

Numero 6
Ottobre
2008

ArsT_EXnica

Rivista italiana di T_EX e L^AT_EX

Atti del G_UIT*meeting* 2008

G_UIT

<http://www.guit.sssup.it/arstexnica>



G_UI_T – Gruppo Utilizzatori Italiani di T_EX

ArsT_EXnica è la pubblicazione ufficiale del G_UI_T

Comitato di Redazione

Gianluca Pignalberi – *Direttore*

Claudio Beccari

Fabiano Busdraghi

Riccardo Campana

Massimo Caschili

Gustavo Cevolani

Massimiliano Dominici

Andrea Fedeli

Enrico Gregorio

Lapo Mori

Ottavio Rizzo

Emiliano Vavassori

Emanuele Vicentini

Raffaele Vitolo

Emanuele Zannarini

ArsT_EXnica è la prima rivista italiana dedicata a T_EX, a L^AT_EX ed alla tipografia digitale. Lo scopo che la rivista si prefigge è quello di diventare uno dei principali canali italiani di diffusione di informazioni e conoscenze sul programma ideato quasi trent'anni fa da Donald Knuth.

Le uscite avranno, almeno inizialmente, cadenza semestrale e verranno pubblicate nei mesi di Aprile e Ottobre. In particolare, la seconda uscita dell'anno conterrà gli Atti del Convegno Annuale del G_UI_T, che si tiene in quel periodo.

La rivista è aperta al contributo di tutti coloro che vogliano partecipare con un proprio articolo. Questo dovrà essere inviato alla redazione di ArsT_EXnica, per essere sottoposto alla valutazione di recensori. È necessario che gli autori utilizzino la classe di documento ufficiale della rivista; l'autore troverà raccomandazioni e istruzioni più dettagliate all'interno del file di esempio (`.tex`). Tutto il materiale è reperibile all'indirizzo web della rivista.

Gli articoli potranno trattare di qualsiasi argomento inerente al mondo di T_EX e L^AT_EX e non dovranno necessariamente essere indirizzati ad un pubblico esperto. In particolare tutorials, rassegne e analisi comparate di pacchetti di uso comune, studi di applicazioni reali, saranno bene accettati, così come articoli riguardanti l'interazione con altre tecnologie correlate.

Di volta in volta verrà fissato, e reso pubblico sulla pagina web, un termine di scadenza per la presentazione degli articoli da pubblicare nel numero in preparazione della rivista. Tuttavia gli

articoli potranno essere inviati in qualsiasi momento e troveranno collocazione, eventualmente, nei numeri seguenti.

Chiunque, poi, volesse collaborare con la rivista a qualsiasi titolo (recensore, revisore di bozze, grafico, etc.) può contattare la redazione all'indirizzo:

arstexnica@sssup.it.

Nota sul Copyright

Il presente documento e il suo contenuto è distribuito con licenza © Creative Commons 2.0 di tipo “Non commerciale, non opere derivate”. È possibile, riprodurre, distribuire, comunicare al pubblico, esporre al pubblico, rappresentare, eseguire o recitare il presente documento alle seguenti condizioni:

Ⓒ **Attribuzione:** devi riconoscere il contributo dell'autore originario.

Ⓓ **Non commerciale:** non puoi usare quest'opera per scopi commerciali.

Ⓔ **Non opere derivate:** non puoi alterare, trasformare o sviluppare quest'opera.

In occasione di ogni atto di riutilizzazione o distribuzione, devi chiarire agli altri i termini della licenza di quest'opera; se ottieni il permesso dal titolare del diritto d'autore, è possibile rinunciare ad ognuna di queste condizioni.

Per maggiori informazioni:

<http://www.creativecommons.com>

Associarsi a G_UI_T

Fornire il tuo contributo a quest'iniziativa come membro, e non solo come semplice utente, è un presupposto fondamentale per aiutare la diffusione di T_EX e L^AT_EX anche nel nostro paese. L'adesione al Gruppo prevede una quota di iscrizione annuale diversificata: 30,00 € soci ordinari, 20,00 (12,00) € studenti (junior), 75,00 € Enti e Istituzioni.

Indirizzi

Gruppo Utilizzatori Italiani di T_EX:

c/o Ufficio Statistica

Scuola Superiore Sant'Anna

Piazza Martiri della Libertà 33

56127 Pisa, Italia.

<http://www.guit.sssup.it>

guit@sssup.it

Redazione ArsT_EXnica:

<http://www.guit.sssup.it/arstexnica/>

arstexnica@sssup.it

Codice ISSN 1828-2369

Stampata in Italia

Pisa: 15 Ottobre 2008

ArsT_EXnica

Rivista italiana di T_EX e L^AT_EX

Numero 6, Ottobre 2008

Editoriale	
<i>Gianluca Pignalberi, Massimiliano Dominici</i>	3
Introduction to METAPOST	
<i>Klaus H^oppner</i>	5
Gestione avanzata delle figure in L ^A T _E X	
<i>Agostino De Marco</i>	10
Una tabella che fa calcoli	
<i>Roberto Giacomelli</i>	28
Bibliografia e L ^A T _E X	
<i>Lapo F. Mori</i>	37
Introduzione allo stile ClassicThesis	
<i>Lorenzo Pantieri</i>	52
T _E X Live 2008 and the T _E X Live Manager	
<i>Norbert Preining</i>	67
L ^A T _E X e grammatiche	
<i>Gianluca Pignalberi, Enrico Bini</i>	76
longmedal: un pacchetto per medaglioni divisi su più pagine	
<i>Agostino De Marco, Massimiliano Dominici</i>	86
Specifying Translated Works in Bibliographies	
<i>Jean-Michel H^ufflen</i>	93

Gruppo Utilizzatori Italiani di T_EX

Editoriale

Gianluca Pignalberi, Massimiliano Dominici

Puntuali, come ogni ottobre, *ArSTeXnica* divulga il *QJrmeeting*. Vi apprestate infatti a leggere il numero contenente gli atti del meeting 2008. Prima di parlarne, permettetemi un po' di aggiornamenti. All'inizio del mio impegno "editoriale" vi avevo promesso lente e costanti miglierie: spero di stare mantenendo la promessa. Da questo numero è presente il codice a barre (lo so, non è generato da *LATeX*, ma Barcode è un prodotto GNU, e per di più italiano, e ci sono così affezionato).

Quando stavo preparando il numero mi sono ricordato dell'esistenza di diversi file da modificare a mano, per rendere coerente il contenuto della rivista. Massimiliano ha fatto un ottimo lavoro, e ha avuto pazienza e precisione nel fare sempre tutte le modifiche. Purtroppo non ha tenuto conto (non poteva saperlo) della pigrizia e della scarsa metodicità del suo successore. Da un po' di tempo avevo in mente l'idea di uno script al quale demandare il lavoro "sporco" e ripetitivo. Non avrei resistito a realizzare un altro numero nel modo usuale (lo script era in programma entro il numero 7), e ho rubato un po' di tempo alla rivista per farlo: già per questo numero ho usato due script bash. Il primo crea la directory del nuovo numero e ci copia dentro i file necessari a realizzare *ArSTeXnica*; il secondo genera il codice a barre, mette i dati giusti al posto giusto e compila il numero. A me non rimane che aprire il master e scriverci dentro i comandi degli autori, i pacchetti aggiuntivi e le inclusioni degli articoli. E sperare per il meglio (ma temere il peggio: l'uso di listings ha portato notevoli problemi, e questo mi ha "costretto" a fornire una funzione di *restore* al suo programmatore. Spero che la trovi utile e la aggiunga, magari in forma migliore, al pacchetto).

Per i prossimi numeri gli script saranno ancora più evoluti; in previsione di ciò chiederò il contributo degli autori. Alcuni articoli qui pubblicati mi hanno offerto lo spunto. Troverete a breve ulteriori notizie sul sito: controllatelo prima di consegnare i prossimi articoli.

È finalmente ora di parlare dei contenuti del meeting. Si apre con Klaus Höppner, a fornire un'introduzione a *METAPOST*, tanto breve quanto preziosa; Agostino De Marco ci parla della gestione avanzata delle figure in *LATeX*. Io sono rimasto esterrefatto della bellezza dei risultati. Roberto Giacomelli ci spiega come realizzare e integrare una tabella "autocalcolante" all'interno di un documento; Lapo Mori invece ci ammannisce una disamina sulla bibliografia in documenti *LATeX*. Lorenzo Pan-

tieri discute approfonditamente, denotando ottime conoscenze, la classe *ClassicThesis*; Norbert Preining presenta il nuovo sistema multiplatforma di installazione di *T_EX Live*. Gianluca Pignalberi si (ci?) diverte con i vari strumenti di composizione di grammatiche, automi e affini, mentre Massimiliano Dominici e Agostino De Marco scrivono un pezzo molto importante nell'economia della composizione di riviste: il loro pacchetto è in grado di generare medaglioni suddivisi su più pagine. Infine, anticipando un futuro prossimo e necessario, Jean-Michel Huffer ci mostra come ottenere bibliografie multilingua in generale, e comprendenti lavori in traduzione in particolare.

Non ci rimane che augurarci buon meeting (per chi partecipa, e prende gli atti in loco), o buona lettura per chi avrà la rivista differita.

Discorso di benvenuto

Come al solito, il convegno nazionale, giunto quest'anno alla sua quinta edizione, è il momento più adatto per tracciare un bilancio dell'anno trascorso. Dopo due anni in cui le novità sono state numerose e le trasformazioni, in alcuni casi, anche profonde (lancio della rivista, nuovo statuto, rinnovo della dirigenza e cambio alla direzione di *ArSTeXnica*), l'ultimo anno ha visto una fase di consolidamento del Gruppo, con un aumento considerevole dei soci iscritti. La politica di istituire quote di iscrizione differenziate per diverse tipologie di soci sembra aver dato, in questo campo, buoni risultati o, quanto meno, non aver ostacolato una tendenza già in atto. Il processo di ricambio del Consiglio Direttivo, avviato un anno fa con l'approvazione del nuovo statuto, prosegue, garantendo l'apporto di energie sempre nuove, anche se sarebbe auspicabile una maggiore partecipazione a livello di candidature e di espressione del voto.

Gode di buona salute anche il Forum, che ho potuto apprezzare particolarmente in questo mio primo anno di presidenza del gruppo, in cui è stato mio compito monitorarlo. È confortante notare come, accanto ad utenti che, com'è naturale, usano il Forum solo per un periodo limitato, ve ne siano altri che invece passano gradualmente dallo status di 'discenti' a quello di 'docenti', una volta acquisita la necessaria esperienza.

Delle novità riguardanti *ArSTeXnica* (novità già realizzate e novità 'incombenti') ha già parlato, qui sopra, Gianluca, che è ovviamente il più adatto a farlo. Non aggiungerò quindi altro, se non che trovo estremamente apprezzabile lo spirito propositivo del direttore in carica.

Oltre alla rivista, il G_JT ha pubblicato quest'anno due monografie: «Introduzione all'arte della composizione tipografica con L^AT_EX» a cura di Claudio Beccari e «L'arte di scrivere con L^AT_EX» di Lorenzo Pantieri. Si tratta di due importanti contributi alla diffusione della conoscenza di L^AT_EX in Italia, e il Gruppo non poteva non fornire il suo supporto, sotto forma di discussioni, suggerimenti di migliorie e assistenza logistica alla pubblicazione. Della seconda, alcune copie della quale sono disponibili al meeting, l'autore sta curando una riduzione da distribuire in formato elettronico come guida di base per il principiante.

Naturalmente, non tutto è rose e fiori. Alcune delle iniziative annunciate all'ultimo meeting stentano ancora a decollare. Per alcune, com'è il caso dei gruppi locali, dobbiamo interrogarci se esista già la massa critica perché possano realizzarsi; per altre, come nel caso di L^AT_EXpedia, la pausa è dovuta invece a una riflessione sulle scelte tecniche da farsi. In entrambi i casi, e in altre situazioni analoghe, è necessario l'impegno di tutti i soci del gruppo per superare il momento di stallo. Colgo

l'occasione per ricordare che siamo ancora alla ricerca di una persona che si occupi in pianta stabile dell'amministrazione del Forum.

Per finire, è doveroso rivolgere un ringraziamento speciale alla Scuola Superiore Sant'Anna, e in particolar modo al Prof. Giulio Bottazzi del Laboratorio di Economia e Management, il cui supporto è stato indispensabile alla buona riuscita del convegno.

Termino questa breve introduzione porgendo a tutti il mio benvenuto al G_JT*meeting2008*.

- ▷ Gianluca Pignalberi
Direttore di *ArsT_EXnica*
g dot pignalberi at
freesoftwaremagazine
dot com
- ▷ Massimiliano Dominici
Presidente del Gruppo
Utilizzatori Italiani di
T_EX
mlgdominici at
interfree dot it

A short introduction to METAPOST

Klaus Höffner

Abstract

METAPOST is a program strongly related to Knuth's original METAFONT. It uses nearly the same graphics language and syntax, but instead of bitmap fonts it produces PostScript output. So it can be used to create high quality graphics. In METAPOST, points and paths may be described by a set of linear equations that are solved by the program. Thus, METAPOST becomes unique among other tools like PSTricks or commercial applications (e.g., Corel Draw). Additionally, the PostScript subset created by METAPOST can be interpreted by pdf \LaTeX . So METAPOST figure can be directly included with e.g., the standard **graphics** package, while normal EPS images have to be converted first to be usable with pdf \LaTeX .

Sommario

METAPOST è un programma fortemente legato al METAFONT originale di Knuth. Usa pressoché uguali sintassi e linguaggio orientato alla grafica, ma produce un output PostScript invece di font bitmap. Dunque può essere usato per creare grafici di alta qualità. In METAPOST, punti e linee (percorsi) possono essere descritti da un insieme di equazioni lineari risolte dal programma. Dunque, METAPOST sventa unico tra strumenti come PSTricks o applicazioni commerciali (ad esempio Corel Draw). In più, il sottoinsieme di PostScript generato da METAPOST può essere interpretato da pdf \LaTeX . Quindi le figure di METAPOST possono essere incluse direttamente tramite, ad esempio, il pacchetto standard **graphics**, mentre le normali immagini EPS devono essere convertite prima di essere usate con pdf \LaTeX .

1 History

When Knuth developed \TeX , he also created a set of new fonts, Computer Modern. For this, he created his own font description language and the program METAFONT, that converts a METAFONT source into a bitmap, usually stored in a file with the extension .pxl or more often .pk. The major feature of METAFONT is that paths may be described with a set of linear equations, that determines how the single points of the path are related, and this equation set is solved by the METAFONT program. Additionally, Knuth extensively used parameters within these equations, so different font series (e.g., bold and medium) could be produced from the same equations by changing some parameters.

John D. Hobby created METAPOST as a system using (nearly) the same programming language, but with PostScript output. It was presented first in TUGboat (HOBBY, 1989), while the first public versions were released in the early 1990s. Some new features were added to the Meta language, e.g., the ability to include stuff typeset by \TeX into a METAPOST drawing (something, that wasn't needed in METAFONT for creating glyphs of a font, but is very useful to put text labels into graphics).

Meanwhile METAPOST is maintained by the METAPOST team, with Taco Hoekwater as chief developer. Since then, many improvements were made. For the future, they plan to release MPlib, a component library that contains the METAPOST engine and can be reused in other applications.

2 Basics

As mentioned before, METAPOST defines its own programming language. It consists of the following elements:

- points,
- pens,
- paths,
- numbers,
- colors (originally RGB only, meanwhile CMYK is supported)

Points are normally named by the letter z , represented by a pair (x, y) .

Paths may contain geometrical elements (e.g., **fullcircle**) or may consist of points that are connected by lines or Bézier curves.

Colors are tuples of three (RGB) or four (CMYK) numbers.

For a short example let's have a look on the following example:

LISTING 1: 1st example

```
filenametemplate "%j-%3c.mps";
beginfig(1);
pickup pencircle scaled 1bp;
draw origin--(1cm,2cm)
    ..(0.5cm,1cm)..cycle;
endfig;
end
```

It shows that each METAPOST figure is put between **beginfig** and **endfig**, with a number identifying the figure. So, a METAPOST source may contain several figures. Originally, when processing the

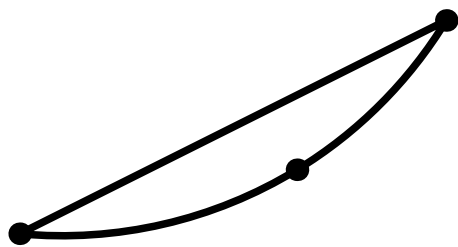


FIGURE 1: Example figure, as defined in listing 1

source (e.g., `ex.mp`) with METAPOST (`mpost ex`), the figure numbers were used as file extension for the resulting PostScript files. In later releases, the command `filenametemplate` was introduced, that uses a syntax like the `printf` command in C. In the example above, we would get a PostScript file with the name `ex-001.mps` (and if we add a figure with number 2, the PostScript output would be written into `ex-002.mps`). Since pdfTEX recognizes files with the extension `.mps` as METAPOST output, the file could be easily used in a LATEX document with `\includegraphics{ex-001.mps}` and the document may be processed both by pdfTEX and, in the original workflow, by compiling to DVI and using dvips.

As in C, all statements may span multiple lines and are finished by the “;” character.

The example figure itself shows a straight line from the origin to the point (1cm, 2cm). Then, the path is closed by a Bézier curve via the point (1.3cm, .3cm). For drawing, a round pen with diameter of one PostScript point is used. METAPOST knows the same units as TEX, like `bp` for PS points, `cm`, `mm`. The result is shown in fig. 1.

3 Defining points by linear equations

While there is nothing exciting about the first example, we will see what makes METAPOST special in a moment. Assume you want to draw a simple rectangle (fig. 2). Then you know it consists of four corners (e.g., with the lower left one in the origin), that we will describe by the following equations:

LISTING 2: Rectangle

```
path p[];
z0 = origin;
x0 = x3;
x1 = x2;
y1 = y0;
y3 = y2;
x1-x0 = 3cm;
y3-y0 = 2cm;
p0 = z0--z1--z2--z3--cycle;
fill p0 withcolor blue;
draw p0 withpen pencircle scaled 1bp;
```

You see, all corners besides z_0 aren't defined directly as (x, y) pairs but described by their re-

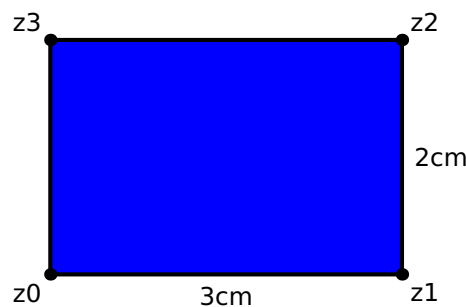


FIGURE 2: Rectangle, resulting from code in listing 2

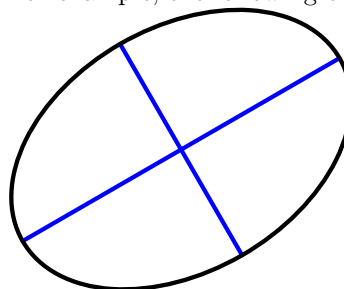
lations. While describing a rectangle with linear equations seems like a bit of overkill, this METAPOST feature becomes really powerful for the construction of complex paths.

4 Transformations of paths

METAPOST supports the following transformations of paths:

- Translations
`p0 shifted (x1, x2)`
- Rotation
`p0 rotated alpha`
- Scaling (in both directions, or in x or y direction individually)
`p0 scaled factor`
`p0 xscaled xfactor`
`p0 yscaled yfactor`
- Slanting
`p0 slanted alpha`

For example, the following ellipsis



is the output of the code

```
draw fullcircle xscaled 3cm
    yscaled 2cm rotated 30;
```

5 Intersection points

Finding the intersection points of paths is another nice METAPOST feature.

Assume you have a triangle. Theory says, if you draw three lines, each of them from one corner of the triangle to the mid point of the opposite side, all these lines will intersect at the same point.

The following code shows how this can be demonstrated in a METAPOST drawing:

LISTING 3: Triangle 1

```

pickup pencircle scaled 1bp;
path p[];
z0 = origin;
z1 - z0 = 3cm*right;
z2 - z0 = 2.7cm*dir(40);
p0 = z0--z1--z2--cycle;
p1 = .5[z0,z1]--z2;
p2 = .5[z1,z2]--z0;
p3 = .5[z2,z0]--z1;
draw p1 withcolor blue;
draw p2 withcolor blue;
draw p3 withcolor blue;
draw p1 intersectionpoint p2
      withpen pencircle scaled 3bp;
draw p0;

```

This code is rather simple. It consists of three parts.

First, the three points $z_0 \dots z_2$ are defined and path p_0 is defined as the triangle with these points as corners.

Second, the paths $p_1 \dots p_3$ are defined. Each consists of a line from one corner to the mid point of the opposite side. This may be easily expressed in METAPOST, since e. g., the statement `.5[z1,z2]` is just the point on half the line from z_1 to z_2 .

Finally, after drawing all the paths defined above, we mark the intersection point of p_1 and p_2 . This is directly given by the command `p1 intersectionpoint p2`. It may be a bit more complicated if two paths have more than one intersection point.

The result of this drawing is shown in fig. 3.

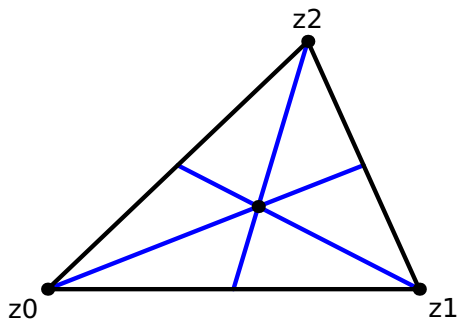


FIGURE 3: Construction of a triangle

6 Whatever it is ...

Coming back to the triangle in the latest example, another interesting task is the following: draw a line from one corner that is orthogonal to the opposite side.

Let's assume we want to draw the line from point z_2 , so the opposite side is just the base line of our triangle. Now we know two things:

1. The line is orthogonal to the base line (connection of z_0 and z_1).

2. The starting point is in z_2 , the end point shall be on the base line.

This may be directly expressed in METAPOST:

LISTING 4: Triangle 2

```

z10-z2 = whatever*((z1-z0) rotated 90);
z10 = whatever[z0,z1];

```

In the code above the end point of the line is named z_{10} .

Here we see both conditions listed before: first, the distance vector between z_{10} and z_2 is given by the distance vector between z_1 and z_0 (i. e. the base line), rotated by 90 degrees, *scaled by an arbitrary factor*.

Second, z_{10} is located *somewhere* on the line defined by the points z_0 and z_1 .

In both cases, I used a numerical value named **whatever**. This may become an arbitrary number. In fact, the value may change from statement to statement, since the variable **whatever** is encapsulated per statement.

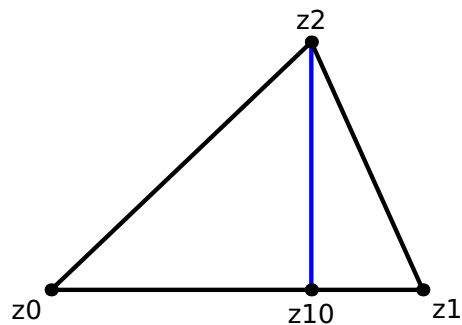


FIGURE 4: Triangle 2

As shown in fig. 4, METAPOST finds the correct position for z_{10} as starting point of a perpendicular line to the base line, with z_2 as end point.

7 Time variables

A path in METAPOST may be imagined as the travel of a vehicle. Paths are parameterized by a time variable (what might be a bit misleading, since of course the drawing is static). So a path has a start and end time, and any point is correlated to a time in between (and vice versa).

Here is some example, where time variables are used:

LISTING 5: Time variables and subpaths

```

pickup pencircle scaled 1bp;
path p[];
p0 = origin{up}..(3cm,2cm);
p1 = (-5mm,2cm)--(3cm,5mm);
draw p0 dashed withdots;
draw p1 dashed withdots;
(t0,t1) = p0 intersectiontimes p1;
draw subpath (0,t0) of p0
      -- subpath (t1,length(p1)) of p1;

```

We have two paths, p_0 and p_1 : a Bézier curve from lower left to upper right, and a straight line from upper left to lower right, drawn with dotted lines.

Now we are not only interested in the intersection point of these paths, but we want to combine the subpath of p_0 before the intersection point with the subpath of p_1 after this point, drawn with a solid line in fig. 5.

In this case, we need the time values of both paths in the intersection point. For this, the statement `p0 intersectiontimes p1` is used. The result of this is a pair (in fact a point), with the time value of p_0 in the intersection point as first part (x -part) and the time value of p_1 as second part (y -part).

As soon as these time values are known, the desired path is constructed using `subpath`.

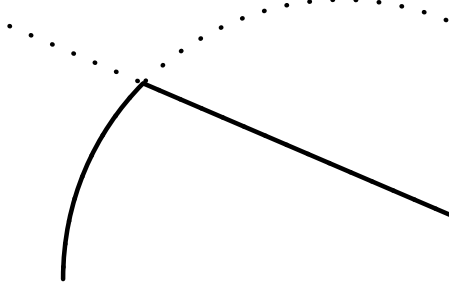


FIGURE 5: Using time variables and subpaths

8 Text and Labels

METAPOST allows to place labels into a figure. In the simplest form, the text may be included directly, without any typesetting done by TEX:

```
defaultfont := "ptmr8r";
defaultscale := 1.2;
label("this is a label",z0);
```

It will just add the text commands to write the label text in 12pt Times Roman (since a font scaling factor of one refers to 10pt) into the PostScript code. Please note that `:=` is used in the code above, since new values are assigned to the variables, while `=` is used in linear equations.

The action of placing a label at z_0 in the example is rather straightforward. The `label` command centers the label at the given point. In many cases, a suffix is appended to the `label` command to define how the label is placed in relation to the given point, i.e. `top`, `bot` (bottom), `lft` (left), `rt` (right) or `ulft`, `llft`, `urt`, `lrt` (e.g., `ulft` means upper left and `lrt` means lower right). The `label` command may be replaced by `dotlabel`, that draws a dot at the given point in addition to the label. For example, the code `dotlabel.urft("this is a label",z0)` draws a dot at z_0 and places the given text in upper right direction from z_0 .

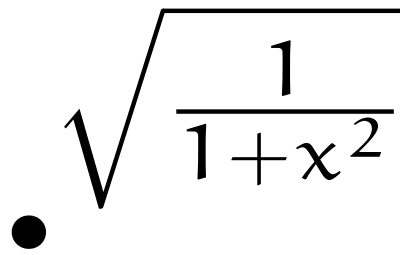


FIGURE 6: Using LATEX for typesetting a label

Only simple text may be used for labels that are included by METAPOST directly. But for real typesetting, we may use one of the best typesetting programs we know, TEX itself. We may include nearly arbitrary TEX stuff into labels. All TEX snippets that occur in the METAPOST source are extracted and typeset with TEX, before the result is included into the figure by METAPOST.

All the TEX stuff has to be embedded into an environment `btex ... etex`, e.g., `label(btex z_0 etex, z0)` will center the label “ z_0 ” at this point. Normally, the plain TEX compiler is used for typesetting the `btex ... etex` fragments. But the name of the program may be passed in command line when calling METAPOST, so to use LATEX we just use the command

```
mpost --tex=latex ex1
```

Let’s have a look how we can typeset a label with LATEX, using Euler math fonts to typeset a formula:

LISTING 6: Typesetting a label with LATEX

```
filenametemplate "%j-%3c.mps";
verbatimtex
\documentclass{article}
\usepackage{euler}
\begin{document}
etex
beginfig(1);
dotlabel.urft(
  btex $\sqrt{\frac{1}{1+x^2}}$ etex,
  origin);
endfig;
```

The example contains some LATEX code within a `verbatimtex ... etex` environment, that is taken verbatim as LATEX code before typesetting all the labels. Please note that it is used to define a preamble with `article` as document class and loading the `euler` package. While the `\begin{document}` is present, the closing `\end{document}` is inserted automatically.

The result of this code, when compiled by METAPOST with using LATEX as typesetter—as explained above—is shown in fig. 6.

Including labels causes some difficulties with fonts. Normally, METAPOST doesn’t embed fonts but just adds a reference to the used fonts into

the PostScript output. This isn't a problem when METAPOST is included in a T_EX document, since T_EX will resolve all of these font references. But the figures won't be usable standalone, since PostScript interpreters like GhostScript will complain about unknown fonts.

In recent versions of METAPOST it is possible to run METAPOST in a mode that will produce standalone PostScript output that contains a "real" EPS with all fonts embedded, that can be displayed in any PS interpreter or may be used in other applications besides T_EX documents.

A switch named `prologues` defines whether METAPOST will embed fonts or not. Just use the definition `prologues:=3;` at the start of your METAPOST file to get a standalone EPS figure. The default value of `prologues` is 0, which means that no fonts will be embedded. The meaning of other values of `prologues` may be looked up in the METAPOST manual, they are only relevant for special cases.

9 Conclusion

This article was intended to just give a short introduction to METAPOST. I left out several things, e. g., how to use colors, defining macros, loops etc. Since the article is originally based on a talk that was part of a comparison of tools, it is focussed on what makes METAPOST unique among other

drawing tools: solving linear equations and having direct access on intersection points and time variables of paths.

If you are interested in learning METAPOST, please have a look into the METAPOST manual (HOBBY, 2008) or into the L^AT_EX Graphics Companion (GOOSSENS *et al.*, 2007), that describes METAPOST (among many other tools).

References

- GOOSSENS, M., MITTELBACH, F., RAHTZ, S., ROEGEL, D. e VOSS, H. (2007). *L^AT_EX Graphics Companion, The, 2nd Edition*. Addison-Wesley Professional.
- HOBBY, J. D. (1989). «A METAFONT-like System with PostScript Output». *TUGboat*, **10** (4). URL <http://www.tug.org/TUGboat/Articles/tb10-4/tb26hobby.pdf>.
- (2008). «METAPOST – A User's Manual». URL <http://www.tug.org/docs/metapost/mpman.pdf>.

▷ Klaus Höppner
Haardtring 230a, 64295 Darmstadt,
Germany
`klaus.hoeppner at gmx dot de`

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 `pdf\text{\LaTeX}`, 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 L^AT_EX 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 L^AT_EX 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 un 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**

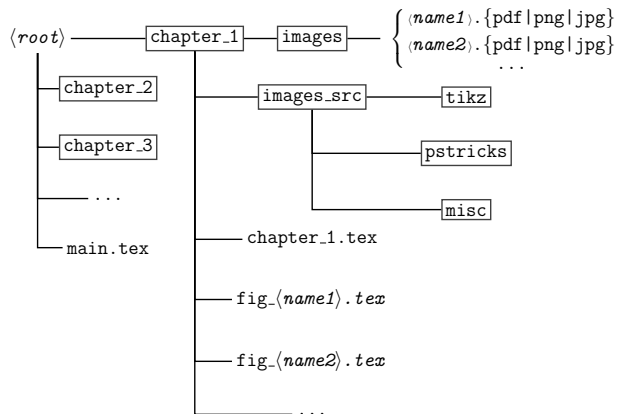
Per gli utilizzatori di LATEX una figura è il contenuto di un ambiente **figure**. Tipicamente questo ambiente è usato per inserire immagini esterne attraverso il comando `\includegraphics` del pacchetto **graphicx**. Per approfondire l'uso di questo ambiente e delle opzioni del pacchetto **graphicx**, 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

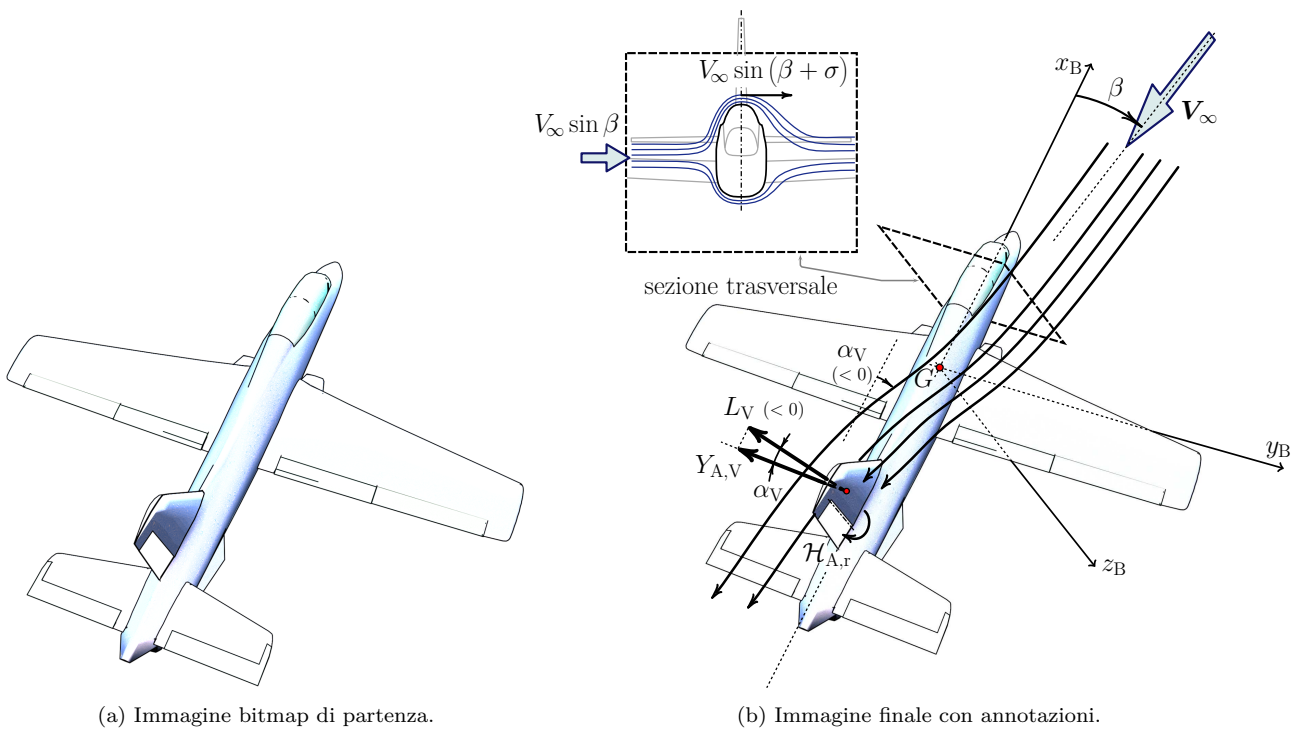


FIGURA 2: Esempio di annotazione di una immagine bitmap.

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 (HOBBS, 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

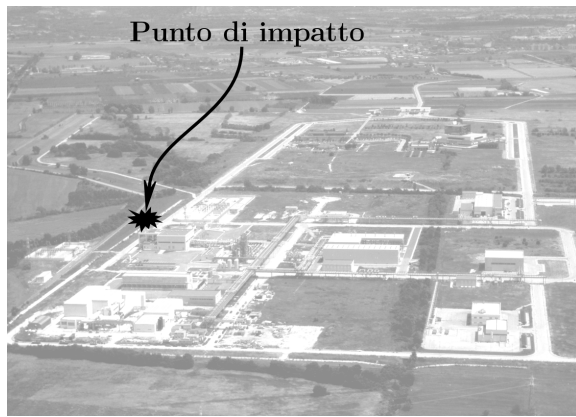


FIGURA 3: Esempio di immagine bitmap con una semplice annotazione. Immagine creata con Gimp. Il font del testo è Latin Modern (formato *Open Type*).

(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_p, y_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

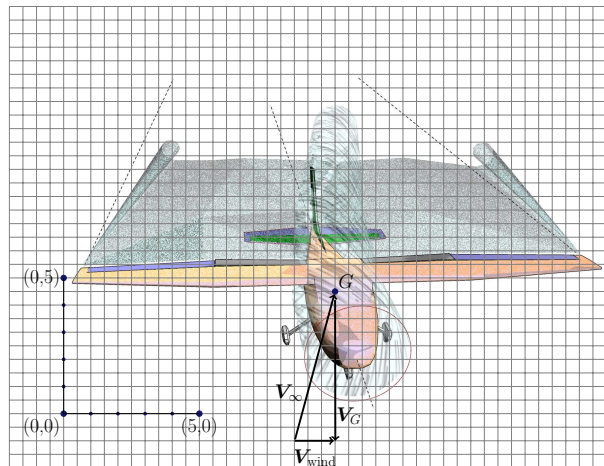


FIGURA 4: Una immagine bitmap, `airplane.pdf`, importata in un ambiente tikzpicture ed annotata con l'ausilio di una griglia.

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`

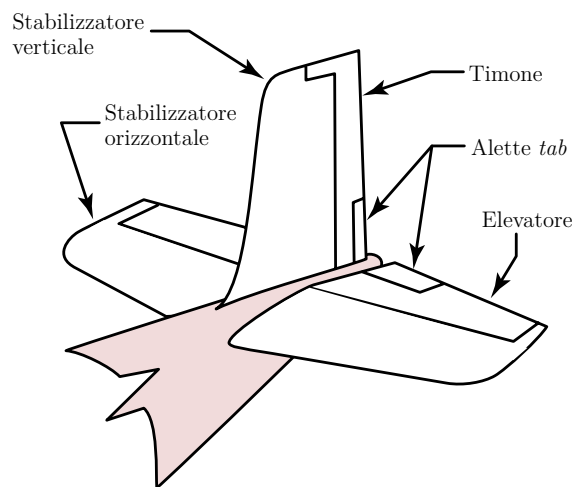


FIGURA 5: Esempio di immagine vettoriale. Le semplici annotazioni testuali sono state create con \LaTeX e `psfrag`.

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 multiplatforma 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* `texttext`, che permette di elaborare oggetti testuali mediante \pdf\LaTeX 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 precedentemente 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`.

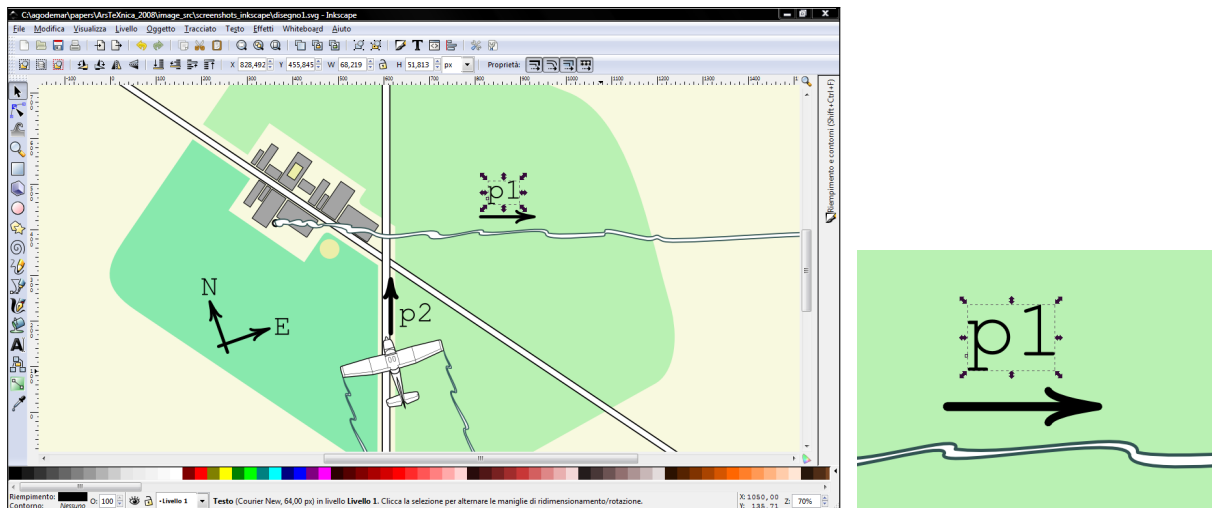
Il primo passo di compilazione con \LaTeX consente all'utente di specificare cosa cercare nell'immagine `picture1.eps` e con cosa ciò deve essere sostituito. Si consideri l'esempio seguente.

```
\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}_{\!\!\!\mathit{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){\boldsymbol{x}_{\!\!\!\mathit{E}}} (Nord)}
\psfrag{E}[l][l]{
  \rput(0,0){\rnode{E}{}}
  \uput{4pt}[0](E){\boldsymbol{y}_{\!\!\!\mathit{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 L^AT_EX 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 `[B1][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 " V_{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.

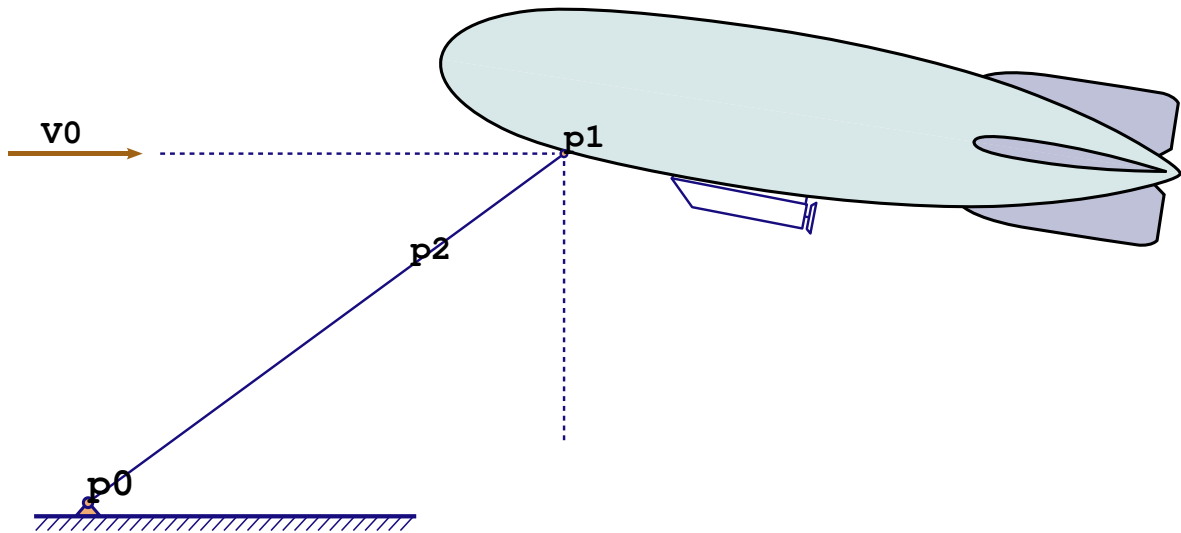
```
\usepackage{psfrag,pstricks,
  pst-eucl,pstricks-add}

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

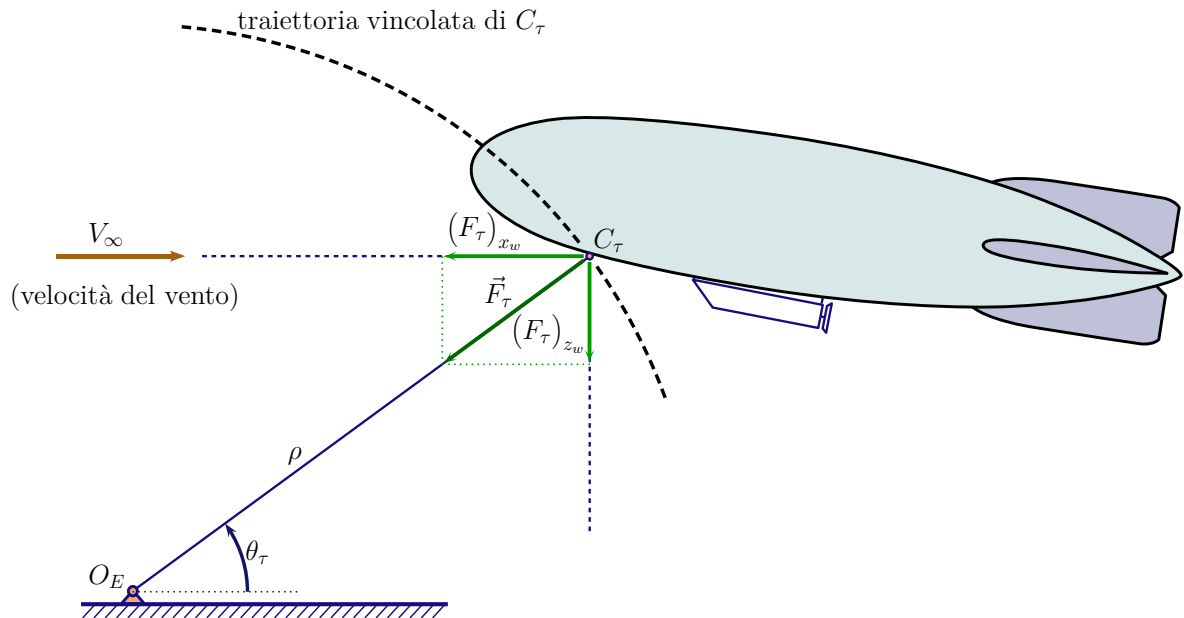
\makeatletter
\def\psMyArc{\pst@object{psMyArc}}
\def\psMyArc@i{\@ifnextchar{\psMyArc@iii}%
  {\psMyArc@ii}}
\def\psarc@ii#1{\addto@par{arrows=#1}%
  \@ifnextchar{\psMyArc@iii}{\psMyArc@iii(0,0)}%
}

\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}[b1][b1]{
\rput(0,0){\rnode{P1}{}}
\uput{3pt}[45](0,0)
{\relsize{1}$C_{\tau}$}
}
% point P2
\psfrag{p2}[b1][b1]{
\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=B1,labelsep=0pt,npos=0.7]
{${\bf F}_{\tau}^{\tau}_{z_w}$}
% the total F vector
\ncline[linewidth=2.0pt,
linecolor=green!40!black,

```



```

        arrows=-D>,nodesepA=3pt]{P1}{P2}
\ncput[ref=B1,labelsep=0pt,npos=0.5]
    { $\vec{F}_\tau$ }
\psParallellLine[linestyle=none]
    (10,0)(-10,0)(P1){10}{P1b}
\psLCNode(P2){1}(0,10){1}{P2a}
% end of Fw component
\psIntersectionPoint(P1)(P1b)(P2)(P2a){P1X}
\ncline[linewidth=2.0pt,
    linecolor=green!60!black,
    arrows=-D>,nodesepA=3pt]{P1}{P1X}
\ncput[ref=B1,labelsep=1pt,npos=0.7]
    { $\big(F_\tau\big)_x$ }
%
\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){\node{P0}{}}
    \uput{4pt}[140](P0){ $E$ }
    \rput(3,0){\node{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}{x}
\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

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

`\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;

`\psParallellLine`: 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`

comandi che servono a produrre le annotazioni desiderate.

