione è assunta come default `[tbp]`.

Ritengo pertanto sia buona cosa ricorrere sempre all'istruzione [!t] (t = *top of a page*) in modo da avere le figure posizionate all'inizio od alla fine della pagina in caso di immagini plurime.

Come si diceva nei consigli pratici forniti (*vedi* a pagina 94), si eviti di scrivere *alla figura seguente*, *alla figura precedente*, ma si faccia sempre riferimento alla `\label` cui la `\caption` della figura fa rinvio, perché non si sa mai in fase di compilazione dove quella immagine andrà a finire.

L'ambiente figure è spesso, come vedremo dagli esempi seguenti, racchiuso in una minipage, già tante volte citata nei sorgenti d'esempio e sfruttato in numerosi esempi. È giunto il momento di dedicargli qualche parola,

L'ambiente minipage

`minipage`

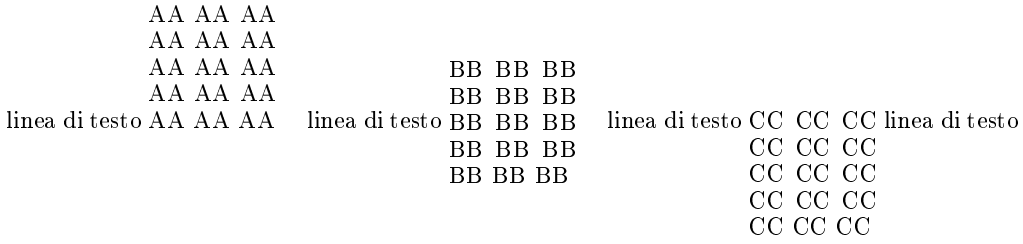
Una minipage è una pagina *in miniatura* che LaTeX si riserva all'interno dell'ambiente di lavoro, individuata dalle classiche istruzioni `\begin{minipage} \end{minipage}`.

Come ogni ambiente che si deve adattare alla pagina, la minipage conosce i parametri del dimensionamento e dell'ampiezza, per cui essa si presenta nella forma `begin{minipage}[cbt]{width}`, dove `\width` è sostituita al solito dall'unità di misura scelta. I parametri opzionali fra parentesi quadre `[cbt]` esprimono il posizionamento della minipage rispetto alla pagina di testo.

Alcuni esempi chiariranno il senso di questi parametri. Appresso sono rappresentate (vedi il relativo sorgente a seguire) tre minipage ciascuna per una larghezza di 17 mm che racchiudono

una coppia di lettere AA, BB, CC. Queste tre minipage sono, a loro volta, racchiuse in un'altra minipage di 134 mm d'ampiezza.

Prima e dopo ciascuna di queste compare la scritta "linea di testo" che traccia in pratica una linea di base che serve a far comprendere l'allineamento dei tre gruppi di lettere (racchiusi ognuno in una minipage) in funzione dell'apposizione dei parametri `[bct]` che è evidente osservano un diverso comportamento rispetto all'ambiente di tabelle e figure.




```
\begin{minipage}{134mm} linea di testo
\begin{minipage}[b]{17mm} AA AA AA AA AA AA AA AA AA AA AA AA AA AA AA
\end{minipage}\hspace{4mm}linea di testo
\begin{minipage}[c]{17mm} BB BB BB BB BB BB BB BB BB BB BB BB BB BB BB
\end{minipage}\hspace{4mm}linea di testo
\begin{minipage}[t]{17mm} CC CC  CC CC CC CC CC CC CC  CC CC CC CC CC CC
\end{minipage} linea di testo \end{minipage}
```

Dei tre parametri, l'unico rispondente a quanto sin qui visto è quello caratterizzato con l'opzione `[c]` (`center`). Il parametro `[t]` pone un gruppo di lettere sopra la linea di base, mentre `[b]` (che sta per `baseline` e non per `bottom`) posiziona il primo gruppo di lettere sulla linea di base. Il posizionamento è dunque al centro, in basso, in alto seguendo una immaginaria linea di base allocata sulla minipage.

È ancora da considerare che il posizionamento predisposto si può completamente *mischiare* se vengono impostati valori di minipage tali da non permettere il posizionamento della quantità di testo compreso nella minipage, se questo è cioè troppo abbondante rispetto alla *width* impostata. in questo caso il testo si disporrà cercando tutti gli spazi disponibili.

Si osservi come più minipage possono essere disposte in serie, separate eventualmente da una distanza `\hspace{xxnn}`, non esistendo altro limite che quello dato dalle dimensioni delle pagine.

Questo era uno degli esempi più articolati. Di solito le minipage si presentano singolarmente, al massimo in numero di due quando, ad esempio, in una deve essere mostrato l'output e nell'altra il sorgente.

Un *paio* di minipage affiancate si rivelano utili quando si vogliano costruire due riquadri all'interno della pagina e posizionare all'interno di questi il testo su due parti (destra e sinistra).

È molto simile, come sequenza di comandi, alle istruzioni date per posizionare le immagini affiancate che vedremo sempre nella parte dedicata alla grafica. Il sorgente mostrato genera il testo subito dopo riportato:

```
\fbox{\begin{center} \begin{minipage}{58mm}La terra era ormai divenuta una striscia....'}
\end{minipage}   \hspace{5mm}
\begin{minipage}{58mm}Le mani dolevano sempre più. Tirai i remi.... }
\end{minipage}  }\end{center}
```

La terra era ormai divenuta una striscia piccola piccola e non s'udivano altri rumori tranne il sordo brontolio del mare e quello del vento.

Le mani dolevano sempre più. Tirai i remi a bordo ed alzata la vela mi misi al timone e mi lasciai portare dal fresco vento del mattino.

È chiaro allora che la medesima costruzione può essere usata anziché per inserire parti di testo, per inserire immagini (affiancate o sovrapposte) poggiando sull'istruzione `\includegraphics`, ed è questo proprio l'uso rilevante in questa parte che interessa che ci ha condotti a questo approfondimento.

In conclusione e più semplicente, queste possono essere le istruzioni *canoniche* per la costruzione di una singola minipage racchiusa da una cornice:

Questo è un esempio di minipage racchiusa in un box. Qui appresso è prodotto il sorgente.

```
\fbox{%
\begin{minipage}{\linewidth}
Questo è un esempio di minipage racchiusa in un box. Qui appresso...
\end{minipage} }
```

L'ambiente figure

figure

figure è l'ambiente privilegiato per il posizionamento delle immagini sulla pagina. Esso si occupa esclusivamente della parte della pagina in cui l'oggetto-immagine deve essere posizionato, mentre altre operazioni di allineamento possono essere demandate ad altri ambienti come center. L'ambiente conosce anche la versione asteriscata che presenta maggiore versatilità di posizionamento.

L'ambiente figure è gestito dai quattro parametri già veduti a proposito delle tabelle: `[htbp]` che stanno, rispettivamente, come ormai è noto, per *here*, *top*, *bottom*, *page*. I parametri possono essere inseriti anche tutti insieme, abilitando un ordine di priorità.

L'eventuale inserimento del segno `[!]` prima di uno di questi specifica al sistema di forzare il posizionamento della figura nella parte di pagina indicata. In caso non venga specificato nulla è privilegiata l'opzione `[tbp]`.

Il numero delle figure ammesse per ogni pagina è definito da tre contatori che fissano, in relazione alla posizione scelta dall'utente, il numero di figure ammissibili per pagina. Questi sono:

- `\topnumber`: fissa in numero di due il massimo delle immagini posizionabili in cima alla pagina;
- `\bottomnumber`: fissa in numero di una il massimo delle immagini posizionabili in fondo alla pagina;
- `\totalnumber`: fissa in tre in numero di immagini posizionabili in una pagina.

L'ambiente figure conosce la variante asteriscata. Personalmente la preferisco alla versione non asteriscata perché mostra una maggiore flessibilità di posizionamento ed obbedienza ai comandi. Tutti gli esempi presentati in questi appunti e relativi all'introduzione di files grafici sono stati adottati con la versione asterisco dell'ambiente.

Packages per l'inserimento d'immagini: epsfig e graphicx

epsfig

Ho già accennato a epsfig; ecco come con questo package sarebbe possibile introdurre un'immogine, con questo sorgente di cui non è riportato l'output.

```
\begin{figure*}
\centerline{\epsfig{file=immagine.eps, width=4cm} \caption{Didascalia immagine} }
\end{figure*}
```

Il package entra in conflitto con graphicx se è mandato in esecuzione prima di questo. Non ho riscontrato alcun problema mandandolo in esecuzione dopo.

Figura 11.2: *Nikao III*

Varrie modalità d'inserimento di files grafici

Dopo queste premesse andremo a vedere come con l'ambiente minipage e figure e con i singoli comandi della famiglia `center` sia possibile allineare orizzontalmente e verticalmente in una pagina i files grafici, sempre più sinteticamente, da qui in avanti, chiamati immagini.

Ci occuperemo adesso, in particolare, dell'allineamento dei files (centrati, affiancati, agli estremi della pagina,...) usando o *routines* interne al sistema o l'ambiente minipage, quindi vedremo packages dedicati che consnetono strutture di posizionamento ed allineamento del tutto particolari rispetto al testo.

In figura 11.2 l'immagine è stata inserita secondo queste istruzioni proprie di `\graphicx`: graphicx

```
\begin{figure*}[!htb] \centering{%
\includegraphics{immagine.eps} \caption{didascalia \label{mia_label} }  }
\end{figure*}
```

In figura 11.2 l'immagine è rappresentata al naturale non essendo state date istruzioni di ridimensionamento. Ma spesso è indispensabile — specie nel caso di immagini plurime — fornire istruzioni particolari che nelle prossime sezioni si andrà ad esaminare.

Secondo coordinate date

Considerando quanto precedentemente detto, un'immagine si può anche posizionare secondo coordinate fornite al sistema, quali possono essere, ad esempio, le seguenti:
`\includegraphics[width=300pt,viewport=10pt 300pt 410pt 350pt]{immagine.eps}`.

Il sorgente mostrato, di cui non è presentato il relativo output, va a posizionare l'immagine secondo le coordinate ricevute, e quindi se incontra del testo vi si sovrappone con pessimo effetto se non desiderato.

Figura 11.3: Barca

Figura 11.4: Barca

In quasto caso (essendo le coordinate fornite del tutto arbitrarie e quasi irrazionali) l'immagine apparirebbe (in questo documento) una ventina di righe dopo le testatine, per un'altezza di 16 righe, allineata a sinistra.

Centrate

La *centratura* dell'immagine si effettua di norma ricorrendo all'ambiente `center` individuato, come sempre da `\begin{center}` ed `\end{center}`.

Tuttavia l'istruzione `\centering%` è da preferire in quanto `center` produce un doppio spazio verticale aggiunto prima di `\begin{center}` e dopo `\end{center}`.

Il segno di percentuale (`%`), usato in genere per commentare parti di testo da non processare, posto al termine della prima riga serve ad indicare a LaTeX di non andare a capo dopo quelle istruzioni, meglio ancora, ad evitare che venga posto uno spazio aggiuntivo del tipo di quello posto tra due righe con l'uso della doppia barra spaziatrice. Questo spazio potrebbe comportare un non perfetto allineamento centrato delle immagini.

Richiamo l'attenzione sul fatto che l'etichetta eventuale della figura, la `\label`, deve comparire vicino alla `\caption`, meglio ancora se all'interno, altrimenti in caso di plurima attivazione delle *cross-references* LaTeX non opera un corretto riferimento all'immagine.

Il sorgente mostrato rappresenta un inserimento d'immagine *scarno*, senza cioè alcuna opzione: l'unica istruzione data, oltre l'inserimento d'immagine, è quella di centrare la stessa. Quindi LaTeX ha tenuto conto soltanto delle dimensioni originarie in pixel, senza operare sull'immagine stessa interventi perché non comandati.

Trattiamo adesso la stessa immagine fornendo istruzioni opzionali ed aggiuntive.

Affiancate e riflesse

Come primo passo si farà in modo di avere due immagini affiancate, incorniciate entrambe, di una impostata dimensione, e che la seconda sia speculare rispetto alla prima, come da questo sorgente:

```
\begin{figure*}[!t]  \centering{%
     {\begin{minipage}{3cm} \centering%
     \includegraphics[width=3cm]{eps/immagine.eps}
     \caption{prima-didascalia} \label{prima-label}
     \end{minipage}  \hspace{17mm}
     \begin{minipage}{3cm} \reflectbox{\fbox{\includegraphics[width=3cm]{eps/immagine.eps} } }
\caption{seconda-didascalia} \label{seconda-label}
\end{minipage} } }  \end{figure*}
```

`\fbox` Le opzioni introdotte sono ormai di facile comprensione e riguardano i comandi già affrontati alla sezione 8.3 circa il posizionamento delle tabelle. Il dimensionamento dell'immagine è operato con

Figura 11.5: Figure agli estremi della pagina

l'istruzione `[width=30mm]`, mentre l'istruzione `\fbox{...}` racchiude l'immagine in una cornice.

Le restanti istruzioni sono quelle già apprese. LaTeX cerca di collocare il file d'immagine nel punto esatto in cui legge il sorgente, in questo caso si trova collocata a fronte, figure 11.3 e 11.4.

L'immagine in figura 11.4 presenta nuove istruzioni rispetto all'immagine 11.3: innanzi tutto `\reflectbox` che presenta una visione speculare dell'immagine, quindi la cornice attorno all'immagine introdotta con l'istruzione `\fbox`, l'inserimento dell'immagine in una minipage appositamente costruita, e l'istruzione fra parentesi quadre [!t]. `\reflectbox`

`\reflectbox` non ha bisogno di spiegazioni per la funzione assolta, anche perché l'effetto è chiaramente visibile nell'immagine. Anche questa istruzione può essere posizionata sulla linea, di modo che la scrittura `\textsf{\reflectbox{Testo riflesso}}` rende oɔɐsǝlᖵᴉɟǝɹ oʇsǝT.

L'ambiente minipage sostituisce in questo caso l'ambiente figure con una diversità sostanziale: le figure non vengono più trattate come oggetti *flottanti*, sebbene come oggetti *statici*. Compaiono quindi, come opzioni, i parametri di allineamento.

Nel caso delle figure 11.3 e 11.4 il posizionamento è avvenuto sfruttando l'ambiente minipage, precisamente due minipage all'interno delle quali si trovano du ambienti grafici con due distinte didascalie, che però l'ambiente figure* gestisce come se si avesse a che fare con una immagine.

Nel caso di testo disposto su di una sola colonna si costruiscono in pratica due (o più) minipage affiancate che accolgono le immagini.

Nel caso di testo su due colonne le immagini (se le dimensioni delle singole immagini sono quasi pari a quelle della colonna) compairiranno una sull'altra.

Un sorgente per allineare le immagini, leggermente diverso nella forma da quello prodotto alla pagina precedente, è il seguente:

```
\begin{figure}[!h] \centering%
   {\begin{minipage}{3cm} \centering
   \includegraphics[width=3cm]{eps/immagine01.eps} \caption{immagine1.eps}
   \end{minipage}  \hspace{17mm}
   \begin{minipage}{3cm} \includegraphics[width=3cm]{eps/immagine2.eps} \caption{didascalia}
   \end{minipage}}   \end{figure}
```

Agli estremi della pagina

Il posizionamento di immagini affiancate ma agli estremi dei margini della pagina, senza la minipage, *vedi* la figura 11.5, si effettua ricorrendo a `\hfill`, con queste istruzioni:

```
\includegraphics[width=20m]{eps/immagine1.eps}\hfill\includegraphics[width=20mm]
{eps/immagine2.eps} \caption{Figure agli estremi della pagina}
```

Avvicinate al centro

Immagini affiancate con spazi proporzionali (figura 11.6) si ottengono con le seguenti istruzioni che generano l'effetto di distribuire un eguale spazio prima, tra e dopo le immagini:


Figura 11.6: Figure con spazi proporzionali

```
\hfill\includegraphics[width=30mm]{eps/immagine1.eps}%
\hfill\includegraphics[width=30mm]{eps/i.eps}\hspace*{\fill}
```

In nota

Una figura come oggetto flottante può anche essere inserita in una nota come è avvenuto a titolo d'esempio per la nota [22] a pagina 24.

L'inserimento segue i comandi: `\footnote{Testo nota....\includegraphics{immagine.eps}}` senza ulteriori specificazioni.

A sfondo pagina

In questo caso siamo abbastanza fuori delle possibilità offerte da `\includegraphics`, in quanto il processo è ottenuto ricorrendo a comandi specifici del LaTeX e definendo tutti i parametri di un nuovo comando chiamato in questo caso `\ImmagineSfondo`.

Occore inolre ricorrere al package eso-pic richiamandolo nel preambolo assieme a graphicx in questa sequenza `\usepackage{eso-pic,graphicx}`, ed inserendo nel punto in cui si vuole far comparire l'immagine di sfondo queste istruzioni: [10]

```
\makeatletter
\newcommand\ImmagineSfondo[2]{%
  \setlength{\unitlength}{1pt}%
   \put(0,\strip@pt\paperheight){%
  \parbox[t][\paperheight]{\paperwidth}{%
    \vfill  \centering\includegraphics[angle=#2]{#1}
    \vfill }}} %
  \makeatother%
  \AddToShipoutPicture*{\ImmagineSfondo{file.eps}{45}}%  <-- Vedi testo
```

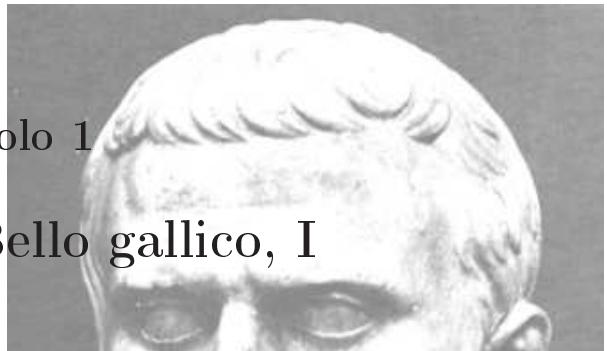
L'immagine compare *su tutte le pagine del documento* a partire dal punto in cui sono posizionate (pagina corrispondente) le istruzioni. Il valore numerico fra parentesi graffe (nel sorgente-esempio è 45°) rappresenta l'angolo di rotazione che si desidera assegnare all'immagine.

Se si desidera un'immagine a sfondo su una singola pagina, esempio mostrato in figura 11.7, si ricorrere alla versione asteriscata del comando in questa forma:
`\AddToShipoutPicture*{\BackgroundPicture{eps/immagine_a_sfondo.eps}{0}}`.

Se non si desidera l'immagine ruotata, occorre assegnare `0` come valore di rotazione, altrimenti il sistema in fase di compilazione restituisce un errore.

Nell'esempio mostrato l'immagine non copre l'intera pagina perché inserita come file d'esempio e scalata.

10. Le istruzioni derivano dal sito `www.texnik.de`.



Capitolo 1

De Bello gallico, I

Gallia est omnis divisa in partes tres, quarum unam incolunt Belgae, aliam Aquitani, tertiam qui ipsorum lingua Celtae, nostra Galli appellantur. Hi omnes lingua, institutis, legibus inter se differunt. Gallos ab Aquitanis Garumna flumen, a Belgis Matrona et Sequana dividit. Horum omnium fortissimi sunt Belgae, propterea quod a cultu atque humanitate provinciae longissime absunt, minimeque ad eos mercatores sae pe commeant atque ea quae ad effeminandos animos pertinent important, proximique sunt Germanis, qui trans Rhenum incolunt, quibuscum continenter bellum gerunt. Qua de causa Helvetii quoque reliquos Gallos virtute praecedunt, quod fere cotidianis proeliis cum Germanis contendunt, cum aut suis finibus eos prohibent aut ipsi in eorum finibus bellum gerunt. Eorum una, pars, quam Gallos obtinere dictum est, initium capit a flumine Rhodano, continetur Garumna flumine, Oceano, finibus Belgarum, attingit etiam ab Sequanis et Helvetiis flumen Rhenum, vergit ad septentriones. Belgae ab extremis Galliae finibus oriuntur, pertinent ad inferiorem partem fluminis Rheni, spectant in septentrionem et orientem solem. Aquitania a Garumna flumine ad Pyrenaeos montes et eam partem Oceani quae est ad Hispaniam pertinet; spectat inter occasum solis et septentriones.

Apud Helvetios longe nobilissimus fuit et ditissimus Orgetorix. Is M. Messala, [et P.] M. Pisone consulibus regni cupiditate inductus coniurationem nobilitatis fecit et civitati persuasit ut de finibus suis cum omnibus copiis exirent: perfacile esse, cum virtute omnibus praestarent, totius Galliae imperio potiri. Id hoc facilius iis persuasit, quod undique loci natura Helvetii continentur: una ex parte flumine Rheno latissimo atque altissimo, qui agrum Helvetium a Germanis dividit; altera ex parte monte Iura altissimo, qui est inter Sequanos et Helvetios; tertia lacu Lemanno et flumine Rhodano, qui provinciam nostram ab Helvetiis dividit. His rebus fiebat ut et minus late vagarentur et minus facile finitimis bellum inferre possent; qua ex parte homines bellandi cupidi magno dolore adficiebantur. Pro multitudine autem hominum et pro gloria belli atque fortitudinis angustos se fines habere arbitrabantur, qui in longitudinem milia passuum CCXL, in latitudinem CLXXX patebant.

