

Le filigrane e le figure di sfondo

Claudio Beccari

Sommario

Il sistema $\text{\TeX}$ offre diversi pacchetti per inserire nei documenti filigrane o immagini di sfondo, ma ognuno di questi pacchetti ha le sue caratteristiche particolari che lo rendono utile in certe applicazioni più che in altre. In questo articolo si vuole esaminare a fondo il modo di introdurre queste decorazioni per comprenderne meglio il meccanismo.

Abstract

Our $\text{\TeX}$ system has several packages available for inserting watermarks or background images. But each one of these packages has specific features that make it more suitable in certain applications rather than others. In this article the mechanism for inserting such “decorations” is thoroughly examined so as to fully understand how it works.

1 Introduzione

Le filigrane e le figure di sfondo hanno qualcosa in comune, sono “disegnate” sotto il testo; in altre parole, ci si scrive sopra. È vero che le immagini di sfondo solitamente impegnano tutta la pagina, mentre le filigrane di solito rimangono all’interno della griglia di stampa, ma queste sono differenze non sostanziali, mitigate dal “di solito” che fa sì che esistano immagini di sfondo che non occupano l’intera pagina oppure filigrane che escono dalla griglia di stampa.

La differenza sostanziale è che le figure di sfondo sono spesso, per non dire sempre, costituite da immagini a colori vivi, più o meno saturi; sopra i quali bisogna scrivere mediante tinte contrastanti con font colorati affinché il testo sia distinguibile dallo sfondo. Le filigrane sono immagini o parole composte con colori poco saturi ossia con una elevata componente di bianco e sulle quali il testo nero risalta senza interferenze.

Le filigrane sono adatte sia per mettere un’immagine di tonalità fortemente attenuata al di sotto del testo di una presentazione, per esempio composta con la classe `beamer`, sia per porre trasversalmente alla pagina scritte di fondo del tipo “BOZZA”, “DRAFT”, “RISERVATO”, “CLASSIFIED”. Queste appariranno sotto il testo in modo da fornire informazioni relative alla natura del documento e rendendo la loro eliminazione impossibile per semplice fotocopiatura.

Le immagini di sfondo vanno bene per le copertine dei documenti; le filigrane possono anda-

re bene per qualsiasi pagina interna di qualsiasi documento.

In ogni caso le une o le altre non debbono venire usate solo perché $\text{\LaTeX}$ consente di farlo.

In questo articolo parlerò essenzialmente di filigrane, anche se molte delle cose che esporrò potranno essere valide anche per le immagini di sfondo.

Ho deciso di scrivere questo articolo stimolato da una discussione sul Forum $\text{\G\TeX}$; chiedeva infatti un partecipante: “Ciao, vorrei inserire la scritta DRAFT in diagonale su ciascun foglio del documento.” Questa discussione si è poi sviluppata con diversi interventi.

Ora, senza scendere nell’analisi del codice di questo o quel pacchetto, vorrei esporre che cosa c’è dietro a queste operazioni.

2 Breve rassegna dei pacchetti esistenti

Ovviamente la soluzione più semplice consiste nell’impiegare i pacchetti già predisposti per questo compito, come `eso-pic` (NIEPRASCHK, 2010) e `watermark` (ROZHENKO, 2004). Anzi, `watermark` è fatto proprio per questo scopo, tanto che il suo uso è talmente facile quanto disarmante:

```
\usepackage{watermark}
\watermark{DRAFT}
```

e da qui in poi il pacchetto fa tutto da solo. Il pacchetto `eso-pic` richiede forse un paio di righe in più. Secondo la mia impressione `eso-pic` è più adatto per le immagini di sfondo, mentre `watermark` è esplicitamente dedicato alle filigrane.

Il risultato può essere ottenuto anche con l’uso del pacchetto `TikZ` (TANTAU, 2010), diventato di fatto il software di prima scelta per eseguire qualsiasi operazione grafica su qualunque documento. Claudio Fiandrino¹ ha fornito la seguente soluzione²:

```
1 \documentclass{article}
2 \usepackage{tikzpagenodes}
3 \usetikzlibrary{calc}
4 \usepackage[contents={}]{background}
5
6 \usepackage{lipsum}% for dummy text
```

1. Collaboratore molto competente del Forum e autore di alcune Guide Tematiche sull’impiego di programmi che utilizzano pesantemente la grafica

2. Il colore e l’opacità della filigrana sono stati modificati per poter disporre di un’immagine in tonalità di grigio, come così pure il comando per l’inserimento in calce del nome dell’autore (estraneo al tema di questo articolo).

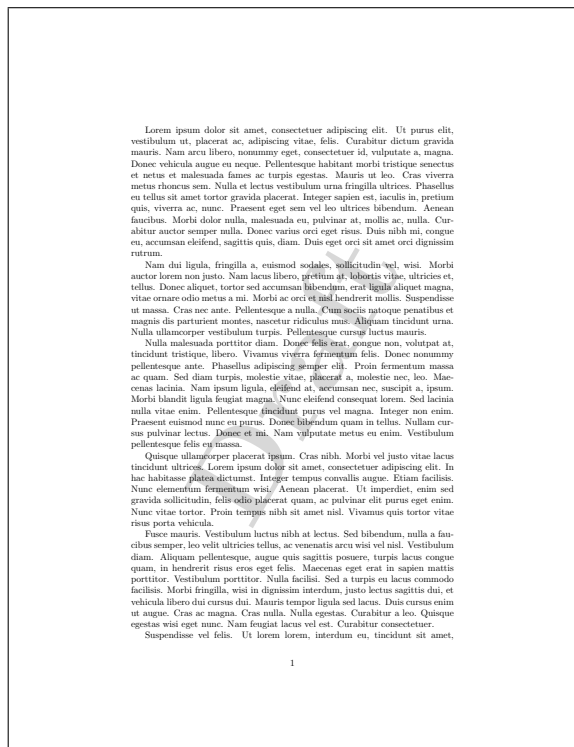


FIGURA 1: Risultato del codice di Claudio Fiandrino mediante l'uso del pacchetto TikZ

```

7
8 \newcommand{\draftlogo}[1][darkgray,rotate=60]{%
9 \tikz[remember picture,overlay]{%
10 \node[scale=5,font=\bfseries\Huge,
11 opacity=0.2,#1] at (current page.center)
12 {\Draft};
13 }%
14 }
15
16 \AddEverypageHook{%
17 \draftlogo
18 \BgMaterial}
19
20 \begin{document}
21 \lipsum[1-8]
22 \end{document}

```

Per ottenere il risultato, mostrato nella figura 1, il codice dell'esempio deve essere compilato due volte.

Leggendo il codice ed osservando la figura si comprende come la filigrana venga centrata nella pagina e non nel centro della griglia di stampa, e come venga ottenuta l'inclinazione di 60° della filigrana stessa. Inoltre, avendo impiegato la classe *article*, lo specchio di stampa è quasi centrato rispetto al foglio di carta (in questo caso di formato *letter*) facendo apparire la filigrana sostanzialmente al centro dello specchio di stampa. Componendo fronte retro la cosa potrebbe non funzionare al meglio, ma comunque le modifiche da apportare sarebbero poche. Non c'è dubbio che TikZ consente di fare operazioni grafiche molto elaborate con pochi comandi relativamente semplici, ma l'impostazione del risultato non è proprio

banale; a confronto, l'uso del pacchetto *watermark* è un giochetto da ragazzi.

Concretamente, mi riferisco qui alle condizioni per un “concorso” scherzoso che ho bandito sul forum $\text{\LaTeX}$:

- creare il codice per inserire una filigrana usando solo le potenzialità del codice di $\text{\LaTeX}$ e del motore di composizione con cui gira il programma *pdflatex*;
- è consentito usare (ma non è obbligatorio farlo) il pacchetto *pict2e*, che estende le funzionalità del linguaggio originale di $\text{\LaTeX}$ secondo le linee indicate da Leslie Lamport stesso nella seconda edizione del suo manuale (LAMPOR, 1994);
- è consentito usare (ma non è obbligatorio farlo) il pacchetto *graphicx* (CARLISLE, 2005).

Nello sviluppare il codice che soddisfi ai requisiti del “concorso” ci si rende conto dei problemi da affrontare e dei “trucchi” a cui bisogna ricorrere. Ma, più importante di tutti, ci si rende conto del funzionamento del meccanismo per inserire le filigrane e, *mutatis mutandis*, per inserire le immagini di sfondo.

