Gestione avanzata delle figure in LaTeX: l'annotazione di illustrazioni e grafici con psfrag/pstricks e PGF/Tikz

Agostino De Marco

Sommario

In questo articolo viene mostrato come utilizzare le potenzialità di LaTeX e dei pacchetti pstricks e PGF/Tikz nella produzione di illustrazioni e grafici. L'argomento è trattato ad un livello tecnico medio/alto. Lo scopo è quello di fornire degli esempi di annotazione di immagini bitmap o vettoriali. Le tecniche illustrate permettono di ottenere delle figure correttamente annotate. Tali figure possono essere incorporate in un documento LaTeX con la certezza che ci sia coerenza tra lo stile tipografico delle annotazioni e quello del testo corrente.

Abstract

This article shows how the combination of LaTeX with the package pstricks or with PGF/Tikz can be used to produce advanced, nice-looking illustrations and plots. This subject is dealt with a technical level biased towards intermediate/advanced users. The aim of the work is presenting a number of examples of how a figure, that is a bitmapped or vector image, might be annotated according to the typographic style of the main LaTeX document and of the displayed math.

1 Introduzione e motivazioni

Questo articolo nasce a conclusione di un anno di lavoro in cui ho curato, sia come coautore che come compositore del manoscritto, la stesura di una monografia destinata ad allievi ingegneri aerospaziali (Coiro *et al.*, 2008). Da utilizzatore abituale di LaTeX ho dato molta importanza in questo lavoro alla cura dei dettagli ed alla resa tipografica del manoscritto. Ho dovuto tener conto della qualità delle molte formule matematiche, dell'impaginazione, ma soprattutto dell'enorme mole di materiale visivo richiesto in un libro di testo. La creazione e la gestione di numerose illustrazioni e soprattutto di tanti grafici è stato uno dei compiti più impegnativi. Ciascuna figura ha rappresentato, per così dire, una storia a sé.

Lo scopo di questo articolo è quello di offrire al lettore gli elementi sufficienti per produrre un documento utilizzando LaTeX o pdfLaTeX, incorporando illustrazioni e grafici *correttamente annotati*, cioè contenenti elementi testuali esplicativi composti nello stesso stile tipografico del testo corrente.

Nella prima parte riassumerò alcuni concetti generali (paragrafo 2), di grande importanza per capire il ruolo delle figure in un manoscritto, ed accennerò alla tecnica di gestione standard delle figure in LaTeX (paragrafo 3). Successivamente fornirò esempi di creazione e di gestione di illustrazioni e grafici (paragrafi 4 e 5) destinati a comparire in un documento professionale di grandi dimensioni.

2 La figura come strumento di comunicazione

2.1 Il livello di comunicazione analogica

Le figure all'interno di un manoscritto sono diventate uno strumento di comunicazione molto usato e molto importante, soprattutto nel mondo tecnico-scientifico.

Nel testo di Matricciani (2008) è ben spiegato come, dal punto di vista del linguaggio comunicativo, le figure rappresentino una forma di comunicazione "analogica". Essendo forme di comunicazione visiva, esse convogliano verso chi le guarda una serie di informazioni che vengono percepite simultaneamente. Da questo punto di vista, le figure all'interno di un documento si contrappongono al testo corrente. Il linguaggio del testo, cioè quello verbale, rappresenta una forma di comunicazione "numerica", sequenziale, perché è fatto di una sequenza di entità discrete, le parole. Il testo, a causa della sua natura sequenziale, è fondamentalmente una lista di concetti e istruzioni forniti al lettore perché ricostruisca le molteplici idee presenti nella mente di chi scrive. Le figure, al contrario, spesso non sono vincolate a una struttura sequenziale (tranne che per le figure multiple). Ciascuna di esse è in grado di esprimere molteplici relazioni presenti originariamente nella mente dell'autore.

Il testo di un documento, in quanto messaggio di tipo numerico, va *decodificato* dal lettore, con un certo grado di fatica che dipende dalle capacità dell'autore ed in parte dalla qualità tipografica. Si può scegliere ad esempio di non leggere un testo, o di saltarne una parte. Le figure, solo perché presenti nel manoscritto, sono informazioni che vengono ricevute comunque dal nostro sistema visivo. Quando vediamo un'immagine non possiamo

decidere di non vederla, la vediamo e basta. Le figure vanno quindi *interpretate* dal lettore, secondo un processo mentale diverso dalla decodifica. Nelle fotografie, nei diagrammi, nei grafici, le informazioni collegate tra loro vengono presentate simultaneamente, analogicamente; appaiono tutte sulla stessa porzione di foglio e la loro efficacia sarà tanto maggiore quanto più esse saranno semplici, chiare, complete e precise. Nella comunicazione in forma analogica, se ben congegnata, la visione sinottica semplifica l'assimilazione delle informazioni e riduce il carico d'elaborazione che grava sul lettore. Il giusto dosaggio fra forme di comunicazione numerica ed analogica fa di un documento, non solo di tipo tecnico-scientifico, un documento di grande qualità.

Nella mia esperienza di ricercatore e docente nell'ambito dell'ingegneria aerospaziale credo di poter citare, come esempio al di sopra di tutti, il testo di Anderson (2004). Questo autore si è guadagnato un titolo di professore emerito per la sua attività didattica e per aver progettato manuali per l'ingegneria di grande qualità, ben strutturati e con un eccellente equilibrio tra testo e figure.

Gli utilizzatori entusiasti di LaTeX sono notoriamente compiaciuti dell'alto livello di qualità tipografica dei loro documenti. I migliori risultati si ottengono comunque con un adeguato livello di conoscenza delle potenzialità del programma e dei suoi pacchetti di estensione. Quando si ha il compito di produrre un manoscritto ricco di figure, vi è la necessità di uno sforzo ulteriore, quello di assicurare che il livello di qualità del materiale visivo sia all'altezza di quello del testo.

2.2 Ruolo delle figure

Le illustrazioni in genere ed i grafici dovrebbero essere adeguati al tipo di documento in cui compaiono, preparate con gran cura per informare, persuadere e motivare il lettore, e non un ornamento del testo, inserite alla fine per motivi "estetici" o per dare un'aria di professionalità.

Sin dalla nascita della comunicazione moderna, in particolare di quella scientifica, scienziati, ingegneri architetti e ricercatori comunicano idee complesse, progetti importanti, particolari costruttivi e molte altre informazioni attraverso molti tipi di figure. I lettori gradiscono molto la loro presenza nei documenti perché possono farsi un'idea abbastanza precisa del contenuto del manoscritto guardando proprio le figure. Le figure danno una rappresentazione sinottica delle informazioni, in un istante visualizzano tanti dati e facilitano la comprensione di un argomento.

Gli stessi autori possono trarre grande utilità dalle figure poiché la fatica richiesta per progettarle e disegnarle aiuta ad approfondire e a chiarire l'argomento. Un autore potrebbe scegliere di comunicare una serie di informazioni soltanto con un testo, magari anche evidenziato in una tabella, ma la chiarezza, la visualizzazione, la comprensione di un dato argomento possono risultare di gran lunga più immediate attraverso una figura.

Le figure non dovrebbero essere viste come parti marginali di un manoscritto, da preparare dopo. Al contrario, spesso, decisi i punti centrali da raffigurare, l'autore scrive il manoscritto *in funzione delle figure*. E questo articolo non rappresenta un'eccezione.

Le illustrazioni o i grafici aiutano a rompere la monotonia della lettura (specialmente di quella tecnica) e, quando mostrano i punti principali di un lavoro, attraggono l'attenzione del lettore facilitando la lettura e la comprensione dell'argomento. *Per far sì che una figura assuma effettivamente un tale ruolo di elemento chiarificatore e di sintesi è necessario conferire ad essa un aspetto appropriato, a partire dalla corretta rappresentazione dei dati per finire con un'adeguata rappresentazione dei simboli.*

Le figure vengono, in genere, richiamate nel testo corrente almeno una volta. Per essere comprese, le figure non richiedono soltanto un commento nel testo corrente, ma anche un testo esplicativo che le accompagni: questo consiste in un titolo, una didascalia e, se è il caso, una legenda ed eventuali annotazioni aggiuntive. Questi elementi devono comunicare tutte le informazioni necessarie per rendere la figura più indipendente possibile dal testo e dalla posizione nel manoscritto. Gli utilizzatori di LaTeX conoscono bene questa necessità. Spesso molti di essi non riescono a resistere alla tentazione di "forzare" a tutti i costi l'algoritmo di posizionamento degli oggetti flottanti perché insoddisfatti della collocazione finale delle figure. In realtà l'uso della macro `\caption` e la possibilità di riferirsi a qualsiasi punto del testo rende questa insoddisfazione inessenziale.

2.3 Tipi di figure

Con il termine "figura" ci si riferisce ad una varietà di materiale iconografico: (a) le fotografie, (b) i disegni tecnici, (c) i diagrammi di dispersione, (d) i grafici, (e) gli istogrammi a barre, (f) i diagrammi di flusso (schemi a blocchi). Ciascun tipo di figura, a seconda della complessità, richiede delle attività di pianificazione e progetto adeguate e particolari. Una guida enciclopedica alla miriade di possibili rappresentazioni grafiche è il libro di Harris (1999). Nel seguito tratteremo in particolare le "illustrazioni", cioè i disegni che spiegano uno o più concetti con una serie di elementi grafici e delle annotazioni esplicative, ed i "grafici", ovvero i diagrammi comunemente usati nell'ingegneria e nelle scienze fisiche e matematiche.

Nella resa finale del documento tutto questo materiale visivo, proveniente da diverse fonti, è riprodotto tipicamente sotto forma di immagini. Nel caso delle fotografie esse provengono da scansioni ottiche o sono direttamente prodotte e immagazzi-

nate in formato digitale dalle moderne macchine fotografiche. Gli altri tipi di materiale iconografico vengono creati con i programmi di grafica più disparati, con i quali è possibile esportare una figura come immagine in un dato formato standard. In ogni caso, nel mondo della stampa si pensa alle figure di un manoscritto come ad oggetti esterni da incorporare, che non provengono, in genere, da una composizione da tastiera come avviene per le tabelle.

Con la diffusione dei personal computer e di potenti programmi di grafica oggi è possibile produrre facilmente molti tipi di figure. Gli stessi sistemi di videoscrittura sincrona permettono di fare grafici e scrivere testi contemporaneamente. Questa è certamente una funzione potente ed utile ma è bene usarla con molta attenzione, specialmente se è richiesto un documento finale di qualità professionale.

Da questo punto di vista LaTeX rappresenta uno strumento di lavoro eccezionale. Esistono dei pacchetti di estensione dedicati alla grafica molto potenti e versatili, che permettono di inserire dei comandi di disegno direttamente negli ambienti flottanti `figure`. È possibile cioè lavorare alle figure come al sorgente di un programma, con il suo particolare linguaggio di programmazione. Dunque, a parte le fotografie, gli utilizzatori di LaTeX hanno a disposizione una vantaggio considerevole nell'impegnativo lavoro di progettazione delle figure di un manoscritto.

3 Gestione delle figure in LaTeX

3.1 L'ambiente `figure` e il pacchetto graphicx

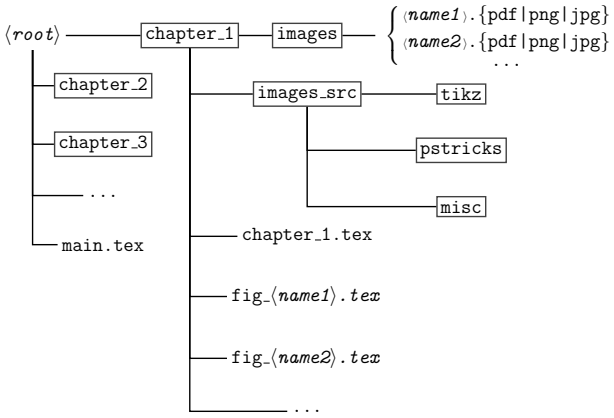
Per gli utilizzatori di LaTeX una figura è il contenuto di un ambiente `figure`. Tipicamente questo ambiente è usato per inserire immagini esterne attraverso il comando `\includegraphics` del pacchetto graphicx. Per approfondire l'uso di questo ambiente e delle opzioni del pacchetto graphicx, il lettore meno avanzato può consultare l'eccellente testo in italiano di Pantieri (2008).

