Claudio Beccari

Font e tipografia

pdfLaTeX, XeLaTeX, LuaLaTeX e i font

Versione v.0.9.81 del 2021-03-06

Associati anche tu al GUIT

Fai click per associarti

L'associazione per la diffusione di TEX in Italia, riconosciuta ufficialmente in ambito internazionale, si sostiene *unicamente* con le quote sociali.

Se anche tu trovi che questa guida tematica gratuita ti sia stata utile, il mezzo principale per ringraziare gli autori è diventare socio.

Divenendo soci si ricevono gratuitamente:

- l'abbonamento alla rivista ArsTEXnica;
- il DVD TEX Collection;
- un eventuale oggetto legato alle attività del GUIT.

L'adesione al GUIT prevede un quota associativa compresa tra 12,00 € e 70,00 € a seconda della tipologia di adesione prescelta e ha validità per l'anno solare in corso.

Font e tipografia
Copyright © 2021, Claudio Beccari

Questa documentazione è soggetta alla licenza LPPL (LATEX Project Public Licence), versione 1.3 o successive; il testo della licenza è sempre contenuto in qualunque distribuzione del sistema TEX e nel sito http://www.latex-project.org/lppl.txt.

Questo documento è curato da Claudio Beccari.

Presentazione

Questa guida tematica relativa ai font usabili con i programmi di composizione del sistema TEX può essere utile a coloro che si avvicinato al sistema TEX dopo essersi abituati a word processor di vario genere, come Microsoft Word o LibreOffice Writer o altri simili programmi, nel loro genere ottimi e adatti ai loro scopi.

In questi word processor basta cliccare su una voce del menù per vedere l'intera lista dei font disponibili, spesso molte decine, ed è possibile scegliere i font che si desiderano nel modo più semplice, cioè cliccando sul nome del font che si desidera usare.

Questo semplice modo di procedere non è usabile con i programmi di composizione del sistema TEX per il semplice motivo che il sistema TEX non è un word processor, ma è un sistema di tipocomposizione. In altre parole il processo di composizione va affrontato in un certo modo come in tipografia, scegliendo le casse dei font che si vogliono usare in modo che essi possano essere composti in modo armonico e funzionale alla comodità di lettura del lettore, non per ottenere effetti grafici particolari come viene normalmente fatto in ogni dépliant pubblicitario.

L'utente del sistema TEX deve quindi saper distinguere i vari font, li deve saper usare e abbinare correttamente e deve comunque conoscerne le particolarità per farne buon uso.

Vorrei che il lettore si familiarizzasse con concetti come *codifica*, *famiglia*, *serie*, *forma*, *corpo*, *scartamento*, (detto anche *avanzamento di riga*, ma spesso chiamato impropriamente *interlinea*), sia per i font da usare per comporre il testo sia per quelli destinati alla composizione della matematica; vorrei che conoscesse l'esistenza dei font bitmapped e sapesse come evitarne i difetti; vorrei che conoscesse i font vettoriali e sapesse distinguere le particolarità dei font Type 1 da quelle dei font TrueType o OpenType; e vorrei che il lettore capisse il meccanismo con il quale i programmi del

sistema TEX accedono a tutte le caratteristiche e particolarità dei vari tipi di font. Tutto ciò dovrebbe aiutare il lettore a capire perché i programmi di composizione talvolta usano font diversi da quelli desiderati nonostante siano stati specificati mediante gli appositi comandi di selezione del mark up di LATEX.

Tra i vari programmi di composizione del sistema TEX mi riferirò poco a latex, prevalentemente a pdflatex, xelatex e lualatex; molto, ma non tutto, di ciò che dirò sarà applicabile, *mutatis mutandis*, anche ad altri programmi di composizione del sistema TEX.

Ringrazio Marco Stara per la paziente revisione "ortografica e semantica" del testo; ha segnalato molti refusi e ha colto alcune frasi poco chiare permettendomi di modificare il testo.

Claudio Beccari
`claudio dot beccari at gmail dot com`

Indice

Presentazione 3

Indice 5

Elenco delle figure 6

Elenco delle tabelle 7

1 Le caratteristiche dei font 9
- 1.1 Premessa . . . 9
- 1.2 Tabelle dei segni . . . 16
- 1.3 Le codifiche dei font . . . 19
- 1.4 Le famiglie dei font . . . 29
  - 1.4.1 Le famiglie dei font per comporre il testo . . . 29
  - 1.4.2 Le serie dei font . . . 31
  - 1.4.3 Le forme dei font . . . 31
  - 1.4.4 I corpi dei font . . . 31
  - 1.4.5 Le famiglie di font per comporre la matematica . . . 32