Aggiungerò alcune altre condizioni:

- la filigrana deve essere collocata al centro dello specchio di stampa, non al centro della pagina;
- l'inclinazione della filigrana deve essere pari a quella della diagonale dello specchio di stampa; è consentita sia una soluzione manuale sia una soluzione automatica;
- deve essere possibile specificare a piacere la grandezza e il colore della filigrana; per questo scopo è accettabile usare almeno il pacchetto *color* (CARLISLE, 2005) o, meglio ancora, il pacchetto *xcolor* (KERN, 2007); supponendo che il codice da scrivere venga conservato in un file *filigrana.sty*, diventa possibile ricorrere a comandi interni di $\text{\LaTeX}$ che sono formati con nomi che contengono il carattere @;
- ogni personalizzazione deve poter essere eseguita dopo il caricamento del file *filigrana.sty*.

3 Premessa operativa

La filigrana deve essere inserita sul supporto di stampa, sia questo lo schermo o un foglio di carta, prima della collocazione del testo sulla medesima pagina. La filigrana altrimenti coprirebbe il testo rendendolo, in mancanza di colori trasparenti, illeggibile.

Quindi per essere sicuri dove la filigrana va inserita bisogna conoscere almeno l'essenziale del funzionamento della routine di uscita di $\text{\LaTeX}$, quella

che spedisce una pagina completamente composta accodandola al file di uscita.

Questa routine è poco documentata nelle distribuzioni del sistema TEX ma l'essenza di questa procedura, al fine di quello che ci interessa qui, è sapere che tale routine pone come prima cosa sulla pagina in costruzione la testatina, poi il blocco del testo, accompagnato dalle eventuali note marginali e dagli altri oggetti mobili (come figure e tabelle) che trovano posto nella pagina, ed infine mette il piedino. Completata questa costruzione la pagina viene accodata al file di uscita. Possiamo dunque immaginare che ciò che va sulla pagina sia costituito da tre strati, il più profondo dei quali contiene la testatina, quello intermedio contiene il corpo del testo e quello più in superficie contiene il piedino.

Ogni filigrana deve stare quindi sotto lo strato del testo, quindi deve essere inserita insieme alla testatina. Questo è un buon posto, anche perché la testatina ha una posizione ben fissa sulla pagina e di solito non subisce alterazioni se la classe del documento e le personalizzazioni inserite dall'utente non violano qualche regola.

Tuttavia, anche se collocata al medesimo livello di profondità, la filigrana non fa parte della testatina nel senso che non giace sulla riga dove solitamente si trovano le informazioni che compaiono nella testatina. Inoltre esistono degli stili di pagina privi di testatina (per esempio gli stili “empty” e “plain”).

In effetti in quest'ultimo caso le testatine sono ancora presenti ma vuote. Il problema iniziale sembra invece più difficoltoso: apparentemente ogni comando di marcatura di LATEX che produce del testo occupa spazio sul supporto di stampa. In realtà ci sono due comandi `\rlap` (*right overlap*, ricopri a destra) e `\llap` (*left overlap*, ricopri a sinistra) raramente usati ma disponibili. Esiste anche il comando `\makebox` a livello utente (BECCARI, 2012) che produce una scatola di larghezza specificata, eventualmente nulla, dove il testo può uscire tranquillamente dalla scatola se più largo della larghezza specificata. Tuttavia questi comandi, pur non occupando spazio orizzontale, o una parte inferiore a quella richiesta dalla larghezza naturale del loro contenuto, occupano sempre un'altezza e una profondità corrispondenti a quelle del loro contenuto.

Tuttavia i comandi originali di marcatura di LATEX contenevano l'ambiente `picture` il quale nella versione originale del 1985 (LATEX 209) era veramente molto limitato, ma ha perso virtualmente ogni limitazione (nel senso che Leslie Lamport stesso ha definito nella seconda edizione del suo manuale come avrebbe dovuto estendersi) con la versione del 1994 (LATEX). L'ambiente è stato “esteso” solo nel 2003 mediante il pacchetto `pict2e` (GÄSSLEIN *et al.*, 2011) e questo pacchetto ha poi subito successive estensioni, l'ultima delle quali risale al 2011.

Dopo tali estensioni, l'ambiente `picture` non ha virtualmente nessuna limitazione, anche se molte delle cose che si possono fare semplicemente con TikZ richiedono invece un grosso lavoro con `picture`.

Per questo motivo l'ambiente `picture`, esteso con il suo pacchetto `pict2e`, non è molto conosciuto ma per cose semplici come le filigrane è facilissimo da usare; anzi, non è nemmeno necessaria l'estensione `pict2e`. I pacchetti `eso-pic` e `watermark` si appoggiano infatti all'ambiente `picture`, e così farò anche qui.

Il pregio fondamentale è che l'ambiente `picture` consente di definire un disegno eseguito su una tela di dimensioni nulle e grazie ai suoi comandi di posizionamento consente di collocare qualunque cosa fuori dall'area del disegno senza bisogno di ricorrere ad alcun artificio specifico.

Ecco dunque che abbiamo trovato come mettere nelle testatine un oggetto di dimensioni nulle, dal quale “fuoriescono” altri oggetti, immagini, scritte o filigrane, collocabili ovunque senza alterare il testo.

4 Soluzione manuale

Creiamo un file `filigrana.sty` che contenga tutte le definizioni che occorrono. Cominciamo dunque a descrivere una prima soluzione manuale al problema della nostra filigrana. Abbiamo bisogno di macro per l'utente per specificare dimensioni e colori della filigrana e per specificare l'angolo di inclinazione della retta sulla quale essa sembra “infilzata”:

```

1 % Pacchetti su cui appoggiarsi
2 \usepackage{graphicx}
3 \usepackage{xcolor}
4 \usepackage{pict2e}[2009/06/01]
5 % Impostazioni predefinite:
6 \def\Draftname{Draft}\def\Draftangle{55}
7 \newlength{\Draftheight}\setlength\Draftheight{43mm}
8 \definecolor{Draftcolor}{gray}{0.8}
9 % Comandi per l'utente
10 \newcommand\AltezzaBOZZA[1]{\setlength\Draftheight{#1}}
11 \newcommand\BOZZA[1]{\gdef\Draftname{#1}}
12 \newcommand\AngoloBOZZA[1]{\xdef\Draftangle{#1}}
13 \newcommand\ColoreBOZZA[3]{\definecolor{#1}{#2}{#3}}
```

Poi occorre una macro per comporre la filigrana nell'ambiente `picture` di dimensioni nulle:

```

1 \newcommand\BOZZA{%
2 \begin{picture}(0,0)\unitlength\p@
3 \put(0,-\strip@pt\headsep){%
4 \put(-\strip@pt\dimexpr\textwidth/2\relax,
5 -\strip@pt\dimexpr\textheight/2\relax){%
6 \makebox(0,0){\normalfont
7 \fontsize{\Draftheight}{\Draftheight}\selectfont
8 \rotatebox{\Draftangle}{\color{Draftcolor}
9 \Draftname}}}
9 \end{picture}}
```

Questa macro contiene un certo numero di “sporchi trucchi”, tipo quelli descritti da Knuth nel suo

T_EXbook, (KNUTH, 1996). In realtà l'unico trucco che non è descritto nel manuale di Lamport è l'uso del comando `\strip@pt` definito nel nucleo di L^AT_EX; esso agisce sul valore esplicito di una dimensione e gli "strappa" via le unità di misura 'pt'. La 'dimensione' può anche essere una espressione dimensionale e il moderno *pdf_{te}x* (il motore di composizione di L^AT_EX) dispone di questi comandi dimensionali dal 2005, quindi se non sono tanto conosciuti è solo dovuto all'inerzia di molti utenti, che preferiscono usare il pacchetto *calc* (KRAB THORUP *et al.*, 2007) che fa esattamente le stesse cose ricorrendo a medesimi comandi. Di solito usare *calc* o `\dimexpr` non comporta alcun risparmio di tempo o di spazio nello scrivere il file di macro. Usare il comando primitivo del linguaggio ϵ -T_EX (BREITENLOHNER, 1998) consente un risparmio di tempo rispetto a medesimi calcoli eseguiti con *calc* ma stiamo parlando di microsecondi, quindi di limitata importanza. La sequenza di controllo `\p@` indica una lunghezza di 1 pt mentre il comando `\fontsize` è disponibile anche per l'utente per specificare un corpo particolare del font e il suo corrispondente avanzamento di riga (L^AT_EX3 PROJECT TEAM, 2005). In questo caso scrivendo una sola parola non dobbiamo preoccuparci dell'interlinea.

Tuttavia questo modo di procedere nasconde un'esigenza ben particolare. La famiglia di font su cui operare deve essere continuamente scalabile, altrimenti il gioco non funziona. Tra i font predefiniti distribuiti con il sistema T_EX ci sono moltissimi font di tipo Type 1 continuamente scalabili³ ma fra quelli che si rifanno ai font originali del sistema T_EX, come i Computer Modern, i CM-super e altri, nessuno è continuamente scalabile a meno che non venga caricato l'uno o l'altro dei pacchetti *type1cm* (codifica OT1) oppure *type1ec* (codifica T1) che, eliminando la scelta dei corpi ottici⁴, consentono lo scalamento continuo. Agire in questo modo sarebbe poco funzionale perché esistono i font Latin Modern, molto simili ai font Computer Modern, che, a differenza di questi, sono continuamente scalabili pur conservando i corpi ottici.