L'ambiente `figure` può racchiudere una sequenza di comandi di disegno, direttamente compilati insieme al documento. Analogamente, si può predisporre un file esterno che contenga la sequenza di comandi di disegno, incorporandolo nel documento principale con il comando `\input`. In alternativa, si può avere un file esterno di comandi, che includa: (*a*) lo stesso ambiente `figure` (si veda, ad esempio, il file `fig_name1.tex` nella figura 1) e (*b*) un comando `\includegraphics` che incorpora un'immagine preconfezionata.

Quest'ultima strategia è preferibile per i manoscritti di notevole lunghezza, specialmente se ricchi di figure. Essa richiede che ciascuna figura sia prodotta a parte, in formato PDF, PNG o JPG.

Una possibile organizzazione dei file, in particolare, delle immagini e dei file di comandi che le incorporano nel documento principale, è rappresentata nella figura 1.

Nella generica cartella `images_src` si potranno conservare i sorgenti con i quali si generano le diverse figure attraverso compilazioni separate. L'autore potrebbe organizzare questa cartella in ulteriori sottocartelle che raggruppino i sorgenti a seconda dei metodi con cui sono create le immagini, ad esempio le cartelle `tikz`, `pstricks`, `misc`.



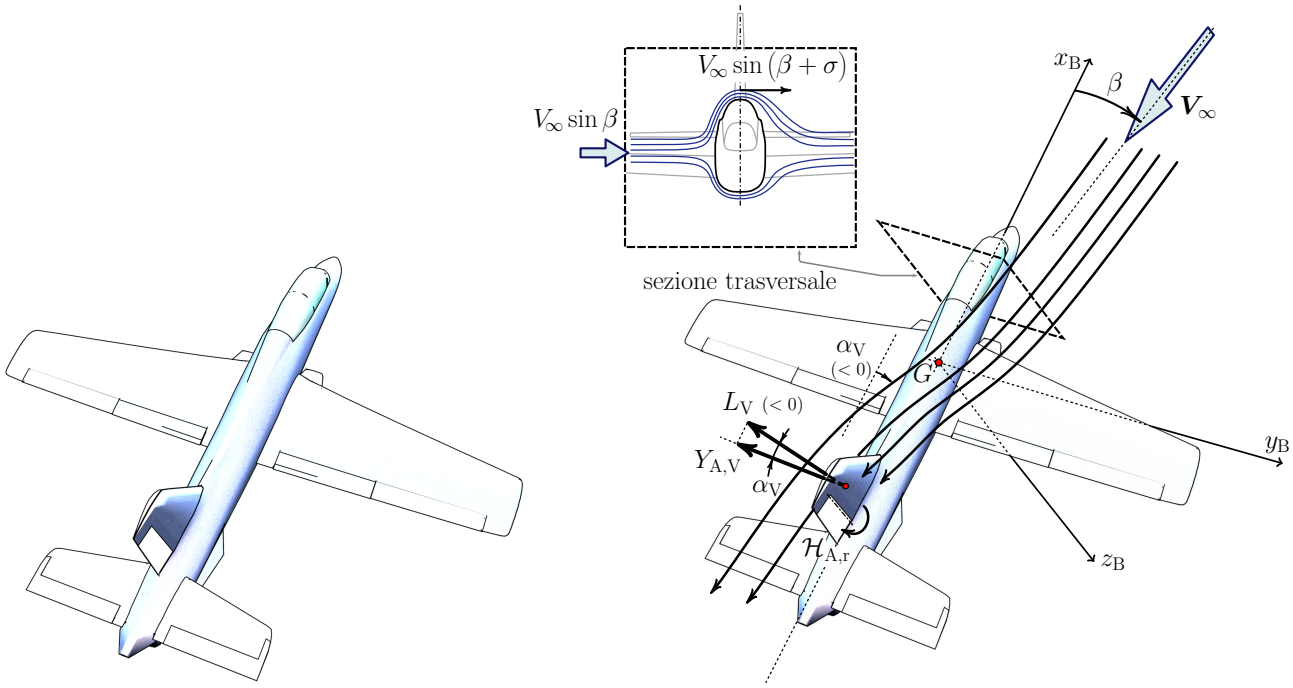

```
%% file main.tex
\documentclass{book}
\usepackage{graphicx}
% ...
\begin{document}
\input{chapter_1/chapter_1.tex}
\input{chapter_2/chapter_2.tex}
% ...
\end{document}

%% file chapter_1.tex
\chapter{Introduzione}
% ...
\input{chapter_1/fig_name1.tex}
% ...

%% fig_name1.tex
\begin{figure}[htbp]
\includegraphics[width=0.9\textwidth]%
  {chapter_1/images/name1}
\caption{...}\label{fig:Nane1}
\end{figure}
```

Figura 1: In alto, una possibile organizzazione dei file e delle cartelle di un documento complesso, con molte figure. In basso, degli estratti dai file sorgenti del documento principale `main.tex` e di alcuni file ausiliari con i comandi di inclusione delle immagini.

3.2 I pacchetti pstricks e PGF/Tikz

Va osservato che LaTeX mette a disposizione l'ambiente `picture` con il quale è possibile produrre semplici disegni o grafici, con relative annotazioni, all'interno di un documento. Si rimanda all'articolo di Caschili (2006) per una introduzione all'uso di questo strumento di disegno.

D'altra parte, tra i pacchetti di estensione creati negli anni dagli utilizzatori di tutto il mondo non



(a) Immagine bitmap di partenza.

(b) Immagine finale con annotazioni.

FIGURA 2: Esempio di annotazione di una immagine bitmap.

mancano quelli dedicati esclusivamente alla grafica, con i quali produrre una figura contestualmente alla composizione di un documento LaTeX. Due pacchetti su tutti, pstricks e PGF/Tikz, costituiscono gli strumenti di lavoro più potenti ed utilizzati.

A questi va naturalmente associato il sistema di disegno MetaPost, lo strumento di creazione di illustrazioni vettoriali che compone gli oggetti testuali sfruttando il "motore" LaTeX.

Si rimanda ai manuali d'uso di questi sistemi di disegno (HOBBY, 1989, 2008; TANTAU, 2008; VAN ZANDT, 2003; VOSS e RODRIGUEZ, 2008) per approfondimenti. La lettura a vari livelli dei manuali d'uso di questi pacchetti, tra l'altro ben scritti e ricchi di esempi, è fortemente consigliata agli utilizzatori di LaTeX, sia che abbiano la necessità di creare semplici disegni sia che vogliano invece produrre illustrazioni più complesse.

Un'ottima introduzione in italiano al pacchetto pstricks si trova nell'articolo di CASCHILI (2007). Per una galleria completa ed aggiornata di esempi d'uso di pstricks si rimanda alla sezione degli esempi del sito ufficiale del pacchetto (HERBERT VOSS, 2008). Una galleria altrettanto completa ed aggiornata di esempi d'uso del pacchetto PGF/Tikz si trova invece nelle sezioni apposite del sito mantenuto da FAUSKE (2008a,b).

Per quanto riguarda le illustrazioni, più avanti ci soffermeremo su alcune tecniche possibili per poter annotare immagini e disegni prodotti con strumenti esterni.

4 Annotazione di una illustrazione

Gli utilizzatori di LaTeX che hanno la necessità di incorporare illustrazioni o grafici in un manoscritto incorrono quasi certamente nel problema di inserire nelle figure delle annotazioni testuali, eventualmente con formule o simboli matematici, che siano coerenti con lo stile tipografico adottato per l'intero documento. Un esempio piuttosto elaborato di figura con annotazioni è dato dalla figura 2. Nella figura 2a è rappresentata l'immagine bitmap di partenza. Essa è stata prodotta usando un programma di creazione di modelli tridimensionali, utilizzando una funzione di resa "non fotorealistica" (*non photorealistic rendering*) ed esportando un fotogramma in un formato bitmap standard. La figura 2b, completa di annotazioni, è invece il prodotto finale.

La coerenza grafica e tipografica tra le figure ed il testo dovrebbe essere ricercata dal generico compositore di documenti, sia che esso utilizzi LaTeX sia che esso faccia uso di altri programmi di impaginazione.

Esistono non pochi libri e svariati tipi di altre pubblicazioni in cui il font utilizzato nel testo esplicativo delle figure è diverso da quello del testo corrente. In quei casi, una regola di buon senso vorrebbe che almeno lo stile delle annotazioni delle figure rimanesse il medesimo per ciascuna di esse. È comunque ovvio che annotazioni coerenti con lo stile del testo rendono il documento più leggibile, perché sottraggono al lettore un motivo di distrazione, e danno un'aria di professionalità

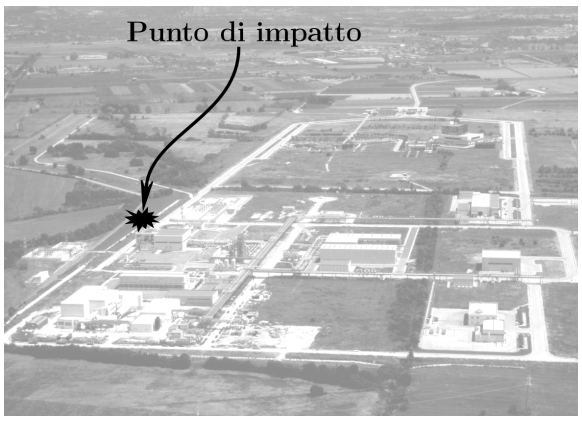


FIGURA 3: Esempio di immagine bitmap con una semplice annotazione. Immagine creata con Gimp. Il font del testo è Latin Modern (formato *Open Type*).

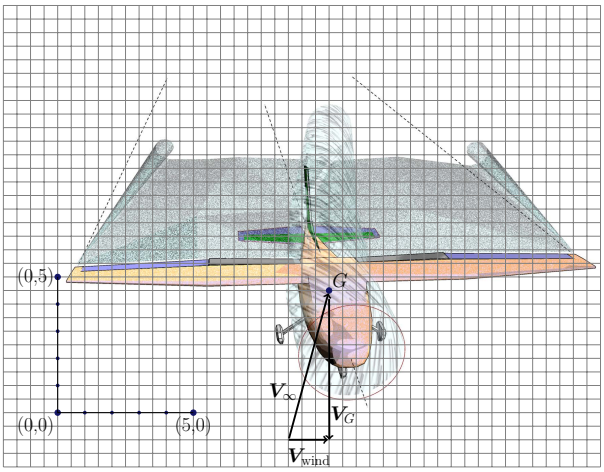


FIGURA 4: Una immagine bitmap, `airplane.pdf`, importata in un ambiente `tikzpicture` ed annotata con l'ausilio di una griglia.

(tipografica).

A questo punto, stabilita la figura da dover incorporare in un documento LaTeX, il quesito importante da porsi è: che metodo scegliere per ottenere delle annotazioni di buona qualità? Ovviamente, la risposta dipende dal tipo di figura, dall'attitudine dell'autore e dal tempo che egli ha a disposizione.

Si consideri, ad esempio, il caso in cui occorra annotare la riproduzione di una fotografia con elementi esplicativi molto semplici: una freccia (un elemento grafico) e l'annotazione "Punto di impatto", come nella figura 3. Questa figura può essere preparata, ad esempio, con un programma come Gimp (GIMP, AUTORI VARI, 2008). Se il documento principale è composto con il font Computer Modern Roman, si può utilizzare con Gimp la famiglia di caratteri Latin Modern (GUST: POLSKA GRUPA UŻYTKOWNIKÓW SYSTEMU TEX, 2008) in formato *Open Type*, liberamente scaricabile da internet.