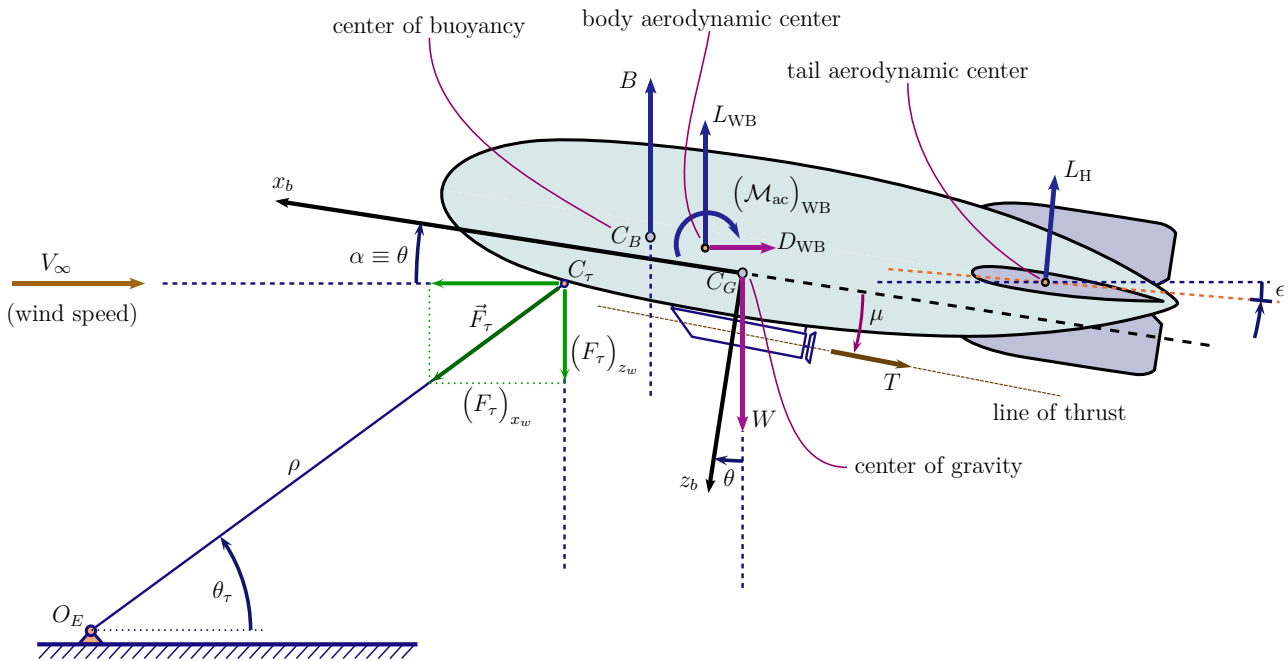


FIGURA 9: Esempio di immagine vettoriale con non poche annotazioni.

(BATTALIA, 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 legende 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 docu-

mentarsi su come riprodurre un grafico sfruttando le caratteristiche di L^AT_EX.

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 (GNUPLLOT, 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 sono 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 L^AT_EX o pdfL^AT_EX, che permettono di gestire gli oggetti testuali di un grafico, coerentemente con lo stile tipografico del testo corrente.

5.1 Annotare con pdfL^AT_EX 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.

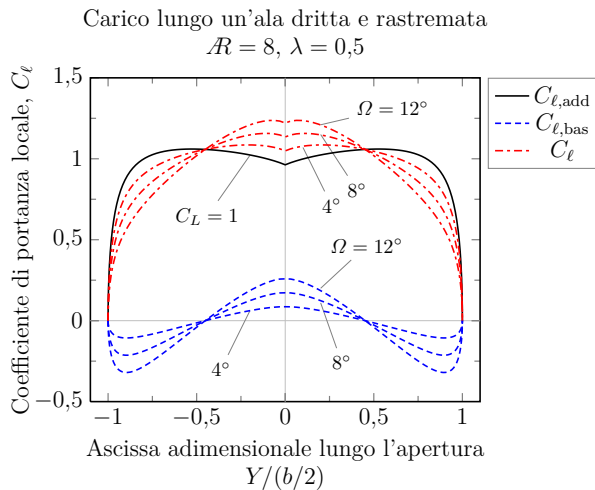


FIGURA 10: Un grafico prodotto con pgfplots. La figura è dotata di una legenda e di molte annotazioni.

```
\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
      $\R=8$, $\lambda=\num{0.5}$
    }
  },
  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{
  $C_{\ell,add}$\\
  $C_{\ell,bas}$\\
  $C_{\ell}$
}%

% 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

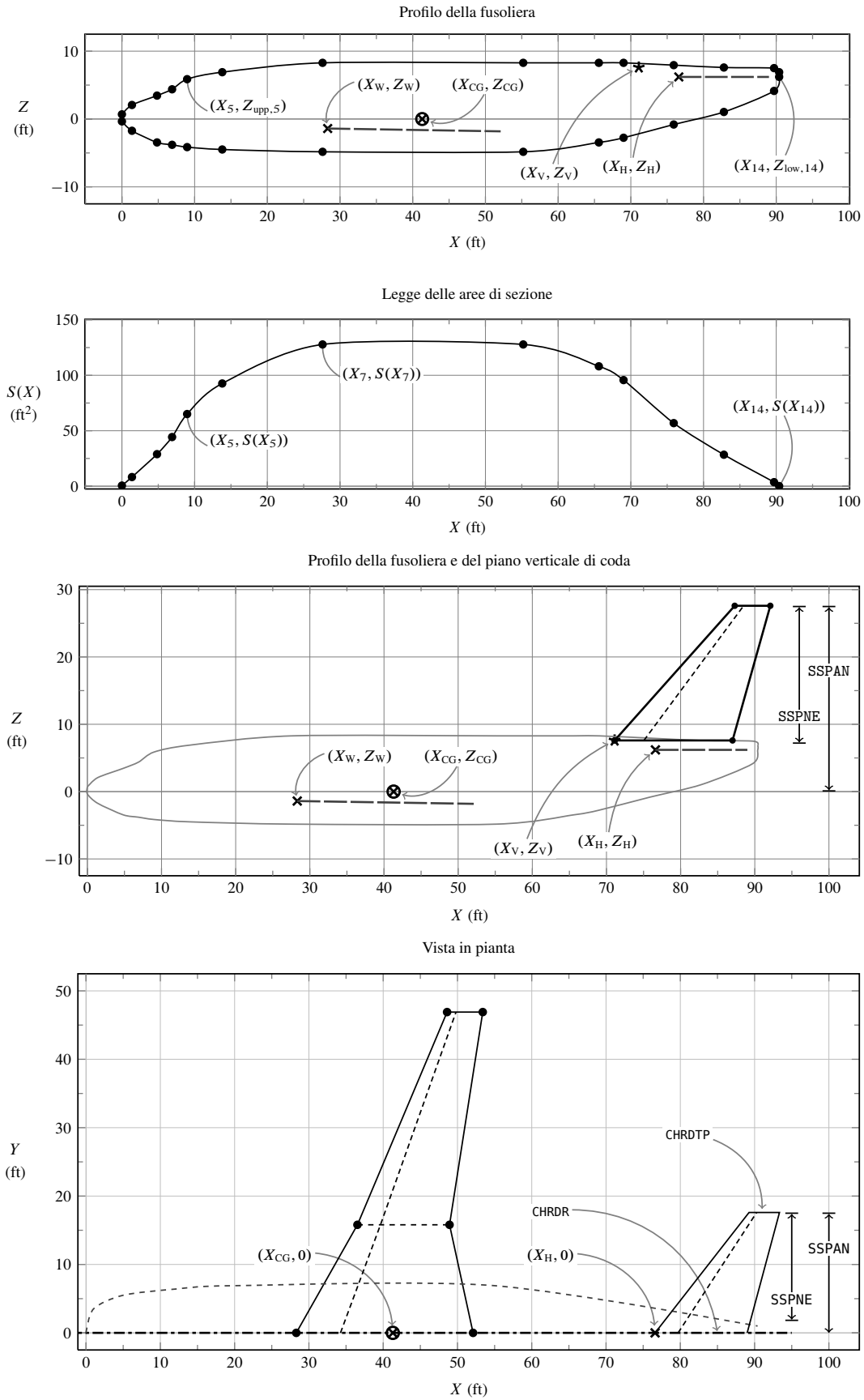


FIGURA 11: Disegni tecnici prodotti con `pgfplots`. Lo scopo di questi grafici è quello di spiegare il significato di alcuni dati di input di un programma di calcolo. Le annotazioni testuali sono composte con i font Times, Math Times Professional e Bera Mono.

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 pdfL^AT_EX.

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), pdfL^AT_EX 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 (GNUPLLOT, 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 (GNUPLLOT, 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 Tikz". 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 ''

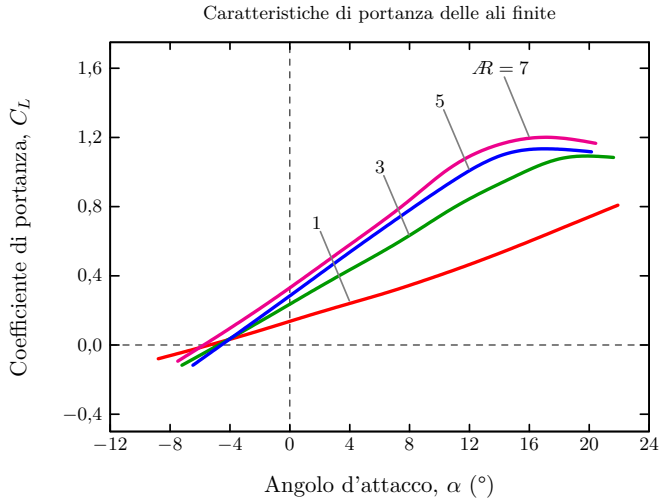
# this sets a tikz node for later reference
set label "" at 16,1.20 font ",gp refnode,name=nA"
set label "" at 12,1.01 font ",gp refnode,name=nB"
set label "" at 8, 0.63 font ",gp refnode,name=nC"
set label "" at 4, 0.25 font ",gp refnode,name=nD"

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
```

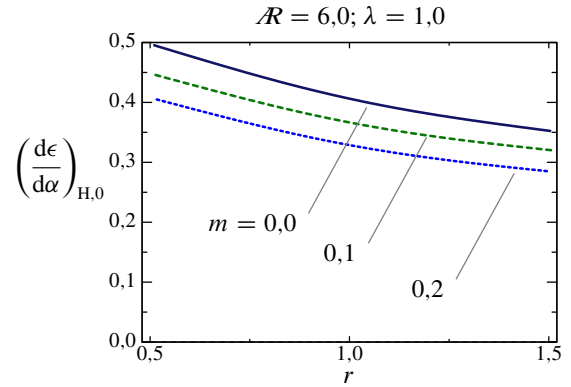
Da una *shell* dei comandi del sistema operativo, il comando `gnuplot alphaCL_AR.plt` richiede a Gnuplot di eseguire sequenzialmente, in modalità *batch*, le istruzioni su riportate. Con l'istruzione `set terminal lua` si richiede un'uscita secondo il terminale Tikz. Il comando `plot` è quello più importante 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 L^AT_EX. Ciò fa capire che l'uso di questi comandi deve essere accoppiato ad un sorgente L^AT_EX 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}
\usepackage{siunitx}
% 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{R} 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{R} 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{plohandlers}
\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}{%
    \pgfpointlineatime{#3}%
    {\pgfpointanchor{gpbb south west #1}%
      {center}}%
    {\pgfpointanchor{gpbb south east #1}%
      {center}}%
  }%
  \pgfextracty{\gptmpdimy}{%
    \pgfpointlineatime{#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 L^AT_EX.

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

5.3 Annotare con L^AT_EX, *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 pa-

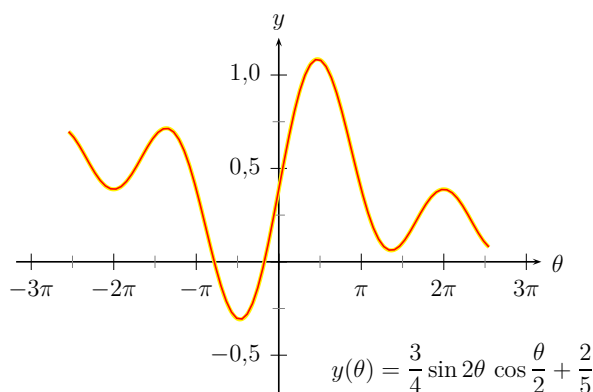
rametricamente (macro *\parametricplot*) e grafici in coordinate polari (opzione *polarplot*). È possibile disegnare grafici attingendo i dati da file esterni con i comandi *\readdata* e *\listplot*.

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.

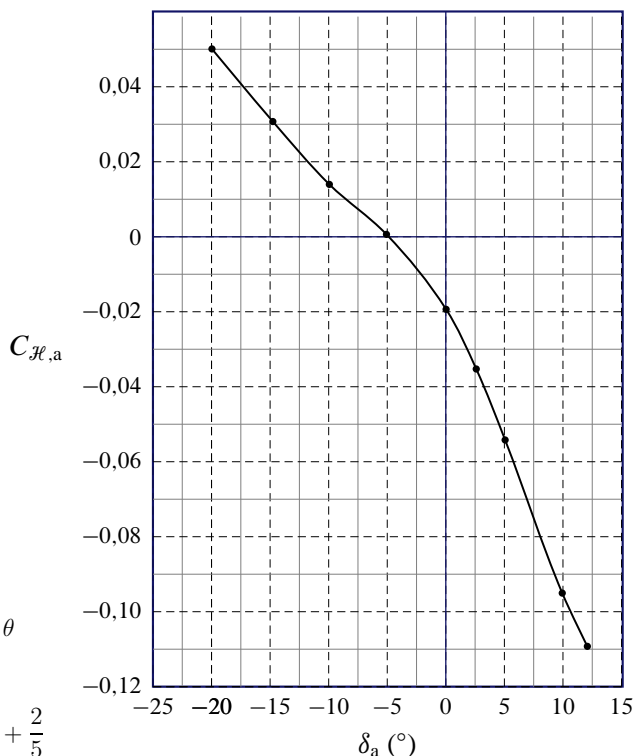
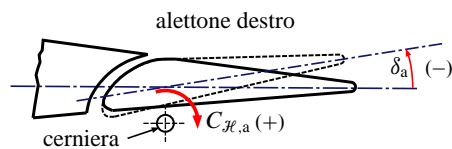
Riferimenti bibliografici

- 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*». *ArsT_EXnica*, (1), pp. 20–28. URL <http://www.guit.sssup.it/arstexnica.php>.
- (2007). «Introduzione a PSTricks». *ArsT_EXnica*, (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/>.
- FEUERSAENDER, 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}\sin 2\theta \cos \frac{\theta}{2}+\frac{2}{5}}
\end{psgraph}
```

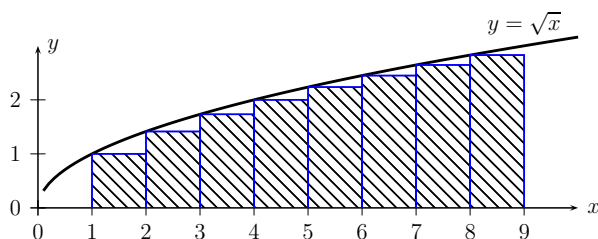


(a) Grafico di una funzione con l'ambiente `psgraph`. In alto è riportato il codice che serve a generare il disegno.



(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/pfguide.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 $\text{\LaTeX} 2_{\epsilon}$* . Gruppo Utilizzatori Italiani di \TeX e \LaTeX .
- PAULI VIRTANEN (2008). Texttext. URL <http://www.elisanet.fi/ptvirtan/software/texttext/>.
- 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 Additional 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

Una tabella che fa calcoli

Roberto Giacomelli

Sommario

Una semplice sintassi di markup del testo è uno dei punti di forza di \LaTeX . L'utente è in grado di costruire il contenuto informativo del documento senza doversi preoccupare di come esso stesso apparirà sulla pagina.

L'articolo vuole mettere alla prova questa caratteristica importante per la produttività dell'autore e per la qualità del suo lavoro costruendo, con particolare attenzione al progetto della sintassi, un nuovo ambiente di tipo tabella utile per fatture, note di spesa o di liquidazione.

Si discuteranno prima gli aspetti linguistici, dando conto di una sperimentazione sul campo condotta coinvolgendo un gruppo di utenti in ambito aziendale, e poi il codice completo del nuovo ambiente, chiamato *calctab*, iniziando dal problema del calcolo in virgola mobile in \TeX e dall'elenco dei potenti pacchetti \LaTeX impiegati.

Si daranno infine alcuni spunti metodologici per l'incremento della diffusione di \LaTeX nelle aziende e negli studi professionali.

Abstract

One of main advantages of \LaTeX is an easy markup syntax of the text. The user is able to build up the informative content of the document without worrying about how it will appear on the page.

This paper aims at testing this remarkable characteristic of \LaTeX , both for the author's productivity and the quality of his work, building, with particular attention to the syntax project, a new environment of type table for invoices, expense notes and liquidation.

The linguistic aspects will be first discussed, taking into account a testing carried out on a group of users working in the business sector. Then the complete code of the new environment called *calctab*, will be proposed, starting with the list of the powerful \LaTeX packages, it is based upon, and the numeric \TeX capability.

In the end some method hints will be given in order to widespread \LaTeX in business and professional offices.

1 Linguaggio: sintassi e semantica

La sintassi base di \LaTeX è particolarmente semplice, basandosi essenzialmente sulla seguente struttura: un carattere di controllo che precede il no-

me del comando, cui segue il testo dell'argomento racchiuso tra parentesi graffe¹.

Durante il processo di compilazione verrà eseguito il codice corrispondente al nome del comando utilizzando come input il testo dell'argomento e ciò produrrà il risultato sulla pagina.

L'utente deve quindi conoscere non solo i nomi dei comandi corrispondenti alle procedure di tipo composizione che desidera effettuare, ma anche il significato che verrà attribuito al testo dell'argomento e, se vi sono più argomenti, l'ordine in cui digitarli.

Questo *ensemble* sintattico/semantico caratterizza non solo \LaTeX , ma i linguaggi informatici di alto livello.

Poiché l'autore del pacchetto o della classe è libero di cambiarne l'implementazione pur mantenendo il più possibile inalterate la sintassi e la semantica a cui si riferiscono gli utenti, in definitiva il linguaggio può esser visto come l'interfaccia pubblica del codice.

Ogni comando converrà che sia progettato per esporre un nome breve che sinteticamente ricordi l'elaborazione che sarà effettuata e per ordinare, secondo una logica intuitiva, la sequenza degli argomenti necessari. Di più, converrà che l'intero set di comandi adotti una "filosofia linguistica" uniforme così da facilitarne l'apprendimento e l'uso da parte degli utenti che gestiranno quindi un unico concetto linguistico.

Questo è particolarmente importante se si tiene conto che \LaTeX è l'opposto di un linguaggio statico. Pacchetti e nuove classi possono estendere le funzionalità compositive, pertanto molti nuovi elementi del linguaggio vengono resi disponibili nei singoli documenti sorgenti.

Tenteremo di applicare correttamente i principi di progettazione del linguaggio per definire la sintassi dell'ambiente di esempio, e di raggiungere l'equilibrio tra il contenuto informativo puramente testuale del sorgente \LaTeX e la restituzione nel documento finale, tipico dell'eleganza dei sistemi asincroni di editoria digitale.

1.1 Un esempio semantico

Per comporre testo sulla pagina si può usare il comando `\parbox`, che prevede due argomenti obbligatori: la larghezza del paragrafo ed il testo.

1. Tralasciamo per un attimo la struttura del testo sorgente e gli ambienti.

TABELLA 1: Una semplice tabella d'esempio

Previsione spese d'impianto

	Descrizione voce	€
A	Fornitura principale	80 000,00
B	Costi gestionali	45 000,00
	Costo impiantistico (A+B)	125 000,00
C	IVA (20,00% su A+B)	25 000,00
	Totale complessivo (A+B+C)	150 000,00

È naturale pensare che i due argomenti debbano essere ordinati nella sequenza sopra citata ed in effetti la sintassi lo conferma:

```
\parbox[⟨position⟩]{⟨width⟩}{⟨text⟩}
```

Fortunatamente non è usuale che il *testo* del paragrafo sia una misura espressa nelle unità comprese da \TeX , dunque ci si accorgerebbe immediatamente dell'errore nel documento finale, come invece potrebbe non accadere nel seguente esempio:

```
\parbox{3.8cm}{2cm}
\parbox{2cm}{3.8cm}
```

che produce (la riquadratura è aggiunta per migliorare la comprensione):

2cm
3.8cm

2 Il nuovo ambiente calctab

Spesso mi capita di dover compilare una tabella di liquidazione in documenti aziendali (come quella mostrata in Tabella 1).

Tali tabelle riportano un testo descrittivo ed alcune righe in cui compaiono voci economiche come spese, imposte e somme.

Per costruirle nel sorgente \LaTeX occorre definire un normale ambiente `tabular`, digitare i giusti comandi per comporre le righe di separazione, calcolare i valori numerici e digitarli a mano nel giusto formato, digitare le descrizioni delle voci, risistemare il sorgente per allineare le colonne più chiaramente, ecc.

E se cambia un valore occorre procedere al ricalcolo manuale ed alla correzione delle cifre.

L'idea è di sviluppare un nuovo ambiente con cui esprimere i dati essenziali direttamente riferiti alle voci economiche, e di lasciare che il codice, dietro le quinte, si occupi sia dei dettagli grafici sia dei calcoli.

Per i diversi tipi di voci economiche si è pensato di associarvi un comando dedicato:

- `\amount` – inserimento voci economiche
- `\perc` – inserimento voci a percentuale
- `\add` – calcolo di somme

Ai comandi dovranno essere passati due argomenti obbligatori: $\langle description \rangle$ e $\langle value \rangle$ in ordine di lettura nel rigo di tabella, tranne per `\add` che non ha bisogno di un valore poiché ottenuto dall'elaborazione. L'argomento $\langle value \rangle$ per il comando `\perc`, sarà interpretato come la percentuale da applicare.

Per generare la Tabella 1 basta scrivere il seguente codice in cui per riferimento, nei commenti a fine riga sono indicate le lettere che verranno assegnate ai righe in tabella:

```
\begin{calctab}[Previsione spese d'impianto]
\amount{Fornitura principale}{80000} % A
\amount{Costi gestionali}{45000} % B
\add{Costo impiantistico}
\perc{IVA}{20} % C
\add{Totale complessivo}
\end{calctab}
```

Regole generali dell'ambiente: ciascun comando genera una riga di tabella. Le operazioni di calcolo richieste vengono effettuate solo sui valori delle righe precedenti ad esclusione di quelli inseriti con il comando `\add`.

Così nell'esempio di Tab. 1 la percentuale del 20% (rigo C) sarà applicata ai valori espressi nelle precedenti righe (A e B), mentre l'ultimo comando sommerà i valori inseriti dai comandi `\amount` e `\perc` relativi ai righe A, B, e C.

I tre nuovi comandi accettano anche un argomento opzionale (dipendente dal comando) che può essere utilizzato per assegnare un nome al rigo generato dal comando stesso, e/o specificare una lista di nomi tra quelli già assegnati a cui limitare l'elaborazione.

I nomi identificativi di riga possono essere scelti dall'utente in modo significativo così da renderne più naturale l'utilizzo, ed attribuiti con sintassi in stile *chiave = valore*. La chiave scelta è semplicemente `id`.

La selezione opzionale dei righe e quindi anche dei valori corrispondenti, si costruisce invece inserendo l'elenco dei relativi nomi separati da virgola nell'argomento opzionale dei comandi.

Di seguito compare il codice d'esempio di un ambiente `calctab` in cui si etichettano alcune righe allo scopo di applicare diverse aliquote d'imposta (il risultato è la Tabella 2):

TABELLA 2: Una seconda tabella d'esempio

Revisione spese d'impianto	
Descrizione voce	€
A Fornitura	82 500,00
B Opere strutturali	28 000,00
C Gestione commessa	46 500,00
D IVA (10,00% su A+B)	11 050,00
E IVA (20,00% su C)	9 300,00
Totale IVA (D+E)	20 350,00
Totale complessivo (A+B+C+D+E)	177 350,00

```
\begin{calctab}[Revisione spese d'impianto]
  \amount[id=impianto]{Fornitura}{82500}
  \amount[id=lavori]{Opere strutturali}{28000}
  \amount[id=gest]{Gestione commessa}{46500}
  \perc[id=iva10,impianto,lavori]{IVA}{10}
  \perc[id=iva20,gest]{IVA}{20}
  \add[iva10,iva20]{Totale IVA}
  \add{Totale complessivo}
\end{calctab}
```

Si noti come in questo caso, sia stata data al comando `\perc`, sia l'etichetta (`iva10`), sia la lista di righe precedenti su cui operare (`impianto,lavori`), ed è l'unico comando che accetta entrambe le cose.

Il comando `\add` non può essere etichettato², mentre al comando `\amount` non può essere passata una lista di identificatori costituendo una voce economica sempre indipendente dalle altre.

2.1 Specifica di sintassi

La sintassi dell'ambiente `calctab` è la seguente:

```
\begin{calctab}[<description paragraph>]
  <calctab commands list>
\end{calctab}
```

dove per `<calctab commands list>` si intende una sequenza dei comandi:

```
\amount[id=<rowID>]{<descr>}{<value>}
```

```
\perc[id=<rowID>,<rowID list>]{<descr>}{<value>}
```

```
\add[<rowID list>]{<descr>}
```

3 Test d'utilizzo

Ho chiesto ad alcuni colleghi di scrivere una classica tabella di liquidazione di una notula professionale avvalendosi di un ambiente `calctab`. Un compito che essi svolgono quasi quotidianamente, dunque un tema di loro interesse, che normalmente viene svolto con l'uso di comuni word processor o fogli di calcolo.

2. Questa scelta mantiene più chiara la sintassi a scapito di un po' di funzionalità. Vedi la sezione 3 per la discussione in merito.

Durante colloqui informali individuali, ho mostrato loro dapprima alcuni esempi di sorgenti compilabili per abituarli al linguaggio di markup (nessuno infatti, aveva mai nemmeno sentito nominare la parola L^AT_EX), infine ho fornito loro una tabella d'esempio cedendogli la tastiera. Ho diligentemente registrato le osservazioni e le domande che durante il test ciascun collega mi poneva nell'intento di misurare la *soddisfazione del cliente*. Gli incontri hanno confermato l'utilità di discussioni simili al fine di migliorare con efficacia il *prodotto* (il lavoro di squadra spesso è insuperabile).

La maggior parte delle domande che mi sono state rivolte riguardanti l'ambiente `calctab`, erano di fatto richieste di estensioni. Ad esempio, la possibilità di cambiare "unità di misura" per i valori in tabella (oltre l'euro), e la gestione di nuove colonne *Quantità* e *Prezzo* tipiche di un prospetto economico in cui il valore di riga corrisponde al prodotto di tali numeri. Od ancora la possibilità di mutare le lettere di riga in prima colonna, in numeri progressivi (vedi le Tabelle 1 e 2).

A quest'ultima richiesta ho obiettato che le lettere maiuscole risultano più leggibili nelle lunghe espressioni di descrizione delle somme: il richiamo (`A+B+C+D+E`), visibile nell'ultima riga in Tab. 2 è forse più chiaro di (`1+2+3+4+5`).

Sull'argomento più importante, la sintassi di `calctab`, sono stati dibattuti due temi, entrambi inerenti alle etichette opzionali dei comandi.

Il primo verte sulla possibilità di ricorrere ad un'etichettatura automatica anziché esplicita delle righe di tabella, per esempio con la sequenza `row1, row2, ...`, dunque per comporre la tabella 2 avremmo dovuto scrivere il codice seguente:

```
% modifica proposta alla sintassi:
\begin{calctab}[Revisione spese d'impianto]
  \amount{Fornitura}{82500} % 1
  \amount{Opere strutturali}{28000} % 2
  \amount{Gestione commessa}{46500} % 3
  \perc[row1,row2]{IVA}{10} % 4
  \perc[row3]{IVA}{20} % 5
  \add[row4,row5]{Totale IVA}
  \add{Totale complessivo}
\end{calctab}
```

Nell'approfondito dibattito che ne è seguito, ho giustificato la diversa scelta adottata, proponendo i seguenti argomenti senz'altro opinabili: in primo luogo ci si deve ricordare i caratteri utilizzati per costruire la sequenza delle etichette, e ciò è più faticoso che non ricordarsi il nome della chiave `id`.

In secondo luogo occorre *contare* le righe inserite per formarne effettivamente i nomi e questo è meno elegante ed espressivo del metodo di assegnare direttamente nomi consistenti alle righe. Infine, in caso di modifica od inserimento di nuove righe occorre che sia aggiornato manualmente l'elenco dei parametri opzionali già digitati.

Il secondo tema invece, sembra una buona idea: poter assegnare un identificatore di riga nell'argomento opzionale del comando `\add`. In altri termini, si tratta di poter assegnare un nome anche alle righe di somma di valori, sia essa totale o selettiva, per compirvi ulteriori elaborazioni senza dover identificare ciascun rigo "addendo" per poi comporne l'elenco nel comando.

In questo modo una tabella di notula professionale come la Tab. 3 (pag. 35) si risolverebbe con l'assegnamento di un'unica chiave `id`. In particolare basterebbe assegnare l'`id` al comando che somma le voci di notula ($A+B+C$) e dichiararlo come lista nel comando che effettua il calcolo della ritenuta d'acconto al rigo F.

Evitare di etichettare tutte le righe solo per riottenerne la somma per righe successivi è uno strumento agile che completa il linguaggio. Tale valore infatti, è già disponibile nella riga di somma ed è naturale che si possa richiamarlo da qualche riga più sotto.

La scelta sintattica iniziale è risultata quindi migliorabile sia dal punto di vista concettuale che funzionale. Il codice dovrà essere modificato con l'aggiunta di un nuovo *stack* di registri in virgola mobile con i relativi *flag booleani* di stato. In questo modo i valori numerici principali inseriti dai comandi `\amount` e `\perc` rimangono indipendenti e le regole base dell'ambiente ancora valide.

4 Implementare il linguaggio

L'implementazione del nuovo ambiente `calctab` è stata sviluppata per stadi, risolvendo un problema alla volta e facendo in caso ricorso, a post sul forum di `qJr` come accennato nella sezione 8.

La strategia di programmazione si è inoltre orientata verso l'uso di pacchetti `LATEX` già disponibili, visti un po' come librerie di codice.

4.1 Eseguire calcoli

Per calcoli con numeri interi, `TEX` dispone dei contatori `\count`, numeri con segno a 32 bit il cui range è $[-2^{31}, 2^{31} - 1]$. Tra l'altro il limite superiore dell'intervallo fissa a poco più di due miliardi il numero massimo di pagine di un documento (vedi GREGORIO (2008) pagine 35 e 42).

Per i numeri decimali si possono impiegare registri dimensionali, per esempio `\dimen`, dalla notazione in *virgola fissa* con 14 bit per la parte intera, 16 bit per quella frazionaria, un bit di segno ed un bit di overflow, per un totale di ancora 32 bit (l'intervallo possibile è $[-2^{14}, 2^{14} - 1]$).

Il limite superiore stabilisce la massima lunghezza a circa 5,76 m ($\approx 2^{14} = 16.384$ pt)³, mentre la precisione raggiungibile è 2^{-16} pt $\approx 5,36$ nm, lunghezza che in `TEX` viene definita punto scalato `sp`. Il valore è assai meno della lunghezza d'onda della luce visibile compreso tra 380 e 750 nanometri, quindi errori dovuti ad approssimazioni numeriche sono irrilevabili nel documento.

Può essere interessante notare che nel modulo `pgfmath`, il motore matematico del noto pacchetto grafico PGF, i calcoli decimali sono svolti con l'ausilio dei registri dimensionali `TEX`, come dichiara il manuale alla sezione 49 (TANTAU, 2008).

Se da un lato questi "tipi" numerici bastano per le esigenze di `TEX`, dall'altro all'epoca del suo sviluppo l'aritmetica in *virgola mobile* non garantiva una sufficiente congruità tra le diverse piattaforme hardware e software, così che Donald Knuth non la implementò per `TEX`.

In effetti, a detta dello stesso Knuth, il problema del calcolo in *virgola mobile* è più complesso e sorprendente di quello che si potrebbe pensare (KNUTH, 1998). Esso coinvolge aspetti teorici, di progettazione dei processori e di sviluppo di standard internazionali⁴ (KAHAN, 1981).

Grazie a più di vent'anni di evoluzione di questo importante settore del calcolo, future incarnazioni di `TEX` potranno disporre di questo tipo di aritmetica per altro utile in molti tipi di elaborazioni di tipocomposizione (BEEBE, 2007).

Ritornando all'ambiente `calctab`, questa necessità di operare con numeri decimali e con valori che superano i limiti dei registri dimensionali.

Per fortuna sono stati implementati pacchetti `LATEX` che svolgono operazioni in *virgola mobile* anche se in maniera non efficiente. Si è optato per `fltpoint` (GUTHÖHRLEIN, 2004) che offre la possibilità dell'uso di comodi registri, impiegati per la memorizzazione dei valori di riga di `calctab`. Il codice della tabella è così in grado di eseguire le operazioni previste con la necessaria precisione e facilmente supportarne altre future.

4.2 I pacchetti di "libreria"

Fondamentale per la semplicità d'implementazione il pacchetto `xkeyval` (ADRIAENS, 2006), con cui non solo si è risolto il problema posto dalla prevista sintassi dell'argomento opzionale dei comandi interni di `calctab`, ma si è ottenuto una notevole eleganza e

3. $1 \text{ pt} = \frac{25,4}{72,27} \text{ mm} \approx 0,35 \text{ mm}$ e $1 \text{ nm} = 10^{-9} \text{ m}$.

4. Lo standard IEEE 754 - "Standard for Binary Floating-Point Arithmetic" del 1985 fu il primo punto di riferimento in materia.

compatezza del codice che lo utilizza (vedi sezione 4.3).

Per procedere all'implementazione degli algoritmi di calcolo, ci si è avvalsi del potente `ifthen` (CARLISLE, 2001), su cui è basato il loop di calcolo principale (comune sia ai comandi `\perc` e `\add`), nonché utilizzato per costruire una serie di flag di controllo.

La formattazione delle cifre a due decimali fissi, è stata invece affidata al pacchetto `numprint` (HARDERS, 2008), che ci si aspetta renda agevoli ulteriori miglioramenti e sviluppi dell'ambiente `calctab`.

Si è invece scartato (solo per il momento) l'uso del pacchetto `datatool` (TALBOT, 2007) per difficoltà incontrate nella composizione grafica della tabella, problema la cui soluzione probabilmente richiedeva l'esperienza dell'averlo risolto con gli strumenti di base. Si tratta comunque di un pacchetto potente che gestisce tabelle di dati in maniera simile a quelle dei database.

I pacchetti utilizzati per l'aspetto grafico della tabella sono stati: `booktabs` per l'ovvio miglioramento dei filetti orizzontali, `eurosym` per il simbolo dell'euro, ed infine `xcolor` con l'opzione `table` utilizzato per colorare alternativamente le righe della tabella.

4.3 Codice delle chiavi

La definizione di una chiave `id` (ordinaria secondo la classificazione in `xkeyval`), relativa all'etichetta di riga comporta la creazione di una seconda chiave il cui nome è il valore stesso dell'`id`.

Nel codice riportato di seguito ^{5 6} si noti come il parametro `#1` venga espanso a formare il nome del registro numero di riga.

```
% the ordinary key is
% useful to assign a name
% to the table rows
% Note the use of different families

\define@key{ctfamId}{id}{
  % save its row number in ctnumrow
  \fpRegSet{ctreg#1}{\thctRowIndex}
  \define@key{ctfamLabel}{#1}[] {
    % load the corresponding row number
    \fpRegGet{ctreg#1}{\tempnumrow}
    % activate ctflag<row>
    \setboolean{ctflag\tempnumrow}{true}
    % global flag is false if a key
    % is specified
    \setboolean{ctGlobflag}{false}
  }
}
```

5. Le linee sono state accorciate per non andare oltre la larghezza di colonna.

6. Il codice va racchiuso tra i comandi `\makeatletter` e `\makeatother` se compare nel preambolo del file sorgente e non in un file di pacchetto `.sty`.

Per memorizzare il numero di riga e risolvere l'associazione con la chiave `id`, si rende necessario creare un registro `fltpoint` per ciascuna di essa⁷.

Quando una chiave compare nella *(rowID list)*, deve essere attivato il flag che indica che la riga corrispondente dovrà essere considerata nel calcolo, mentre lo stato del flag globale dovrà essere `false` ad indicare un calcolo selettivo.

In altre parole, il flag globale `ctGlobflag` indica lo stato del calcolo: il `true` indica che il calcolo dovrà riguardare tutti i valori già introdotti in tabella (con l'ausilio di comandi `\amount` e `\perc`), il `false` indica che solo le righe i cui `ctflag<row>` sono `true` saranno elaborate nel calcolo.

4.4 Codice dei comandi interni

L'esecuzione del calcolo viene portata a termine con un comando privato, interno all'ambiente, che restituisce il risultato dell'elaborazione nel registro `ctSum`, e nella stringa descrittiva memorizzata in `\ctlabelflag`, quest'ultima utilizzata in *append* su se stessa, effettuando un ciclo per ciascuna riga.

L'importante comando `\ctLoop` utilizza il costruito `\whiledo` contenente l'istruzione condizionale `\ifthenelse`, come si può notare nel seguente listato:

```
\newcommand\ctLoop{
\fpRegSet{ctSum}{0}
\setcounter{ct}{1}
\edef\ctlabelflag{}% for text of label rows
\edef\ctplustext{}% for '+' char
\whiledo
{\not(\value{ct}>\value{ctRowIndex})}{
  \ifthenelse
    {(\boolean{ctflag\thct}%
      \OR%
      \boolean{ctGlobflag})}
    {\fpRegAdd{ctSum}{\ctRowIndex\thct}%
      % row flag status reset
      \setboolean{ctflag\thct}{false}
      \edef\ctlabelflag
        {\ctlabelflag\ctplustext\Alph{ct}}
      \edef\ctplustext{+}
    }{}% else clause denied
  \stepcounter{ct}%
}%
  % reset of the global flag status
  \setboolean{ctGlobflag}{true}
}%
```

Il contatore LATEX chiamato `ctRowIndex` detiene il numero di righe fino a quel momento inserite.

Per convertire il numero di riga in lettere si è fatto uso della funzione del kernel di LATEX `\Alph` che gestisce l'intervallo da 1 a 26 (eventualmente esiste il pacchetto `alphalph` di Heiko Oberdiek (OBERDIEK, 2007) che estende oltre tale limite la traduzione in lettere dei numeri interi utilizzando più caratteri alfabetici).

7. Occorre migliore l'implementazione sostituendo l'uso dei registri `fltpoint`.

Ad ogni ciclo viene “spento” il flag di riga eventualmente attivo per effetto del codice relativo alle chiavi di lista, e, all’uscita, si riattiva comunque il flag globale `ctGlobflag` riportandolo nello stato corrispondente all’elaborazione di tutte le righe.

Poiché i comandi di riga sono definiti internamente all’ambiente `calctab` (usarli all’esterno non avrebbero senso), nel seguente codice del comando `\amount`, gli argomenti hanno un doppio #.

```
\newcommand\amount[3][]{%
  \stepcounter{ctRowIndex}%          new row
  % definition of the bool flag for each row
  \newboolean{ctflag\thectRowIndex}
  % starting value of ctflag<n>
  \setboolean{ctflag\thectRowIndex}{false}
  \setkeys{ctfamId}{##1}
  % save the amount value
  \fpRegSet{ctRowValue\thectRowIndex}{##3}
  % append the row to the token register
  \toks255=\expandafter{\the\toks255\Alph{
    ctRowIndex}&##2\rule{\ctsep}{0pt}&
    \numprint{##3}}\}
}
```

È stato previsto uno spazio orizzontale regolabile, fissato a 8 mm, detto `\ctsep`. Tale spazio viene inserito in seconda colonna per impedire che in alcuni casi la tabella possa apparire troppo stretta.

In effetti, l’output dei comandi interni consiste in una sequenza di token memorizzati nel registro T_EX `\toks255`. Questo permette di risolvere il problema conseguente al fatto che il contenuto di ciascuna cella della tabella composta dall’ambiente `tabular`, viene racchiuso in un gruppo implicito. Tale oggetto, descritto per esempio in GREGORIO (2008) al Capitolo 2, costituisce una barriera per le definizioni non globali, interne all’ambiente.

Il contenuto del registro verrà poi “liberato” solo dal codice di chiusura dell’ambiente `calctab`.

Il comando `\add` elabora la lista di chiavi e poi lancia il ciclo che renderà disponibile il valore della somma nel registro `ctsum`. Dopodiché non dovrà far altro che aggiungere i token al registro `\toks255` che alla fine conterrà l’intera tabella:

```
\newcommand\add[2][]{%
  \setkeys{ctfamLabel}{##1}
  \ctLoop
  \fpRegGet{ctSum}{\atempnumber}
  \toks255=\expandafter{\the\toks255%
    \midrule\rowcolor{gray!15}&%
    \bfseries##2 (}
  \edef\ctnum{\noexpand\ctlabelreg}%
    noexpand\rule{\ctsep}{0pt}}
  \toks255=\expandafter\expandafter%
    \expandafter{\expandafter\the%
    \expandafter\toks255\ctnum&\bfseries}
  \edef\ctnum{\noexpand%
    \numprint{\atempnumber}}
  \toks255=\expandafter\expandafter%
    \expandafter{\expandafter\the%
    \expandafter\toks255\ctnum\\}
}
```

Nelle prossime linee di codice è riportato il comando `\perc`, quello più complesso dei tre.

Si noti come, prima di lanciare il ciclo `\ctLoop`, deve essere momentaneamente decrementato il contatore di riga per evitare che la riga stessa sia conteggiata nel calcolo.

Dopodiché viene richiesta l’elaborazione delle chiavi: prima quelle di lista (famiglia `ctfamLabel`), poi quella di etichettatura (famiglia `ctfamId`), onde evitare errori nel caso in cui l’utente assegni il nome di riga ed allo stesso tempo lo inserisca nella lista di etichette.

Vengono poi eseguiti i calcoli in virgola mobile per applicare la percentuale indicata dall’utente alla somma contenuta, dopo l’esecuzione di `\ctLoop`, nel registro `fltpoint` `ctsum`.

Infine è la volta dei comandi di aggiornamento del registro `\toks255`.

```
\newcommand\perc[3][]{%
  % another new row:
  \stepcounter{ctRowIndex}
  % a control bool flag for each row:
  \newboolean{ctflag\thectRowIndex}
  % starting value of ctflag<n>:
  \setboolean{ctflag\thectRowIndex}{false}
  % processing of the user id label:
  \setkeys{ctfamLabel,ctfamId}{##1}
  % get the value of the sum:
  \addtocounter{ctRowIndex}{-1}
  \ctLoop
  \stepcounter{ctRowIndex}

  % calculation
  \fpDiv{\tempPerc}{##3}{100}%
  \fpRegSet{tempreg}{\tempPerc}
  \fpRegMul{ctSum}{tempreg}
  \fpRegRound{ctSum}{-2}%
  \fpRegGet{ctSum}{\atempnumber}
  \fpRegSet{ctRowValue\thectRowIndex}
    {\atempnumber}
  \toks255=\expandafter{\the\toks255%
    \Alph{ctRowIndex}&##2 %
    (\numprint{##3}% su }
  \edef\ctnum{\noexpand\ctlabelreg}%
    \noexpand\rule{\ctsep}{0pt}}
  \toks255=\expandafter\expandafter%
    \expandafter{\expandafter\the%
    \expandafter\toks255\ctnum&
  \edef\ctnum{\noexpand\numprint{%
    \atempnumber}}
  \toks255=\expandafter\expandafter%
    \expandafter{\expandafter\the%
    \expandafter\toks255\ctnum\\}
}
```

4.5 Codice dell’ambiente `calctab`

Occorre inizialmente allocare contatori, lunghezze e box. Il codice di chiusura del nuovo ambiente `calctab` sistema i token del registro, e quindi la tabella, in un box per misurarne la larghezza con il comando primitivo `\wd`.

Tale dimensione diventerà la larghezza del materiale di intestazione della tabella, passato all'ambiente come argomento opzionale.

Ecco il listato:

```
% length for tuning width
% of the central column:
\newlength{\ctsep}\setlength{\ctsep}{0.8cm}

% main row counter of table:
\newcounter{ctRowIndex}
% loop index:
\newcounter{ct}

% table box allocation
\newsavebox{\ctTabBox}

\newenvironment{calctab}[1][]{%
  % set font family
  \sffamily

  % set the token register
  \toks255={\rowcolors{1}{gray!15}{}}%
  \begin{tabular}{clr}\midrule&%
    Descrizione voce & \euro \\\midrule

  % reset counter to starting value
  \setcounter{ctRowIndex}{0}

  % save paragraph table header to use it
  % in the environment end code
  \newcommand\ctFirstPar[1]{#1}

  % set standard decimal position
  \nprouddigits{2}

  % new boolean to control numeric loop
  \newboolean{ctGlobflag}
  \setboolean{ctGlobflag}{true}

  % insert \ctLoop code ...
  % insert \amount code ...
  % insert \perc code ...
  % insert \add code ...