Il codice precedente mostra come il comando `\B@ZZA`, che contiene il disegno di dimensioni nulle sia in larghezza sia in altezza e profondità, può essere messo all'estremità destra delle testatine sia di destra che di sinistra. Il primo comando

3. I font di tipo Type 1 sono font vettoriali ed hanno il vantaggio che possono venire ingranditi, distorti, stirati in mille modi, ma comunque resi in modo impeccabile sia su schermo, sia a stampa.

4. Uno dei vantaggi dei font originali del sistema T_EX è che sono fra le poche collezioni di font che dispongono di una vasta serie di corpi ottici, cioè di font affini nello stile grafico, ma disegnate in modo tale che i corpi più piccoli non perdano i loro dettagli più minuti mentre i corpi più grandi non risultino troppo neri. Questo è un enorme vantaggio, specialmente quando si devono usare molti apici e pedici come in matematica.

`\put` sposta in basso il riferimento esattamente di `\headsep`, quindi lo abbassa allo spigolo superiore destro dello specchio di stampa; il successivo `\put` pone il suo contenuto in basso e a sinistra di mezza altezza e di mezza larghezza dello specchio di stampa, quindi esattamente al centro dello specchio. Il contenuto di questo comando `\put`, a sua volta, è una scatola di dimensioni nulle il cui contenuto viene esattamente centrato in altezza e in larghezza. Questo contenuto, la lettura è chiara, è costituito dal testo specificato con `\Draftname`, colorato come specificato dal colore `Draftcolor`, composto con il font di corpo pari a `\Draftheight` e ruotato dell'angolo `\Draftangle`.

Quindi ora non abbiamo più problemi con la filigrana, ma dobbiamo mettere il comando che la produce dentro le testatine con i vari stili di pagina. Se l'utente usasse il pacchetto *fancyhdr*, l'impostazione delle testatine sarebbe più semplice, ma pensando ad un file di macro personali va benissimo ricorrere a comandi di basso livello:

```

1 \let\ori@ps@empty\ps@empty
2 \let\ori@ps@plain\ps@plain
3 \let\ori@ps@headings\ps@headings
4
5 \renewcommand\ps@plain{\ori@ps@plain
6 \def\@oddhead{\null\hfill\B@ZZA}%
7 \let\@evenhead\@oddfoot}
8
9 \renewcommand\ps@empty{\ori@ps@empty
10 \def\@oddhead{\null\hfill\B@ZZA}%
11 \let\@evenhead\@oddfoot}
12
13 \renewcommand\ps@headings{\ori@ps@headings
14 \toks@=\expandafter{\@oddhead}%
15 \edef\@oddhead{\the\toks@\noexpand\B@ZZA}%
16 \toks@=\expandafter{\@evenhead}
17 \edef\@evenhead{\the\toks@\noexpand\B@ZZA}}
```

Ho esplicitato solo i tre stili di pagina più frequentemente usati; dentro il nucleo di L^AT_EX e nei file di classe questi stili sono identificati dal loro nome aggiunto al prefisso `\ps@`. Basta quindi memorizzare gli stili originali mediante dei comandi "sinonimi" (gli stessi nomi con il prefisso `\ori@ps@`) e poi ridefinirli in modo che contengano le testatine con l'aggiunta del comando `\B@ZZA` all'estremità destra.

Per gli stili *empty* e *plain* basta aggiungere le definizioni opportune. Per lo stile *headings* (e lo stesso bisognerebbe fare per ogni altro stile che contenga qualcosa nelle testatine) bisogna ricorrere allo "sporco trucco" di memorizzare la definizione (il testo sostitutivo) delle definizioni delle testatine in un registro "token", il registro numero 'zero', che dispone di un nome particolare `\toks@`. Quindi ridefinire ciascun comando per le testatine pari o dispari espandendo il registro token ma senza espandere il comando `\B@ZZA`. Lo "sporco trucco" può essere semplicemente evitato usando il pacchetto *etoolbox* che definisce i comandi `\preto` e `\addto` per aggiungere del testo a quello contenuto nella definizione di una macro rispettivamente

prima o dopo il testo preesistente. Ma qui non lo facciamo perché le condizioni che ci siamo dati non consentono di usare comandi di altri pacchetti, se non di quelli specificamente elencati.

Infine, affinché le modifiche agli stili di pagina abbiano luogo, bisogna eseguire almeno il comando per lo stile di default:

```
\pagestyle{headings}
```

Ora basta creare un file di prova che carichi il file `filigrana.sty` oltre agli altri pacchetti necessari.

```
1 \documentclass[a4paper,12pt]{book}
2 \usepackage[utf8]{inputenc}
3 \usepackage[T1]{fontenc}
4 \usepackage{lmodern}
5 \usepackage[italian]{babel}
6 %\usepackage{canoniclayout}
7 %\usepackage{layaureo}
8 \usepackage{filigrana}
9 \usepackage{lipsum}
10 \begin{document}\Huge
11
12 \chapter{Prima pagina in stile plain}
13   \thispagestyle{plain}
14
15 Le pagine successive sono in stile headings.
16
17 \lipsum[1-5]
18
19 \lipsum[6-20]
20
21 \chapter{Pagine classificate}
22 \BOZZA{CLASSIFIED}
23
24 \lipsum[5-25]
25 \end{document}
```

Salvato questo file, ad esempio come `BOZZA.tex`, si può vedere che cosa succede con il layout di pagina della classe `book` oppure con il layout che si ottiene con il pacchetto `canoniclayout` oppure con il pacchetto `layaureo`. Si noterà ad occhio nudo che la filigrana ha bisogno di un aggiustamento per l'inclinazione perché le diagonali degli specchi di stampa nei tre casi sono inclinate in modo diverso (vedi più avanti la figura 7).

L'inclinazione non è tuttavia un problema, basta misurare l'altezza e la larghezza di uno specchio di stampa in una pagina sullo schermo o stampata su carta. L'angolo da introdurre mediante il comando `\AngoloBOZZA` sarà l'arcotangente in gradi sessagesimali del quoziente altezza diviso larghezza. Per molti lettori è dai tempi delle scuole secondarie che non capita di calcolare un'arcotangente, ma oggi ogni calcolatrice elettronica tascabile ha il tasto per eseguire questo calcolo. In mancanza di tale strumento c'è sempre l'applicazione/programma *Calculator* in ogni computer, che consente di eseguire il calcolo senza alzarsi dalla scrivania. D'altra parte questo calcoletto è da fare una sola volta per l'intero documento, non è una cosa da ripetere in ogni pagina, per cui il "disturbo" di doverlo calcolare a mano è veramente trascurabile. Il risultato

di questo codice, con l'angolo corretto è mostrato nella figura 3.

5 Prima soluzione automatica

È possibile lasciare fare i calcoli al programma *pdf-tex*? Ebbene sì, è possibile ma con qualche acrobazia. Infatti, tranne il modulo per eseguire calcoli del pacchetto *TikZ*, non ho trovato nessun altro pacchetto che sia in grado di calcolare l'arcotangente precedentemente eseguita a mano.

Non possiamo chiamare il modulo dei calcoli di *TikZ*, perché sarebbe vietato dalle condizioni del concorso. D'altra parte non abbiamo bisogno di fare una grande elaborazione per calcolare l'arcotangente di un angolo che sicuramente giace nel primo quadrante; al massimo potremmo calcolare l'arcocotangente (cioè l'arcotangente dell'angolo complementare) per poter mantenere i calcoli con una precisione decente. Inoltre la procedura di *TikZ* fa riferimento ad una tabella di un migliaio di valori per calcolare l'angolo di cui è data la tangente; usa metodi di interpolazione che ritengo siano inutilmente complicati per replicarli qui in forma semplificata.

Il concorso ha messo in luce l'inventiva dei partecipanti. In particolare Roberto Giacomelli ha fornito una soluzione basata sull'uso delle funzioni disponibili per eseguire calcoli in virgola mobile contenuti nella libreria `13fp.sty` che accompagna la distribuzione del linguaggio sperimentale messo a punto dal *L^AT_EX 3 Team*. Questa libreria sperimentale, già stabile, fa parte di ogni distribuzione aggiornata del sistema *T_EX* in quanto parte integrante di *X_YL^AT_EX* e ad essa molti pacchetti si appoggiano sistematicamente; non sempre fanno parte di *pdfL^AT_EX*, ma ora cominciano ad essere presenti diversi pacchetti basati su questo linguaggio anche per *pdfL^AT_EX*.