Figura 11.7: Immagine a sfondo pagina

Più immagini affiancate: subfigure e subfig

subfigure si deve a STEVEN DOUGLAS COCHRAN, permette d'inserire immagini affiancate sulla stessa pagina.

Esso fa ampio riferimento al package caption ed al package subfloat. In pratica si possono creare delle miniature con gli elementi tipici di ogni immagine. Questo package è stato aggiornato dal-

Figura 11.8: Raggruppamento d'immagini

l'Autore (2004) con la creazione di un nuovo package subfig, derivato dal primo. La compatibilità col precedente package è quasi totale.

Il vantaggio principale è quello di riunire un gruppo d'immagini che mostrano fra loro una qualche attinenza. Il sorgente per un'applicazione di subfigure è mostrato alla pagina successiva. Le istruzioni fondamentali sono `\subfloat` e `\subfigbox`.

`\qquad` Un'altra modalità di raggruppare più figure assieme è alla figura 11.8, sorgente a pagina 302: si nota come anche questa scrittura ricorra all'istruzione `\qquad`. La sequenza di istruzioni opera in questo modo: prima viene creata una figura, quindi all'interno di questa si centrano le sottofigure. L'effetto diverso delle impostazioni consiste nella possibilità che ognuna delle immagini abbia una propria etichetta.

La didascalia di *riepilogo* delle immagini viene elencata in `\tableoffigures`: anche in questo caso le immagini risultano separate dall'istruzione `\qquad`.

L'istruzione `\subfigure` crea all'interno della figura un'area cui si possono associare le didascalie. La sottonumerazione delle figure è trattata dettagliatamente da D. Cochran in [7, IV].

Il sorgente per le immagini d'esempio è stato prelevato di *sana pianta* dall'url: `http://www.texnik.de/floats/subfigure.phtml`, uno dei migliori siti per quanto concerne il supporto a LaTeX.

Operando in classe memoir non bisogna richiamare il package subfloat, che è invece indispensabile nelle classi standard di LaTeX (ed in altre classi non standard), dal momento che la classe contiene una *routine* che assolve alla funzioni del package. Sempre operando nelle classi standard, il richiamo al package non è di per sé sufficiente occorrendo anche richiamare, prima delle istruzioni rappresentate alla pagina successiva, la seguente serie di comandi:

```
\newbox\subfigbox \makeatletter
\newenvironment{subfloat}%
{\def\caption##1{\gdef\subcapsave{\relax##1}}%
  \let\subcapsave\@empty%
  \setbox\subfigbox\hbox%
  \bgroup}%
{\egroup%
  \subfigure[\subcapsave]{\box\subfigbox}}%
\makeatother
```

Le istruzioni `\qquad` e `%` si occupano del posizionamento dell'immagine. Questo il sorgente delle immagini mostrate in questa pagina:

```
\begin{figure*}[!ht]\centering%
{\includegraphics[width=25mm]{eps/barca01.eps}}\qquad
{\includegraphics[width=25mm]{eps/barca01.eps}}\qquad
{\includegraphics[width=25mm]{eps/barca01.eps}}\qquad
{\includegraphics[width=25mm]{eps/barca01.eps}}\caption{Raggruppamento d'immagini}
\end{figure*}
```

Figura 11.9: Altro raggruppamento di immagini

Miniature di figure: captcont

Il package, anch'esso opera di STEVEN DOUGLAS COCHRAN, gestisce miniature di figure e permette anche di costruire tabelle di figure: risulta utile quando, attraverso una sequenza di immagini, si voglia fornire una significativa rappresentazione delle trasformazioni che l'immagine originaria ha subito nel tempo od a seguito di agenti che ne abbiano alterato la primitiva rappresentazione.

Il package va caricato nel preambolo prima del *gemello* (dello stesso Autore) subfigure ed implementato con le opzioni: `\usepackage[TABTOPCAP,FIGBOTCAP]{subfigure}`.

Le opzioni fanno sì che gli indici di tabelle e figure, `listoftables` e `listoffigures`, vengano modificati in modo da includere eventuali didascalie inserite per ogni sub-figura.

A parte questo, nel preambolo, occorre fornire, le seguenti istruzioni fornite da S. D. Cochran nel suo esempio, i cui valori (numerici) possono essere modificati dall'utente:

Sorgente per le immagini di subfigure

```
  \centering%
  \begin{subfloat}
{\includegraphics[height=30mm]{eps/w.eps}}%
\caption{\emph{height} }%
\end{subfloat}%
\hspace{1.5cm}%
\begin{subfloat}%
{\includegraphics[width=25mm]{eps/w.eps}}
\caption{\emph{width} }%
\end{subfloat}%
\hspace{1.5cm}%
\begin{subfloat}%
{\includegraphics[totalheight=35mm]{eps/w.eps}}
 \caption{\emph{totalheight} }%
 \end{subfloat}%
\hspace{1.5cm}%\reflectbox
\begin{subfloat}%
{\includegraphics[scale=0.6]{eps/w.eps}}
 \caption{\emph{scale} }%
\end{subfloat}%
\hspace{1.5cm}%
\begin{subfloat}%
{\includegraphics[width=30mm,angle=90]{eps/w.eps}}
\caption{\emph{width} ed \emph{angle} }%
\end{subfloat}%
\hspace{1.2cm}%
\begin{subfloat}%
{\includegraphics[bb=90 90 120 150]{eps/w.eps}} %
\caption{\emph{bb} }%
\end{subfloat}%{\footnotesize
\hspace{1.5cm}%
\begin{subfloat}%
{\includegraphics[width=38mm,angle=144,origin=br]
{eps/w.eps}} \caption{\emph{width}, \emph{angle}
 ed \emph{origin} }%
\end{subfloat}%
\caption{Esempi sui....   }
  \label{label}
```

```
\newcommand{\figbox}[1]{%
  \fbox{%
    \vbox to .6in{%
    \vfil \hbox to .8in{%
      \hfil    #1%
      \hfil}%
    \vfil}}}
\newcommand{\bigfigbox}[1]{%
  \fbox{%
    \vbox to 6in{%
    \vfil  \hbox to 4in{%
      \hfil    #1%
      \hfil}%
    \vfil}}}  \makeatletter
\def\subfigtopskip{4pt}
\def\subfigbottomskip{4pt}
\def\subfigcapskip{2pt}
```

La legenda delle istruzioni ormai è chiara: il package ridisegna una serie di box in cui verranno poi posizionate le singole figure.

Le altre istruzioni da fornire all'interno del documento sono nella forma:

```
\setcounter{lofdepth}{2}  \listoffigures  \clearpage
\setcounter{lotdepth}{2}  \listoftables   \clearpage
```

vanno ad integrare le classiche istruzioni `listoffigures` ed `listoftables`.

L'indice di *profondità* `lofdepth` espresso nell'esempio dal numero `{2}` può essere naturalmente modificato a piacere dall'utente.

```
\centering%
   \subfigure[{I}]%
   {\includegraphics[width=25mm]{eps/immagine.eps}}\qquad\qquad%
   \subfigure[{II}]%
   {\includegraphics[width=25mm]{eps/immagine.eps}}\qquad\qquad%
   \subfigure[{III}]%
   {\includegraphics[width=25mm]{eps/immagine.eps}}\qquad\qquad%
   \subfigure[{IV}]%
   {\includegraphics[width=25mm]{eps/immagine.eps}} \qquad\qquad%
   \subfigure[{V}]%
   {\includegraphics[width=25mm]{eps/immagine.eps}}\qquad\qquad%%
   \subfigure[{VI}]%
   {\includegraphics[width=25mm]{eps/immagine.eps}}\qquad\qquad%%
   \subfigure[{VII}]%
   {\includegraphics[width=25mm]{eps/immagine.eps}}\qquad\qquad%%
   \subfigure[{VIII}]%
   {\includegraphics[width=25mm]{eps/immagine.eps}}\qquad\qquad%%
   \caption{Altro raggruppamento di immagini\label{label}}
```

Un sorgente-tipo relativo al posizionamento delle figure-miniature è dato dalle seguenti linee:

```
\begin{figure}  \begin{center}%
    \subfigure{\figbox{\includegraphics[width=xxmm]{foto1.eps}}} \hspace{10pt}%
    \subfigure{\figbox{{\includegraphics[width=xxmm]{foto2.eps}}}}\\
    \subfigure{\figbox{{\includegraphics[width=xxmm]{foto3.eps}}}} \hspace{10pt}%
    \subfigure{\figbox{{\includegraphics[width=xxmm]{foto4.eps}}}} \end{center}
  \caption{Miniature}\end{figure}
```

Il risultato prodotto è visibile a pagina 304. Le singole immagini vengono nominate con le prime lettere dell'alfabeto.

Va notato a questo proposito il diverso comportamento dell'istruzione `[\label{figura_A}]` dopo l'istruzione `\subfigure`.

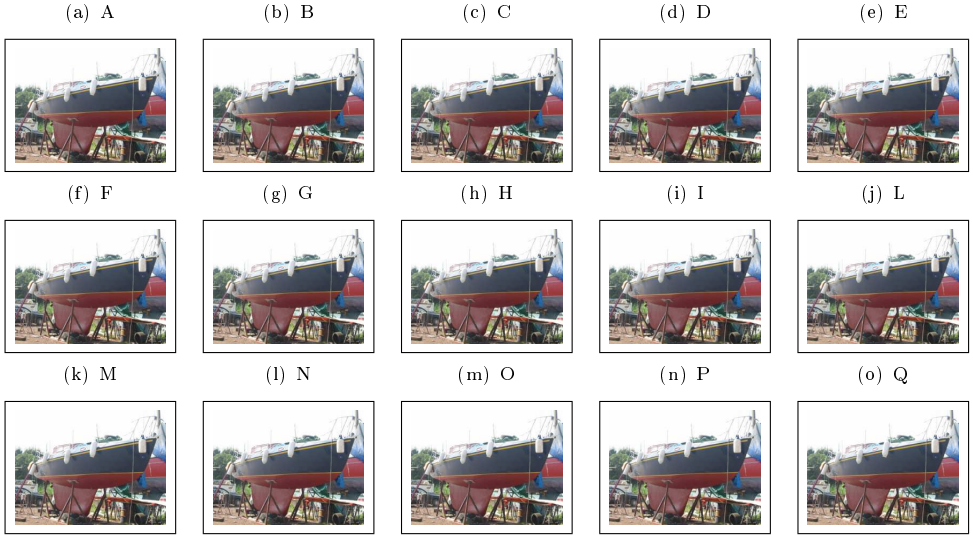


Tabella 11.4: Tabelle di figure

In questo caso la label non si limita a svolgere solo le funzioni di etichetta di riferimento per una eventuale *cross-reference*, bensì presiede anche al posizionamento numerico della lettera dell'alfabeto latino che individua l'immagine: mantenendo le due parentesi quadre [ ], anche senza alcuna specificazione all'interno delle stesse, viene posizionata la lettera indicativa, togliendole si ha l'immagine senza lettera di riferimento.

Il package contiene un contatore che si occupa di questa necessità, ma l'autore avverte che si tratta di un'istruzione delicata, e richiama alla prudenza per ogni eventuale modifica effettuata: [8, IV], pag. 11. Nell'esempio prodotto, S. Cochran (dopo `\begin{document}`) pone istruzioni tipiche come: `\setcounter{lofdepth}{2}\listoffigures\clearpage \setcounter{lotdepth}{2}\listoftables \clearpage` finalizzate ad eviate incongruenze.

Un'altra potenzialità notevole del package è quella di creare tabelle *di immagini* continue, tramite la definizione di nuovi comandi (quali `\subtable`) che ridefiniscono dei veri e propri miniambienti. A questa nuova istruzione se ne trova associata un'altra di didascalia: `\captcont`, che si presenta anche nella variante asterisco. `\subtable`

Se per ogni serie di immagini si fosse usata la variante asterisco nella forma `\captcont*` le immagini sarebbero state rappresentate senza riferimento, in puro elenco dopo l'ultima didascalia. `\captcont*`

Le immagini prodotte in questo nuovo ambiente compaiono in questa pagina. La specificazione per ogni `subtable`, fra parentesi quadre dell'eventuale istruzione del tipo `[\label{tab:sub1_1_a}]` permetterà di attivare le *cross-references* per quella singola immagine. Il sorgente è presentato alla pagina successiva.

Come si può notare andando a visionare l'indice delle tabelle queste immagini compaiono con il loro proprio nome (anche se in questo caso è una lettera), e la posizione delle immagine stesse appare (nell'indice delle tabelle) riferita alla didascalia dell'immagine.

Sorgente per le immagini tabellari mostrate alla pagina precedente: (captcont)

```
\subtable[{E}]{\figbox{{\includegraphics
    [height=15mm] {eps/file.eps}}}}%
    \hspace{10pt}%
    \subtable[{F}]{\figbox{{\includegraphics
    [height=15mm] {eps/file.eps}}}}%
    \hspace{10pt}%
    \subtable[{G}]{\figbox{{\includegraphics
    [height=15mm]  {eps/file.eps}}}}%
    \hspace{10pt}%
    \subtable[{H}]{\figbox{{\includegraphics
    [height=15mm]{eps/file.eps}}}}%
    \hspace{10pt}%
    \subtable[{I}]{\figbox{{\includegraphics
    [height=15mm] {eps/file.eps}}}}%
    \hspace{10pt}%
    \subtable[{J}]{\figbox{{\includegraphics
    [height=15mm] {eps/file.eps}}}}%
    \hspace{10pt}%
    \subtable[{H}]{\figbox{{\includegraphics
    [height=15mm] {eps/file.eps}}}}%
    \hspace{10pt}%
    \subtable[{L}]{\figbox{{\includegraphics
    [height=15mm] {eps/file.eps}}}}%
    \hspace{10pt}%
    \subtable[{M}]{\figbox{{\includegraphics
    [height=15mm] {eps/file.eps}}}}%
    \hspace{10pt}%
     %------------
    \subtable[{N}] {\figbox{{\includegraphics
    [height=15mm] {eps/file.eps}}}}%
    \hspace{10pt}%
    \subtable[{O}] {\figbox{{\includegraphics
    [height=15mm] {eps/file.eps}}}}%
    \hspace{10pt}%
    %------------
    \subtable[{P}] {\figbox{{\includegraphics
    [height=15mm]{eps/file.eps}}}}%
    \hspace{10pt}%
    \subtable[{Q}] {\figbox{{\includegraphics
    [height=15mm]{eps/file.eps}}}}%
    \hspace{10pt}%
    \subtable[{R}] {\figbox{{\includegraphics
    [height=15mm]{eps/file.eps}}}}%
    \hspace{10pt}%
    \subtable[{S}]{\figbox{{\includegraphics
    [height=15mm] {eps/file.eps}}}}%
    \listsubcaptions
```

Figura 11.10: Miniature con captcont

Stop dei processi flottanti

Esaminiamo ora un problema che talvolta si verifica quando si ha che fare con testi complessi forniti di un notevole numero di immagini e tabelle (oggetti flottanti). Può capitare di vedere arrestarsi la compilazione e di leggere il diagnostico: `! LaTeX Error: Too many unprocessed floats.`

Il problema non è grave, perché se si forza la compilazione ad andare avanti questa procede e senza errori. Tuttavia bisogna tener conto che in quel punto LaTeX ha segnalato una presenza *ingombrante* di oggetti flottanti, ed è bene risolvere subito il problema. Tanto per spiegarmi al

meglio, nel presente lavoro io ho avuto questo diagnostico dopo l'inseriemnto del package hvfloat (*vedi* a pagina 316), e l'errore mi veniva segnalato nella parte successiva di questo lavoro, (la V), e parecchio avanti, su un comando di `\marginpar`, trattato anch'esso da LaTeX come oggetto flottante.

Se un'oggetto flottante non può essere immediatamente processato, esso viene piazzato in una sorta di *limbo* di oggetti non gestiti finché, appunto, non viene processato. Siccome il limite di LaTeX in questa coda è di 18 oggetti, quando il limite è superato viene restituito il diagnostico riportato. Il contrasto risiede sovente nell'opzione di posizionamento `[htb]` fornita agli oggetti: se questa è identica, ad esempio `[!t]` per tutti gli oggetti, il problema può facilmente verificarsi.

A parte l'uso del comando di sistema `\suppressfloat[tb]` che posto immediatamente prima della figura ne previene lo scivolamento in altre parti del documento, esistono due modi di avvicinarsi al problema, e questi sono in funzione dello stato del lavoro. `\suppressfloat`

Mi spiego: se si è appena iniziato a scrivere un capitolo e gli oggeti (immagini e tabelle) sono state *buttati* lì senza alcun criterio e devono ancora essere spostati, posizionati, ridimensionati,... è inutile preoccuparsi del problema; quasi sicuramente ponendo ordine nel lavoro, il problema si eliminerà da solo. Se invece si è quasi alla fine del lavoro il problema va affrontato e risolto: come ho già detto, non è bene trascinarsi dietro diagnostici d'errore.

Occorre in questo caso fornire al sistema le necessarie istruzioni per far terminare i processi flottanti invasivi tramite alcuni comandi, di cui è comunque raccomandato un uso parco. I comandi sono:

- il classico e già conosciuto `\clearpage`, che facendo iniziare una nuova pagina ferma i processi in corso. Non sempre questo comando dà gli effetti desiderati, specie quando gli oggetti flottanti sono numerosi e vicini;
- ricorrere al package placeins, [11] nella forma `\usepackage[section]{placeins}`, soluzione adottata. Occorre inserire l'ulteriore comando `\FloatBarrier`, il quale va posizionato immediatamente prima della sezione che segue l'oggetto che ha creato problemi. `\FloatBarrier`
- la variante di placenis: `\usepackage[below]{placeins}`, è meno restrittiva della precedente, e consente agli oggetti flottanti di essere posizionati dopo l'inizio di una nuova sezione.
- si può infine ricorrere al package afterpage già visto nella parte II a pagina 129. Il comando tipico del package dato nella forma `\afterpage{clearpage}`, è usato anche per gestire piccole pagine con oggetti flottanti.

Per un approfondimento si consiglia di consultare *Using Imported Graphics in LaTeX 2ε* di KEITH RECKDAHL [16, IV, pag. 44 e segg.] .

11.4 Figure ruotate sulla pagina: lscape e rotating

landscape e rotating sono due packages che prendono, in modalità diverse, il predominio della pagina, occupandosi di collocare un'immagine sull'intera pagina, accogliendo, a seconda dei casi, testo ed immagine, rotando l'immagine.

lscape

Il package lscape che si deve a DAVID CARLISLE, parte integrante dei pacchetti di grafica forniti assieme a LaTeX, presiede al collocamento delle figure nell'omonimo ambiente.

La pagina viene quindi ruotata assieme al suo contenuto e le istruzioni si limitano ad un `\begin{landscape}` ed un `\end{landscape}`: i restanti comandi non abbisognano di spiegazioni.

La differenza fra quest'ambiente ed altri che si comportano all'identica maniera (*vedi* appresso rotating) risiede nel fatto che landscape colloca le figure su una pagina apposita e che può accogliere

11. Il package si deve a DONALD ARSENEAU.

sia testo, quanto figure che tabelle. In aggiunta lscape può essere usato con ottimi risultati in congiunzione con il package endfloat. Un esempio di ambiente lscape, con relativo sorgente, è mostrato a pagina 318.

rotating

\sideways-figure

Il package rotating tramite l'appropriata istruzione d'ambiente sidewaysfigure consente, con piccole varianti, di ottenere i medesimi risultati del precedente package, e come il precedente accetta anch'esso tabelle, figure e grafica.

Data la similarità dei risultati di quest'ultimo è mostrato solo il sorgente:

```
\begin{sidewaysfigure}
\includegraphics[width=10cm]{immagine.eps}
\caption{figura con il package sidewaysfigure}
\label{mia_label}
\end{sidewaysfigure}
```

Grafica con rotating

rotating permette anche il trattamento di parti di testo in modalità grafica come nell'esempio mostrato in figura 11.11. [12]

L'esempio riprodotto rappresenta in una certa maniera, alla fine di questo capitolo, il passaggio dall'inserimento di immagini già composte ad una grafica più avanzata che sarà esaminata in un successivo capitolo.

11.5 Figure incorniciate nel testo

Alcuni packages sovrintendono a questa necessità. Ne passerò in rassegna soltanto alcuni: window, wrapfig, picinpar (che contempla due ambienti) e floatflt. Sono tutti *ruvidi*, abbastanza ostici, ed in fondo di non grande utilità.

window

Questo package si deve a ELMAR SCHALÜCK e consente di posizionare un'immagine nel corpo del testo, come per la foto incorniciata in riquadro in figura 11.12. Diversamente da floatflt (*vedi* appresso) questo non mi ha dato eccessivi problemi.