Se il documento contiene, al più, figure semplici come la 3, il compositore non dovrà compiere grandi sforzi per produrle in accordo allo stile tipografico prescelto. Anche se dovesse esserci la necessità di inserire un'annotazione leggermente più complessa, del tipo "Punto di impatto previsto, $(x_\mathrm{p}, y_\mathrm{p})$", con programmi come Gimp si potrebbero sfruttare gli strumenti di formattazione del testo, ottenendo comodamente i simboli matematici necessari. È ovvio che questa tecnica non risulta adeguata per annotazioni con simboli matematici o formule matematiche di maggiore complessità.

Per annotare una figura con oggetti testuali più complessi esistono almeno due possibilità: l'uso di pdfLaTeX con l'estensione PGF/Tikz, oppure l'uso di LaTeX con l'estensione psfrag, ed eventualmente di pstricks, e del convertitore dvips.

4.1 Annotare con pdfLaTeX e PGF/Tikz

Questa tecnica è particolarmente conveniente se l'utente ha la necessità di apporre annotazioni ad un'immagine bitmap. I comandi chiave, da utilizzare all'interno di un ambiente `tikzpicture`, sono: `\pgftext` e `\pgfuseimage`.

La figura 4 costituisce un esempio di immagine in formato PDF importata in un ambiente `tikzpicture` ed annotata con l'ausilio di una griglia. Si riporta un frammento del codice che genera la figura 4:

```
\begin{tikzpicture}
% includo airplane.pdf
\pgftext[at=\pgfpoint{0cm}{0cm},left,base]
    {\pgfuseimage{airplane}};
\draw[help lines,step=0.5cm] (-2,-2) grid (20,15);
% ...
\node (G) at (10,4.5)
    [circle,draw=black,inner sep=2pt,fill=black]
    {};
\node at (G) [draw=none,above right] {$G$};
\draw[line width=2pt,arrows=->,shorten <=6pt]
    (G) -- +(0.01,-5.5)
    node[draw=none,inner sep=0pt,fill=none] (V1)
    {};
\draw[line width=2pt,arrows=<-]
    (V1) -- +(-1.5,0)
    node[draw=none,inner sep=0pt,fill=none] (V2)
    {};
\draw[line width=2pt,arrows=->] (V2) -- (G) {};

\path (V1)
    to node [above,pos=0.08]
    {$\boldsymbol{V}_G$} (G);
\path (V1)
    to node [below=0.2cm]
    {$\boldsymbol{V}_\mathrm{wind}$} (V2);
\path (V2)
    to node [below,pos=0.4]
    {$\boldsymbol{V}_\infty$} (G);
\end{tikzpicture}
```

Con i comandi `\pgftext` e `\pgfuseimage` è possibile importare l'immagine esterna `airplane.pdf` subito dopo l'apertura dell'ambiente `tikzpicture`. I rimanenti comandi sono delle normali istruzioni di disegno di tikz. In particolare, il comando `\draw` con la direttiva `grid` ha disegnato la griglia utile al posizionamento degli oggetti. I comandi `\path` sono serviti a posizionare delle etichette testuali lungo le rette che congiungono coppie di nodi.

È chiaro che questo sistema richiede di procedere per gradi. L'importazione dell'immagine di sottofondo e la predisposizione della griglia costituiscono il primo passo. La griglia serve a dare un'idea delle coordinate dei nodi che occorre posizionare nel disegno, ai quali agganciare le annotazioni. Per modifiche successive, si procede ad aggiustare i comandi di posizionamento dei nodi, fino ad ottenere un risultato esteticamente soddisfacente. Aiutano molto in questo caso le opzioni `xshift` e `yshift` accettate dalla direttiva `node`. Degli eleganti effetti grafici si ottengono con la direttiva `pin`. Si rimanda al manuale d'uso per approfondimenti sulle tecniche di gestione dei nodi, delle loro possibili interconnessioni e sul posizionamento delle etichette testuali.

Tra le recenti funzioni aggiunte alla versione 2.0 del pacchetto PGF/Tikz si segnalano quelle messe a disposizione dalla nuova libreria di calcolo. Caricando la libreria di funzioni nel preambolo con il comando `\usetikzlibrary{calc}` è possibile effettuare dei calcoli sulle coordinate dei nodi definiti nell'ambiente `tikzpicture`. Poter disporre di queste funzioni può rivelarsi molto utile se si devono annotare immagini di contenuto tecnico.

4.2 Annotare con LaTeX e psfrag

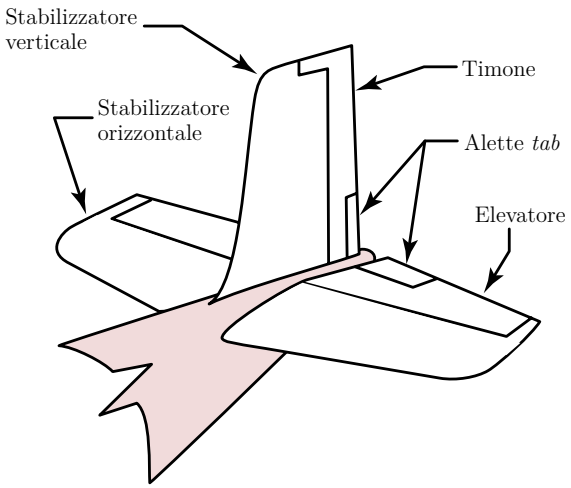


Figura 5: Esempio di immagine vettoriale. Le semplici annotazioni testuali sono state create con LaTeX e psfrag.

4.2.1 Uso semplice di psfrag

Questa tecnica risulta particolarmente efficiente se si intende lavorare con immagini genuinamente vettoriali e necessita di un'immagine di partenza in formato EPS (*Encapsulated PostScript*). Va osservato che, anche quando si dispone di un'immagine bitmap ad alta risoluzione, il carico di lavoro di un moderno personal computer nel convertirla in formato EPS e conservarla in memoria è pressoché irrilevante per l'utente. Pertanto, questa procedura di annotazione può applicarsi in generale. C'è comunque chi ritiene concettualmente assurdo convertire un formato bitmap in uno vettoriale; ma ciò a volte è indispensabile se si vogliono usare delle strategie che portano a dei buoni risultati qualitativi.

Così come per la procedura illustrata nel paragrafo precedente, si parte dal presupposto che l'utente trovi comodo e vantaggioso produrre la prima versione di un disegno o di un'immagine con uno strumento esterno. Un esempio di strumento molto valido è dato dal programma di grafica vettoriale Inkscape (Bah, 2008; Inkscape, Autori Vari, 2008). Questo software multipiattaforma è liberamente scaricabile da internet e viene distribuito con licenza GPL (*GNU General Public License*). Inkscape è espandibile dal generico utente con delle estensioni scritte in linguaggio Python. Tra queste va annoverato il *plug-in* textext, che permette di elaborare oggetti testuali mediante pdfLaTeX e di renderli nel *viewport* dell'applicazione come tracciati (Pauli Virtanen, 2008). In questo paragrafo vogliamo focalizzarci sulla possibilità di esportare con Inkscape un disegno in formato EPS, per poi trasformarlo con LaTeX mediante l'uso del pacchetto psfrag (Grant, 2008).

La figura 5 è un classico esempio di immagine vettoriale, realizzata inizialmente con Inkscape, con semplici annotazioni testuali. La figura 2 nella pagina 13, che abbiamo precedente commentato, rappresenta un esempio in cui un'immagine bitmap è stata importata in Inkscape, organizzando il disegno in un "livello" di sottofondo contenente l'immagine originale ed un livello superiore degli oggetti vettoriali (curve, cerchi, oggetti testuali).

Il flusso di lavoro si basa sulla funzionalità, semplice ma fondamentale, messa a disposizione da psfrag, che consiste nel cercare un oggetto testuale in un dato file di formato EPS e rimpiazzarlo con un testo composto con LaTeX.

Il meccanismo di modifica dell'immagine originaria, sia essa ad esempio `picture1.eps`, avviene in due passi successivi:

(*i*) Con il comando `latex main_picture1` si compila il file sorgente `main_picture1.tex`, contenente le istruzioni `\psfrag` e l'istruzione di inclusione `\includegraphics{picture1.eps}`.

(*ii*) Con il comando `dvips main_picture1` si trasforma il risultato del comando precedente dal formato DVI al formato Postscript.

Se si vuole, si può successivamente trasformare il file `main_picture1.ps` nel formato PDF tramite il comando `ps2pdf main_picture1.ps` e rinominare il risultato finale come `picture1.pdf`.

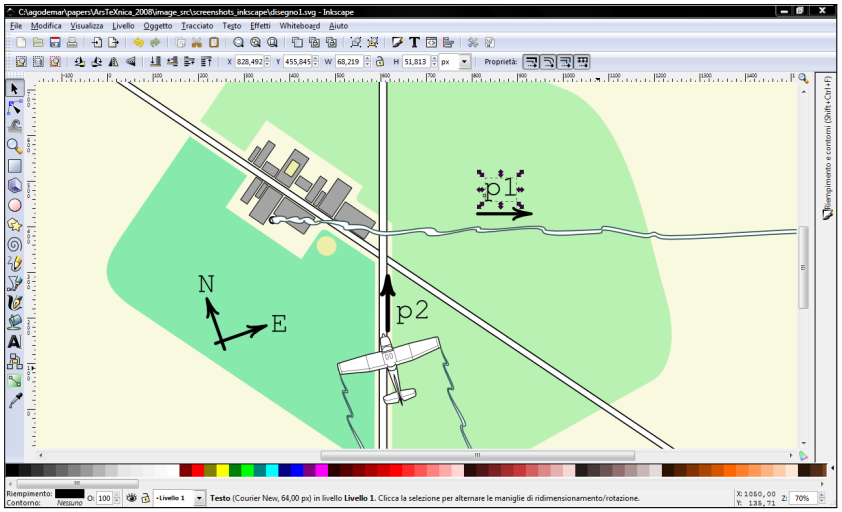
Il primo passo di compilazione con LaTeX consente all'utente di specificare cosa cercare nell'immagine `picture1.eps` e con cosa ciò deve essere sostituito. Si consideri l'esempio seguente.

```
\documentclass{article}
\usepackage[latin1]{inputenc}
```

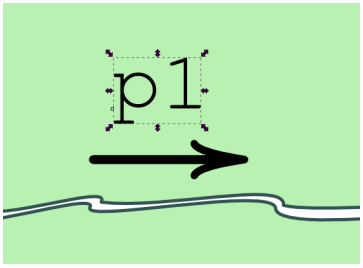
(a) L'immagine prodotta con Inkscape in formato EPS (`disegno1.eps`).

(b) Il risultato finale ottenuto attraverso il pacchetto psfrag. L'immagine in formato PDF è stata ritagliata e ridimensionata.

FIGURA 6: Annotazioni inserite in un'immagine vettoriale prodotta con Inkscape (si veda la figura 7).



(a) Una schermata della finestra di Inkscape. In lavorazione il file `disegno1.svg`, poi esportato nel file `disegno1.eps`. Accanto, l'ingrandimento di una regione del disegno. Inkscape mostra i contorni ed il punto di riferimento della linea di base dell'oggetto testuale selezionato.



(b) L'oggetto testuale "p1".