Non riporto il codice della soluzione trovata da Giacomelli, che comunque è reperibile nel filone di discussione citato a proposito del concorso, ma voglio esporne i principi, molto interessanti ed efficaci.

Premetto che la libreria `13fp.sty` definisce già una serie di funzioni tra le quali quelle trigonometriche seno, coseno e tangente, oltre alla radice quadrata di cui parleremo più avanti. Non contiene le definizioni delle funzioni trigonometriche inverse, quindi anche con quella libreria bisogna arrangiarsi.

Giacomelli ha usato il metodo delle tangenti, è un semplice metodo iterativo che viene usato spessissimo per risolvere equazioni per le quali non è disponibile una formula risolutiva esplicita. Si parte da un valore approssimato della soluzione e con un formula iterativa, che illustrerò fra poco, si migliora l'approssimazione. Ovviamente la procedura iterativa dell'algoritmo può essere applicata solo un numero finito di volte. I motori di composi-

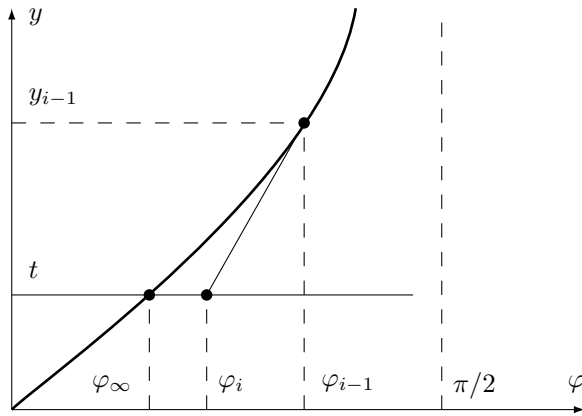


FIGURA 2: Metodo delle tangenti

zione del TEX lavorano a virgola fissa e dispongono di 16 cifre binarie fratte, corrispondenti a circa 5 cifre decimali fratte. Quindi è perfettamente inutile iterare l'algoritmo delle tangenti oltre a questo limite per cercare di ottenere valori approssimati migliori: sarebbe impossibile. Anzi, con una simile macchina dalle prestazioni di calcolo modeste è già un buon risultato raggiungere una precisione di tre o quattro cifre decimali esatte.

Il metodo delle tangenti è basato sul disegno della figura 2 e mostra quanto lavoro sarebbe richiesto per determinare l'angolo φ_∞ la cui tangente vale t . Nel nostro problema questa tangente è data dal quoziente fra l'altezza e la larghezza dello specchio di stampa, quindi l'assumiamo come quantità nota. Dobbiamo risolvere allora l'equazione:

$$\varphi_\infty = \arctan t$$

ovvero:

$$\tan \varphi_\infty = t \quad (1)$$

Facendo riferimento alla figura 2 dobbiamo trovare l'intersezione della curva della tangente con la retta orizzontale $y = t$. Supponiamo di essere partiti da un'approssimazione dell'ascissa del punto di intersezione φ_0 e dopo un certo numero di iterazioni $i - 1$ di essere arrivati all'ascissa φ_{i-1} dove però la funzione tangente vale y_{i-1} e non vale ancora t , come richiesto dall'equazione (1). Possiamo tuttavia determinare la pendenza della funzione tangente nel punto di coordinate (φ_{i-1}, y_{i-1}) impiegando la trigonometria "elementare"⁵ per cui la pendenza della retta tangente alla curva in quel punto vale:

$$y'_{i-1} = y'(\varphi_{i-1}) = \frac{1}{\cos^2 \varphi_{i-1}} \quad (2)$$

5. La parola "elementare" è racchiusa fra virgolette e molti lettori potrebbero non conoscere nemmeno la trigonometria. Coloro che hanno fatto il liceo scientifico o alcuni indirizzi degli istituti tecnici potrebbero confermare che si tratta di un concetto elementare, ma gli altri? Non è il caso di spaventarsi: chi non conosce la trigonometria e le basi del calcolo differenziale prenda quanto segue con leggerezza e scoprirà che non è difficile da seguire anche senza le basi teoriche.

Possiamo allora intersecare questa retta tangente, di cui conosciamo un punto e l'inclinazione in quel punto, con la retta orizzontale $y = t$ che è il nostro obiettivo e determinare la nuova ascissa φ_i che risulta più vicina a φ_∞ di quanto non lo fosse φ_{i-1} :

$$\varphi_i = \varphi_{i-1} - \cos^2 \varphi_{i-1} (\tan \varphi_{i-1} - t) \quad (3)$$

Possiamo quindi ripetere la procedura iterativamente, incrementando i ad ogni iterazione, fino a quando, a causa delle limitazioni di calcolo della macchina, il valore aggiornato φ_i ottenuto smette di avvicinarsi a φ_∞ oppure quando la precisione del valore aggiornato è ritenuta soddisfacente, per esempio:

$$|\varphi_i - \varphi_{i-1}| < 0.001$$

In definitiva l'algoritmo potrebbe essere:

Inizializza $i = 1$ e $\varphi_0 = \pi/4$

finché $|\varphi_i - \varphi_{i-1}| > 0.001$

ripeti

$$\varphi_i = \varphi_{i-1} - \cos^2 \varphi_{i-1} (\tan \varphi_{i-1} - t)$$

$$i \leftarrow i + 1$$

fine

Giacomelli assicura che bastano tre o quattro iterazioni per ottenere l'angolo la cui tangente è assegnata e, pur non avendole conteggiate, non ho dubbi sulla correttezza dell'affermazione in quanto tipicamente il metodo delle tangenti converge quadraticamente e quindi ad ogni iterazione le cifre esatte possono raddoppiare.

Mi pare giusto descrivere questo algoritmo senza riportarne il codice perché il linguaggio di LATEX 3 è sperimentale, ancora non molto conosciuto e, comunque, non fa ancora parte integrale del nucleo di pdfLATEX, ma solo di alcuni pacchetti. Inoltre i concetti elementari di trigonometria che ho esposto non sono familiari a tutti i lettori, quindi anche il codice per realizzare l'algoritmo presenterebbe per loro delle difficoltà in più.

Mi sono rivolto invece all'algoritmo delle frazioni continue che serve per eseguire certi calcoli numerici in modo molto semplice usando solo numeri interi. Ho trovato che il valore di $\pi/4$ è calcolabile con la precisione consentita dai mezzi di *pdf*tex

mediante la seguente frazione continua:

$$\begin{aligned}\frac{\pi}{4} &= 0.7853981634 \\ &= \frac{1}{1 + 0,27323954474} \\ &= \frac{1}{1 + \frac{1}{3 + 0,65979236633}} \\ &= \frac{1}{1 + \frac{1}{3 + \frac{1}{1 + 0,51562832648}}} \\ &= \dots\end{aligned}$$

che va proseguita fino ad avere tante frazioni quante sono le cifre esatte desiderate.

Ne segue che:

$$\frac{\pi}{4} = \arctan 1 = \cfrac{1}{1 + \cfrac{1}{3 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{1 + \cfrac{1}{15 + \cfrac{1}{2 + \cfrac{1}{73}}}}}}}}} \quad (4)$$

Ho allora pensato di ottenere un'approssimazione dell'arcotangente sostituendo gli '1' nei vari numeratori con il valore della tangente di cui calcolare l'arcotangente, in modo tale che l'approssimazione ottenuta sia ancora una funzione dispari come lo è la funzione arcotangente. Pertanto, siccome $[\arctan(t)]/t$ deve essere una funzione pari, l'equazione (4) porta a:

$$\arctan t \approx \cfrac{t}{1 + \cfrac{t^2}{3 + \cfrac{t^2}{1 + \cfrac{t^2}{1 + \cfrac{t^2}{1 + \cfrac{t^2}{15 + \cfrac{t^2}{2 + \cfrac{t^2}{73}}}}}}}}} \quad (5)$$

Il procedimento che ho seguito è tutt'altro che rigoroso ed è molto empirico tuttavia nell'intervallo

2

CAPITOLO 1. PRIMA PAGINA IN STILE PLAIN

stibulum urna fringilla ultrices. Phaselus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis.

FIGURA 3: Filigrana inserita con la procedura manuale descritta nel paragrafo 4

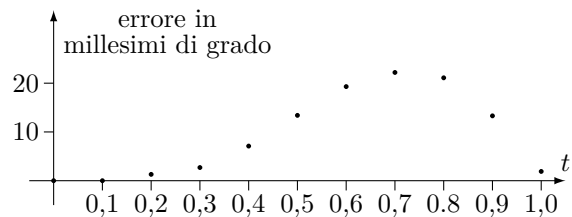


FIGURA 4: Errore fra la funzione approssimante e la funzione reale

$0 \leq t \leq 1$ l'errore di approssimazione è rappresentato nella figura 4. Come si vede il massimo errore di approssimazione è pari a 22,2 millesimi di grado, corrispondenti a circa 0,4 millesimi di radiante – un errore perfettamente accettabile.