12. Il sorgente del grafico mostrato a fronte e mostrato qui appresso, si trova in tutti i testi di LaTeX dedicati alla grafica a cominciare da quello di HERBERT VOSS citato in bibliografia [18, IV], e da lì è stato ripreso. In quei testi la scritta è rappresentata come una ruota. Io ho ottenuto questa struttura con piccole modifiche. Questo il sorgente:

```
\usepackage[dvips]{graphicx} \usepackage{rotating}
\newcount\wang \newsavebox{\wangtext} \newdimen\wangspace
\def\wheel#1{\savebox{\wangtext}{#1}%
  \wangspace\wd\wangtext  \advance\wangspace by 10mm%
  \centerline{%
    \rule{0pt}{\wangspace}%
    \rule[-\wangspace]{0pt}{\wangspace}%
   \wang=-180\loop\ifnum\wang<360 \rlap{\begin{rotate}{\the\wang}
       \rule{10mm}{0pt}#1\end{rotate}} \advance\wang by 10\repeat}}
\wheel{La Grafica con \LaTeXe }
```

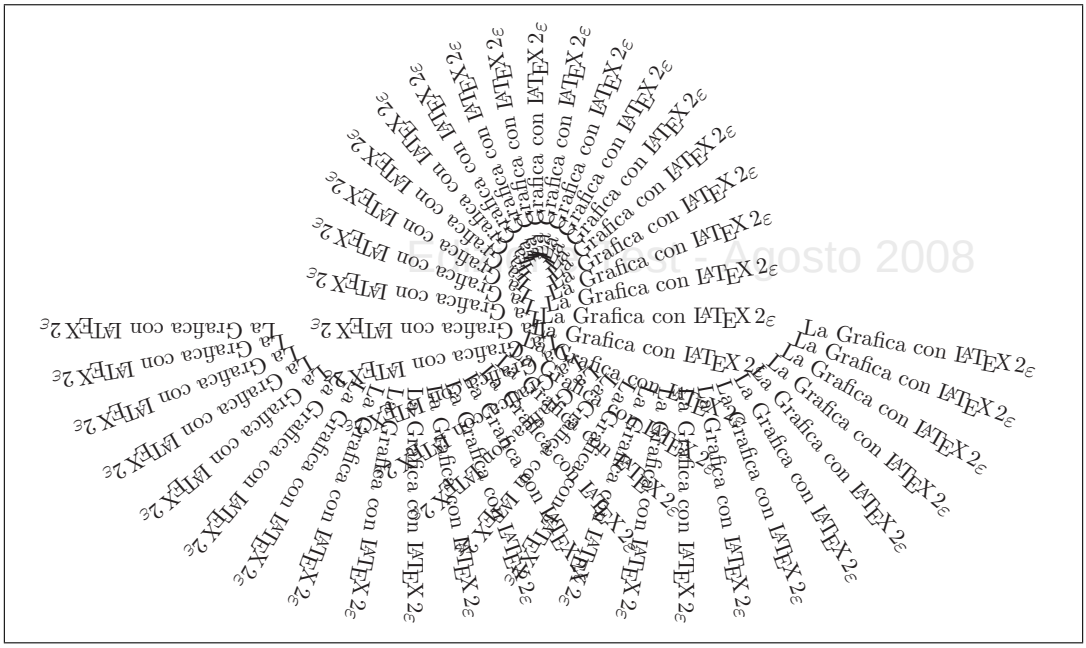


Figura 11.11: Grafico con rotating

Esaminiamo le istruzioni da dare dopo aver fatto una precisazione: il paragrafo che incornicia l'immagine deve essere abbastanza *lungo*, anche se questo è purtroppo antiestetico in quanto spezzare il paragrafo con dei rientri a capo porta ad una impostazione non solo ancor più antiestetica, ma ancora meno efficiente di quanto si vuole mostrare.

Nella compilazione di questo sorgente ho avuto continui messaggi d'errore in fase di compilazione prima di raggiungere un equilibrio accettabile. Non ho potuto neanche inserire degli `\item` descrittivi perché la figura non sarebbe risultata incorniciata. È questa la *rusticità* cui facevo cenno sopra e consiglio di ricorrere al package soltanto se ce n'è vero bisogno. I *sacri* testi citati nella bibliografia (BENJAMIN BAYART, [2, II]) avvertono che il problema si potrebbe risolvere facendo leggere nel preambolo window prima di babel. Purtroppo non ho riscontrato l'esattezza di questa affermazione per nessuna delle classi testate, e non dolo per quella in uso (memoir). Credo che si debba convenire con Bayart sulla considerazione che questo package è un *residuato* del vecchio LaTeX2.09 e che quindi mal si adatta alla nuova impostazione del LaTeX 2ε.

Questo package ha un'altra caratteristica: se usato insieme ad altri packages che fanno uso di contatori,[13] li fa letteralmente *schizzare* fuori numero, rendendo un errore in fase di compilazione. Per questo motivo, per realizzare l'esempio riportato alla pagina seguente, sono ricorso ad un file postscript precedentemente composto e compilato che poi ho inserito come immagine con il suffisso .ps, operazione spiegata in dettaglio alla sezione 6.4.

Le regole sono le seguenti:

- `toplines:[n]`: indica le linee da lasciare prima di posizionare l'immagine: come si nota dal sorgente mostrato si è specificato di lasciare sei righe sopra la figura;

13. A proposito dei problemi che questo package può dare, *vedi* a pagina 156.

Figura 11.12: A sinistra esempio del package window, a destra del package wrapfig

- `inwindow:image`: specifica l'inserimento nominale dell'immagine;
- `ratio n n`: per *ratio* s'intende la collocazione dell'immagine:
  - `ratio 0 1` specifica di posizionare l'immagine a sinistra del testo;
  - `ratio 1 1` specifica di posizionare l'immagine al centro (caso in questione);
  - `ratio 1 0` specifica di posizionare l'immagine a destra del testo.

Ancora non è tutto. Il paragrafo in cui è contenuta l'immagine deve iniziare con l'istruzione `\parskip1ex` e deve terminare con `\par`, deve essere racchiuso fra due parentesi graffe altrimenti si modifica la spaziatura fra paragrafi.

Ecco il sorgente per l'immagine mostrata in figura 11.12, a sinistra:

```
{\parskip1ex \windowbox[toplines:6][inwindow:{\fbox{\shortstack{\includegraphics[width=1cm]
{immagine.eps}}}}]%
[ratio: 1 1] %
---- t e s t o ----   ----t e s t o----  ----t e s t o----   ----t e s t o----      \par }
```

L'istruzione mostrata deve essere attaccata alle righe di testo.

wrapfig

wrapfig si deve a DONALD ARSENEAU e nonostante l'autorità della firma è anch'esso un package che a quano potuto rilevare dà adito a parecchi problemi, specie per quel che riguarda il posizionamento.

Ha una differenza rispetto agli fogli di stile che è sostanziale: permette di gestire non solo figure e le tabelle, e così gli ambienti generati dal foglio di stile (wrapfigure e wraptable posizionano la figura o la tabella all'interno di porzioni di tetso, anche se tale posizionamento è spesso un poco difficoltoso e costringe a continui aggiustamenti –lo ripeto– per quanto ho potuto constatare.

Il posizionamento della figura non è limitato alle solite tre opzioni, ma è governato da un'abbondanza di potenziali parametri che sono: `r l i o R L I O`.

Le prime tre lettere stanno, naturalmente, per i parametri *right* e *left*, mentre le restanti indicano, rispettivamente, *inside*, *outside*. Le lettere maiuscole specificano gli stessi parametri ma in valori assoluti forzando la posizione dell'immagine.

Figura 11.13: picinpar: a sinistra ambiente window, a destra ambiente figwindow

L'immagine mostrata alla figura 11.12 a destra, e di cui appresso è riportato il sorgente, mostra le istruzioni date per generare la figura.

Alla riga 1) si trova commentata l'istruzione `[0.1\width]` perché mi ha generato dei problemi in fase di compilazione: in altri casi può rivelarsi invece utilissima.

```
---- t e s t o ----   ----t e s t o----  ----t e s t o----   ----t e s t o----
1) \begin{wrapfigure}[5]{l}{0pt}%[0.1\width]
2) \centering%
3) \includegraphics[width=25mm]{immagine.eps}%
4) \caption{Didascalia}
5) \end{wrapfigure}
6) ---- t e s t o ----   ----t e s t o----  ----t e s t o----   ----t e s t o----
```

picinpar

Il package che si deve a FRIEDHELM SOWA [14] deve considerarsi un residuo del LaTeX2.09, essendo datato (1993). Lo si riporta per la *remota* necessità cira il suo uso.

Come tale si consiglia di usarlo con parsimonia ed in documenti che si articolino su pochi packages, perché anche questo è destinato a far salire il numero dei contatori in maniera esponenziale.

Il package contempla due ambienti fondamentali: window e figwindow, a seconda che nella finestra generata si voglia posizionare del testo od un'immagine.

14. L'autore del package nelle prime righe del foglio di stile tributa un credito di riconoscimento nei confronti di ALAN HÖNIG, primo creatore di un essenziale package che assolveva alle medesime funzioni.

ambiente window

L'effetto che si vuole ottenere, visibile nella figura 11.5, è quello di posizionare una qualsiasi scritta all'interno di una porzione di testo.

L'effetto desiderato è voluto grazie a queste poche istruzioni:

```
\begin{window}[14,c,%
      \large{%
      \textbf{%
      \fbox{\shortstack{L\\i\\n\\u\\x}}}},]
      testo testo testo testo testo testo
      ..................................
      testo testo testo testo testo testo
      \end{window}
```

Le opzioni si riducono a due:

- il numero delle righe dopo le quali si desidera che compaia la scritta verticale (in questo caso 14),
- l'allineamento della scritta stessa che nel caso dell'esempio risulta centrata: le opzioni sono dunque `c` per *center*, `l` per *left*, `r` per *right*.

`\shortstack` Non esistono altre avvertenze da dare né sull'istruzione `\shortstack` che governa il posizionamento della scritta, né sulle modalità di posizionare la scritta, se si eccettua il fatto che la rudimentalità dell'istruzione richiede che ogni singola lettera, per avere un effetto appena gradevole, sia separata dal doppio segno di backslash.

ambiente figwindow

L'ambiente presenta sostanziale identità con il precedente, ma sovrintende al posizionamento di un'immagine e non di una scritta. La figura 11.5 mostra anche in questo la scarnità di comandi.

Una volta attivato l'ambiente figwindow, resta soltanto da determinare il numero delle righe dopo le quali si vuole far comparire l'immagine e l'allineamento di questa determinato, come di consueto, dalle lettere `l`, `c`, `r`.

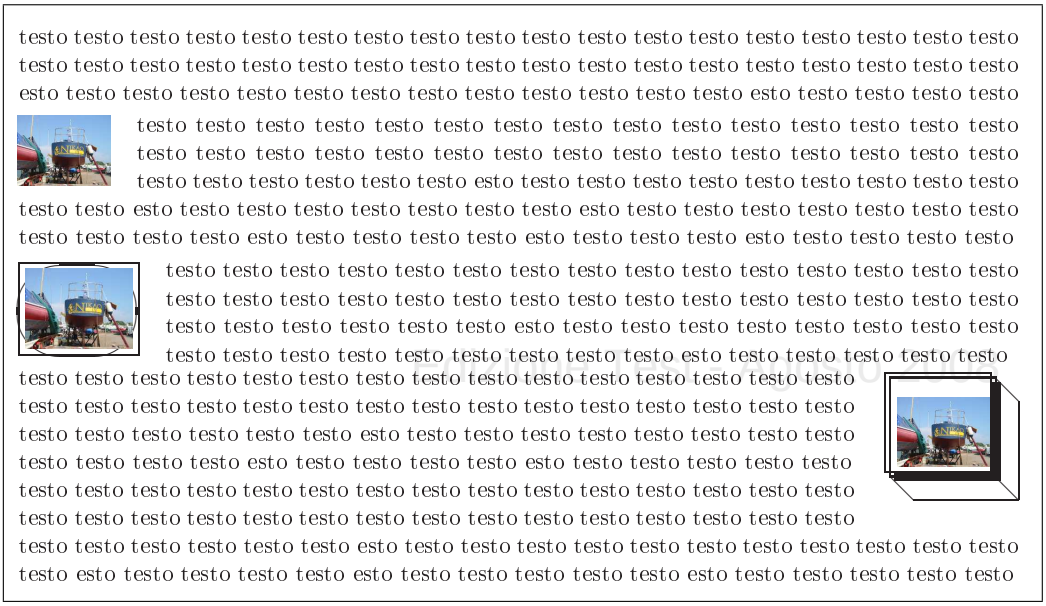
```
\begin{figwindow}[6,r,%
      \fbox{\includegraphics[width=50mm]{immagine.eps}},%
                           {didascalia}]
      testo testo testo testo testo testo
      testo testo testo testo testo testo
\end{figwindow}
```

picins

picins di JOACHIM BLESER e EDMUND LANG indexnomi propri!persone!Bleser Joachim indexnomi propri!persone!Lang Edmund che permette di posizionare l'immagine a destra o sinistra del testo. Rispetto agli applicativi finora visti presenta una maggiore stabilità

Il package introduce una nuova istruzione `\parpic` che attraverso una serie di opzioni determina
`\parpic` il posizionamento dell'immagine.

L'istruzione completa si presenta nella forma `\parpic(w,h)(x,y)[opz.][pos.]{immagine}`, dove i primi valori fra parentesi tonde rappresentano la larghezza e l'altezza dell'immagine; la seconda serie di valori fra parentesi tonde rappresenta il *segnaposto* dell'immagine riferito all'angolo superiore sinistro dei suoi `BoundingBox`: se la specifica è assente il comando usa gli argomenti di `[pos]`; l'argomento `[opz]` esprime le opzioni che possono essere `l` o `r` (immagine a destra o sinistra del testo), ovvero `d` (che sta per *dash* (linea tratteggiata), `o` che sta per *oval*, `f` che sta per *frame*, `s` che sta per *shadow*, ed infine `x` che sta per *box*.



testo testo testo testo testo testo testo testo testo testo testo testo testo testo testo testo testo testo
testo testo testo testo testo testo testo testo testo testo testo testo testo testo testo testo testo testo
esto testo testo testo testo testo testo testo testo testo testo testo testo esto testo testo testo testo
testo testo testo testo testo testo testo testo testo testo testo testo testo testo testo
testo testo testo testo testo testo testo testo testo testo testo testo testo testo testo
testo testo testo testo testo testo esto testo testo testo testo testo testo testo testo testo
testo testo esto testo testo testo testo testo testo testo esto testo testo testo testo testo testo testo
testo testo testo testo esto testo testo testo testo esto testo testo testo esto testo testo testo testo

testo testo testo testo testo testo testo testo testo testo testo testo testo testo testo
testo testo testo testo testo testo testo testo testo testo testo testo testo testo testo
testo testo testo testo testo testo esto testo testo testo testo testo testo testo testo
testo testo testo testo testo testo testo testo testo esto testo testo testo testo testo
testo testo testo testo testo testo testo testo testo testo testo testo testo testo testo
testo testo testo testo testo testo testo testo testo testo testo testo testo testo testo
testo testo testo testo testo testo esto testo testo testo testo testo testo testo testo
testo testo testo testo esto testo testo testo testo esto testo testo testo testo testo
testo testo testo testo testo testo testo testo testo testo testo testo testo testo testo
testo testo testo testo testo testo testo testo testo testo testo testo testo testo testo
testo testo testo testo testo testo esto testo testo testo testo testo testo testo testo testo testo testo
testo esto testo testo testo testo esto testo testo testo testo testo esto testo testo testo testo testo

Figura 11.14: Applicazioni di picins

Si faccia attenzione che l'ordine di inserimento di queste opzioni è rilevante, e serie diverse creano costruzioni garfiche diverse.

Il posizionamento dell'immagine, anch'esso argomento opzionale, colloca con le lettere `l` e `r` (*left* e *right*) l'immagine alla sinistra o alla destra della cornice, mentre collocamenti verticali sono ammessi con `t` (*top*) e `b` (*bottom*).

Infine fra parentesi quadre il nome dell'immagine da includere.

Una possibile costruzione è mostrata in figura 11.14, e qui di seguito è riportato il sorgente.

```
\usepackage{graphicx,picins}
\noindent testo testo ......
\parpic[d]{\includegraphics[width=14mm]{immagine}}
\noindent testo testo ......
\parpic(18mm,14mm)[f]{\includegraphics[width=16mm]{immagine}}
\noindent testo testo ......
\parpic(16mm,15mm)(2mm,14mm)[rbsx]{\includegraphics[width=14mm]{poppa}}
\noindent testo testo ...... \par  \picskip{2}
\noindent testo testo ......
```

picins presenta notevoli potenzialità nella costruzioni di box particolari e le sue applicazioni saranno esaminate a quel capitolo.

floatflt

floatflt si deve a MATS DAHLGREN: questo posiziona –quando gli va!– una figura all'interno di un paragrafo e si mostra estremamente utile nel caso di testo scritto su colonne. Ma anche questo pacchetto è un residuo del LaTeX2.09.

Nell'elaborazione della prima versione di questi *Appunti*, il package ha smesso di funzionare (sic!) senza alcun plausibile motivo dopo qualche mese che lo usavo. Installato, reinstallato, ricompilato, letto il sorgente, scaricato un nuovo foglio di stile,... tutto inutile. Il mancato funzionamento risede certo in qualche conflitto d'istruzioni con altri packages installati successivamente: la cosa strana è che quello stesso sorgente posizionato in altre parti e capitoli funziona!

Come si vede dalle linee mostrate,

```
\begin{figure}[!h]
\begin{floatingfigure}[r]{4.0cm}
\includegraphics[width=xx\textwidth]{immagine.eps} \caption{Didascalia figura nel testo}
\end{floatingfigure}
\end{figure}
```

le istruzioni si riducono a due: `\begin{floatinfigure}` ed `\end{floatinfigure}`. floatinfigure può contenere opzioni in questa forma: `\begin{floatinfigure}[x]{width}` dove la "x" va sostituita con una delle seguenti lettere:

`r`: forza la figura a comparire alla destra del paragrafo,
`l`: forza la figura a comparire alla sinistra del paragrafo;
`p`: forza la figura a comparire a destra del paragrafo se il numero di pagina è dispari, a sinistra se è pari;
`v`: applica le opzioni del package: se le opzioni non sono specificate fa comparire la figura a destra del paragrafo se il numero della pagina è dispari ed a sinistra se è pari.

Fra queste istruzioni ci sono, ovviamente, i rituali comandi d'inclusione di grafica e di specifica dell'immagine ormai appresi, cioè `\includegraphics` e `\caption`.

Il package conosce delle opzioni che sono:

`rflt`: forza il package a far flottare gli oggetti a destra del paragrafo;
`lflt`: forza il package a far flottare gli oggetti alla sinistra del paragrafo;
`vflt`: è il default.

Anche in questo caso, come per window, le istruzioni che presiedono al posizionamento della figura non devono essere *staccate* dal testo che precede e segue. In conseguenza si esige che sia digitato un testo abbastanza lungo in modo che l'immagine creata venga incorniciata sopra, sotto ed a lato al fine di evitare antiestetiche sovrapposizione di testo. In aggiunta le istruzioni, se poste nelle immediate vicinanze del precedente window fanno sì che il testo si vada a sovrapporre all'immagine, il foglio di stile si *urta* se vicino c'è un ambiente verbatim, ecc. ecc. ecc.

`\bigskip` L'istruzione `\bigskip` alla quinta riga perché può avere bisogno di aggiustamenti, e sul fatto che prima dell'inizio della minipage il testo debba terminare con il doppio segno di backslash (\\) altrimenti l'immagine risulterà indentata.

`\centering` L'istruzione `\centering %` presente alla linea due del sorgente va tolta se si desidera ottenere il bordo dell'immagine allineato al testo, come è stato fatto nel presente caso.

Queste le istruzioni:

```
TESTO SOPRA L'IMMAGINE........ \\
1 \begin{minipage}{.30\textwidth}%
2 \centering%
3 \includegraphics[width=.95\textwidth]
4 {immagine.eps}\\%
5 \hbox{\kern 6em {\emph{\small didascalia immagine}}}\bigskip \end{minipage}%
6 \begin{minipage}{.70\textwidth}%
7 TESTO A FIANCO DELL'IMMAGINE........  %
8 \end{minipage}\\
TESTO SOTTO L'IMMAGINE........
```

Se dovessero sorgere problemi si ricomprendano le istruzioni fra `\begin{figure*}` ed `\end{figure*}`.

Nikao III pronto per il varo

A volte, ancora, le istruzioni si portano via intere porzioni di testo (generalmente quelle a loro seguenti) e le fanno *flottare* lì dove nessuno le *vede* più... senza che in fase di compilazione si abbia un solo messaggio di errore.
In conclusione: meglio non usarlo, e se proprio ci fosse bisogno di un effetto del genere, ricorrere alla creazione di una minipage e ricordate le istruzioni appena date.
Il testo *abbastanza... lungo*, di cui si diceva sopra, fa sì che l'immagine creata venga incorniciata sopra, sotto ed a lato per evitare antiestetiche sovrapposizione di testo.
L'immagine in riquadro presente in questa pagina è stata creata ricorrendo ad una minipage.