}{%closing code
\begin{center}
\begin{lrbox}{\ctTabBox}
\the\toks255\bottomrule
\end{tabular}
\end{lrbox}
\sffamily
\parbox{\wd\ctTabBox}{\ctFirstPar\par%
\smallskip\par
\usebox{\ctTabBox}}
\end{center}
}
```

5 Un comando, una tabella

È interessante notare che si possono costruire nuovi comandi per la composizione immediata di tabelle di tipo `calctab`.

Nei documenti della contabilità dei lavori pubblici, per esempio, può essere necessario riassumere

le voci economiche riguardanti l'emissione di uno Stato d'Avanzamento Lavori (SAL) che consistono in una prima riga con l'importo dei lavori, una seconda con l'IVA, ed un'ultima con il totale.

Per ottenere la tabella può essere predisposto un nuovo comando, chiamato `\saltab`, a cui passare i tre argomenti numerici in un ordine ritenuto essere il più logico:

1. il numero progressivo del SAL emesso,
2. l'importo dei lavori,
3. l'aliquota IVA da applicare.

Il nuovo comando `\saltab` ha dunque la sintassi seguente:

`\saltab{<SAL number>}{<amount>}{<rate>}`

e questa semplice implementazione:

```
\newcounter{cttemp}
\newcommand\saltab[3]{% tabella per SAL
  \setcounter{cttemp}{#1}
  \begin{calctab}[Tabella liquidazione %
    del \Roman{cttemp} %
    Stato d'avanzamento Lavori]
    \amount{Lavori}{#2}
    \perc{IVA}{#3}
    \add{Totale liquidato}
  \end{calctab}}
```

6 L'esempio classico

L'ultimo esempio riguarda la tabella 3 in cui è necessario calcolare una percentuale con due cifre decimali significative ed apporre un testo descrittivo su più linee. È il classico caso della liquidazione di una notula professionale.

7 Conclusioni

Il lavoro svolto ha consentito la messa a punto di una sintassi semplice e l'implementazione di un nuovo ambiente tabella in grado di effettuare calcoli.

Ciò dimostra l'utilità di LATEX anche in ambiti aziendali dove può anche superare in produttività i software di Office Automation WYSIWYG.

La maggior diffusione di LATEX dipende molto dal modo di interagire con il linguaggio.

Comandi intuitivi ed ambienti dal giusto insieme di opzioni, nonché nuovi pacchetti o classi di documento personalizzati, consentono all'utente di dedicarsi ai contenuti, e produrre documenti con minori errori e qualità tipografica elevata.

Tuttavia, senza un forte impegno e motivazione da parte degli utenti, LATEX è destinato ai margini delle scelte IT soprattutto delle imprese, nonostante la gratuità e la libertà concessa dalla licenza LPPL.

La risposta dei colleghi che hanno compiuto le prove mi induce a concludere che l'investimento

TABELLA 3: Esempio classico: la notula professionale

```

\begin{calctab}[Notula per prestazioni professionali:\\
    Progetto impiantistico per la pizzeria ‘‘La Margherita’’]
\amount[id=aria]{Progettazione impianto trattamento aria}{5400}
\amount[id=video]{Progettazione impianto elettrico e di sorveglianza video}{8000}
\perc[id=spese]{Spese generali}{8,55}
\add{Importo totale}
\perc{Contributo cassa nazionale}{2}
\perc{IVA}{20}
\perc[aria,video,spese]{Ritenuta d’acconto}{-20}
\add{Totale complessivo}
\end{calctab}

```

Notula per prestazioni professionali:
 Progetto impiantistico per la pizzeria ‘‘La Margherita’’

Descrizione voce	€
A Progettazione impianto trattamento aria	5 400,00
B Progettazione impianto elettrico e di sorveglianza video	8 000,00
C Spese generali (8,55% su A+B)	1 145,70
Importo totale (A+B+C)	14 545,70
D Contributo cassa nazionale (2,00% su A+B+C)	290,91
E IVA (20,00% su A+B+C+D)	2 967,32
F Ritenuta d’acconto (-20,00% su A+B+C)	-2 909,14
Totale complessivo (A+B+C+D+E+F)	14 894,79

iniziale in risorse e cultura informatica da mettere in campo per il proficuo uso di L^AT_EX sia notevole, ma il paradigma offerto dalla sintassi elegante ed espressiva del linguaggio crea forse uno strumento migliore con cui lavorare.

8 Ringraziamenti

Ringrazio in primo luogo il Professor Enrico Gregorio per avermi aiutato, per mezzo del forum di G_UI_T, a risolvere il problema costituito dai gruppi impliciti in cui sono racchiuse le celle dell’ambiente `tabular`, per avermi suggerito il modo di uguagliare la larghezza del `\parbox` d’intestazione a quella della tabella, e per avermi guidato nella *giungla* degli `\expandafter`.

Un ringraziamento va anche ai miei colleghi, che si sono sottoposti ad una ‘‘quindicina di minuti di perplessità’’ durante i test sintattici, all’amica Marzia Dati per l’aiuto con la lingua inglese, ed alla mia famiglia, sempre felice di concedermi tempo e comunicarmi incoraggiamento per i progetti come questo.

Riferimenti bibliografici

- ADRIAENS, H. (2006). «The xkeyval package». Disponibile su CTAN, <http://www.ctan.org/tex-archive/macros/latex/contrib/xkeyval>.
- BEEBE, N. H. F. (2007). «Extending T_EX and METAFONT with floating-point arithmetic». In *The 28th Annual Meeting of the TeX Users Group*.
- Testo riassuntivo in <http://www.math.utah.edu/~beebe/talks/2007/tug2007/tug2007.pdf>.
- CARLISLE, D. (2001). «The ifthen package». Disponibile su CTAN, <http://www.ctan.org/tex-archive/macros/latex/base>.
- GREGORIO, E. (2008). «Appunti di programmazione in L^AT_EX e T_EX». <http://profs.sci.univr.it/~gregorio/introtex.pdf>.
- GUTHÖHRLEIN, E. (2004). «The fltpoint package». Disponibile su CTAN, <http://www.ctan.org/tex-archive/macros/latex/contrib/fltpoint>.
- HARDERS, H. (2008). «The numprint package». Disponibile su CTAN, <http://www.ctan.org/tex-archive/macros/latex/contrib/numprint>.
- KAHAN, W. (1981). «Why do we need a floating-point arithmetic standard?». Scaricabile da <http://www.cs.berkeley.edu/~wkahan/ieee754status/why-ieee.pdf>.
- KNUTH, D. E. (1998). *The Art of Computer Programming: Seminumerical Algorithms*, volume 2. Addison-Wesley Professional.
- OBERDIEK, H. (2007). «The alphalph package». Disponibile su CTAN, <http://www.ctan.org/tex-archive/macros/latex/contrib/oberdiek/alphalph.pdf>.

TALBOT, N. (2007). «The datatool package». Disponibile su CTAN, <http://www.ctan.org/tex-archive/macros/latex/contrib/datatool>.

TANTAU, T. (2008). «The TikZ and PGF packages». Scaricabile da <http://www.ctan.org/tex-archive/graphics/pgf/base/doc/>

`generic/pgf/pgfmanual.pdf`. Manual for version 2.00.

▷ Roberto Giacomelli
Carrara (MS), Italia
`giaconet at tin dot it`

Gestire la bibliografia con L^AT_EX

Lapo F. Mori*

Sommario

La bibliografia è una parte fondamentale delle pubblicazioni scientifiche. Questo articolo presenta e analizza i principali strumenti che L^AT_EX offre per creare, gestire e personalizzare sia i riferimenti nel testo che l'elenco delle voci bibliografiche.

Abstract

The bibliography is a fundamental part of most scientific publications. This article presents and analyzes the main tools that L^AT_EX offers to create, manage, and customize both the references in the text and the list of references at the end of the document.

1 Premessa

Il presente articolo tratta la bibliografia sia dal generico punto di vista tipografico che dallo specifico punto di vista applicativo. Mentre le considerazioni di tipografia prescindono dal particolare programma di composizione utilizzato e non hanno dunque prerequisiti, le parti dell'articolo in cui si mostrano applicazioni richiedono conoscenze elementari di L^AT_EX che possono essere ottenute da qualunque guida di base (AUTORI VARI, 2007; BAUDOIN, 1998; OETIKER *et al.*, 2000; THE TUTORIAL TEAM, 2000) o da un manuale cartaceo (GOOSSENS *et al.*, 1995; KOPKA e DALY, 1995).

2 Introduzione

I riferimenti bibliografici rappresentano una parte importante, talvolta fondamentale, di qualunque scritto accademico. In passato la redazione della bibliografia presentava non poche difficoltà dovute soprattutto al fatto che la numerazione delle voci veniva fatta manualmente e, quindi, modifiche in corso d'opera potevano comportare interventi in più punti del documento. L^AT_EX, che è stato pensato per documenti di questo tipo, offre una varietà di strumenti per gestire la bibliografia in modo automatico, rendendo il compito dell'autore molto più facile.

Nel par. 3 viene spiegato come comporre una bibliografia con L^AT_EX, partendo dalle basi ed arri-

vando fino a personalizzazioni avanzate. Nel par. 4 viene invece affrontato il tema degli stili bibliografici sia per quando riguarda i riferimenti nel testo che per l'elenco delle voci alla fine del documento.

Il presente articolo non entra nei dettagli delle regole per i documenti tecnici, per le quali ci si riferisce alle norme vigenti. In Italia fino al 2007 è stata in vigore la norma UNI 10168 (1993), successivamente sostituita dalla norma ISO 690 (1987).

3 La bibliografia con L^AT_EX

L^AT_EX offre due modalità per comporre la bibliografia: la modalità automatica che si avvale del programma BIB_TE_X e si appoggia a *database* bibliografici esterni (vedi par. 3.2), e quella manuale che utilizza l'ambiente *thebibliography* e ingloba tutte le informazioni bibliografiche all'interno del file sorgente *.tex* (vedi par. 3.3). Indipendentemente dalla strategia adottata per raccogliere le informazioni bibliografiche, le citazioni nel testo vengono effettuate con gli stessi comandi, come spiegato nel par. 3.1.

3.1 Riferimenti nel testo

Indipendentemente da come vengono definite le voci bibliografiche (si vedano i par. 3.2 e 3.3), il loro richiamo nel testo viene effettuato con il comando `\cite{chiave}` (e sue varianti) dove *chiave* è la *citekey* nel file *.bib* (se si usa l'ambiente BIB_TE_X, par. 3.2) oppure la chiave della voce `\bibitem` (se si usa l'ambiente *thebibliography*, par. 3.3). Quando il sorgente viene compilato, `\cite{chiave}` viene collegato ipertestualmente (se si usa il pacchetto *hyperref*) al rispettivo `\bibitem` e sostituito dall'appropriato riferimento (numerico, autore-anno o in nota a seconda dello stile scelto).

Per effettuare citazioni multiple è sufficiente inserire tutte le chiavi bibliografiche separate da virgole all'interno dello stesso comando `\cite`, come ad esempio

```
\cite{Goossens1995,Kopka1995}
```

che produce

```
(Goossens et al., 1995; Kopka and Daly, 1995)
```

Per far apparire in bibliografia una voce che non ha un riferimento nel testo è sufficiente usare il comando `\nocite{chiave}` prima della bibliografia. Per richiamare tutte le voci non citate è possibile usare `\nocite{*}`.

*Ringrazio in primo luogo Massimiliano Dominici che, oltre ad avermi consigliato su molti aspetti dell'articolo, ha anche scritto il par. 6 su *biblatex* e Gustavo Cevolani che ha scritto il par. 6.4 sull'indice analitico con *biblatex*. Desidero inoltre ringraziare Valeria Angeli, Claudio Beccari, Caterina Mori e Gianluca Pignalberi per gli utili suggerimenti forniti nelle fasi di stesura e revisione di questo articolo.

3.2 Composizione automatica con BibT_EX

BibT_EX è un programma esterno a L^AT_EX che permette di creare la bibliografia a partire da un *database* esterno (file `.bib`) che può essere condiviso da più documenti L^AT_EX. Questo è spesso più conveniente rispetto ad inserire le informazioni bibliografiche direttamente dentro il sorgente `.tex` di ogni documento. BibT_EX, descritto nei seguenti paragrafi, presenta molti altri vantaggi rispetto all'ambiente `thebibliography` tra cui la possibilità di formattare ed ordinare automaticamente le voci della bibliografia.

3.2.1 Funzionamento di BibT_EX

Per utilizzare BibT_EX è necessario:

1. creare un *database* bibliografico `.bib`;
2. nel documento `.tex` vanno inclusi: la dichiarazione dello stile `.bst` da utilizzare con il comando `\bibliographystyle`, il riferimento al database `.bib` con il comando `\bibliography` e le citazioni nel testo con i comandi `\cite` e simili;

```
\documentclass{...}
\usepackage{...}

\begin{document}
Leggi \citet{Kopka1995}.

\bibliographystyle{plainnat}
\bibliography{database}
\end{document}
```

3. compilare il file nel seguente ordine (si assuma che il file principale si chiami `documento.tex`)

```
latex documento
bibtex documento
latex documento
latex documento
```

Alla prima compilazione con L^AT_EX, il comando `\bibliographystyle` scrive nel file `.aux` quale sia lo stile `.bst` da usare, ogni `\cite` scrive una nota sulla citazione nel file `.aux` e il comando `\bibliography` scrive nel file `.aux` il nome del *database* `.bib` da usare. In questo stadio L^AT_EX non sta ancora sostituendo i `\cite` presenti nel testo: se si apre il file `.dvi` si troverà un punto interrogativo per ogni `\cite` e nel `.log` sarà presente un *warning* dovuto a riferimenti sconosciuti.

Quando si lancia BibT_EX, questo leggerà il file `.aux` tenendo conto dello stile bibliografico `.bst` e del *database* `.bib` specificati nel documento `.tex`. A questo punto, per ogni `\cite` scriverà le informazioni presenti nel `.bib`, formattate in base allo stile `.bst`. Alla fine, se lo stile lo prevede, metterà in ordine le voci bibliografiche e scriverà il risultato nel file `.bbl`. Il file `.bbl` contiene le voci bibliografiche organizzate in un ambiente `thebibliography` prodotto da BibT_EX formattando i contenuti del file `.bib` in base alla formattazione impostata dallo

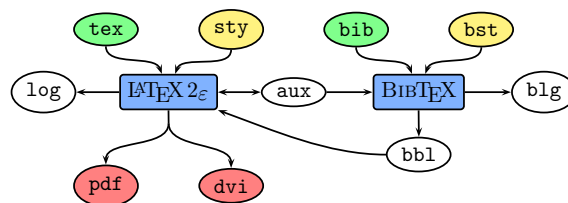


FIGURA 1: Flusso dei dati tra L^AT_EX and BibT_EX per creare un `.pdf` o un `.dvi` (■) a partire dai file di dati `.tex` e `.bib` (■) e dai file di stile `.sty` e `.bst` (■).

stile `.bst`. Questo file viene letto tutte le volte che si compila il file sorgente `.tex`.

Quando si compila nuovamente con L^AT_EX, questo produrrà un *warning* dovuto al fatto che le citazioni sono ancora non definite, ma, giunto al comando `\bibliography` leggerà il file `.bbl`. A questo punto, per ogni `\bibitem` trovato scriverà una nota nel file `.aux`.

Alla successiva compilazione, L^AT_EX troverà le definizioni per ogni `\cite` dentro il file `.aux`. Il file `.blg` è il log prodotto da BibT_EX, analogo del file `.log` per L^AT_EX.

Questo flusso, che va ripetuto se vengono fatte modifiche alle citazioni nel testo, al database `.bib` oppure allo stile `.bst`, è rappresentato nella Fig. 1.

3.2.2 Struttura dei database bibliografici

I file `.bib` sono *database* bibliografici per BibT_EX che contengono le informazioni di ogni voce bibliografica. Un esempio potrebbe essere

```
@BOOK{Kopka1995,
  title = {A Guide to {\LaTeX} -- Document Preparation
    for Beginners and Advanced Users},
  publisher = {Addison-Wesley},
  year = {1995},
  author = {H. Kopka and P. W. Daly},
}
```

Ogni tipo di voce (BOOK, ARTICLE, PROCEEDINGS, ecc.) offre molti campi in aggiunta a quelli mostrati nell'esempio (`title`, `publisher`, `year`, `author`). Per approfondimenti si faccia riferimento a PATASHNIK (1998).

3.2.3 Regole per l'uso di BibT_EX

La creazione delle voci per BibT_EX è molto intuitiva ma devono essere rispettate alcune semplici regole. Di seguito si riportano le più comuni, per le altre si faccia riferimento a PATASHNIK (1998).

Lettere maiuscole

Gli stili di BibT_EX di solito controllano l'uso delle lettere maiuscole, specialmente nel titolo. Questo comportamento è del tutto ragionevole perché assicura uniformità nella formattazione delle voci ma può causare problemi in situazioni specifiche in cui si usino acronimi, formule chimiche, ecc. La soluzione consiste semplicemente nel racchiudere la parola o le lettere che vogliamo controllare direttamente tra parentesi graffe come nell'esempio seguente per "CO":

```
title = {Inquinamento da {CO}},
```

Se si vuole avere un particolare utilizzo delle maiuscole nei titoli (ad esempio si vogliono tutte le iniziali maiuscole) è consigliabile scegliere o creare uno stile `.bst` che lo faccia automaticamente invece che forzare a mano la gestione, ad esempio racchiudendo tutti i titoli tra parentesi graffe come nel seguente esempio

```
title = {{La Tradizione dei Classici dall'Antichità ai  
Tempi Moderni}},
```

Se, infatti, in un secondo tempo si volesse modificare l'uso delle maiuscole, è possibile farlo semplicemente cambiando stile invece che modificando a mano tutte le voci del database `.bib`.

Comandi

Proprio per il fatto che B_IB_TE_X reinterpreta maiuscole e minuscole in base allo stile `.bst` in uso, in alcuni casi i comandi L^AT_EX possono dare errori di compilazione. Se ad esempio si usa uno stile in cui solo la prima lettera del titolo è maiuscola e in uno dei titoli compare il comando `\LaTeX`, questo verrebbe interpretato come `\latex` e restituirebbe il seguente errore

```
! Undefined control sequence.
```

La soluzione consiste semplicemente nel racchiudere i comandi tra parentesi graffe come `{\LaTeX}`.

Accenti e caratteri speciali

Un problema analogo si presenta quando si usano i comandi per gli accenti ed i caratteri speciali come “ö” (`\"o`), “ç” (`\c{c}`), “ñ” (`\~{n}`), ecc. In tutti questi casi è sufficiente racchiudere tra parentesi graffe il comando.

```
title = {Come scrivere il curriculum vit{\ae}},
```

Elenco dei nomi

Quando è presente più di un nome nei campi `author` o `editor`, questi vanno separati con `and` preceduto e seguito da uno spazio come:

```
author = {Autore1 and Autore2},
```

Ogni nome è costituito da quattro parti: Nome, von, Cognome, Jr. Ogni parte essere composta da una lista di nomi (anche abbreviati). Ad eccezione del cognome, le altre parti possono non essere presenti. B_IB_TE_X accetta due possibili modi di scrivere un nome:

- **Nome von Cognome:** ad esempio se il nome è “Pico della Mirandola” va scritto come `Pico della Mirandola`.
- **von Cognome, Jr, Nome:** ad esempio se il nome è “Pico della Mirandola II” va scritto come `della Mirandola, II, Pico`.

La seconda forma è sempre preferibile perché la prima non può essere usata nel caso che sia presente la parte Jr oppure quando il cognome è formato da più parole e la parte von non è presente. Gli stili bibliografici `.bst` si preoccupano di mantenere esteso oppure abbreviare il nome. Questo non toglie che, se si vogliono utilizzare solamente nomi abbreviati, si componga il file `.bib` già in questa forma. Ad esempio per scrivere abbreviato “Lapo Filippo Mori” basta inserire

```
author = {Mori, L. F.},
```

Se l'elenco dei nomi degli autori o degli editori è troppo lungo, può essere terminato da `and others` che viene poi formattato dagli stili standard come “et al.”:

```
author = {Conte, G. B. and Pianezzolla, E. and Chiesa,  
P. and Rossi, G. and others},
```

URL

Gli stili standard di B_IB_TE_X non prevedono un campo per gli indirizzi di siti web. Se si usa uno di questi stili, una possibile soluzione consiste nell'usare il campo `howpublished` della tipologia `misc` come nell'esempio seguente

```
@MISC{...,  
...,  
howpublished = {\url{http://...}},  
}
```

Una soluzione migliore consiste nell'usare stili che prevedano il campo `url`. Ad esempio gli stili `plainnat`, `unsrnat` e `abbrevnat` che sono un'estensione degli stili standard di B_IB_TE_X da usarsi con `natbib` (vedi il par. 5). Anche gli stili personalizzati creati con `makebst` (vedi il par. 4.2.2) prevedono il campo `url`. Il pacchetto `babelbib` (vedi il par. 3.7.3), da usarsi per avere bibliografie multilingua, provvede come `natbib` degli stili che estendono quelli standard e che prevedono il campo `url`. In tutti questi casi la sintassi è la seguente

```
@ARTICLE{...,  
...,  
url = {http://...},  
}
```

In ogni caso è consigliabile utilizzare anche il pacchetto `breakurl`.¹ Quando si riporta l'indirizzo di un sito web si possono avere dei problemi quando questo è troppo lungo o vicino al margine destro. Se il documento viene compilato con pdfL^AT_EX non ci sono problemi dato che il driver è in grado di spezzare i link su più righe. Quando però si usa il driver `dvips` (ovvero quando si segue il percorso L^AT_EX → .dvi → PostScript → .pdf), a causa di ragioni interne, gli indirizzi web non possono essere spezzati su più righe. Nei casi migliori questo causa dei poco estetici paragrafi con `underfull` e `overfull` ma nei casi peggiori i link possono

1. Il pacchetto `breakurl` è stato scritto da Vilar Camara Neto e l'ultima versione è del 2006.

anche uscire dai margini della pagina. Il pacchetto `breakurl` risolve questo problema spezzando i link in più parti (ognuna delle quali mantiene il collegamento ipertestuale nel `.pdf`) invece che produrli come un unico pezzo inescabibile. Per fare questo il pacchetto provvede il comando `\burl` ed inoltre interpreta gli eventuali `\url` come `\burl`.

Mesi

BIB \TeX ha delle macro per la gestione automatica del nome del mese inserito con il campo `month`; queste macro gestiscono in automatico la forma estesa (March) o quella abbreviata (Mar.) e la lingua (Marzo) in base allo stile `.bst` usato. Per usufruirne, è necessario introdurre il mese con la forma abbreviata inglese.²

3.2.4 Creazione e gestione dei database bibliografici

Come risulta evidente dall'esempio riportato nel par. 3.2, i file `.bst` possono essere facilmente creati manualmente utilizzando un qualunque editor di testo. Esistono tuttavia una serie di programmi dedicati alla creazione e gestione di *database* `.bib`. Tra questi si segnalano BIB \TeX Mng³ (*shareware* per Windows), BibDesk⁴ (*open-source* per Mac OS X) KBIB \TeX ⁵ (*open-source* per Linux), Pybliographer⁶ (*open-source* per Linux), bibliographer⁷ (*freeware* per Linux), Bibwiki⁸ (basato su internet), cb2Bib⁹ (*freeware* per Windows e Linux), Zotero¹⁰ (plugin *open-source* multiplatforma per Firefox), e JabRef¹¹ (*open-source* multiplatforma). Degni di nota sono gli ultimi tre programmi citati che verranno trattati nei seguenti paragrafi.

cb2Bib

cb2bib (che sta per “clipboard to BIB \TeX ” ovvero “dalla *clipboard* a BIB \TeX ”) è un programma non per la gestione dei *database* bibliografici, ma per l'estrazione di informazioni delle voci bibliografiche da fonti non formattate o non standard come `.pdf`, siti e email. cb2bib legge il testo presente nella *clipboard* e lo analizza seguendo degli schemi predefiniti.¹² Se il riconoscimento secondo questi

2. La forma abbreviata inglese riporta le prime tre lettere del mese: jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec.

3. <http://www.latexsoft.com/bibtexmng.htm>

4. <http://bibdesk.sourceforge.net/>

5. <http://www.unix-ag.uni-kl.de/~fischer/kbibtex/>

6. <http://www.pybliographer.org/Welcome/>

7. <http://bibliographer.homelinux.net/>

8. <http://wolfgang.plaschg.net/bibwiki/>

9. http://www.molspaces.com/d_cb2bib-overview.php

10. <http://www.zotero.org/>

11. <http://jabref.sourceforge.net/>

12. Sono già implementati schemi di riconoscimento per i seguenti formati: schede degli articoli della PNAS (<http://www.pnas.org/>), schede dello ISI Web of Knowledge (<http://isiknowledge.com/>), *abstract* di Wiley InterScience Journal (<http://www.interscience.wiley.com/>), pubblicazioni della American Chemical Society (<http://pubs.acs.org/>), schede di ScienceDirect

schemi va a buon fine, le informazioni vengono inserite nei rispettivi campi, altrimenti l'utente può effettuare l'estrazione manuale.

Zotero

Zotero è un plugin *open-source* multiplatforma (Windows, Linux e Mac OS X) per il browser Firefox¹³ che permette di raccogliere, organizzare e analizzare riferimenti bibliografici. Data la sua natura di estensione di un *web browser*, Zotero è particolarmente utile per estrarre riferimenti bibliografici da pagine web. Le “librerie” di riferimenti possono poi essere esportati come `.pdf` (potendo formattare le voci con un certo numero di stili predefiniti), file di testo, database per BIB \TeX e molti altri formati. In aggiunta, Zotero si interfaccia anche con programmi di scrittura WYSIWYG¹⁴ come Microsoft Word¹⁵ e OpenOffice.¹⁶ Zotero permette di effettuare ricerche avanzate sulle proprie librerie e di importare voci da un gran numero di formati. Caratteristica unica di Zotero è la possibilità di accedere alle proprie librerie da qualunque computer per mezzo di internet, oltre che poter essere usato *offline*.

JabRef

JabRef è un programma *open-source* multiplatforma (Windows, Linux e Mac OS X) per la creazione e gestione di *database* di voci bibliografiche in formato BIB \TeX basato su Java VM. La creazione delle voci avviene attraverso la compilazione di maschere i cui campi dipendono dal tipo di riferimento (book, article, proceedings, ecc.); le maschere possono anche essere personalizzate dall'utente aggiungendo e togliendo campi.

JabRef permette di effettuare ricerche di articoli sui motori Medline¹⁷ e CiteSeer¹⁸ e di importare voci bibliografiche da archivi in numerosi formati tra cui BibTeX, CSA, Refer/Endnote, ISI Web of Science, SilverPlatter, Medline/Pubmed (xml), Scifinder, OVID, INSPEC, Biblioscape, Sixpack, JSTOR and RIS. Il programma offre funzioni avanzate di ricerca e catalogazione delle voci bibliografiche con parole chiave. Permette anche di stampare le voci o esportarle nei formati HTML, Docbook, BibTeX, MODS, RTF, Refer/Endnote and OpenOffice.

JabRef permette di creare automaticamente le BIB \TeX keys (ad esempio prendendo il cognome

(<http://www.sciencedirect.com/>), voci della Digital Bibliography & Library Project (<http://dblp.uni-trier.de/>), articoli di Nature dalle informazioni contenute nel primo paragrafo (<http://www.nature.com/nature/>), schede dello IOP Electronic Journals (<http://www.iop.org/EJ/>), JSTOR (<http://www.jstor.org/>), formato ISI (<http://scientific.thomson.com/isi/>), formato RIS.

13. <http://www.mozilla.com/firefox/>

14. Acronimo per “What You See Is What You Get”.

15. <http://office.microsoft.com/>

16. <http://www.openoffice.org/>

17. <http://www.ncbi.nlm.nih.gov/sites/entrez>

18. <http://citeseer.ist.psu.edu/>

me del primo autore e l'anno di pubblicazione) e di inserire le citazioni in diversi editor di testo tra cui LyX, Kile and WinEdt. Il programma si interfaccia con altre applicazioni esterne come programmi per la visualizzazione di PDF/PS, permettendo anche di associare alle voci bibliografiche un collegamento al rispettivo file .pdf. Ad ogni voce è anche possibile associare un url o un DOI;¹⁹ in entrambi i casi, JabRef apre il *web browser* sulla corrispondente pagina.

Fonti bibliografiche su internet

Molti siti, sia di riviste che di *database* bibliografici, permettono di esportare voci bibliografiche direttamente nel formato Bib_TE_X.

Tra le riviste si ricordano il portale di ACM,²⁰ Science,²¹ PNAS,²² The Journal of Chemical Physics.²³

Tra i *database* bibliografici si ricordano Lead2Amazon²⁴ (sito che usa Amazon.com, .ca, .co.uk, .de, .fr e .co.jp per generare automaticamente voci Bib_TE_X), Google Scholar²⁵ (motore di ricerca di Google dedicato a pubblicazioni scientifiche; selezionare "Bib_TE_X" tra le "Scholar Preferences"), BibSonomy²⁶ (portale per condividere link a pubblicazioni), CiteSeer²⁷ (motore di ricerca pubblico e biblioteca digitale per articoli scientifici), CiteULike²⁸ (portale per condividere link a pubblicazioni), The Collection of Computer Science Bibliographies²⁹ (*database* bibliografico di informatica con oltre due milioni di voci), HubMed³⁰ (interfaccia alternativa di PubMed³¹ che offre anche la possibilità di esportare le voci in formato Bib_TE_X), T_EXMed³² (altra interfaccia alternativa di PubMed per esportare le voci in formato Bib_TE_X).

3.3 Composizione manuale con thebibliography

L'ambiente thebibliography va posizionato nel corpo del documento sorgente nel punto in cui si vuole la bibliografia, tipicamente subito prima di `\end{document}`.

19. DOI è l'acronimo di (Digital Object Identifier) e rappresenta il futuro per i riferimenti a pubblicazioni in formato elettronico. Rispetto agli url ha il vantaggio di essere associato all'oggetto a cui si riferisce (articolo scientifico) e non al luogo dove esso è memorizzato (sito web), garantendo una maggiore longevità del collegamento. Maggiori informazioni possono essere trovate su <http://www.doi.org/>.

20. <http://portal.acm.org/>

21. <http://www.sciencemag.org/>

22. <http://www.pnas.org/>

23. <http://jcp.aip.org/>

24. <http://keijisaito.info/lead2amazon/e/>

25. <http://scholar.google.com/>

26. <http://www.bibsonomy.org/>

27. <http://citeseer.ist.psu.edu/>

28. <http://www.citeulike.org/>

29. <http://liinwww.ira.uka.de/bibliography/>

30. <http://www.hubmed.org/>

31. <http://pubmed.gov/>

32. <http://www.sbg.bio.ic.ac.uk/~mueller/TeXMed/>

```
\begin{thebibliography}{argomento}
...
\end{thebibliography}
```

L'argomento dell'ambiente thebibliography definisce la lunghezza massima delle etichette nell'elenco delle voci della bibliografia. Se non si usano etichette personalizzate (ovvero si lascia a L^AT_EX il compito di numerare in ordine crescente le voci della bibliografia) generalmente si mette come argomento "9" nel caso in cui le voci siano meno di dieci, "99" se meno di cento, ecc. Se, invece, si usano etichette personalizzate (ad esempio [Mori 06]), va messa l'etichetta più lunga.

L'ambiente thebibliography funziona in modo simile all'ambiente itemize. Ogni voce della bibliografia inizia con il comando `\bibitem` che prende come parametro la stringa che identifica il riferimento bibliografico (analogamente a `\label`). Di seguito vanno riportate le informazioni del riferimento (ad esempio autore, titolo, curatore, anno di pubblicazione) con la loro formattazione e punteggiatura.

```
\begin{thebibliography}{9}

\bibitem[Kopka1995]{\textsc{Kopka}, H. e \textsc{Daly},
P.-W. (1995). \emph{A Guide to {\LaTeX}} -- Document
Preparation for Beginners and Advanced Users}.
Addison-Wesley.

\end{thebibliography}
```

produce la Fig. 2.

3.4 Da Bib_TE_X a thebibliography

Alcune riviste impongono che gli articoli abbiano la bibliografia creata con l'ambiente thebibliography, ovvero che sia contenuta nel file sorgente .tex. Anche in questi casi è possibile utilizzare la bibliografia con un *database* esterno .bib da gestire con Bib_TE_X. Infatti, Bib_TE_X non fa altro che estrarre dal *database* .bib le informazioni riguardanti le voci citate nel testo e formattarle secondo lo stile .bst scelto per formare il corrispondente ambiente thebibliography, che viene inserito nel file .bb1. Sfruttando questa logica, una volta terminato il lavoro, è sufficiente compilare il *database* .bib con Bib_TE_X ottenendo il file .bb1. Questo file contiene tutte le voci bibliografiche costruite con l'ambiente thebibliography ed è quindi sufficiente copiare tutto il contenuto del file ed incollarlo nel file sorgente .tex.

3.5 Da thebibliography a Bib_TE_X

Non esistono metodi automatici per convertire il contenuto di thebibliography in formato Bib_TE_X. Spesso può convenire importare ogni singola voce da *database* su internet. In alternativa si può provare ad convertire il testo di thebibliography con cb2Bib. Per entrambi i metodi si faccia riferimento al par. 3.2.4.

References

- [1] KOPKA, H. e DALY, P. W. (1995). *A Guide to L^AT_EX – Document Preparation for Beginners and Advanced Users*. Addison-Wesley.

FIGURA 2: Bibliografia ottenuta con l'ambiente thebibliography.

3.6 Quale metodo usare

Automatizzare la gestione della bibliografia con BIB_T_EX presenta qualche svantaggio:

- aumenta la complessità dell'ambiente di lavoro;
- nonostante che sia uno strumento flessibile, BIB_T_EX non permette di personalizzare a piacere il contenuto delle voci bibliografiche.

Tuttavia i vantaggi associati al suo uso superano gli svantaggi:

- automatizza funzioni noiose come ordinare le voci della bibliografia;
- rende automatica la loro formattazione;
- esistono molti strumenti per creare e gestire database per BIB_T_EX.

3.7 Specialità

3.7.1 Cambiare il nome della bibliografia

Il nome della bibliografia dipende dalla classe e dalla lingua usata. Le classi **report** e **book** assegnano al titolo della bibliografia il comando `\bibname` a cui il pacchetto **babel** assegna a sua volta il valore “Bibliography” in inglese e “Bibliografia” in italiano. La classe **article**, invece, assegna al titolo della bibliografia il comando `\refname` a cui il pacchetto **babel** assegna a sua volta il valore “References” in inglese e “Riferimenti bibliografici” in italiano.

Per modificare i nomi che vengono assegnati di *default* è sufficiente utilizzare il comando `\renewcommand` come nel seguente esempio:

```
\renewcommand\bibname{Riferimenti utili}
```

3.7.2 Bibliografie multiple

Diversi sono i motivi per cui si possono volere bibliografie multiple in uno stesso documento e analogamente differenti sono i pacchetti che permettono di farlo.

Il pacchetto **chapterbib**³³ permette di creare bibliografie separate per ogni documento inserito con un comando `\include`, e non solo per i capitoli come farebbe pensare il nome. L'utilizzo del pacchetto è particolarmente semplice in quanto le

33. Il pacchetto **chapterbib** è stato scritto da Donald Arseneau e l'ultima versione è del 2004.

bibliografie separate sono create automaticamente quando si utilizza il comando `\include` senza la necessità di dover utilizzare comandi *ad hoc* per i richiami nel testo. In questo caso BIB_T_EX deve essere lanciato sui singoli file inclusi e non sul documento master.

Il pacchetto **bibunits**³⁴ permette di creare bibliografie separate per differenti parti del documento come capitoli, sezioni, ecc. Il pacchetto funziona separando le citazioni che BIB_T_EX deve poi processare. In aggiunta alle bibliografie per le singole parti di testo, al termine del documento può essere riportata una bibliografia generale. Anche l'utilizzo di questo pacchetto è molto semplice dato che è sufficiente indicare nel preambolo quali sono le parti per cui devono essere create bibliografie separate con il comando `\bibliographyunit{unità}` dove *unità* può essere **chapter**, **section**, ecc. Anche in questo caso non servono comandi *ad hoc* per i richiami nel testo. Per ogni bibliografia separata (*bibunit*), il pacchetto genera un file `.aux` che deve essere compilato con BIB_T_EX.

Il pacchetto **multibib**³⁵ permette di creare bibliografie multiple in uno stesso documento ma, in aggiunta ai pacchetti **bibunits** e **chapterbib** che permettono di creare bibliografie multiple per differenti parti del documento, **multibib** permette di posizionare le bibliografie multiple dove si desidera. Ogni bibliografia può avere un proprio stile `.bst` e un proprio database `.bst` e per ognuna BIB_T_EX crea un proprio file di output `.bbl`. A differenza dei precedenti pacchetti, **multibib** richiede dei comandi *ad hoc* per i richiami nel testo: in pratica per ogni categoria deve essere usato un differente tipo di *cite* e dei comandi specifici anche per la `\bibliographystyle` e `\bibliography`. Per questo motivo è difficile separare la bibliografia di un documento pensato per averne una unica dato che questo comporta dover modificare tutti i comandi *cite* nel testo. Come per **bibunits**, il pacchetto **multibib** genera un file `.aux` per ogni bibliografia e questo deve essere compilato con BIB_T_EX.

Il pacchetto **multibbl**³⁶ funziona in modo molto simile a **multibib** ma non permette di ottenere

34. Il pacchetto **bibunits** è stato scritto da Thorsten Hansen e l'ultima versione è del 2004.

35. Il pacchetto **multibib** è stato scritto da Thorsten Hansen e l'ultima versione è del 2004.

36. Il pacchetto **multibbl** è stato scritto da Apostolos Syropoulos e l'ultima versione è del 2004.

una bibliografia generale in aggiunta a quelle per argomenti.

I pacchetti `bibtopic` e `splitbib` permettono di ottenere risultati simili a quelli di `multibib` e `multibbl` ma con una strategia differente: invece di richiedere dei comandi *ad hoc* per i richiami nel testo, chiedono che vengano definite a priori quali sono le voci di ogni singola bibliografia. Il pacchetto `bibtopic`³⁷ richiede che ogni singola bibliografia abbia un proprio database `.bib`. Questo risulta essere un enorme vantaggio nel caso in cui si usino programmi intelligenti per la gestione dei file `.bib` come `JabRef` con cui è molto semplice creare differenti *database* a partire da uno singolo.³⁸ Il pacchetto `splitbib`³⁹, invece, richiede che si dichiari quali voci appartengono ad ogni categoria nel preambolo del documento. Anche questo approccio permette di modificare facilmente un documento pensato per avere una bibliografia unica.

3.7.3 Bibliografie multilingua

Il pacchetto `babelbib`,⁴⁰ usato in coppia con il pacchetto `babel`, permette di generare bibliografie multilingua in cui:

- ogni voce viene scritta in una determinata lingua, oppure
- tutta la bibliografia viene formattata in una lingua scelta dall'autore.⁴¹

Mentre per il secondo caso (bibliografia con tutte le voci in una medesima lingua) è possibile anche scriversi agevolmente uno stile `.bst` come spiegato nel par. 4.2.2, per il primo (bibliografia con voci in diverse lingue) il pacchetto offre un eccellente aiuto: per assegnare la lingua ad ogni voce del *database* `.bib` è sufficiente specificarla nel campo `language` come nel seguente esempio:

```
@BOOK{Kopka1995,
  title = {A Guide to {\LaTeX} -- Document Preparation
    for Beginners and Advanced Users},
  publisher = {Addison-Wesley},
  year = {1995},
  author = {H. Kopka and P. W. Daly},
  language = {english},
}
```

4 Gli stili bibliografici

Con stile bibliografico si possono intendere due cose:

37. Il pacchetto `bibtopic` è stato scritto da Pierre Basso e Stefan Ulrich e l'ultima versione è del 2002.

38. Nel caso specifico di `JabRef` per copiare alcune voci da un database all'altro è sufficiente copiarle ed incollarle.

39. Il pacchetto `splitbib` è stato scritto da Nicolas Markey e l'ultima versione è del 2005.

40. Il pacchetto `babelbib` è stato scritto da Harald Harders e l'ultima versione è del 2004.

41. Le lingue attualmente supportate dal pacchetto sono l'afrikaans, il danese, il belga, l'inglese, il francese, il tedesco, l'italiano, il norvegese, il portoghese, lo spagnolo e lo svedese.

- lo stile delle voci della bibliografia (generalmente presente alla fine del documento),
- lo stile dei riferimenti che compaiono nel testo.

Le tre principali famiglie di stili per i riferimenti nel testo (numerato, autore-anno, in nota) sono trattate nel par. 4.1 dove sono anche mostrati esempi sia di utilizzo in generale che nello specifico con L^AT_EX. Sebbene la scelta dello stile dei riferimenti influenzi almeno in parte lo stile delle voci bibliografiche, L^AT_EX permette di separare i due aspetti: lo stile della bibliografia può essere controllato con il file `.bst` come spiegato nel par. 4.2.

4.1 Stile dei riferimenti nel testo

Esistono tre principali famiglie di stili di riferimenti nel testo: numerato, autore-anno ed in nota. Ogni disciplina adotta degli standard specifici per soddisfare le proprie esigenze e l'uso che viene fatto della bibliografia (GARCIA, 2007).

4.1.1 Numerato

I riferimenti numerati appaiono generalmente in parentesi quadre e seguono una numerazione araba (ad esempio [1]).

Il vantaggio principale di questo tipo di riferimenti è che possono essere usati sia quando si fa un riferimento diretto (ad esempio 'si veda [1] come riferimento per la teoria') che quando si fa un riferimento indiretto (ad esempio 'questo è già stato dimostrato in precedenza [1]'). Altro vantaggio è la possibilità di usare i riferimenti di questo tipo insieme a parentesi tonde in frasi come '(per approfondimenti si veda [1])'. In sostanza la forma del riferimento, che in L^AT_EX viene ottenuto con il semplice `\cite`, è indipendente dalla frase in cui compare e questa flessibilità ha reso questo stile piuttosto diffuso.

4.1.2 Autore-anno

Quando si cita un articolo per dare il riferimento ad un teorema o teoria che non si riporta per intero non ha molta importanza che il lettore sappia chi ha scritto l'articolo e quando. Il lettore interessato può ottenere tutte queste informazioni dall'elenco dei riferimenti bibliografici riportato alla fine dello scritto. Questa situazione fa sì che lo stile numerato sia il più diffuso nelle discipline scientifiche. Nelle discipline letterarie e filosofiche, però, il riferimento ad un autore piuttosto che ad un altro e ad un periodo storico piuttosto che ad un altro ha di per sé un significato ed è importante che il lettore possa ottenere queste informazioni direttamente dal testo. Per questo motivo, nelle discipline non scientifiche è molto diffuso il così detto stile autore-anno che cerca di riassumere nel riferimento bibliografico tutte le informazioni essenziali, ovvero il cognome del primo autore e l'anno di pubblicazione: un esempio potrebbe essere (Mori et al., 2006).

Il fatto che le parentesi tonde abbiano un loro significato fa sì che possano nascere fraintendimenti. Ad esempio è possibile fare riferimenti generali come ‘questo è già stato mostrato (Mori, 2006).’ ma non dei riferimenti diretti come ‘(Mori, 2006) ha mostrato che’. Per questo, gli stili autore-anno tendono a mostrare numerose varianti per i riferimenti nel testo per risolvere problemi grammaticali o estetici. Ecco alcuni esempi:

- ‘Questo è già stato mostrato (Mori, 2006).’
- ‘Per questo si veda Mori (2006).’
- ‘(si veda a tal proposito [Mori, 2006])’

Non essendo possibile automatizzarla, la scelta del tipo di riferimento è lasciata all’autore. La presenza di tante possibili forme di riferimento comporta la necessità di molti comandi differenti al posto del tipico `\cite` e quindi i pacchetti LATEX che supportano le citazioni autore-anno offrono un numero molto ampio di comandi per la citazione.

Pacchetti

Per le citazioni autore-anno i tre pacchetti più diffusi sono **harvard**, **achicago** e **natbib**. I tre pacchetti offrono più o meno le stesse potenzialità (anche se **natbib** è molto più flessibile) ma i loro comandi seguono logiche differenti. Il pacchetto **harvard**⁴² assegna i nomi in base alla funzione logico-grammaticale che il riferimento ha nella frase. Per esempio quando il riferimento è un nome, si usa `\citenoun`. Il pacchetto **achicago**⁴³ assegna i nomi in base alla forma del riferimento. Se ad esempio la citazione deve contenere solo l’autore si usa `\citeA` (dove “A” sta per “author”) se deve contenere anche il nome si usa `\citeN` (dove “N” sta per “name”). Il pacchetto **natbib**,⁴⁴ basato sulla logica di **harvard**, è il pacchetto per la gestione di citazioni con lo stile autore-anno più flessibile e quindi verrà trattato estesamente nel par. 5.

4.1.3 Note a piè di pagina

In alcune discipline, prevalentemente umanistiche, si utilizzano i riferimenti a piè di pagina. Questo uso è diffuso soprattutto nelle riviste che non prevedono la lista dei riferimenti bibliografici alla fine degli articoli.

Pacchetti

Il pacchetto **footbib**⁴⁵ definisce il comando `\footcite` che mette i riferimenti bibliografici in apice racchiusi tra parentesi quadre (ad esempio

42. Il pacchetto **harvard** è stato scritto da Peter Williams e Thorsten Schnier e l’ultima versione è del 1996.

43. Il pacchetto **achicago** è stato scritto da Matt Swift e l’ultima versione è del 2001.

44. Il pacchetto **natbib** è stato scritto da Patrick Daly e l’ultima versione è del 2006.

45. Il pacchetto **footbib** è stato scritto da Eric Domenjoud e l’ultima versione è del 2004.

[1]). Le informazioni del riferimento bibliografico sono riportate a piè di pagina ma la numerazione dei riferimenti non segue quella delle note a piè di pagina. I comandi `\footcite` non interferiscono con i `\cite` ed è quindi possibile affiancare una bibliografia tradizionale a quella a piè di pagina.

Il pacchetto **opcit**⁴⁶ crea i riferimenti come vere e proprie note a piè di pagina, seguendone quindi la numerazione (ad esempio ¹). Il pacchetto tiene conto dei riferimenti che sono già stati citati per evitare la ripetizione delle stesse note, utilizzando formule convenzionali come “Idem” e “op. cit.”. Oltre al manuale che accompagna il pacchetto, si consiglia la lettura di GARCIA (2007).

Il pacchetto **jurabib**,⁴⁷ originariamente pensato per testi di diritto tedesco, offre molti strumenti per la manipolazione dei riferimenti. Analogamente ad **opcit**, il pacchetto **jurabib** crea i riferimenti bibliografici utilizzando le note a piè di pagina standard di LATEX.

Il pacchetto **natbib** mette a disposizione l’opzione **super** che, analogamente a **footbib**, produce voci bibliografiche con una numerazione indipendente dalle note a piè di pagina ma non racchiuse in parentesi quadre (ad esempio ¹). Le voci della bibliografia sono riportate al termine del documento e non a piè di pagina. Sebbene **natbib** offra meno opzioni per i riferimenti bibliografici a nota, rispetto agli altri due pacchetti ha il vantaggio di utilizzare la stessa sintassi per le note e per gli stili numerato o autore-anno. Questo permette di passare da uno stile all’altro semplicemente cambiando le opzioni con cui viene caricato il pacchetto senza intervenire sui comandi presenti nel testo del documento.

Il pacchetto **inlinebib**⁴⁸ può essere usato per i riferimenti bibliografici a piè di pagina ma è sconsigliato in quanto il pacchetto è piuttosto vecchio e non offre molte personalizzazioni per l’utente. Una limitazione su tutte è il fatto che può essere usato solamente con lo stile **indexing.bst**.

4.1.4 Approcci ibridi

Tra autore-anno e numerato

Una via di mezzo tra lo stile autore-anno, che produce riferimenti come ‘(Mori, 2006)’, e quello numerato, che produce riferimenti come ‘[1]’, è data dallo stile **alpha.bst** che produce riferimenti come ‘[Mor06]’. I risultati di questo stile sono dubbi dato che non è sintetico come lo stile numerato e al tempo stesso non riporta sufficienti informazioni se il nome dell’autore e l’anno di pubblicazione sono essenziali per la comprensione della frase (come avviene per lo stile autore-anno). **natbib** supporta questo stile.

46. Il pacchetto **opcit** è stato scritto da Federico Garcia e l’ultima versione è del 2006.

47. Il pacchetto **jurabib** è stato scritto da Federico Garcia e l’ultima versione è del 2006.

48. Il pacchetto **bibentry** è stato scritto da René Seindal e l’ultima versione è del 1995.

Tra autore-anno e in nota

In alcune riviste umanistiche le voci bibliografiche compaiono direttamente nel testo. Il pacchetto **bibentry**⁴⁹ permette di farlo ed inoltre impone al file **.bst** solo poche restrizioni quindi può in linea di massima essere usato con qualunque stile.

Il pacchetto **inlinebib**, pensato per i riferimenti a piè di pagina, può essere usato anche per avere i riferimenti bibliografici nel testo ma il suo uso è sconsigliato in quanto il pacchetto è piuttosto vecchio e non offre molte personalizzazioni per l'utente.

Anche il pacchetto **jurabib**, pensato per le note a piè di pagina, permette l'inserimento dei riferimenti nel testo e può essere usato con quattro stili: **jurabib.bst**, **jhuman.bst** e due del tipo 'Chicago'.

4.2 Stile delle voci bibliografiche

4.2.1 Stili esistenti

Ogni rivista e casa editrice, o quasi, ha le proprie regole per formattare la bibliografia sebbene le differenze tra uno stile e l'altro siano di solito piccole (come l'uso del neretto o del corsivo per il numero o il volume, l'uso delle parentesi e della punteggiatura). La maggioranza delle riviste hanno reso disponibile il proprio stile bibliografico per B^IB^T_EX e quindi quasi tutte le distribuzioni L^AT_EX hanno un certo numero di stili bibliografici già pronti; sul Comprehensive T_EX Archive Network (CTAN)⁵⁰ ne sono disponibili anche di più. Questi stili coprono un'ampia gamma di casistiche e quindi, nella maggior parte dei casi, sono sufficienti per formattare la propria bibliografia.

B^IB^T_EX viene distribuito con quattro stili (**plain**, **unsrt**, **abbrv** e **alpha**) creati dallo stesso autore del programma, Oren Patashnik. Questi stili presentano la grossa limitazione di non supportare il formato autore-anno (vedi il par. 4.1.2) che è invece molto diffuso in alcuni campi.

Il sito di Ken Turner⁵¹ riporta un testo di esempio con esempi dei più comuni stili bibliografici **.bst** disponibili. Un altro eccellente esame degli stili disponibili, ricco di esempi e organizzato in base al loro argomento, è riportato sul sito del Reed College.⁵²

In rete esistono molti più stili di quelli trattati nei due siti riportati sopra. Per vedere come si comporta uno stile **.bst** la cosa migliore è utilizzare il file **xampl.bib** che viene fornito insieme alla documentazione di B^IB^T_EX e poi compilare un file di questo tipo (si assume che lo stile bibliografico di prova si chiami **stileprova.bst**):