Il procedimento è empirico ma è sufficientemente preciso ed è inutile spingersi oltre. Quando $t = 1$ si dovrebbe ritrovare il valore di 45° ma invece c'è un errore di 1,9 millesimi di grado, da dove arriva? Dall'aritmetica interna di *pdftex* che è a "virgola" fissa con 16 cifre binarie fratte e sulle quali si accumulano gli errori di troncamento e di arrotondamento. Raramente si può pensare o sperare che calcoli eseguiti con l'aritmetica interna di *pdftex* possano produrre errori inferiori. Nella fattispecie l'errore indicato deriva dall'accumulazione degli errori sulle ultime sette cifre binarie fratte e ce ne sono ancora nove su cui contare, per cui un errore dell'ordine del millesimo è del tutto accettabile. Ragionando in radianti il massimo errore di 22,2

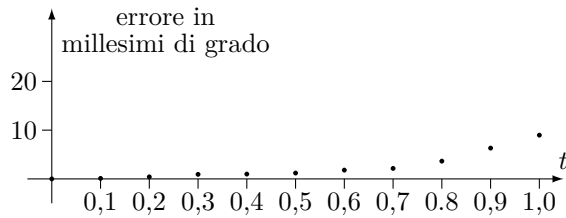


FIGURA 5: Errore assoluto fra la forma di Padé di grado 5 e la funzione reale

millesimi di grado corrisponde ad un errore inferiore a 0,4 millesimi di radiante, dello stesso ordine di grandezza di quello accettabile con il metodo iterativo descritto in precedenza.

Al di là di queste considerazioni di calcolo numerico, è ovvio che un errore di pochi millesimi di grado sull'inclinazione della filigrana passa del tutto inosservato per l'applicazione proposta.

Ma per chi non si accontentasse della semplice evidenza empirica è disponibile una teoria più solida basata sulle forme di Padé (CARRERI, 2005). Queste sono approssimazioni razionali di serie di potenze troncate a quozienti polinomiali che generalmente hanno numeratore e denominatore formati da polinomi di grado diverso; ma quest'ultima, però, non è una condizione essenziale. Chi usa queste approssimazioni può scegliere liberamente i gradi del numeratore e del denominatore; ad esempio con polinomi pari di grado 4 l'approssimazione della funzione dispari arcotangente può essere ricavata nella forma:

$$\frac{\arctan t}{t} \approx \frac{1 + \frac{7}{9}t^2 + \frac{64}{945}t^4}{1 + \frac{10}{9}t^2 + \frac{5}{21}t^4}$$

Tenuto conto che l'aritmetica interna di *pdfTeX* (e anche di *x_εLaTeX*) non è molto precisa, è meglio riscrivere l'espressione in una forma tale da minimizzare il numero di operazioni da eseguire:

$$\arctan t \approx t \frac{945 + 735t^2 + 64t^4}{945 + 1050t^2 + 225t^4} \quad (6)$$

La precisione di questo algoritmo è riportata nella figura 5. Come si vede l'errore assoluto della differenza tra la funzione approssimante e la funzione reale è molto minore dell'errore ottenuto con la frazione continua (5). Questo prova, se mai ci fosse qualche dubbio, che un problema matematico risolto con i metodi della matematica dà risultati migliori che non con metodi empirici. Non trascuriamo la possibilità di usare anche la frazione continua (5), ma in questo caso sarebbe stupido farlo.

Ciò premesso, il problema si riduce a creare il software per calcolare automaticamente l'angolo della diagonale rispetto all'orizzontale. Poiché l'errore dell'approssimazione tende ad aumentare con il valore della tangente, riduciamo gli angoli

al primo ottante e quindi a valori della tangente non superiori all'unità. Le formule trigonometriche ci aiutano nello svolgere questo compito e ne teniamo conto nel codice, infatti se la tangente è maggiore dell'unità ne prendiamo il reciproco, calcoliamo l'angolo che risulta inferiore a 45°, poi ne calcoliamo il valore cercato come complemento a 90°:

```
1 % Definizione di alcuni registri lunghezza e di
  costanti
2 \newlength{\tang}
3 \newlength{\tanquad}
4 \newcommand{\gradrad}{57.2957795130823}% gradi al
  radiante
5 \newlength{\arctang}
6 \ifx\undefined\@tdG \newdimen\@tdG \fi %
  numeratore
7 \ifx\undefined\@tdH \newdimen\@tdH \fi %
  denominatore
8 \ifx\undefined\I \newcount\I \fi % contatore
  booleano
9 % Inizializzazione
10 \setlength\tang{\dimexpr\p@*\textwidth/\textheight
  \relax}% tangente
11 \setlength\tanquad{\dimexpr\tang*\tang/\p@
  \relax}% t^2
12 % Calcolo della approssimante di Pade'
13 \ifdim\tang>\p@
14 \I=\@ne % arcotangente > 1
15 \tang\dimexpr\p@*\p@/\tang\relax% cotangente
16 \else
17 \I=\@z % arcotangente <= 1
18 \fi
19 \tanquad=\dimexpr\tang*\tang/\p@\relax
20 %
21 \@tdG = \dimexpr 64\tanquad+735\p@\relax
22 \@tdG = \dimexpr \strip@pt\@tdG \tanquad + 945\p@
  \relax % numeratore senza t
23 \@tdH = \dimexpr 225 \tanquad + 1050\p@ \relax
24 \@tdH = \dimexpr \strip@pt\@tdH \tanquad + 945\p@
  \relax % denominatore
25 \arctan = \gradrad\dimexpr\tang*\@tdG/\@tdH\relax
26 \ifnum\I>0
27 \arctan=\dimexpr90\p@ - \arctan\relax%
  complemento a 90 gradi
28 \fi
29 % Inizializzazione dell'angolo di rotazione
30 \AngoloBOZZA{\strip@pt\arctan}%
31
```

Con l'ultimo comando viene inizializzato l'angolo di rotazione della filigrana e quindi, oltre all'aggiunta di queste poche righe di codice, il contenuto del file *filigrana.sty* diventa capace di ruotare la filigrana dell'angolo corretto. La figura 3 di fatto continua a presentare la filigrana collocata correttamente in posizione e queste poche righe di codice risparmiano all'utente dal doversi calcolare a mano l'arcotangente dell'angolo di rotazione. Non solo, il codice tratta correttamente sia il caso della gabbia di stampa in verticale (*portrait*), sia quello della gabbia panoramica (*landscape*).

6 Seconda soluzione avanzata

Le soluzioni presentate sopra richiedono l'uso del comando *\rotatebox* del pacchetto *graphicx*. Se ne può fare a meno?

La risposta è sì, ma bisogna “sporcarsi le mani” con comandi di bassissimo livello del linguaggio di descrizione della pagina specificato dal formato PDF (THÀNH *et al.*, 2010).

In questo formato vengono eseguite delle operazioni grafiche che vengono svolte dai comandi del linguaggio PDF inseriti nel file finale PDF dai driver d’interfaccia che l’utente usa quando compone con *pdf_latex*. Il motore di resa grafica per visualizzare o per stampare il documento comprende solo il linguaggio PDF.

La dilatazione e la contrazione, eventualmente diverse nelle due direzioni, la rotazione e la traslazione costituiscono delle trasformazioni affini e vengono controllate da una matrice di trasformazione che ora verrà brevemente illustrata. Naturalmente bisogna anche essere certi che definendo una certa matrice di trasformazione, questa non rimanga per sempre in vigore ma piuttosto esaurisca il suo compito appena eseguita la trasformazione dell’oggetto. In altre parole, la matrice di trasformazione predefinita corrisponde alla matrice identità e il vettore di traslazione è nullo; al termine di ogni diversa trasformazione è necessario ripristinare la matrice predefinita.

Qui siamo interessati ed eseguire solo la rotazione di un angolo φ e, geometricamente parlando, nel nostro caso la ‘matrice’ di trasformazione che usa il linguaggio PDF è semplicemente data da

$$\begin{pmatrix} \cos \varphi & -\sin \varphi & 0 \\ \sin \varphi & \cos \varphi & 0 \end{pmatrix} \quad (7)$$

dove l’ultima colonna rappresenta il vettore di traslazione.