11.6 Le didascalie

L'inserimento standar delle didascalie per le immagini (come per le tabelle) avviene con l'istruzione `\caption`; quest'argomento è stato affrontato più volte nel corso della parte II, specie nel capitolo 8. Tanto per le tabelle quanto per le immagini il font (ed il relativo corpo) usati sono gli stessi di quelli del testo. Può risultare più gradevole *staccare* queste scritte dalla restante parte del documento, sia usando un altro set di caratteri, sia usando un font di corpo più piccolo con lo stesso set.

L'uso italiano prevede infatti, almeno nella generalità dei casi esaminati, che il corpo-testo della didascalia sia leggermente scalato rispetto a quello presente nella restante parte del testo, scrittura che, personalmente, trovo più elegante ed adatta, e questo stile ho seguito per le didascalia in uso nel documento. Per evitare comandi dedicati per ogni singola didascalia del tipo, ad esempio, di `\caption{\ffotnotesize{testo didascalia}}`, occorre dare istruzioni che valgano per l'intero documento, e di queste ci occuperemo ora.

Personalizzazione delle didascalie

Nel presente documento, per mutare la scrittura io ho digitato queste istruzioni:

```
\captionnamefont{\small}
\captionstyle{\small}
```

Ovviamente i comandi posti fra parentesi graffe possono essere sostituiti da altri secondo il gradimento, come, ad esempio, `\sffamily`.

caption

Una prima semplice personalizzazione si può ottenere richiamando il package caption di HARALD AXEL SOMMERFELDT richiamando nel preambolo queste istruzioni:

```
\usepackage{graphic,caption}
\DeclareCaptionStyle{italic}
  {labelfont={sf,bf},textfont={rm,it},indention=15pt,
```

```
  labelsep=period,justification=raggedright}
    \captionsetup[figure]{style=italic}
    \captionsetup{style=default,labelfont={sf,bf}}
```

Successivamente, le istruzioni verranno introdotte in questo modo:

```
\begin{figure*}[t]
\centering{%
\includegraphics{immagine.eps}
\caption{Didascalia figura}      }
\end{figure*}
```

Analoga procedura si può ovviamente usare per le didascalie in tabelle. Il sorgente è tratto da [17, II, pagg. 312-313].

caption2

In alternativa si può ricorrere all'uso del package caption2 che si deve anch'esso a HARALD AXEL SOMMERFELDT, che permette di procedere alla personalizzazione delle didascalie.

\caption

Se nel preambolo è già presente l'istruzione `\usepackage{caption}`, occorre commentarla perché interagisce negativamente con questo package restituendo un errore. Il package va richiamato assieme ad eventuali opzioni nella forma `\usepackage[sc]{caption2}`.

Per rendere sensibile LaTeX alle nuove impostazioni va anche ridefinito un comando in una forma *simile* a questa: `\renewcommand{\captionlabeldelim}{\space\$\rightarrow\$}`.

Nelle immagini che non sono mostrate perché queste istruzioni mostrano contrasti con la classe in uso (testate sulle classi standard hanno mostrato validità), appariranno le applicazioni di questa istruzione mostrando nella prima una freccia che indica l'immagine (istruzione `\rightarrow`), e nella seconda (istruzione `\RIGHTarrow`) un indicatore.

```
\begin{figure}[!ht]
     \centering%
     \mbox{%
     \begin{minipage}{.30\textwidth}%
     \includegraphics[width=\textwidth]{eps/immagine-a.ps}%
     \end{minipage}%
     \quad%
     \begin{minipage}[c]{.35\textwidth}%
     \caption*{{\footnotesize Prima didascalia}}
     \caption*{{\footnotesize Seconda didascalia}}
     \end{minipage}%
     \quad%
     \begin{minipage}{.30\textwidth}%
     \includegraphics[width=\textwidth]{eps/immagine-b.ps}
     \end{minipage}%
   }%
\end{figure}\label{label}
```

Una possibilità di posizionare le didascalie al centro delle immagini è dato dal sorgente qui sotto rappresentato.

```
\begin{figure}[!tb]
     \centering%
     \mbox{%
     \begin{minipage}{.30\textwidth}%
```

```
    \includegraphics[width=\textwidth]{eps/file1.eps}%
    \end{minipage}%
    \quad%
    \begin{minipage}[c]{.35\textwidth}%
    \caption*{{\footnotesize Esempio di didascalia}}
    \end{minipage}%
    \quad%
    \begin{minipage}{.30\textwidth}%
    \includegraphics[width=\textwidth]{eps/file2.eps}
    \end{minipage}%
  }%
\end{figure}
```

Le immagini relative sono in figura 11.6. In questi casi il ridimensionamento delle immagini è affidato all'istruzione `{n\textwidth}`, ove [n] va sostituito con un fattore numerico opportuno. Se ad esempio si usa un fattore (fattore .20) si riducono le dimensioni originarie delle due foto, mentre il fattore `{.35\textwidth}` determina le dimensioni d'ampiezza dello spazio riservato alla didascalia. [15]

Figura 11.15: Esempio di didascalia

float

Questo package in combinazione con il package font permette delle didascali con i font modificati rispetto a quelli in uso. `font`

Figura 11.16: Didascalia con float e font

Si tratta di una serie d'istruzioni che personalmente non ritengo particoalrmente utili, mentre per altri versi float può senz'altro essere sfruttato.

La serie di istruzioni è la seguente:

15. In un ottimo manuale reperibile in rete Caucci & Spadaccini [6, IV, pagg. 33-37] illustrano le potenzialità del package caption2 fornendo peraltro un sorgente che mostra un'abbondanza di esempi per le varie modifiche apportabili alle didascalie.

```
\usepackage{graphicx,float}
\usepackage[font={sf,bf},textfont=md]{caption}
\begin{figure*}[t]
\centering{%
\includegraphics{immagine.eps}   \caption*{Didascalia immagine}   }
\end{figure*}
```

hvfloat

hvfloat di HERBERT VOSS assolve a varie funzioni, sia a quelle tipiche di un ambiente landscape sia alla capacità di poter disporre le didascalie delle immagini in maniera del tutta diversa dal consueto, ossia non solo sotto l'immagine, ma anche a lato della stessa, rovescita,...

In questo caso quindi l'Autore ha riservata una particolare attenzione non tanto alle figure, che pure sono suscettibili di notevoli trattamenti, bensì alle didascalie che possono, in un certo modo, anche se il termine è improprio, *flottare* accanto all'immagine in diversa posizione.

Le opzioni del package sono mostrate nella tabella 11.5.

Opzione	default	Descrizione
`floatPos`	`htb`	Posizionamento dell'oggetto
`rotAngle`	0	Valore angolare *identico* l'immagine e la didascalia
`capWidth`	0.8	Larghezza discalia: `w`(larghezza), `h` (altezza). Scalabile anche con `\columnwidth`
`capAngle`	0	Valore angolare (antiorario) per ruotare la didascalia
`capPos`	`b`	Posizione didascalia relativa all'immagine: *left bottom top right*
`capVpos`	`c`	Se `capPos=l|r` la didascalia può essere posizionata soltanto `b|c|t`
`objectPos`	`c`	Posizionamento dell'immagine nei valori `l|c|r`
`objectAmgle`	0	Valore angolare (antiorario) di rotazione dell'immagine
`floatCapSep`	5	Spazio aggiuntivo (destra o sinistra) (espresso in punti)fra immagine e didascalia
`useOBox`	`false`	Non *passa* all'immagine i parametri di `hvFloat`
`nonFloat`	`false`	L'immagine non è trattata in ambiente flottante e posizionata in maniera standard

Tabella 11.5: Ozioni di hvfloat

In quest'ambiente il posizionamento (d'immagine e didascalia) si effettua con quest'istruzioni: `\hvFloat{figure}{\includegraphics{eps/immagine.eps}}{Didascalia immagine}{fig:0}`, ma è chiaro che non questa la massima *ambizione* del package di H. Voß.

In figura 11.17 è mostrato un esempio di quest'applicazione. Credo che questo sia più che sufficiente per mostrare le potenzialità del package; eventuali altre possono essere trovate nel manuale di Voß. Il sorgente qui di seguito mostrato:

```
\hvFloat[%
floatPos=htb,%
capWidth=0.5,%
capPos=r,%
capVPos=c,%
objectPos=c]{figure}{\includegraphics{eps/barca03}}%
[Didascalia posizionata con \textsf{hvfloat}]{%
Testo di prova in didascalia posizionato con \textsf{hvfloat} -- Testo di prova.....} {fig:1}
```

mostra l'applicazione di alcune opzioni indicate in tabella 11.5 assieme all'indicazione di una serie di valori di dimensionamento ed allineamento.

Figura 11.17: Testo di prova in didascalia posizionato con hvfloat – Testo di prova in didascalia posizionato con hvfloat – Testo di prova in didascalia posizionato con hvfloat – Testo di prova in didascalia posizionato con hvfloat – Testo di prova in didascalia posizionato con hvfloat – Testo di prova in didascalia posizionato con hvfloat

Rilevante il posizionamento in ambiente d'immagini in maniera simile a lscape: Relative istruzioni al manuale citato: [19, IV, pagg. 10-12].

Figura 11.18: 29 luglio 2002: Il varo del Nikao III dopo due anni di restauro

```
\begin{landscape}\begin{figure} \includegraphics[width=6cm]{eps/parte03/varo.eps}
\caption{.....} \label{.....} \end{figure} \end{landscape}
```

11.7 Brevi cenni sul trattamento del colore in LaTeX

Accenni sul colore s'erano fatti trattando del package ulem, a pagina 103 e nel capitolo dedicato alle tabelle, a pagina 182.

Qui appresso affronterò molto superficialmente, i modelli di colore, e rinvio pertanto a trattazioni appropriate. [16] I modelli di colore presenti in LaTeX sono:

— RGB, espressione di tre colori fondamentali Red, Green e Blue. Questa combinazione rappresenta lo standard in informatica. La loro combinazione genera il bianco. È il sistema usato per la visualizzazione a monitor.
- GRAY (grigio), una combinazione ridotta espressamente pensata per le tonalità di grigio, del modello RGB;
- CMYK, che raccoglie i colori Cyan, Magenta, Yellow e Black. Contrariamente al sistema RGB è un modello sottrattivo. È il sistema usato nell'editoria.

Il colore è stato trattato in LaTeX prima con il package color, quindi da xcolor che meglio risponde alle esigenze di integrazione con i pacchetti della serie ps-tricks.

I packages presentano le seguenti opzioni:
- `monochrome` che converte tutti i comandi di colore in bianco e nero;
- `dvipsnames` rende i nomi dei colori disponibili agli altri drivers;
- `nodvipsnames` opzione inversa alla precedente;
- `usenames` carica i nomi dei colori come definiti.

Trattando il colore la prima operazione da fare è quello di definirlo, e quest'operazione può effettuarsi in uno dei seguenti modi qui appresso mostrati:

```
\definecolor{gray3}{gray}{.3}
\definecolor{verdeguit}{rgb}{0,0.6,0}
\definecolor{giallo}{rgb}{0.9,0.9,0.9}
\definecolor{rossovivo}{rgb}{0.7,0.2,0.2}
```

Come si nota negli scarni sorgenti mostrati mostrati il primo argomento fra parentesi graffe è quello che assegna il nome al colore; il secondo individua le specifiche del colore (toni di grigio o colori) ed il terzo, il più delicato, ne fissa le tonalità.

Nell'esempio riportato alla seconda linea è stata riportata, di pari passo, la definizione del colore usata per creare il logo e la scritta GUIT. Si noti come in questo caso, come anche nella quarta riga degli esempi, i nomi possano essere di *fantasia*, quello che rileva è la codifica `rgb` ed il valore di colore assegnato. `rgb`

Il comando `\color` può ancora essere combinato con altre istruzioni. `\color`

Così, `\pagecolor{verdeguit}` — una volta che sia stato definito `verdeguit`;— porrà in questo colore la pagina in cui è presente il comando e quelle seguenti; l'istruzione `\normalcolor`, che agisce come l'istruzione `\normalsize` per il trattamento dei caratteri, restituirà i colori originali quando questi, però!, siano stati definiti nel preambolo del documento. Parimenti agiranno allo stesso modo `\textcolor{verdeguit}`, `\colorbox{verdeguit}`, `\colorbox{verdeguit}`,...

Tutto questo è valido naturalmente se non si usano istruzioni di colore *assolute* del tipo: `{\fcolorbox{green}{red}{\textcolor{yellow}{Riquadro}\textcolor{cyan}{a più colori} di \textcolor{blue}{\LaTeX}}}`, come quella vista ad esempio poche pagine indietro.

Non si dimentichi comunque un accorgimento pratico di primissima importanza: in poche materie come questa la teoria è una cosa e la pratica tutt'altra!

Con questo intendo dire che nel fare le prove di colore bisogna avere a portata di mano una stampante laser (a colori) di primissima qualità, perché non ci si può fidare dei colori che appaiono sullo schermo a seguito delle istruzioni impartite, in quanto questi risentono di un procedimento di trattamento del colore che no viene tradotto in rapporto 1:1 in fase di stampa.

16. È questa una parte che in futuro mi riservo di approfondire.

Valori per GRAY		Valori per RGB		Valori per CMYK	
0		0,0,0		0,0,0,0	bianco
.05		.05,.05,.05		.05,.05,.05,05	
.1		.1,.1,.1		.1,.1,.1,	
.2		.2,.2,.2		.2,.2,.2,0	
.3		.3,.3,.3		.3,.3,.3,0	
.4		.4,.4,.4		.4,.4,.4,0	
.5		.5,.5,.5		.5,.5,.5,0	
.6		.6,.6,.6		.6,.6,.6,0	
.7		.7,.7,.7		.7,.7,.7,0	
.8		.8,.8,.8		.8,.8,.8,0	
.9		.9,.9,.9		.9,.9,.9,0	
.95		.95,.95,.95		.95,.95,.95,0	
1	bianco	1,1,1	bianco	1,1,1,0	
				1,1,1,1	

Tabella 11.6: Valori per GRAY, RGB, CMYK

Nella tabella 11.6 sono riportati i valori numerici per le tonalità di grigio nei modelli Gray, RGB e CMYK. [17] Una completa mappatura del colore si può reperire nell'opera *Chroma: a reference book of LaTeX color*, di UWE KERN.

17. La tabella è stata tratta dal manuale di LUIGI CARUSILLO *I modelli di colore in LaTeX*, febbraio 2006, sesta revisione, reperibile in linea all'url `www.webalice.it/lgcrsll/pgl/index.html` L'opera fa parte di una collezione di guide intitolate dall'Autore *PGL – Piccole Guide Linux*

Capitolo 12

I box

Introduzione

I comandi della famiglia `\box` permettono di ottenere effetti grafici di evidenziazione dell'oggetto ricompreso nel box, che può essere incorniciato o meno, e racchiudendo testo o immagine in una scatola che può essere posizionata a piacere: appunto con il termine di *scatola*, l'equivalente italiano della parola inglese, sono tradotti nella nostra lingua i comandi che ora andremo a trattare.

Più precisamente, i box sono — in genere — oggetti di forma rettangolare o quadrata, di cui è possibile impostare altezza, larghezza e profondità e definire elementi opzionali. Spesso i box risultano legati per il loro posizionamento da stretta interconnessione con la `minipage` (*vedi* a pagina 292) e ricorrono anche assai spesso, all'istruzione `\rule`, già vista a pagina 105, e su questa *si appoggiano* per arricchire di nuovi elementi la loro presentazione. `\rule`

I box si presentano in conclusione come una singola unità di elaborazione, e ciò significa, come accennato di passaggio ad inizio di questa parte, che posseggono lo stesso valore del testo, perché come questo si possono liberamente muovere, avendo il medesimo peso di un carattere.

L'utilizzo di questi strumenti è diffusissimo nell'editoria e DONALD KNUTH li concepì sin dalla prima versione di TEX, ed ad essi in questi *Appunti* si è spesso ricorso per produrre esempi. Le principali istruzioni sinora adottate sono state:

`\makebox{testo}`	`\framebox{testo}`	`\fbox{testo}`	`\boxed{$a + b = c$}`
testo	testo	testo	$a + b = c$

A queste istruzioni se ne aggiungono altre sia tipiche delle famiglie dei box nel LATEX base, sia integrate con altri comandi, sia ancora di packages dedicati che permettono particolari costruzioni. Così l'istruzione `\hbox{\sffamily testo}`, [1] usata singolarmente, costruisce un box senza riquadro: testo, e la stessa integrata con `\kern` (`\hbox{\kern 2em{testo}}`) sposta lateralmente il testo fra parentesi graffe secondo la *quantità* indicata producendo testo. `\hbox`

`mbox` inscatola anch'esso il testo, ma questo viene posizionato sulla linea come se si trattasse di una continuazione di un discorso. Bisogna prestare attenzione alla lunghezza della frase perché altrimenti questa sfora la riga non essendo previsti parametri opzionali di calibrazione. Tuttavia il comando può risultare utilissimo in ambito matematico per inscatolare e far apparire sulla linea espressioni matematiche che tendono a *sfuggire* dall'impostazione desiderata. `\mbox`

Un'altra istruzione `\raisebox` permette d'innalzare ed abbastanza il testo fra parentesi graffe (l'argomento) come in questo sorgente: `\raisebox`

1. Negli esempi a seguire le modiche della famiglia di fonts ottenute con `\sffamily` non compariranno nei sorgenti.

```
{\sffamily Imparare \raisebox{3mm}{\LaTeX} \raisebox{5mm}{è apprendere}
\raisebox{7mm}{con un poco} \raisebox{9mm}{di fatica} \raisebox{8mm}{nuo-}\raisebox{7mm}{ve}
\raisebox{6mm}{for-}\raisebox{3mm}{me} \raisebox{1mm}{di} conoscenza}
```

che rende

{\sffamily Imparare \raisebox{3mm}{\LaTeX} \raisebox{5mm}{è apprendere} \raisebox{7mm}{con un poco} \raisebox{9mm}{di fatica} \raisebox{8mm}{nuo-}\raisebox{7mm}{ve} \raisebox{6mm}{for-}\raisebox{3mm}{me} \raisebox{1mm}{di} conoscenza}

dove l'innalzamento e l'abbassamento del testo è governato da `\raisebox` rispetto alla linea delimitata dalle parole "Imparare" e "conoscenza".

`\usebox` `\tbox` Alcune di queste istruzioni, ed altre ancora come `\usebox` e `\tbox`, saranno esaminate assai spesso in combinazione con altre, anche perché una delle caratteristiche dei box è la potenzialità di poter mischiare fra loro vari comandi in modo da ottenere oggetti di un particolare risalto che evidenzino il contenuto.

12.1 Famiglie dei comandi box nel LaTeX base

Volendo operare allora, al tempo stesso, un riepilogo delle istruzioni già accennate ed anche una trattazione più organica dell'argomento, riepiloghiamo per prima cosa i comandi che la famiglia dei box ricomprende nel LaTeX standard; questi sono: a) `\parbox`; b) `\fbox`; c) `\framebox`; d) `\hbox`; e) `\mbox`; f) `\makebox`, mentre altre famiglie come quelle delle istruzioni `\shadowbox`, `\doublebox`, `\ovalbox`, `\Ovalbox`, sono proprie di applicazioni diverse.

Le sezioni qui a seguire tratterano dei vari comandi indistintamente, essendo sufficiente quanto appena detto per l'individuazione dei comandi box del LaTeX base e di quelli di altri applicativi.

I singoli comandi

`\fbox` L'istruzione `\fbox` è stata già esaminata in numerosi sorgenti d'esempio, specie al capitolo precedente, e come ormai s'è appreso, permette d'inserire testo, tabelle ed immagini in un riquadro. L'espressione `\fbox{testo in riquadro}` genera ad esempio: \fbox{testo in riquadro} .

`\fbox`, come altri comandi della medesima famiglia, può essere implementato con istruzioni tipo: `{\setlength{\fboxrule}{1mm}\setlength{\fboxsep}{3mm}\fbox{\textsf{testo in box}}}` che generano {\setlength{\fboxrule}{1mm}\setlength{\fboxsep}{3mm}\fbox{\textsf{testo in box}}} disegnando attorno al testo una cornice spaziata della misura precisata.

Per fare un ulteriore esempio, il sorgente `{\setlength{\fboxrule}{2pt}{\setlength{\fboxsep}{.3mm}\fbox{\textsf{testo in box}}}}` produce {\setlength{\fboxrule}{2pt}{\setlength{\fboxsep}{.3mm}\fbox{\textsf{testo in box}}}} .

Si presti attenzione che le varie istruzioni sono racchiuse agli estremi fra due parentesi graffe, perché si tratta di comandi a lunga valenza che altrimenti estenderebbero la loro influenza su tutti gli altri `\fbox` seguenti nel documento.

Rotazione dei box

La rotazione dei box può avvenire principalmente in due maniere: si può costruire il box e poi ruotarlo con strumenti finalizzati, oppure si può ricorrere alle istruzioni caratteristiche dell'istruzione `\rotatebox` e farlo ruotare attorno al proprio punto di rotaione: *vedi* figura 12.1.