FIGURA 7: Una sessione di lavoro con Inkscape ed alcuni oggetti testuali.

```
\usepackage{lmodern,amsmath,fixmath}
\usepackage{graphicx,psfrag}
\usepackage{pstricks,pstricks-add}
\begin{document}
\pagestyle{empty}
\psfrag{p1}[b][b]{
    \rput(0,0){\rnode{VW}{}}
    \uput{4pt}[90](VW)%
        {$\boldsymbol{V}_{\!\mathrm{wind}}$}
}
\psfrag{p2}[l][l]{
    \rput(0,0){\rnode{V}{}}
    \uput{2pt}[0](V){$\boldsymbol{V}$}
}
\psfrag{N}[b][b]{
    \rput(0,0){\rnode{N}{}}
    \uput{4pt}[90](N){$x_{\mathrm{E}}$ (Nord)}
}
\psfrag{E}[l][l]{
    \rput(0,0){\rnode{E}{}}
    \uput{4pt}[0](E){$y_{\mathrm{E}}$ (Est)}
}
\includegraphics[width=1.0\textwidth]%
    {disegno1.eps}
\end{document}
```

Con questo sorgente si vuole annotare l'immagine `disegno1.eps` riportata nella figura 6a, contenente alcune caselle di testo. Si tenga presente

che un file in formato EPS è un file di caratteri ASCII, quindi è ispezionabile con un editor di testi. Se si va a ispezionare l'immagine in questione, si individueranno comandi Postscript ben precisi che posizionano gli oggetti testuali (ad esempio `p1` o `N`) all'interno del *bounding box*.

La struttura del sorgente LaTeX riportato come esempio è molto semplice. Oltre che dal preambolo, in cui viene richiamato il pacchetto psfrag, vi è una sequenza di chiamate alla macro `\psfrag`. Infine vi è la chiamata alla macro `\includegraphics` che richiama il file `disegno1.eps` da assoggettare al processo di sostituzione operato dalle chiamate a `\psfrag`.

Ciascuna delle chiamate alla macro `\psfrag` si presenta nella forma:

`\psfrag{`⟨*testo*⟩`}[`⟨*posn*⟩`][`⟨*psposn*⟩`]%`
`{`⟨*testo LaTeX*⟩`}`

Il ⟨*testo*⟩ verrà cercato e sostituito dal ⟨*testo LaTeX*⟩. Se nel file postscript originario il punto di riferimento dell'oggetto ⟨*testo*⟩ era ⟨*psposn*⟩, quello del ⟨*testo LaTeX*⟩ nel nuovo file prodotto dalla compilazione sarà ⟨*posn*⟩. Le possibili scelte al riguardo assomigliano a quelle che servono a posizionare le etichette testuali con i comandi di pstricks. Ad esempio, nel sorgente appena riportato il comando di sostituzione dell'oggetto testuale `p1` usa la coppia `[b][b]` per indicare un punto di riferimento sul bordo inferiore e centrato rispetto all'oggetto nel senso orizzontale. Se si scegliesse la coppia `[Bl][tr]` si avrebbe che il punto sulla linea di base e all'estrema sinistra del ⟨*testo LaTeX*⟩ andrebbe a capitare dove originariamente era l'estremità superiore destra dell'oggetto `p1` (`B` sta per *baseline*, `l`, `r`, `t`, `b` stanno per *left*, *right*, *top* e *bottom*).

Oltre all'indicazione dei punti di riferimento è possibile passare alla macro `\psfrag` altri parametri con i quali si potrà scalare e ruotare il ⟨*testo LaTeX*⟩. Si rimanda al manuale d'uso di psfrag (Grant e Carlisle, 2008) per ulteriori dettagli sull'uso della macro `\psfrag`.

Per avere un'idea di come è stata creata l'immagine di partenza `disegno1.eps`, si vada a guardare la figura 7, dove si è riportata la schermata di una sessione di lavoro con Inkscape. Nella figura 7a si distinguono gli oggetti testuali `N`, `E`, `p1` e `p2`. L'ingrandimento nella figura 7b mostra la selezione della casella di testo dell'oggetto `p1`.

4.2.2 Uso avanzato di psfrag e pstricks

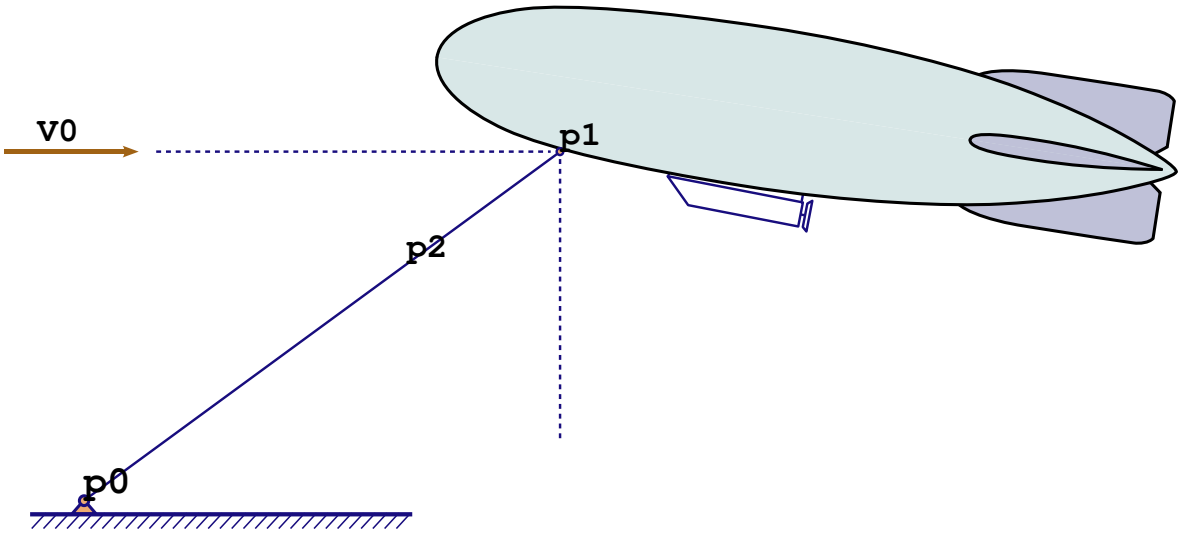
Fino a questo punto non abbiamo ancora citato i comandi del pacchetto pstricks, che pure sono stati utilizzati per produrre l'illustrazione finale della figura 6b. Abbiamo anche detto che l'uso di psfrag è legato principalmente al vantaggio di poter usare programmi di grafica dedicati alla produzione di immagini vettoriali. Qui va osservato che vi è anche la possibilità di usare in maniera avanzata il meccanismo di sostituzione del testo, utilizzando le potenti funzionalità di pstricks. In tal modo è possibile aggiungere ad un disegno degli ulteriori elementi grafici per via programmatica.

L'esempio riportato nel paragrafo 4.2.1 contiene già questo approccio ma lì i comandi pstricks sono usati in maniera limitata, soltanto per un comodo posizionamento degli oggetti ⟨*testo LaTeX*⟩. Ad esempio, si guardi il corpo delle istruzioni passate a `\psfrag` per posizionare il testo "$\boldsymbol{V}_{\mathrm{wind}}$". Si sfrutta il fatto che il punto di riferimento del ⟨*testo LaTeX*⟩ è un punto di coordinate (0,0) per eventuali comandi pstricks. Di conseguenza, viene definito il nodo `VW` mediante l'uso delle macro `\rput` ed `\rnode`. Il posizionamento dell'etichetta testuale può essere fatto in diversi modi. Nell'esempio viene utilizzato il comando `\uput`.

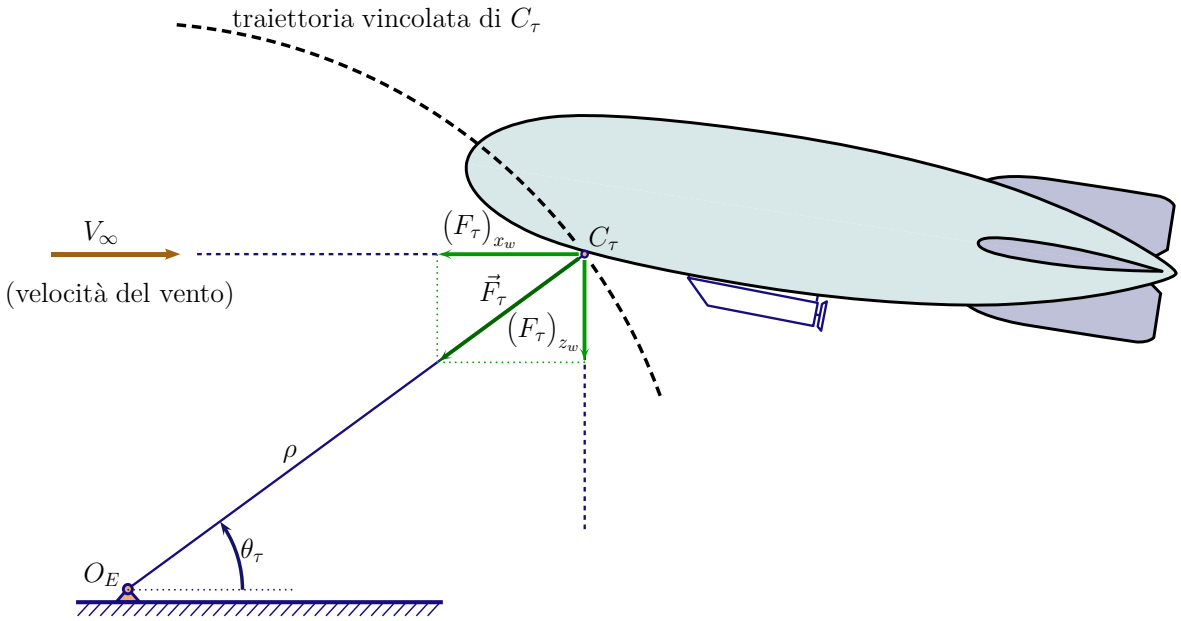
A questo punto, andiamo a considerare le illustrazioni riportate nelle figure 8a e 8b. La prima rappresenta la versione iniziale, la seconda ne costituisce la versione finale. La versione iniziale, creata con Inkscape, è ricca di elementi grafici che sarebbe stato troppo laborioso ottenere altrimenti, ad esempio con pstricks. Con l'ausilio di psfrag e di pstricks/pstricks-add è stato possibile sfruttare il posizionamento degli oggetti testuali `V0`, `p0`, `p1` e `p2` per ottenere le costruzioni geometriche presenti nella versione finale.

Ecco un estratto del sorgente che ha prodotto la figura 8b.

```
\usepackage{psfrag,pstricks,
           pst-eucl,pstricks-add}

% Versione personalizzata di \psArc,
% suggerita da Herbert Voss, si veda:
%  fr.comp.text.tex
%  "Calculer distances avec Pstricks"

\makeatletter
\def\psMyArc{\pst@object{psMyArc}}
\def\psMyArc@i{\@ifnextchar({\psMyArc@iii}%
    {\psMyArc@ii}}
\def\psarc@ii#1{\addto@par{arrows=#1}%
   \@ifnextchar({\psMyArc@iii}{\psMyArc@iii(0,0)}%
}

\def\psMyArc@iii(#1)(#2)#3#4{%
   \begin@OpenObj
     \pst@getangle{#3}\pst@tempa
     \pst@getangle{#4}\pst@tempb
     \ifx\pst@tempa\pst@tempb \else
       \pst@@getcoor{#1}%
       \addto@pscode{%
        \psMyArc@iv{#1}{#2} \psarc@v}%
       \gdef\psarc@type{0}%
       \showpointsfalse
     \fi
   \end@OpenObj%
}

\def\psMyArc@iv#1#2{%
   \pst@coor /y ED /x ED
   /r \pstDistAB{#1}{#2} def
   /c 57.2957 r \tx@Div def
   /angleA
     \pst@tempa
     \psk@arcsepA c mul 2 div
     \ifcase\psarc@type add \or sub \fi
   def
   /angleB
     \pst@tempb
     \psk@arcsepB c mul 2 div
     \ifcase\psarc@type sub \or add \fi
```



(a) Immagine originale prodotta con Inkscape in formato EPS. Si notino gli oggetti testuali `p0`, `p1`, `p2` e `V0`.