Il comando `\rotatebox` genera i due valori del seno e del coseno di φ . Se conosciamo i valori del seno e del coseno di tale angolo, conservati nelle macro `\seno` e `\coseno`, basta far precedere l’oggetto da ruotare dai comandi in linguaggio PDF introdotti mediante il comando `\pdfliteral` in questo modo⁶:

```
1 \pdfliteral{ q \coseno\space \seno\space
2             -\seno\space \coseno\space
3             0 0 cm}
```

e poi far seguire l’oggetto da ruotare dal comando di ripristino della matrice di trasformazione preesistente:

```
1 \pdfliteral{ Q}
```

In entrambi gli argomenti gli spazi indicati come tali o esplicitati con il token implicito `\space` sono essenziali per il linguaggio PDF. Il comando del linguaggio base `q` serve per memorizzare l’attuale

6. I comandi gestiti dal programma di composizione *pdf_latex* includono anche i comandi `\pdfsave`, `\pdfsetmatrix` e `\pdfrestore` che separano le tre azioni qui descritte e impongono solo la matrice di rotazione senza dover preoccuparsi della matrice di traslazione. Dovendo scrivere un codice conviene usare i comandi più compatti forniti dall’unico comando `\pdfliteral` e con gli argomenti appositi.

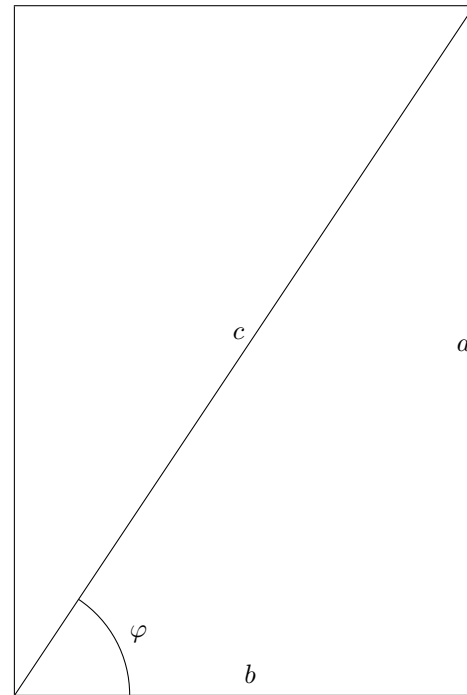


FIGURA 6: Lo specchio di stampa, la sua diagonale e l’angolo fra la diagonale e la base

matrice di trasformazione, i sei elementi successivi sono la ‘matrice’ scritta per colonne; il comando `cm` serve per dichiarare la precedente sequenza di valori come la “current matrix”, la matrice corrente di trasformazione; il comando di ripristino `Q` ripristina la matrice di trasformazione precedentemente memorizzata.

Perciò, non usando il comando `\rotatebox`, dobbiamo ridefinire la macro che costruisce la filigrana. La definizione non è molto diversa da quella che abbiamo già usato con `\rotatebox`, solo che ora dobbiamo inserire prima e dopo la scatola che contiene la filigrana i comandi in linguaggio PDF che definiscono la matrice di rotazione:

```
1 \newcommand\B@ZZA{%
2 \begin{picture}(0,0)\unitlength\p@
3 \put(0,-\strip@pt\headsep){%
4 \put(-\strip@pt\dimexpr\textwidth/2\relax,
5 -\strip@pt\dimexpr\textheight/2\relax){%
6 \pdfliteral{ q \coseno\space \seno\space
7 -\seno\space \coseno\space 0 0 cm}%
8 \setbox\z@\hbox{\makebox(0,0){\normalfont
9 \fontsize{\Drafterheight}{\Drafterheight}\selectfont
10 \color{Draftcolor}\Draftname}}\box\z@
11 \pdfliteral{ Q}}
12 \end{picture}}
```

Nel paragrafo precedente sono dovuto ricorrere a calcoli non semplici per determinare il valore dell’angolo φ a partire dall’altezza e dalla larghezza dello specchio di stampa; non si può evitare tutto ciò? Visto che poi `\rotatebox` deve determinare il seno e il coseno dell’angolo di rotazione, non sarebbe preferibile calcolare subito tali valori a partire dalle dimensioni dello specchio di stampa, invece di tribolare con la formula approssimata

dell'arcotangente? Certamente. Infatti detti a e b rispettivamente l'altezza e la base dello specchio di stampa, come si vede nella figura 6, basta calcolare le grandezze seguenti:

$$a = \text{textheight} \quad (8)$$

$$b = \text{textwidth} \quad (9)$$

$$c = \sqrt{a^2 + b^2} \quad (10)$$

$$\cos \varphi = \frac{b}{c} \quad (11)$$

$$\sin \varphi = \frac{a}{c} \quad (12)$$

Ora però siamo di fronte ad un altro piccolo problema, bisogna far calcolare a *pdf_{tex}* una radice quadrata senza usare altri pacchetti.

Ho già ricordato che la libreria *l3fp.sty* contiene le macro in linguaggio LATEX 3 per eseguire calcoli con valori numerici a virgola mobile compreso il calcolo per la radice quadrata e ho già spiegato perché non vorrei ricorrere a quella libreria mentre uso pdfLATEX. Ma per il calcolo della radice quadrata l'algoritmo è così semplice che programmarlo con le espressioni di ϵ -TEX, facenti parte di pdfLATEX, non è assolutamente difficile.

Infatti la radice quadrata può venire calcolata con l'algoritmo di Newton che ha una convergenza quadratica, quindi è molto veloce e preciso; esso si basa sul calcolo di una sequenza di valori sempre meglio approssimati ricavati mediante una formula ricorsiva: sia R il radicando e sia r_i l' i -esima approssimazione della radice di questa sequenza; sia r_0 il primo elemento con cui iniziare i calcoli. La formula ricorsiva diventa la seguente:

$$r_{i+1} = \frac{r_i + R/r_i}{2} \quad \text{per } i = 0, 1, 2, 3, \dots \quad (13)$$

Se il valore iniziale è scelto con accuratezza, il procedimento ricorsivo (13) raddoppia le cifre esatte ad ogni iterazione. A questo scopo, visto che l'altezza dello specchio di stampa è (quasi) sempre maggiore della base, allora (quasi) sempre è $x = b/a \leq 1$. Possiamo perciò riscrivere l'espressione (10) nella forma

$$c = a\sqrt{1 + x^2} \quad (14)$$

e scegliere:

$$r_0 = 1 + \frac{x^2}{2} \quad (15)$$

per iniziare le iterazioni per il calcolo della radice.

Facciamo due conti di massima. Per uno specchio di stampa piuttosto tozzo come quello ottenibile con uno specchio di stampa dalle proporzioni simili a quelle della carta in formato "letter", il rapporto x varrebbe circa 0,773, poco più di $3/4 = 0,75$; quindi il valore esatto di $\sqrt{1 + 0,75^2}$ sarebbe 1,25. Il valore r_0 con cui iniziare le iterazioni sarebbe 1,28125, come si vede molto vicino al valore finale della successione. Invece, con uno specchio di

stampa molto allungato con un rapporto $x = 0,5$, la radice esatta di $\sqrt{1 + 0,5^2}$ sarebbe 1,1180... mentre il valore iniziale di r_0 sarebbe 1,125; anche in questo caso molto vicino al valore finale della successione.

Ecco allora che posso stendere il codice per determinare il seno e il coseno dell'angolo di rotazione in questo modo⁷:

```

1 % Definizione di alcuni registri lunghezza
2 \newlength\rappcateti
3 \newlength\radicando\newlength\radice
4 \ifx\I\undefined\newcount\I\fi
5 % Macro per il calcolo del seno e del coseno
6 \newcommand*\sincos{%
7 \rappcateti=\dimexpr\textwidth*\p@/\textheight
8 \relax % b/h
9 % Calcolo del radicando e di di r_0
10 \radicando=\dimexpr\rappcateti*\rappcateti/\p@
11 + \p@\relax
12 \radice=
13 \dimexpr\p@+(\rappcateti*\rappcateti/\p@)/2
14 \relax
15 % Formula di Newton con 6 iterazioni
16 \I=0
17 \@whilenum\I<6\do{%
18 \radice=
19 \dimexpr(\radice+\radicando*\p@/\radice)/2
20 \relax
21 \advance\I\@ne}%
22 % Calcolo dell'ipotenusa
23 \radice=\dimexpr\radice*\textheight/\p@\relax
24 % Trovata l'ipotenusa ora si calcolano seno e
25 coseno
26 \edef\seno{%
27 \strip@pt\dimexpr\textheight*\p@/\radice}%
28 \edef\coseno{%
29 \strip@pt\dimexpr\textwidth*\p@/\radice}%
30 }
31 % Si esegue ora la macro in modo da definire
32 % effettivamente i valori del seno e coseno
33 \sincos

```

Ma se volessi fissare a mano il valore dell'angolo di rotazione per avere una filigrana centrata, sì, nel centro dello specchio di stampa, ma ruotata di un angolo diverso rispetto alla sua diagonale? Avrei bisogno di un comando *\AngoloBOZZA* simile a quello definito in precedenza ma che *non* conservi soltanto il valore di questo angolo in una opportuna macro, ma che provveda a calcolare anche il seno e il coseno necessari alla rotazione. Ho dunque bisogno di un comando che accetti un argomento facoltativo. Se questo manca, allora il seno e il coseno necessari vengono determinati in base alle dimensioni dello specchio di stampa, con l'algoritmo che ho appena descritto. Se invece l'argomento facoltativo viene specificato, allora il seno e il coseno necessari vengono calcolati a partire da questo valore. Per comodità dell'utente, l'an-

7. Nella macro qui riportata le iterazioni sono 6; probabilmente è eccessivo, e un valore come 3, o meglio 4, sarebbe più che sufficiente; tuttavia i calcoli eseguiti con *\dimexpr* sono velocissimi perché vengono svolti nel "processore" del calcolatore e non impegnano la scrittura sul disco, se non per i risultati finali di ogni iterazione; quindi un paio di iterazioni in più non danno nessun fastidio.