La prima via si segue ricorrendo all'istruzione `\turn` del package rotating. Qui appresso è mostrato un esempio di quest'applicazione vista al capitolo precedente e ricorrendo alle istruzioni `\parbox` `\turn` (che vedremo meglio a breve) e `\fbox`. Si ottiene questa costruzione:

testo testo
testo testo

testo testo
testo testo

testo testo
testo testo

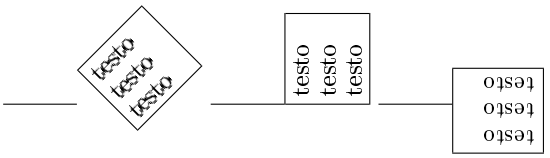
\parbox

fornita da questo sorgente:

```
\fbox{\parbox[b]{15mm}{testo testo testo testo}}
\hspace{10mm}\begin{turn}{-45} \fbox{\parbox[b]{15mm}{testo testo testo testo}}
\end{turn} \hspace{10mm}\begin{turn}{-90}
\fbox{\parbox[b]{15mm}{testo testo testo testo}} \end{turn}
```

L'altra via è data, come si diceva, dall'istruzione `rotatebox`. Se la rotazione di un box è espressa da due soli argomenti: l'angolo di rotazione ed il materiale (il box stesso) in rotazione, esistono però le opzioni di posiionamento. Le immagini di seguito riportate, dove il comando `\rotatebox` è stato ancora una volta implementato con `\fbox` e `\parbox`, mostrano la rotazione del box rispetto alla linea di base espressa da `\rule{3em}{0.4pt}`.

\rotatebox




```
\rule{3em}{0.4pt}\rotatebox{45}{%
\fbox{\parbox{3em}{testo\\testo\\testo}}}
\rule{3em}{0.4pt}\rotatebox{90}{%
\fbox{\parbox{3em}{testo\\testo\\testo}}}
\rule{3em}{0.4pt}\rotatebox{180}{%
\fbox{\parbox{3em}{testo\\testo\\testo}}}
```

Per comprendere a pieno ora il meccanismo di posizionamento in funzione della rotazione, riprendiamo gli stessi sorgenti appena riportati, ed assegnamo a ciascuno di essi un angolo di rotazione di 45 gradi. Quindi implementiamo l'istruzione con le opzioni ammesse `[tbcB]`, in modo da ottenere questi sorgenti:

```
\rule{3em}{0.4pt}\rotatebox{45}{%
\fbox{\parbox[c]{3em}{testo\\testo\\testo}}}\hspace{2mm}
\rule{3em}{0.4pt}\rotatebox{45}{%
\fbox{\parbox[t]{3em}{testo\\testo\\testo}}}\hspace{2mm}
\rule{3em}{0.4pt}\rotatebox{45}{%
\fbox{\parbox[b]{3em}{testo\\testo\\testo}}}\hspace{2mm}
\rule{3em}{0.4pt}\rotatebox{45}{%
\fbox{\parbox[B]{3em}{testo\\testo\\testo}}}
```

ed osserviamo la costruzione che ne deriva. Tenendo presente la raffigurazione geometrica del box comprensiva dei suoi elementi basilari come riportata in figura 12.1, si avrà questa costruzione,

dove si nota il diverso comportamento in altezza del box, rispetto alla linea di base, a seconda dell'opzione specificata fra parentesi quadre.

Qualora questi parametri non dovessero ancora ritenersi soddisfacenti, si può ricorrere alla specifica delle coordinate di origine di posizionamento del box secondo questi sorgenti il cui output va sempre guardato tenendo presente la costruzione grafica in figura 12.1:

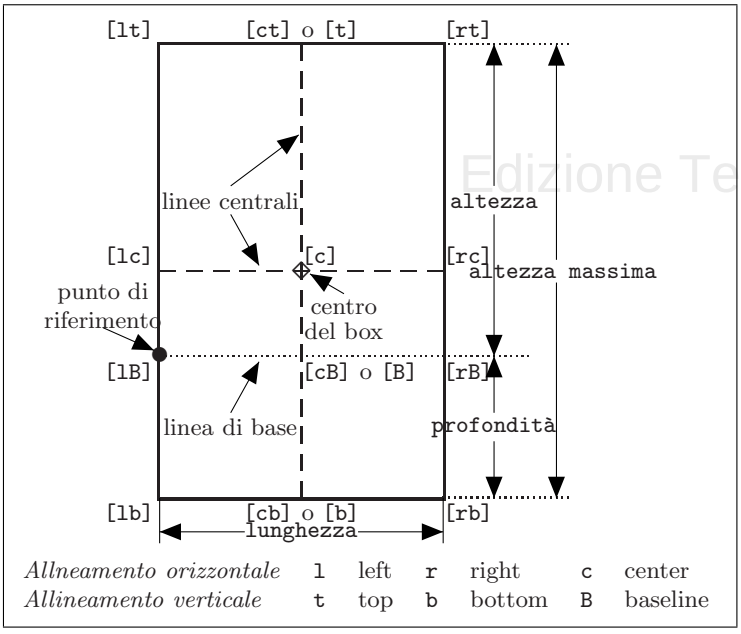
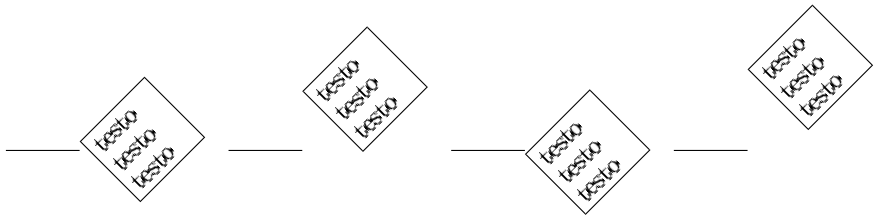


Figura 12.1: Costruzione geometrica di un box in LaTeX

```
\rule{3em}{0.4pt}\rotatebox[x=25mm,y=49mm]{45}{%
\fbox{\parbox{3em}{testo\\testo\\testo}}}\hspace{2mm}
\rule{3em}{0.4pt}\rotatebox[x=-25mm,y=-49mm]{45}{%
\fbox{\parbox{3em}{testo\\testo\\testo}}}\hspace{2mm}
\rule{3em}{0.4pt}\rotatebox[x=5mm,y=-5mm]{45}{%
\fbox{\parbox{3em}{testo\\testo\\testo}}}\hspace{2mm}
\rule{3em}{0.4pt}\rotatebox[x=-15mm,y=-15mm]{45}{%
\fbox{\parbox{3em}{testo\\testo\\testo}}}
```

che rendono:



`\fbox` e `\parbox`

Con `\fbox` e `\parbox`, che si vedrà ancora più approfonditamente a breve, è possibile inserire uno o più paragrafi di testo all'interno di un doppio riquadro:
`\fbox{\fbox{\parbox{.95\linewidth} {Nonostante il freddo.... }` che generano:

Edizione Test - Agosto 2008

Opzioni di	
`pos`	lettere `t`, `b`, `c`: indicano l'allineamento verticale del testo (*center*, *top*, *bottom*);
`height`	indica l'altezza del paragrafo box espressa nella unità di misura che si desidera;
`innerpos`	le lettere `t`, `b`, `c`: indicano l'allineamento verticale del testo contenuto;
`width`	indica la larghezza del paragrafo;
`text`	è il testo contenuto

Tabella 12.1: Opzioni specifiche dei comandi `\parbox`. Vedi anche la figura 12.1.

Nonostante il freddo continuai a camminare finché le forze tennero. Il bujo lentamente mi avvolse ed in breve non riuscii più a scorgere né un cammino da percorrere né eventuali ostacoli da evitare. Non si vedeva la Luna, doveva essere già tramontata, e le stelle non mi dicevano nulla né come indicazione fisica, né, tantomeno, spirituale.

`\framebox` permette di distribuire il testo all'interno di un riquadro secondo le varianti qui indicate: `\framebox`

`\par\framebox {testo in riquadro }` genera:

testo in riquadro

`\par\framebox[3.52\width ][s]{testo in riquadro }` genera:

testo in riquadro

`\par\framebox[2\width]{testo in riquadro }` genera:

testo in riquadro

L'istruzione `\parbox`, diverse volte usata nella II parte per la creazione di esempi risulta ancora utile per l'evidenziazione di paragrafi o di parti di testo `\parbox`

come in questo caso, dove è stata assegnata l'istruzione `\parbox{4.4cm}{testo}` prima delle parole "come in questo caso, dove..."

L'istruzione permette di distribuire due paragrafi in maniera affiancata sulla stessa pagina tramite la determinazione di parametri.

L'istruzione completa del comando è: `\parbox[pos][height][innerpos]{width}{text}`.

Aber Tonio war Konsul Krögers Sohn, dessen Getreidesäcke mit dem breiten schwarzen Firmendruck man Taf für Tag durch die Straßen kutschieren sah;...

Tonio era il figlio del console Kröger, i cui sacchi di grano con il grosso timbro nero della ditta, giorno dopo giorno venivano trasportati per le strade;...

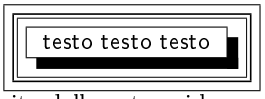
E questo è il sorgente dell'esempio mostrato:

```
\parbox[t][2cm][t]{6.2cm}{\em Aber Tonio war Konsul Kr\"{o}gers Sohn.... }
\hspace{4mm}
\parbox[t][2cm][t]{6.2cm}{\em Tonio era il figlio del console Kr\"oger.... }
```

L'istruzione `\parbox` può ancora essere usata in questa forma implementata dell'istruzione `\tbox` `\usebox`

```
\parbox{.45\textwidth}{ \usebox{\tbox} }%                                    \tbox
\parbox{.3\textwidth}{\fbox{\doublebox{\shadowbox{ testo testo testo}}}}
```

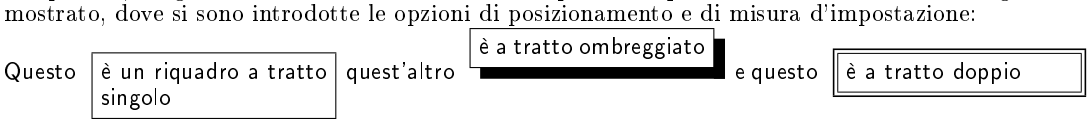
che rende:



Un'applicazione di \parbox si vedrà ancora a proposito delle note nei box, a fronte.

\doublebox
\shadowbox

\parbox, congiunta a \doublebox e \shadowbox permette queste costruzioni secondo il sorgente mostrato, dove si sono introdotte le opzioni di posizionamento e di misura d'impostazione:




```
Questo \fbox{\parbox[t]{3cm}{è un riquadro a tratto singolo}}
quest'altro \shadowbox{\parbox[b]{3cm}{è a tratto ombreggiato}}
e questo \doublebox{\parbox[c]{3cm}{è a tratto doppio}}
```

Si guardi l'effetto prodotto dalle opzioni [tbc] che innalzano, abbassano e centrano (rispetto alla linea) i singoli box.

Box dinamici

Box dinamici si possono ottenere tramite una serie di istruzioni da impartire nel preambolo e nel documento come mostrato nel riquadro presente in questa pagina.

SCRIPT PER I BOX DINAMICI DA INSERIRE NEL PREAMBOLO

```
\makeatletter
\newenvironment{Annotation}[1]{%
\def\AnnotBoxName{Annotation#1}%
\@ifundefined{\AnnotBoxName}{%
\expandafter\newsavebox\csname\AnnotBoxName\endcsname  }{}%
\expandafter\let\expandafter\AnnotBox\csname\AnnotBoxName\endcsname
\begin{lrbox}{\AnnotBox}%
\begin{minipage}{\linewidth}%
}{% \end{minipage}%
\end{lrbox}%
\global\setbox\AnnotBox=\copy\AnnotBox }  \makeatother
```

SCRIPT DA INSERIRE NEL DOCUMENTO

```
\begin{Annotation}{bar}
\textit{Questa è una annotazione \\redatta a titolo di esempio \\e creata con le istruzioni sopra
riportate.} \end{Annotation} \usebox{\Annotationbar} \usebox{\Annotationbar}
```

che genera [a]

Questa è una annotazione
redatta a titolo di esempio
e creata con le istruzioni sopra riportate.

a. I sorgenti sono tratti dal sito http://www.texnik.de

Riquadri colorati e ombreggiati

Istruzioni che permettono di ottenere testi incorniciati con evidenziazione del colore:

Le istruzioni richiedono che sia stato richiamato nel preambolo il package color: in genere non è necessario richiamare nel preambolo; tuttavia, a seconda delle varie versioni di LaTeX installate può richiedersi l'istruzione \usepackage{color} o del più avanzato xcolor.

Alcuni anni fa, non mi ricordo quanti precisamente, e disponendo di poco, o quasi niente, denaro, non avendo alcun particolare interesse per le cose di questo mondo, mi risolsi a navigare per vedere la parte della terra ricoperta dalle acque: è questo un modo che ho per riprendere il controllo su me stesso, per cacciare il malumore e ristabilire l'equilibrio. [1]
Come mi accorgo che il volto va assumendo un ghigno torvo, [2] che la mia anima assomiglia tutta ad un umido e piovigginoso novembre, come mi sorprendo a sostare, quasi senza accorgermene, dinanzi alle agenzie di pompe funebri e di andare appresso a tutti i funerali che incontro; e specialmente quando l'ipocondria aumenta così tanto che mi deve soccorrere un forte principio morale per impedirmi di scendere risoluto in strada e gettare con rabbia per terra il cappello alla gente [3], allora mi accorgo che è tempo di riprendere il mare al più presto. In questo modo aggiro il suicido.

a. Traduzione letterale: *Ogni qualvolta io mi trovo (scopro) un (atteggiamento) torvo sulla bocca.*
b. Reso *methodically* con *rabbia*: non ha senso dire: "buttare giù metodicamente il cappello alla gente".
c. Pistola e palla suonano ridicoli in italiano. Il testo inglese è: *This is my substitute for pistol and ball.*

L'inserimento del testo in un riquadro *elementare* segue queste istruzioni:
`{\large \textsf{\fcolorbox{green} {red} {\textcolor{yellow} {Riquadro} \textcolor{cyan}{a più colori} di \textcolor{blue}{\LaTeX }}} }` che rendono:

Riquadro a più colori di LaTeX .

Un ulteriore esempio: L'output LaTeX 2ε è prodotto da questo sorgente:

```
\setlength\fboxrule{1pt}\setlength\fboxsep{1mm}
\definecolor{giallo}{rgb}{0.9,0.9,0.9}
\fcolorbox{giallo}{yellow}
{\textcolor{red}{ \LaTeXe }}
```

Note nei box

L'inserimento di note nei box avviene tramite le istruzioni qui sotto riportate che forniscono l'output nel riquadro presente in questa pagina, la cui ampiezza del testo è dovuta al fatto che si sono volute posizionare più note:

```
\framebox[130mm]{\parbox{120mm}{
Alcuni anni fa .....  e ristabilire l'equilibrio.~\footnotemark[1]
Come mi accorgo che il volto va assumendo un ghigno torvo, \footnotemark[2] ........
......il cappello della gente\footnotemark[3], allora mi accorgo che
è tempo di riprendere il mare al più presto. In questo modo aggiro il suicido.} }
\addtocounter{footnote}{-3}
  \footnotetext[1]{La traduzione letterale ....}   \stepcounter{footnote}
  \footnotetext[2]{Ho reso \emph{methodically} ....}
  \stepcounter{footnote}   \footnotetext[3]{Pistola e palla suonano ridicoli in italiano. }
```

12.2 Packages dedicati

Esamineremo adesso alcuni packages che arricchiscono le funzioni dei box del LaTeX standard.
Uno di questi, slashbox è stato già mostrato in una particolare applicazione a pagina 188, e lì si fa rinvio.

boxedminipage

boxedminipage, un package abbastanza datato, residuo dei primi pacchetti per LaTeX2.09, risale infatti al 1992, opera di MARIO WOLCZKO, che consente di ridimensionare la pagina ed inserire il testo in una sorta di minipage costruita in accordo alle istruzione del package. Questo sorgente

boxedminipage

```
{ \hsize = 4 in     \leftskip = 1 in      \rightskip = 1 in
\begin{boxedminipage}[l]{0.70\linewidth} \hsize = 4 in
La terra era ormai divenuta...
\end{boxedminipage}    \par           }
```

produce:

La terra era ormai divenuta una striscia piccola piccola e non s'udivano altri rumori tranne il sordo brontolio del mare e quello del vento.

framed

L'istruzione tipica del package framed che si deve a DONALD ARSENEAU, consente di far apparire un paragrafo incorniciato in maniera ancora più marcata disegnando due nuovi ambienti framed e shared. La sequenza di comandi:

```
\setlength{\FrameRule}{3pt}
\begin{framed} questo è il primo paragrafo composto in ambiente framed \end{framed}
\definecolor{shadecolor}{rgb}   {0.4,0.7,1}
\begin{shaded} e questo è un paragrafo composto in ambiente shared \end{shaded}
```

genera:

questo un paragrafo composto in ambiente framed

e questo è un paragrafo composto in ambiente shared

shadow

shadow, di MAURO ORLANDINI definisce il comando `\shabox` che genera un effetto ombra più marcato rispetto all'istruzione di base `shadowbox`.

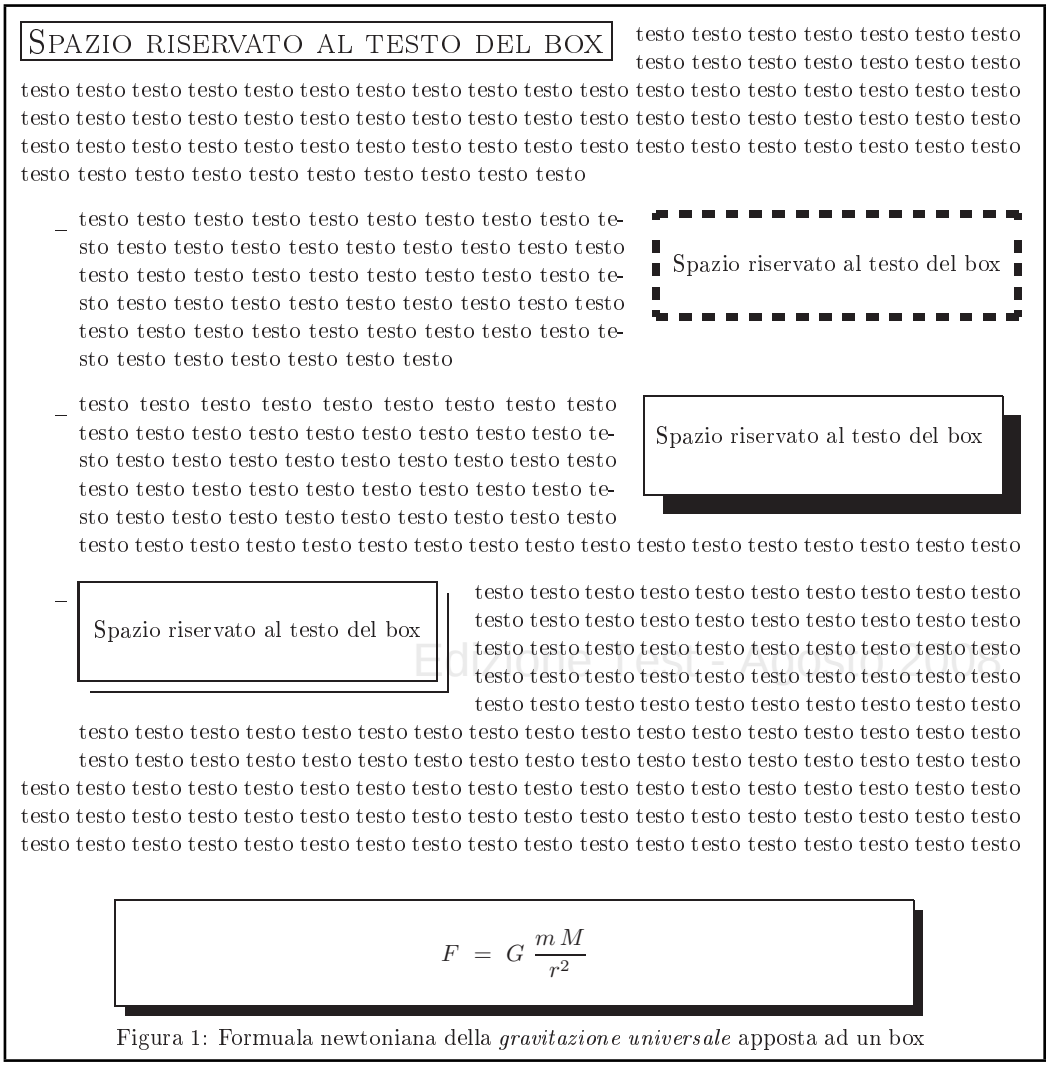
Il sorgente `\shabox{\LaTeXe\ombreggiato}` genera

LaTeX 2_ε ombreggiato

picins

Questo package, già visto a pagina 311 per le sue caratteristiche di posizionare le immagini in un riquadro a destra o sinistro dello schermo, permette le stesse operazioni anche sui box, dimostrando anche in quest'ambiente grande versatilità.

L'istruzione principale resta la stessa (`\parpic`), e resta pure la stessa impostazione di valori fra parentesi tonde come eguali sono le opzioni di posizionamento dell'oggetto-box. Le istruzioni sono riportate qui appresso, ed in figura 12.2 alcuni output prodotti.