```
\documentclass{article}
\begin{document}
\bibliographystyle{stileprova}
\nocite{*}
\bibliography{xampl}
\end{document}
```

Dato che **xampl.bib** è piuttosto estremo in alcuni degli esempi di voci bibliografiche che riporta, il suo utilizzo farà anche vedere un interessante selezione dei messaggi di errore che B^IB^T_EX può produrre.

4.2.2 Personalizzare lo stile con **makebst**

Dato che moltissimi aspetti dello stile bibliografico possono essere personalizzati, non è detto che ne esista uno che soddisfa le proprie esigenze o quelle della rivista per cui si scrive (che hanno quasi sempre regole ferree sullo stile bibliografico ma non sempre forniscono un file di stile per B^IB^T_EX) e quindi talvolta può risultare necessario creare un proprio stile **.bst**. Altra evenienza in cui può risultare necessario creare uno stile bibliografico personalizzato è quando si voglia scrivere la bibliografia in una lingua diversa dall'inglese, dato che quasi tutti gli stili disponibili sono appunto in inglese. La realizzazione di uno stile personalizzato può essere fatta modificandone uno esistente ma questo richiede la conoscenza del linguaggio di programmazione B^IB^T_EX che è tutt'altro che intuitivo.

Patrick Daly, lo stesso autore di **natbib** e coautore dell'ottimo manuale KOPKA e DALY (1995), ha scritto il programma **makebst** che consente di creare in modo interattivo stili personalizzati per le voci bibliografiche **.bst** da usarsi con B^IB^T_EX (ed ovviamente compatibili con **natbib**). Il programma e tutti gli stili vengono di solito distribuiti come un unico pacchetto col nome di **custom-bib**.

Il cuore del programma è il file di stile generico **merlin.mbs** che contiene codice alternativo per numerosi aspetti dello stile bibliografico ed è analizzato estesamente in DALY (2007). Questo file è compilato dal programma **docstrip**⁵³ che produce il file di stile in base alle opzioni scelte. Dato che il numero di opzioni è molto alto (circa cento), il programma prevede un'interfaccia grafica tramite il file **makebst.tex**. Il primo passo consiste nel compilare con T_EX o L^AT_EX il file **makebst.tex**: a questo punto l'utente deve rispondere in modo interattivo alle domande poste che compaiono sullo schermo. All'inizio della compilazione viene chiesto quale file **.mbs** si desidera utilizzare e, in base alla scelta, viene costruito un file **batch** per **docstrip** per produrre uno stile con le caratteristiche bibliografiche contenute dal file **.mbs** stesso. Questo significa che le opzioni che l'utente può scegliere interattivamente dipendono dal file di stile **.mbs** selezionato. **merlin.mbs** è uno stile bibliografico

49. Il pacchetto **bibentry** è stato scritto da Patrick Daly, lo stesso autore di **natbib**, e l'ultima versione è del 2000.

50. <http://www.ctan.org/>

51. <http://www.cs.stir.ac.uk/~kjt/software/latex/showbst.html>

52. <http://web.reed.edu/cis/help/LaTeX/bibtexstyles.html>

53. **docstrip**, scritto da Frank Mittelbach, fa ormai parte di tutte le distribuzioni L^AT_EX.

di terza generazione che ha rimpiazzato l'iniziale `genbst.mbs` (introdotto nel novembre 1993) ed il suo corrispondente multilingua `babel.mbs`. A differenza dei due stili precedenti, in `merlin.mbs` tutte le parole come “editor” sono state sostituite da funzioni (in questo caso da `bb1.editor`) che assumono valori diversi a seconda della lingua selezionata (“curatore” in italiano, “editor” in inglese, “Redakteur” in tedesco e “redacteur” in francese, ecc.). `merlin.mbs` supporta come opzioni solo le opzioni `English` e `babel`; le definizioni per tutte le altre lingue sono contenuti in file `.mbs` separati (ad esempio `italian.mbs`). La scelta della lingua deve essere effettuata nelle prime fasi della compilazione di `makebst.tex`: quando viene chiesto che stile master utilizzare (“Enter the name of the MASTER file”) lasciare `merlin.mbs`, quando invece viene chiesto il file delle definizioni per la lingua (“Name of language definition file”) indicare il `.mbs` della lingua desiderata.

Se non esiste il file `.mbs` per la lingua che si vuole usare,⁵⁴ è sempre possibile selezionare come lingua `babel` che, invece di inserire la traduzione delle parole, inserisce al loro posto dei comandi (in questo caso `\bbleditor{}`) le cui definizioni devono essere contenute nel file `babelbst.tex`.

A parte la lingua, alcune delle personalizzazioni offerte da `merlin.mbs` sono:

- citazioni autore-anno o numeriche;
- criterio per ordinare le voci: in ordine di citazione, alfabetico, in ordine di pubblicazione cronologico diretto o inverso, ecc.;
- formato dei nomi degli autori: nome e cognome, iniziali e cognome, cognome ed iniziali, ecc.;
- numero dei nomi da riportare prima di usare “et. al.”;
- formattazione per il nome degli autori;
- posizione della data;
- formato del volume, numero e pagine per gli articoli;
- punteggiatura.

Al termine della procedura viene prodotto un file `.dbj` che deve essere compilato con LATEX per ottenere il file di stile `.bst`. Nel caso che vogliano essere apportate delle modifiche allo stile prodotto dalla procedura sopra descritta, può essere conveniente aprire e modificare il file `.dbj` piuttosto che ripartire da zero e rispondere alle domande interattive.

Per maggiori approfondimenti su `makebst` si consiglia la lettura di DALY (2003, 2007).

54. Attualmente vengono forniti i file `.mbs` per catalano, danese, esperanto, finlandese, francese, inglese, italiano, norvegese, olandese, polacco, portoghese, sloveno, spagnolo e tedesco.

5 Il pacchetto natbib

Il pacchetto `natbib` permette moltissime personalizzazioni ed è altamente consigliato; le più comuni verranno trattate in questo paragrafo ma si consiglia la lettura di DALY (2006) per un quadro più esaustivo sul pacchetto.

5.1 Stili bibliografici compatibili

`natbib` funziona solo con particolari stili bibliografici che ne supportano tutte le funzioni; i tre che vengono distribuiti insieme al pacchetto (`plainnat.bst`, `abbrvnat.bst` e `unsrnat.bst`) sostituiscono i corrispondenti stili standard di BIBTEX (`plain.bst`, `abbrv.bst` e `unsrt.bst`) portando il vantaggio di poter essere usati sia in modalità numerata (unica disponibile con i tre stili standard) e autore-anno. Molti altri stili che supportano il formato di `natbib` sono disponibili in rete e, soprattutto, questo formato è supportato da `makebst` (vedi par. 4.2.2) il che rende possibile la creazione di stili personalizzati.

5.2 Comandi per le citazioni nel testo

`natbib` offre due comandi di base per le citazioni (che vanno a sostituire il tradizionale `\cite`): `\citet` per le citazioni nel testo e `\citep` per le citazioni con parentesi. Entrambi hanno la versione asteriscata (`\citet*` e `\citep*`) che produce la lista degli autori completa al posto di quella abbreviata. Tutti i comandi prevedono due argomenti opzionali per aggiungere testo prima o dopo il riferimento. Analogamente a `\cite`, questi comandi permettono l'utilizzo di citazioni multiple. Vengono anche forniti comandi per produrre riferimenti senza parentesi: `\citealt` è l'analogo di `\citet` senza parentesi e `\citealp` è l'analogo di `\citep` senza parentesi. Esempi di utilizzo dei precedenti comandi con stile autore-anno (opzione `authoryear`) sono riportati nella Tab. 1 mentre la Tab. 2 riporta l'utilizzo dei comandi con stile numerato (opzione `number`).

Il comando standard `\cite` può ancora essere utilizzato e viene interpretato da `natbib` come `\citet` quando la bibliografia è autore-anno (opzione `authoryear`) oppure come `\citep` quando la bibliografia è numerata (opzione `number`).

5.3 Opzioni del pacchetto

5.3.1 Scegliere il tipo di parentesi

Il tipo di parentesi che racchiudono i riferimenti bibliografici può essere scelto con le seguenti opzioni:

- `round`, di default, seleziona parentesi tonde come “si veda Mori (2006)” o “si veda (2)”;
- `square` seleziona parentesi quadre come “si veda Mori [2006]” o “si veda [2]”;
- `curly` seleziona parentesi graffe come “si veda Mori {2006}” o “si veda {2}”.

TABELLA 1: Elenco dei comandi per i riferimenti nel testo e del loro effetto con lo stile autore-anno (opzione `author-year`).

Riferimenti nel testo	
<code>\citet{mori06}</code>	⇒ Mori et al. (2006)
<code>\cite{mori06}</code>	⇒ Mori et al. (2006)
<code>\citet[cap.~2]{mori06}</code>	⇒ Mori et al. (2006, cap. 2)
Riferimenti con parentesi	
<code>\citep{mori06}</code>	⇒ (Mori et al., 2006)
<code>\citep[cap.~2]{mori06}</code>	⇒ (Mori et al., 2006, cap. 2)
<code>\citep[vedi][]{mori06}</code>	⇒ (vedi Mori et al., 2006)
<code>\citep[vedi][cap.~2]{mori06}</code>	⇒ (vedi Mori et al., 2006, cap. 2)
Riferimenti con lista degli autori completa	
<code>\citet*{mori06}</code>	⇒ Mori, Lee, and Krishnan (2006)
<code>\citep*{mori06}</code>	⇒ (Mori, Lee, and Krishnan, 2006)
Riferimenti multipli	
<code>\citet{mori06,rossi07}</code>	⇒ Mori et al. (2006); Rossi et al. (2007)
<code>\citep{mori06,rossi07}</code>	⇒ (Mori et al., 2006; Rossi et al. 2007)
<code>\citep{mori06,mori07}</code>	⇒ (Mori et al., 2006, 1991)
<code>\citep{mori06a,mori06b}</code>	⇒ (Mori et al., 2006a,b)
Riferimenti senza parentesi	
<code>\citealt{mori06}</code>	⇒ Mori et al. 2006
<code>\citealt*{mori06}</code>	⇒ Mori, Lee e Krishnan 2006
<code>\citealp{mori06}</code>	⇒ Mori et al., 2006
<code>\citealp*{mori06}</code>	⇒ Mori, Lee e Krishnan, 2006
<code>\citealp{mori06,rossi07}</code>	⇒ Mori et al., 2006; Rossi et al., 2007
<code>\citealp[pag.~32]{mori06}</code>	⇒ Mori et al., 2006, pag. 32

TABELLA 2: Elenco dei comandi per i riferimenti nel testo e del loro effetto con lo stile numerato (opzione `numbered`).

Riferimenti nel testo	
<code>\citet{mori06}</code>	⇒ Mori et al. [11]
<code>\citet[cap.~2]{mori06}</code>	⇒ Mori et al. [11, cap. 2]
Riferimenti con parentesi	
<code>\citep{mori06}</code>	⇒ [11]
<code>\cite{mori06}</code>	⇒ [11]
<code>\citep[cap.~2]{mori06}</code>	⇒ [11, cap. 2]
<code>\citep[vedi][]{mori06}</code>	⇒ [vedi 11]
<code>\citep[vedi][cap.~2]{mori06}</code>	⇒ [vedi 11, cap. 2]
Riferimenti multipli	
<code>\citep{mori06a,mori06b}</code>	⇒ [11, 18]

Per modificare il tipo di parentesi, è anche possibile utilizzare il comando `\bibpunct` per il quale si rimanda a DALY (2007).

5.3.2 Scegliere il tipo di punteggiatura

Il tipo di punteggiatura per separare voci di riferimenti multipli può essere scelto con le seguenti opzioni:

- **colon**, di default, seleziona il punto e virgola come “(Mori et al., 2006; Rossi et al., 2007)” o “Mori et al. (2006); Rossi et al. (2007)”;
- **comma** seleziona la virgola come “(Mori et al., 2006, Rossi et al., 2007)” o “Mori et al. (2006), Rossi et al. (2007)”;

Anche la punteggiatura può essere modificata con il comando `\bibpunct` per il quale si rimanda a DALY (2007).

5.3.3 Scegliere lo stile bibliografico

Il pacchetto `natbib` permette di impostare lo stile bibliografico semplicemente specificando la rispettiva opzione al momento del richiamo del pacchetto. **authoryear**, di default, carica lo stile autore-anno, **numbers** lo stile numerato e **super** lo stile in nota a piè di pagina. Lo stile `.bst` usato deve supportare per lo meno l’opzione scelta.

5.3.4 Ordinare e comprimere i riferimenti multipli

Riferimenti multipli come `\cite{a,b,c,d}` di default producono qualcosa di orribile come “[2,6,4,3]”. Chiaramente ordinando a mano i riferimenti (ad esempio `\cite{b,c,d,a}`) si può ottenere “[2,3,4,6]” ma questo non è ancora accettabile. Il pacchetto `natbib`, quando usato con l’opzione **numbers**, offre l’ulteriore opzione **sort&compress** che au-

automaticamente ordina e comprime i riferimenti multipli. In questo caso ad esempio `\cite{a,b,c,d}` produrrebbe “[2–4,6]”.

5.3.5 Riferimenti dalla bibliografia al testo

In documenti lunghi può essere utile avere nella bibliografia dei riferimenti alle pagine in cui compaiono i riferimenti bibliografici di ogni voce. Sia il pacchetto `backref` che `citeref` permettono di farlo ma si consiglia l’uso del primo perché più moderno e integrato con `hyperref` (produce quindi anche riferimenti ipertestuali se usato insieme a `hypernat`). Nessuno dei due pacchetti supporta la compressione della lista delle pagine (“5, 6, 7” non viene trasformato in “5–7”) ma non ripetono una pagina se uno stesso riferimento vi compare più volte.

5.3.6 Ridurre lo spazio tra le voci della bibliografia

La bibliografia viene composta come una lista (in modo simile dunque ad `itemize`, `enumerate` e `description`) e quindi lo spazio tra le voci può essere modificato intervenendo su `\itemsep` nel seguente modo (UK TUG, 2008):

```
\let\oldbibliography\thebibliography
\renewcommand{\thebibliography}[1]{%
  \oldbibliography{#1}%
  \setlength{\itemsep}{0pt}%
}
```

Il pacchetto `natbib` offre una soluzione ancora più semplice in quanto permette di intervenire direttamente su `\bibsep` come nel seguente esempio

```
\setlength{\bibsep}{0pt}
```

5.3.7 Stile dei numeri nella bibliografia

Di default L \TeX formatta i numeri delle voci della bibliografia nel seguente modo:

- ```
[1] GARCIA, F. (2007). L \TeX and the different
 bibliography styles. The Prac \TeX Journal, (2).
[2] GOOSSENS, M., MITTELBAACH, F. e SAMARIN, A. (1995).
 The L \TeX Companion. Addison-Wesley.
```

Per modificarlo è possibile mettere nel preambolo un comando del tipo (UK TUG, 2008):

```
\makeatletter
\renewcommand*{\@biblabel}[1]{\hfill#1.}
\makeatother
```

oppure, se si usa `natbib`,

```
\renewcommand{\bibnumfmt}[1]{#1.}
```

Entrambi producono

- ```
1. GARCIA, F. (2007). L $\text{\TeX}$  and the different
    bibliography styles. The Prac $\text{\TeX}$  Journal, (2).
2. GOOSSENS, M., MITTELBAACH, F. e SAMARIN, A. (1995).
    The L $\text{\TeX}$  Companion. Addison-Wesley.
```

6 bibl \TeX

Il pacchetto `bibl \TeX` si propone di offrire all’utente una soluzione generale per la personalizzazione degli stili bibliografici e di citazione. Oltre ad includere le funzionalità di molti pacchetti esaminati in precedenza in questo articolo, consente di effettuare modifiche, anche profonde, ad un dato stile bibliografico, tramite comandi L \TeX . Il pacchetto, scritto da Philipp Lehman, è ancora in fase di sviluppo e, pur non essendo incluso nelle distribuzioni ufficiali, è reperibile su CTAN.⁵⁵

Il funzionamento del pacchetto richiede $\epsilon\text{-}\text{\TeX}$, il pacchetto `etoolbox`, anch’esso in fase di sviluppo, e i pacchetti standard `keyval`, `ifthen` e `calc`. L’uso di `babel` e di `csquote`, anche se non indispensabile è raccomandato per il buon funzionamento di `bibl \TeX` . Ovviamente non può essere usato insieme ai pacchetti che si propone di sostituire.

Una discussione approfondita sull’uso di `bibl \TeX` richiederebbe un articolo a sé. Nel seguito verranno esaminate solo le caratteristiche principali, rimanando il lettore alla documentazione del pacchetto (LEHMANN, 2007).

6.1 Stili bibliografici e di citazione

Il principale limite di BIB \TeX è il fatto che, per poter avere un controllo totale dello stile bibliografico, è necessario imparare un linguaggio *ad hoc*, notevolmente diverso da quello cui è abituato l’utente \TeX /L \TeX . `custom-bib`, come si è visto nel paragrafo 4.2.2, aiuta molto ma può non essere sufficiente ad ottenere il risultato voluto. Inoltre, prevede la compilazione di un file ogni volta che si apportano modifiche e quindi non è molto comodo per visualizzare immediatamente le modifiche effettuate.

Con `bibl \TeX` la formattazione dei riferimenti bibliografici e delle citazioni è interamente gestibile tramite comandi L \TeX . Gli stili non sono più contenuti in file `.bst`, nella tradizionale sintassi BIB \TeX , ma in file `.bbx` (stili bibliografici) o `.cbx` (schemi di citazione). Il file `.bb1` che si ottiene dalla compilazione con `bibtex` non contiene, in questo caso, l’ambiente `thebibliography`, ma una serie di macro che contengono i dati delle singole voci della bibliografia. La formattazione vera e propria viene eseguita al momento di comporre i riferimenti bibliografici, seguendo le istruzioni contenute nello stile bibliografico prescelto.

Lo stile viene specificato tramite un’opzione del pacchetto:

```
\usepackage[style=numeric]{bibl $\text{\TeX}$ }
```

oppure:

```
\usepackage[bibstyle=authortitle,%
  citestyle=verbose-trad1]{bibl $\text{\TeX}$ }
```

55. [http://www.ctan.org/tex-archive/macros/latex/exptl/bibl \$\text{\TeX}\$ /](http://www.ctan.org/tex-archive/macros/latex/exptl/bibl\TeX/)

Nel primo caso il valore `numeric` è assegnato sia a `bibstyle` che a `citestyle`.

Il pacchetto fornisce alcuni stili predefiniti; quelli bibliografici coprono i quattro schemi tradizionali: numerico (`numeric`), alfabetico (`alphabetic`), autore-anno (`authoryear`), autore-titolo (`authortitle`). A ciascuno di questi stili è possibile associare uno schema di citazione appropriato. A seconda dello schema scelto è possibile avere riferimenti numerici, con etichetta alfanumerica, autore-anno, autore-titolo, in nota, ecc. Lo schema `verbose`, che usa la citazione completa alla prima occorrenza e una forma abbreviata le volte successive, prevede alcune varianti in cui vengono usate, a seconda del contesto, le formule convenzionali *idem*, *ibidem*, *op. cit* e *loc. cit*.

È possibile modificare gli stili predefiniti, all'interno del proprio documento, o crearne di nuovi a partire da quelli esistenti. Ad esempio, se si vuole che i titoli degli articoli appaiano in corsivo e il nome della rivista sia racchiuso tra virgolette basse, è sufficiente scrivere i seguenti comandi nel preambolo:

```
\DeclareFieldFormat{article}{title}%
{\mkbibemph{#1\isdot}}
\DeclareFieldFormat{journaltitle}%
{\mkbibquote{#1}}
```

oppure, nel caso si voglia modificare in questo senso lo schema autore-anno, in modo da poterlo riutilizzare in più di un documento, si dovrà scrivere nel file `myauthoryear.bbx`:

```
\RequireBibliographyStyle{authoryear}
\DeclareFieldFormat{article}{title}%
{\mkbibemph{#1\isdot}}
\DeclareFieldFormat{journaltitle}%
{\mkbibquote{#1}}
\endinput
```

e assegnare `myauthoryear` come valore all'opzione `bibstyle`.

6.2 Comandi per le citazioni

`biblatex` fornisce comandi per le citazioni che possono essere usati in diversi contesti. Oltre ai classici `\cite` e `\nocite` sono disponibili: `\parencite`, che racchiude la citazione tra parentesi, `\footcite`, che inserisce la citazione in nota, `\textcite`, da usare quando la citazione è parte integrante del discorso, `\supercite` (solo per schemi numerici) per citazioni in esponente, e `\fullcite` che inserisce l'intera voce bibliografica. `\autocite`, a seconda del contesto, si comporta come uno dei comandi citati sopra. Esempi di utilizzo di questi comandi sono riportati nelle Tab. 3 e 4.

Infine sono disponibili comandi per citare singole parti di una voce bibliografica: `\citeauthor`, `\citetitle`, `\citeyear`, e `\citeurl`.

6.3 Comandi per la bibliografia

`biblatex` si discosta dall'uso tradizionale di L^AT_EX per quanto riguarda l'inserimento dei riferimenti bibliografici. Come visto in precedenza, lo stile biblio-

grafico viene specificato come opzione del pacchetto, senza usare il comando `\bibliographystyle`. Il comando `\bibliography` ha la sola funzione di specificare i database da usare per recuperare le voci bibliografiche, ma non inserisce nessun elenco. Tale elenco deve essere inserito con il comando apposito `\printbibliography`, che accetta un argomento opzionale in cui inserire una serie di istruzioni di controllo. Le più interessanti sono quelle che eseguono dei filtri sulle voci da stampare. Tali filtri possono essere determinati tramite un particolare campo della voce bibliografica (`type` o `keyword`), oppure tramite la definizione, nel preambolo, di categorie (`\DeclareBibliographyCategory`) e l'assegnazione delle singole voci a tali categorie (`\addtocategory`), o ancora a seconda della parte del documento in cui la citazione si trova, se tra le opzioni del pacchetto è stato specificato `refsection` o `refsegment`.⁵⁶

In questo modo è facile avere una bibliografia divisa per capitoli, o per argomenti, usando più volte il comando `\printbibliography` con diversi filtri attivati. Addirittura, se le varie bibliografie, pur divise, devono essere inserite tutte alla fine del documento, basta usare i comandi `bibbysection`, `bibbysegment` o `bibbycategory`.

6.4 Comandi per l'indice analitico

Un'interessante funzionalità di `biblatex` è l'indicizzazione automatica delle voci bibliografiche. L'indice analitico finale, e soprattutto l'indice dei nomi, è molto utile al lettore di qualsiasi libro, comprese le tesi, ed è di norma presente in tutte le pubblicazioni scientifiche. L'opzione `indexing` di `biblatex` permette di inserire automaticamente nell'indice dei nomi (o in un indice a scelta) tutti gli autori citati in bibliografia e nel testo. `biblatex` si appoggia, per creare l'indice analitico, al pacchetto `makeidx` (per le funzionalità di base) o al pacchetto `index` (per le funzionalità avanzate, come gli indici multipli). Di seguito, descriviamo un esempio di base di indicizzazione automatica dei nomi, rimandando a LEHMANN (2007, in particolare il paragrafo 3.1.2 e i modelli d'esempio che accompagnano la documentazione) per una spiegazione più dettagliata di questa funzionalità.

Supponendo di avere un database bibliografico che si chiama `database.bib`, il seguente codice per il documento principale permette di provare le funzionalità di `biblatex`:

```
\documentclass{...}

\usepackage{indexing}{biblatex}
\bibliography{database}

\usepackage{makeidx}
\makeindex

\begin{document}
```

56. Le sezioni possono anche essere definite manualmente, racchiudendo una porzione del documento tra `\begin{refsection}` e `\end{refsection}`.

TABELLA 3: Elenco dei comandi di biblatex per i riferimenti nel testo e del loro effetto con lo stile autore-anno compatto (opzione `authoryear-comp`).

Riferimenti nel testo	
<code>\textcite{mori06}</code>	⇒ MORI et al. (2006)
<code>\textcite[cap.~4]{mori06}</code>	⇒ MORI et al. (2006, cap. 4)
Riferimenti con parentesi	
<code>\autocite{mori06}</code>	⇒ (MORI et al. 2006)
<code>\parencite{mori06}</code>	⇒ (MORI et al. 2006)
<code>\parencite[cap.~4]{mori06}</code>	⇒ (MORI et al. 2006, cap. 4)
<code>\parencite[vedi] []{mori06}</code>	⇒ (vedi MORI et al. 2006)
<code>\parencite[vedi] [cap.~4]{mori06}</code>	⇒ (vedi MORI et al. 2006, cap. 4)
Riferimenti multipli	
<code>\cite{mori06,rossi07}</code>	⇒ MORI et al. 2008; ROSSI et al. 2007
<code>\textcite{mori06,rossi07}</code>	⇒ MORI et al. (2008); ROSSI et al. (2007)
<code>\parencite{mori06,rossi07}</code>	⇒ (MORI et al. 2008; ROSSI et al. 2007)
<code>\cite{mori06,mori08}</code>	⇒ MORI et al. 2006, 2008
<code>\cite{mori06a,mori06b}</code>	⇒ MORI et al. 2006a,b
Riferimenti senza parentesi	
<code>\cite{mori06}</code>	⇒ MORI et al. 2006
<code>\cite[cap.~4]{mori06}</code>	⇒ MORI et al. 2006, cap. 4

TABELLA 4: Elenco dei comandi di biblatex per i riferimenti nel testo e del loro effetto con lo stile numerato (opzione `numeric-comp`).

Riferimenti nel testo	
<code>\textcite{mori06}</code>	⇒ MORI et al. [6]
<code>\textcite[cap.~4]{mori06}</code>	⇒ MORI et al. [6, cap. 4]
Riferimenti con parentesi	
<code>\cite{mori06}</code>	⇒ [6]
<code>\parencite{mori06}</code>	⇒ [6]
<code>\autocite{mori06}</code>	⇒ [6]
<code>\cite[cap.~4]{mori06}</code>	⇒ [6, cap. 4]
<code>\cite[vedi] []{mori06}</code>	⇒ vedi [6]
<code>\cite[vedi] [cap.~4]{mori06}</code>	⇒ [vedi [6, cap. 4]
Riferimenti multipli	
<code>\cite{mori06,mori08}</code>	⇒ [3, 6]

```
Come nota \textcite{Kopka1995}...
\clearpage

E come afferma \citeauthor{Goossens1995} nel suo...
\clearpage

\nocite{*}
\printbibliography
\printindex
\end{document}
```

Assumendo che il nostro file principale si chiami `documento.tex`, compilandolo con:

```
latex    documento
bibtex   documento
latex    documento
makeindex documento
latex    documento
```

si ottiene un documento con la normale bibliografia seguita da un indice analitico contenente tutti i nomi citati nel testo (con uno dei comandi di citazione discussi nel paragrafo 6.2) e in bibliografia (se contenuti nel campo `author` del file `.bib`). Specificando `indexing=cite` o `indexing=bib` è possibile

inserire nell'indice, rispettivamente, *solo* i nomi citati nel testo o quelli citati in bibliografia.

Benché non sia forse troppo diffuso elencare nell'indice i nomi che appaiono nelle pagine dei riferimenti bibliografici (quelli, per intendersi, stampati da `\printbibliography`), farlo è particolarmente utile per controllare, almeno in prima approssimazione, la coerenza e la completezza dell'indice: infatti, ogni nome dovrà apparire almeno due volte nell'indice, la prima nella pagina in cui è citato e l'ultima nella pagina della bibliografia. Da questo punto di vista, il problema che sorge è che `biblatex`, sia con l'opzione globale `indexing`, sia con quella specifica `indexing=bib`, inserisce automaticamente nell'indice anche tutti i *titoli* delle opere citate, cosa questa del tutto inusuale. Una possibile soluzione è ridefinire nel preambolo il campo utilizzato da `biblatex` per gestire l'indicizzazione dei titoli (LEHMANN, 2007, paragrafo 4.4.2), come segue:

```
\DeclareIndexFieldFormat{indextitle}{\{ }\{ }
```


In questo modo, l'indicizzazione dei titoli viene soppressa, e nell'indice appariranno solo i nomi degli autori, sia alla pagina di citazione sia alla corrispondente pagina dei riferimenti bibliografici.

6.5 Supporto multilingue

`babel` include anche le funzionalità di `babelbib`. Se è stata attivata l'opzione `babel`, con il valore `hyphen` o `other`, il pacchetto controlla, per ogni voce della bibliografia, se è stata specificata una lingua tramite il campo `hyphenation` e, in caso affermativo, usa per quella voce le regole di sillabazione, ed eventualmente la traduzione di termini quali "curatore" o "volume", ecc., associate a tale lingua.

In generale, le traduzioni e le localizzazioni di formati di data o di altri elementi, usate da `babel`, sono contenute in file con estensione `.lbn` e il cui nome corrisponde ad un identificatore riconosciuto da `babel`.

Riferimenti bibliografici

- AUTORI VARI (2007). *Introduzione all'arte della composizione tipografica con L^AT_EX*. Curato da C. Beccari. URL <http://www.guit.sssup.it/downloads/GuidaGuIT.pdf>.
- BAUDOIN, M. (1998). *Impara L^AT_EX (...e mettilo da parte)*. URL http://www.mat.uniroma1.it/centro-calcolo/manuali/impara_latex.pdf.
- DALY, P. W. (2003). «Customizing bibliographic style files». URL <ftp://tug.ctan.org/pub/tex-archive/macros/latex/contrib/custom-bib/makebst.pdf>.
- (2006). «Natural sciences citations and references (author-year and numerical schemes)». URL [aggiungiurl](http://aggiungiurl.org).
- (2007). «A master bibliographic style file for numerical, author-year, multilingual applications». URL <ftp://tug.ctan.org/pub/tex-archive/macros/latex/contrib/custom-bib/makebst.pdf>.
- FENN, J. (2006). «Managing citations and your bibliography with BIB_TE_X». *The PracT_EX Journal*, (4). URL <http://tug.org/pracjourn/2006-4/fenn>.
- GARCIA, F. (2007). «L^AT_EX and the different bibliography styles». *The PracT_EX Journal*, (2). URL <http://tug.org/pracjourn/2007-2/garcia>.
- GOOSSENS, M., MITTELBAACH, F. e SAMARIN, A. (1995). *The L^AT_EX Companion*. Addison-Wesley.
- ISO 690 (1987). *Documentation—Bibliographic references—Content, form and structure*. International Organization for Standardization, Geneva, Switzerland, 2^a edizione.
- KOPKA, H. e DALY, P. W. (1995). *A Guide to L^AT_EX – Document Preparation for Beginners and Advanced Users*. Addison-Wesley.
- LEHMANN, P. (2007). «The `babel` package». URL <http://www.ctan.org/tex-archive/macros/latex/exptl/babel/doc/babel.pdf>.
- OETIKER, T., PARTL, H., HYNÄ, I. e SCHLEGEL, E. (2000). *Una (mica tanto) breve introduzione a L^AT_EX 2_ε*. URL <http://www.ctan.org/tex-archive/info/lshort/italian/itlshort.pdf>.
- PATASHNIK, O. (1998). «BIB_TE_Xing». URL <ftp://ftp.ctan.org/tex-archive/biblio/bibtex/contrib/doc/btxdoc.pdf>.
- THE TUTORIAL TEAM (2000). *On-line Tutorial on L^AT_EX*. Indian T_EX Users Group. URL <http://www.tug.org.in/tutorials.html>.
- UK TUG (2008). *The UK TeX FAQ*. URL <http://www.tex.ac.uk/tex-archive/help/uk-tex-faq/newfaq.pdf>.
- UNI 10168 (1993). *Documentazione. Riferimenti bibliografici. Contenuto, forma e struttura*. Ente nazionale italiano di unificazione, Milano, Italy.
- WIKIPEDIA (2007). «Wikibook on L^AT_EX: Bibliography management». URL http://en.wikibooks.org/wiki/LaTeX/Bibliography_Management.

▷ Lapo F. Mori
Dipartimento di Ingegneria Meccanica,
Nucleare e della Produzione
Università di Pisa
lapo dot mori at ing dot unipi
dot it

Introduzione allo stile ClassicThesis

Lorenzo Pantieri*

Sommario

Lo scopo di questo lavoro è fornire agli utenti di \LaTeX di lingua italiana alcuni strumenti per scrivere un documento utilizzando lo stile ClassicThesis, di André Miede, ispirato all'opera di Robert Bringhurst *Gli Elementi dello Stile Tipografico* (1992).

Tale obiettivo è perseguito in primo luogo presentando una mia rielaborazione della documentazione dello stile (MIEDE, 2007), opportunamente integrata con svariati temi (tratti dai testi riportati nella bibliografia), che ritengo utili complementi. In secondo luogo vengono analizzati alcuni problemi tipici incontrati durante la stesura di una pubblicazione accademica o professionale, specialmente in lingua italiana, indicando le soluzioni che ritengo migliori.

Abstract

The purpose of this work is to provide to the Italian \LaTeX users some tools to write a document using the ClassicThesis style, by André Miede, inspired to the Robert Bringhurst's masterpiece *The Elements of the Typographical Style* (1992).

This aim is pursued introducing my personal reworking of the style documentation (MIEDE, 2007) and analyzing the typical problems faced during the writing of an academic or professional publication, especially in Italian language, showing the solutions I think better.

Chiarimenti

La scelta delle soluzioni adottate in questo lavoro deriva sia dalla mia esperienza sia dalle numerose discussioni riguardanti ClassicThesis presenti sul forum del Gruppo Utilizzatori Italiani di \TeX e \LaTeX (<http://www.guit.sssup.it/phpbb/index.php>), che resta sempre un eccellente

*Desidero ringraziare i membri dello Staff del Gruppo Utilizzatori Italiani di \TeX e \LaTeX , in particolare il Prof. Claudio Beccari, Fabiano Busdraghi, Gustavo Cevolani, Massimiliano Dominici, Daniele Ferone, il Prof. Enrico Gregorio, Lapo F. Mori, Ottavio Rizzo, Andrea Tonelli, Emiliano Vavassori ed Emanuele Vicentini, per l'impagabile aiuto fornito nella redazione di questo lavoro, le spiegazioni dettagliate, la pazienza e la precisione nei suggerimenti, le soluzioni fornite, la competenza e la disponibilità: grazie mille, ragazzi! Ringrazio inoltre André Miede per aver elaborato lo splendido stile ClassicThesis e per averne discusso con me a lungo, prodigo di preziose osservazioni e di validi consigli.

E un grazie davvero speciale a Monia, che mi è stata vicino in questo periodo intenso, piacevole e costruttivo.

riferimento per tutti i temi trattati nel presente documento.

Il testo presume che il lettore conosca già i rudimenti di \LaTeX , ovvero che abbia letto una delle numerose guide di base disponibili gratuitamente in Rete (BECCARI, 2008; PANTIERI, 2008), oppure un manuale cartaceo.

La prassi seguita è quella di non approfondire i vari temi nei dettagli, ma di indirizzare il lettore alla letteratura specifica o ai manuali dei pacchetti suggeriti, quando necessario. Tutte le volte che si cita un pacchetto, non si fornisce una descrizione completa del suo funzionamento, per cui si rimanda alla relativa documentazione, ma si analizzano le opzioni più importanti e se ne suggerisce l'utilizzo.

L'esposizione del lavoro è articolata come segue:

Nella prima parte viene fornita una visione d'insieme dello stile ClassicThesis e ne vengono descritte le caratteristiche fondamentali e le peculiarità. Dopo aver letto questa parte, si dovrebbe avere una conoscenza di base del funzionamento di ClassicThesis.

Nella seconda parte vengono esposti alcuni suggerimenti su come personalizzare i propri documenti scritti con ClassicThesis, facendo in modo che vengano prodotti risultati diversi da quelli predefiniti.

1 Le basi di ClassicThesis

1.1 Storia e filosofia

La *suite* ClassicThesis, sviluppata per \LaTeX da André Miede, è composta da due pacchetti, `classicthesis` e `classicthesis-ldpkg`, e da un modello di tesi pronto per l'uso.¹

La *suite* ha due obiettivi:

1. fornire un modello, pronto per l'uso, di tesi di laurea o di dottorato (in realtà, il modello può essere usato anche per comporre libri e relazioni);
2. fornire uno stile “classico” e di alta qualità, ispirato al capolavoro di Robert Bringhurst *Gli Elementi dello Stile Tipografico* (1992).

La *suite* è configurata per essere eseguita su un'installazione *completa* di \MiKTeX o \TeX Live e

1. Il nome ClassicThesis indica dunque lo “stile” (che comprende i due pacchetti e il modello), mentre il pacchetto che contiene la maggior parte delle impostazioni è denotato con `classicthesis`: ciò non dovrebbe essere fonte di eccessiva confusione.

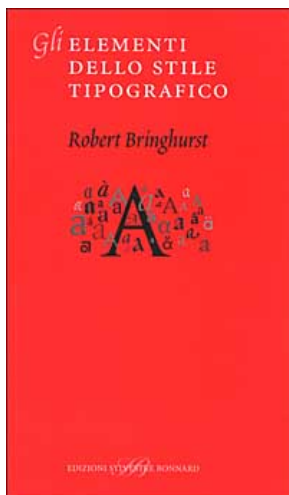


FIGURA 1: Copertina dell'edizione italiana degli *Elementi dello Stile Tipografico* (BRINGHURST, 1992).

usa font liberamente disponibili. (Chi volesse usare i font Minion Pro può comunque farlo facilmente).²

Lo stile *ClassicThesis* dovrebbe consentire a chiunque abbia una minima conoscenza di \LaTeX di produrre documenti eleganti con relativa facilità.

La prima versione della *suite* (che è distribuita con licenza GNU) è stata pubblicata all'inizio del 2006. Nel giugno del 2007 è stata pubblicata la versione 2.0, che contiene diversi piccoli miglioramenti. Attualmente (ottobre 2008) è disponibile la versione 2.3, che è estremamente stabile e collaudata: ad oggi sono stati composti con *ClassicThesis* numerosi documenti, e lo stile è ora diffuso in tutto il mondo.

Chi fosse interessato ad approfondire i dettagli tipografici su cui *ClassicThesis* si basa, può farlo leggendo il capolavoro di Bringhurst (1992).

Importante

Alcuni aspetti di questo stile potrebbero sembrare insoliti, ad una prima occhiata: è un'impressione piuttosto comune all'inizio. Tuttavia, si tratta di caratteristiche appositamente studiate per essere così come sono, specialmente le seguenti:

- Lo stile non usa il neretto: il corsivo o il maiuscoletto lo sostituiscono egregiamente.
- Le dimensioni dei margini sono definite appositamente così come sono. In questo modo vengono prodotti documenti eleganti e facilmente leggibili, e al contempo si ha una quantità sufficiente di informazioni sulla pagina. E, no: le righe *non* sono troppo corte.
- Le tabelle non usano linee verticali. Si consiglia di leggere la documentazione del pacchetto

2. Il file `LISTOFFILES` contiene l'elenco dei pacchetti richiesti. In effetti, *ClassicThesis* funziona con la maggior parte delle distribuzioni e dei sistemi operativi per cui \LaTeX è disponibile.

to `booktabs` per un'esauriente discussione a questo proposito.

- E, infine, viene fornito al lettore un accesso più semplice ai numeri di pagina, negli indici: i numeri di pagina si trovano subito di fianco alle relative voci. Sì, i numeri *non* sono ordinatamente allineati a destra e *non* sono connessi alle rispettive voci con punti che guidano l'occhio lungo una distanza non necessaria. Per chi non fosse ancora convinto: il lettore è interessato a conoscere il numero di pagina o vuole fare la somma dei numeri?

Se si modificano queste impostazioni si corre il rischio di distruggere la bellezza e l'armonia dello stile, per cui è decisamente consigliabile lasciarle invariate (a meno che non si sappia davvero che cosa si sta facendo).

1.2 Uso dello stile

1.2.1 Installazione

L'installazione di *ClassicThesis* non presenta particolari difficoltà. Si scarica il file `classicthesis.zip` da <http://www.ctan.org/> e, dopo averlo decompresso, si installano i pacchetti `classicthesis` e `classicthesis-ldpkg` nel solito modo (la procedura per l'installazione di un pacchetto dipende dalla propria distribuzione di \LaTeX : i dettagli sono riportati nel manuale della distribuzione che si usa).³

1.2.2 Impiego

Lo stile *ClassicThesis* si carica semplicemente con

```
\documentclass[...]{scrreprt}
%\documentclass[...]{scrbook}
%\documentclass[...]{scrartcl}
\usepackage{...}
\usepackage[...]{classicthesis-ldpkg}
\usepackage[...]{classicthesis}
\usepackage{...}

\begin{document}
...
\end{document}
```

Le classi `scrreprt`, `scrbook` e `scrartcl` sono le controparti delle classi standard di \LaTeX `report`, `book` e `article`, rispettivamente (vedi il paragrafo 1.8 nella pagina 55).

3. Per esempio, se si usa la distribuzione \TeX Live è sufficiente posizionare il relativo file `.sty` nel cosiddetto "albero personale", ovvero la cartella in cui il singolo utente può inserire i pacchetti che utilizza e che non sono inclusi nella distribuzione di \LaTeX (su Mac tale cartella è `$HOME/Library/texmf/tex/latex/`; su Linux essa dipende dalla distribuzione utilizzata). Se invece si usa la distribuzione \MiKTeX è sufficiente avviare il programma \MiKTeX Package Manager e selezionare il pacchetto da installare; una volta concluso il processo di installazione è necessario aggiornare la base di dati dei pacchetti (PANTIERI, 2008, p. 25).

Se si sta scrivendo una tesi, è consigliabile modificare uno alla volta i file del modello: solo dopo essersi assicurati che tutto funzioni, è possibile eventualmente personalizzare il proprio documento.

1.3 Organizzazione del lavoro

Un fattore molto importante per il successo nella scrittura di una tesi (ma anche di un libro o di una relazione) è l'organizzazione del proprio materiale. Si suggerisce di adottare una struttura come la seguente:

- La cartella **Capitoli/** contiene il materiale principale, suddiviso in capitoli, come **Capitolo01.tex** ecc.
- La cartella **MaterialeInizialeFinale/** contiene il materiale iniziale e finale, come i ringraziamenti, la dedica, la bibliografia, ecc.
- La cartella **Immagini/** contiene tutte le figure usate nella tesi. Se si hanno molte figure, potrebbe essere conveniente organizzare la cartella in sottocartelle, contenenti ciascuna le figure di un capitolo.
- **Bibliografia.bib**: la base di dati di **BIB_{TEX}** che contiene tutti i riferimenti bibliografici del documento.
- **ClassicThesis.tex** (o un file **.tex** dal nome appropriato): il file principale della tesi, che richiama tutti gli altri.

Questo piccolo accorgimento permette di semplificare notevolmente il proprio lavoro.

1.4 Opzioni del pacchetto *classicthesis*

Il pacchetto *classicthesis* ha alcune opzioni che consentono di modificare (moderatamente) l'aspetto dei propri documenti.

drafting stampa la data e l'ora in fondo ad ogni pagina del documento, in modo da avere sempre sott'occhio la versione a cui si sta lavorando. Se si sta scrivendo una tesi di laurea o di dottorato, questa opzione è utile per evitare il rischio di consegnare al relatore una versione non aggiornata del proprio lavoro.

eulerchapternumbers imposta, per i numeri dei capitoli, i font *AMS Euler*, di Hermann Zapf e Donald Knuth (il font predefinito è il Palatino).

linedheaders cambia l'aspetto della pagina del titolo dei capitoli, aggiungendo una linea orizzontale sopra e sotto il titolo. Il numero del capitolo viene spostato in alto, sopra il titolo.

listsseparated separa, nell'indice, le voci di figure e tabelle relative a capitoli diversi, aggiungendo uno spazio extra.

totaligned allinea a sinistra tutte le voci dell'indice.

subfig attiva la compatibilità con il pacchetto **subfig**.

nochapters va attivata se si desidera usare il pacchetto *classicthesis* con una classe che non è suddivisa in capitoli, come **scrartcl**. Disattiva automaticamente le opzioni **eulerchapternumbers**, **linedheaders**, **listsseparated** e **parts**.

beramono imposta i font Bera Mono come font a larghezza fissa. (Non c'è alcun font a larghezza fissa predefinito.)

eulermath imposta gli splendidi font *AMS Euler* come font per le formule matematiche (i font predefiniti per la matematica sono quelli della famiglia Palatino.)

parts va selezionata se si usa una suddivisione in parti del documento. Fornisce il comando **\myPart**, che si occupa dello stile delle relative voci nell'indice e nel corpo del testo. (L'opzione non può essere usata insieme a **nochapters**.)

a5paper adatta le proporzioni di pagina al formato A5 (questa caratteristica è ancora sperimentale).

pdfspacing realizza la spaziatura tra i caratteri del maiuscolo mediante il pacchetto **microtype** (è possibile usare l'opzione **DVloutput** di **microtype** per generare documenti in formato DVI). Questa opzione corregge, fra l'altro, alcuni problemi nella composizione dei titoli dei paragrafi e dei capitoli.

minionpro imposta i font Minion Pro, di Robert Slimbach, come font principali del documento. Le proporzioni di pagina sono modificate di conseguenza.

minionprospacing realizza la spaziatura tra i caratteri del maiuscolo mediante il pacchetto **MinionPro**. Questa opzione abilita automaticamente l'opzione **minionpro** e annulla l'opzione **pdfspacing**.

Il modo migliore per capire come funzionano queste opzioni è provare le differenti possibilità e scegliere quelle che risultano più gradite.

1.5 Il pacchetto *classicthesis-ldpkg*

Il pacchetto *classicthesis-ldpkg* ne carica altri utili. Il pacchetto ha solo un'opzione disponibile, **nochapters**, il cui valore predefinito è **false**. L'opzione va attivata se si desidera usare il pacchetto con una classe che non è suddivisa in capitoli, come ad esempio la classe **scrartcl**.

TABELLA 1: Comandi di sezionamento del documento.

<code>\myPart</code>	<code>\subsubsection</code>
<code>\myChapter</code>	<code>\paragraph</code>
<code>\section</code>	<code>\subparagraph</code>
<code>\subsection</code>	

1.6 Non solo tesi

È possibile avvalersi dello stile *ClassicThesis* per scrivere articoli, relazioni, libri e curriculum vitae, anche senza avvalersi del relativo modello di tesi. Nella documentazione dello stile vengono proposti alcuni esempi pronti per l'uso.

1.7 Alcuni comandi e ambienti notevoli

I comandi e gli ambienti sono strumenti estremamente potenti di cui si fa un uso intensivo in \LaTeX . In questo paragrafo vengono presentati alcuni comandi ed ambienti specifici di *ClassicThesis*.

1.7.1 Comandi di sezionamento del documento

Strutturare un documento (suddividerlo in parti, capitoli, paragrafi, sottoparagrafi, ...) è molto facile da realizzare con \LaTeX . Se si usa *ClassicThesis*, è necessario usare i comandi della tabella 1. I comandi `\myPart` e `\myChapter` sono disponibili nelle classi `scrreprt` e `scrbook`, ma non nella classe `scrartcl`.

1.7.2 Elenchi numerati e descrizioni

Gli ambienti che realizzano elenchi sono usati molto spesso in \LaTeX . Permettono infatti:

- A. di “dare respiro” al testo;
- B. di migliorarne la leggibilità;
- C. di strutturare le proprie idee.

Un elenco come quello precedente, con le etichette in maiuscolo, si può ottenere con l'ambiente `aenumerate` di *ClassicThesis*, il cui uso è analogo a quello dell'ambiente standard `enumerate`.

1.7.3 Note a margine

Una nota a margine si ottiene molto semplicemente con il comando