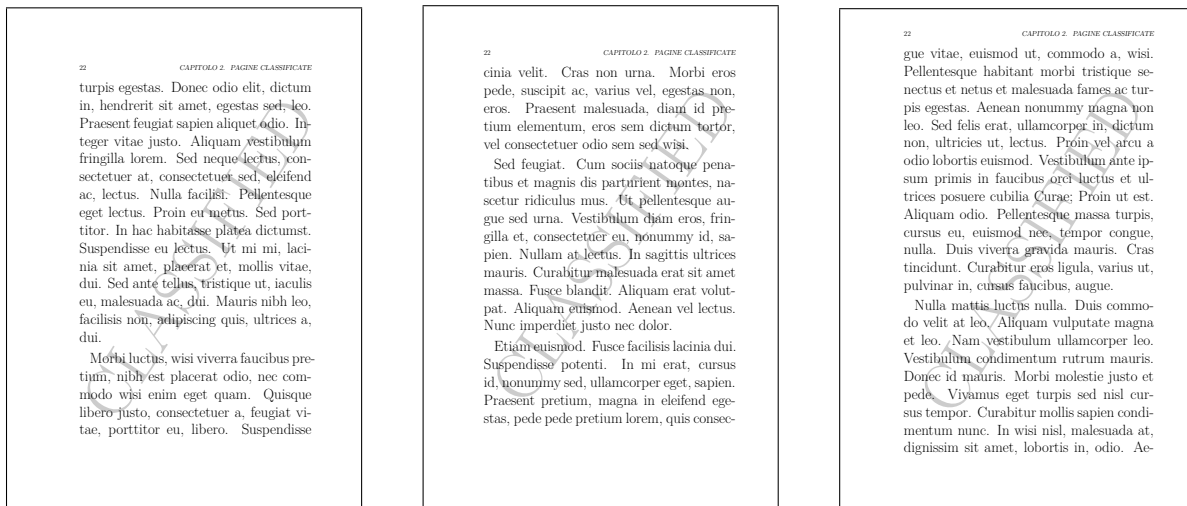


FIGURA 7: Tre layout di pagina diversi con le relative filigrane

golo facoltativo deve venire specificato in gradi sessagesimali.

Il comando in questione va definito con la sintassi $\LaTeX$ per i comandi con argomenti facoltativi in questo modo:

```
1 \newcommand\AngoloB0ZZA[1] []{\def\@tempA{#1}%
2 \ifx\@tempA\empty\sincos\else\FIL@senocoseno{#1}
\fi}
```

Il valore di default dell'argomento facoltativo è predefinito come “vuoto” e lo sviluppo del comando prevede inizialmente la verifica che tale argomento sia effettivamente vuoto. Se lo è, esegue il comando $\backslash\text{sincos}$ che determina il seno e il coseno necessari per la rotazione a partire dai lati dello specchio di stampa. Se non lo è, provvede a determinare il seno e il coseno richiesti mediante un algoritmo abbastanza semplice: determina numericamente il (reciproco del) valore in radianti della metà dell'angolo specificato, poi calcola la tangente, la cotangente e infine determina il seno e il coseno mediante le formule parametriche:

$$\sin \varphi = \frac{2}{\cot \varphi/2 + \tan \varphi/2} \quad (16)$$

$$\cos \varphi = \frac{\cot \varphi/2 - \tan \varphi/2}{\cot \varphi/2 + \tan \varphi/2} \quad (17)$$

I calcoli per determinare il seno e il coseno richiesti sono semplici, ma tutto sta a vedere come si calcola la tangente del semiangolo (in radianti). Posto:

$$x = \frac{57,2957795}{\varphi/2} \quad (18)$$

la frazione continua che dà la tangente richiesta è

data da:

$$\tan \varphi/2 = \frac{1}{x - \frac{1}{3x - \frac{1}{5x - \frac{1}{7x - \frac{1}{9x - \frac{1}{11x - \dots}}}}}} \quad (19)$$

Per i nostri calcoli con l'aritmetica disponibile con $\text{pdf} \text{latex}$ ci basta troncare la frazione continua alla parte esplicitata, eliminando i puntini. Siccome $x \rightarrow \infty$ quando $\varphi \rightarrow 0$, risulta evidente che in casi in cui si voglia usare questo algoritmo con angoli qualsiasi è meglio eseguire dei test per sostituire il calcolo per valori di $|\varphi| \lesssim 0,1$ (circa 6°) con altri sviluppi approssimati, per esempio con la serie di Maclaurin troncata al secondo termine. Un test simile potrebbe essere eseguito per valori di φ prossimi a 180° ; qui li ometto, perché mi sembrerebbe piuttosto insolito richiedere una filigrana ruotata di 180° , cioè capovolta, mentre non escluderei una filigrana orizzontale, non ruotata affatto.

Ciò premesso, la realizzazione degli algoritmi indicati nelle equazioni (16)–(19) avviene mediante il codice seguente:

```
1 \def\FIL@senocoseno#1{\bgroup%
2 % Trasforma gradi in radianti, divide per due,
3 % e calcola la tangente di fi/2
4 \@tDE=#1\p% angolo in gradi
5 \ifdim\dimexpr\@tDE*\@tDE/\p < 36\p@
6 \@tDE=0.01745329252393\@tDE% angolo in
radianti
7 \xdef\seno{\strippt\dimexpr\@tDE\relax}%
8 \xdef\coseno{\strippt\dimexpr\p@-
9 0.5\@tDE*\@tDE/\p\relax}%
10 \else
11 \@tDB=\dimexpr 114.591559\p@\p@\@tDE
\relax % 2/fi
```

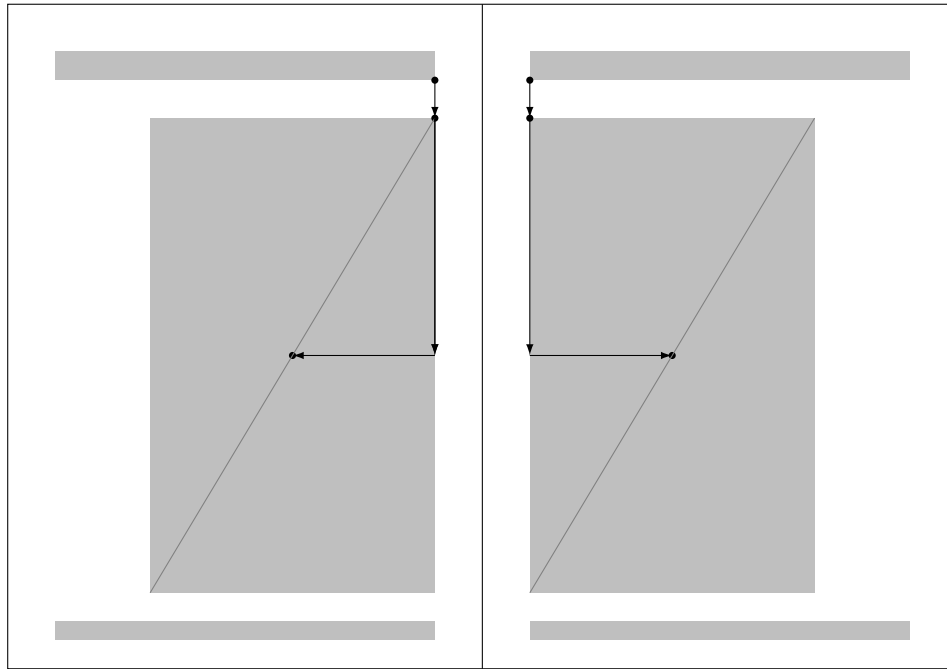


FIGURA 8: Punti di riferimento per collocare le filigrane in uno *spread* con testatine sporgenti. Le frecce indicano i successivi spostamenti dei punti di riferimento dalle estremità interne delle testatine, fino al centro degli specchi di stampa. Le diagonali dello specchio hanno la stessa direzione nelle due pagine affacciate.

```

12 \countdef\I=254\I=11\relax
13 \let\tan\@tdA \let\cot\@tdC
14 \tan=\z@
15 \@whilenum\I>\z@\do{
16   \@tdD=\dimexpr\I\@tdB-\tan\relax
17   \tan=\dimexpr \p@*\p@/\@tdD\relax
18   \advance\I-2\relax}%
19   \cot=\dimexpr\p@*\p@/\tan\relax
20 % Calcola \seno e \coseno con le formule
    parametriche
21 \xdef\seno{\strip@pt\dimexpr2\p@*\p@/(\cot+\tan)
    \relax}%
22 \xdef\coseno{\strip@pt\dimexpr(\cot-\tan)*\p@/(\cot+\tan)
    \relax}%
23 \fi\egroup\ignorespaces}%