(b) Immagine finale annotata con direttive `\psfrag` e comandi di disegno di `pstricks`. Si osservi che l'arco di circonferenza è stato tracciato deducendo programmaticamente la distanza tra i punti di riferimento degli oggetti testuali `p0` e `p1` dell'immagine originale (si veda la funzione `\psMyArc` definita nel paragrafo 4.2.2).

FIGURA 8: Esempio di immagine vettoriale con diverse annotazioni.

```
   def
   \ifshowpoints\psarc@showpoints\fi
   \ifx\psk@arrowA\@empty
     \ifnum\psk@liftpen=2
       r angleA \tx@PtoC
       y add exch x add exch moveto
     \fi
   \fi}
\makeatother

\begin{document}
\pagestyle{empty}

\psfrag{V0}[bc][bc]{
    $V_\infty$
    \uput{1.2\baselineskip}[-90](0,0)
        {(velocità del vento)}
}
\psfrag{p1}[bl][bl]{
    \rput(0,0){\rnode{P1}{}}
    \uput{3pt}[45](0,0)
        {\relsize{1}$C_\tau$}
}

% point P2
\psfrag{p2}[bl][bl]{
    \rput(0,0){\rnode{P2}{}}
    \psParallelLine[linestyle=none]
        (0,10)(0,-10)(P1){10}{P1a}
    % end of Fzw component
    \psIntersectionPoint(P1)(P1a)(P2)(5,0){P1Z}
    % the Fzw component
    \ncline[linewidth=2.0pt,
            linecolor=green!60!black,
            arrows=-D>,nodesepA=3pt]{P1}{P1Z}
    \nbput[ref=Bl,labelsep=0pt,npos=0.7]
        {$\big(F_\tau\big)_{z_w}$}
    % the total F vector
    \ncline[linewidth=2.0pt,
            linecolor=green!40!black,
```

```
            arrows=-D>,nodesepA=3pt]{P1}{P2}
    \nbput[ref=Bl,labelsep=0pt,npos=0.5]
        {$\vec{F}_\tau$}
    \psParallelLine[linestyle=none]
        (10,0)(-10,0)(P1){10}{P1b}
    \psLCNode(P2){1}(0,10){1}{P2a}
    % end of Fxw component
    \psIntersectionPoint(P1)(P1b)(P2)(P2a){P1X}
    \ncline[linewidth=2.0pt,
            linecolor=green!60!black,
            arrows=-D>,nodesepA=3pt]{P1}{P1X}
    \nbput[ref=Bl,labelsep=1pt,npos=0.7]
        {$\big(F_\tau\big)_{x_w}$}
    %
    \ncline[linewidth=1.0pt,
            linecolor=green!60!black,
            linestyle=dotted]{P1X}{P2}
    \ncline[linewidth=1.0pt,
            linecolor=green!60!black,
            linestyle=dotted]{P2}{P1Z}
    % length of the cable
    \psRelNode(P1)(P2){2}{P3}
    \uput{4pt}[90](P3){$\rho$}
}

% point P0, tether fixed point
\psfrag{p0}[bl][bl]{
    \rput(0,0){\rnode{P0}{}}
    \uput{4pt}[140](P0){$O_E$}
    \rput(3,0){\rnode{P0a}{}}
    \ncline[linewidth=1.0pt,
            linecolor=blue!60!black,
            linestyle=dotted]{P0}{P0a}
    % pst-eucl related stuff
    \pstGeonode[PointSymbol=none,PointName={}]
        (P0){x}
    \pstGeonode[PointSymbol=none,PointName={}]
        (P0a){x}
    \pstGeonode[PointSymbol=none,PointName={}]
        (P2){x}
    \pstMarkAngle[MarkAngleRadius=2.1,
                  linewidth=1.8pt,
                  linecolor=blue!60!black,
                  arrows=->]{P0a}{P0}{P2}{}
    \uput{2em}[135](P0a){$\theta_\tau$}
    % constrained Ctau trajectory
    \psMyArc[linestyle=dashed,linewidth=1.8pt]
        (P0)% center
        (P1)% on the circle, defines radius
        {20}{35}% start/end angles
    \psMyArc[linestyle=dashed,linewidth=1.8pt]
        (P0)% center
        (P1)% on the circle, defines radius
        {38}{85}% start/end angles
    \uput{10.35}[70](P0)
        {traiettoria vincolata di $C_\tau$}
}
\includegraphics[width=1.0\textwidth]%
    {disegno1.eps}
\end{document}
```

L'esempio appena riportato non è banale. Esso richiede un minimo di studio, soprattutto per capire gli aspetti avanzati di programmazione pstricks utilizzati per ottenere il risultato finale. L'idea che c'è alla base è comunque semplice: attraverso l'uso di psfrag si sostituisce a ciascun oggetto testuale un nodo di pstricks; un nodo creato con una data chiamata alla macro `\psfrag` risulta "visibile" ai comandi pstricks utilizzati all'interno delle successive chiamate a `\psfrag`; i vari nodi che vengono via via definiti sono opportunamente manipolati e sfruttati per mezzo di appositi comandi pstricks, in particolare definiti dal pacchetto pstricks-add;[1] si tiene conto che in ciascuna chiamata alla macro `\psfrag` il punto di riferimento del ⟨*testo LaTeX*⟩ ha coordinate (0,0).

Alcune macro molto utili per la manipolazione dei nodi sono fornite dal pacchetto pstricks-add ed estendono quelle del pacchetto pst-node. Se ne riporta una breve descrizione:

`\psRelNode`: dati due nodi, ne definisce un terzo lungo la linea che forma un dato angolo rispetto a quella che passa per i primi due. Un caso particolare si ha quando si vuole un terzo nodo posto da qualche parte lungo la congiungente i primi due;

`\psRelLine`: analoga alla precedente, in aggiunta disegna anche una linea;

`\psParallelLine`: dati tre nodi, disegna una linea di una data lunghezza, parallela alla congiungente i primi due, che parte dal terzo nodo;

`\psIntersectionPoint`: date due rette, definite da quattro nodi, definisce un quinto nodo come loro intersezione;

`\psLNode`: dati due nodi, definisce un terzo nodo lungo la retta congiungente i primi due (caso particolare di `\psRelNode`);

`\psLCNode`: definisce un nodo, in quanto coppia di coordinate, come la combinazione lineare di altri due.

Andando a scorrere il sorgente che genera la figura 8b si troveranno esempi d'uso di alcune di queste macro.

Un altro particolare notevole di questo esempio è dato dal modo in cui è stato disegnato l'arco di cerchio tratteggiato, che è stato poi annotato con il testo "traiettoria vincolata di C_τ". Resta ben inteso che la tecnica di produzione delle illustrazioni che stiamo discutendo presuppone inevitabilmente un certo numero di cicli di compilazione; questo perché in qualche caso l'utente dovrà posizionare un testo per tentativi. Dunque, anche l'arco in questione potrebbe essere stato disegnato per tentativi, con la macro primitiva `\psarc`, fornendo come centro un nodo opportuno. Tuttavia, con questo disegno ho voluto fornire un esempio di implementazione di una macro per uno scopo particolare, non previsto dagli sviluppatori ufficiali.

Il problema è quello di tracciare un arco di circonferenza a partire da due nodi, il primo coincidente con il centro, il secondo con un punto sulla circonferenza, ed assegnando due parametri angolari per definirne gli estremi. La posizione dei due nodi, per giunta, è determinata dall'esterno, cioè dai punti del foglio in cui si sono collocati gli oggetti testuali *nella sessione di Inkscape*. La funzione chiave che permette di effettuare il disegno è `\psMyArc`, riportata nel preambolo e derivata dalla funzione `\psArc` fornita dal pacchetto pstricks-add. La funzione personalizzata si basa sulla macro di utilità `\pstDistAB` dell'eccellente pacchetto pst-eucl

1. Una buona strategia potrebbe essere quella di usare tutte le chiamate a `\psfrag`, eccetto l'ultima, per definire esclusivamente i nodi. L'ultima chiamata conterrà tutti i comandi che servono a produrre le annotazioni desiderate.

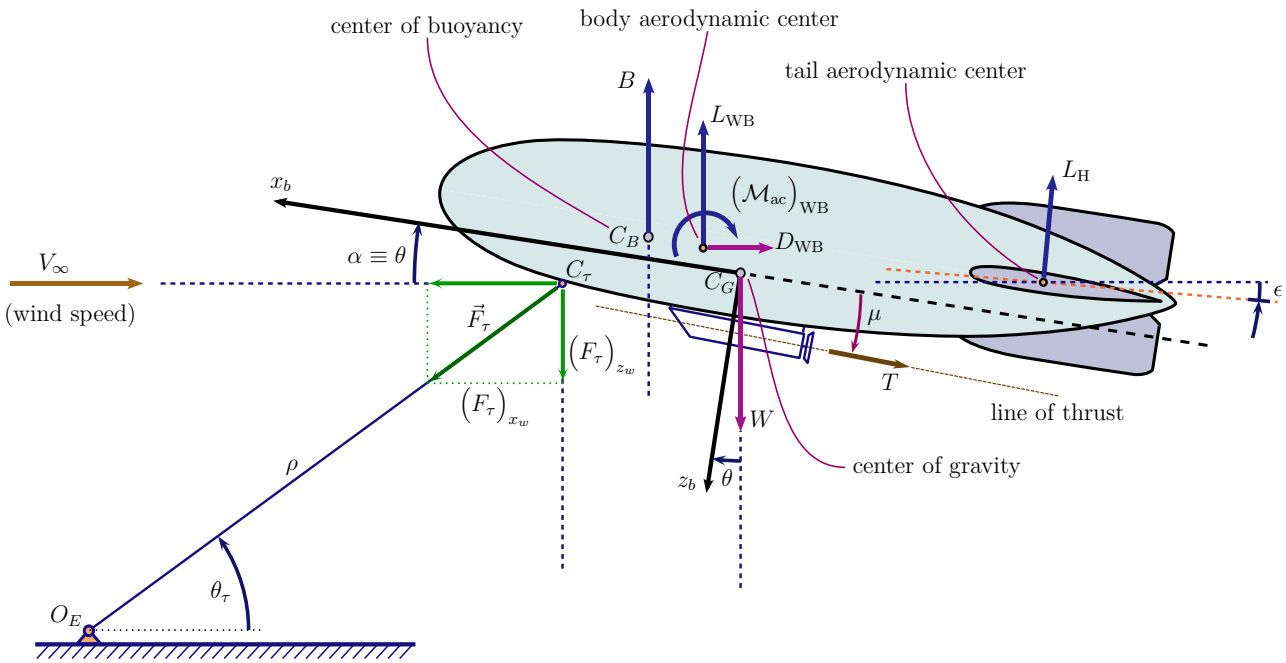


FIGURA 9: Esempio di immagine vettoriale con non poche annotazioni.

(BATTAIA, 2007; RODRIGUEZ, 2005, 2008) che permette di calcolare, a livello di linguaggio Postscript, la distanza tra due nodi.

Si segnala che anche l'arco che indica l'angolo θ_τ nella figura 8b è stato ottenuto grazie a pst-eucl, con la macro `\pstMarkAngle`.

Per finire, si riporta nella figura 9 una versione più articolata dell'illustrazione esaminata in questo paragrafo. Si possono osservare le numerose annotazioni, ottenute attraverso l'uso adeguato delle macro primitive di pstricks e delle macro dei pacchetti di estensione.

5 Annotazione di un grafico

Un altro problema che necessita di particolare attenzione quando si producono documenti tecnico-scientifici è quello della preparazione di grafici e diagrammi. Essi devono essere corredati di un testo esplicativo adeguatamente composto.

Gli autori che usano un grafico per discutere un certo argomento si concentrano, giustamente, sulla corretta rappresentazione dei dati e sulla più efficace soluzione possibile per la esposizione di un dato concetto. Tuttavia, spesso si verifica che per l'ottenimento di un grafico ci si accontenta che una data curva abbia la forma attesa. Viene giudicato eccessivo lo sforzo di dare al grafico anche un aspetto tipograficamente corretto. Cioè si accetta una penalizzazione in estetica e leggibilità.