```
\graffito{...}
```

Nei documenti fronte-retro, le note sono poste nel margine destro nelle pagine dispari e nel margine sinistro nelle pagine pari. Nei documenti solo-fronte, le note sono sempre poste nel margine destro.

1.7.4 Comandi di cambiamento di stile

Di regola, \LaTeX sceglie il carattere appropriato in base alla struttura logica del documento (capitoli, paragrafi, testatine, ...). In alcuni casi, si potrebbe però voler cambiare lo stile dei caratteri,

manualmente; per far ciò con *ClassicThesis*, si usano i comandi elencati nella tabella 2 nella pagina successiva.

1.8 Le classi KOMA-Script

Com'è noto, \LaTeX mette a disposizione quattro classi standard per la composizione di libri, relazioni, articoli e lettere: esse sono rispettivamente `book`, `report`, `article` e `letter`. Lo stile *ClassicThesis*, però, funziona non con le usuali classi standard di \LaTeX , bensì con le classi *KOMA-Script*, predisposte da Markhus Kohm (da cui il prefisso “Koma”).

Le classi *KOMA* sono progettate come controparti delle classi standard. Alle classi `book`, `report`, `article` e `letter` corrispondono rispettivamente le classi `scrbook`, `scrreprt`, `scrartcl` e `scrlettr` (vedi la tabella 3 nella pagina seguente). I nomi delle classi *KOMA-Script* sono composti dal prefisso “scr” e dal nome abbreviato della corrispondente classe standard (al fine di accorciare la lunghezza dei nomi ad otto lettere).

1.8.1 Opzioni delle classi KOMA-Script

Le classi *KOMA-Script* hanno svariate opzioni, che hanno effetto sull'intero documento. Di seguito vengono descritte le opzioni globali più comuni per le tre classi principali `scrartcl`, `scrreprt` e `scrbook` (dal momento che la classe `scrlettr` è fondamentalmente diversa dalle altre tre, per essa si rimanda alla documentazione di *KOMA-Script* (2007, p. 139)). Molte delle opzioni delle classi *KOMA-Script* sono analoghe a quelle delle classi standard; la loro descrizione è inclusa qui per comodità.

La tabella 4 nella pagina 57 riporta le opzioni globali più comuni delle tre classi principali *KOMA-Script*, mostrando per ciascuna classe se l'opzione è predefinita o meno (un'opzione indefinita in una classe è incompatibile con quella classe, e non può essere scelta).

10pt, 11pt, 12pt Hanno lo stesso effetto delle corrispondenti opzioni standard. Impostano la dimensione del font principale nel documento. Il valore predefinito è di 11 punti.

a4paper, a5paper, ... Definiscono le dimensioni del foglio. La dimensione predefinita è **a4paper**.

oneside, twoside Queste opzioni hanno lo stesso effetto che nelle classi standard. Specificano se deve essere generato un documento a doppia o singola facciata. Le classi `scrartcl` e `scrreprt` sono a singola facciata e la classe `scrbook` è a doppia facciata per impostazione predefinita.

openany, openright Queste opzioni hanno lo stesso effetto che nelle classi standard. L'opzione **openright**, predefinita nella classe `scrbook`, fa iniziare i capitoli sempre in una pagina destra, mentre l'opzione **openany**, predefinita nella classe `scrreprt`, li fa iniziare nella successiva

TABELLA 2: Le varianti di stile del font Palatino usate da ClassicThesis.

Comando	Stile
<code>\textit</code>	<i>Corsivo</i>
<code>\textsc</code>	MAIUSCOLETTTO
<code>\spacedlowsmallcaps</code>	MAIUSCOLETTTO BASSO SPAZIATO
<code>\spacedallcaps</code>	MAIUSCOLETTTO SPAZIATO

TABELLA 3: Corrispondenza tra le classi KOMA-Script e le classi standard.

Classe standard	Classe KOMA-Script
<code>article</code>	<code>scrartcl</code>
<code>report</code>	<code>scrreprt</code>
<code>book</code>	<code>scrreprt</code>
<code>letter</code>	<code>scrlettr</code>

pagina a disposizione. Queste opzioni non sono disponibili con la classe `scrartcl`, che non ha capitoli.

BCOR*(correzione)* Nelle classi KOMA-Script, questa opzione permette di specificare il valore della correzione per la rilegatura, lasciando uno spazio uguale a *(correzione)* per la rilegatura del documento. Per la correzione, è possibile usare una qualsiasi unità tipografica riconosciuta da L^AT_EX.

headinclude, headexclude, footinclude, footexclude Per calcolare le proporzioni di pagina è necessario comunicare al pacchetto `typearea` (che nelle classi KOMA-Script determina tali proporzioni) se le testatine e i piè di pagina vadano considerati come parte del corpo del testo o dei bordi. Le opzioni di classe `headinclude` e `footinclude` fanno sì che le testatine e i piè di pagina vengano considerati come parte del testo, le opzioni `headexclude` e `footexclude` fanno sì che essi vengano considerati come parte dei bordi.

*(valore)***headlines** Abbiamo visto come per calcolare le proporzioni di pagina sia necessario specificare se le testatine e i piè di pagina vadano considerati come parte del corpo del testo o dei bordi. Bisogna però anche specificare l'altezza delle testatine. L'opzione *(valore)***headlines** permette di farlo, dove *(valore)* indica il numero di righe di ciascuna testatina. Il valore predefinito è di 1.25 (è un valore di compromesso, sufficiente quando le testatine sono sottolineate, ma tale da non sovrastimare eccessivamente il valore del bordo superiore quando le testatine non sono sottolineate). Generalmente è possibile lasciare invariato il valore predefinito di *(valore)***headlines** e modificarlo solo in casi particolari.

cleardoubleempty, cleardoubleplain Se si desidera che le pagine vuote create dal comando `\cleardoublepage` non abbiano né testatina né piè di pagina, con le classi standard l'unica possibilità è ridefinire tale comando. Le classi KOMA-Script, invece, forniscono un'apposita opzione: se si usa l'opzione `cleardoubleempty`, alle pagine lasciate vuote viene applicato lo stile `empty` (vedi il paragrafo 1.8.2 a fronte). Se invece si usa l'opzione `cleardoubleplain`, alle pagine lasciate vuote viene applicato lo stile `plain`.

titlepage, notitlepage Queste opzioni hanno lo stesso effetto di quelle standard. Specificano se dopo il titolo del documento debba avere inizio o no una nuova pagina. La classe `scrartcl` non dà inizio a una nuova pagina per impostazione predefinita, al contrario delle classi `scrreprt` e `scrbook`.

abstracton, abstractoff Nelle classi standard `report` e `article`, l'ambiente `abstract` antepone l'intestazione centrata "Sommario" al testo del sommario. Questa, in passato, era la prassi usuale. Nel frattempo, però, la lettura di giornali e riviste ha abituato i lettori a riconoscere immediatamente un testo centrato posto all'inizio di un articolo o di una relazione come il sommario, senza bisogno di ulteriori precisazioni. Questo capita a maggior ragione quando il sommario precede l'indice. Le classi KOMA-Script forniscono la possibilità di includere o escludere il titolo del sommario con le opzioni `abstracton` e `abstractoff`, rispettivamente. I libri generalmente non hanno un sommario, ma eventualmente un apposito capitolo introduttivo. È questo il motivo per cui la classe `scrbook` non ha alcun ambiente predefinito per il sommario.

pointednumbers, pointlessnumbers L'opzione `pointednumbers` fa seguire un punto alla numerazione di parti, capitoli, paragrafi, sottoparagrafi e didascalie. L'opzione `pointlessnumbers`, predefinita (e consigliata), non usa alcun punto.

fleqn Compone le formule visualizzate allineandole a sinistra (rispetto a un margine rientrato) invece che centrandole.

TABELLA 4: Le opzioni più comuni delle classi KOMA-Script.

Opzione	scrbook	scrreprt	scrartcl
11pt	predefinita	predefinita	predefinita
a4paper	predefinita	predefinita	predefinita
oneside		predefinita	predefinita
twoside	predefinita		
openany		predefinita	<i>indefinita</i>
openright	predefinita		<i>indefinita</i>
headexclude	predefinita	predefinita	predefinita
footexclude	predefinita	predefinita	predefinita
cleardoubleplain	predefinita	predefinita	predefinita
titlepage	predefinita	predefinita	
notitlepage			predefinita
abstractoff	<i>indefinita</i>	predefinita	predefinita
pointlessnumbers	predefinita	predefinita	predefinita
final	predefinita	predefinita	predefinita

leqno Dispone la numerazione delle formule sulla sinistra invece che sulla destra.

draft, final L'opzione **draft** Evidenzia le righe che L^AT_EX non è riuscito a comporre adeguatamente, e che quindi fuoriescono dal margine, con una spessa linea nera sul margine destro, rendendole più facili da individuare. Ciò non accade con l'opzione **final**. Le due opzioni sono caricate anche dagli altri pacchetti e hanno effetto sul loro funzionamento. Per esempio, il pacchetto **graphicx** non carica le immagini quando è specificata l'opzione **draft**: al loro posto viene mostrato un riquadro, di dimensioni corrispondenti, che contiene solo il nome dell'immagine.

1.8.2 Stili di pagina

Analogamente alle classi standard, le classi KOMA-Script accettano diverse combinazioni di testatina/piè di pagina (i cosiddetti stili di pagina). L'argomento *<stile>* del comando

```
\pagestyle{<stile>}
```

stabilisce quale stile sarà utilizzato. Gli stili di pagina più diffusi delle classi di KOMA-Script sono i seguenti.

plain stampa i numeri di pagina al piè di pagina, lasciando vuota la testatina. Questo è lo stile di pagina predefinito.

empty imposta le testatine e i piè di pagina in modo che non vi sia stampato nulla.

headings stampa il titolo del capitolo corrente sulla testatina di ciascuna pagina e il numero di pagina al piè di pagina.

scrheadings consente di disporre di testatine e piè di pagina personalizzati, come ad esempio quelli usati in questo lavoro (richiede il pacchetto **scrpage2**, caricato da **classicthesis**).

1.9 Introduzione ai principali tipi di font

1.9.1 Che cos'è un font?

In tipografia la parola *font* indica un insieme di caratteri accomunati da un certo stile grafico. I caratteri tipografici contenuti in un font, detti anche *glifi*, sono solitamente lettere, numeri e segni di punteggiatura, ma possono anche essere simboli matematici, note musicali, segni geografici, icone, disegni e altro ancora (VAVASSORI e ZANNARINI, 2005).

Una *famiglia di font* è un insieme di font la cui rappresentazione grafica si mantiene fedele a caratteristiche comuni. Una famiglia di font contiene il font con rappresentazione grafica standard e alcune sue varianti.

Esempi di famiglie di font di uso comune sono Palatino, Computer Modern Roman e Times, con le rispettive varianti. Le varianti di stile del font Palatino usate da ClassicThesis sono riportate nella tabella 2 nella pagina precedente.

1.9.2 Larghezza del font: fissa o variabile?

Un parametro che si può prendere in considerazione per una prima classificazione dei diversi font è la larghezza del carattere, che può essere fissa o variabile.

I font a larghezza fissa (detti anche *monospaced* o *typewriter*) producono un testo con caratteristiche simili a quello ottenibile da una macchina per scrivere, che ha la peculiarità di mantenere la stessa larghezza predefinita per ogni carattere della collezione. Tale caratteristica può rivelarsi vantaggiosa se il font è utilizzato all'interno di un editor di testo di un calcolatore o per la stampa di dati incolonnati, ma per le parti testuali di un documento abbastanza lungo è preferibile utilizzare famiglie di font a larghezza variabile.

Esempi di font a larghezza variabile sono Times, Helvetica e Palatino; esempi di font a larghezza



FIGURA 2: Famiglie di font a larghezza variabile.



FIGURA 3: Famiglie di font a larghezza fissa.

fissa sono Courier e Monaco (vedi le figure 2 e 3 nella pagina seguente).

1.9.3 Abbellimenti: con “grazie” o senza?

Un altro parametro che permette di classificare i font è la presenza o meno delle cosiddette “grazie”, che sono piccole rifiniture presenti alle estremità dei caratteri.

I caratteri con grazie sono generalmente considerati più leggibili rispetto a quelli senza grazie in lunghi passaggi (perché le grazie creano, per effetto ottico, una sorta di riga continua che “guida” l’occhio alla lettura di una riga), e sono quindi più comunemente utilizzati in materiale pubblicato professionalmente, come libri e giornali, almeno per le parti testuali.

Per contro, alcuni tipi di font senza grazie offrono una resa migliore rispetto a font serif quando vengono impiegati nella composizione di pagine Web, cataloghi e brochure commerciali. Un’altra applicazione che lascia spazio all’utilizzo di un font senza grazie è quella della visualizzazione di caratteri molto piccoli sullo schermo di un calcolatore: infatti la carenza di dettaglio che contraddistingue il carattere sans serif può conferirgli una maggiore chiarezza.

Famiglie molto utilizzate di font con grazie sono Computer Modern Roman, Times, Palatino e Courier. Alcuni esempi di famiglie senza grazie sono Helvetica, Monaco ed Optima.

TABELLA 5: Alcuni stili delle quattro famiglie Computer Modern.

Computer Modern	
Stile:	Dritto
	<i>Corsivo</i>
	<i>Inclinato</i>
	Neretto
	MAIUSCOLETTA
	Sans serif
	Typewriter

1.9.4 La scelta dei font

In un documento si utilizzano normalmente quattro famiglie di font: una famiglia con grazie, una senza grazie, una famiglia a larghezza fissa e una per le formule matematiche. Scegliere famiglie di font che si combinino bene insieme non è facile, e richiede un’approfondita conoscenza e un’ottima padronanza della tipografia per ottenere risultati adeguati.

In mancanza di istruzioni specifiche, LATEX impiega le quattro famiglie Computer Modern, create da Donald Knuth con il programma METAFONT:

Computer Modern Roman è una famiglia di font con grazie progettata per il testo del documento. È disponibile negli spessori medio e neretto, e nelle varianti dritto, inclinato, corsivo e maiuscoletto.

Computer Modern Sans Serif è una famiglia di font senza grazie, disponibile negli spessori medio e neretto, e nelle varianti dritto ed inclinato.

Computer Modern Typewriter è una famiglia di font a larghezza fissa, disponibile nello spessore medio e nelle varianti dritto, inclinato, corsivo e maiuscoletto.

Computer Modern Mathematics è una famiglia di font specificamente progettata per le formule matematiche. Comprende il corsivo matematico, il neretto e una collezione di caratteri calligrafici.⁴

I font Computer Modern sono descritti da Knuth con grande dettaglio (compreso l’intero codice sorgente) nel libro *Computer Modern Typefaces*, Volume E della serie *Computers and Typesetting* (1986).⁵

4. A rigore, la famiglia matematica Computer Modern è costituita a sua volta da tre sottofamiglie: la prima contiene essenzialmente il corsivo matematico e il greco; la seconda contiene gli operatori binari e i caratteri calligrafici; la terza contiene i grandi operatori e i grandi delimitatori espandibili.

5. Computer Modern non è l’unica famiglia di caratteri disegnata con METAFONT, ma è sicuramente una delle più mature e usate. In particolare, variando i parametri, si pos-

Lo stile ClassicThesis non usa le famiglie Computer Modern: impiega invece la famiglia di font Palatino (o, in alternativa, la famiglia Minion Pro) per il testo, i Bera Mono come font a spaziatura fissa e i font \mathcal{AMS} Euler per la matematica (o, in alternativa, i font matematici della famiglia Palatino); ClassicThesis non usa alcun font sans serif.

Palatino è una famiglia di font con grazie progettata per il testo del documento. È disponibile negli spessori medio, neretto e neretto esteso (questi ultimi due spessori non sono però usati da ClassicThesis), e nelle varianti dritto, inclinato, corsivo e maiuscoletto.

Bera Mono è una famiglia di font a spaziatura fissa, disponibile nello spessore medio e nelle varianti dritto, inclinato, corsivo e maiuscoletto.

\mathcal{AMS} Euler è una famiglia di font progettata per le formule matematiche. Comprende, fra l'altro, il corsivo matematico e una raccolta di caratteri calligrafici.

La famiglia Palatino

Palatino è una famiglia di font con grazie creata da Hermann Zapf nel 1948. È una delle famiglie di caratteri di ispirazione neumanista più diffuse. Equilibrati ed eleganti, i font Palatino sono tra i più celebri tra i caratteri di Zapf, e sono anche tra i più imitati. Maiuscoletto e numeri minuscoli sono elementi essenziali della famiglia.

Chiamati così in onore di Giambattista Palatino, un maestro della calligrafia italiano del sedicesimo secolo, i font Palatino sono basati sui tipi di carattere del Rinascimento italiano, che imitano la scrittura calligrafica. Mentre però i tipi rinascimentali tendenzialmente usavano lettere piccole con linee verticali più lunghe (ascendenti e discendenti) con tratti più fini, i font Palatino hanno proporzioni più ampie, e sono considerati molto più facili da leggere.

La fonte ultima del nome è il *Mons Palatinus*, il colle Palatino a Roma, sito del Tempio di Apollo e di molti palazzi imperiali.

I font Palatino sono stati adattati praticamente a tutte le tecnologie tipografiche; ne esistono versioni liberamente disponibili e commerciali (le fonderie Linotype e Adobe Systems commercializzano entrambe versioni autentiche del Palatino).

La famiglia Bera Mono

Bera Mono è una versione della famiglia di font Bitstream Vera Mono ottimizzata per L^AT_EX. Si tratta di una famiglia di font con una licenza d'uso libera. È stata disegnata da Jim Lyles della

sono generare tipi di caratteri con una perfetta dimensione ottica (i font dal corpo più minuto sono modificati in modo da rimanere leggibili).

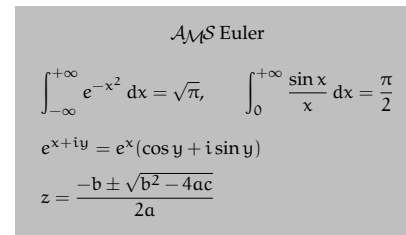


FIGURA 4: Font \mathcal{AMS} Euler.

Bitstream, ed è basata sul disegno della famiglia Bitstream Prima, progettata dallo stesso Lyles. Si tratta di font attraenti e leggibili anche quando sono usati su uno schermo di un calcolatore.

Sebbene la famiglia Bera Mono contenga solo i simboli di punteggiatura comuni e l'alfabeto latino con qualche segno diacritico, la sua licenza permette, con qualche restrizione, di modificarla e distribuire versioni derivate da essa, ed il progetto di sviluppo DejaVu sta espandendo il numero di glifi in modo da coprire tutte le lingue europee.

La famiglia \mathcal{AMS} Euler

Gli \mathcal{AMS} Euler sono una famiglia di font matematici commissionata dalla American Mathematical Society e disegnata da Hermann Zapf e Donald Knuth. Cercano di emulare lo stile della calligrafia di un matematico che scriva simboli matematici sulla lavagna, che è dritto, piuttosto che inclinato. Si mescolano molto bene con altri tipi di carattere disegnati da Zapf, come Palatino, Aldus e Melior, ma molto male con i Computer Modern (i font predefiniti di L^AT_EX).

Il nome \mathcal{AMS} Euler è stato scelto in onore di Leonhard Euler: realizzati in origine con METAFONT, i font \mathcal{AMS} Euler sono stati usati per la prima volta nel libro *Concrete Mathematics* (1988), di cui Knuth fu co-autore, che era dedicato ad Euler.

I font \mathcal{AMS} Euler sono molto gradevoli ed eleganti; va segnalato tuttavia che manca qualche carattere speciale (per esempio, il simbolo della costante di Planck ridotta \hbar non è disponibile: al suo posto viene utilizzato il carattere \hbar del comando `\hslash`). Generalmente, ciò non costituisce un problema.

La famiglia Minion Pro

I Minion Pro sono caratteri disegnati da Robert Slimbach e commercializzati dalla Adobe dal 1989. I Minion Pro costituiscono una famiglia di caratteri da testo neumanisti pienamente sviluppata. Inoltre, in senso tipografico, sono notevolmente economici per quel che riguarda la composizione; infatti permettono di inserire qualche carattere in più per riga, in ogni corpo, rispetto alla maggior parte degli altri caratteri da testo, senza apparire schiacciati o compressi. Il maiuscoletto e i numeri minuscoli, disponibili per ogni variante di peso, in



FIGURA 5: La famiglia Minion Pro.

tondo e corsivo, sono parti essenziali del disegno. La famiglia comprende anche un font di ornamenti tipografici, caratteri con svolazzi e una serie cirillica. Il Minion Pro greco, tondo e corsivo, esiste in forma sperimentale, ma non è stato ancora reso pubblico. Un altro disegno di Slimbach, il carattere cavalleresco corsivo Poetica, si adatta bene ad accompagnare i Minion Pro. I Minion Pro sono i font usati da Robert Bringhurst per comporre i suoi *Elementi dello Stile Tipografico*.

2 Personalizzare ClassicThesis

La ricerca di personalizzazioni di particolari oggetti o dell'intero documento è un processo che presto o tardi tutti si trovano ad affrontare. Di seguito vengono esposti alcuni suggerimenti su come fare in modo che ClassicThesis produca risultati diversi da quelli predefiniti.

2.1 Il tormentone dei margini di pagina

Una delle lamentele più frequenti di chi si accosta a ClassicThesis per la prima volta è che i margini utilizzati sarebbero «troppo ampi» e che il foglio non sarebbe «sufficientemente ben riempito». Prima di buttarsi nella frenesia dell'«allarghiamo un po' questa strettissima pagina» è però bene riflettere. Come per la maggior parte delle cose in L^AT_EX, ci sono ottime ragioni per cui i margini sono quelli che sono.

Uno dei punti di forza di L^AT_EX è che consente di disinteressarsi completamente delle questioni tipografiche, per dar modo all'autore di concentrarsi unicamente sulla struttura e sui contenuti del proprio documento. Questo fatto dovrebbe sempre essere tenuto presente: utilizzando uno stile scritto da altri l'utente accetta per buone tutte le impostazioni tipografiche scelte per lui dall'autore dello stile e non è più tenuto a studiare tipografia per mettere a punto l'aspetto delle proprie pubblicazioni. Questo vale anche per ClassicThesis: modificarne i margini significa contraddire questa filosofia e, di conseguenza, comporta l'obbligo di studiare un (bel) po' di tipografia per ottenere risultati accettabili.

L'esperienza mostra che leggere diventa più difficile se ci sono troppi caratteri per riga: ciò dipende

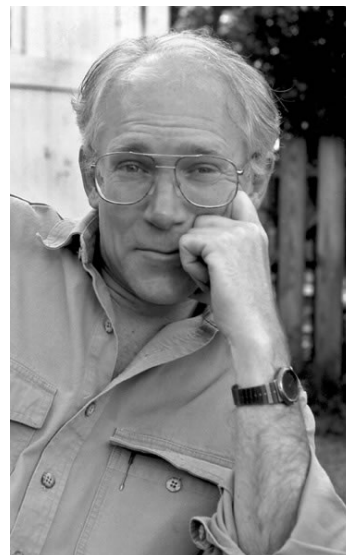


FIGURA 6: Robert Bringhurst (immagine tratta da <http://www.griffinpoetryprize.com>).

dal fatto che l'occhio si affatica spostandosi dalla fine di una riga all'inizio della successiva (questo è il motivo per cui i giornali vengono stampati su più colonne). Questa esperienza è stata codificata da Bringhurst nella celebre regola che considera come ottimale il valore di circa 66 caratteri per riga, contando anche gli spazi, indipendentemente dal font usato. Come ampiezza media di un carattere viene assunto il rapporto tra la lunghezza dell'alfabeto inglese minuscolo ed il numero di lettere che lo compongono (26). La regola di Bringhurst fissa la lunghezza di riga in funzione del font impiegato, in modo da avere un numero medio di caratteri per riga ottimale.

Per esempio, per caratteri Palatino di 10 punti (usati da ClassicThesis in mancanza di istruzioni specifiche), la lunghezza dell'alfabeto è 133 punti, il che porta a un valore ottimale della lunghezza di riga compreso tra 288 punti, cui corrispondono circa 61 caratteri per riga, e 312 punti (che è proprio il valore adottato da ClassicThesis), cui ne corrispondono circa 66 (BRINGHURST, 1992, p. 26).

Bisogna tener conto del fatto che la lunghezza di riga adottata da ClassicThesis, di suo, è già al limite superiore di quanto stabilito dalla regola di Bringhurst per consentire un maggiore riempimento delle pagine in formato A4, su cui si assume avvenire la stampa (come di norma, in Europa, in ambiente universitario o casalingo).

Gli eleganti margini ampi, inoltre, consentono l'impiego delle note a margine, che sono utili per riassumere il corpo del testo e per dare un tocco di vitalità alla pagina, e lasciano spazio per eventuali annotazioni del proprio relatore (se si sta scrivendo una tesi).

Per queste ragioni è *decisamente sconsigliabile* modificare i margini predefiniti di ClassicThesis. Le proporzioni di pagina sono qualcosa di molto

personale e relativo, quindi se l'utente è soddisfatto da quelle messe a disposizione da *ClassicThesis* e trova allettante l'idea di disinteressarsi del problema della loro definizione, allora potrà utilizzare con soddisfazione *ClassicThesis*; altrimenti, se un autore ha esigenze diverse o non è soddisfatto dalla resa grafica dello stile, allora può provare a rivolgersi ad altre classi o pacchetti, o a costruirsi da solo le dimensioni di pagina che corrispondono alle proprie aspettative.

Se un utente necessita di proporzioni di pagina differenti da quelle predefinite di *ClassicThesis*, per esempio perché è obbligato a seguire delle indicazioni imposte dalla sua università, va tenuto presente che le classi *KOMA-Script*, su cui *ClassicThesis* si basa, hanno il proprio sistema per impostare i parametri tipografici, e per determinare le proporzioni di pagina usano il pacchetto *typearea* (la cui documentazione è un capitolo del manuale di *KOMA-Script*), che ha una serie di impostazioni predefinite, calcolate in base al carattere scelto, e che è *incompatibile* con i pacchetti *layaureo* e *geometry* (funzionanti con le classi *book*, *report* e *article*). In caso di necessità è possibile impostare a mano le dimensioni del corpo del testo, mediante un comando del tipo

```
\areaset[⟨rilegatura⟩]{⟨larghezza⟩}{⟨altezza⟩}
```

dove i parametri $\langle rilegatura \rangle$, $\langle larghezza \rangle$ e $\langle altezza \rangle$ rappresentano rispettivamente la correzione per la rilegatura, la larghezza e l'altezza del corpo del testo, tenendo conto delle opzioni di classe *headinclude*, *headexclude*, *footinclude* e *footexclude* (discusse nel paragrafo 1.8 nella pagina 56).

I valori predefiniti per una pagina A4 con il font Palatino in corpo 10 (e le opzioni di classe *headinclude* e *footinclude*) sono

```
\areaset[5mm]{312pt}{699pt}
```

dove il valore di 699 punti è stato ottenuto sommando i 624 punti dell'altezza del corpo del testo con i 33 punti delle testatine ed i 42 del piè di pagina.

2.2 I miniindici

Quando i capitoli hanno una struttura particolarmente complessa, può essere conveniente riportare nella pagina iniziale l'indice del capitolo. Questi miniindici possono essere prodotti automaticamente con il pacchetto *minitoc*:

```
\usepackage[tight,italian]{minitoc}
```

Per avere le intestazioni "Indice" di ciascun miniindice in maiuscolo spaziatto e non in neretto (che *ClassicThesis* non usa) è sufficiente scrivere

```
\def\ptctitle{indice}
\def\mtctitle{indice}
\def\stctitle{indice}
\setlength{\mtcindent}{0pt}
```

```
\renewcommand{\mtifont}{\normalsize%
\scshape\lsstyle}
```

nel preambolo.

Naturalmente, devono essere presenti il comando `\dominitoc` prima di `\tableofcontents` ed i comandi `\minitoc`/`\mtcskip` subito dopo i vari `\myChapter`.

2.3 Migliorare la spaziatura del maiuscolo

In generale, \LaTeX cerca di produrre sempre le migliori interruzioni di riga possibili; se non riesce a trovare il modo di spezzare le righe secondo i suoi severi criteri, lascia che la riga fuoriesca dal margine destro, avvertendo l'utente con un messaggio di "overfull hbox".⁶

Nelle versioni di *ClassicThesis* precedenti la 2.0, le situazioni di fuoriuscita di una riga dal margine destro nei titoli dei paragrafi e dei capitoli erano abbastanza comuni. Questi problemi erano dovuti al fatto che *ClassicThesis* usava il pacchetto *soul* per realizzare la spaziatura tra i caratteri del maiuscolo: la spaziatura (formata da uno spazio fisso più un'eventuale correzione, la cosiddetta "crenatura") era realizzata "carattere per carattere" e non riusciva a tenere conto di tutti i fattori necessari.

L'uso di *soul* per la spaziatura del maiuscolo dovrebbe essere eliminato a favore del metodo adottato dal pacchetto *microtype*, che non soffre delle fragilità di *soul* e realizza la spaziatura agendo direttamente al livello del font. Perché ciò avvenga, è sufficiente caricare *classicthesis* con l'opzione *pdfspacing*:

```
\usepackage[⟨...⟩,pdfspacing]{classicthesis}
```

Occorre una versione aggiornata di *microtype* (almeno la 2.0).⁷

2.4 Scrivere un'introduzione con collegamenti ipertestuali

Supponiamo di avere un documento composto da un'introduzione e da diversi capitoli:

```
\documentclass[⟨...⟩]{scrreprt}
\usepackage{classicthesis-ldpkg}
\usepackage[⟨...⟩]{classicthesis}
\begin{document}
\myChapter{Introduzione}
\myChapter{La pizza}
\myChapter{La birra}
\end{document}
```

6. Benché \LaTeX avverta l'utente quando ciò accade, le righe a cui si riferisce non sempre sono facili da trovare. Usando l'opzione *draft* nel comando `\documentclass`, queste righe saranno evidenziate con una spessa linea nera sul margine destro.

7. Se si usa l'opzione globale *draft*, il pacchetto *microtype* viene disabilitato. Se si vuole che venga usato sempre, è necessario caricarlo con l'opzione *final*.

Se si desidera inserire nell'introduzione un'indicazione del tipo:

```
L'esposizione del lavoro è articolata
come segue:
\begin{description}
\item[Nel primo capitolo] si parla di pizza.
\item[Nel secondo capitolo] si parla
di birra.
\end{description}
```

in modo che cliccando su “secondo capitolo”, “terzo capitolo”, ..., il lettore venga portato nelle pagine corrispondenti, è sufficiente strutturare il documento nel modo seguente:

```
\documentclass[...]{scrreprt}
\usepackage{classicthesis-ldpkg}
\usepackage[...]{classicthesis}
\begin{document}
\myChapter{Introduzione}
\begin{description}
\item[{\hyperref[cap:pizza]{Nel primo
capitolo}}] si parla di pizza.
\item[{\hyperref[cap:birra]{Nel secondo
capitolo}}] si parla di birra.
\end{description}
\myChapter{La pizza}\label{cap:pizza}
\myChapter{La birra}\label{cap:birra}
\end{document}
```

2.5 Inserire uno sfondo colorato in un'immagine

Se si desidera inserire uno sfondo colorato in un'immagine (per esempio del colore grigio chiaro utilizzato come sfondo della figura 7b), è possibile definire un comando ad hoc nel preambolo:

```
\newcommand{\myincludegraphics}[2][]{%
\begin{group}\setlength{\fboxsep}{0pt}%
\colorbox{lightgray}{%
\includegraphics[#1]{#2}}%
\end{group}}
```

da usare nel modo seguente:

```
\myincludegraphics[...]{(immagine)}
```

Affinché tutto funzioni, è fondamentale che l'immagine da inserire abbia lo sfondo trasparente. Per sfondi di colore diverso si rimanda alla ricca documentazione di `xcolor`.

2.6 Personalizzare la numerazione delle didascalie

Con *ClassicThesis*, le figure vengono numerate progressivamente come “Figura 1”, “Figura 2”, ...

Se si sta scrivendo un libro o una relazione e si desidera ripristinare la numerazione standard collegata al capitolo corrente (“Figura 1.1”, “Figura 1.2”, ..., “Figura 2.1”, “Figura 2.2”, ...) è sufficiente scrivere



(a) Immagine senza sfondo.



(b) Immagine con sfondo.

FIGURA 7: Inserire uno sfondo colorato in un'immagine.

```
\numberwithin{figure}{chapter}
```

dopo aver caricato *classicthesis* con le opzioni `pointlessnumbers` e `listsseparated`:

```
\usepackage[...]{pointlessnumbers,
listsseparated}{classicthesis}
```

Naturalmente, se si desidera fare lo stesso con le tabelle, basta sostituire `table` al posto di `figure` nel codice appena scritto.

2.7 La matematica

Uno dei principali punti di forza di \LaTeX è sicuramente la sua capacità nella composizione e nella gestione delle formule matematiche. Se si sta scrivendo un lavoro di tipo scientifico, è conveniente abilitare i font matematici forniti dall'American Mathematical Society (che costituiscono una preziosa estensione dei font matematici disponibili) con il comando

```
\usepackage{amsmath}
```

Il pacchetto `amsmath`, che fornisce svariate estensioni per il miglioramento della gestione di documenti che contengono formule matematiche, è caricato automaticamente da *classicthesis-ldpkg*.

2.7.1 Personalizzare lo stile di definizioni e teoremi

Componendo documenti matematici, è utile disporre di un metodo per introdurre e numerare definizioni, teoremi e strutture simili; \LaTeX mette a disposizione il comando `\newtheorem`.

Per evidenziare adeguatamente definizioni e teoremi è possibile impiegare, al posto del neretto

(che ClassicThesis non usa), il maiuscolo per l'intestazione, accompagnandolo con una spaziatura prima e dopo l'enunciato, per "staccarlo" visivamente dal resto. A tal fine, è possibile usare il pacchetto `amsthm` ed il seguente codice:

```
\newtheoremstyle{classicdef}% Nome
{12pt}% Spazio che precede l'enunciato
{12pt}% Spazio che segue l'enunciato
{}% Stile del font dell'enunciato
{}% Rientro (se vuoto, non c'è rientro;
%\parindent = rientro dei capoversi)
{\scshape}% Stile del font dell'intestazione
{:}% Punteggiatura che segue
% l'intestazione
{.5em}% Spazio che segue l'intestazione:
% " " = normale spazio inter-parola;
%\newline = a capo
{}% Specifica l'intestazione
% dell'enunciato
% (normalmente viene lasciata vuota)
```

```
\newtheoremstyle{classicthm}% Nome
{12pt}% Spazio che precede l'enunciato
{12pt}% Spazio che segue l'enunciato
{\itshape}% Stile del font dell'enunciato
{}% Rientro (se vuoto, non c'è rientro;
%\parindent = rientro dei capoversi)
{\scshape}% Stile del font dell'intestazione
{:}% Punteggiatura che segue
% l'intestazione
{.5em}% Spazio che segue l'intestazione:
% " " = normale spazio inter-parola;
%\newline = a capo
{}% Specifica l'intestazione
% dell'enunciato
% (normalmente viene lasciata vuota)
```

```
\theoremstyle{classicdef}
\newtheorem{definizione}{Definizione}
% [section] o [chapter]

\theoremstyle{classicthm}
\newtheorem{teorema}{Teorema}
% [section] o [chapter]
```

2.7.2 Lettere greche minuscole

Alcune lettere greche minuscole hanno delle forme varianti i cui nomi cominciano con l'abbreviazione `var` (vedi la tabella 6); esse negli Stati Uniti sono delle varianti, ma in Europa (ad eccezione di `\varsigma`) sono considerate le forme principali.⁸

Per scrivere documenti in accordo con il gusto europeo, è utile ridefinire le lettere greche varianti (ad eccezione di `\varsigma`) come normali. A tal fine, è sufficiente scrivere

```
\renewcommand{\epsilon}{\varepsilon}
\renewcommand{\theta}{\vartheta}
```

8. Con i font \mathcal{AMS} Euler, `\rho` e `\varrho` producono lo stesso risultato; lo stesso accade per `\sigma` e `\varsigma` (vedi la tabella 6). Dal momento che, in un documento, per ciascuna lettera si sceglie in alternativa la forma principale o la sua variante, non c'è pericolo di confusione.

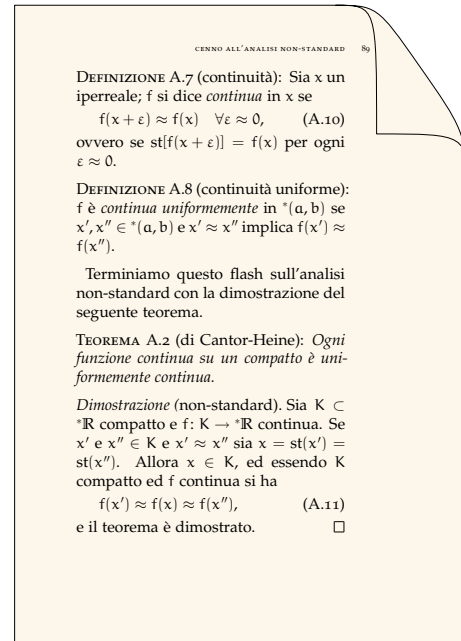


FIGURA 8: Definizioni e teoremi dallo stile personalizzato.

TABELLA 6: Lettere greche minuscole: forme principali e varianti.

ϵ	<code>\epsilon</code>	ε	<code>\varepsilon</code>
θ	<code>\theta</code>	ϑ	<code>\vartheta</code>
ρ	<code>\rho</code>	ϱ	<code>\varrho</code>
ϕ	<code>\phi</code>	φ	<code>\varphi</code>

(a) Font Palatino.

ϵ	<code>\epsilon</code>	ε	<code>\varepsilon</code>
θ	<code>\theta</code>	ϑ	<code>\vartheta</code>
ρ	<code>\rho</code>	ϱ	<code>\varrho</code>
ϕ	<code>\phi</code>	φ	<code>\varphi</code>

(b) Font \mathcal{AMS} Euler.

```
\renewcommand{\rho}{\varrho}
\renewcommand{\phi}{\varphi}
```

dopo aver caricato il pacchetto `classicthesis`. Il codice deve seguire il caricamento del pacchetto in quanto ClassicThesis si occupa di scegliere i font matematici, e questa azione definisce anche il significato delle lettere greche.

2.7.3 Incompatibilità con `mathtools` e francese

Se si desidera usare contemporaneamente con ClassicThesis il pacchetto `mathtools` e il pacchetto `babel` con l'opzione `french` (che abilita la lingua francese), il pacchetto `mathtools` va caricato prima di `babel` e dopo `classicthesis`. In altre parole, bisogna caricare i pacchetti nell'ordine seguente:

```
\usepackage[...]{classicthesis-ldpkg}
\usepackage[...]{classicthesis}
\usepackage{mathtools}
\usepackage[...]{french, ...}{babel}
```

Se non si rispetta quest'ordine, si ottiene un errore. Il motivo è che sia `mathtools` sia l'opzione `french` di `babel` rendono attivo il carattere `:`, dandone ovviamente una definizione diversa.

2.8 Le appendici

Dal momento che ClassicThesis non scrive l'intestazione "Capitolo" né nei titoli dei capitoli né nelle testatine né nell'indice, è consigliabile, qualora il proprio documento contenga delle appendici, adottare lo stesso criterio. A tal fine, è sufficiente strutturare il proprio documento semplicemente così:

```
\documentclass{scrreprt}
\usepackage{classicthesis-ldpkg}
\usepackage[...]{classicthesis}
\usepackage{lipsum}
\begin{document}
\tableofcontents
\myChapter{Pizza}
\lipsum
\myChapter{Birra}
\lipsum
\appendix
\myChapter{Caffè}
\lipsum
\end{document}
```

2.9 La bibliografia

2.9.1 Creare la bibliografia

La bibliografia può essere gestita con grande efficienza e flessibilità da \LaTeX . Essa può essere creata manualmente per mezzo dell'ambiente `thebibliography`, tuttavia è decisamente preferibile l'utilizzo del programma `BIB \TeX` , che permette di separare il contenuto dei riferimenti bibliografici (archiviati in apposite basi di dati) dalla loro forma (gestita da stili). Per generare la bibliografia con ClassicThesis bisogna:

1. compilare il documento con \LaTeX ;
2. eseguire `BIB \TeX` sul documento;
3. ricompilare *tre* volte con \LaTeX per includere la bibliografia e aggiornare tutti i riferimenti (compresi quelli finali, generati da ClassicThesis mediante il pacchetto `backref`, che contengono l'indicazione delle pagine in cui le voci bibliografiche sono citate).

Uno strumento particolarmente utile e versatile nella gestione della bibliografia è costituito dal pacchetto `natbib` (caricato automaticamente da `classicthesis-ldpkg`), che è accompagnato da un'esauriente documentazione.

2.9.2 Personalizzare la bibliografia

Per creare dei riferimenti bibliografici ottimizzati per la lingua italiana e per ClassicThesis, è possibile usare lo stile `classic.bst`.⁹

Una volta opportunamente posizionato `classic.bst` — la posizione dipende dalla distribuzione di \LaTeX che si usa, ma in genere è sufficiente collocare il file nella cartella `/bibtex/bst/` dell'albero personale (PANTIERI, 2008, p. 25) — basta scrivere

```
\manualmark
\markboth{\spacedlowsmallcaps{\bibname}}%
         {\spacedlowsmallcaps{\bibname}}
\refstepcounter{dummy}
\addcontentsline{toc}{chapter}%
{\tocEntry{\bibname}}
\bibliographystyle{classic}
\bibliography{Bibliografia}
```

nel punto in cui si desidera che compaia la bibliografia.

Lo stile ClassicThesis (mediante il pacchetto `backref`) permette di avere, nelle voci bibliografiche, l'indicazione delle pagine in cui esse sono citate ("Citato a pagina..."). Per avere i riferimenti in italiano anziché in inglese, è sufficiente scrivere

```
\renewcommand*{\backreftwosep}{ e-}
\renewcommand*{\backreflastsep}{ e~}
\renewcommand*{\backrefalt}[4]{%
  \ifcase #1 %
    \relax%
  \or
    (Citato a pagina~#2.)%
  \else
    (Citato alle pagine~#2.)
  \fi}
```

dopo aver caricato il pacchetto `classicthesis-ldpkg`.

2.10 L'indice analitico

2.10.1 Creare l'indice analitico

Come la bibliografia, anche l'indice analitico può essere gestito automaticamente da \LaTeX . Creare l'indice analitico con ClassicThesis è molto semplice: basta posizionare il codice

```
\manualmark
\markboth{\spacedlowsmallcaps{\indexname}}%
         {\spacedlowsmallcaps{\indexname}}
\refstepcounter{dummy}
\pagestyle{scrheadings}
\addcontentsline{toc}{chapter}%
{\tocEntry{\indexname}}
\printindex
```

nel punto in cui si desidera che compaia l'indice analitico (tipicamente, alla fine del documento, subito prima di `\end{document}`). In questo modo,

9. Il file `classic.bst` è scaricabile da <http://www.lorenzopantieri.net/LaTeX.html>.

nel documento finale vengono creati correttamente il relativo segnalibro e le testatine.

Naturalmente, è necessario caricare il pacchetto `makeidx` e servirsi del programma `MakeIndex`. Nel preambolo deve essere presente anche il comando `\makeindex`.

2.10.2 Personalizzare l'indice analitico

L'indice analitico che si ottiene con le impostazioni predefinite non è molto elegante. È possibile ricorrere a un apposito file, che chiameremo `classic.ist`, che contiene le istruzioni per personalizzare l'indice analitico, così definito:

```
headings_flag 1
heading_prefix "\n \item \textsc{"
heading_suffix "}"
symhead_positive "Simboli"
symhead_negative "simboli"
numhead_positive "Numeri"
numhead_negative "numeri"
```

Una volta registrato e opportunamente posizionato `classic.ist` — la posizione, al solito, dipende dalla distribuzione di \LaTeX che si usa, ma in genere è sufficiente collocare il file nella cartella `/makeindex/` dell'albero personale (PANTIERI, 2008, p. 25) — le istruzioni per la compilazione dell'indice analitico sono le seguenti:

1. si compila una prima volta con \LaTeX il sorgente;
2. si elabora l'indice analitico (dalla linea di comando: `makeindex -s classic <nome del file>`);
3. si compila altre due volte con \LaTeX .

Per bilanciare le colonne dell'ultima pagina dell'indice analitico è sufficiente inserirlo all'interno di un ambiente `multicols` (deve essere caricato il pacchetto `multicol`). A questo scopo è possibile ridefinire l'ambiente `theindex` con il seguente codice

```
\let\orgtheindex\theindex
\let\orgendtheindex\endtheindex
\def\theindex{%
    \def\twocolumn{\begin{multicols}{2}}%
    \def\onecolumn{}%
    \clearpage
    \orgtheindex}
\def\endtheindex{%
    \end{multicols}%
    \orgendtheindex}
```

da scrivere nel preambolo.

Riferimenti bibliografici

AMERICAN MATHEMATICAL SOCIETY (1999). *Manuale d'uso del pacchetto amsmath*. <http://tug.ctan.org/tex-archive/info/italian/amslldoc/itamsldoc.pdf>.

BECCARI, C. (2008). *Introduzione all'arte della composizione tipografica con \LaTeX* . <http://www.guit.sssup.it/downloads/GuidaGuIT.pdf>.

BRINGHURST, R. (2001, ed. or. 1992). *Gli Elementi dello Stile Tipografico*. Sylvestre Bonnard, Milano.

BUSDRAHI, F. (2004). *Documentazione del pacchetto layaureo*. <http://www.guit.sssup.it/downloads/layaureo.pdf>.

CEVOLANI, G. (2006). «Norme tipografiche per l'italiano in \LaTeX ». *ArSTeXnica*, 1. <http://www.guit.sssup.it/arstexnica.php>.

FEAR, S. (2005). *Publication quality tables in \LaTeX* . Manuale d'uso del pacchetto `booktabs`, <http://www.ctan.org/tex-archive/macros/latex/contrib/booktabs/booktabs.pdf>.

GOOSSENS, M., MITTELBACH, F. e SAMARIN, A. (2004). *The \LaTeX Companion*. Addison-Wesley, Reading (Massachusetts).

GRAHAM, R. L., KNUTH, D. E. e PATASHNIK, O. (1994). *Concrete Mathematics*. Addison-Wesley, Reading (Massachusetts).

GREGORIO, E. (2006). *\LaTeX , breve guida ai pacchetti di uso più comune*. <http://profs.sci.univr.it/~gregorio/breveguida.pdf>.

HIMMELMANN, M. e MORI, L. F. (2007). «Scrivere il curriculum vitae». *ArSTeXnica*, 4. <http://www.guit.sssup.it/arstexnica.php>.

KNUTH, D. E. (1984). *The \TeX book*. Addison-Wesley, Reading (Massachusetts).

KOHM, M. (2007). *KOMA-Script, a versatile \LaTeX 2_ε bundle*. Manuale d'uso della classe KOMA-Script, <http://www.ctan.org/tex-archive/macros/latex/contrib/koma-script/scrguien.pdf>.

LAMPORT, L. (1994). *\LaTeX : a document preparation system*. Addison-Wesley, Reading (Massachusetts).

MIEDE, A. (2007). *A Classic Thesis style*. Manuale d'uso dello stile ClassicThesis, <http://www.ctan.org/tex-archive/macros/latex/contrib/classicthesis/ClassicThesis.pdf>.

MORI, L. F. (2007). «Scrivere la tesi di laurea con \LaTeX 2_ε». *ArSTeXnica*, 3. <http://www.guit.sssup.it/arstexnica.php>.

OETIKER, T. (2000). *Una (mica tanto) breve introduzione a \LaTeX 2_ε*. <http://www.ctan.org/tex-archive/info/italian/lshort/itlshort.pdf>.

PAKIN, S. (2005). *The Comprehensive L^AT_EX Symbol List*. <http://www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-a4.pdf>.

PANTIERI, L. (2008). *L'arte di scrivere con L^AT_EX*. Un'introduzione a L^AT_EX 2_ε, http://www.lorenzopantieri.net/LaTeX_files/ArteLaTeX.pdf.

VAVASSORI, E. G. e ZANNARINI, E. (2005).

L^AT_EX e i font: installazione pratica. <http://www.guit.sssup.it/guitmeeting/2005/articoli/zannarini-vavassori.pdf>.

▷ Lorenzo Pantieri
www.lorenzopantieri.net
sit6113 at iperbole dot
bologna dot it

T_EX Live 2008 and the T_EX Live Manager

Norbert Preining

Abstract

T_EX Live 2008 has been released recently, and the DVDs are ready to go gold. This is the first release of T_EX Live shipping the T_EX Live Manager, `tlmgr` for short.

Besides taking over some of the tasks from `texconfig` (which has never been available for Windows) it finally brings many new features to the T_EX Live world, most importantly the option for dynamic updates.

This article will present the new T_EX Live Installer, the T_EX Live Manager, and at the end lists other changes in T_EX Live 2008.

Sommario

T_EX Live 2008 è stata recentemente rilasciata, e i DVD sono pronti per la produzione. Questo è il primo rilascio di T_EX Live equipaggiata di T_EX Live Manager, in breve `tlmgr`.

Oltre a farsi carico di alcune delle procedure di `texconfig` (mai stato disponibile per Windows), reca finalmente molte nuove caratteristiche nel mondo di T_EX Live: principalmente l'opzione per gli aggiornamenti dinamici.

L'articolo presenta il nuovo T_EX Live Installer, il T_EX Live Manager, ed elenca infine le altre modifiche in T_EX Live 2008.

Important note

This article describes the status of the T_EX Live Manager as it will be shipped around October 2008, and not the one on DVD. The version on the DVD works fine for local configuration tasks (which is why we felt it could be shipped), but is not sufficiently robust for reliable updates over the Internet. Users' first update will be to get the new `tlmgr`.

1 Introduction

After more than one year of development work T_EX Live 2008 has been released with a complete new infrastructure (PREINING, 2007). At first these infrastructure changes were only relevant for the developers themselves, since it made life (a bit) easier and the system more consistent due to the elimination of duplicated information.

As a first user-visible change came the unification of the installer, so that all supported platforms now share the same installer. Furthermore, this installer has gotten a GUI which also is uniform across all platforms. On Unix systems the only

prerequisites are a Perl installation, and for the GUI the installation of Perl/Tk. On Windows we ship a minimal Perl with the necessary modules.

The first part of this article will give an overview of the new installer.

The second user-visible change came from the addition of the T_EX Live Manager, or `tlmgr` for short, to the list of programs. It manages an existing T_EX Live installation, both packages and options. Besides performing many of the same actions as `texconfig` it has the ability to install additional packages, update and remove existing ones, make backups, search within and list all packages.

2 The new installer

The creation of a new T_EX Live installer was necessitated by the new package infrastructure (PREINING, 2007). From a user's point of view the new installer has only one visual change, but there other significant changes. In particular:

- It is possible to install T_EX Live from the Internet.
- There is just one installer, which can run either in text mode, emulating the former `install-tl.sh` shell script, or in GUI mode (more or less emulating the T_EX Live 2007 `tlpmgui`).
- The Windows installation is much closer to using the same implementation as Unix.

2.1 Install T_EX Live from the Internet

If you got (by the time you read this article) a T_EX Live DVD you can just start the installer as usual. On Windows this will be by default the GUI installer (see below), on Unix the text mode installer.

We also ship an installation package (KOTUCHA *et al.*, 2008) containing all the necessary files for an installation over the network. By default, normal installation will use a CTAN mirror, selected using the <http://mirror.ctan.org> service. (See <http://tug.org/ctan.html#sites>.)

Two installers for network downloads are provided. `install-tl-unx.tar.gz` supports Unix only. `install-tl.zip` additionally contains a small subset of Perl for Windows which is required to bootstrap the system. The latter works on all platforms supported by T_EX Live. The sole reason for providing a separate package for Unix is its significantly smaller size.

In any case you can override the source from which you want to install with the command line option `-location`.

2.2 The text mode installer

If you have used \TeX Live in recent years you will see no big changes in the text mode installer (see fig. 1); we tried to keep it as close as possible to the one used in former releases. One new option is at the bottom of the menu, namely *set up for running from DVD*. This is what we call *live* installation: it sets up a minimal writable environment on your computer, while all the input files and binaries remain on the DVD.