2 L'accesso ai font con LaTeX 2_ε 38
- 2.1 L'accesso ai font TrueType . . . 39
- 2.2 L'accesso ai font Type 1 . . . 39
- 2.3 I font virtuali . . . 40
- 2.4 Il file di descrizione dei font . . . 42
- 2.5 Accesso ai font tramite i loro file di descrizione . . . 43
- 2.6 Scelta dei font matematici . . . 53
- 2.7 Ricapitolazione . . . 56

3 L'interfaccia per l'utente 57
3.1 Selezione di font alternativi 58
3.2 I comandi di basso livello 61
3.3 Installazione di nuovi font 64
3.3.1 I font OpenType da usare con lualatex e xelatex . . . 64
3.3.2 I font vettoriali da usare con pdflatex 65
3.3.3 Un caso più semplice per usare un font Type 1 con pdflatex 66
3.3.4 Le mappe generali dei font 70
3.4 Font per scrivere con alfabeti diversi 72
3.4.1 Alfabeti diversi con lualatex e xelatex 72
3.4.2 Alfabeti diversi con pdflatex 73
3.5 Ricapitolazione 76

Bibliografia 78

Indice analitico 83

Elenco delle figure

1.1 Confronto fra un glifo di un font vettoriale con lo stesso glifo del corrispondente font bitmapped 10
1.2 Il disegno del glifo bitmapped della omega con diacritici 13
1.3 Il disegno del glifo vettoriale della omega con diacritici 14

2.1 Corpi specificati e ottenuti 50

3.1 Schema di flusso dell'uso di fontinst 67

Elenco delle tabelle

1.1 Font Computer Modern tondo con grazie di serie media in corpo 10 20
1.2 Font Computer Modern teletype monospaziato in corpo 10 . . 20
1.3 Font Latin Modern tondo con grazie di serie media in corpo 10 23
1.4 Font Text Companion tondo con grazie di serie media in corpo 10 24
1.5 Font Latin Modern tondo con grazie di serie media in corpo 10 con codifica LY1 25
1.6 Font Latin Modern tondo con grazie di serie media in corpo 10 con codifica T5 26
1.7 Font cirillico tondo con grazie di serie media in corpo 10 27
1.8 Font greco tondo con grazie di serie media in corpo 10 28
1.9 Il font per le lettere matematiche in corpo 10 34
1.10 Il font per i simboli matematici in corpo 10 34
1.11 Il font per i grandi operatori e per i delimitatori estensibili in corpo 10 35
1.12 Il font per la prima serie di simboli matematici dell'American Mathematical Society in corpo 10 36
1.13 Il font per la seconda serie di simboli matematici dell'American Mathematical Society in corpo 10 36

2.1 La stessa frase composta con vari font 52

Le caratteristiche dei font 1

La parola font, ormai entrata nell'uso del linguaggio di chiunque usi un calcolatore, deriva dal francese *la fonte* che indica la fusione e anche il prodotto della fusione. È chiaro il riferimento ai caratteri metallici fusi in una particolare lega di piombo e antimonio, usati per la stampa per più di cinque secoli da Gutenberg in poi. Oggi di fatto non si usano più i caratteri mobili metallici, se non in poche stamperie artigiane di altissima professionalità, oppure per piccoli lavori come la produzione di carta intestata o di biglietti da visita, oppure anche per scritte "cubitali" dove si usano al loro posto caratteri in legno duro – sempre più rari.

La discussione se la parola font sia grammaticalmente maschile o femminile lascia il tempo che trova; e ci saranno sempre coloro che dicono e scrivono *le font* o *le fonti*, e quelli che invece dicono e scrivono *i font* o *i fonti*; quello che è certamente sbagliato, è indicare il plurale con la 's'.

In italiano si usa questa parola più per indicare una collezione di caratteri, riservando la parola 'carattere' al singolo segno o al singolo blocchetto metallico con cui si componeva a mano in tempi passati, non così remoti. Il singolo segno può anche essere chiamato *glifo*, simile all'inglese *glyph* che ha lo stesso significato e la stessa etimologia.

1.1 Premessa

Nella tipografia elettronica, come quella praticata con i programmi di tipocomposizione del sistema TEX, i singoli segni sono delle entità abbastanza virtuali che prendono forma quando vengono resi visibili all'uomo o sullo schermo o sulla carta. In quanto oggetti quasi virtuali, essi possono venire collocati sul supporto visibile all'uomo in posizioni qualsiasi, anche sovrapposti; è chiaro che questo non si poteva fare con i caratteri metallici, se non creando i punzoni per glifi compositi che imitassero caratteri sovrapposti.