SPAZIO RISERVATO AL TESTO DEL BOX

testo testo

- testo

  Spazio riservato al testo del box

- testo

  Spazio riservato al testo del box

- Spazio riservato al testo del box

  testo
testo testo

$$F \;=\; G\,\frac{m\,M}{r^2}$$

Figura 1: Formuala newtoniana della *gravitazione universale* apposta ad un box

Figura 12.2: Applicazioni picins

```
\usepackage{graphicx,picins}
\parpic{\fbox{\Large\scshape Spazio riservato al testo del box}}
\noindent testo testo testo......
\begin{itemize}
\item [--] \dashlength{2mm} \linethickness{1mm}
\parpic(54mm,15mm)[dr]{Spazio riservato al testo del box}
\noindent testo testo testo......
```

```
\item [--]  \shadowthickness{3mm}  \linethickness{.4pt}
\parpic(54mm,15mm)(2mm,7mm)[sr]{Spazio riservato al testo del box}
\noindent testo testo testo......
\item [--] \boxlength{2mm}\parpic(54mm,15mm)[x]{Spazio riservato al testo del box}
\noindent testo testo testo......
\end{itemize} \noindent testo testo testo......
\centering%
\pichskip{6mm}
\piccaption{Gravitazione universale}
\parpic(120mm,16mm)[s]{\(\displaystyle F = \frac{m M}{r^2} \)}
```

\piccaption

Una delle applicazioni può rilevanti di questo package, consiste comunque nella capacità di poter applicare le didascalie ai box ricorrendo all'istruzione `\piccaption`.

L'inserimento della caption segue l'istruzione `\piccaption{Didascalia}`, mentre il box continua a ad essere individuato dai soliti paramatri come quelli all'ultima riga del sorgente in questione.

Le dimensioni del box (in lunghezza) influiscono naturalmente sulle dimensioni della `\caption`.

fancybox

fancybox è un pacchetto sviluppato nel 1997 da TIMOTHY VAN ZANDT, e rivisto nel 2000 da HEIKO OBERDIEK. Van Zandt lo sviluppò assai probabilmente allorché lavorava alla classe seminar per la generazione di slides: *vedi* in proposito la sezione 17.1. Il package di Van Zandt ridefinisce nuovi ambienti assai rilevanti per le famiglie dei box.

Allineamento del testo nel box

Questi riguardano innanzitutto la possibilità di posizionare i box secondo le opzioni `t` (*top*) e `b` (*bottom*), specificando le opzioni nei singoli ambienti che il package ridefinisce, e che sono Bcenter, Bflushleft e Bflushright. Tuttavia esistono alcuni limitazioni e contrasti che appresso vedremo, a ragione soprattutto della databilità del pacchetto.

I singoli comandi non spostano il box al centro, destra o sinistra, ma hanno influenza sul testo ricompreso all'interno del box, come mostrato dal sorgente a seguire:

```
\setlength{\fboxsep}{10pt}%
\fbox{%
\begin{minipage}{33mm}  \begin{Bflushright}
testo in box testo \\in box testo in box testo \\in box
\end{Bflushright} \end{minipage} }
```

testo in box testo
in box testo in box testo
in box

Il package presenta quindi la possibilità di comporre ambienti itemizzati nei box con l'ambiente Bitemize racchiuso fra i classici comandi `\begin{Bitemize}` ed `\end{Bitemize}`. Queste istruzioni permettono d'inserire in un box a parte le liste d'interesse collocandole dove si preferisce con le opzioni `[b]` o `[t]` che seguono l'istruzione d'apertura d'ambiente: `\begin{Bitemize}[t]`.

```
\usepackage{fancybox}
      % In alternativa \usepackage{myfancybox}
      % Vedi testo

\ovalbox{\begin{Bcenter}
Liste con \textsf{fancybox}\hspace{3mm}
\begin{Bitemize}[b] \item primo elemento
\item secondo elemento \item terzo elemento
\item quarto elemento
\end{Bitemize}  \end{Bcenter}  }
```

Liste con fancybox
- primo elemento
- secondo elemento
- terzo elemento
- quarto elemento

Si noti l'effetto dell'opzione [b] dopo l'istruzione \begin{Bitemize}, che sposta verso il basso tutta la scritta Liste con \textsf{fancybox}.

Espressioni matematiche nei box

L'introduzione di un ambiente dedicato alle applicazioni matematiche, Beqnarray, che ammette anch'esso —in via teorica— opzioni di posizionamento, permette di scrivere testo matematico all'interno di un box in maniera piu professionale che non ricorrendo ad un comando standar di queste famiglie. Beqnarray

Occorre tuttavia premettere che quest'ambiente che non dà alcun problema con le classi standard, è entrato in conflitto con le impostazioni della classe di lavoro (memoir), in quanto il foglio di stile del package, riga 325, usa un'istruzione, abbastanza datata, \rm(theequation), rifiutata da memoir in quanto obsoleta.

Per far lavorare comunque nella classe quest'ambiente, occorre pertanto agire secondo le procedure descritte a pagina 67 e riscrivere nel package l'istruzione in \textrm(theequation). Un altro limite, in questo limitato ambiente, è rappresentato dalle opzioni [t] e [b] che non producono alcun effetto salvo quello di spostare le parentesi quadre e la lettera fra esse compresa immediatamente prima dell'espressione, e questo anche per le classi standard.

Qui appresso è mostrata un'applicazione dell'ambiente con il relativo sorgente.

```
\Ovalbox{\begin{Beqnarray}
a^2 + b^2 & = & c^2\\
a^2 + 2ab + b^2 & = & (a + b)^2
\end{Beqnarray}  }
```

$$\begin{aligned} a^2 + b^2 &= c^2 && (12.1)\\ a^2 + 2ab + b^2 &= (a+b)^2 && (12.2) \end{aligned}$$

Van Zandt ha introdotto l'ulteriore ambiente landfloat in caso fosse necessario rappresentare espressioni di notevole lunghezza che dovessero essere ruotate per essere rappresnetate nella loro interezza sulla parte più lunga della pagina. landfloat

Si tratta in sostanza di qualcosa molto simile all'ambiente landscape visto al capitolo precedente e che, congiuntamente all'istruzione rotateleft ruota l'intera espressione matematica attraverso questa *routine*:

```
\begin{equation}
\begin{landfloat}{table}{\rotateleft}
 a + b + c +d + f + g + h+ i +l + m..... = z
\end{landfloat}
\end{equation
```

Di quest'ambiente non è riportato l'output per non appesantire il testo ulteriormente il tetso con una pagina vuota, ma la *routine* è veramente utile.

Per chi fosse interessato ad approfondire ulteriormente le possibilità dei box (e relativi ambienti) in ambito matematico, rinvio al manuale di Van Zandt [17, IV], dove l'autore fornisce interessanti *routines* per la creazione di nuovi ambienti.

In particolare i box

Un ulteriore ambiente introdotto è Sbox, il quale in congiunzione con l'istruzione \TheSbox mostra in *display* il box costruito nell'ambiente secondo queste istruzioni: Sbox TheSbox

```
\begin{Sbox}
\begin{minipage}{60mm}
Testo nel box in ambiente \textsf{Sbox}
\end{minipage} \end{Sbox} \fbox{\TheSbox}
```

Testo nel box in ambiente Sbox

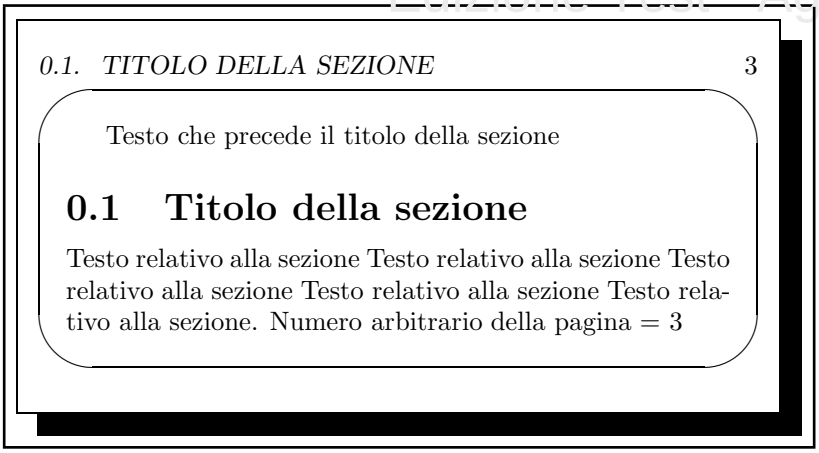


Figura 12.3: Applicazione di `\fancypage`

`\ovalbox` `\Ovalbox` `\doublebox` `\fancyoval` `\cornersize`

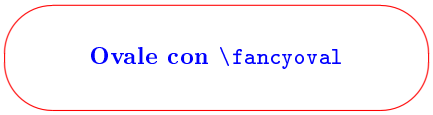
In chiusura di fancybox, le nuove istruzioni introdotte per i box. Innanzi tutto `\ovalbox` e `\Ovalbox`, di cui non presento ulteriori esempi essendo visibili le loro applicazioni nei due precedenti sorgenti, ed ancora la già vista istruzione `\doublebox` di cui : anche di questa l'output è stato già abbondantemente prodotto.

Accanto a queste, ne esistono altre istruzioni come `\fancyoval` che lavora in ambiente picture (*vedi* il capitolo 13) in congiunzione con un'altra istruzione introdotta dal package (`\cornersize`), che definisce, come da nome, le dimensioni dell'argomento ricompreso fra parentesi graffe .

Un'applicazione di queste istruzioni è la seguente:

```
\usepackage{fancybox}
\cornersize{6.7} \begin{picture}(40,40) \color{red}
  \put(45,20){\makebox(0,0){\color{blue}
  \textbf{Ovale con {\ttfamily \textbackslash fancyoval}}} }
  \put(45,20){\fancyoval(160,40)}
```

che restituisce:



`\fancypage` e `\fancyput`

`\fancypage`

Due altre istruzioni, anche se ai limiti della famiglia dei box, sono rappresentate da `\fancypage`, che per mezzo della ridefinizione di alcuni comandi da fornire nel preambolo e di altri specifici nel documento, permette un'impostazione della pagina come quella rappresentata in figura 12.3.

Nel preambolo occorre fornire istruzioni di questo tipo o similari:

```
\usepackage{fancybox}
\makeatletter
\renewcommand\section{\@startsection {section}{1}{\z@}%
       {-2.5ex \@plus -1ex \@minus -.2ex}%
       {1.3ex \@plus.2ex}%
       {\normalfont\Large\bfseries}}
\makeatother
```

e quindi nel documento comparirà quest'altra serie di istruzioni.

```
\fancypage
  {\setlength\fboxsep{10pt}\ovalbox}
  {\setlength{\fboxsep}{8pt}%
   \setlength{\shadowsize}{8pt}%
   \shadowbox}
Testo che precede il titolo della sezione \section{Titolo della sezione}
Testo relativo alla sezione...... .....Numero arbitrario della pagina = 3
```

Si nota che compaiono diverse istruzioni tipiche delle famiglie dei box, governate comunque dall'istruzione `\fancypage`. Quest'istruzione, che credo di rara applicazione nella scrittura di un documento, può invece rivelarsi utile per creare documenti d'esempio da inserire (appunto) nel documento che si sta scrivendo. I sorgenti sopra mostrati e relativi a quest'applicazione sono tratti con modifiche da *The LaTeX Companion*, [17, II, pag. 398].

Una variante all'esempio appena mostrato si ha ricorrendo ad un'altra istruzione del package `\fancyput`, nella forma `\fancyput(2in,-1.2in) {\HUGE\sffamily \textcolor{red}{TEST}}`.

`\fancyput`

Questa, come s'intuisce e verifica dal sorgente appena riprodotto, posiziona la scritta voluta nella parte del documento secondo le coordinate specificate.

L'istruzione si può rilevare utile quando una copia del docuemnto è rilasciata in versione *beta-test* e si desidera apporre quest'indicazione all'inizio di un capitolo o di ogni sezione. È, se si vuole, una sort di variante delle righe di sorgente viste a pagina 85, che appongono una scritta su tutte le pagine del documento.

`\fancyput` conosce due varianti interessanti:

- `\thisfancyput` che produce effetti soltanto sulla pagina in cui l'istruzione è apposta, e l'istruzione del tutto analoga — `\thisfancyput`
- `\fancypage}` che produce i suoi effetti, come appunto la precedente, sulla pagina. — `\fancypage`

Entrambe le istruzioni conoscono la versione asteriscata, che se usate aggiungono l'argomento di `\fancyput*` `\fancypage*` (nell'esempio di sopra la scritta "TEST") in una pagina seguente.

`\varianti asterisco`

Edizione Test - Agosto 2008

Capitolo 13

I grafici

Introduzione

Per la grafica LaTeX poggia su diversi ambienti, il basilare e standard è picture, che sovrintende alla composizione di grafici elementari: linee, cerchi, diagrammi elementari, e poco più.

Le conoscenze che servono, almeno da questo punto di vista, sono limitate, riducendosi a nozioni elementari di geometria. Più complesso è il sistema della rappresentazione con le applicazioni avanzate, ma anche con queste, una volta che ci si è *leticato* un poco sino a comprenderne il meccanismo di funzionamento, tutto procede senza problemi. Gli interessati dovranno quindi procurarsi (stampandoli e non leggendoli *on line*!) i vari manuali, studiarli, e poi procedere alla creazione di grafici. Un discorso a parte meriterebbe anche in questo caso il trattamento del colore, affontato quasi di passaggio a pagina 319. Per questo, come ho già detto in quelle pagine, mi riservo in futuro una trattazione più completa.

Ricordarsi di quanto si è detto all'inizio di questa parte circa l'output desiderato, se PDF o PS. Per molti ambienti grafici l'output è indifferente, mentre per i pacchetti avanzati della serie pst-... l'output deve necessariamente essere in formato PostScript. Esistono, è vero, e li vedremo, alcuni applicativi che aggirano quest'ostacolo, ma la via, a mio parere, non è perseguibile in vista di un ottimale output. Ho notato spesso piccoli *accomodamenti* che non sempre soddisfacevano i comandi inseriti.

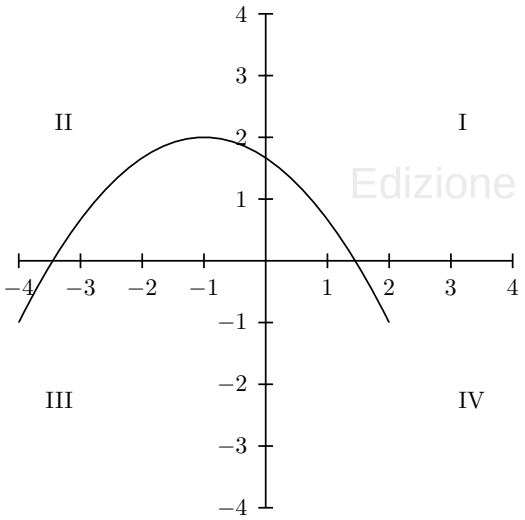
13.1 I vari ambienti

L'ambiente originario picture è nato nel LaTeX2.09, e ne supporta tutte le limitazioni.

Per sopperire alle sue limitazioni, diverse applicazioni si sono succedute nel tempo: pspicture, , pict2e, pgf, pspicture, epic ed eepic, alcuni altri e soprattutto le applicazioni della serie pstricks. In tutti questi packages, compreso l'ambiente picture, creazione e posizionamento dell'immagine-grafico avvengono secondo il sistema delle coordinate cartesiane. Il singolo applicativo legge cioè innanzi tutto le istruzioni relative al grafico da rappresentare (linea, curva,...) quindi le informazioni sulle sue caratteristiche (lunghezza, spessore del tratto,...) e poi procede a rappresentarlo secondo il sistema delle cordinate cartesiane date.

picture

Inizio e fine dell'ambiente sono marcati con i consueti comandi `\begin{picture}` ed `\end{picture}`. Le coordinate `(x, y)` comprendono, fra parentesi tonde, le espressioni nemeriche che si riferiscono



Piano cartesiano

```
%Preambolo
\documentclass{minimal}
\usepackage{pstricks,pst-plot}
%Istruzioni
\begin{pspicture}(-4.5,-4.5)(4.5,4.5)
\psaxes(0,0)(-4,-4)(4,4)
\uput[ur](3,2){I}
\uput[ul](-3,2){II}
\uput[dl](-3,-2){III}
\uput[dr](3,-2){IV}
\parabola(2,-1)(-1,2)
\end{pspicture}
```


Figura 13.1: Piano cartesiano

alle coordinate dell'immagine e dei singoli elementi.

L'unità di lunghezza va specificata una volta per tutte prima dell'inizio dell'ambiente. Come ho detto spesso, è bene scegliere non una misura assoluta, bensì una misura *legata* al font in uso, cioè, meglio definire la misura `1ex` piuttosto che `1mm`.[1] Un modo di definire questo parametro può essere il seguente: `\setlength{\unitlength}{60mm}`. il valore di default dell'unità di lunghezza è `1pt`.

Le modalità operative di picture, sostanzialmente analoghe a quelle di altri pacchetti più potenti quali pstricks, sono semplici. Agendo secondo il sistema delle coordinate cartesiane, per ogni punto del piano l'applicazione individua il punto che deve disegnare e comporre la curva o la retta a seconda dell'istruzione ricevuta, considerando in quale dei quattro quadranti il punto si trova potendo assumere, in funzione del quadrante, valori positivi o negativi per ascisse ed ordinate.

Nella figura 13.1 (disegnata però, come si nota dal sorgente, secondo l'ambiente avanzato pspicture di pstricks che sarà esaminato alla sezione 13.1) è rappresentato un piano cartesiano diviso nei quattro quadranti sul quale è disegnato un qualsiasi paraboloide. La curva passa per due punti (positivi e negativi) sull'asse delle ascisse e per due punti (positivi e negativi) dell'asse delle ordinate.

La struttura delle istruzioni all'interno del singolo ambiente assume la consegna di tradurre i comandi in una retta, in una curva, od in un'altra determinata figura a seconda delle coordinate che siano state impartite.

Comandi di picture

I principali comandi utilizzabili in questo ambiente sono:

1. Negli esempi mostrati si trova tuttavia usata l'unità di lunghezza in millimetri, perché sono disegni della precedente edizione degli *Appunti* non modificati.

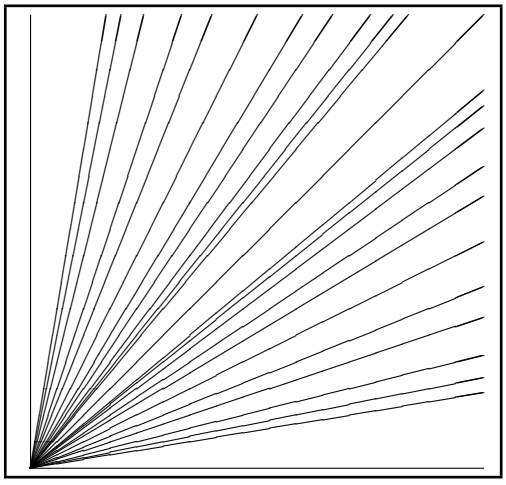
- \put: nella forma: \put(x,y){oggetto};
- \multiput: nella forma: \multiput(x,y)(Δx,Δy) {n} {oggetto}; dove l'oggetto può essere tanto l'istruzione \line(3, 6) {2.3}, o \circle*{0.75}. Le cosiddette *curve di Bézier* sono rappresentate nella forma: \qbezier(1.2, 0.6) (3.3, 1.4) (2.6, 4.5). L'ambiente picture conosce ancora le seguenti istruzioni finalizzate:
  - \line: istruzione che presenta i seguenti parametri:
    a) direzione del vettore: i valori sono compresi fra -6,-5 - 5,6;
    b) lunghezza del vettore rappresentato.
  - \circle: l'argomento è relativo all'unità di lunghezza e determina il diametro del cerchio. L'ambiente ha *forti* restrizioni perché sono ammessi diametri sino a 14 mm;
  - circle* produce cerchi riempiti;
  - \arrows: istruzione vettoriale con valori compresi fra -4,-3 - 3,3;
  - \oval: crea ovali;
  - \thinlines: crea linee sottile;
  - \thicklines: crea linee più marcate;
  - \linethickness: regola lo spessore della linea.\linethickness.

I disegni a seguire sono tratti, salvo piccole modifiche ed adattamenti, dal manuale *Graphics in LaTeX 2ε* [15, IV] e dal sito http://www.texnik.de, entrambi di URS OSWALD.