```

Nella figura 7 sono riportate alcune pagine filigranate ricavate da documenti composti con la classe *book* e con tre diversi specchi di stampa, con quello di default oppure con lo specchio generato mediante il pacchetto *canoniclayout* oppure con quello generato con il pacchetto *layaureo*. Come si vede, le diverse proporzioni dello specchio di stampa richiedono diverse inclinazioni della filigrana ma questa appare sempre e comunque centrata sulla diagonale dello specchio di stampa.

7 Conclusione

Questa lunga spiegazione, con tre soluzioni diverse per inserire la filigrana, mostra in modo abbastanza approfondito quello che fanno dietro le quinte i vari pacchetti per inserire le filigrane o le immagini di sfondo. Naturalmente le macro che abbiamo sviluppato, volendo, potrebbero diventare il contenuto di un file di estensione personale, per esempio *filigrana.sty*. Ma sarebbe anche il caso di esten-

dere le macro per gestire il caso di composizione fronte retro creato con testatine che sporgono nel margine esterno, dove quindi l'estremità destra delle testatine delle pagine pari e di quelle dispari non hanno la stessa posizione rispetto allo specchio di stampa. In quel caso sarebbe meglio inserire la macro che disegna la filigrana all'estremo interno delle testatine, quindi i comandi di posizionamento sarebbero diversi per le pagine pari rispetto a quelle dispari: figura 8.

Potrebbe essere anche una buona idea quella di non aggiungere le filigrane alle testatine degli stili di pagina standard, ma di creare dei nuovi stili di pagina con nomi diversi, ma che contengano le macro per le filigrane. In questo modo la decisione di mettere o meno la filigrana consiste semplicemente nel cambiare lo stile delle pagine.

Invece, come sottoprodotto di questa esposizione, ho usato spesso le espressioni dimensionali dimostrando che si possono fare calcoli abbastanza complessi anche con la modesta aritmetica dei motori di composizione del sistema T_EX (esclusi quelli che si basano sull'interprete *luatex*, naturalmente, ma qui mi sono rivolto principalmente se non esclusivamente al programma *pdf_latex*). Si possono calcolare anche le funzioni trascendenti e devo dire che le approssimazioni con frazioni continue che ho usato non sono impiegate spesso negli altri pacchetti di calcolo del sistema T_EX. Sono particolarmente soddisfatto delle funzioni trigonometriche calcolate mediante le formule parametriche, che permettono di evitare tanti test per cercare di riportare il calcolo nel primo quadrante o addirittura nel primo ottante. Talvolta può sembrare che abbia voluto

complicare apposta i calcoli ma per mia fortuna ho usato a lungo il calcolo numerico e ho sviluppato una certa sensibilità per evitare alcune trappole che si manifestano sovente con il calcolo automatico⁸. Posso capire che i lettori, che non hanno questa esperienza, trovino gli sviluppi numerici che ho fatto un po' strani. Non si spaventino, per comporre tipograficamente non è necessario essere buoni matematici (anche se non guasta: Gutenberg a suo tempo conosceva molto bene la geometria), infatti basta usare pacchetti già creati da altri. Ma comprendendo quello che c'è nascosto dentro il pacchetto lo si usa certamente meglio.

8 Ringraziamenti

Ringrazio Claudio Fiandrino che ha messo a disposizione sul sito del Forum g_{IT} la sua soluzione al problema posto dal primo richiedente che si è firmato con lo pseudonimo @virgo; ringrazio molto anche questa persona che mi ha dato lo spunto per illustrare il problema “da dentro”. Ringrazio Roberto Giacomelli per aver voluto sportivamente partecipare al “concorso” e fornire una soluzione per il calcolo dell'arcotangente che applica in modo creativo concetti imparati nel primo corso di Analisi Matematica seguito all'Università. Ringrazio Francesco Endrici, un altro partecipante sportivo al “concorso”; si è cimentato con il problema trovando la soluzione giusta per rendere la filigrana indipendente dal testo.

Riferimenti bibliografici

L^AT_EX3 PROJECT TEAM (2005). «L^AT_EX 2_ε font selection». Leggibile con `texdoc fntguide` con la distribuzione TeX Live.

BECCARI, C. (2012). «Il L^AT_EX Reference Manual commentato». Scaricabile dal sito <http://www.guitex.org/home/images/doc/GuideGuIT/latexhandbookcommentato.pdf>.

8. Per esempio, mi piace molto l'artificio sviluppato nel realizzare il comando `\dimexpr` di eseguire lo “scalamento” in due mosse: lo scalamento consiste nel moltiplicare una lunghezza per un quoziente fra due lunghezze. I registri lunghezza e le lunghezze in generale contengono al massimo 32 bit, un paio di servizio e gli ultimi 16 da considerare come parte fratta della misura in punti di quella lunghezza. Prima viene eseguito il prodotto in un registro a 64 bit interno al processore, il che non comporta nessuna perdita di informazione dovuta a troncamenti o arrotondamenti, poi viene eseguita la divisione con il taglio del risultato mediante arrotondamento. In questo modo mediamente i calcoli sono più affidabili ma scegliendo adeguatamente le lunghezze su cui operare si possono eseguire moltiplicazioni fra grandezze, divisioni fra grandezze, inversioni e simili. Si tratta sempre di usare correttamente i tre ingredienti dell'espressione di “scalamento”. Nel codice che ho scritto ho usato spessissimo questa operazione di scalamento ottenendo risultati migliori di quelli che avrei ottenuto con i “vecchi” comandi indicati nel T_EXbook.

BREITENLOHNER, P. (1998). «The ε -T_EX manual». Leggibile con `texdoc etex` con la distribuzione TeX Live.

CARLISLE, D. (2005). «Packages in the ‘graphics’ bundle». Leggibile con `texdoc graphicx` con la distribuzione TeX Live.

CARRERI, A. (2005). *Approssimante di Padé*. Tesi di Laurea, Università degli Studi della Calabria. Questa tesi è in rete all'indirizzo <http://lan.unical.it/Persone/Dellaccio/careri.pdf>. Probabilmente si tratta di un monografia di laurea triennale; il testo è ben focalizzato sul tema e, a parte qualche refuso, è sostanzialmente corretto.

GÄSSLEIN, H., NIEPRASCHK, R. e TKADLEC, J. (2011). «The `pict2e` package». Leggibile con `texdoc pict2e` con la distribuzione TeX Live.

KERN, U. (2007). «Extending L^AT_EX's color facilities: the `xcolor` package». Leggibile con `texdoc xcolor` con la distribuzione TeX Live.

KNUTH, D. E. (1996). *The T_EXbook*. Addison Wesley, Reading, Mass., 16^a edizione.

KRAB THORUP, K., JENSEN, F. e ROWLEY, C. (2007). «The `calc` package – Infix notation arithmetic in L^AT_EX». Leggibile con `texdoc calc` con la distribuzione TeX Live.

LAMPORT, L. (1994). *L^AT_EX: A document preparation system – User guide and reference manual*. Addison Wesley, 2^a edizione.

NIEPRASCHK, R. (2010). «The `eso-pic` package». Leggibile con `texdoc eso-pic` con la distribuzione TeX Live.

ROZHENKO, A. I. (2004). «The `watermark` package». Leggibile con `texdoc watermark` con la distribuzione TeX Live.

TANTAU, T. (2010). «The `TikZ` & `PGF` packages». Leggibile con `texdoc tikz` con la distribuzione TeX Live.

THÀNH, H., RATZ, S., HAGEN, H., JAKOWSKI, P. e SCHRÖDER, M. (2010). «The `pdfTEX` manual». Leggibile con `texdoc pdftex` con la distribuzione TeX Live.

▷ Claudio Beccari
Villarbasse
claudio dot beccari at gmail
dot com