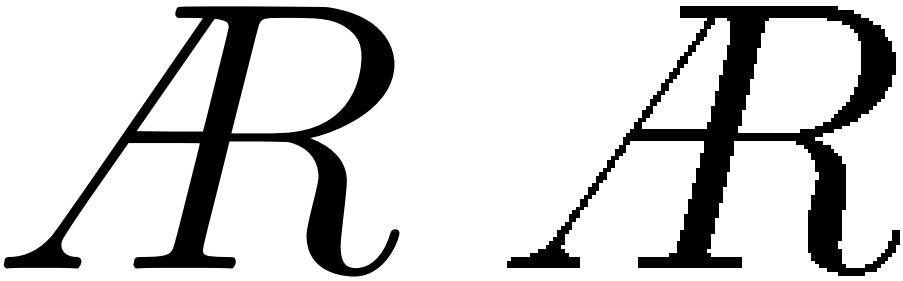


Figura 1.1 Confronto fra un glifo di un font vettoriale con lo stesso glifo del corrispondente font bitmapped

La libertà che si ha con i font elettronici è molto maggiore di quella che si aveva con i font metallici. Per esempio con i caratteri metallici era semplice scrivere: $\mathscr{CB}$, ma era difficilissimo scrivere: $\mathscr{CB}$, se non creando un punzone, una matrice e un unico carattere metallico apposta.

Tuttavia l'elettronica non riesce a rendere visibili i vari glifi se non campionandoli in tanti puntini accostati e talvolta anche parzialmente sovrapposti; è la natura discreta del segnale elettronico-informatico che richiede questa rappresentazione. Che questa discretizzazione avvenga prima della visualizzazione oppure durante la visualizzazione è in un certo senso irrilevante; il fatto è che viene sempre eseguita. La differenza, invece, è sostanziale, perché se la discretizzazione avviene prima della visualizzazione siamo di fronte a *font bitmapped*, mentre se avviene durante la visualizzazione siamo di fronte a *font vettoriali*.

Affinché non ci siano dubbi sul significato del periodo precedente si osservi la figura 1.1; il glifo a sinistra nella figura è un glifo vettoriale (appartiene al font usato per indicare il coefficiente che gli ingegneri aerospaziali chiamano *aspect ratio* ed è la legatura di una 'A' con una 'R') campionato durante la visualizzazione, mentre il secondo è lo stesso glifo prodotto da METAFONT in forma bitmapped ed è stato campionato per poi essere reso alla perfezione da una stampante da 600 pixel al pollice. Il glifo vettoriale e quello bitmapped nella loro grandezza naturale di 12 pt (circa 4mm) sono praticamente indistinguibili ma, ingranditi fino a circa 35 mm, si vede come la discretizzazione preliminare alla visualizzazione del font bitmapped è bruttissima a confronto con la resa del font vettoriale, discretizzato durante

la visualizzazione; se si agisce sulla scala di resa di questa pagina sullo schermo, il segno vettoriale viene ingrandito mantenendo sempre il suo contorno massimamente "liscio" compatibilmente con la bassa densità di pixel dello schermo; mentre il glifo bitmapped ingrandisce mantenendo la stessa scalettatura dei suoi contorni.

Il sistema TEX, nel suo primo stato embrionale del 1978, non poteva fare altro che gestire font già campionati, ma non li gestiva direttamente; semplicemente gestiva le informazioni sulle loro dimensioni "stampate" sui loro accostamenti e sulle legature mediante una tabella nella quale queste informazioni metriche erano codificate in modo adeguato; si trattava dei file `.tfm` (TEX font metrics) e il primo motore di composizione non si occupava di gestire i glifi stessi; esso produceva un file in formato DVI (DeVice Independent) nel quale ogni glifo era indicato con un numero accompagnato dalle coordinate dove collocare il suo punto di riferimento sul supporto[1]. Il file DVI conteneva anche il nome del file con la descrizione grafica dei glifi, ma TEX non se ne occupava direttamente. Era poi il *driver* per lo schermo o per la stampante che doveva reperire le informazioni grafiche di ogni glifo, sulla base del suo numero identificativo e del file contente le informazioni grafiche, e di metterlo al punto giusto sullo schermo o, rispettivamente, sulla carta.

Allora, nel 1978, non esistevano i font vettoriali, ma solo quelli che nel file descrittivo della forma dei glifi erano descritti da matrici di bit che valevano 1 in corrispondenza dei puntini da annerire (o da colorare) e 0 in corrispondenza dei puntini da lasciare bianchi. Il driver di stampa doveva quindi accedere a file diversi per "scrivere" sullo schermo, con una densità di puntini anneribili di circa 72 pixel al pollice, rispetto a scrivere sulla carta dove non era difficile arrivare (a quei tempi) a 150 pixel al pollice o, con macchine particolarmente costose e professionali, anche a 600 pixel al pollice. In altre parole il driver per lo schermo o per la stampa

[1] Le coordinate del foglio nel file DVI hanno l'origine nel punto in alto a sinistra; questo avviene ancora oggi, ma nei file PS e PDF le coordinate hanno l'origine nel punto in basso a sinistra, per cui per eseguire il cambio di riferimento è necessario conoscere le dimensioni reali del supporto su cui scrivere. Per fortuna l'utente non deve occuparsi di questi dettagli, ma capisce perché lavorando con dvips o con pdflatex è necessario che esplicitamente o implicitamente il programma conosca le dimensioni reali del supporto. Con latex e l'uscita in formato DVI le dimensioni reali del supporto non hanno nessuna importanza.