Linee

```
\setlength{\unitlength}{6cm}
\begin{picture}(1, 1)
\put(0, 0){\line(0, 1){1}}
\put(0, 0){\line(1, 0){1}}
\put(0, 0){\line(1, 1){1}}
\put(0, 0){\line(1, 2){.5}}
\put(0, 0){\line(1, 3){.33333}}
\put(0, 0){\line(1, 4){.25}}
\put(0, 0){\line(1, 5){.2}}
\put(0, 0){\line(1, 6){.16667}}
\put(0, 0){\line(2, 1){1}}
\put(0, 0){\line(2, 3){.6667}}
\put(0, 0){\line(2, 5){.4}}
\put(0, 0){\line(3, 1){1}}
\put(0, 0){\line(3, 2){1}}
\put(0, 0){\line(3, 4){.75}}
\put(0, 0){\line(3, 5){.6}}
\put(0, 0){\line(4, 1){1}}
\put(0, 0){\line(4, 3){1}}
\put(0, 0){\line(4, 5){.8}}
\put(0, 0){\line(5, 1){1}}
\put(0, 0){\line(5, 2){1}}
\put(0, 0){\line(5, 3){1}}
```

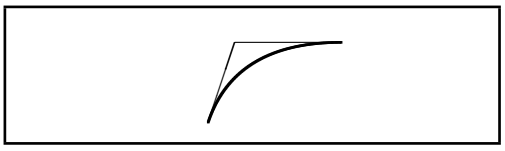
(*segue alla colonna a fianco*)



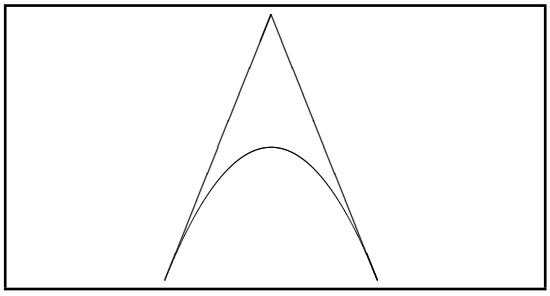
```
\put(0, 0){\line(5, 4){1}}
\put(0, 0){\line(5, 6){.8333}}
\put(0, 0){\line(6, 1){1}}
\put(0, 0){\line(6, 5){1}}
\end{picture}
```

Curve di Bézier

```
\begin{picture}(50,30)(-10,10)
\linethickness{1pt}
\qbezier(0,0)(10,30)(50,30)\thinlines
\put(0,0){\line(1,3){10}}
\put(50,30){\line(-1,0){40}} \end{picture}
```

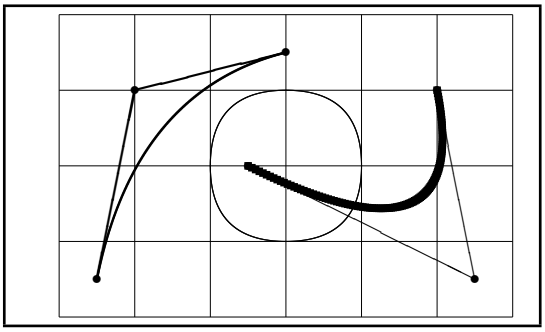


```
\begin{picture}(80,100)(0,0)
\bezier{400}(0,0)(40,100)(80,0)
\put(0,0){\line(2,5){40}}
\put(40,100){\line(2,-5){40}}
\end{picture}
```



```
\setlength{\unitlength}{1cm}
\begin{picture}(6, 4)
\linethickness{0.075mm}
\multiput(0, 0)(1, 0){7}{\line(0, 1){4}}
\multiput(0, 0)(0, 1){5}{\line(1, 0){6}}
\put(.5, .5){\circle*{.1}}
\put(1, 3){\circle*{.1}}
\put(3, 3.5){\circle*{.1}}
\thicklines
\put(.5, .5){\line(1, 5){.5}}
\put(1, 3){\line(4, 1){2}}
\qbezier(.5, .5)(1, 3)(3, 3.5) \thinlines
\put(2.5, 2){\circle*{.1}}
\put(5.5, .5){\circle*{.1}}
\put(5, 3){\circle*{.1}}
\put(2.5,2){\line(2, -1){3}}
\put(5.5,.5){\line(-1,5){.5}}
\linethickness{1mm}
\qbezier(2.5, 2)(5.5,.5)(5,3) \thinlines
```

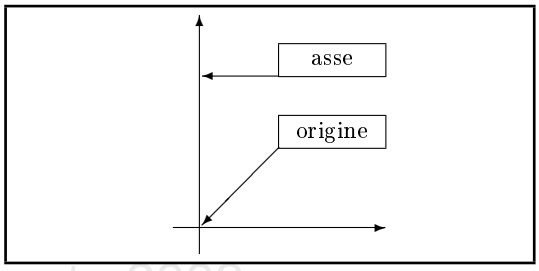
(*segue alla colonna a fianco*)



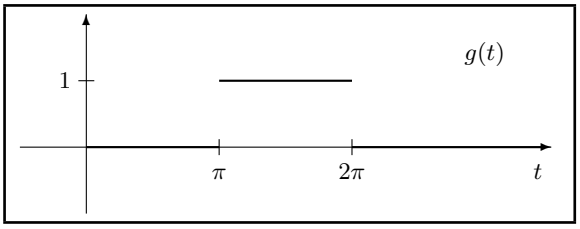
```
\qbezier(4, 2)(4, 3)(3, 3)
\qbezier(3, 3)(2, 3)(2, 2)
\qbezier(2, 2)(2, 1)(3, 1)
\qbezier(3, 1)(4, 1)(4, 2)
\end{picture}
```

Vettori e Coordinate

```
\begin{picture}(80,100)(-10,-10)
\put(-10,0){\vector(1,0){80}}
\put(0,-10){\vector(0,1){100}}
\put(30,30){\framebox(40,12){origine}}
\put(30,30){\vector(-1,-1){29}}
\put(30,57){\framebox(40,12){asse}}
\put(30,57){\vector(-1,0){29}}
\end{picture}
```

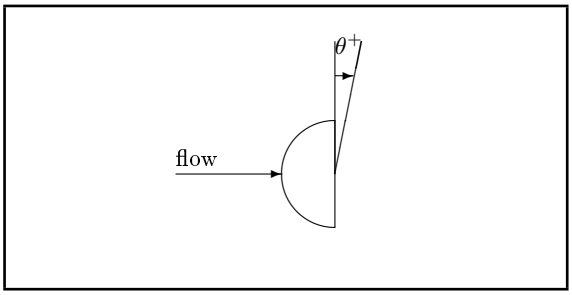


```
\begin{picture}(200,75)
\put(0,25){\vector(1,0){200}}
\put(25,0){\vector(0,1){75}}
\put(75,22){\line(0,1){6}}
\put(125,22){\line(0,1){6}}
\put(22,50){\line(1,0){6}}
\thicklines  \put(25,25){\line(1,0){50}}
\put(75,50){\line(1,0){50}}
\put(125,25)\line(1,0)72
\put(17,50){\makebox(0,0){$1$}}
\put(75,13){\makebox(0,0)[b]{$\pi$}}
\put(125,13){\makebox(0,0)[b]{$2\pi$}}
\put(195,13){\makebox(0,0)[b]{$t$}}
\put(175,60){\makebox(0,0){$g(t)$}}
\end{picture}
```



Misure angolari

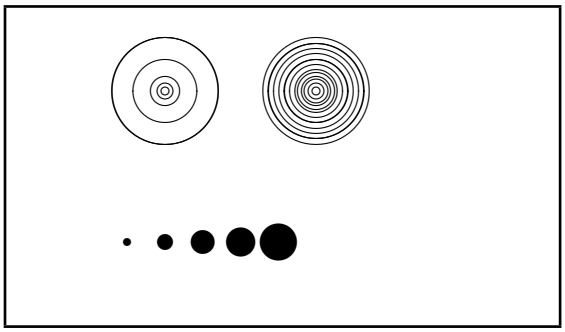
```
\begin{picture}(80,80)(0,0)
\put(60,77){\vector(1,0){7}}
\put(60,85){$\theta\hat{}$}
\put(60,40){\line(1,5){10}}
\put(60,40){\line(0,1){50}}
\put(60,40){\oval(40,40)[l]}
\put(60,20){\line(0,1){40}}
\put(0,40) {\vector(1,0){40}}  \put(0,43)
{flow}
\end{picture}
```



Cerchi

```
\setlength{\unitlength}{1mm}
\begin{picture}(60, 40)
\put(20, 30){\circle{1}}
\put(20, 30){\circle{2}}
\put(20, 30){\circle{4}}
\put(20, 30){\circle{8}}
\put(20, 30){\circle{16}}
\put(20, 30){\circle{32}}
\put(40, 30){\circle{1}}
\put(40, 30){\circle{2}}
\put(40, 30){\circle{3}}
\put(40, 30){\circle{4}}
\put(40, 30){\circle{5}}
\put(40, 30){\circle{6}}
\put(40, 30){\circle{7}}
\put(40, 30){\circle{8}}
\put(40, 30){\circle{9}}
\put(40, 30){\circle{10}}
\put(40, 30){\circle{11}}
```

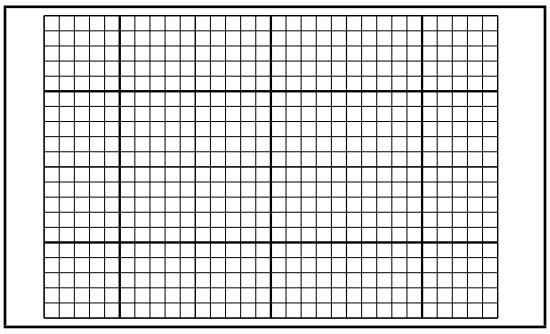
(*segue nella colonna a fianco*)



```
\put(40, 30){\circle{12}}
\put(40, 30){\circle{13}}
\put(40, 30){\circle{14}}
\put(15, 10){\circle*{1}}
\put(20, 10){\circle*{2}}
\put(25, 10){\circle*{3}}
\put(30, 10){\circle*{4}}
\put(35, 10){\circle*{5}}
\end{picture}
```

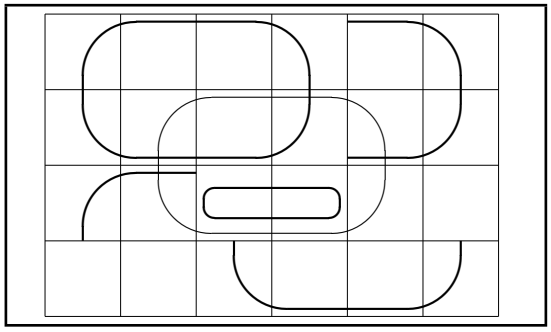
Griglie \multiput e \linethickness

```
\setlength{\hunitlength}{2mm}
\begin{picture}(30, 20)
\linethickness{0.075mm}
\multiput(0,0)(1,0){31}{\line(0,1){20}}
\multiput(0,0)(0,1){21}{\line(1,0){30}}
\linethickness{0.15mm}
\multiput(0,0)(5,0){7}{\line(0,1){20}}
\multiput(0,0)(0,5){5}{\line(1,0){30}}
\linethickness{0.3mm}
\multiput(5,0)(10,0){3}{\line(0,1){20}}
\multiput(0,5)(0,10){2}{\line(1,0){30}}
\end{picture}
```



\oval

```
\setlength{\unitlength}{10mm}
\begin{picture}(6,4)
\linethickness{0.075mm}
\multiput(0,0)(1,0){7}{\line(0,1){4}}
\multiput(0,0)(0,1){5}{\line(1,0){6}}
\thicklines   \put(2,3){\oval(3,1.8)}
\thinlines   \put(3,2){\oval(3,1.8)}
\thicklines
\put(2,1){\oval(3,1.8)[tl]}
\put(4,1){\oval(3,1.8)[b]}
\put(4,3){\oval(3,1.8)[r]}
\put(3,1.5){\oval(1.8, 0.4)}
\end{picture}
```

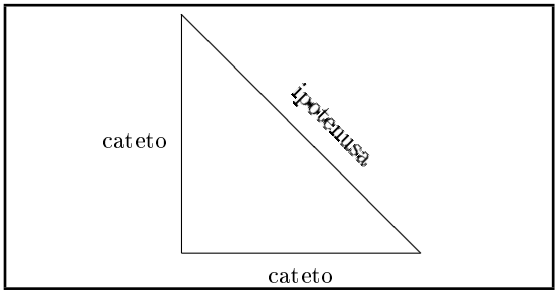


Nel manuale di U. Oswald si trovano varie costruzioni grafiche anche di una certa complessità. La presenza di sorgenti favorisce la comprensione della metodologia di lavoro dell'ambiente.

L'ambiente inoltre può essere arricchito anche con diverse altre istruzioni grafiche per ottenere particolari figure, come in quest'applicazione:

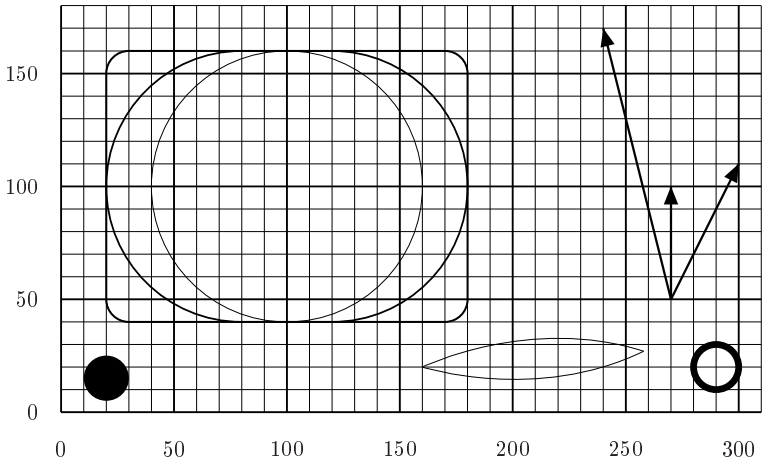
```
\usepackage{graphicx,rotating}

\begin{picture}(100,100)
\put(60,100){\line(0,-1){90} }
\put(60,10){\line(1,0){90}
    \makebox(80,-17){cateto} }
\put(60,100){\line(1,-1){90} }
\put(40,50){\makebox(5,5){cateto} }
\put(100,70){\begin{turn}{-45}
    \makebox{ipotenusa}\end{turn} }
\end{picture}
```



pspicture

DAVID CARLISLE ha rilasciato il package nel 1999 per estendere le funzioni dell'ambiente picture. L'applicazione poggia sui comandi PostScript ma non costituisce in fondo un gran passo avanti, perché se è più versatile nel disegno dell'angolazione delle linee e nella regolazione dello spessore di queste, soffre di limitazioni nel creare i cerchi, nel senso che anche in questo caso si tratta di




```
\usepackage{pspicture}\usepackage{graphpap}

\begin{picture}(180,150)
\graphpaper(0,0)(310,180)
\put(160,20){\Curve(98,7){2}}
\put(160,20){\Curve(98,7){-2}}
\linethickness{0.4pt} \put(100,100){\oval(120,120)}
\linethickness{0.8pt} \put(100,100){\oval(160,120)}
\linethickness{0.9pt}\put(100,100){\oval[10](160,120)}
\thinlines \put(20,15){\circle*{20}}
       \linethickness{3pt} \put(290,20){\circle{20}}
       \linethickness{1pt}
               \put(270,50){\Vector(0,50)}
               \put(270,50){\vector(5,10){30}}
               \put(270,50){\vector(-5,20){30}}
\end{picture}
```


Figura 13.2: Grafici con il package pspicture

misure prederminate e non alterabili più di tanto. Lo stesso manuale [5, IV] è assai stringato: due paginette con le istruzioni ed una con i grafici, se si eccettuano le poche restanti pagine destinate allo stile.

Le istruzioni basilari di picture, quali ad esempio `\circle` sopravvivono, ma le dimensioni del cerchio nella variante asteriscata (`\circle*`) sono *sciolte* dai parametri dello spessore di linea (`\linethickness`).

Una delle poche innovazioni è costituita dall'istruzione `\arrowlength`, che in congiunzione con le istruzioni `\vector` e `\Vector` permette di determinare lo spessore delle frecce, come prima si diceva. L'altra rilevante, rispetto all'ambiente picture del LaTeX, sta nel fatto che pspicture si sgancia dalla definizione ad inizio d'ambiente dell'unità e misura di lunghezza che s'era vista nella forma `\setlength{\unitlength}{misura}`.

In figura 13.2 i grafici sono rappresentati su una griglia per fat apparire le coordinate di lavoro dell'ambiente: si osservino i grafici in griglia leggendo anche i relativi sorgenti.

Per `\put(160,20){\Curve(98,7){2}}` e per `\put(160,20){\Curve(98,7){-2}}` vengono dise- `\Curve`

gnati due tratti curvilenei che contrapposti l'uno all'altro assomigliano vagamente ad un ellisse. La prima serie di numeri individua le coordinate sulla griglia, la seconda esprime le misure del tratto curvilineo, ed il terzo numero infine l'altezza rispetto a gli estremi: per il valore `{-2}` un tratto è speculare all'altro (valore `{2}`).

`\oval`

Alle istruzioni successive con `\linethickness{0.8pt}` è stabilito lo spessore di tre ovali sovrapposti che si toccano in due punti della circonferenza. Le coordinate di posizionamento sono eguali per tutti `\put(100,100)`, e come si nota sono riferite ai rispettivi centri. Le ultime coppie di numeri `{\oval(120,120)}`, `{\oval(160,120)}` e `{\oval[10](60,120)}` esprimono quindi rispettivamente: un cerchio, perché per `\oval(120,120)` l'ovale degenera in un cerchio, e i due ovali. La misura fra parentesi quadre (in questo caso `[10]`) determina, in ovvio rapporto alle dimensioni, l'entità dello *schiacciamento*. In questo caso ponendo per il sorgente mostrato un valore di `[57]`, i due ovali si andrebbero a sovrapporre.

`\circle`

`\circle*`

I due cerchietti posti quasi agli estremi sono espressione di `\circle`, e `\circle*`, e disegnano un cerchietto pieno ed uno vuoto. .

`\vector`

`\Vector`

Le istruzioni `\vector` e `\Vector` a questo punto dovvrebbero apparire chiare senza altre spiegazioni. Un ultimo richiamo è sull'istruzione `\linethickness{1pt}` che governa i tre vettori e che può essere sostituita dall'altra `\thinlines`.

graphpap

Il package graphpap di cui ci si è serviti all'esempio precedente per focalizzare su una griglia figure geometriche, è opra di LESLIE LAMPORT e come s'è visto non disegna una griglia astratta come quella composta a pagina 340, ma permette in base a parametri stabiliti dall'utente di dimensionare automaticamente la griglia e le sue coordinate.

Le istruzioni si esauriscono nella forma `\graphpaper(x,y)(nn,nn)`, dove i valori di `x` ed `y` rappresentano le coordinate di origine, e l'altra serie di numeri disegna dimensionalmente la griglia rappresentata con un valore costante di 10. Nella griglia appena vista le coordinate erano `\graphpaper(0,0)(310,180)`.

pict2e

pict2e è un package che ha avuto una *gravidanza* lunga e travagliata: annunciato da LESLIE LAMPORT addirittura nel 1994 in occasione della seconda edizine del LaTeX Book, fino al 2003, non fu mai realmente scritto poichè il team del progetto LaTeX declinava di produrrlo. Sino all'anno della comparsa il foglio di stile forniva solo l'avvertenza che... il package non era ancora disponibile.

Nel 2003 ROLF GÄSSLEIN, NIEPRANSCHK ROLF e HOSEPH TKADLEC rilasciavano il package che è stato di recente (luglio 2008) aggiornato. Accompagna la distribuzione un manualetto di poche pagine [10, IV] per il quale gli autori non si sono davvero sprecati, ispirato più a un *tirar via* che a una versione *beta-test*: poteva essere scritto in maniera più chiara e con qualche esempio in più.

Documentazione su pict2e è reperibile a firma di CLAUDIO BECCARI: *LaTeX 2ε* `pict2e` *and complex numbers* apparso nel 2007 sul vol. 27 del TUGboat [4, IV]. L'autore nello stesso anno, in occasione del XXVII meeting dei TeX Users Group ha trasformato l'articolo in una serie di slides disponibili all'indirizzo: `www.ucam.ac.ma/fssm/tug2006/presentations/tug2006-Beccari.pdf`.

Impostazioni

R. Niepraschk e J. Tkadlec in un manualetto d'accompagnamento al pacchetto (*Graphics drivers for* pict2e) [14] illustrano le impostazioni d'opzione da dare per il richiamo del package. Queste sono riportate in tabella 13.1, ed indicano, ad esempio, che il package può essere richiamato nella forma `\usepackage[dvips]{pict2e}`.

Opzioni driver		Opzioni di sistema
dvips	dvipsone	ltxarrows
xdvi	dviwindo	pstarrows
pdftex	dvipdf	original
vtex	textures	debug
dvipdfm	pctexps	hide
oztex	pctex32	

Tabella 13.1: Opzioni di pict2e

Nella tabella originaria i drivers recano accanto una x che esprime costantemente la legenda *supported but untested* o *not yet implemented* a testimonianza che la scrittura deve considerarsi ancora non definitiva. Le opzioni della terza colonna riguardano (prime due righe) implementazioni di pict2e che usa lo stile di pstricks, e opzioni d'uso per sviluppatori e test di di istruzioni Nel pacchetto è disponibile un file (pct2e.cfg) accompagnato da un altro (pic2e-example.cfg) per le ventuali configurazioni di sistema

JJJJJJJJJJJ

kkkkkkkkkkkkkkkkkkkkkkkkkkkk

curve2e

IIIIIIIII

curves

KKKKKKKKKKKK

pgf

pgf che si deve a TILL TANTAU è un package che in passato ha goduto nella grafica di un certo successo che in molti ambienti continua ancora. T. Tantau ha in verità sviluppato tutta una serie di applicativi (pgfbaseimage, pgfbaseplot, e diversi altri) che rispondono con la creazione di distinti ambienti alle esigenze della grafica.