Tipicamente, le etichette degli assi o le legenda di un grafico richiedono l'uso di simboli matematici. È altrettanto tipico il caso in cui l'autore non abbia informazioni adeguate o scelga di non documentarsi su come riprodurre un grafico sfruttando le caratteristiche di LATEX.

Si tenga presente che oggi gli autori hanno a disposizione una vasta scelta di programmi ed ambienti di lavoro con cui confezionare grafici, sia commerciali che liberamente usabili. Nell'ambito ingegneristico, vale la pena di citare due esempi su tutti: Matlab (THE MATHWORKS, 2008) e Gnuplot (GNUPLOT, AUTORI VARI, 2008). In ogni caso, un diagramma, per quanto complesso e articolato possa essere concepito, si basa su un insieme di dati numerici che esso vuole rappresentare. Nella grande maggioranza dei casi, i dati sonno conservati o esportabili in un file di testo, nel quale sono tipicamente organizzati per colonne.

Nei paragrafi successivi verranno illustrate alcune possibili soluzioni basate su LATEX o pdfLATEX, che permettono di gestire gli oggetti testuali di un grafico, coerentemente con lo stile tipografico del testo corrente.

5.1 Annotare con pdfLATEX e pgfplots

Il pacchetto pgfplots (FEUERSAENGER, 2008a) è un'estensione di PGF/Tikz relativamente recente. Il manuale d'uso di pgfplots (FEUERSAENGER, 2008b) è ricco di esempi e ciò rispecchia l'ottimo livello di qualità del pacchetto. Esso mette a disposizione l'ambiente `axis`, da usare all'interno di un ambiente `tikzpicture`. Ciascuna curva o insieme di dati da mostrare nel sistema di assi prescelto viene aggiunta con il comando `\addplot`.

La figura 10 è un esempio di grafico ottenuto con pgfplots. Ecco un estratto del codice sorgente che la genera.

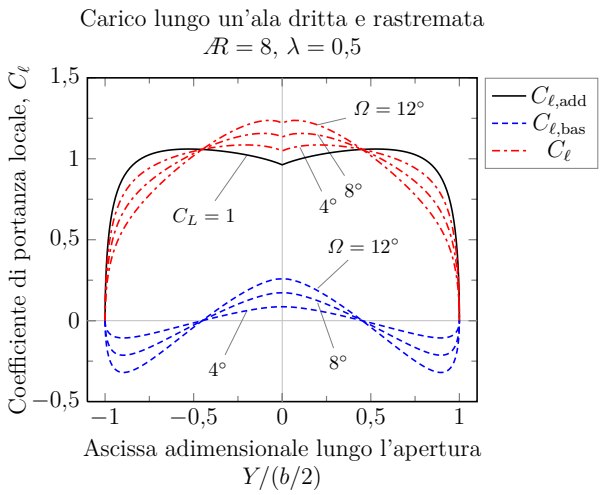


FIGURA 10: Un grafico prodotto con pgfplots. La figura è dotata di una legenda e di molte annotazioni.

```
\usepackage{lmodern}
\usepackage{amsmath,fixmath}
\usepackage{ar}
\usepackage{relsize,calc}
\usepackage{siunitx}% command \num
\sisetup{decimalsymbol=comma}
\usepackage{pgfplots}

\begin{document}

\pgfplotsset{
    every axis legend/.append style={
        at={(1.02,1)},
        anchor=north west,thin,draw=gray
        }
}
\pgfplotsset{every axis/.append style={
    font=\relsize{1}}
}

\pgfkeys{
    /pgf/number format/.cd,
    fixed, precision=2, use comma
}
\pgfplotsset{
    every axis y label/.append style={
        rotate=0, yshift=-0.1em
        }
}
\pgfplotsset{
    every axis/.append style={
        thick,
        tick style={semithick}
    }
}
\tikzstyle{every pin}=[
    fill=none, draw=none, thick,
    font=\relsize{0}
    ]

\begin{tikzpicture}%
\begin{axis}[
    xmin=-1.1,xmax=1.1,
    ymin=-0.5,ymax=1.5,
    minor tick num = 1,
    title={
        \parbox{18em}{
            \centering
    Carico lungo un'ala dritta e rastremata\\
            \centering
            $\AR=8$, $\lambda=\num{0.5}$
        }
    },
    xlabel={
        \parbox{18em}{
            \centering
    Ascissa adimensionale lungo l'apertura\\
            \centering
            $Y/(b/2)$
        }
    },
    ylabel={
       Coefficiente di portanza locale, $C_\ell$
    }
]

% Annotations
\node[
    coordinate,
    pin={[pin distance=0.8cm,
        pin edge={bend right=0}]-95:{$C_L=1$}}
    ] at (axis cs:-0.2,1.02) {};
\node[
    coordinate,pin={[pin distance=0.5cm,
        pin edge={bend right=0}]
            85:{$\Omega=\SI{12}{\degree}$}}
    ] at (axis cs:0.2,0.2) {};
% ...

\legend{%
    $C_{\ell,\mathrm{add}}$\\%
    $C_{\ell,\mathrm{bas}}$\\%
    $C_{\ell}$\\%
}%

% Curves

\addplot[color=black,thick]
    plot coordinates {
    (-1.000000, 0.000000)
    (-0.999507, 0.092636)
    % ...
    (0.999507,  0.092636)
    (1.000000,  0.000000)
};
% ...
\addplot[color=red,thick,
    dash pattern=on 4pt off 2pt on 1.5pt off 2pt]
    plot coordinates {
    (-1.000000, 0.000000)
    (-0.999507, 0.079083)
    % ...
    (0.998027,  0.156007)
    (0.999507,  0.079083)
    (1.000000,  0.000000)
};
\end{axis}
\end{tikzpicture}%
\begin{document}
```

Scorrendo il codice di questo esempio si vede che prima dell'ambiente `tikzpicture` vengono impostati gli stili della legenda, delle etichette degli assi e degli assi stessi. Ciò viene fatto secondo la prassi della recente versione 2.0 di PGF/Tikz, attraverso dei comandi `pgfplotsset`. Si osservi, in particolare, che la virgola decimale viene impostata con un comando `\pgfkeys`, dando la direttiva `use comma`. L'ambiente `axis` viene aperto passando delle opzioni che stabiliscono i limiti degli assi, il testo del titolo e quello delle etichette. La legenda è creata con il comando `\legend`. Ciascuna curva è inserita tramite il comando `\addplot` al quale vengono passate anche le direttive di stile.

La figura 11 mostra degli ulteriori esempi di grafici creati con il pacchetto pgfplots componendo le annotazioni testuali con i font Times, Math Times Professional e Bera Mono.

Come si vede dagli esempi, il pacchetto pgfplots è molto potente e permette di ottenere dei risultati di

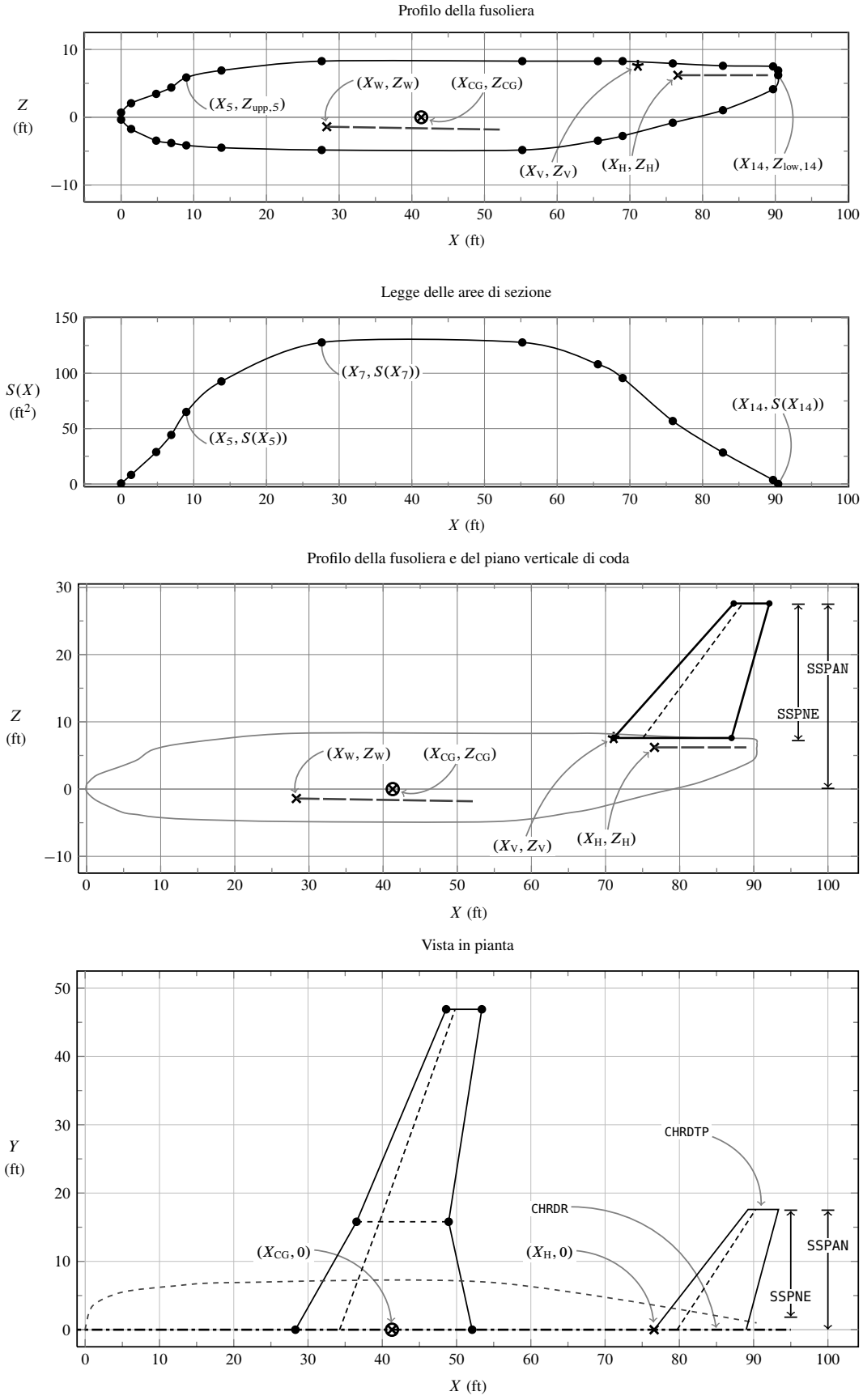


FIGURA 11: Disegni tecnici prodotti con pgfplots. Lo scopo di questi grafici è quello di spiegare il significato di alcuni dati di input di un programma di calcolo. Le annotazioni testuali sono composte con i font Times, Math Times Professional e Bera Mono.

ottima qualità. La distribuzione ufficiale contiene anche il file `matlab2pgfplots.m`, un programma di utilità in linguaggio Matlab. Con questa funzione, dopo aver creato un grafico in una finestra apposita con una sessione di lavoro di Matlab, è possibile esportarne il contenuto (curve ed altri elementi grafici compresi) in un file di comandi pgfplots pronto per essere compilato con pdfLaTeX.

L'uso di pgfplots è particolarmente indicato per disegnare curve basate su insiemi di dati di dimensioni ragionevoli. Il passaggio ad `\addplot` di un gran numero di coppie di coordinate potrebbe causare problemi di gestione della memoria.

Si rimanda al manuale d'uso del pacchetto per maggiori approfondimenti, ad esempio sui diversi modi di utilizzare file di dati esterni.