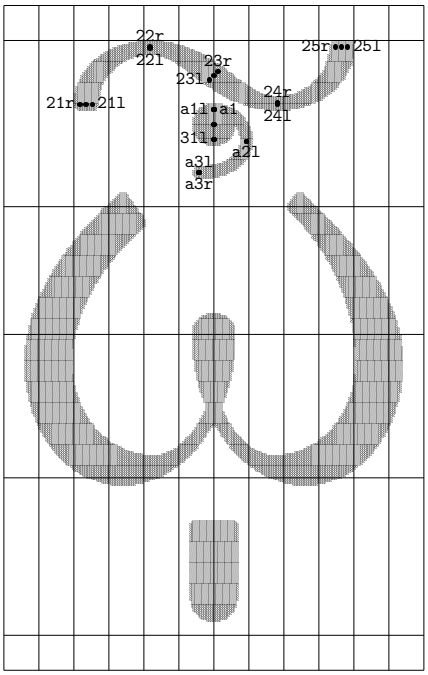
doveva poter accedere a file contenenti le matrici di 'zeri' e 'uni' creati appositamente per il particolare dispositivo di rappresentazione grafica usato per rendere i segni visibili agli uomini. Rappresentare i vari segni con un file corrispondente ad una densità di pixel sbagliata, voleva dire alterare la scala di quanto scritto sulla pagina, oppure sgranare i singoli segni rendendo visibili anche gli spazi fra un puntino e l'altro. Con la densità di pixel giusta la rappresentazione visibile agli uomini era invece perfetta.

Donald E. Knuth insieme a TEX creò anche METAFONT, il programma per creare i file bitmapped insieme ai file metrici. Esso è disponibile ancora oggi e, per mia esperienza personale, trovo che sia il miglior programma per creare font, anche se i suoi font sono bitmapped; esistono programmi per tracciare i contorni di questi font in modo da creare i font vettoriali e il risultato è eccellente, (Nienhuys, 2010). Tanto per fare un esempio si esàminino le figure 1.2 a fronte e 1.3 della pagina 14 tratte dai miei font *CBfonts* (Beccari, 2008b), la cui collezione fa parte integrante delle distribuzioni TEX complete, e sono i font di default per la composizione in greco.

Nella figura 1.2 a fronte, ingrandendo l'immagine, si vedono distintamente i puntini che formano il glifo; attenzione, questi non sono i puntini reali che vengono anneriti sullo schermo o stampando su carta; sono semplicemente i puntini usati da certi font speciali che Knuth ha disegnato per rappresentare i glifi dei suoi font; il concetto è diverso, ma il risultato è simile; il disegno mostrato è quello che si ottiene lanciando METAFONT in modalità proof, che ingrandisce molto il disegno facendolo corrispondere ad una densità di pixel molto grande di 2602 punti al pollice, che porta alla rappresentazione molto grande che appare nella figura 1.2 della prossima pagina.

Le righe verticali che compaiono nel disegno indicano i punti tipografici di larghezza del glifo quando è stampato al suo corpo di progetto di 10 pt. Le righe orizzontali indicano dall'alto: l'altezza degli ascendenti, l'altezza della 'x', cioè la x-height (quella che dà il nome all'unità di misura dimensionale `ex`), l'asse matematico, la linea di base, la profondità dei discendenti. Tutte queste misure sono quelle di "progetto" ma non è vietato che i singoli tratti del glifo oltrepassino leggermente questi valori, come si vede chiaramente per la omega nelle sue estremità superiori e nelle curve inferiori che fuoriescono dai limiti; è un perfezionismo ottico che permette di vedere

METAFONT output 2013.01.01:1835 Page 167 Character 198 "Lowercase omega with smooth breathing, circumflex and iota subscript"




```
21 = 21r + (3.5,0)
22 = 22r + (0,-0.5)
23 = 23l + (2.4,2.4)
24 = 24l + (0,0.5)
25 = 25l + (-3.5,0)
a2 = a2l + (0,0)
a3 = a3r + (0,0)
31 = a1l + (0,-8.5)
a1r = a1 + (0,0)
a2r = a2l + (0,0)
31r = a1 + (0,0)
31l = 31l + (0,0)
31 = a1l + (0,-8.5)
31r = a1 + (0,0)
```

Figura 1.2 Il disegno del glifo bitmapped della omega con diacritici come rappresentato da METAFONT. Se si ingrandisce l'immagine con i mezzi propri del visualizzatore di file PDF, si può vedere bene non solo la seghettatura dei contorni, ma anche i puntini che formano l'immagine.

meglio proporzionate le lettere affiancate che formano un'intera parola o un'intera riga di parole.