```
===== TeX Live installation procedure =====
==> Note: Letters/digits in <angle brackets> indicate menu items <==
==>         for commands or configurable options         <==

Proposed platform: Intel x86_64 with GNU/Linux

<B> binary systems: 1 out of 15

<S> Installation scheme (scheme-full)

Customizing installation scheme:
<C> standard collections
<L> language collections
83 collections out of 84, disk space required: 1426 MB

<D> directories:
  TEXDIR (the main TeX directory):
    /usr/local/texlive/2008
  TEXMFLOCAL (directory for site-wide local files):
    /usr/local/texlive/texmf-local
  TEXMFSYSVAR (directory for variable and automatically generated data):
    /usr/local/texlive/2008/texmf-var
  TEXMFSYSCONFIG (directory for local config):
    /usr/local/texlive/2008/texmf-config
  TEXMFHOME (directory for user-specific files):
    ~/texmf

<O> options:
  [ ] use letter size instead of A4 by default
  [X] create all format files
  [X] install macro/font doc tree
  [X] install macro/font source tree
  [ ] create symlinks in standard directories

<V> set up for running from DVD

Other options:
=====
<I> start installation to hard disk
<H> help
<Q> quit

Enter command:
```

FIGURE 1: Main menu of the text mode installer

2.3 The GUI Installer

The GUI installer has nearly the same functionality as the text version; the option to set up for live installation is the only missing piece. It is written in Perl/Tk and thus should run on all platforms (on Unix Perl/Tk has to be installed).

The main window can be seen in fig. 2. It should remind you very much of the text mode installer. As with the text mode installer it allows you to change which binary systems should be installed (fig. 3), select the scheme to be installed (fig. 4), where a *scheme* is a pre-defined set of collections to be installed, and further specify in more detail which collections (a collection is a set of packages) and which language packages to install (fig. 5 and 6). You can select the installation directories for \TeX Live and toggle some options, all in line

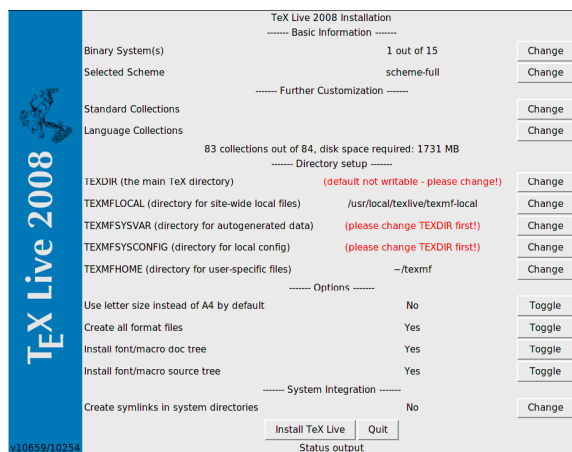


FIGURE 2: Main window of the GUI installer

with the installer of recent years and the text mode installer.

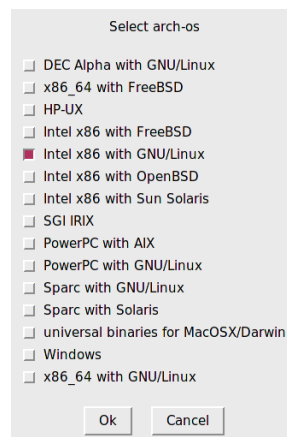


FIGURE 3: Binary system select window

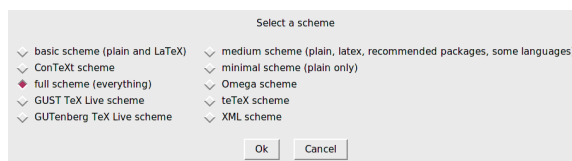


FIGURE 4: Scheme select window

During installation the main window's status line will indicate what is going on currently, and at the same time the program will print out to the terminal the same output as the normal installer.

For both the text mode installer and the GUI mode installer, a log file with more details is created in the installation directory as `install-tl.log`. (If you report installation problems, please send us this log file.)

2.4 Bringing Windows in line with Unix

\TeX Live 2008 supports Windows 2000 and later. By dropping older Windows versions, there is much less need to treat Windows specially.

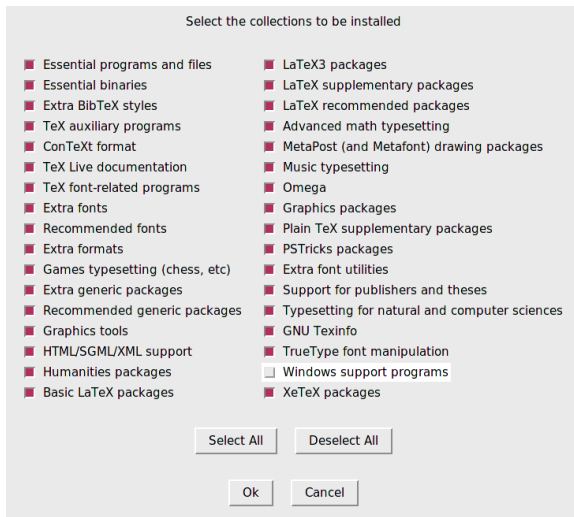


FIGURE 5: Collections select window

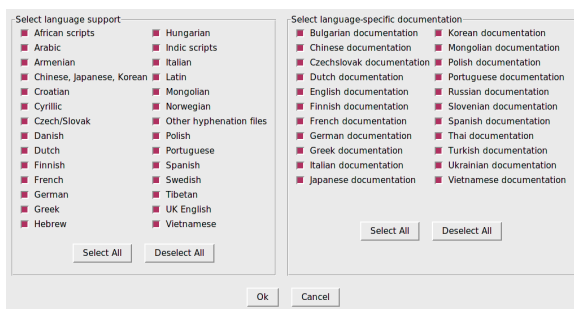


FIGURE 6: Language packs select window

Under Windows 2000 and later, users have a real home directory, viz. `%USERPROFILE%`, usually `C:\Documents and Settings\username`.

This is now reflected in tilde expansion by Kpathsea: `~/texmf` is expanded to `%USERPROFILE%\texmf` under Windows and to `$HOME/texmf` under Unix.

It is also possible to differentiate between system settings and user settings. Happily, there is no longer any need to have a different set of `texmf` trees on Windows, or to leave out scripts such as `fmtutil-sys` and `updmap-sys`. We also have a single `texmf.cnf` used on all platforms.

3 The TeX Live Manager

TeX Live Manager provides a wealth of options and commands; we will explain them all here, some tersely, some in more detail. Of course we expect to add more features in the future.

3.1 The TeX Live Database

First of all, it is important to understand where all the information about installed packages and other options are saved. This is the TeX Live database, which normally can be found in `ROOT/tlpkg/texlive.tlpdb` (where `ROOT` is the destination folder you have chosen at installation time).

It contains the list of all packages, all the files installed, and in addition to that collects configuration information like the default installation source, and the options you have set at installation time (e.g., whether you want A4 or letter-size paper by default).

Most of `tlmgr`'s actions will load the local database, and many actions will also load a remote database: If you want to install a package, the TeX Live Manager will load the database of the specified installation server and checks whether this package exists.

Although we say *remote*, it is not necessarily a remote network location. If you install from DVD the default installation source will be the DVD, and `tlmgr` will load the database located on the DVD when needed.

3.2 General syntax of `tlmgr`

The general syntax of `tlmgr` is

```
tlmgr [opt]... action [opt]... [arg]...
```

The first set of options before the main **action** configure the general operation of `tlmgr`, while the second set of options are specific to the chosen **action**. We do not support mixing and reshuffling of all these options, partly for the sake of clarity, and also for programming reasons.

The first set of options can contain

- `--location loc` specifies the location from which packages should be installed or updated, overriding the location found in the installation's TeX Live Package Database (TLPDB).
- `--gui` starts the GUI version of `tlmgr`. The GUI does not support all the bells and whistles of the command-line program. It is in fact a separate program that calls the command-line version to do the actual work. The difference between this `--gui` option and the `gui` action (see below) is that given the option, `tlmgr` tries to open the GUI directly at the screen for the specified **action**.
- `--gui-lang ll` selects the language in which the GUI will appear. Normally the GUI tries to deduce your language from the environment (on Windows via the registry, on Unix via `LC_MESSAGES`). If that fails you can select a different language by giving this option a two-letter language code.

Furthermore, some standard options are supported: `--help` (also `-h`) for getting help, `-q` which suppresses informational messages, and `-v` (verbose) for turning on debugging options. With `--version` the script will show you the version of your TeX Live system and of itself.

3.3 The actions

There is a (permanently growing) list of actions, currently: `help`, `version`, `gui`, `install`, `update`, `backup`, `restore`, `remove`, `option`, `paper`, `arch`, `search`, `show`, `list`, `check`, `uninstall`, `generate`.

The general actions

- **search** [*option*...] *what* Without any options, search in the list of locally installed packages for package names or descriptions matching *what*. If you give the option `--global` it also searches the remote database. This might differ in case you have not installed all of \TeX Live, but only a part of it. Finally, if you specify `--file` then files are searched, and not package names.
- **show** *pkg*... gives you more detailed information on the listed packages. If all packages are installed locally, it does not consult the remote database.
- **list** [*collections*|*schemes*] With no argument, lists all packages available at the default install location, prefixing those already installed with "i ". With an argument lists only collections or schemes, as requested.
- **uninstall** This action will ask for confirmation, and then remove the entire installation. Don't do it or we will be sad. If you give the `--force` option, it does not even ask, but proceeds immediately with the removal.
- **check** [*files*|*collections*|*all*] Executes one (or all) check(s) on the consistency of the installation. For *files* it checks that all files listed in the local database are actually present, and lists those missing. The option `--use-svn` will use the `svn` command to check for the files.
- **gui** starts the GUI, as explained above at `--gui`.
- **version** is the same as `--version`.
- **help** is the same as `--help`.

The configuration actions

- **option** [*show*] Shows all configuration settings currently saved in the local database. The *show* option is accepted and ignored.
- **option** *key* [*value*] Without the *value*, shows the current value of the configuration option *key*; with *value*, sets this configuration option. Currently accepted *keys* are *location* (default installation location), *formats* (create formats at installation time), *docfiles* (install documentation files), *srcfiles* (install

source files). These are the options you have set at installation time and will be honoured at *later* install and upgrade actions. For example, changing the *docfiles* options from false to true will not install or remove the already present documentation files. But a subsequent update will install or remove them.

- **paper** *paper* Sets the default papersize; possible values are *a4* and *letter*.
- **program** *paper* [*help*|*paper*] This allows setting different paper sizes for the specified *program*: *xdvi*, *dvips*, *pdftex*, *dvipdfm*, *dvipdfmx*, *context*. Without any additional argument it reports the currently selected papersize. With *help*, it issues all the supported paper sizes for that program. And if you specify a paper size, it will be set as default papersize for the given program.
- **generate** *what* This command generates one or more configuration files, as follows:

Giving *language.dat* for *what* generates the *language.dat* file which specifies the hyphenation patterns to be loaded for \LaTeX -based formats. Giving *language.def* for *what* generates the *language.def* file which specifies hyphenation patterns to be loaded for *etex*-based formats. Specifying *language* for *what* generates both of these files.

Giving *fmtutil* for *what* generates the *fmtutil.cnf* file which contains the definitions of all formats available.

Giving *updmap* for *what* generates the *updmap.cfg* file which lists all the installed font map files.

For *fmtutil* and the language files, recreating is normal and both the installer and *tlmgr* routinely call that.

For *updmap*, however, neither the installer nor *tlmgr* use **generate**, because the result would be to disable all maps which have been manually installed via *updmap-sys --enable*, e.g., for proprietary or local fonts. Only the changes in the *--localcfg* file mentioned below are incorporated by **generate**.

On the other hand, if you only use the fonts and font packages within \TeX Live, there is nothing wrong with using **generate** *updmap*. Indeed, we use it to generate the *updmap.cfg* file that is maintained in the live source repository.

If the files *language-local.dat*, *language-local.def*, *fmtutil-local.cnf*, or *updmap-local.cfg* are present under *TEXMFLOCAL* in the respective directories, their contents will be simply merged into the final files, with no error checking of any kind.

The package management actions

- **install** *pkg*... installs the packages given as argument. By default, installing a package also installs all of its dependencies. The following options are supported: **--no-depends** will not install dependent packages. There is also **--no-depends-at-all** which in addition disregards the tightly coupled packages architecture-specific executables; for example, **bin-bibtex** and **bin-bibtex.i386-linux**. That is something you should never use unless you are sure you know what you are doing. **--dry-run** fakes the installation without changing anything.

- **update** *pkg*... updates the packages given as arguments. In addition, if the *pkg* is a collection, and the remote server has new packages in this collection, they will be installed, following the dependencies specified in the collection. Options:

--list Lists the packages which would be updated or newly installed, but does not do the update. It also lists the revision numbers of the local and the remote packages.

--all Update all out-of-date packages.

--dry-run Fake the updates without changing anything.

--backupdir *directory* Save a snapshot of the current package (as installed) in *directory*, before the package is updated. This way one can easily recover in case an update turned out as not working. See the **restore** action for details.

--no-depends Do not install normal dependencies.

--no-depends-at-all See **install** above for this option.

- **remove** *pkg*... removes the packages given as arguments. Removing a collection will remove all package dependencies (but not collection dependencies) in that collection, unless **--no-depends** is specified. However, when removing a package, dependencies are never removed.

Options:

--no-depends Do not remove dependent packages.

--no-depends-at-all See **install** above for this option.

--force By default, when removing a package or collection would invalidate a dependency of another collection/scheme, the package is not be removed and an

error is given. With this option, the package will be removed unconditionally. Use with care.

--dry-run Fake the removals without actually changing anything.

- **backup** *pkg*... makes a backup of the given packages, or all packages with **--all**, to the directory specified with **--backupdir** (it must exist and be writable).

The following options are supported:

--backupdir *directory* The directory is a required argument and must specify an existing directory where backups are to be placed.

--all Make a backup of all packages in the TeX Live Live installation. This will take quite a lot of space and time.

- **restore** **--backupdir** *dir* [*pkg* [*rev*]] If no *pkg* is given (and thus no *rev*), lists the available backup revisions for all packages.

With *pkg* given but no *rev*, list all available backup revisions of *pkg*.

With both *pkg* and *rev*, tries to restore the package from the specified backup.

The option **--backupdir** *dir* is required, and must specify a directory with backups.

The option **--dry-run** is also supported, as usual.

- **arch operation arg**... If *operation* is **list**, this lists the names of architectures (**i386-linux**, ...) available at the default install location.

If *operation* is **add**, adds the executables for each of the following arguments (architecture names) to the installation.

The option **--dry-run** is also supported, as usual.

3.4 Typical usage of tlmgr

Here we present some typical usage examples of the TeX Live Manager.

Installing a new collection

Suppose that you installed **scheme-medium** and then realize that the hyphenation patterns for some language you are using haven't been installed, say, Norwegian. First you fire up **tlmgr** to search for the support:

```
$ tlmgr search --global norwegian
collection-langnorwegian - Norwegian
hyphen-norwegian -
```

and then to install this collection:

```
$ tlmgr install collection-langnorwegian
install: collection-langnorwegian
install: hyphen-norwegian
regenerating language.dat
regenerating language.def
```

and then it continues to regenerate all the format files depending on either `language.dat` or `language.def`. (If the `formats` option is changed to false in the local database, the format rebuilding will be skipped. The default is to do so, to keep them up to date without manual intervention.)

Searching for a package

You want to typeset an invitation in a special form, say in the shape of a heart. Your first try is

```
$ tlmgr search paragraph
```

but that yields no output. Maybe it's not installed? So try a global search:

```
$ tlmgr search -global paragraph
tlmgr: installation location /src/TeX/
  ↳ texlive-svn/Master
bigfoot - Footnotes for critical editions
edmargin - Multiple series of endnotes for
  ↳ critical editions
footmisc - A range of footnote options
genmpage - Generalization of LaTeX's
  ↳ minipages
hanging - Hanging paragraphs
ibycus-babel - Use the Ibycus 4 Greek font
  ↳ with Babel
insbox - A TeX macro for inserting
  ↳ pictures/boxes into paragraphs
layouts - Display various elements of a
  ↳ document's layout
lettrine - Typeset dropped capitals
lineno - Line numbers on paragraphs
lipsum - Easy access to the Lorem Ipsum
  ↳ dummy text
moresize - Allows font sizes up to 35.83pt
ncctools - A collection of general
  ↳ packages for LaTeX
paralist - Enumerate and itemize within
  ↳ paragraphs
picinpar - Insert pictures into paragraphs
plari - Typesetting stageplay scripts
seqsplit - Split long sequences of
  ↳ characters in a neutral way
shapepar - A macro to typeset paragraphs
  ↳ in specific shapes
vwcol - Variable-width multiple text
  ↳ columns
```

and here we are, `shapepar` seems to be what's needed. So let us see what it is:

```
$ tlmgr show shapepar
tlmgr: installation location /src/TeX/
  ↳ texlive-svn/Master
Package: shapepar
Category: Package
```

```
ShortDesc: A macro to typeset paragraphs in
  ↳ specific shapes.
LongDesc: \shapepar is a macro to typeset
  ↳ paragraphs in a
special ...
Installed: No
Collection:collection-latexextra
```

Ok, confirmed, now we can either install the respective collection using

```
$ tlmgr install collection-latexextra
```

which will install quite a lot of packages, or only that one single package in the hope that it does not depend on anything else:

```
$ tlmgr install shapepar
tlmgr: installation location /src/TeX/
  ↳ texlive-svn/Master
install: shapepar
running mktexlsr
...
```

These examples are about finding uninstalled packages. The default for TEX Live is a full installation, i.e., everything is installed that is available.

Updating your installation

After the initial installation you want to get the latest and greatest of everything, but first you want to see what that means:

```
$ tlmgr update --list
tlmgr: installation location /mnt/cdrom
Cannot load TeX Live database from /mnt/
  ↳ cdrom at /home/norbert/tltest/2008/
  ↳ bin/i386-linux/tlmgr line 1505, <TMP>
  ↳ line 1982.
```

Hmm, there seems to be an error, it tries to install from the DVD which you returned to your friend last week. Well, then you should switch to the network installation source; best to do it for all future sessions by saving it as default location. But what was that strange address again? Fortunately you can tell `tlmgr` to use CTAN and it will know what to do:

```
$ tlmgr option location ctan
tlmgr: setting default installation
  ↳ location to http://mirror.ctan.org/
  ↳ systems/texlive/tlnet/2008
```

Fine. Now let us see what we can upgrade:

```
$ tlmgr update --list
shapepar: local: 10400, source: 10567
bin-texlive: local: 10693, source: 10750
pdftex: local: 10622, source: 10705
texlive.infra: local: 10685, source: 10748
```

Well, some things are there, so let us update all of them at once:

```
$ tlmgr update --all
update: shapepar (10400 -> 10567) ... done
update: bin-texlive (10693 -> 10750) ... 2
    done
update: pdftex (10622 -> 10705) ... done
update: texlive.infra (10685 -> 10748) ... 2
    done
running mktexlsr ...
```

Paper size configuration

You are moving to Japan and want letter as your default paper size; nothing easier:

```
$ tlmgr paper letter
```

will switch to letter for the most important programs, and also recreate the formats.

4 The GUI for tlmgr

To make most Windows users and some Unix users happy we provide a front end for the TeX Live Manager written in Perl/Tk. It does not do the actual work, but leaves that for tlmgr. It also does not provide quite the full functionality of tlmgr, but almost all of it is there.

This program features several screens for different functionalities: installation, update, and removal of packages, removal of TeX Live as a whole, architecture support and configuration.

The GUI is started with either `tlmgr gui` or `tlmgr -gui` action where action is one of the actions given above. In the latter case it tries to open the respective screen of the GUI.

4.1 The install screen

The first window to be seen normally is the package installation screen (fig. 7).

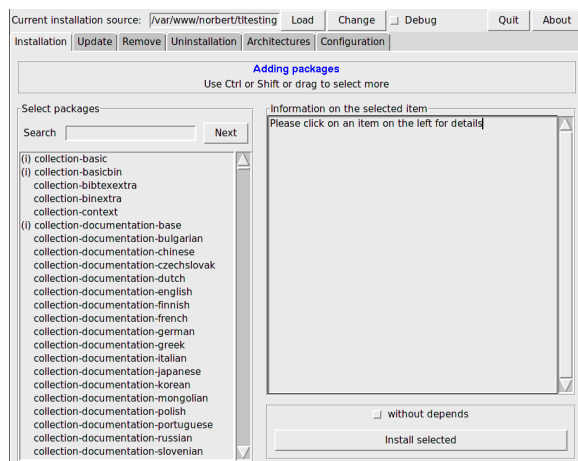


FIGURE 7: TeX Live Manager GUI install screen

At the top you see the current installation source, as given either on the command line of tlmgr, or in the absence of a command line argument as

taken from the default option. It is *not* loaded automatically, you have to press the Load button, or the Change button to select a different installation source for this run only. Below you see the list of available packages on the left, first all the collections and schemes, then all the other packages in alphabetic order. You can search by entering a string into the search text field, which immediately jumps to the first entry. The button Next jumps to the next match. After selecting one package you can see its description in the right half of this screen. Below there is the action button for installing the selected package(s), and also a switch that allows you to install a package without those it depends on.

4.2 The update screen

The update screen is similar to the install screen, but only lists those packages which have an upgrade available on the installation location. The upper part of the right pane gives you information on the package, and in the action area below you see two buttons, one for updating only the selected packages, and one for updating all packages.

In fig. 8 you can see the update screen with updates available and the information for the selected package shown in the right part of the screen.

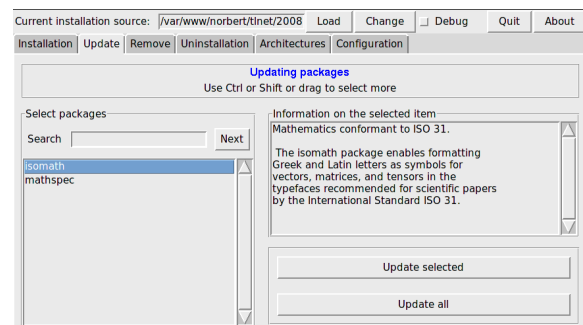


FIGURE 8: TeX Live Manager GUI update screen

4.3 The remove screen

The remove screen is also similar to the install screen, with the list of all installed packages in the left part, the information window in the upper right part, and the action area with two toggles and the remove button in the lower right part; see fig. 9.

The two toggles correspond to the option `--force` and `--no-depends` of the tlmgr remove action, see above.

4.4 The uninstallation screen

This screen only sports one button which allows you to completely remove the TeX Live installation from your system. This button is not present on Windows systems, being replaced by an informational note that you should use the Add/Remove entry from the Control Panel.

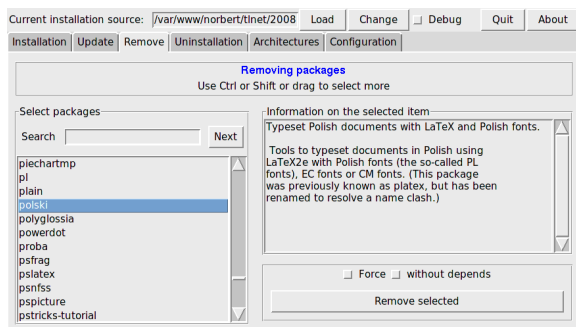


FIGURE 9: T_EX Live Manager GUI remove screen

4.5 The architectures screen

T_EX Live allows you to install the binaries for several architecture-operating system combinations in case you want to distribute your installation via NFS or other means in an inhomogen local network, see fig. 10.

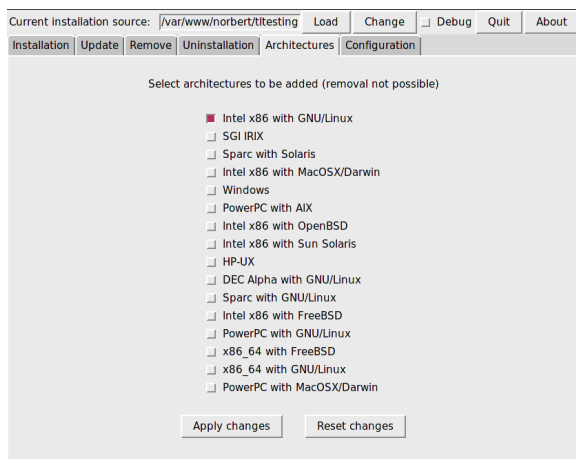


FIGURE 10: T_EX Live Manager GUI architectures screen

This screen lists the available architectures at the current installation source, and allows you to select new architectures to be installed by pressing the **Apply changes** button.

Note that the *removal* of architectures is currently not supported, and that the whole screen is disabled on Windows systems since Windows does not support normal symbolic links.

4.6 The config screen

This screen allows the user to comfortably examine and set the various options of the T_EX Live installation, see figure 11.

In the upper part you can change the defaults for the installation source, whether formats should be created (and updated) by default, and whether macro/font documentation and source files should be installed.

In the lower left part you can set the letter for all the programs to either **A4** or **letter**, or for each program individually. In the latter case you

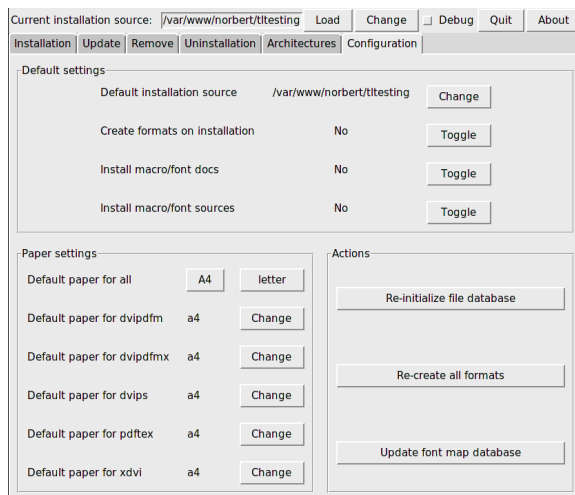


FIGURE 11: T_EX Live Manager GUI config screen

can choose from a wide range of paper formats depending on the programs support.

In the lower right part there are some convenience buttons for updating the **ls-R** databases, the outline font list (**updmap-sys**) and rebuilding all formats.

4.7 Execution of the commands

As mentioned above, this GUI is only a front end and leaves the actual work to **tlmgr** itself. So every action you do (installation, removal, etc.) will pop up a window where the output of the **tlmgr** process is shown.

On Unix systems that output will be shown immediately. Windows lacks good support for forking in Perl/Tk, and thus you have to wait until the whole process has terminated before the output appears. That can take quite some time, so please be patient.

We are working on merging the **tlmgr** and its GUI into one program so that the output would become more immediate in all cases.

5 What else is there?

Besides reworking the whole infrastructure, which is only user-visible in the new installer and the T_EX Live Manager, as with every year, all the programs and packages have been updated. We currently ship around 1400 normal packages, e.g., L^AT_EX and font packages, and around 300 other packages, mostly documentation and a few packages which are T_EX Live internal.

The new player in the game this year is the new engine LuaT_EX (<http://luatex.org>); besides a new level of flexibility in typesetting, this provides an excellent scripting language for use both inside and outside T_EX documents.

5.1 Windows-specific features

To be complete, a TEX Live installation needs support packages that are not commonly found on a Windows machine. TEX Live provides the missing pieces:

Perl and Ghostscript. Because of the importance of Perl and Ghostscript, TEX Live includes ‘hidden’ copies of these programs. TEX Live programs that need them know where to find them, but they don’t betray their presence through environment variables or registry settings. They aren’t full-scale distributions, and shouldn’t interfere with any system installations of Perl or Ghostscript.

Command-line tools. A number of Windows ports of common Unix command-line programs are installed along with the usual TEX Live binaries. These include `gzip`, `chktex`, `jpeg2ps`, `unzip`, `wget` and the command-line utilities from the `xpdf` suite. (The `xpdf` viewer itself is not available for Windows, but the Sumatra PDF viewer is based on it: <http://blog.kowalczyk.info/software/sumatrapdf>.)

fc-cache helps X_YTEX to handle fonts more efficiently.

PS_View. Also installed is PS_View, a new PostScript viewer that is free software; see fig. 12. It also supports viewing of PDF files and is extremely fast. Please contact us with any suggestions, this program is in active development.

dviout This is a DVI previewer which is shipped only in the support directory of the DVD, but you will get it if you use the network update procedure. See fig. 13 for a screenshot.

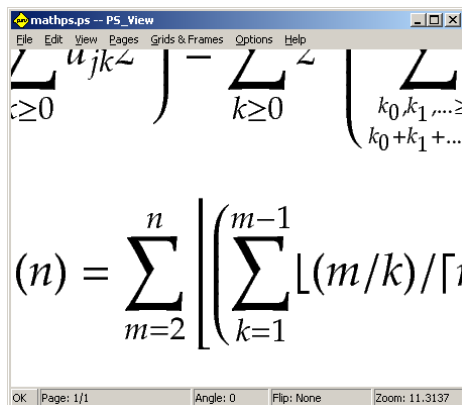


FIGURE 12: PS_View allows very high magnification, and renders PDF, too

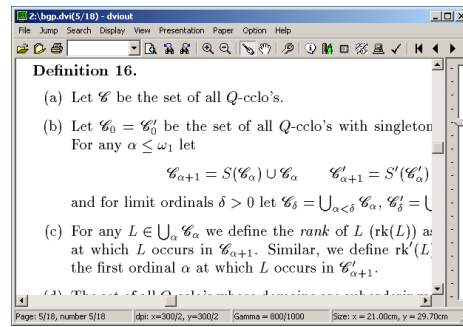


FIGURE 13: DVIout on Windows

6 Final remarks and other resources

The TEX Live Manager is very much work in progress, and its GUI even more. We are adding new functionality frequently, and improving existing functionality to make it more robust. If you see any anomalies don’t hesitate to contact us at tex-live@tug.org, and we will try to improve it further.

As with most volunteer projects the group of core programmers is quite small. Most of `tlmgr` and its GUI has been programmed by the author with some minor contributions from others. Anyone being more or less able to program Perl is heartily invited to join forces and help us, there are long lists of TODOS for the TEX Live Manager, let alone for all of TEX Live.

If you are searching more information on TEX Live your first starting place should be <http://tug.org/texlive/> and the documentation specific page <http://tug.org/texlive/doc.html>.

The list of people to thank is too long to be included here, please see the online TEX Live documentation, Chapter 9 (Acknowledgments), for the ever growing list. Of course one name has to be mentioned and that is Karl Berry who, with great enthusiasm and perpetual support (and a sometimes critical voice if I was too fast in implementing something!), prepared the TEX Live 2008 release.

References

- KOTUCHA, R., KROONENBERG, S. e PREINING, N. (2008). «A new installer for TEX live». In *Proceedings of the BachTeX 2008*. GUST.
- PREINING, N. (2007). «TEX live’s new infrastructure». *ArsTeXnica*, (4), pp. 69–73. URL <http://www.guit.sssup.it/arstexnica.php>.

▷ Norbert Preining
Vienna University of Technology
[preining at logic dot at](mailto:preining@logic.at)

L^AT_EX e grammatiche (la faccia triste dell'informatica)

Gianluca Pignalberi*, Enrico Bini†

Sommario

Le grammatiche, i diagrammi sintattici, gli automi, sono strumenti fondamentali nell'informatica; è grazie a loro se i nostri computer sono in grado di fare quello che sanno. L^AT_EX deve essere in grado di comporli. In questo articolo vedremo quali strumenti abbiamo per aggiungere questi elementi ai nostri documenti.

Abstract

Grammars, syntax diagrams and automata are Computer Science's basic tools; thanks to them our computers can do what they do. L^AT_EX has to be able to typeset them. In this paper we'll overview which tools we can use to add those elements to our documents.

1 Introduzione

Contrariamente a quanto pensano gli studenti delle scuole di grado inferiore, la grammatica non è una materia arida e noiosa (lo è, semmai, il metodo usato per insegnarla). È alla base del linguaggio, ne fornisce le regole per la costruzione. In pratica, è il fondamento della comunicazione.

In informatica le grammatiche sono il fondamento dei linguaggi di programmazione e di meccanismi quali automi (ARBIB *et al.*, 1981; KFOURY *et al.*, 1982; MOLL *et al.*, 1988). Con un intuito felicissimo, Noam Chomsky ne traccia la gerarchia e le corrispondenze con meccanismi di generazione.

L^AT_EX, in quanto strumento d'elezione della comunità scientifica, non poteva non avere a disposizione degli strumenti adatti a scrivere e disegnare grammatiche, regole di riscrittura o produzioni, automi e altro. L'articolo vuole essere un compendio dei pacchetti esistenti e una guida, non breve né esaustiva, al loro uso. Riprendo qui, espandendoli, alcuni argomenti affini da me pubblicati alcuni anni fa su Dev (PIGNALBERI, 2004a,b), rivista del defunto e resuscitando Gruppo Editoriale Infomedia; sono certo di non infrangere alcun diritto d'autore.

2 Grammatiche BNF

Una grammatica G è una quadrupla (X, V, S, P) in cui X e V sono alfabeti finiti disgiunti e

1. X è l'alfabeto terminale della grammatica;
2. V è l'alfabeto non-terminale, o variabile, della grammatica;
3. S è il simbolo iniziale della grammatica;
4. P è l'insieme di produzioni della grammatica. P è un insieme di coppie (v, w) con v stringa di $(X \cup V)$ contenente almeno un simbolo non-terminale, mentre w è un elemento arbitrario di $(X \cup V)^*$. Un elemento (v, w) di P è di solito scritto $v \rightarrow w$, ed è definito regola di riscrittura, o produzione.

Data la precedente definizione, possiamo ora scrivere una grammatica; una semplice è quella delle parentesi corrispondenti: $G = (\{ (,) \}, \{ S \}, S, \{ S \rightarrow (, S \rightarrow (S), S \rightarrow SS \})$. Siccome dal contesto è comprensibile quale simbolo appartenga a quale alfabeto, possiamo fornire la grammatica dando solo le sue produzioni.

Uno degli standard per denotare regole e produzioni grammaticali è il cosiddetto BNF: Backus-Naur Form. Gli strumenti per scrivere agevolmente dette regole sono forniti da `syntax`. Questo aggiunge a L^AT_EX anche altri comandi per altre finalità: formattazione di elementi sintattici, ridefinizione del carattere underscore (`_`), abbreviazione della sintassi del comando `\verb`. Mentre gli ultimi due comandi/caratteristiche possono essere considerate marginali per la composizione di grammatiche, non altrettanto deve essere considerata la formattazione di elementi sintattici.

All'interno di una serie di produzioni ci sono simboli terminali e non-terminali. I simboli terminali possono essere scritti tra apici oppure no. I comandi per ottenere le due rappresentazioni sono `\lit` e `\lit*`. I simboli non-terminali sono in genere rappresentati tra parentesi angolari. Il comando necessario introdotto da `syntax` è `\synt`. I comandi `\synt{S}` e `\lit{()}` generano, rispettivamente, $\langle S \rangle$ e $()$.

Ora abbiamo in potenza tutti gli strumenti per scrivere una grammatica BNF; il problema, evidenziato più volte in GREGORIO (2004), è la totale mancanza di chiarezza del codice: il sorgente di una grammatica scritta con `\synt` è molto lungo e non agevole da seguire.

2.1 L'ambiente `grammar`

Per evitare sessioni di scrittura lunghe e passibili di errori (o di orrori), `syntax` fornisce l'ambiente `grammar`, realizzato proprio per rendere facile la

*Autore

†Presentatore

scrittura di grammatiche BNF. Scrivere una grammatica qualunque è ora estremamente semplice. Vediamo la forma BNF per la grammatica relativa al linguaggio delle parentesi corrispondenti:

$$\begin{aligned} \langle S \rangle &::= () \\ &| \langle S \rangle \rangle \\ &| \langle S \rangle \langle S \rangle \end{aligned}$$

Queste produzioni dicono semplicemente: il simbolo non-terminale S può essere sostituito con una coppia di parentesi (simbolo terminale), oppure con un simbolo non-terminale S chiuso in parentesi, oppure con due simboli non-terminali S . Possiamo generare così qualunque stringa di parentesi corrispondenti, sia annidate che affiancate. Il codice della grammatica, pulitissimo, è:

```
\begin{grammar}
<S> ::= ()
\alt (<S>)
\alt <S><S>
\end{grammar}
```

Vediamo come ottenere la stringa $((()()))$ a partire dalle produzioni grammaticali. $S \rightarrow (S) \rightarrow (SS) \rightarrow ((S)) \rightarrow ((SS)) \rightarrow ((()S)) \rightarrow ((()()))$.

3 Diagrammi sintattici

Un diagramma sintattico è uno schematismo grafico teoricamente in grado di rendere semplice a chiunque la lettura di una grammatica: non tutti sono in grado di comprendere le produzioni di una grammatica BNF.

Il pacchetto `syntax` è dotato di un ambiente per formattare i diagrammi sintattici: `syntdiag`. Nella sua forma base `syntdiag` compone i diagrammi entro i limiti della pagina; nella versione *starred* (`syntdiag*`) siamo invece liberi di eccedere tali limiti, a meno di non fissare un limite, secondo la seguente sintassi:

```
\begin{syntdiag*}[dichiarazioni][ampiezza]
<testo>
\end{syntdiag*}
```

Le dichiarazioni a cui ci riferiamo riguardano il tipo di freccia sinistra e destra. Queste sono dichiarate dai comandi `\left{<tipo>}` e `\right{<tipo>}`, e i tipi ammessi sono riportati di seguito:

>>-	→→→	→→→
-><	←←←	←←←
>-	→→	→→
...	→→→	→→→
->	→→	→→
-	→	→

Vediamo finalmente cos'è un diagramma sintattico e cosa permette di rappresentare. Prendiamo sempre il caso della grammatica delle parentesi corrispondenti. Il diagramma sintattico rappresenta

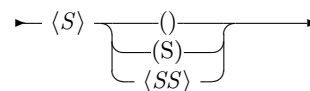
graficamente i simboli che formano la grammatica, concatenandoli tramite uno o più percorsi da seguire. Il simbolo non terminale di partenza viene rappresentato all'inizio del diagramma, mentre di seguito riportiamo le produzioni. Essendo quelle della grammatica in esame delle alternative, queste andranno rappresentate una sull'altra, con dei flussi di scelta, simulati dall'ambiente `stack`. Ogni sovrapposizione deve essere separata da `\\`. I flussi alternativi sono rivolti all'esterno, a significare che il flusso prosegue il normale percorso dopo aver scelto l'alternativa. Esiste anche la possibilità di rappresentare dei cicli, cioè dei flussi rientranti: l'ambiente necessario è `rep`.

Quando includiamo il pacchetto possiamo indicare come opzione la forma dei raccordi dei cicli e delle alternative. Scrivendo o no `rounded` otterremo delle linee arrotondate o squadrate.

Rappresentare la grammatica in esame come un diagramma sintattico è semplice, vista anche l'esiguità del sorgente:

```
\begin{syntdiag*}[] [4cm]
<S>
\begin{stack}()\\(S)\\<SS>\end{stack}
\end{syntdiag*}
```

e questo è il risultato:



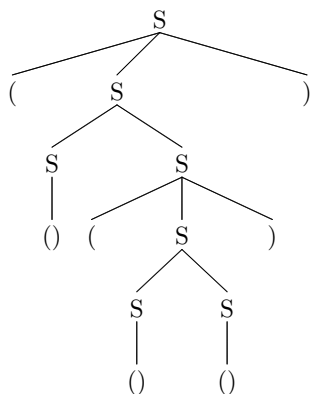
4 Alberi di derivazione

Un albero di derivazione è un formalismo grafico per visualizzare come una stringa di un linguaggio viene derivata dal simbolo non-terminale. Il pacchetto d'elezione per comporre alberi di derivazione è `synttree`. Il suo uso è molto semplice: l'unico comando `\synttree` prende come parametro la lista dei nomi dei nodi. Il primo nodo da passare è la radice dell'albero, e di seguito i figli. Ogni nodo e relativi figli devono essere racchiusi tra parentesi quadre. Possiamo anche dare, come parametro opzionale, la profondità massima dell'albero.

Come esempio, vediamo come derivare la stringa $((()()))$ dal non-terminale. Il seguente codice, indentato per comodità di comprensione,

```
\synttree[S [(]
[S [S [(] ]
[S [(]
[S [S [(] ]
[S [(] ]
]
[)]
]
[)]
]
```

genera il seguente albero di derivazione:



Ora, leggiamo i simboli terminali (le foglie dell'albero) da sinistra a destra: otteniamo la stringa $((()())())$, esattamente quella da derivare.

5 Automi

Se possiamo scrivere una grammatica, possiamo convertirla in automa. Il tipo più semplice di automa è il cosiddetto Automa a Stati Finiti (FSA – Finite State Automaton), corrispondente a un linguaggio regolare. Questo automa è del tutto simile ad un grafo con nodi (stati) e archi (transizioni). L'unico cambiamento rilevante è relativo agli stati iniziale e finale, evidenziati rispetto agli altri nodi. Possiamo fare un altro distinguo: in genere la grammatica viene considerata un generatore poiché dalle produzioni generiamo un linguaggio. Un automa viene invece considerato un accettore perché a partire da una stringa possiamo vedere se l'automa giunge allo stato finale (accetta la stringa come facente parte del linguaggio) o no.

Non sono molti i pacchetti a permettere di disegnare automi e grafi in LATEX. A dire la verità, degli unici tre trovati su CTAN uno solo è esattamente quanto sperato. Il primo pacchetto da vedere è **finomaton**, uno strumento Tcl per il disegno interattivo di automi, con successiva esportazione in METAPOST. Il secondo, **gastex**, è un vero e proprio pacchetto: fornisce comandi per comporre il grafo descritto. Purtroppo può generare solo immagini PostScript; un documento includente **gastex** dovrà essere compilato con LATEX, e non con PdfLATEX. Il terzo, dal promettente nome di **automata**, è disponibile solo per MiKTEX, ma non per TEX Live. Parleremo dei primi due.

5.1 Grafi con finomaton

Come già accennato, **finomaton** è un editor grafico, scritto in Tcl. Grazie al suo ambiente interattivo possiamo disegnare il grafo secondo il nostro gusto, per poi esportarlo in METAPOST ed includerlo nei nostri documenti. La cosa, naturalmente, comporta dei problemi: li vedremo presto.

Per prima cosa disegneremo un automa. Non sarà quello relativo alla grammatica delle parentesi

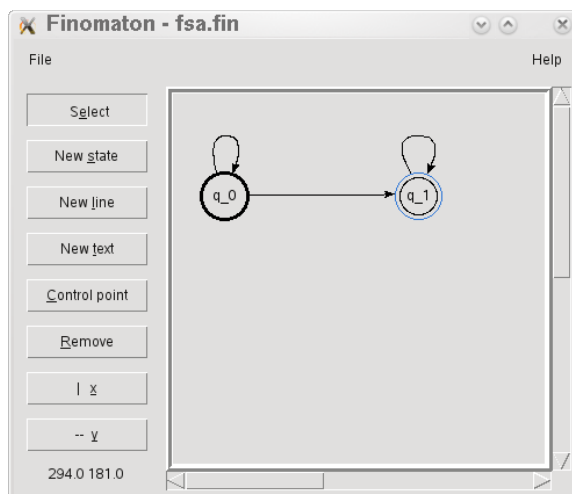


FIGURA 1: FSA del linguaggio delle parentesi corrispondenti disegnato con **finomaton**

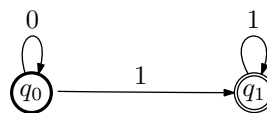


FIGURA 2: Il grafo convertito in METAPOST e poi in PDF

corrispondenti; questa è una grammatica *context-free* (libera dal contesto), riconosciuta da un *automa push-down*, ma non da un FSA. Potremmo anche disegnare un FSA in grado di riconoscere le stringhe generate dalla grammatica delle parentesi corrispondenti. Quest'automa riconoscerà anche stringhe non generate dalla grammatica, dunque non riconosce tutte e sole le stringhe di quel linguaggio, e quindi non è un accettore del linguaggio generato da quella grammatica. Allora disegneremo il grafo relativo alla seguente grammatica regolare, riconosciuta da un FSA:

$$\begin{aligned} \langle S \rangle &::= 0\langle S \rangle 1 \\ &\quad | \langle S \rangle 1 \\ &\quad | 1 \end{aligned}$$

Questa grammatica genera il linguaggio 0^*1^+ . In figura 1 vediamo il corrispondente grafo nell'editor.

Una volta finito il disegno, possiamo esportare il file in METAPOST con l'opzione dedicata nel menù File. Otteniamo un file con estensione **.mp**. Per processarlo daremo **mpost <nome_file>.mp**, ottenendo un file PostScript con estensione **.1**. Se intendiamo includere quest'ultimo in un documento da processare con PdfLATEX, dobbiamo assicurarci di aver caricato il supporto a PDF (`\input supp-pdf.tex`), oppure di aver incluso **graphics** o **graphicx**, dopodiché possiamo convertire il file (richiede ConTEXt) inserendolo nel punto in cui lo includiamo: `\convertMPtoPDF{<file>}{scala X}{scala Y}`. La soluzione più semplice però consiste nel cambiare estensione al file, da **.1** a **.mps** e fare un'inclusione normale.

Il risultato è visibile in figura 2.

5.2 Grafi con gastex

L'altro pacchetto considerato è GasTeX (Graphs and Automata Simplified in TeX). Se la sezione relativa in TeX Live appare molto deficitaria, senza neanche un manuale (per fortuna il file di stile è ben commentato), il sito internet dedicato (GASTIN, 2005) è molto ricco di esempi didattici, con un'ampia miscellanea di grafi e automi.

Disegniamo il solito automa con i comandi di *gastex*, tenendo presente questo: gli stati iniziale e finale sono qui denotati con archi rispettivamente entrante e uscente, non collegati ad altri nodi.

Attenzione: anche GasTeX non permette di processare i propri documenti con PdfLaTeX, poiché usa PostScript per disegnare ed eseguire calcoli. Il consiglio dell'autore è processare il documento con LaTeX, e poi convertirlo in PDF.