5.2 Annotare con Gnuplot (estensione Lua), pdfLaTeX e gnuplot-lua-tikz

Questa tecnica è da prendere in considerazione da coloro che preferiscono utilizzare il popolare programma Gnuplot, distribuito gratuitamente in rete e disponibile per i principali sistemi operativi (Gnuplot, Autori Vari, 2008). Gnuplot è in grado di realizzare grafici di funzioni matematiche in due e tre dimensioni ed è diventato uno dei programmi di disegno più usati poiché permette anche di riportare in grafico delle curve definite per punti, le cui coordinate sono memorizzate in un file. Tali curve possono essere disegnate con diverse possibilità di *smoothing* e con una notevole efficienza nel caso di file di dati di grandi dimensioni. Altra caratteristica molto utilizzata è la capacità di Gnuplot di produrre un'uscita sia sullo schermo sia su file (nei principali formati grafici, vettoriali e bitmap). Si rimanda alla documentazione ufficiale per approfondire le potenzialità di questo programma (Gnuplot, Autori Vari, 2008; Janert, 2008; Kawano, 2005; Williams e Kelley, 2008).

Una importante caratteristica di Gnuplot è data dalla possibilità di impostare i parametri del disegno, la provenienza e la modalità di rappresentazione dei dati attraverso un apposito linguaggio di *scripting*. Recentemente Peter Hedwig (Hedwig, 2008) ha reso disponibile una versione di questo programma di disegno ricompilata in modo da poter sfruttare le caratteristiche del linguaggio Lua. In aggiunta, con questa versione di Gnuplot egli ha messo a disposizione un "terminale Ti*k*z". Per un dato insieme di curve, il terminale produce come uscita un file di testo con le coordinate dei punti da incorporare in un ambiente `tikzpicture`.

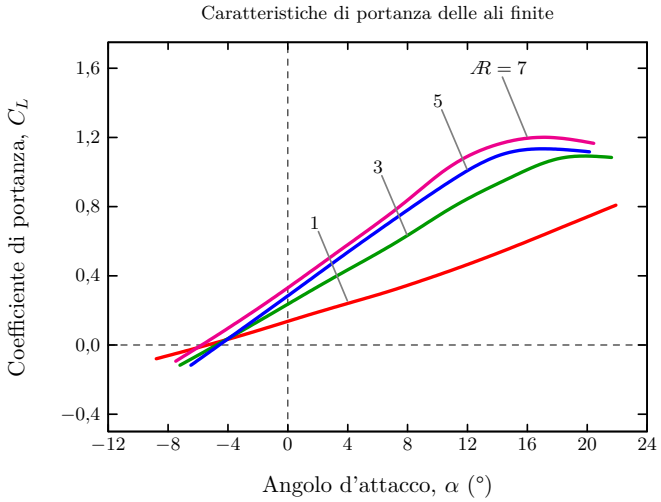
Quello che segue è un esempio di file contenente una sequenza di comandi di Gnuplot. Nel caso specifico il nome del file è `alphaCL_AR.plt`.

```
# file alphaCL_AR.plt
#--------------------
set terminal lua solid \
    nopicenvironment originreset \
    plotsize 9cm,6.5cm
set output 'alphaCL_AR.tex'

set datafile separator ","
set format x "\\num{%4.0f}"
set format y "\\num{%5.1f}"
unset key
set ytics offset 1,0
set xtics offset 0,0.5
set xtics 4
set ytics 0.4
set border lw 2
set xrange [ -12. : 24. ] noreverse nowriteback
set xlabel ''
set yrange [ -.5 : 1.75 ] noreverse nowriteback
set ylabel ''

# this sets a tikz node for later reference
set label "" at 16,1.20 font ",gp refnode,name=nA"
set label "" at 12,1.01 font ",gp refnode,name=nB"
set label "" at 8, 0.63 font ",gp refnode,name=nC"
set label "" at 4, 0.25 font ",gp refnode,name=nD"

set yzeroaxis
set xzeroaxis
set samples 200, 200
plot \
'alphaCL_AR1.csv' using 1:2:(90.) \
   smooth acsplines with lines lw 4 ,\
'alphaCL_AR3.csv' using 1:2:(90.) \
   smooth acsplines with lines lw 4 ,\
'alphaCL_AR5.csv' using 1:2:(90.) \
   smooth acsplines with lines lw 4 ,\
'alphaCL_AR7.csv' using 1:2:(90.) \
   smooth acsplines with lines lw 4
```

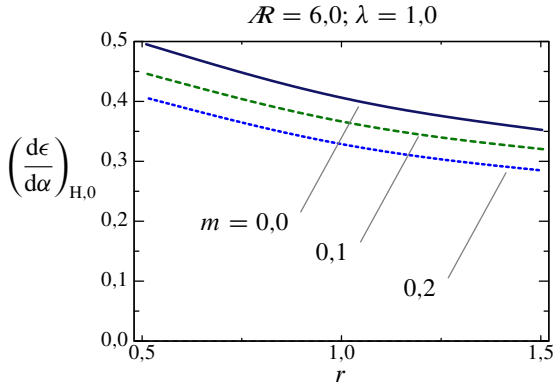
Da una *shell* dei comandi del sistema operativo, il comando `gnuplot alphaCL_AR.plt` richiede a Gnuplot di eseguire sequenzialmente, in modalità *batch*, le istruzioni su riportate. Con l'istruzione `set terminal lua` si richiede un'uscita secondo il terminale Ti*k*z. Il comando `plot` è quello più importante perché esegue effettivamente il disegno delle curve rispetto agli assi di riferimento prestabiliti. Nell'esempio specifico i dati sono letti dai file `alphaCL_AR1.csv`, `alphaCL_AR3.csv`, `alphaCL_AR5.csv` e `alphaCL_AR7.csv`. L'istruzione `set output` serve a produrre un'uscita indirizzata al file `alphaCL_AR.tex`. Con i comandi `set format` si specifica il formato dei numeri che etichettano le tacche degli assi. Si noterà la presenza del comando `\num` fornito dal pacchetto siunitx; con questa tecnica si va a controllare il formato del separatore decimale al livello del linguaggio di LaTeX. Ciò fa capire che l'uso di questi comandi deve essere accoppiato ad un sorgente LaTeX che carichi nel preambolo i pacchetti di estensione opportuni.

Il file che incorpora `alphaCL_AR.tex` potrà essere chiamato `main_alphaCL_AR.tex`. Un esempio è dato dal seguente listato con il quale si è prodotto il grafico della figura 12a.

```
% file main_alphaCL_AR.tex
%-------------------------
\documentclass[10pt,a4paper]{article}
\usepackage[T1]{fontenc}
\usepackage[latin1]{inputenc}
\usepackage{amsmath,fixmath}
\usepackage{lmodern}
\usepackage{ar}
% siunitx is better than SIunits
\usepackage{siunitx}
\sisetup{
  load=derived,
  unitsep=thin,
  valuesep=thin,
```



(a) Il testo è stato composto con il font Latin Modern. Si noti la legatura $\mathcal{AR}$ disegnata da Claudio Beccari.



(b) Il testo è stato composto con il font Times ed il font commerciale Math Times Professional. Si noti la legatura $\mathcal{AR}$ appositamente disegnata dall'autore nello stile Times.

FIGURA 12: Esempi di grafici prodotti con una versione di Gnuplot dotata di un terminale esterno per tikz in linguaggio Lua. L'uscita di Gnuplot è un file di comandi tikz necessari al tracciamento delle curve e degli assi. Questo file viene incorporato in un documento principale che carica il pacchetto gnuplot-lua-tikz e completa il grafico con eventuali comandi di disegno aggiuntivi.

```
  decimalsymbol=comma,
  expproduct=cdot,
  digitsep=thin,
  sepfour=false
}
\pagestyle{empty}
\usepackage[version=latest]{pgf}
\usepackage{pgfkeys}
\usepgflibrary{arrows,snakes,patterns}
\usepgflibrary{plothandlers}
\usepackage{gnuplot-lua-tikz}
\usepackage{relsize}
\definecolor{mydarkgreen}{rgb}{0.03,0.47,0.03}
\definecolor{mydarkblue}{rgb}{0.07,0.08,0.4}

% #1 plot number
% #2 point name
% #3 rel. x pos
% #4 rel. y pos
\newdimen\gptmpdimx
\newdimen\gptmpdimy
\newcommand{\gpsetplotcoord}[4]{%
  \pgfextractx{\gptmpdimx}{%
    \pgfpointlineattime{#3}%
      {\pgfpointanchor{gpbb south west #1}%
        {center}}%
      {\pgfpointanchor{gpbb south east #1}%
        {center}}}%
  \pgfextracty{\gptmpdimy}{%
    \pgfpointlineattime{#4}%
      {\pgfpointanchor{gpbb south west #1}%
        {center}}%
      {\pgfpointanchor{gpbb north west #1}%
        {center}}}%
  \coordinate (#2) at (\gptmpdimx,\gptmpdimy);%
}
%%
%%  e.g. \gplegendline{gp_lt_color1}{gp_lt_plot1}
%%
\newcommand{\gplegendline}[2]{%
  \tikz[gnuplot,baseline,yshift=.6ex]%
    {\draw[#1,#2](0,0)--(.6,0);}%
}
%%
%%  e.g. \gplegendmark{gp_lt_color0}{gp_mark1}
%%
\newcommand{\gplegendmark}[2]{%
  \tikz[gnuplot,baseline,yshift=.6ex]%
    {\color{#1}\gppoint{#2}{(0,0)}}
}
%%
%%  e.g. "time / t \quad\gplabelarrow"
%%
\newcommand{\gplabelarrow}{%
  \tikz[baseline]%
    {\draw[yshift=.7ex,->](0,0)--(.7,0);}%
}
\begin{document}
\begin{tikzpicture}[gnuplot]
%%
%% include plot file
%%
\input{alphaCL_AR.tex}
%%
%% "disable" \gpsolidlines
\renewcommand{\gpsolidlines}{}
\gpdashedlines
%
% change the color of default gnuplot line types
\colorlet{gp_lt_color0}{mydarkblue}%
\colorlet{gp_lt_color1}{mydarkgreen}%
%%
%% customize line types
%%
%\tikzstyle{gp_lt_plot0} = [loosely dotted]%
%\tikzstyle{gp_lt_plot1} = [loosely dashed]%
%\tikzstyle{gp_lt_plot1} = [dashed]%
%%
%% draw title
%%
\gpsetplotcoord{1}{gp title}{0.5}{1}
\node[above=.2cm,
    text badly centered,
    text width=7cm]
    at (gp title)
{Caratteristiche di portanza delle ali finite};
%%
%% draw x-label
%%
\gpsetplotcoord{1}{gp xlabel}{0.5}{0}
\node[below=.6cm]
    at (gp xlabel)
{Angolo d'attacco, $\alpha$ (\SI{}{\degree})};
%%
%% draw y-label
%%
```

```
\gpsetplotcoord{1}{gp ylabel}{0}{0.5}
\node[left=1.2cm, anchor=south, rotate=90]%
  at (gp ylabel)
{Coefficiente di portanza, $C_L$};

\tikzstyle{point_style} =
    [draw=none,
    /tikz/inner sep=0pt,
    /pgf/inner sep=0pt,
    line width=0pt,
    fill=none,fill opacity=1.0]

\node at (nA)
  [point_style,
   pin={
      [pin distance=30pt,
       inner sep=0.2ex,
       pin edge={bend right=0,
                 gray,thick}]
                 110:%
                   {\makebox[1ex][c]{$\AR=7$}}
      }]{};
\node at (nB)
  [point_style,
   pin={
      [pin distance=30pt,
       inner sep=0.2ex,
       pin edge={bend right=0,
                 gray,thick}]
                 110:%
                   {5}
      }]{};
\node at (nC)
  [point_style,
   pin={
      [pin distance=30pt,
       inner sep=0.2ex,
       pin edge={bend right=0,
                 gray,thick}]
                 110:%
                 {3}
      }]{};
\node at (nD)
  [point_style,
   pin={
      [pin distance=35pt,
       inner sep=0.2ex,
       pin edge={bend right=0,
                 gray,thick}]
                 110:%
                   {1}
      }]{};
\end{tikzpicture}
\end{document}
```

Sia la figura 12a che la figura 12b sono state create con questa tecnica. Essa si rivela molto versatile, soprattutto perché può essere automatizzata attraverso un sistema di *makefiles*. D'altra parte alcuni utenti potrebbero trovarla laboriosa rispetto all'uso di pgfplots per la necessità di dover lavorare contemporaneamente al file dei comandi Gnuplot ed al file sorgente LaTeX.