I numeri riportati in corrispondenza dei puntini evidenziati nei diacritici superiori, i cui valori sono riportati a destra, sono i punti guida con cui si sono descritti i contorni dei singoli segni; non entro nei dettagli, perché sarebbe necessario descrivere a fondo il programma METAFONT; tuttavia i valori numerici scritti nella parte destra della figura aiutano il disegnatore a mettere in posizione con precisione le singole parti del disegno.

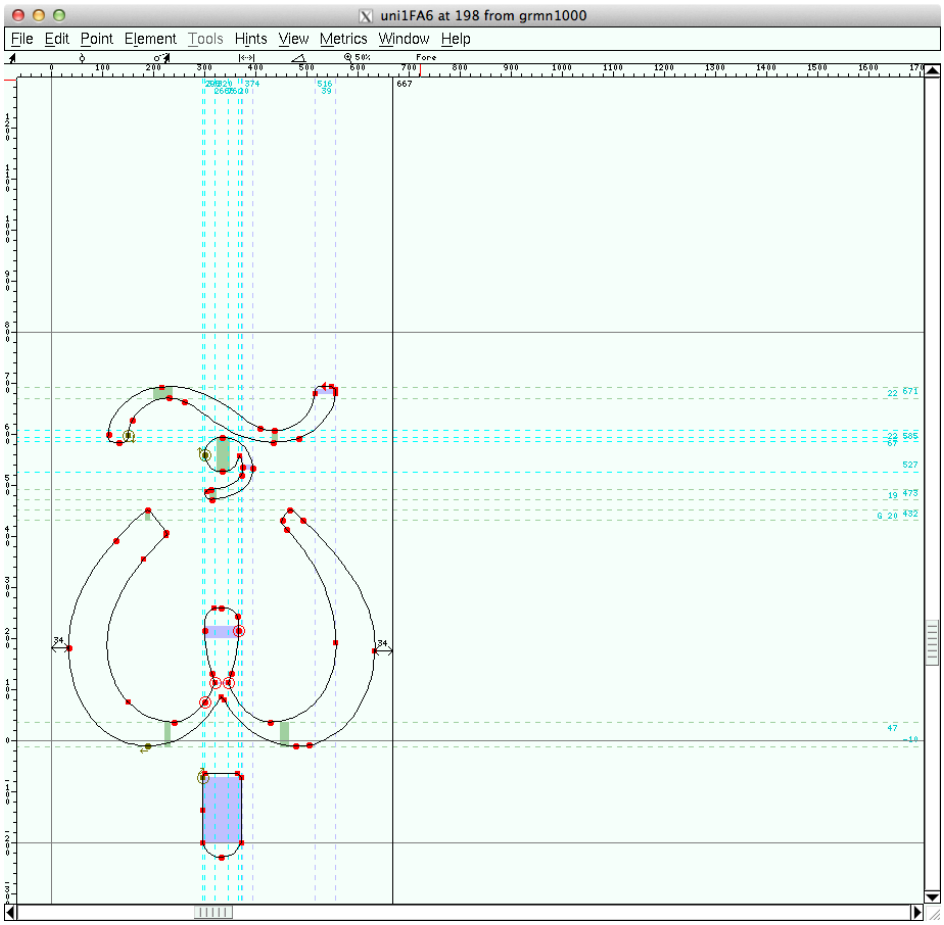


Figura 1.3 Il disegno del glifo vettoriale della omega con diacritici visualizzato attraverso il programma di editing dei font vettoriali FontForge (Wilson, 2000)

La figura 1.3 presenta lo stesso glifo ottenuto tracciando i contorni delle sue varie parti con il programma mftrace (Nienhuys, 2010); questo programma serve per determinare le *spline di Bézier* di terzo grado (Wikipedia, 2012)[2] che approssimano al meglio il contorno seghettato del

[2]Il sito web di Wikipedia non è probabilmente il più autorevole in merito, ma questo

disegno bitmapped. Purtroppo l'immagine della schermata è una immagine bitmapped, quindi se la si ingrandisce molto, si vedono i singoli puntini di cui è composta. Non ci si lasci ingannare: da un lato non sarebbe possibile fare diversamente, visto che si tratta della "cattura" dei pixel del monitor del calcolatore; dall'altro il contorno che vi appare disegnato è puramente virtuale; le sue "coordinate" sono costituite dai nodi delle spline di Bézier, i punti rossi e neri nell'immagine, e queste coordinate vengono passate al driver di visualizzazione (schermo o stampante) che (a) scala le coordinate ricevute; (b) ridisegna virtualmente il contorno a partire dalle coordinate scalate; (c) annerisce (o colora) i punti del supporto grafico che cadono all'interno del contorno[3]; quindi il contorno è una informazione per il driver che deve annerire il suo interno, non è una linea tracciata veramente sul supporto di visualizzazione.