```
\unitlength=4pt
\begin{picture}(25, 10)(0,-5)
\gasset{Nw=5,Nh=5,Nmr=2.5,curvedepth=0}
\thinlines
\node[Nmarks=i,iangle=90]
(A1)(0,0){$q_0$}
\drawloop[loopangle=180](A1){$0$}
\node[Nmarks=f,fangle=90]
(A2)(25,0){$q_1$}
\drawloop[loopangle=0](A2){$1$}
\drawedge(A1,A2){$1$}
\end{picture}
```

Il precedente codice è di semplice interpretazione, e contiene quanto basta per disegnare un grafo anche più complesso. Il comando `\node` specifica se il nodo da disegnare è iniziale o finale ($Nmarks=i$ | f) e qual è la posizione angolare degli archi di inizio/fine ($i/fangle$): nel nostro caso (90°) sono nella parte superiore del nodo; qual è il nome, la posizione e l'etichetta. Il comando `\drawloop` serve a disegnare un autoarco, e specifica la posizione angolare dell'arco (nel nostro caso 0 e 180° , cioè uno alla sinistra e uno alla destra dei rispettivi nodi), il nodo a cui si applica e l'etichetta. Infine `\drawedge` disegna un arco tra due nodi, chiedendo solo i nomi dei nodi e l'etichetta dell'arco. Il comando `\gasset` si incarica di settare le dimensioni comuni: Nw e Nh sono larghezza e altezza di nodi ovali, Nmr è il raggio massimo di tali nodi (in genere $Nw/2$ o $Nh/2$); $curvedepth$ indica la curvatura di un arco, cioè la distanza verticale tra il punto di massima curvatura dell'arco e la sua proiezione sulla retta unente i due estremi dell'arco; 0 indica un arco rettilineo.

Il risultato del codice è visibile in figura 3.

6 Formattazione di sorgenti

Un programma scritto in un qualsiasi linguaggio di programmazione funziona, in parole povere, grazie

alla grammatica sottostante: questa ne assicura il funzionamento coerente. Anche evidenziare le parti del sorgente in base al significato implica evidenziarne le strutture grammaticali. Dunque possiamo vedere quei pacchetti in grado di comporre un sorgente ben formattato (qualcuno lo definirebbe abbellito) come degli strumenti concernenti la grammatica dei sorgenti da trattare.

Finora abbiamo parlato di programmi: sebbene non siano la stessa cosa, includeremo nel discorso anche gli algoritmi e lo pseudocodice.

Discuteremo i pacchetti *algorithms*, *algorithmicx*, *program* e *listings*.

6.1 Algoritmi e pseudocodice

Il primo strumento a venire in mente per formattare algoritmi è *verbatim*. Mi sembra ridondante parlarne ancora qui: l'ambiente è una delle prime cose descritte nei manuali, anche per principianti. Il suo compito è comporre il testo letteralmente, e ciò non equivale a formattare un algoritmo, visto anche l'impedimento di comporre eventuali formule in maniera adeguata. Anche usare *alltt*, o adottare altre soluzioni più casereccio (come inserire spazi arbitrari o parole bianche), non significa formattare un codice. Ricadremmo in uno dei casi non descritti, ma contemplati implicitamente, in GREGORIO (2004).

6.1.1 Algoritmi con algorithms

Il pacchetto *algorithms* ci offre due validi ambienti per formattare pseudocodice. Entrambi fungono da pacchetto, quindi sono questi da includere nel preambolo. L'ambiente *algorithmic* permette la formattazione di algoritmi e pseudocodice grazie all'adozione di comandi specifici. Questi comandi servono a riprodurre i costrutti tipici dei linguaggi imperativi: affermazioni, cicli e scelte. In tabella 1 sono riassunti i comandi disponibili per l'ambiente citato, comprendenti anche i comandi per pre e post condizioni, commenti, restituzione di valori, valori true e false e stampa di messaggi.

In fase di apertura dell'ambiente possiamo anche passare l'opzione relativa alla frequenza di numerazione delle righe del sorgente. Ad esempio, il seguente sorgente

```
\begin{algorithmic}[2]
\IF{$S \neq \emptyset$}
\STATE $S = S \setminus e$
\ELSE
\STATE exit
\ENDIF
\caption{Ricorsione su insiemi}
```

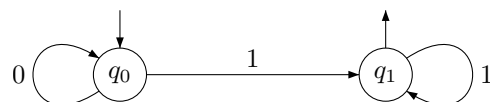


FIGURA 3: Il grafo di 0^*1^+ disegnato con GasTeX

TABELLA 1: Comandi forniti dall'ambiente `algorithmic` per comporre pseudocodici

Costrutto	Comando L ^A T _E X
Comando	<code>\STATE <testo></code>
Ciclo	<code>\FOR{<condizione>} <testo> \ENDFOR</code> <code>\FORALL{<condizione>} <testo> \ENDFOR</code> <code>\WHILE{<condizione>} <testo> \ENDWHILE</code> <code>\REPEAT <testo> \UNTIL{<condizione>}</code> <code>\LOOP <testo> \ENDLOOP</code>
Scelta	<code>\IF{<condizione>} <testo> \ENDIF</code> <code>\IF{<condizione>} <testo> \ELSE <testo> \ENDIF</code> <code>\IF{<condizione>} <testo> \ELSIF <testo> \ELSE <testo> \ENDIF</code>
Precondizione	<code>\REQUIRE <testo></code>
Postcondizione	<code>\ENSURE <testo></code>
Commento	<code>\COMMENT{<testo>}</code>
Restituzione di valori	<code>\RETURN <testo></code>
Vero	<code>\TRUE</code>
Falso	<code>\FALSE</code>
Stampa di messaggi	<code>\PRINT <testo></code>

`\label{alg:alg}`
`\end{algorithmic}`

`\end{algorithmic}`
`\end{algorithm}`

genera il seguente codice:

```

    if  $S \neq \emptyset$  then
2:    $S = S \setminus e$ 
    else
4:   exit
    end if

```

```

if  $S \neq \emptyset$  then
     $S = S \setminus e$ 
else
    exit
end if

```

Algorithm 1: Ricorsione su insiemi

L'altro ambiente citato, `algorithm`, è il floating body aggiuntivo a quello appena visto. Come tutti i floating body fornisce il noto meccanismo di numerazione; il numero è reso visibile dal comando `\caption` e dal riferimento nel testo; l'etichetta è solo in inglese ma, un minimo di manualità ci permette di cambiarne la lingua:

```

\makeatletter
\renewcommand{ALG@name}{Algoritmo}
\makeatother

```

solo dopo aver incluso il pacchetto nel documento.

Possiamo anche decidere di aggiungere una cornice all'algoritmo con le opzioni *plain*, *boxed* e *ruled* (di default) passate al caricamento del pacchetto; ne vediamo l'esempio di seguito:

```

\usepackage[boxed]{algorithm}
\begin{algorithm}
\begin{algorithmic}
\IF{$S \neq \emptyset$}
    \STATE  $S = S \setminus e$ 
\ELSE
    \STATE exit
\ENDIF
\caption{Ricorsione su insiemi}
\label{alg:alg}

```

6.1.2 Postilla agli algoritmi

Se avete notato delle minime differenze di composizione tra gli ultimi due algoritmi di esempio e il resto dell'articolo, avete senz'altro un occhio di falco, ma anche assolutamente ragione. In realtà gli algoritmi immersi nel testo sono due documenti PDF, compilati a parte a causa di un conflitto tra i pacchetti `algorithmic` e `program`, entrambi inclusi nel sorgente dell'articolo. Entrambi definiscono il comando `\COMMENT`. Quasi rimpiango la moda di anteporre al nome del comando una specie di *namespace*, secondo lo stile *nome_pacchetto.nome_comando*.

6.1.3 Algoritmi alternativi con `algorithmicx`

In alternativa (sarebbe più corretto dire "in aggiunta", come vedremo) ad `algorithms` possiamo usare `algorithmicx`. Questo pacchetto è stato scritto per sopperire ad alcune mancanze in `algorithms` riscontrate dall'autore del pacchetto in esame. Questo pacchetto non definisce alcun comando per gli algoritmi, ma fornisce delle macro per definirne; fornisce anche alcuni set di comandi da usare direttamente. Questi set (detti layout) sono: *algpseudo-code*, molto simile nell'aspetto a quello definito in `algorithmic`, *algcompatible*, pienamente compatibile col pacchetto preesistente, *algpascal*, tendente a for-

mattare, con tanto di indentazione, un programma Pascal, e *algc*, incompleto, mirante a formattare un programma C. I layout vanno caricati nel preambolo come se fossero pacchetti.

Ogni algoritmo inizia con la dichiarazione `\begin{algorithmic}[lines]` e termina con `\end{algorithmic}`. Con il parametro *lines* definiamo la numerazione del codice: 0 non numera, 1 numera tutte le righe, *n* numera le righe multiple di *n*.

Una riga di testo è preceduta dal comando `\State`. Esiste anche la macro `\Statex`; questa definisce una riga non numerabile. I commenti sono definiti grazie a `\Comment`, senza le limitazioni imposte da `algorithms`. Ad esempio,

```
\begin{algorithmic}[0]
\State x++ \Comment{Incrementa x}
\end{algorithmic}
```

genera

```
x++                                ▷ Incrementa x
```

ma nessuno ci vieta di ridefinire il commento:

```
\algnewcommand{\algorithmiccomment}[1]
{\hskip3em$\rightarrow$ #1}
```

e ottenere

```
x++          → Incrementa x
```

Non ci dilungheremo nella descrizione di tutte le caratteristiche del pacchetto: come già detto, ciò esula dai fini dell'articolo; vediamo piuttosto un esempio di algoritmo ottenuto con l'opzione *algpseudocode*: il calcolo ricorsivo del fattoriale.

```
\begin{algorithmic}[0]
\Function{fatt}{x}
\If{x < 0}
\State exit(0);
\EndIf
\If{x = 0}
\State return(1);
\Else
\State return(x * fatt(x - 1));
\EndIf
\EndFunction
\end{algorithmic}
```

genera il seguente pseudocodice già formattato:

```
function FATT(x)
  if x < 0 then
    exit(0);
  end if
  if x = 0 then
    return(1);
  else
    return(x * fatt(x - 1));
  end if
end function
```

Vediamo un riassunto di tutti i costrutti possibili con *algpseudocode* in tabella 2.

Infine, possiamo usare `algorithm` del paragrafo 6.1.1 come contenitore per floating body.

6.1.4 Pseudocodici con *program*

Un altro pacchetto promettente riguardo la formattazione di pseudocodici è *program*. L'autore afferma di aver programmato la formattazione in base al proprio stile preferito, e non necessariamente deve essere lo stile preferito di altri. Lo stile dei comandi è molto simile a quello visto nel paragrafo 6.1.1, sebbene ridotto e con dei piccoli cambiamenti.

Anche il pacchetto *program* fornisce due ambienti: *program* e *programbox*. Questi sono simili nella sostanza, ma differenti nella forma: il secondo formatta il sorgente in una sola pagina.

In tabella 3 riassumiamo i comandi dell'ambiente in esame.

Dato il seguente pseudocodice ne vediamo la formattazione ad opera di *program*:

```
\FUNCT |fatt|(x) \BODY
\IF x < 0 \THEN |exit|(0);
\FI
\IF x = 0 \THEN |return|(1);
\ELSE |return|(x * |fatt|(x - 1));
\FI
\ENDFUNCT
```

```
funct fatt(x) ≡
  if x < 0 then exit(0);
  fi
  if x = 0 then return(1);
  else return(x * fatt(x - 1));
  fi
.
```

6.2 Sorgenti formattati

Il pacchetto *listings* ci aiuta a risolvere il problema di formattare, evidenziandone le parole chiave, un sorgente scritto in un linguaggio di programmazione. La lista delle opzioni disponibili al caricamento del pacchetto è veramente vasta, ma qui vedremo solo alcune di esse.

La prima cosa da fare quando dobbiamo inserire un codice sorgente in un documento è stabilire il linguaggio del sorgente: `\lstloadlanguages{C,csh}` stabilisce l'intenzione di formattare un sorgente contenente codice C e Cshell. La lista dei linguaggi riconosciuti ne conta circa 80. Il comando visto va inserito nel preambolo del documento.

Quando dobbiamo inserire il sorgente nel documento dobbiamo avere cura di selezionare il linguaggio giusto, nonché l'eventuale dialetto: `\lstset{language=[<dialetto>]<linguaggio>}`. Con lo stesso comando possiamo stabilire la dimensione della tabulazione (`tabsize=<numero>`), lo stile delle parole chiave (ne vedremo un esempio

TABELLA 2: Costrutti disponibili con `algpseudocode`

Costrutto	Risultato
<code>\For{<testo>}</code> <corpo> <code>\EndFor</code>	for <testo> do <corpo> end for
<code>\While{<testo>}</code> <corpo> <code>\EndWhile</code>	while <testo> do <corpo> end while
<code>\Repeat</code> <corpo> <code>\Until{<testo>}</code>	repeat <corpo> until <testo>
<code>\If{<testo>}</code> <corpo> [<code>\ElsIf{<testo>}</code> <corpo> ...] [<code>\Else</code> <corpo>] <code>\EndIf</code>	if <testo> then <corpo> [else if <testo> then <corpo> ...] [else <corpo>] end if
<code>\Procedure{<nome>}{<parametri>}</code> <corpo> <code>\EndProcedure</code>	procedure <NOME>(<parametri>) <corpo> end procedure
<code>\Function{<nome>}{<parametri>}</code> <corpo> <code>\EndFunction</code>	function <NOME>(<parametri>) <corpo> end function
<code>\Loop</code> <body> <code>\EndLoop</code>	loop <body> end loop
<code>\Require <testo></code> <code>\Ensure <testo></code> <code>\Statex</code> <code>\State <istruzione></code> <code>\Call{<nome>}{<parametro>}</code> <code>\Comment <testo></code>	Require: <testo> Ensure: <testo> <istruzione> <NOME>(<parametro>) → <testo>

TABELLA 3: Riassunto dei costrutti disponibili grazie al pacchetto `program`

Costrutto	Significato e uso
<code>\FUNCT <nome> (x) \BODY \ENDFUNCT</code>	Delimitatori di funzione: questa va scritta dopo <code>\BODY</code>
<code>\PROC <nome> (x) \BODY \ENDPROC</code>	Delimitatori di procedura
<code>\IF <condizione> \THEN <azione> \ELSE <azione> \FI</code>	Delimitatori di scelta; la sezione <code>\ELSE</code> è opzionale
<code>\WHILE <condizione> \DO <istruzioni> \OD</code>	Delimitatori di ciclo

a breve), la visibilità degli spazi e delle tabulazioni (`showspace=true` o `false`, `showtabs=true` o `false`, `tab=<comando>`), la numerazione delle righe. Demandiamo la visione di questa e delle altre caratteristiche modificabili alla lettura del manuale del pacchetto, esulando queste (e anche qualcuna di quelle descritte) dalle finalità dell'articolo.

Per formattare il documento useremo l'ambiente `lstlisting`. La sintassi è

```
\begin{lstlisting}[<opzioni>]{<parametro>
  <sorgente>
\end{lstlisting}
```

Il parametro obbligatorio è un'etichetta: tutti i listati nel documento che hanno la stessa etichetta avranno lo stesso contatore di righe, e quindi manterranno la continuità della numerazione delle righe di codice.

Tra le opzioni abbiamo *caption* per le didascalie (etichetta solo in inglese; vale un'osservazione analoga fatta per *algorithm*, tesa però a modificare `\lst@UserCommand\lstlistingname{Listing}`); *captionpos* per posizionare le didascalie (valori *t o b*), *float* per rendere il listato un floating body.

È ora di vedere un esempio: il codice seguente

```
\lstset{language=[plain]TeX}
\begin{lstlisting}
\def\toclevel@title{0}
\let\@oldtoc\tableofcontents
\def\BlackBoxes{\overfullrule=5\p@}
\def\NoBlackBoxes{\overfullrule=\z@}
\def\ATname{\Ars}
\newcommand*\pkgname[1]{\textsf{#1}}
\newcommand*\clsname[1]{\textsf{#1}}
\newcommand*\optname[1]{\textsf{#1}}
\newcommand*\envname[1]{\textsf{#1}}
\DeclareRobustCommand\cmdname[1]
  {\texttt{\char'\@#1}}
\DeclareRobustCommand\meta[1]{%
  \ensuremath{\langl e}\emph{#1}
  \ensuremath{\rangl e}}
\def\@cfgfound{\ClassInfo{arstexnica}
  {Info loaded from \jobname.cfg.}}
\def\@cfgnotfound{\ClassWarning
  {arstexnica}
  {Config file not found.}}

\def\@usenempty#1{%
  \csname #1\endcsname
  \global\expandafter\let\csname
    #1\endcsname\@empty
}
\end{lstlisting}
```

fornisce questo output:

```
\def\toclevel@title{0}
\let\@oldtoc\tableofcontents
\def\BlackBoxes{\overfullrule=5\p@}
```

```
\def\NoBlackBoxes{\overfullrule=\z@}
\def\ATname{\Ars}
\newcommand*\pkgname[1]{\textsf{#1}}
\newcommand*\clsname[1]{\textsf{#1}}
\newcommand*\optname[1]{\textsf{#1}}
\newcommand*\envname[1]{\textsf{#1}}
\DeclareRobustCommand\cmdname[1]
  {\texttt{\char'\@#1}}
\DeclareRobustCommand\meta[1]{%
  \ensuremath{\langl e}\emph{#1}
  \ensuremath{\rangl e}}
\def\@cfgfound{\ClassInfo{arstexnica}
  {Info loaded from \jobname.cfg.}}
\def\@cfgnotfound{\ClassWarning
  {arstexnica}
  {Config file not found.}}

\def\@usenempty#1{%
  \csname #1\endcsname
  \global\expandafter\let\csname
    #1\endcsname\@empty
}
```

Supponiamo di voler cambiare aspetto della sola parola chiave *def*. Dovremo sempre passare uno o più parametri a `\lstset`. Nel nostro caso, volendo la parola chiave colorata, basterà specificare, prima del sorgente, lo stile da applicare, ad esempio:

```
\lstset{emph={def},
  emphstyle=\color{blue}}
```

Ecco lo stesso sorgente secondo il nuovo stile:

```
\def\toclevel@title{0}
\let\@oldtoc\tableofcontents
\def\BlackBoxes{\overfullrule=5\p@}
\def\NoBlackBoxes{\overfullrule=\z@}
\def\ATname{\Ars}
\newcommand*\pkgname[1]{\textsf{#1}}
\newcommand*\clsname[1]{\textsf{#1}}
\newcommand*\optname[1]{\textsf{#1}}
\newcommand*\envname[1]{\textsf{#1}}
\DeclareRobustCommand\cmdname[1]
  {\texttt{\char'\@#1}}
\DeclareRobustCommand\meta[1]{%
  \ensuremath{\langl e}\emph{#1}
  \ensuremath{\rangl e}}
\def\@cfgfound{\ClassInfo{arstexnica}
  {Info loaded from \jobname.cfg.}}
\def\@cfgnotfound{\ClassWarning
  {arstexnica}
  {Config file not found.}}

\def\@usenempty#1{%
  \csname #1\endcsname
  \global\expandafter\let\csname
    #1\endcsname\@empty
}
```

Il pacchetto ci permette altre due utili funzioni: la prima, `\lstinline`, rende possibile l'inserimento

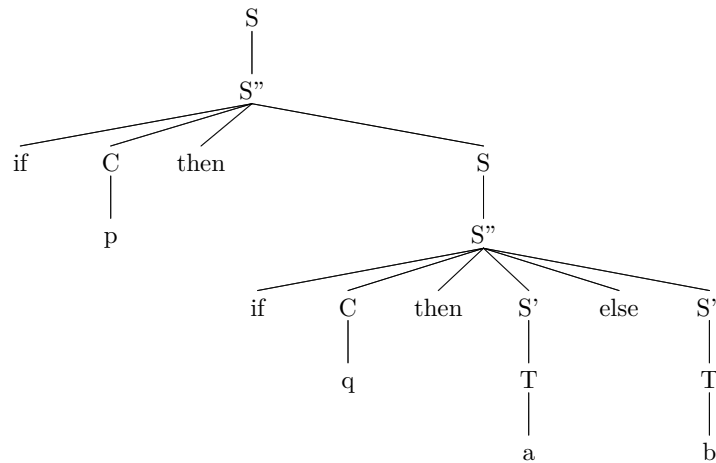


FIGURA 4: Albero di derivazione per if-then-else

inline di un comando, allo stesso modo di `\verb`; la seconda, `\lstinputlisting`, ci permette di inserire dei sorgenti nel documento senza copiarceli fisicamente.

6.3 Supplemento grammaticale

All'inizio del capitolo abbiamo fatto cadere dall'alto la similitudine "sorgente evidenziato – grammatica evidenziata". La cosa potrebbe sembrare strana a qualche persona; per ovviare a ciò proponiamo un esempio classico: if-then-else.

Si abbia la seguente grammatica:

$$\begin{aligned}
 \langle S \rangle &::= \langle S' \rangle \\
 &\quad | \quad \langle S'' \rangle \\
 \langle S' \rangle &::= \text{if } \langle C \rangle \text{ then } \langle S' \rangle \text{ else } \langle S'' \rangle \\
 &\quad | \quad T \\
 \langle S'' \rangle &::= \text{if } \langle C \rangle \text{ then } \langle S \rangle \\
 &\quad | \quad \text{if } \langle C \rangle \text{ then } \langle S' \rangle \text{ else } \langle S'' \rangle \\
 &\quad | \quad T \\
 \langle C \rangle &::= p \\
 &\quad | \quad q \\
 \langle T \rangle &::= a \\
 &\quad | \quad b
 \end{aligned}$$

con S , S' , S'' e C simboli non-terminali (rispettivamente affermazione/Statement e condizione/Condition), a e b comandi terminali, p e q condizioni terminali, e *if*, *then* e *else* parole riservate terminali.

Questa grammatica genera una sola possibile derivazione della stringa `if p then if q then a else b`, come vediamo in figura 4.

Avere grammatiche non ambigue, cioè in grado di generare un solo albero di derivazione per istruzione, è essenziale per avere specificazioni sintattiche ben definite; al di là di questo fatto, pure importante nella teoria dei linguaggi di programmazione, possiamo rimarcare quanto segue: riuscire

ad evidenziare un sorgente equivale ad effettuarne il parsing, e dunque a riconoscere il sorgente come scritto nel linguaggio scelto. Dunque la composizione di sorgenti non è fuori tema rispetto all'argomento trattato.

7 Conclusioni

Nel corso dell'articolo abbiamo introdotto alcuni argomenti ostici anche per molti informatici, ma fondamentali per inquadrare la portata dell'articolo stesso. Per ognuno degli argomenti (grammatiche, diagrammi sintattici, alberi di derivazione, automi a stati finiti, formattazione di algoritmi e sorgenti di programmi) abbiamo descritto almeno un pacchetto in grado di aggiungere a *L^AT_EX* la capacità di comporre gli elementi citati.

Semmai ce ne fosse bisogno, sapremmo da dove partire per i nostri futuri documenti su grammatiche e linguaggi formali.

8 Ringraziamenti

Un doveroso ringraziamento al Comitato Scientifico del *q_UIt* e ai revisori: con i loro suggerimenti e osservazioni hanno notevolmente migliorato la qualità dell'articolo.

Un grazie sentito a Enrico Bini, senza il cui aiuto non sarei stato in grado di presentare il lavoro al meeting.

Oltre a dover ringraziare tutti gli autori dei pacchetti citati, ringrazio e mi scuso con Pallavicini e Conte (e Jannacci), inconsapevoli e incolpevoli ispiratori del titolo.

Riferimenti bibliografici

ARBIB, M., KFOURY, A. e MOLL, R. (1981). *A Basis for Theoretical Computer Science*. Springer, New York, N.Y.

- GASTIN, P. (2005). «GasT_EX». URL `http://www.lsv.ens-cachan.fr/~gastin/gastex/gastex.html`.
 - GREGORIO, E. (2004). «Errori in L^AT_EX: come maltrattare L^AT_EX e rendere infelice un *copy editor*». In *Atti del GJrmeeting*. Pisa.
 - KFOURY, A., MOLL, R. e ARBIB, M. (1982). *A Programming Approach to Computability*. Springer, New York, N.Y.
 - MOLL, R., ARBIB, M. e KFOURY, A. (1988). *An Introduction to Formal Language Theory*. Springer, New York, N.Y.
 - PIGNALBERI, G. (2004a). «L^AT_EX, diagrammi sintattici e grammatiche». *Dev*, (122).
 - (2004b). «L^AT_EX e la formattazione di pseudocodice e sorgenti». *Dev*, (123).
- ▷ Gianluca Pignalberi
Free Software Magazine
g dot pignalberi at
freesoftwaremagazine dot com

▷ Enrico Bini
Scuola Superiore Sant’Anna
Pisa, Italia
e dot bini at sssup dot it

longmedal: un pacchetto per medaglioni divisi su più pagine

Agostino De Marco, Massimiliano Dominici

Sommario

Alcuni manuali usano riquadri incorniciati, eventualmente estesi su più pagine, per separare dal testo principale materiale a carattere avanzato o di altro tipo. Il pacchetto `longmedal` si propone di fornire un metodo semplice per riprodurre questo tipo di oggetti in un documento prodotto con `LATEX`.

Abstract

Some textbooks organize different kinds of advanced or secondary material in framed boxes, possibly spanning several pages. The `longmedal` package aims at providing an easy interface to reproduce such objects in a `LATEX` document.

1 Introduzione

In manuali di argomento tecnico, ma non solo, accade spesso che una parte del testo, considerata non appartenente al testo principale, venga separata da questo anche visivamente, oltre che strutturalmente. Si può trattare di approfondimenti, di note storiche, di materiale a carattere esercitativo. Questo materiale viene di solito evidenziato da un riquadro munito di un'intestazione che ne identifica il contenuto, e può eventualmente estendersi per più di una pagina. Inoltre, in alcuni casi il materiale contenuto nel riquadro è separato anche dal flusso normale del testo e confinato in una precisa collocazione della pagina (di norma nella parte superiore). La figura 1, tratta da ANDERSON (2004), illustra appunto uno di questi riquadri.

La gestione di un layout di pagina come questo può risultare complicata e macchinosa con programmi di impaginazione WYSIWYG, anche con quelli progettati per un uso professionale; e la modifica di poche righe di testo può comportare la perdita di ore di lavoro, in particolare se a seguito di essa cambia la lunghezza del testo stesso. Con `LATEX` questo processo può invece essere, entro certi limiti, automatizzato, rendendo più semplice il compito di gestire un layout del genere. Sottolineare questo aspetto, che è uno dei punti di forza di `LATEX`, è ancora più importante in un momento come quello attuale in cui assistiamo ad una fase di transizione nel mondo di `TEX`. Il definitivo perfezionamento di `XYLATEX` porterà, infatti, alla possibilità di una gestione senza precedenti dei font di sistema, un settore in cui fino ad ora era innegabile il van-

taggio acquisito dalle applicazioni commerciali. A questo va aggiunto l'auspicato avvento di `LUATEX`. Se questa applicazione verrà resa facilmente fruibile in tempi brevi, gli utilizzatori di `LATEX` avranno a disposizione uno strumento di lavoro straordinariamente più potente dell'attuale, sotto tutti gli aspetti, non ultimo quello della programmabilità.

Il pacchetto `longmedal` si propone di fornire una soluzione di facile uso al problema di riprodurre il tipo di oggetti in questione, utilizzando a questo scopo il meccanismo di gestione degli oggetti mobili (*float*) implementato in `LATEX`. Questa soluzione comporta alcune limitazioni, quali l'impossibilità di avere all'interno del riquadro oggetti mobili o note a piè di pagina. Queste limitazioni, tuttavia, non sembrano irragionevoli e rappresentano un male minore rispetto ai problemi che sarebbero sorti se si fosse voluto percorrere la via di una ridefinizione della routine di output. Questa alternativa, oltre a rappresentare tecnicamente un compito di rilevante difficoltà, avrebbe potenzialmente introdotto incompatibilità con altri pacchetti che, in maniera simile, vanno a ritoccare la medesima routine.

Lo strumento più adatto per la creazione e gestione di tali *float* è risultato essere il pacchetto `float`, che insieme a pacchetti come `calc`, `ifthen`, `chngpage`, `multicol`, `xkeyval` e `xcolor` è servito a facilitare notevolmente i compiti di programmazione degli autori.¹

Le “radici” del pacchetto `longmedal` si possono trovare in due discussioni di fine 2007 all'interno del Forum del GuIT: <http://www.guit.sssup.it/phpBB2/viewtopic.php?t=3186&highlight=e> e <http://www.guit.sssup.it/phpBB2/viewtopic.php?t=3316&highlight=>. Il Prof. Claudio Beccari ha fornito, poi, un preziosissimo apporto nelle fasi iniziali del progetto, con i suoi suggerimenti e con la revisione del codice, oltre ad aver ispirato il nome del pacchetto (il termine “medaglione” appare infatti in BECCARI (2007, pp. 206-207)).

2 L'algoritmo principale

L'idea fondamentale alla base dell'implementazione del pacchetto è, come si è detto, quella di sfruttare il meccanismo di posizionamento dei *float*, così co-

1. Per maggiori dettagli sui singoli pacchetti si rimanda alla relativa documentazione: ADRIAENS (2008); CARLISLE (2001); KERN (2007); LINGNAU (2001); MITTELBAH (2006); THORUP *et al.* (2005); WILSON (2003).

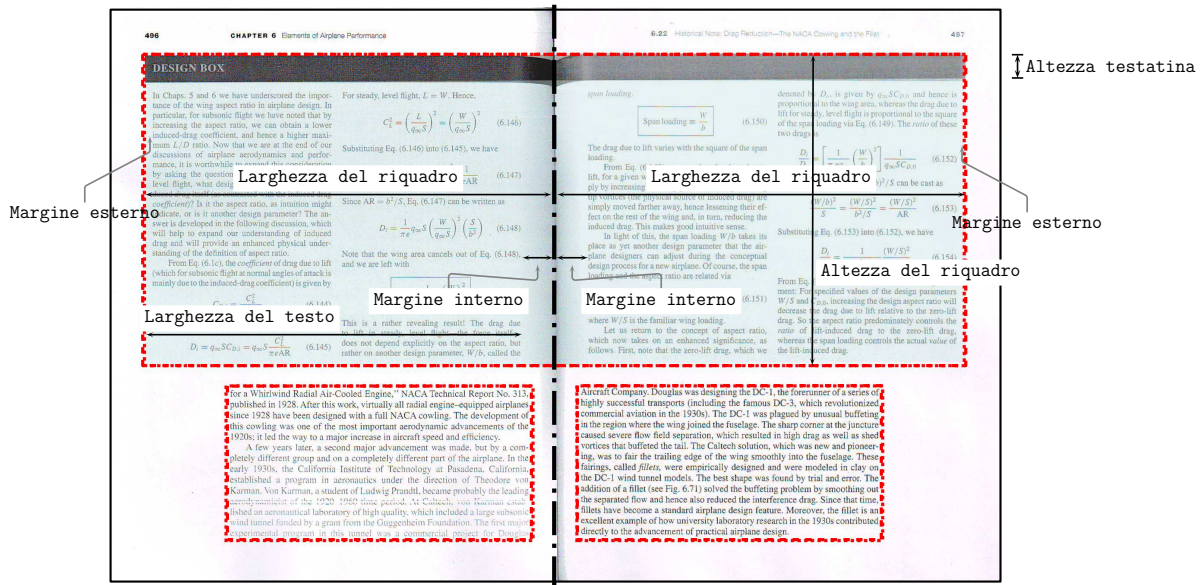


FIGURA 1: Illustrazione di un riquadro di approfondimento.

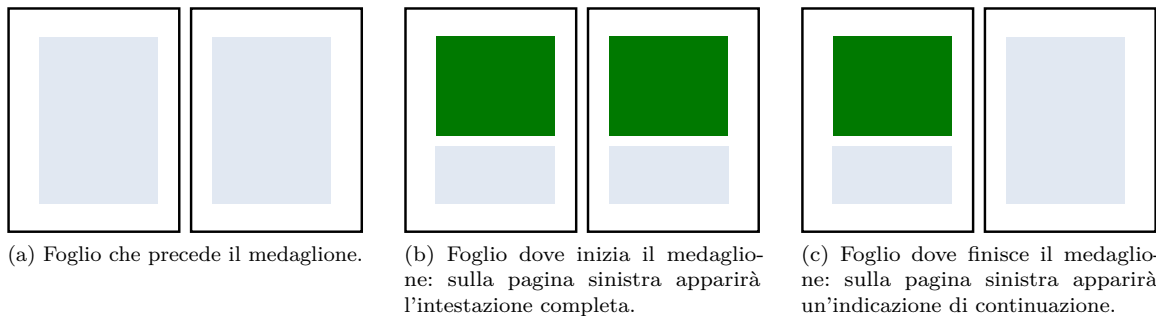


FIGURA 2: Schema di posizionamento delle singole parti di un medaglione.

me è definito in `LATEX`. Il contenuto del medaglione viene suddiviso in un numero sufficiente di `float` dotati di adeguata intestazione e posizionato in pagine adiacenti.

Il comando che l'utente usa nel proprio documento per avere un medaglione, (`\longmedal`), non agisce direttamente sul materiale in questione, ma si limita semplicemente a inserirlo in un box (`\LMD@box`), dopo aver raccolto eventuali indicazioni dall'utente sul valore dei parametri che controllano l'aspetto del medaglione ed aver eseguito una serie di calcoli sulle varie dimensioni che lo caratterizzano. In particolare vengono determinate l'altezza e la larghezza dell'area del testo vera e propria del riquadro, e queste due dimensioni vengono immagazzinate nei registri `\LMD@height` e `\LMD@widthc`. Inoltre viene incrementato il contatore interno che regola il numero progressivo del riquadro.²

Il lavoro vero e proprio viene poi eseguito dalla macro interna `\LMD@makeboxes`. Essa prende, come unico argomento, il nome dell'ambiente mobile

2. Questa operazione va compiuta adesso, e non quando il materiale viene diviso nei singoli `float`.

in cui verrà inserito l'argomento di `\longmedal`.³

All'interno di `\LMD@makeboxes` viene eseguito un ciclo che termina soltanto quando `\LMD@box` risulta vuoto. Ad ogni iterazione del ciclo, la primitiva `\vsplit` estrae da `\LMD@box` tanto materiale quanto basta per riempire un box di dimensioni `\LMD@height × \LMD@widthc`. A questo materiale vengono aggiunti eventuali sfondi e cornici, dopo di che può essere inserito in un ambiente mobile.

Alla fine del ciclo avremo quindi ottenuto una serie di ambienti mobili, tutti dello stesso tipo, e tutti con lo stesso posizionamento (di norma nella parte superiore della pagina), pronti per essere accodati ed emessi uno per pagina.⁴ Se il documento che si sta scrivendo è un documento a doppia facciata, al momento dell'effettivo posizionamento dei singoli `float`, viene controllata la parità della pagina su

3. Come vedremo in seguito nella sezione 4.5, questo è utile per definire nuovi tipi di riquadro, ognuno corredato del proprio, esclusivo, ambiente mobile.

4. `LATEX` mantiene code diverse per diversi tipi di oggetti mobili, per cui non c'è pericolo di interferenza, tuttavia tabelle e figure posizionate in alto possono essere bloccate da questa coda e spostate più o meno lontano dal punto di richiamo.

cui ci si trova⁵ e viene di conseguenza generata l'intestazione più pertinente: sul primo *float* apparirà comunque l'intestazione completa; a partire dal secondo, se questo si trova su una pagina pari apparirà l'indicazione di continuazione, altrimenti l'intestazione sarà vuota (la figura 2 illustra la situazione indicata). Se invece si sta scrivendo un documento a facciata singola, apparirà l'intestazione completa sul primo *float* e l'indicazione di continuazione su tutti gli altri.

Inoltre ad ogni primo *float* emesso verranno anche scritte su un file ausiliario tutte le informazioni necessarie per generare un elenco dei medaglioni.

2.1 Accorgimenti per hyperref

Alcuni accorgimenti devono essere presi affinché il pacchetto possa collaborare bene con *hyperref*. In particolare alcuni problemi sorgono in quanto *hyperref* cerca di inserire un'ancora per i suoi riferimenti, ogni volta che viene emesso un nuovo *float*, contando sul fatto che il contatore sia di volta in volta incrementato. Questo ovviamente nel nostro caso non succede, perché singoli *float* appartengono allo stesso medaglione e di conseguenza devono condividere lo stesso numero progressivo. In questo modo verrebbero definite in punti diversi ancora con lo stesso nome, confondendo il meccanismo dei riferimenti, e *hyperref* è costretto a segnalare l'errore.

Inoltre un'ulteriore ancora viene definita nel momento in cui viene incrementato il contatore `longmedalFloat`. Poiché a noi serve in realtà soltanto questa, è sufficiente inibire il comportamento di *hyperref* nei confronti dei nuovi *float* generati. Il codice che segue controlla che *hyperref* sia stato caricato e, in caso affermativo, fa in modo che esso non posizioni ancora all'inizio di ogni *float*.