Con questa tecnica non sono stati riscontrati problemi di memoria con file di dati di grandi dimensioni.

5.3 Annotare con LaTeX, pstricks e pst-plot

Rispetto alle tecniche precedenti, questa è probabilmente la più potente. Si basa sul pacchetto pst-plot e sulle estensioni apportate da pstricks-add.

Esistono numerose opzioni e possibilità di configurazione di un grafico. Si possono disegnare i grafici di funzioni analitiche e delle loro funzioni derivate (macro `\psplot` e funzione Postscript `Derive`). Si possono disegnare funzioni definite parametricamente (macro `\parametricplot`) e grafici in coordinate polari (opzione `polarplot`). È possibile disegnare grafici attingendo i dati da file esterni con i comandi `\readdata` e `\listplot`.

Si rimanda alla documentazione dei pacchetti pst-plot e pstricks-add per approfondimenti ed esempi. La figura 13a mostra un semplice esempio di grafico di una funzione in cui le tacche dell'asse delle ascisse sono etichettate con multipli di π. La figura 13b mostra un esempio più complesso nel quale è presente anche un'illustrazione esplicativa, realizzata anch'essa con i comandi di pstricks e contestualmente al disegno del grafico sottostante. La figura 13c, infine, è adattata da uno dei tanti esempi forniti dal manuale di pstricks-add. Essa mostra l'uso della macro `\psStep` per il disegno di un insieme di rettangoli sottesi dal grafico di una funzione.

Riferimenti bibliografici

Anderson, J. D. (2004). *Introduction to Flight*. McGraw Hill Higher Education, 5th Edition.

Bah, T. (2008). *Inkscape: Guide to a Vector Drawing Program (Digital Short Cut), 2nd Edition*. Prentice Hall.

Battaia, L. (2007). *Pst-eucl, Geometria euclidea con Pstricks*. URL `http://www.batmath.it/latex/pdfs/guida_eucl.pdf`.

Caschili, M. (2006). «Semplici figure con l'ambiente picture». *ArsTeXnica*, (1), pp. 20–28. URL `http://www.guit.sssup.it/arstexnica.php`.

— (2007). «Introduzione a PSTricks». *ArsTeXnica*, (4), pp. 25–44. URL `http://www.guit.sssup.it/arstexnica.php`.

Coiro, D. P., De Marco, A. e Nicolosi, F. (2008). *Elementi di dinamica del velivolo (Titolo provvisorio)*. In via di pubblicazione.

Fauske, K. M. (2008a). Examples of TikZ and PGF version 2.00. URL `http://www.fauskes.net/pgftikzexamples/pgf-version-2/`.

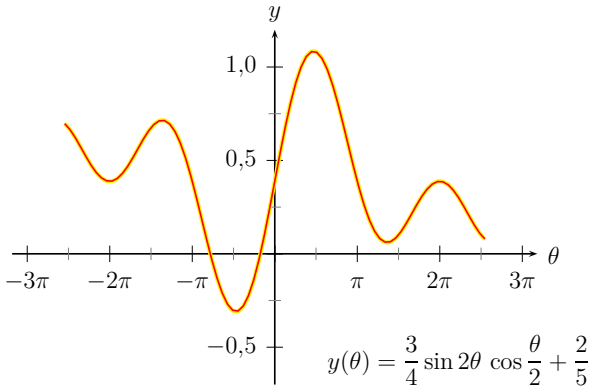
— (2008b). A PGF and TikZ examples gallery. URL `http://www.fauskes.net/nb/pgftikzexamples/`.

Feuersaenger, C. (2008a). Package PGFPlots. URL `http://www.ctan.org/tex-archive/graphics/pgf/contrib/pgfplots/`.

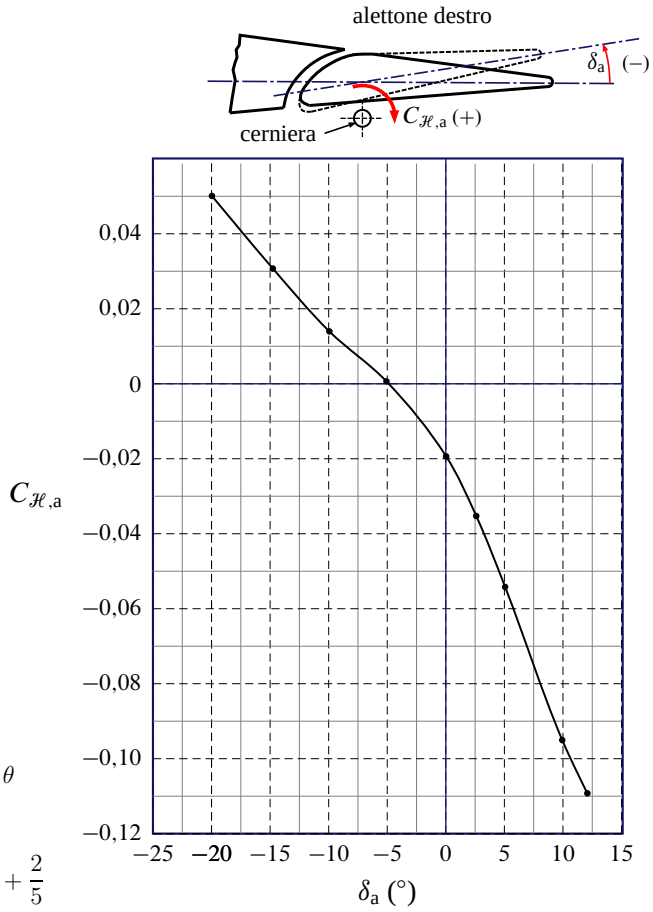
— (2008b). Manual for Package PGFPlots. URL `http://www.ctan.org/tex-archive/graphics/pgf/contrib/pgfplots/doc/latex/pgfplots.pdf`.

Gimp, Autori Vari (2008). Gimp website. URL `http://www.gimp.org/`.

```
\usepackage{pstricks,pstricks-add}
% ...
\psset{
  lly=-0.5cm,
  xAxisLabel=$\theta$,yAxisLabel=$y$,
  algebraic,plotpoints=1000
}
\begin{psgraph}[
  comma,trigLabels,subticks=2,
  dx=\psPiH,dy=0.5,Dy=0.5]{->}
  (0,0)(-5,-0.7)(5,1.2){10cm}{6cm}
\psplot[linecolor=yellow,linewidth=2pt]
  {-4}{4}{0.75*sin(2*x)*cos(x/2)+0.4}
\psplot[linecolor=red]
  {-4}{4}{0.75*sin(2*x)*cos(x/2)+0.4}
\uput{2pt}[-90](3,-0.4)
  {$\displaystyle
  y(\theta)=\frac{3}{4}\sin2\theta\,
     \cos\frac{\theta}{2}+\frac{2}{5}$}
\end{psgraph}
```

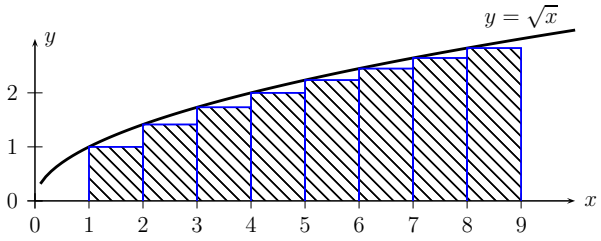


(a) Grafico di una funzione con l'ambiente psgraph. In alto è riportato il codice che serve a generare il disegno.



(b) Grafico accompagnato da un'illustrazione esplicativa. Testo composto con i font Times e Math Times Professional.

```
\begin{pspicture}(-0.5,-0.5)(10,3)
\psaxes{->}(10,3)[$x$,0][$y$,0]
\psplot[plotpoints=100,linewidth=1.5pt,algebraic]{0.1}{10}{sqrt(x)}
\psStep[linecolor=blue,fillstyle=vlines](1,9){8}{x sqrt }
\uput{1pt}[90](9,3.2){$y=\sqrt{x}$}
\end{pspicture}
```



(c) Esempio d'uso della macro \psStep.

FIGURA 13: Grafici prodotti con pstricks e pstricks-add.

GNUPLOT, AUTORI VARI (2008). Gnuplot homepage. URL http://gnuplot.sourceforge.net/.

GRANT, M. C. (2008). Package PSFrag. URL http://www.ctan.org/tex-archive/macros/latex/contrib/psfrag/.

GRANT, M. C. e CARLISLE, D. (2008). The PSfrag System, version 3. URL http://www.ctan.org/tex-archive/macros/latex/contrib/psfrag/pfgguide.pdf.

GUST: POLSKA GRUPA UŻYTKOWNIKÓW SYSTEMU TEX (2008). Latin Modern (LM) Collection of Fonts. URL

http://www.gust.org.pl/projects/e-foundry/tex-gyre/latin-modern.

Harris, R. L. (1999). *Information Graphics. A Comprehensive Illustrated Reference*. Oxford University Press, New York.

Hedwig, P. (2008). Gnuplot TikZ terminal. URL http://peter.affenbande.org/gnuplot/.

Herbert Voss (2008). Pstricks website. URL http://tug.org/PSTricks/.

Hobby, J. (1989). «A Metafont-like system with postscript output». *Tugboat, The TeX User's Group Newsletter*, (10), pp. 505–512. URL http://www.tug.org.

— (2008). *A User Manual for METAPOST*. URL http://www.tug.org/mpost.

Inkscape, Autori Vari (2008). Inkscape website. URL http://www.inkscape.org/.

Janert, P. K. (2008). *Gnuplot in Action. Understanding Data with Graphs*. Manning Publications Co.

Kawano, T. (2005). Gnuplot not so Frequently Asked Questions. URL http://t16web.lanl.gov/Kawano/gnuplot/index-e.html.

Matricciani, E. (2008). *La scrittura tecnico-scientifica*. Casa Editrice Ambrosiana.

Pantieri, L. (2008). *L'arte di scrivere con LaTeX. Un'introduzione a LaTeX 2ε*. Gruppo Utilizzatori Italiani di TeX e LaTeX.

Pauli Virtanen (2008). Textext. URL http://www.elisanet.fi/ptvirtan/software/textext/.

Rodriguez, D. (2005). *The Pst-euclide Package*. URL http://www.ctan.org/tex-archive/graphics/pstricks/contrib/pst-eucl/euclide_english.pdf.

— (2008). Pst-eucl Package. URL http://www.ctan.org/tex-archive/graphics/pstricks/contrib/pst-eucl/.

Tantau, T. (2008). *The TikZ and PGF Packages. Manual for version 2.00*. URL http://sourceforge.net/projects/pgf.

The MathWorks (2008). Matlab homepage. URL http://www.mathworks.com/.

Van Zandt, T. (2003). *PSTricks. PostScript macros for Generic TeX*. URL http://www.ctan.org/tex-archive/graphics/pstricks/base/doc/pstricks-doc.pdf. Version 97.

Voss, H. e Rodriguez, D. (2008). *Pstricks-add Additionals Macros for Pstricks*. URL http://www.perce.de/LaTeX.

Williams, T. e Kelley, C. (2008). Gnuplot – An Interactive Plotting Program. URL http://www.gnuplot.info/docs/gnuplot.pdf.

▷ Agostino De Marco
Università degli Studi di Napoli "Federico II"
Dipartimento di Ingegneria Aerospaziale
agostino dot demarco at unina dot it