I font vettoriali hanno proprio il pregio che al driver vengono passate le coordinate del contorno di ogni glifo e che l'annerimento dell'interno del contorno avviene al momento della visualizzazione e viene ricomposto immediatamente se si cambia il fattore di ingrandimento della immagine riprodotta. Per questo motivo i font vettoriali possono essere ingranditi a piacere e rimpiccioliti con deformazioni trascurabili; si vedono infatti delle fasce viola (orizzontali) e verdi (verticali) che indicano delle distanze da curare in modo particolare; suggeriscono al driver che riproduce i glifi di non avvicinare troppo i bordi del contorno "distanziati" da quelle barre colorate; costituiscono i cosiddetti *hints*, una caratteristica di cui sono dotati i migliori font vettoriali, e che permettono di non rendere irriconoscibili i glifi troppo rimpiccioliti.

Invece cambiando scala all'immagine riprodotta quando si usano i font bitmapped, i singoli "quadratini" che costituiscono i puntini anneriti vengono scalati insieme all'intera immagine; se l'immagine viene fortemente ingrandita si riconoscono i singoli quadratini, si vedono i bordi seghettati e in sostanza il disegno del singolo glifo appare pesantemente deteriorata.

---

particolare sito presenta dei disegni dinamici nei quali si vede benissimo da dove nasce la curva o spline di Bézier.

[3]Per sapere quale sia l'interno del contorno, su questo è segnato un verso positivo di percorrenza e l'interno del contorno è la parte che viene mantenuta a destra del percorso. Guardando attentamente la figura, ingrandendola, si vede la freccina che indica il verso positivo di percorrenza, per esempio in prossimità del punto nero nella curva in basso a sinistra, in prossimità della linea di base.

I visualizzatori di file PDF generalmente contengono un filtro digitale bidimensionale[4] che smorza i contrasti al bordo dei glifi, per cui i glifi bitmapped possono apparire correttamente disegnati se la densità di pixel della discretizzazione si accorda con quella dello schermo; ma non possono fare più di tanto se l'immagine viene ingrandita, cosicché la lettura di quanto è scritto sulla pagina, malgrado l'ingrandimento e malgrado il filtro bidimensionale, appare mal definita e con i contorni sfumati.

Attenzione: di sua natura latex usa i font bitmapped; pdflatex, xelatex e lualatex di loro natura usano i font vettoriali; non sono sicuro che xelatex possa usare font bitmapped, ma pdflatex li può usare se non ne dispone di vettoriali equivalenti. Per l'uscita in formato DVI, prodotta normalmente da latex, sarebbero necessari i font bitmapped, ma spessissimo, se non sempre, il formato DVI viene trasformato in PS mediante dvips che, se ne dispone, preferisce usare i font vettoriali; oppure il formato DVI viene trasformato in PDF mediante dvipdfmx, oppure il formato PS viene trasformato in formato PDF tramite pstopdf, e questi sono tutti programmi che, se ne dispongono, preferiscono usare i font vettoriali. Ecco perché il programma latex viene usato pochissimo se il risultato finale da ottenere è un file in formato PDF, che si può ottenere direttamente con pdflatex. In tutti questi casi, comunque si arrivi al formato finale PS o PDF, sono sempre preferiti i font vettoriali, cosa che i programmi indicati fanno sempre anche se fossero disponibili i font bitmapped equivalenti. Oggi direi che i font bitmapped sono caduti in disuso e riappaiono soltanto in certi rari casi, oppure quando si disegnano dei nuovi font mediante METAFONT (che, d'altronde, come spiegato sopra, si possono sempre trasformare in font vettoriali).

1.2 Tabelle dei segni

Il file metrico `.tfm` relativo ad un dato font individua ogni glifo con un numero; d'ora in poi chiamerò questo numero l'*indirizzo* del glifo. In inglese

[4] Un filtro digitale bidimensionale è un algoritmo che prende in esame un certo numero non troppo grande di pixel adiacenti – coprenti perciò una superficie, non allineati lungo una linea – e colora il pixel centrale in base al suo colore proprio e al colore medio pesato dei pixel adiacenti. È chiaro che un pixel nero circondato da pixel neri rimane nero, ma un pixel sul contorno di un glifo può essere nero, ma non tutti i pixel adiacenti sono neri, quindi il suo colore diventa di un grigio più o meno chiaro a seconda di quanto sono i pixel bianchi adiacenti. La presenza di questo algoritmo smorza la seghettatura al contorno, grazie alla presenza dei pixel più o meno grigi.

assume vari nomi, fra i quali quello più usato, almeno con i font UNICODE, è *code point*. Anche il file che contiene la descrizione grafica dei glifi individua ogni segno con il suo indirizzo; talvolta anche con il suo nome. I nomi sono più o meno codificati, ed esiste la denominazione Adobe, derivante dalla lista dei nomi usati dalla compagnia Adobe quando ha "inventato" i font PostScript (vettoriali) e il linguaggio PostScript. I nomi, per quanto riguarda il sistema TEX, servono solo per scopi particolari.