Disponendo oggi delle potenzialità di pstricks questo package se non obsoleto, credo possa considerarsi oggi superato, a meno che la sua continua evoluzione non riservi nuove sorprese.

L'ambiente che il package mette a disposizione (tikzpicture) conosce una sintassi diversissima da quella di picture e presenta comunque notevoli potenzialità.

Ma dovendo imparare a conoscere un ambiente per le proprie necessità, mi sembra più naturale invitare ad approfondire pstricks che, seppur gestibile nell'ambiente PostScript, e questa, a mio giudizio, è tutt'altro che una limitazione!, ha potenzialità in continuo divenire.

pstricks

Introduzione

Le limitate capacità in ambiente puramente grafico di TEX e LATEX furono affrontate fa TIMOTHY VAN ZANDT il quale concepì l'idea di un package che potesse adattare il ruvido linguaggio di LATEX alle esigenze di una grafica professionale e moderna.

Nacque così pstricks, un pacchetto dalle potenzialità e mezzi espressivi notevoli ma anche di una certa complessità manipolativa, che dette origine ad una serie di pacchetti caratterizzati dalle iniziali pst- e finalizzati ad ampliare l'ambiente grafico-matematico anche in 3d, con applicazioni davvero sofistificate.

pstricks è uno standard nelle distribuzioni Unix e Linux, ma, almeno fino a poco tempo fa, non era di serie con MiKTEX. Nel caso va scaricato dai siti dedicati, CTAN o DANTE. Resta sottinteso che è d'obbligo comunque per l'eventuale utilizzatore una visita al sito `www.pstricks.de`.

pstricks e PDF

Gli strumenti della serie pst-, e questa forse è un'altra loro *rusticità*, mal si coniugano –come già detto– con i comandi della famiglia `pdflatex`, non tanto, e non soltanto, perchè sono stati pensati per il formato PS, quanto perché le istruzioni del formato PS possono definirsi istruzioni di alto livello, incompatibili con il formato PDF.

L'istruzione `ps2pdf1.x` (da PS a PDF) [2] risolve quasi completamente il problema, ma, specie per gli utenti di MiKTEX, questo può talvolta creare problemi, ed allora non resta che percorrere la via già illustrata alla sezione Produzione di output in PS come esempi, [3] producendo a parte un file, magari con l'utilizzo della classe `minimal`, ed introducendolo quindi nel documento come immagine con l'istruzione `\includegraphics{file_immagine.ps}`.

pdftricks

In alternativa, pensando ad un lavoro dal taglio professionale e non *adattato* al formato PDF, si può ricorrere a pdftricks, un pacchetto ideato da C. V. RADHAKRISHNAN, che può esssere usato per soccorre a tale carenza e che lavora bene sotto i comandi `pdflatex`.

Le istruzioni da fornire nel preambolo si presentano in questa forma:

```
\usepackage[shell]{pdftricks}
\begin{psinputs}
\usepackage{pstricks,color,pstcol} \usepackage{multido}
\usepackage{pst-plot,pst-tree,pst-eps,pst-node,pst-eps}
\end{psinputs}
\begin{document} testo ed istruzioni \end{document}
```

L'elenco dei pacchetti citati nel breve sorgente-esempio suffraga l'affermazione svolta all'inizio della presentazione che il package fa parte di una famiglia di pacchetti dedicati e finalizzati a specifiche applicazioni, uniti fra loro dal medesimo linguaggio di programmazione, dalla stessa tipologia di istruzioni, dall'identica finalità applicativa: realizzazione di grafica eccellente in LaTeX.

\usepackage {pst-all}

Se allora s'intende servirsi di tutti questi pacchetti della serie, non conviene ricorrere all'usuale istruzione `\usepackage{pst-...,pst-...,pst-...}`, dal momento che esiste un comando globale `\usepackage{pst-all}` che carica tutti le applicazioni collegate e derivate.

Quest'istruzione però va usata con cautela e deve essere utlizzata in quei casi in cui non molti altri pacchetti siano presenti, sia perché possono sorgere problemi di compatibilità, sia perché i contatori possono saltare facilmente a ragione delle numerose istruzioni attivate.

Tanto premesso, in questa parte saranno discusse le generalità di quest'applicazione, ma prima ritengo necessaria un'ulteriore precisazione, dedicata specialmente al *critico* prima che al lettore. [4]

Nella parte V, dedicata alle applicazioni avanzate, saranno esaminati altri pacchetti della serie pst- e con questo collegati, nonché le loro relative applicazioni.

2. Vedi sezione Da PS a PDF tramite ps2pdf, a pagina 86.

3. Vedi la sezione 6.4.

4. La trattazione riservata a questo package apparirà sicuramente ai più, specie ai tecnici, sproporzionata in un *presuntuoso* manuale, redatto per di più in forma di *Appunti*, di semplice introduzione a LaTeX.

Ma in questo non vi è solo la constatazione, puramente soggettiva, che pstricks rappresenta, al pari di ledmac e di qualche altro package, una delle più riuscite creazioni per questo linguaggio di programmazione, ma anche una motivazione psicologica d'interesse per i risultati che nel mio campo il pacchetto mi ha consentito di raggiungere.

A ciò si aggiunga che dal punto di vista emotivo, se mai questo aggettivo ha senso in ambito informatico, le creazioni con pstricks sono fra quelle che più mi hanno colmato di soddisfazione nel vedere la compilazione andata a buon fine, superate forse soltanto, agli inizi della mia esperienza, dalle compilazioni di musixtext. È questo in fondo anche un modo di tributare il personale mio riconoscimento all'autore dei packages della serie pst- ed ai successivi sviluppatori.

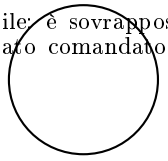
Modalità di lavoro e caratteristiche

Come per picture anche l'ambiente disegnato da pstricks: pspicture, lavora secondo il sistema delle coordinate cartesiane, inserendo ogni figura in un box individuato sulla pagina dal punto in basso a sinistra e dal punto in alto a destra e così rappresentato.

In un primo avvicinamento si possono dunque individuare queste coordinate come `(xs,ys)` e `(xd,yd)`, ove per `s` e `d` s'intenderanno, rispettivamente, le coordinate di sinistra e di destra.

Tanto premesso, la minimale istruzione `\pscircle(2,0){1}` volta a disegnare un cerchio apparirà nella forma mostrata in questa pagina.

È evidente che il disegno è inservibile: è sovrapposto al testo perché il sistema ha letto le coordinate e l'ha posizionato dove è stato comandato nel punto del testo in cui ha trovato le istruzioni. Un'istruzione del tipo:



```
\begin{pspicture}(1,1)(2,2)
\pscircle(2,0){1}
\end{pspicture}
```

avrebbe posizionato in locazione fisica diversa la figura rendendola comunque di eguali dimensioni.

Va notato inoltre che in assenza di ulteriori istruzioni, il pacchetto ha usato le istruzioni dimensionali di default scalate per 1 cm. Per ottenere un disegno più pulito e coerente bisogna aggiungere alcune istruzioni.

Il disegno, visibile questa volta sulla riga di testo  –e rappresentato secondo le istruzioni `\pscircle[fillcolor=red,fillstyle=solid](0,0){.1}`– disegna un cerchio di dimensioni più ragionevoli `{.1}` la rappresentazione dell'oggetto, colorandone l'interno di rosso con l'istruzione `[fillcolor=red]`.

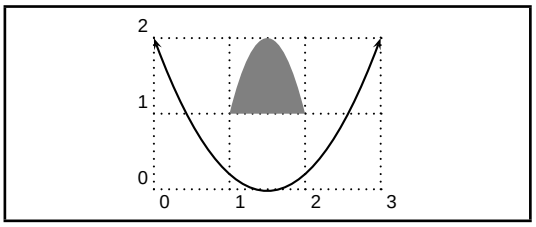
Questi due semplici esempi non volevano costituire alcuna anticipazione di un discorso più articolato che ora si andrà a fare e che si riprenderà nella parte V dedicata alle applicazione avanzate, ma solo una significativa rappresentazione del *modus operandi* di questo pacchetto.

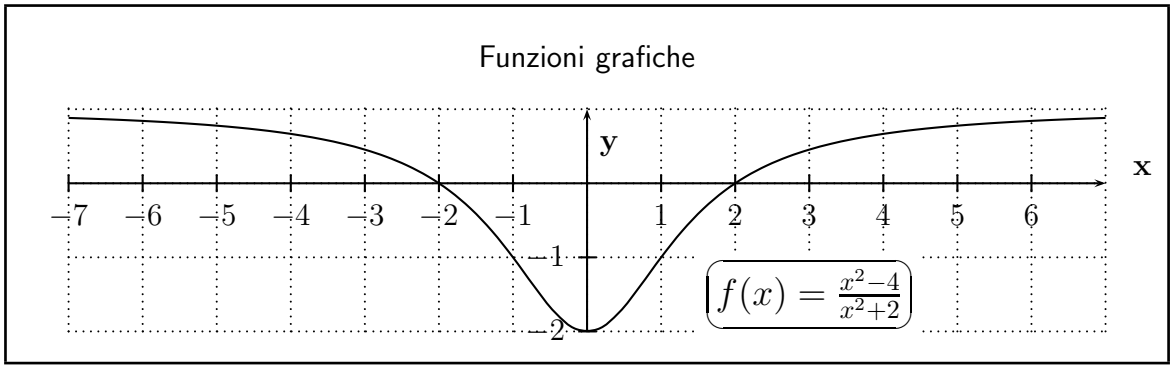
Prima ancora di addentrarci nelle particolarità tecniche, intendo però descrittivamente riassumere le caratteristiche di pstricks:

- le figure geometriche (cerchi ed *ovali*) possono avere del testo inscritto, e questa particolarità non è più riservata solo ai box;
- le famiglie delle linee e delle frecce si distinguono per una gran quantità di istruzioni personalizzate che permettono di adeguarle ad ogni esigenza;
- una linea può essere disegnata su una sequenza di punti fornendole una sola istruzione, senza che la sua inclinazione sia preventivamente calcolata;
- l'introduzione del concetto fisico-grafico dei *nodi* dà accesso ad una nuova via di programmazione grafica;
- frecce e forme sono largamente aggiustabili e variabili, ed il tracciamento di curve è abbastanza semplice, e contempla anche le cosiddette curve di Bezier.

Parabole

```
\psset{xunit=1cm,yunit=1cm,runit=1cm}
\begin{pspicture}(0,0)(3,2)
\psgrid[subgriddiv=1,%
griddots=10,%
gridlabels=7pt](0,0)(3,2)
\parabola*[linecolor=gray](1,1)(1.5,2)
\parabola<->(0,2)(1.5,0)
\end{pspicture}
```





Funzioni grafiche

Preambolo

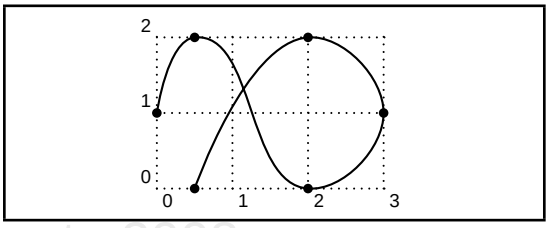
```
\usepackage[dvips]{graphicx,pstricks,pst-plot,fancybox,ammymb,color}
\newpsobject{showgrid}{psgrid}{subgriddiv=1,griddots=10,gridlabels=0pt}
```

Istruzioni

```
\psset{unit=1cm}
\begin{pspicture}(-7,-2)(7,1) \showgrid
\psaxes{->}(0,0)(-7,-2)(7,1)
\psplot[plotstyle=curve]{-7}{7}{x x mul 4 sub x x mul 2 add div}
\rput(7.5,0.2){$\mathbf{x}$} \rput(0.3,0.5){$\mathbf{y}$}
\large{%
\rput(3,-1.5){\colorbox{white}{\ovalbox{$f(x)=\frac{x^2-4}{x^2+2}$}}} }    \end{pspicture}
```

Curve complesse

```
\psset{xunit=1cm,yunit=1cm,runit=1cm}
\begin{pspicture}(0,0)(3,2)
\psgrid[subgriddiv=1,%
griddots=10,%
gridlabels=7pt](0,0)(3,2) \pscurve[showpoints=true]{-}%
(0,1)(0.5,2)(2,0)(3,1)%
(2,2)(0.5,0) \end{pspicture}
```



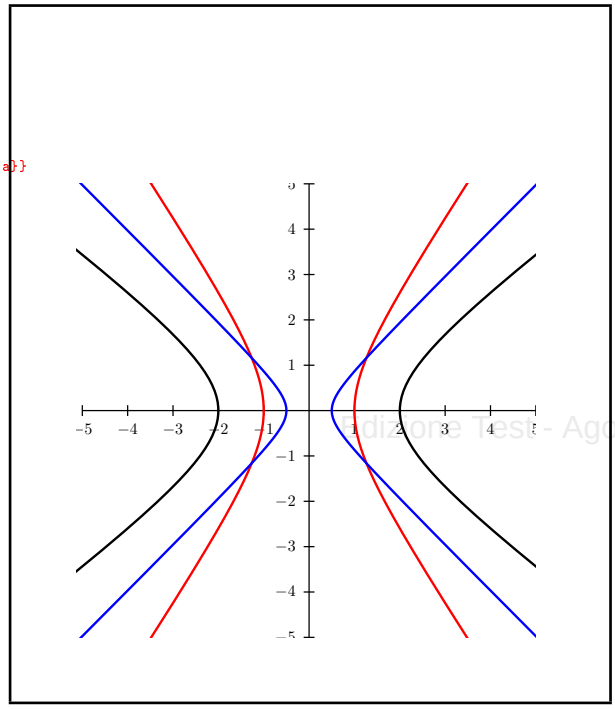
Iperbole

Preambolo

```
\usepackage{graphicx}
\usepackage{pstricks,pst-plot,amssymb,color,fancybox}
\makeatletter \def\psthyperbola{\pst@object{psthyperbola}}
\def\psthyperbola@i#1#2{%
\pst@killglue \begingroup \use@par
\psthyperbola@ii{#1}{#2}
\psthyperbola@ii{#1 neg}{#2}
\endgroup }
\def\psthyperbola@ii#1#2{% a b
\addto@pscode{%
/a {#1} bind def /b #2 def
/d {1 t dup mul sub} bind def }
\parametricplot{-.99}{.99}{%
a t dup mul 1 add d div mul b t 2 mul d div mul}  }
\makeatother
```

Istruzioni

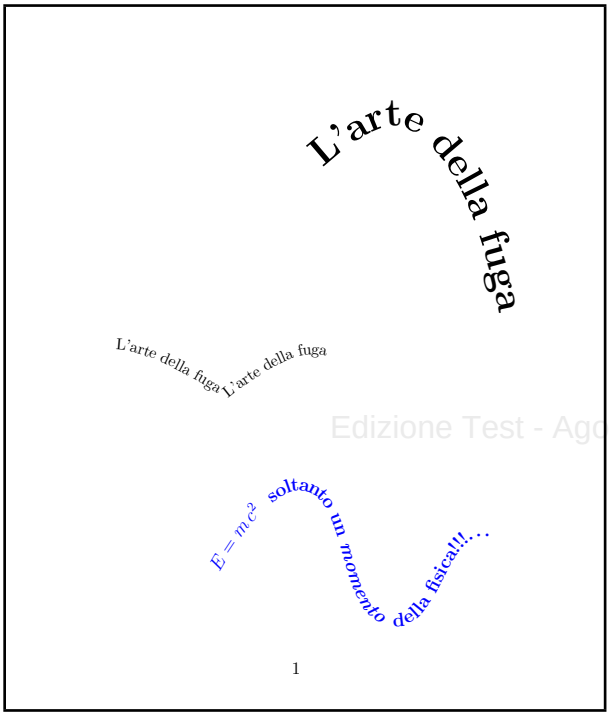
```
\begin{pspicture*}(-5,-5)(5,5)
\psaxes(0,0)(-5,-5)(5,5) \psset{linewidth=1.5pt}
\psthyperbola{2.0}{1.5}
\psthyperbola[linecolor=red]{1.0}{1.5}
\psthyperbola[linecolor=blue]{0.5}{0.5}
\end{pspicture*}
```



Testo in grafica

```
\usepackage{pstricks,pst-text,color}
\begin{pspicture}(6,10)
\small\psset{linestyle=none}
\pstextpath[r]{\pscurve(0.3,3.2)(2,4.3)(4.45,3.3)}
{L'arte della fuga}
\pstextpath[l]{\pscurve(4.55,3.2)(6.5,4.2)(9,2.8)}
{L'arte della fuga}
\pstextpath[l]{\pscurve(6.55,8.2)(8.5,9.2)(11,2.8)}
{ {\bfseries\Huge L'arte della fuga }}
\end{pspicture} \begin{pspicture}(-2, -1.2)(3, 0.2)

\psset{linecolor=white}
\pstextpath
{\pscurve(-4, -2)(-2, 0)(0, -3)(2, -1)%
(3, -2)(5, -3)} {\color{blue} {\large \textbf{$E = mc^2$ è soltanto un \emph{momento} della
fisica!!!\ldots } } }
\end{pspicture}
```



Bibliografia

[1] Anonimo, *The Tale of* floatflt, versione non numerata, Autore, versione e data sconosciuti.
htt://www.homenet.sc/matsd/latex/floatexem.pdf

[2] Luciano Battaia, *PSTricks, ovvero ... la cilegina sulla torta...* , versione non numerata, 20 gennaio 2006.
http://www.batmath.it/latex/pdfs/guida_pstricks.pdf

[3] Claudio Beccari, *Graphics in LaTeX*, The pracTeX Journal, n. 1, 2007, Article revision, 27 agosto 2007
http://dw.tug.org/pracjourn/2007-1/beccari/beccari.pdf

[4] Claudio Beccari, *LaTeX 2ε, pict2e and complex numbers*, TUGboat, volume 0 2006, N.° 0, pagg. 1001-1011, draft, 23 agosto 2005
www.tug.org/TUGboat/Articles/tb27-2/tb87becc.pdf

[5] David Carlisle, *The* pspicture *package*, versione 2.02, 11 aprile 1999
www.hep2.fzu.cz/tex/texmf-dist/doc/latex/carlisle/pspicture.pdf

[6] Luca Caucci Mario Spadaccini, *Gestione di figure e Tabelle con LaTeX* , versione 1.0.0
http:www.guit.sssup.it/downloads/fig-tut.pdf, 21 ottobre 2004.

[7] Douglas Steven Cochran, *The subfigure package*, versione 2.14, 2 luglio 2002.
http:www-2.cs.cmu.edu/afs/cs/usr/sdc/www/latex/subfigure.214.pdf

[8] Douglas Steven Cochran, *The captcont package*, versione 2.0, 14 febbraio 2004.
http:www-2.cs.cmu.edu/afs/cs/usr/sdc/www/latex/captcont.20.pdf

[9] Patrick W. Daly, *Graphics and Colours with LaTeX*, versione non numerata, 4 giugno 1998,
http://tex.loria.fr/graph-pack/grf/grf.pdf

[10] Hubert Gässlein Rolf Niepraschk Joseph Tkadlec, *The pic2e package*, versione 0.2, 19 luglio 2008.
www.ctan.org/tex-archive/macros/latex/contrib/pict2e/pict2e.pdf

[11] Michel Gossens Frank Mittelback Sebastian Rahtz Denis Roegel Herbert Voss, *The LaTeX Graphics Companion*
Addison-Wesley, seconda edizione, 2007

[12] Indian TeX Users Group - AA.VV., *Online LaTeX Tutorial Part II - Graphics*, versione non numerata, 2002
http://sarovar.org/projects/pstricks

[13] Michael Niedermair, *Zeichnungen mit PSTricks erstellen*, versione non numerata, Bayerischer TEX-Stammtisch, 9 agosto 2003
http://www.uwe-siart.de/typografie/pstricks_20030809.pdf

[14] Rolf Niepraschk Joseph Tkadlec, *TGraphics drivers for* pict2e, versione 0.1r, 25 giugno 2004.
www.ctan.org/tex-archive/macros/latex/contrib/pict2e/p2e-drivers.pdf

[15] Urs Oswald, *Graphics in LaTeXe*, versione non numerata, 10 marzo 2003,
http://www.ursoswald.ch/LaTeXGraphics/overview/latexgraphics.pdf

[16] Keith Reckdahl, *Using Imported Graphics in LaTeX 2ε*, versione 2.0, 15 dicembre 1997.
ftp://ftp.dante.de/tex-archive/info/epslatex.pdf

[17] Timothy Van Zandt, *Documentation for* fancybox.sty: *Box tips and tricks*, versione 1.3, 19 settembre 2000
www.ctan.org/tex-archive/macros/latex/contrib/fancybox/fancybox.sty

[18] Herbert Voss, *Multicols with Pictures*, versione non numerata, 31 dicembre 2000
http://www.texnik.de/multicols/2columnArticleWithPics.pdf.

[19] Herbert Voss, *Package* `hvfloat` *Rotating objects and captions*, versione 1.10, 16 giugno 2004
http://dante.ctan.org/CTAN/macros/latex/contribut/hvfloat/hvfloat.pdf