```
\AtBeginDocument{%
  \ifpackageloaded{hyperref}{%
    \def\LMD@nomultipleanchors{%
      \let\float@makebox\HyOrg@float@makebox%
    }
    \let\LMD@nomultipleanchors\relax
  }
}
```

2.2 Le cornici

Le cornici vengono ottenute grazie ad una serie di comandi modificati del pacchetto *framed*. In particolare, le modifiche apportate consentono di assemblare insieme i diversi *box* che compongono il medaglione (intestazione e area del testo) e di inserire un colore di sfondo dietro l'area del testo.

5. A questo scopo viene, per il momento, impiegato il pacchetto *chnpage*. Poiché il suo stesso autore lo considera ormai obsoleto, esso verrà sostituito, nelle prossime versioni, con il suo successore *change page*.

3 Comandi per l'utente

3.1 Il comando principale

Il comando principale messo a disposizione dal pacchetto è `\longmedal`. Il comando accetta un argomento opzionale e uno obbligatorio:

```
\longmedal[⟨opzioni⟩]{⟨testo⟩}
```

L'argomento obbligatorio (*⟨testo⟩*) contiene il materiale che dovrà essere posizionato nel medaglione, mentre l'argomento opzionale contiene le istruzioni per modificare l'aspetto di default del medaglione stesso. Esamineremo in seguito (sezione 4) i dettagli relativi a queste ultime.

3.2 Elenco dei medaglioni

È possibile anche generare automaticamente un elenco dei singoli medaglioni, così come può essere fatto per figure e tabelle, e, più in generale, per tutti i *float*. Tuttavia il pacchetto non mette a disposizione, almeno per il momento, un comando apposito, ma si appoggia a quanto fornito da *float*, cioè:

```
\listof{longmedalFloat}{lom}
```

Il comando `\listof{⟨nome⟩}{⟨ext⟩}` è stato infatti progettato per trattare in maniera generale ogni tipo di elenco, senza che si debba definire ogni volta un comando apposito. Si noti però che nel primo argomento di `\listof` è stato inserito non `longmedal`, ma `longmedalFloat` che è il nome del tipo di *float* per medaglioni. Questo non è del tutto soddisfacente e potrebbe essere cambiato in futuro.⁶

4 Personalizzazione dell'ambiente

Alcuni parametri relativi all'aspetto dei medaglioni possono essere modificati dall'utente secondo le proprie preferenze o le esigenze specifiche del progetto. Si è deciso di lasciare all'utente la possibilità di configurare il valore di questi parametri, che esamineremo con maggior dettaglio in seguito, sia ad un livello globale, valido per tutti i medaglioni di uno stesso tipo,⁷ sia a livello del singolo medaglione.

In un'ottica di questo tipo, e considerando l'elevato numero di parametri modificabili dall'utente, si è ritenuto che la soluzione più semplice fosse quella di associare a ciascun parametro una variabile contrassegnata da una parola chiave. Esistono diversi pacchetti che permettono di utilizzare una struttura di questo tipo. Quello fino ad oggi più

6. Tuttavia il meccanismo più generale di `\listof` continuerà comunque a funzionare, cosicché non si avranno problemi di compatibilità.

7. Vedremo in seguito (sezione 4.5) che è possibile definire comandi per nuovi tipi di medaglione, nel caso il documento ne preveda più di uno.

TABELLA 1: Opzioni per il layout generale.

Variabile	Tipo	Valore di default
<code>height</code>	dim.	<code>0.6\textheight</code>
<code>width</code>	dim.	<code>\textwidth</code>
<code>innermargin</code>	dim.	<code>0.5cm</code>
<code>outermargin</code>	dim.	<code>0.5cm</code>
<code>framesep</code>	dim.	<code>0pt</code>
<code>framerule</code>	dim.	<code>2pt</code>
<code>frameheadrule</code>	dim.	<code>2em</code>
<code>framecolor</code>	col.	<code>black</code>
<code>framebgcolor</code>	col.	<code>gray</code>
<code>columns</code>	int.	<code>1</code>
<code>drawframe</code>	alt.	<code>none</code>

usato, e contenuto in tutte le principali distribuzioni anche nella versione di base, è `keyval`. Tuttavia, con `keyval` è difficile implementare un meccanismo a cascata che permetta di ereditare modifiche precedenti, perché a quelle variabili a cui l'utente non assegna un valore specifico, viene assegnato, se è stato definito, un valore di default iniziale, che non può essere cambiato, sovrascrivendo le precedenti modifiche.

Una soluzione, ovviamente, può essere quella di specificare nuovamente, assieme alla variabile da modificare, anche tutte le altre. Ma questa soluzione, oltre ad essere estremamente scomoda, è anche soggetta ad errori, perché, in caso di un successivo ripensamento su quelli che sono i parametri ottimali, costringe l'autore ad intervenire sulla stessa variabile in diversi luoghi del documento.

Fortunatamente, il pacchetto `xkeyval` ha introdotto un sistema per memorizzare le assegnazioni. In questo modo è possibile assegnare alle variabili valori che sono ereditati nel momento in cui vengono effettuate nuove assegnazioni.

Questa possibilità di poter gestire in maniera semplice differenti livelli di configurazione è stato il motivo determinante nella scelta di `xkeyval` nei confronti di `keyval`. Del resto, con l'ultima versione 2008 anche T_EX Live, dopo M_iKT_EX, si è dotata di un sistema per la gestione dei pacchetti, per cui diventano sempre meno rilevanti le obiezioni all'adozione di pacchetti meno diffusi.

Terminata questa introduzione alla gestione delle variabili, possiamo esaminarle nei dettagli, dopo averle distinte in tre diverse categorie:

- variabili che controllano il layout generale;
- variabili che controllano l'aspetto dell'intestazione;
- altre.

4.1 Layout generale

Le variabili che rientrano in questa categoria sono elencate nella tabella 1. Ne fanno parte, in primo luogo, i parametri che specificano le dimensioni

TABELLA 2: Opzioni per l'intestazione.

Variabile	Tipo	Valore di default
<code>headernamelfont</code>	font	<code>\bfseries</code>
<code>headernumfont</code>	font	<code>\bfseries</code>
<code>headertitlefont</code>	font	<code>\bfseries</code>
<code>headercontfont</code>	font	<code>\bfseries\footnotesize</code>
<code>headerconttext</code>	testo	(continua da pagina precedente)
<code>headername</code>	testo	<code>Insight Box</code>
<code>headernum</code>	testo	<code>\thelongmedalFloat</code>
<code>headertitle</code>	testo	
<code>label</code>	testo	
<code>caption</code>	testo	<code>\LMDtitle</code>
<code>headersep</code>	testo	<code>{\ }</code>
<code>headertitlesep</code>	testo	<code>{\ }</code>
<code>headeralignment</code>	macro	
<code>headertextcolor</code>	col.	<code>white</code>
<code>preheaderskip</code>	dim.	<code>1em</code>

dei vari elementi che compongono il medaglione. Queste dimensioni, tranne quelle che si riferiscono allo spessore della cornice, sono indicate nella figura 1. Va specificato che le variabili `height` e `width` indicano, rispettivamente, l'altezza e la larghezza *totali* del riquadro. Altezza e larghezza dell'area del testo vengono calcolate a partire da queste, sottraendo i margini e altri *spazi bianchi*, come `framesep` che indica lo spazio che separa la cornice dall'area interna del riquadro (area del testo + margini). Le dimensioni possono essere passate anche sotto forma di espressioni, dato che `longmedal` carica comunque il pacchetto `calc` per eseguire internamente i propri calcoli. La seguente assegnazione, quindi, è del tutto legittima, all'interno dell'argomento opzionale di `\longmedal`:

```
width=\textwidth+2cm+2\fbboxsep
```

Le variabili con prefisso `frame` controllano invece elementi visuali come lo spessore della cornice, il suo colore, o il colore di sfondo. Può essere specificata, come colore, ogni espressione valida per `xcolor` (che viene caricato automaticamente da `longmedal`).

Infine `columns` specifica il numero di colonne in cui va divisa l'area del testo (il pacchetto `multicol` viene caricato appositamente a questo scopo), mentre `drawframe` controlla quale parte della cornice vada tracciata, oltre a quella superiore che riporta l'intestazione. Le alternative possibili sono quattro: `none` (opzione di default, verrà tracciata solo la parte superiore), `bottom` (solo la parte inferiore), `side` (solo i lati), `all` (tutta la cornice verrà tracciata).

4.2 Intestazioni

Nella tabella 2 sono indicate le variabili relative alla composizione dell'intestazione dei vari riquadri, insieme ai loro valori di default.

TABELLA 3: Altre opzioni.

Variabile	Tipo	Valore di default
defaults	bool.	false

L'intestazione è composta dal nome dell'ambiente, da un numero progressivo e da un eventuale titolo, specifico del singolo medaglione, che l'utente può scegliere di inserire oppure no. Questi dati controllano indirettamente anche la voce nell'elenco dei medaglioni. In particolare, la variabile `caption` indica ciò che va aggiunto a numero e nome dell'ambiente (di default, l'eventuale titolo).

4.3 Altre opzioni

L'ultima opzione serve a ripristinare i valori di default impostati dal pacchetto (o al momento della definizione di un nuovo tipo di medaglioni, si veda la sezione 4.5). Si tratta, naturalmente di una variabile che può assumere solo i valori *vero* o *falso* (default).

4.4 Impostazioni globali

Per quanto sia conveniente avere la possibilità di scegliere i valori dei parametri per ogni singolo medaglione, è ancora più conveniente poter disporre di un meccanismo per impostare questi parametri globalmente per tutti i medaglioni appartenenti a un determinato tipo. Il pacchetto definisce a questo scopo il comando `\LMDSetup` con la seguente sintassi:

`\LMDSetup{⟨comando⟩}{⟨opzioni⟩}`

dove `⟨comando⟩` è il nome del tipo di medaglione a cui si vuole applicare la configurazione e `⟨opzioni⟩` è la solita lista di assegnazioni. Come è stato già spiegato in precedenza (sezione 4), `\LMDSetup` usa internamente il comando `\presetkeys` di `xkeyval` per memorizzare i valori assegnati, in modo che un successivo uso di `\setkeys` nell'argomento opzionale, ad esempio, di `\longmedal` non faccia sì che alle variabili non specificate venga associato il valore di default.

4.5 Nuovi tipi di medaglione

`\longmedal` è abbastanza configurabile da poter rispondere a esigenze di tipo diverso. Tuttavia è possibile che il disegno di un documento richieda la presenza contemporanea di diversi tipi di medaglioni, ad esempio degli *Approfondimenti tecnici* e delle *Note storiche*. Questi diversi medaglioni devono essere tenuti rigorosamente separati, quanto ad aspetto, numerazione, ecc. senza tener conto del fatto che di norma avranno un nome differente.

È utile, quindi avere un comando per definire nuovi ambienti:

`\DeclareNewMedalFloat{%`
`⟨comando⟩}{⟨nome⟩}{⟨pos⟩}{⟨ext⟩}`

I quattro argomenti indicano nell'ordine:

`⟨comando⟩` il nome del comando che verrà poi usato dall'utente nel documento (deve essere specificato *senza* la barra rovescia);

`⟨nome⟩` il nome dell'ambiente, come deve apparire nell'intestazione;

`⟨pos⟩` la posizione del medaglione (una di quelle possibili per un ambiente mobile, ovviamente `t` è la più adatta);

`⟨ext⟩` l'estensione del file ausiliario per generare l'elenco (si veda la documentazione del pacchetto `float`).

Di fatto, anche il comando `\longmedal` viene definito dal pacchetto proprio usando `\DeclareNewMedalFloat`:

`\DeclareNewMedalFloat{%`
`longmedal}{Insight Box}{t}{lom}`

Per avere un'idea di come opera effettivamente `\DeclareNewMedalFloat` riportiamo il codice relativo:

```
\newcommand{\DeclareNewMedalFloat}[4]{%
  \@namedef{LMD@#1@name}{#2}
  \LMD@newfloat{#1Float}{#3}{#4}
  \LMD@definekeys{#1}
  \LMD@keydefaults{#1}
  \LMD@makecommand{#1}
  \@namedef{theH#1Float}{%
    \@nameuse{the#1Float}}
}
```

Come si può vedere dal codice, i nomi dei vari ambienti e comandi vengono costruiti a partire dal nome del comando (primo argomento) suggerito dall'utente. `\LMD@newfloat` combina insieme le funzioni di `\newfloat` e `\floatname`, definiti dal pacchetto `float`, dichiarando, così, un nuovo ambiente mobile e assegnandogli, allo stesso tempo, il nome che deve apparire nell'intestazione. I tre comandi che seguono associano al nuovo tipo di medaglione le variabili che ne controllano l'aspetto, assegnano i valori di default e definiscono il comando che sarà usato per generare i singoli medaglioni. Infine viene creato un doppione del contatore che regola la numerazione dei successivi medaglioni, doppione che verrà usato da `hyperref`, se caricato.

All'atto di definire un nuovo tipo di medaglione sarà bene accertarsi di aver scelto come estensione del file ausiliario (argomento `⟨ext⟩`) un suffisso che non confligga con altri già esistenti e usati dal compilatore per altri scopi: come `lof` (*list of figures*), `lot`, (*list of tables*), `lom`, o altri eventualmente definiti in precedenza dall'utente.⁸

8. Diamo per scontato che l'utente non usi, come suffisso, `aux`, `toc`, ecc.

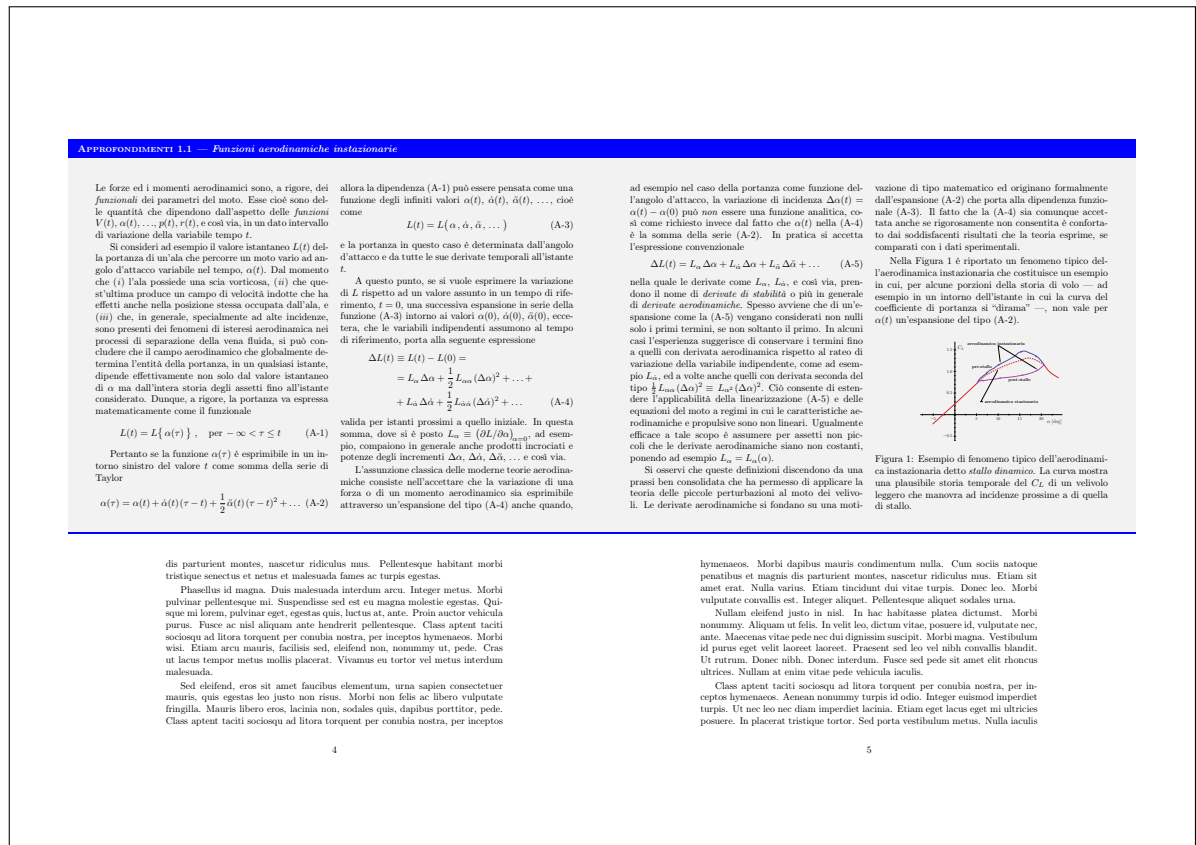


FIGURA 3: Un medaglione su due pagine, in un documento a doppia facciata.

5 Un esempio

Nella figura 3 è possibile vedere un esempio di medaglione diviso su due pagine, in un documento a doppia facciata. L'esempio è stato ottenuto con il seguente codice (per ragioni di spazio è stato omissso il contenuto del medaglione):

```
\definecolor{mylightgray}{rgb}{%
  0.95,0.95,0.95}
\columnsep 12pt
\columnseprule 0pt
\longmedal[%
  width=\textwidth+2\oddsidemargin+2in,
  innermargin=1.0cm,
  outermargin=1.0cm,
  height=0.62\textheight,
  columns=2,
  drawframe=bottom,
  framerule=2pt,
  framesep=2mm,
  framecolor=mydarkblue,
  framebgcolor=mylightgray,
  headernamefont=\bfseries\scshape,
  headertitlefont=\bfseries\itshape,
  headername=Approfondimenti,
  headertitlesep={\ ---\ },
  headertitle={Funzioni aerodinamiche
    istazionarie},
  label={\lmd:Aerodinamica:Instazionaria}
]{%

```

{testo}
}

L'opzione `width`, come detto in precedenza, è quella che controlla la larghezza totale dell'ambiente. Volendo che i medaglioni su due pagine affacciate si tocchino, come si vede nella figura, è necessario aggiungere, a destra e a sinistra, una misura pari allo spazio bianco che costituisce il margine interno. Nelle classi standard, questa misura è definita pari a `\oddsidemargin+1in`, il che spiega perché nel codice sopra riportato vengano mescolate diverse unità di misura (pratica da evitare ogni volta che è possibile).

6 Sviluppi futuri

Il pacchetto si trova già in uno stadio di sviluppo abbastanza avanzato e, naturalmente, una volta adeguatamente testato, potrà essere rilasciato su CTAN. Tuttavia ci sono ancora una serie di punti che meritano di essere ulteriormente approfonditi ed eventualmente adottati nelle versioni future di longmedal:

- attualmente l'altezza dei singoli elementi è fissa per un dato medaglione, mentre l'utente dovrebbe avere la possibilità di effettuare, entro certi limiti, una regolazione fine;

- attualmente il nome del medaglione non è configurato per adeguarsi automaticamente alla lingua selezionata con `babel`;
- attualmente `longmedal` si limita ad usare il comando `\listof`, del pacchetto `float` per generare l'elenco dei vari tipi di medaglione, mentre sarebbe più conveniente avere, per ogni elenco, comandi del tipo `\listoflongmedal`, più in linea con `\listoffigures`, ecc.;
- attualmente, ogni tipo di medaglione ha il proprio elenco separato, mentre l'utente potrebbe desiderare un elenco comprendente tutti i tipi di medaglione.

7 Ringraziamenti

Gli autori desiderano ringraziare in primo luogo il Prof. Claudio Beccari, al quale si devono preziosi suggerimenti nelle fasi iniziali della progettazione di questo pacchetto.

Un altro ringraziamento va a tutti coloro che hanno partecipato alle due discussioni sul Forum di GuIT menzionate nell'introduzione.

Infine è giusto menzionare qui l'anonimo recensore di Ar_sT_EXnica che ha contribuito a rendere più chiari per il lettore diversi punti di questo articolo.

Riferimenti bibliografici

- ADRIAENS, H. (2008). «The xkeyval package». <http://www.ctan.org/tex-archive/macros/latex/contrib/xkeyval/doc/xkeyval.pdf>.
- ANDERSON, J. D. (2004). *Introduction to Flight*. McGraw Hill Higher Education, 5th Edition.
- BECCARI, C. (2007). «Introduzione all'arte della composizione tipografica con L^AT_EX». <http://www.guit.sssup.it/downloads/GuidaGuIT.pdf>.
- CARLISLE, D. (1999). «The keyval package». <http://www.ctan.org/tex-archive/macros/latex/required/graphics/keyval.dtx>.
- (2001). «The ifthen package». <http://www.ctan.org/tex-archive/macros/latex/base/ifthen.dtx>.
- KERN, U. (2007). «Extending L^AT_EX's color facilities. the xcolor package». <http://www.ctan.org/tex-archive/macros/latex/contrib/xcolor/xcolor.pdf>.
- LINGNAU, A. (2001). «An improved environment for floats». <http://www.ctan.org/tex-archive/macros/latex/contrib/float/float.pdf>.
- MITTELBAACH, F. (2006). «An environment for multicolumn output». <http://www.ctan.org/tex-archive/macros/latex/required/tools/multicol.pdf>.
- THORUP, K. K., JENSEN, F. e ROWLEY, C. (2005). «The calc package. infix arithmetic notation in L^AT_EX». <http://www.ctan.org/tex-archive/macros/latex/required/tools/calc.pdf>.
- WILSON, P. (2003). <http://www.ctan.org/tex-archive/obsolete/macros/latex/contrib/misc/chngpage.sty>.
- ▷ Agostino De Marco
Università degli Studi di Napoli "Federico II"
Dipartimento di Ingegneria Aerospaziale
`agostino dot demarco at unina dot it`
- ▷ Massimiliano Dominici
`mlgdominici at interfree dot it`

Specifying Translated Works in Bibliographies

Jean-Michel Hufflen

... [The Tower of Babel:] In the middle of the precinct there was a tower of solid masonry, a furlong in length and breadth, upon which was raised a second tower, and on that a third, and so on up to eight. The ascent to the top is on the outside, by a path which winds round all the towers. When one is about half-way up, one finds a resting-place and seats, where persons are wont to sit some time on their way to the summit. On the topmost tower there is a spacious temple, and inside the temple stands a couch of unusual size, richly adorned, with a golden table by its side. [...]

Herodotus (ca 484 BC, ca 425 BC).
History, Book I (Clio), § 181, translated by
George Rawlinson.

Abstract

First we recall the layout recommended within a bibliography for a translation of a document. Then we explain why entries for translated works cannot be specified nicely if we use `BIBTEX`. A solution is proposed for future implementation in `MLBIBTEX`.
Keywords Bibliographies, translated works, `BIBTEX`, cross-referencing, `MLBIBTEX`.

Sommario

L'articolo espone innanzitutto qual è il formato consigliato, all'interno di una bibliografia, per la traduzione di un documento. Viene poi spiegato perché le voci per le opere tradotte non possono essere specificate in maniera soddisfacente usando `BIBTEX`. Infine viene proposta una soluzione da implementare nelle future versioni di `MLBIBTEX`.
Parole chiave Bibliografie, opere tradotte, `BIBTEX`, riferimenti incrociati, `MLBIBTEX`.

1 Introduction

English has become the international language, especially in scientific topics, so most of books and articles in Mathematics, Computer Science, Physics, ... are written in English. Often the 'References' sections of such articles only list bibliographical items written in English, in order for these items to be usable by readers, supposed to understand English texts. Obviously, this point is not universal, very simple counterexamples being given by articles published in Italian by the `ArSTeXnica` journal. In such case, citing an Italian translation of an article or book originally written in English—or

another language—may be more judicious, especially for Italian readers. Last, there is a domain where translations can indisputably be more accessible than original texts: that is History. An article using ancient texts as material—texts by ancient writers such as Herodotus or Thucydides—would probably be connected with a modern translation—possibly in a bilingual edition—rather than the original text only¹. In addition, such texts often have been translated several times, and some translations yielded actual 're-creations'², so knowing the edition and the translators' names with precision may be relevant within a bibliography.

In the next section, we recall how documents' translations should appear within the references of a bibliography. Let us also recall that when the `LATEX` word processor (Lamport, 1994) is used to typeset a document, 'References' sections are usually generated by the `BIBTEX` bibliography processor (Patashnik, 1988), from citation keys belonging to an auxiliary (`.aux`) file, some entries stored in bibliography database (`.bib`) files³, and a bibliography style ruling references' layout, written using the `bst` language described in Mittelbach et al. (2004, § 13.6). Section 3 shows that the specification of such translations using `BIBTEX`'s `.bib` format may yield *information redundancy*. Section 4 explains how we plan to solve this problem in `MLBIBTEX`⁴ (Hufflen, 2003), our multilingual reimplement of `BIBTEX`. Reading this article only requires a basic practice of `LATEX` and `BIBTEX`. Section 4 also uses some notions related to XML⁵.

1. About the two examples of ancient writers given above, we personally studied ancient Greek at grammar school. Translating Herodotus' or Thucydides' texts is a worthwhile exercise if you are interested in ancient History. However, it requires a great amount of time, and when we were doing that, we often thought that when a Greek people living at this time asked to eat, he would probably be dead of starvation if, for his contemporaries, it took as much time as us to understand him. A strong argument for the availability of modern translation of such texts.

2. For example, there are many prose translations of *Odyssey*, an epic poem attributed to Homer. As different work, it has also been translated into French verse by Victor Bérard (1864–1931).

3. In this article, we refer to precise terminology, coming from `MLBIBTEX`'s (Hufflen, 2003): bibliographical *entries* are specified in `.bib` files, and bibliographical *references*—in `.bbl` files when they are generated by `BIBTEX` or `MLBIBTEX` (Mittelbach et al., 2004, § 12.1.3)—are to be typeset by a word processor.

4. `MultiLingual BIBTEX`.

5. `eXtensible Markup Language`. Readers interested in an introductory book to this formalism can consult Ray (2001).

2 Translations in references

Typography manuals only give a few details about citations of translations. If the only information about the original work consists of its author and title, Butcher (1992, § 10) recommends that such a reference is organised as follows:

Hesiod. *Theogony—Works and Days*, tr.
M. L. West (Oxford University Press,
1988).

The original title may follow the translated title, as shown by the more expanded reference below:

Hesiod. *Theogony — Works and Days*
[Θεογονία — Ἔργα καὶ ἡμέραι], trans-
lated from ancient Greek by Martin
Lichtfield West (Oxford University
Press, 1988).

—parentheses may be used instead of square brackets to surround the original title⁶ (Chicago, 1993, § 15.228)—what is accurate for an ancient text⁷. If we would like to give more information about the original publisher and publication date, these details may come at first:

Kenneth Robeson. *Hex*. No. 37 in *Doc Savage Series* (Bantam, June 1969);
tr. Angela Campana as ‘L’autostrada
stregata’, (Arnoldo Mondadori Editore,
ottobre 1974).

or at second (Chicago, 1993, § 15.169):

Umberto Eco. *The Name of the Rose*. Har-
court, 1983, originally published as *Il
nome della rosa* (Bompiani, 1980).

the *Chicago manual of Style* recommends that a translation of a work does not include the original edition unless the full citation of the latter is given. Of course, all these conventions and some other variations can be programmed with accurate bibliography styles of BibTEX (Mittelbach et al., 2004, §§ 13.5 & 13.6), once new fields related to translations are added.

6. In typography, square brackets usually mark up editorial omissions or additions within texts (Butcher, 1992, § 11.5.2). Square brackets can also be used within bibliographies to surround author names when a work’s authorship is known but not revealed on the title page (Chicago, 1993, § 15.90):

Marion Zimmer Bradley [and Diana L. Paxson].
Priestess of Avalon. HarperCollins Publish-
ers, 2001.

a question mark being added when the author’s identity is merely surmised:

[Mersenne Marin?] *Campanae Parisienses —
Aria*. In Ottorino Respighi’s *Ancient Airs
and Dances, 2nd Suite*.

7. To be complete, we should make precise the Greek form of the author’s name, i.e., ‘Hesiod [Ἡσίοδος]’, since we give the Greek title. But in this article, we will not go thoroughly into the transliteration of words written with non-Latin alphabets.

3 Translations in database files

The problem we address here is related to *information redundancy*. Let us assume that we are written an article in Italian, and cite the following Italian translation of a book originally written in English:

Kenneth Robeson. *Taz*. Traduzione italia-
na di *Doc Savage #27, Mystery Un-
der the Sea* (Bantam, 1968) di Bianca
Russo. Arnoldo Mondadori Editore,
settembre 1975.

If our text is processed by LATEX and our ‘References’ section built by BibTEX, the latter reference could be derived from the following entry:

```
@BOOK{robeson-it1975,  
  AUTHOR = {Kenneth Robeson},  
  TITLE = {Taz},  
  ...  
  NOTE = {Traduzione italiana...},  
  ...}
```

some new fields relevant for the information about the translation may be added, provided that bibliography styles used are able to process them (Mittelbach et al., 2004, § 13.6.3).

Of course, if we write an article in English about the *Doc Savage* series, we cite the original text of this science-fiction story:

```
@BOOK{robeson1968f, ...}
```

In this way, the original title, publisher, and year of publication would be duplicated in our bibliography database (.bib) file⁸, which induces some redundancy. If there is a mistake about these fields, it must be fixed in several places: the entry for the original text and each entry specifying one of this text’s translations.

A first solution is given in Figure 1: each specification of a translation refers to the original text’s entry. If you only cite the `robeson-fr1972` or `robeson-it1975` entry, you just need an additional BibTEX and two additional LATEX runs since the `robeson1968f` entry is added automatically at BibTEX’s first run. That may be good technique if the two translations—`robeson-fr1972a` and `robeson-it1975`—are cited, since all the information about the original text is given only once. If only the Italian translation is cited, adding a reference for the original text may be viewed as artificial increase of the ‘References’ section’s size.

As an alternative, the Refer format⁹, recognised by TIb (Alexander, 1989), a bibliography processor

8. We can use several .bib files, but the keys used throughout such files belong to the same namespace. So we can assume that there is only one .bib file, without loss of generality.

9. The `pybibliographer` program can be used as a converter from the .bib format to the Refer format: see Mittelbach et al. (2004, § 13.4.5) for more details.

```

@BOOK{robeson1968f,
  AUTHOR = {Kenneth Robeson},
  TITLE = {Mystery Under the Sea},
  SERIES = {Doc Savage},
  NUMBER = 27,
  PUBLISHER = {Bantam},
  YEAR = 1968,
  MONTH = aug}

@BOOK{robeson-fr1972a,
  AUTHOR = {Kenneth Robeson},
  TITLE = {Le myst\‘{e}re sous la mer},
  SERIES = {Doc Savage},
  NUMBER = 27,
  PUBLISHER = {Pocket Marabout},
  NOTE = {Traduction française de
    \cite{robeson1968f} par
    Claude Olivier},
  YEAR = 1972}

@BOOK{robeson-it1975,
  AUTHOR = {Kenneth Robeson},
  TITLE = {Taz},
  SERIES = {Doc Savage},
  NUMBER = 15,
  PUBLISHER = {Arnoldo Mondadori
    Editore},
  NOTE = {Traduzione italiana di
    \cite{robeson1968f} di Bianca
    Russo},
  YEAR = 1975,
  MONTH = sep}

```

FIGURE 1: Original book and two translations.

designed for use with the Plain T_EX typeset engine (Knuth, 1984), allows a translated work to be specified using more fields than in BibT_EX, as shown in Figure 2. Let us remark that this formulation is more structured than the first version of our `robeson-it1975` entry. However, dealing with it also induces an information redundancy problem if both the original text and the Italian translation are cited throughout the same document, because the entry given in Figure 2 cannot be considered sometimes partially—that is, by ignoring the fields `%t`, `%a`, `%i`, `%s`, `%n`, `%o`, `%d`—sometimes globally when processed by a bibliography style. The same problem of information redundancy holds if several translations of a same work are to be considered.

In fact, BibT_EX provides a way to share some information common to several entries, as we show it in Figure 3 with the specification of another edition of *Mystery Under the Sea*. As explained in Mittelbach et al. (2004, § 13.2.5), the special `CROSSREF` field tells BibT_EX that the missing fields of the `robeson1983cb` entry can be found within the `robeson1983c` entry. If this last entry is cross-referenced by a certain number of entries (defaulting to 2), BibT_EX will automatically put it on the reference list. Otherwise, if

```

%A Kenneth Robeson
%T Mystery Under the Sea
%P Bantam
%S Doc Savage
%N 27
%D 1968
%t Taz
%a Bianca Russo
%i Arnoldo Mondadori Editore
%s Doc Savage
%n 15
%o Traduzione italiana di Bianca Russo
%d 1975

```

FIGURE 2: Translation’s specification in the Refer format.

```

@INBOOK{robeson1983cb,
  TITLE = {Mystery Under the Sea},
  CROSSREF = {robeson1983c},
  VOLUME = 27,
  PAGES = {135--260}}

@BOOK{robeson1983c,
  AUTHOR = {Kenneth Robeson},
  TITLE = {Death in Silver---Mystery
    Under the Sea},
  SERIES = {Doc Savage},
  NUMBER = {26--27},
  PUBLISHER = {Bantam},
  YEAR = 1983,
  MONTH = nov}

```

FIGURE 3: Using a cross-reference in BibT_EX.

the `robeson1983c` entry is not cited throughout the document and if only the `robeson1983cb` entry cross-references it, the information originating from the `robeson1983cb` entry—corresponding to the `AUTHOR`, `SERIES`, `PUBLISHER`, `YEAR`, and `MONTH` fields—will be directly added to the reference built from the `robeson1983cb` entry. On modern BibT_EX’s versions, the `--min-crossrefs` option allows end-users to adjust this behaviour, the default setting being:

```
bibtex --min-crossrefs=2 job-name
```

`job-name` being the name of the `.aux` file processed.

Unfortunately, this kind of inheritance does not apply to translations: the translation’s title is given by the `TITLE` field, and the `TITLE` field of the original text’s entry will not be used if it is cross-referenced by the translation’s entry. When a `@PROCEEDINGS` entry is cross-referenced by some articles belonging to these proceedings, the `BOOKTITLE` field is used for the proceedings’ title—whereas the `TITLE` field stands for an article’s title—but this `BOOKTITLE` field is irrelevant for a `@BOOK` entry.

4 Future

As abovementioned, MIBIB_TE_X is a reimplementation of BIB_TE_X with particular focus on multilingual features. Its present version could replace BIB_TE_X, except for cross-referencing entries, that is planned for MIBIB_TE_X's next version¹⁰, and we also plan to be able to deal with translations nicely. MIBIB_TE_X's architecture has been presented in Hufflen (2006), let us just recall that parsing .bib files results in an XML tree.

Figure 4 shows the planned implementation of ‘actual’ cross-references among entries and translations referring to another entry: by means of two attributes, **crossref** and **translates**. We have also added language information, modelled by the **language** attribute: in the .bib files MIBIB_TE_X can process, it is given by a **LANGUAGE** field.

We can notice that this specification avoids information redundancy as much as possible. In addition, when a bibliography style requires that an original edition is fully cited in its translations, we could implement the same behaviour as for ‘classical’ cross-references. That is, if an ‘original entry’ is referred by a certain number of translations, it will be automatically put in the reference list. Otherwise, the full citation of an original edition will be appended to the information related to one of its translations. Of course, this ‘certain number of translations’ could be controlled by an option.

Another important advantage is related to the information about natural languages used to write the different parts of a bibliographical entry. If an entry’s language is **french**, the whole information is supposed to be in English, unless another language is put explicitly by means of a **foreigngroup** element. That is why we put this element in the specification of the **series** element of the **robeson-fr-1972** entry in Figure 4. This field should be specified as:

```
SERIES = {[Doc Savage] : english},
```

as part of the extended concrete syntax recognised by MIBIB_TE_X's parser. If the **AUTHOR** field was given in the **robeson-fr1972** entry, we should also make precise:

```
AUTHOR = {[Kenneth Robeson] : english},
```

in order to prevent incorrect hyphenation of these words, since the French and English rules for that are different. This information about the author

10. More precisely, MIBIB_TE_X is able to process cross-referencing entries if you use the compatibility mode, allowing ‘old’ bibliography styles written in the **bst** language to be run. Whereas BIB_TE_X requires that a cross-referenced entry appears *after* the first entry referencing it (as we do in Figure 3), this limitation has been overcome in MIBIB_TE_X's compatibility mode. On the other hand, MIBIB_TE_X can be used with multilingual bibliography styles written in **nbst** (New Bibliography **S**Tyles) (Hufflen, 2003), but these styles cannot deal with cross-referencing entries presently.

```
<book id="robeson1968f" language="english">
  <author>
    <name>
      <personname>
        <first>Kenneth</first>
        <last>Robeson</last>
      </personname>
    </name>
  </author>
  <title>Mystery Under the Sea</title>
  <publisher>Bantam</publisher>
  <year>1968</year>
  <month><aug/></month>
  <number>27</number>
  <series>Doc Savage</series>
</book>

<inbook id="robeson1983ca" language="english"
  crossref="robeson1983c">
  <title>Death in Silver</title>
  <volume>26</volume>
  <pages>
    <firstpage>1</firstpage>
    <lastpage>133</lastpage>
  </pages>
</inbook>

<book id="robeson-fr1972a" language="french"
  translates="robeson1968f">
  <translator>
    <name>
      <personname>
        <first>Claude</first>
        <last>Olivier</last>
      </personname>
    </name>
  </translator>
  <title>Le mystère sous la mer</title>
  <publisher>Pocket Marabout</publisher>
  <year>1972</year>
  <number>27</number>
  <series>
    <foreigngroup language="english">
      Doc Savage
    </foreigngroup>
  </series>
</book>

<book id="robeson-it1975" language="italian"
  translates="robeson1968f">
  <translator><name>...</name></translator>
  <title>Taz</title>
  <publisher>Arnoldo Mondadori...</publisher>
  <year>1975</year>
  <month><sep/></month>
  <number>15</number>
  <series>...</series>
</book>
```

FIGURE 4: XML-like tree for translations.

belonging to the entry referred by the **translates** attributes, it is supposed to be in English since this entry, **robeson1968f**, is declared as an entry for a work in English.

Some experiments about this implementation are promising, but we have to update the bibliography styles in order for them to be able to deal with cross references and translations. We plan to propose several `nbst` bibliography styles, according to the amount of information imported from the original work's specification. As we explained in Section 2, some styles just import the authors' names, some also put the original work's title, possibly with the publisher and date of publication. We have to design some new concrete syntax for `.bib` files: two new fields, `TRANSLATES` and `TRANSLATOR`, this last field's taxonomy being the same as `AUTHOR`'s or `EDITOR`'s in 'classical' `BIBTEX`.

Last, let us notice that such a specification eases the search for translations if we consider XML trees and expressions belonging to `XPath`, the language used to address parts of an XML documents (W3C, 1999). For example, all the translations of the `robeson1968f` entry in the XML tree in Figure 4 can be caught by the `XPath` expression:

```
/mlbiblio/*[@translates="robeson1968f"]
```

`mlbiblio` being the root element of the trees resulting from parsing `.bib` files by `MLBIBTEX`.

5 Conclusion

We confess that when we cited translations, we often used specifications like those in Figure 1 for a long time. And recently, we studied the meaning of the fields handled by the `Refer` format, since we wish `MLBIBTEX` to be able to process this format in its next version. This led us to decide how to deal with `Refer` fields related to translations and to try to solve this problem of information redundancy nicer than we did before. Like cross-referencing entries, our proposal for translation specifications eases information sharing. We plan to look into other possible information sharing within bibliographies before starting implementation up.

Acknowledgements

I am grateful to Massimiliano Dominici, who kindly provided the Italian fragments that appear throughout the document, including the abstract's translation. I also thank the referee who conscientiously proof-read the first version. Thanks to this issue's Chief Editor, Gianluca Pignalberi, too.

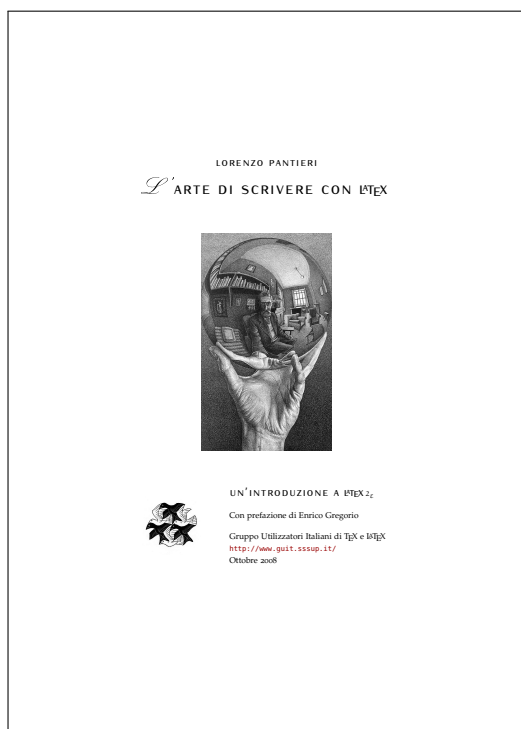
References

- James C. Alexander. *Tib: a T_EX Bibliographic Pre-processor*, 1989. Version 2.2, see CTAN:`biblios/tib/tibdoc.tex`.
- Judith Butcher. *Copy-Editing. The Cambridge Handbook for Editors, Authors, Publishers*. Cambridge University Press, 3rd edition, 1992.
- Chicago. The Chicago manual of style. The University of Chicago Press, 1993. The 14th edition of a manual of style revised and expanded.
- Jean-Michel Hufflen. `MLBIBTEX`'s version 1.3. *TUGboat*, 24(2):249–262, July 2003.
- Jean-Michel Hufflen. `MLBIBTEX` architecture. *Ar_sT_EX_nica*, 2:54–59, October 2006. In GUIT 2006 meeting.
- Donald Ervin Knuth. *Computers & Typesetting. Vol. A: The T_EXbook*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1984.
- Leslie Lamport. *L_AT_EX: A Document Preparation System. User's Guide and Reference Manual*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1994.
- Frank Mittelbach and Michel Goossens, with Johannes Braams, David Carlisle, Chris A. Rowley, Christine Detig, and Joachim Schrod. *The L_AT_EX Companion*. Addison-Wesley Publishing Company, Reading, Massachusetts, 2nd edition, August 2004.
- Oren Patashnik. *BIB_TE_Xing*. Part of the `BIBTEX` distribution, February 1988.
- Erik T. Ray. *Learning XML*. O'Reilly & Associates, Inc., January 2001.
- W3C. *XML Path Language (XPath). Version 1.0*. <http://www.w3.org/TR/xpath>, November 1999. W3C Recommendation. Edited by James Clark and Steve DeRose.

▷ Jean-Michel Hufflen
LIFC (EA CNRS 4269)
University of Franche-Comté
16, route de Gray
25030 BESANÇON CEDEX
FRANCE
hufflen at lifc dot
univ-fcomte dot fr

Eventi e novità

Presentazione del libro di Lorenzo pantieri *L'arte di scrivere con L^AT_EX*



Il Gruppo Utilizzatori Italiani di T_EX è lieto di presentare la nuova guida introduttiva all'uso di L^AT_EX, realizzata da Lorenzo Pantieri. L'opera, frutto dell'esperienza personale dell'autore e delle numerose discussioni sul forum del Gruppo, è stata concepita con l'intento di presentare, ad un pubblico non solo di principianti, un'esposizione organica e sistematica, per quanto sintetica, dei concetti fondamentali di L^AT_EX, e una disamina, corredata di esempi, di alcuni problemi tipici che si incontrano nella stesura di una pubblicazione scientifica o professionale.

Per mantenere le caratteristiche di sintesi e concisione a cui è improntato, il manuale non appesantisce la lettura con un'analisi dettagliata dei vari temi, lasciando che il lettore li approfondisca attraverso lo studio di guide più avanzate, o dei manuali dei singoli pacchetti.

Tutti gli argomenti sono affrontati dalla prospettiva di un utente che componga il proprio documento in lingua italiana (particolarmente utile, a questo scopo, risulta la breve appendice che riassume le norme tipografiche italiane), fornendo così indicazioni che difficilmente potrebbero essere reperite in altri manuali.

Il testo, di cui alcune copie sono state rese disponibili durante il G_LTmeeting, è liberamente scaricabile all'indirizzo <http://www.guit.sssup.it/downloads/>.

BachoTeX 2009

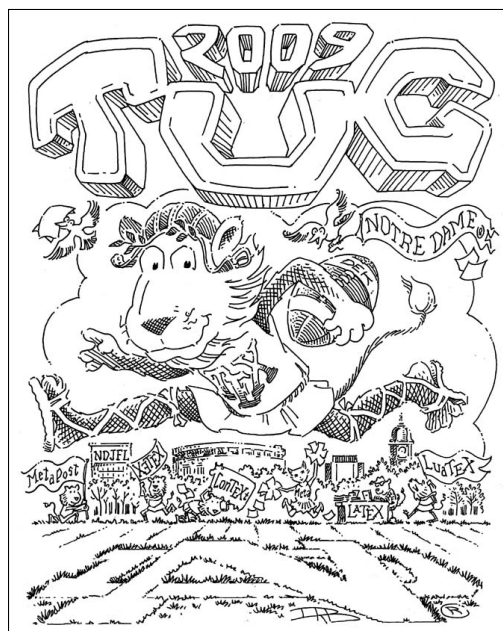
Il XVIII convegno BachoTeX, organizzato dallo User Group polacco (GUST), si terrà dal 29 Aprile al 3 Maggio 2009 a Bachotek, nei pressi di Brodnica, in Polonia.

TUG Conference 2009

Il convegno annuale 2009 del T_EX Users Group si terrà dal 28 al 31 luglio 2009 presso l'Università di Notre Dame, a Notre Dame, Indiana, USA.

L'organizzazione locale è curata principalmente da Martha Kummerer, del *Notre Dame Journal of Formal Logic* (<http://www.nd.edu/~ndjfl/>).

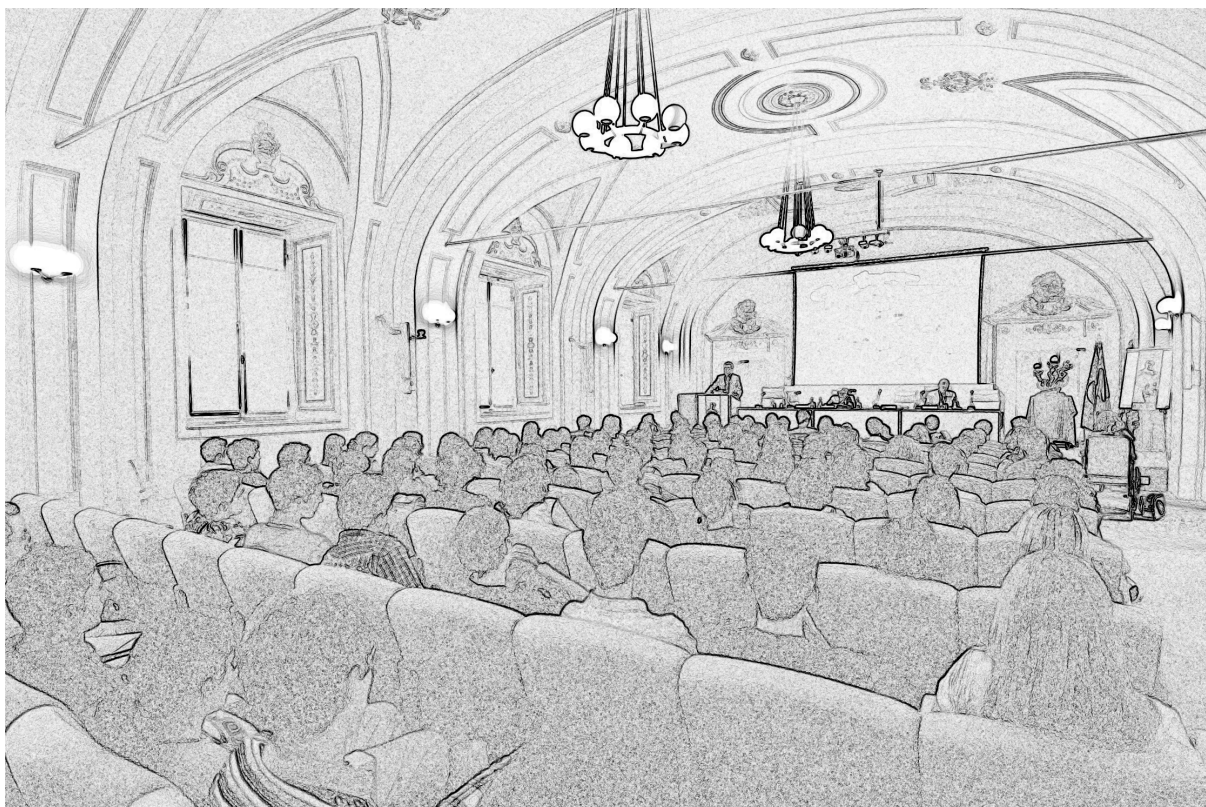
Ulteriori informazioni saranno via via pubblicate sulla pagina internet dedicata al convegno: <http://www.tug.org/tug2009/>.



EuroTeX Conference 2009 e 3rd ConTeXt Meeting

Nel 2009, i convegni EuroTeX e ConTeXt si terranno entrambi congiuntamente all'Aia, in Olanda, dal 24 al 28 Agosto, organizzati dal locale User Group (NTG).

Il programma e altri particolari sull'organizzazione del convegno verranno comunicati in seguito sulla pagina internet <http://www.ntg.nl/EuroTeX2009/index.html>.



Programma del Convegno

Sessione Mattutina

09:00 – Registrazione

09:30 – A short introduction to MetaPost

Klaus Höppner, DANTE e.V.

10:00 – Gestione avanzata delle figure in \LaTeX :
l'annotazione di illustrazioni e grafici con
`psfrag`/`PSTricks` e `PGF/Tikz`

*Agostino De Marco, Università degli Studi di Napoli
“Federico II”*

10:30 – Una tabella che fa calcoli

Roberto Giacomelli

11:00 – Coffee break

11:30 – Gestire la bibliografia con \LaTeX

Lapo Mori, Università degli Studi di Pisa

12:00 – Introduzione allo stile `ClassicThesis`

Lorenzo Pantieri

12:30 – Pausa Pranzo

Sessione Pomeridiana

14:30 – \TeX Live 2008 and the \TeX Live Manager

Norbert Preining, Vienna University of Technology

15:00 – \LaTeX e grammatiche (la faccia triste dell'informatica)

Gianluca Pignatelli, Free Software Magazine

15:30 – Coffee break

16:00 – Longmedal: un pacchetto per medaglioni
divisi su più pagine

Massimiliano Dominici,

*Agostino De Marco, Università degli Studi di Napoli
“Federico II”*

16:30 – Specifying translated works in bibliographies

Jean Michel Hufflen, Université de Franche Comté

17:00 – \LaTeX Help Desk (esperti di \LaTeX a vostra
disposizione)

18:00 – Chiusura dei lavori e cena sociale

Questa rivista è stata stampata
presso Logo S.r.l. Servizi di Stampa Digitale, Borgoricco (PD)
su carta ecosostenibile Vision Trend White
prodotta da Steinbeis Temming Papier GmbH & Co.



ArTeXnica – Call for Paper

La rivista è aperta al contributo di tutti coloro che vogliano partecipare con un proprio articolo. Questo dovrà essere inviato alla redazione di ArTeXnica, per essere sottoposto alla valutazione di recensori entro e non oltre il 14 Marzo 2009. È necessario che gli autori utilizzino la classe di documento ufficiale della rivista; l'autore troverà raccomandazioni e istruzioni più dettagliate all'interno del file d'esempio (.tex).

Gli articoli potranno trattare di qualsiasi argomento inerente al mondo di L^AT_EX e non dovranno necessariamente essere indirizzati ad un pubblico esperto. In particolare tutorials, rassegne e analisi comparate di pacchetti di uso comune, studi di applicazioni reali, saranno bene accettati, così come articoli riguardanti l'interazione con altre tecnologie correlate.

Di volta in volta verrà fissato, e reso pubblico sulla pagina web <http://www.guit.sssup.it/arstexnica>, un termine di scadenza per la presentazione degli articoli da pubblicare nel numero in preparazione della rivista. Tuttavia gli articoli potranno essere inviati in qualsiasi momento e troveranno collocazione, eventualmente, nei numeri seguenti.

Chiunque, poi, volesse collaborare con la rivista a qualsiasi titolo (recensore, revisore di bozze, grafico, etc.) può contattare la redazione all'indirizzo arstexnica@sssup.it.

ArsT_EXnica

Rivista italiana di T_EX e L^AT_EX

Numero 6, Ottobre 2008

- 3 Editoriale
Gianluca Pignalberi, Massimiliano Dominici
- 5 Introduction to METAPOST
Klaus Höppner
- 10 Gestione avanzata delle figure in L^AT_EX
Agostino De Marco
- 28 Una tabella che fa calcoli
Roberto Giacomelli
- 37 Bibliografia e L^AT_EX
Lapo F. Mori
- 52 Introduzione allo stile ClassicThesis
Lorenzo Pantieri
- 67 T_EX Live 2008 and the T_EX Live Manager
Norbert Preining
- 76 L^AT_EX e grammatiche
Gianluca Pignalberi, Enrico Bini
- 86 longmedal: un pacchetto per medaglioni divisi su più pagine
Agostino De Marco, Massimiliano Dominici
- 93 Specifying Translated Works in Bibliographies
Jean-Michel Huppen