Ogni particolare font, dunque, è costituito da un certo numero di glifi; Knuth inizialmente creò la collezione di font Almost Computer Modern, che poi, in forma definitiva, tolti alcuni errori di disegno, chiamò Computer Modern, sigla CM. Questi sono i font predefiniti da ogni programma di tipocomposizione del sistema TEX tranne XƎTEX e XƎLATEX, oppure LuaTEX e LuaLATEX; per questi ultimi i font predefiniti sono i Latin Modern, sigla LM. Vedremo fra poco la differenza.

I font CM formano una collezione abbastanza completa di font di varie famiglie, serie e forme e godono della rara particolarità di essere dotati di corpi ottici. Le famiglie sono diverse, ma sostanzialmente sono formate da quella dei font con grazie, da quelli senza grazie e da quelli a spaziatura fissa (*teletype*, cioè telescrivente o macchina da scrivere; ma la parola *teletype* ha generato la sigla `tt` per indicare questa famiglia). Le serie sono quella media e quella nera estesa per quasi tutti i font; le forme sono molto varie, vanno dal tondo all'inclinato, al corsivo, al maiuscoletto; non tutte le famiglie sono dotate di tutte le serie e le forme. La collezione CM contiene anche delle forme rare come quelle dei font dunhill, dei funny font, e dei CMinch, i grandi font alti un pollice (inch) che Knuth usa per numerare i capitoli dei suoi testi: Knuth (1984) e Knuth (1986).

La genialità dei font CM sono i corpi ottici; che siano bitmapped o vettoriali, i font vanno usati in diversi corpi; per un testo composto in corpo 10 pt, si usa il corpo 10 per il testo corrente; il corpo 9 per le citazioni e talvolta per le didascalie; il corpo 8 per le note a piè di pagina; il corpo 7 per i pedici e gli apici di primo livello; il corpo 5 per gli apici e i pedici di secondo livello; il corpo 14 per i titoli dei paragrafi, il corpo 12 per i titoli dei sottoparagrafi, il corpo 10 (spesso in nero o in maiuscoletto) per i titoli dei sotto sottoparagrafi. I corpi 17, 20, 25 e successivi sono usati solo per gli elementi che formano le pagine dei titoli o i frontespizi[5].

[5] Per impostazione predefinita i corpi ottici dei font CM sono in progressione lineare

Se un font fosse disponibile in un solo corpo, bisognerebbe ingrandirlo o rimpicciolirlo per ottenere quei caratteri di corpo maggiore o minore; i più piccoli potrebbero diventare illeggibili e in particolare i tratti sottili, specialmente in mancanza di *hinting*, potrebbero non venire riprodotti; quelli più grandi diventerebbero talmente neri da poter sembrare di serie nera, malgrado siano ingrandimenti di font di serie media. Per questo i font CM e pochi altri font come questi sono dotati di corpi ottici; cioè il font tondo con grazie diritto di serie media di corpo 5, assomiglia a quello di corpo 10, ma i suoi tratti non sono ridotti proporzionalmente al corpo; si osservi questo testo composto in corpo 10 con questo composto in corpo 5 ingrandito del doppio. Al contrario si osservi questo testo composto in corpo 10 alle sue dimensioni naturali con questo composto in corpo 17 ridotto alle dimensioni del corpo 10. Da questi due esempi si vede chiaramente che il font di corpo 5 rispetto a quello in corpo 10 è proporzionalmente più largo e con le aste più spesse. Analogamente il font di corpo 17 ridotto al corpo 10 presenta delle aste più sottili ed appare tutto più smilzo.

Il difetto dei font CM è che funzionano bene solo per comporre in inglese moderno, una delle poche lingue occidentali scritte con l'alfabeto latino privo di segni diacritici. D'altra parte Kunth è statunitense e, come è noto, gli statunitensi spesso conoscono solo la loro lingua madre. Ma Knuth, oltre ad essere una persona molto colta, è di origine danese, quindi ha avuto un occhio particolare per gli accenti, i caratteri speciali e gli altri segni diacritici, ma in una maniera che è risultata poco efficiente quando il sistema TEX è arrivato in Europa. Egli ha voluto conservare piccoli i file dei font, di soli 128 caratteri, e per questo ha rinunciato a disegnare glifi dotati dei segni diacritici, come invece è stato fatto con la collezione dei font European Computer Modern, sigla EC; ovviamente la cosa andava benissimo per i primordi del sistema TEX, ma appena questo è arrivato in Europa verso il 1984 sono cominciati i guai. Inoltre non ha conservato la stessa codifica per le varie famiglie dei font Computer Modern; si confrontino le tabelle del font *cmr10* (tondo con grazie di serie media)

---

discendente partendo dal corpo 10, mentre sono in progressione geometrica ascendente di ragione 1,2 per i corpi superiori al corpo 10. I valori effettivamente usati sono i corpi 10,95 ($\approx 10\sqrt{1,2}$), 12, 14,4, 17,28, 20,74, 24,88, 29,86, 35,84; gli ultimi due corpi in realtà sono usati solo per i font European Computer Modern di cui si parlerà più avanti.

con il font *cmtt10* (teletype monospaziato) raffigurati rispettivamente nelle tabelle 1.1 e 1.2 della prossima pagina.

Come si vede la disposizione dei segni dall'indirizzo 32 all'indirizzo 126 per i font teletype coincide con la tabella ASCII[6]; mentre per il font tondo questa corrispondenza manca, o, se vogliamo, esiste solo per i caratteri dall'indirizzo 33 all'indirizzo 122. Questa discrepanza col tempo si è rivelata grave. Infatti si confronti quali caratteri si trovano agli indirizzi 123 e 125: si tratta delle parentesi graffe { e } nella tabella ASCII e nel font *cmtt10*; ma con il font *cmr10* esse appaiono come – e ″. Questo difetto è stato eliminato con i font a 256 caratteri.

In ogni caso, in entrambe le tabelle si nota la presenza delle lettere greche maiuscole da usarsi in matematica e dei segni diacritici quasi tutti inseriti nei primi 32 indirizzi; la tilde all'indirizzo 126 è presente in entrambe le tabelle, ma mentre nel font tondo svolge la funzione di accento, nel font teletype essa appare più come il segno che in informatica viene usato per indicare la cartella `HOME` nei sistemi UNIX (`~`), anche se può servire da accento.

Gli accenti e gli altri segni diacritici vengono collocati sopra o sotto le lettere che devono riceverli mediante un algoritmo di sovrapposizione; il risultato è abbastanza buono, ma l'algoritmo di divisione in sillabe in fin di riga (*cesura*[7] in fin di riga) dei vari programmi di composizione viene impedito dall'algoritmo per la collocazione dei segni diacritici. Poiché tutte le lingue europee, tranne l'inglese moderno, fanno uso di diacritici, questo fatto spiega perché i font CM in Europa sono virtualmente abbandonati; in realtà i font CM vengono ancora usati in matematica come font per gli operatori scritti in lettere latine e per disporre delle lettere greche maiuscole diritte.

1.3 Le codifiche dei font

LaTeX divenne disponibile in forma preliminare circa nel 1984 e il suo primo manuale fu pubblicato nel 1986 (Lamport, 1986); questa fu la cosiddetta

[6] Lo spazio nella tabella all'indirizzo 32 è reso visibile mediante il segno ␣.

[7] Per essere precisi bisognerebbe distinguere la divisione in sillabe secondo le regole grammaticali, che viene chiamata *sillabazione*, e la divisione in sillabe tipografica, chiamata *cesura*; i punti di divisione della cesura sono un sottoinsieme di quelli della sillabazione essenzialmente per motivi psicosociologici, estetici e pratici.

Tabella 1.1 Font Computer Modern tondo con grazie di serie media in corpo 10

	’00	’01	’02	’03	’04	’05	’06	’07	’10	’11	’12	’13	’14	’15	’16	’17	
’000	Γ 0	Δ 1	Θ 2	Λ 3	Ξ 4	Π 5	Σ 6	Υ 7	Φ 8	Ψ 9	Ω 10	ff 11	fi 12	fl 13	ffi 14	ffl 15	"00
’020	ı 16	ȷ 17	` 18	´ 19	˘ 20	ˇ 21	¯ 22	° 23	˛ 24	ß 25	æ 26	œ 27	ø 28	Æ 29	Œ 30	Ø 31	"10
’040	˙ 32	! 33	” 34	# 35	$ 36	% 37	& 38	’ 39	(40	) 41	* 42	+ 43	, 44	- 45	. 46	/ 47	"20
’060	0 48	1 49	2 50	3 51	4 52	5 53	6 54	7 55	8 56	9 57	: 58	; 59	¡ 60	= 61	¿ 62	? 63	"30
’100	@ 64	A 65	B 66	C 67	D 68	E 69	F 70	G 71	H 72	I 73	J 74	K 75	L 76	M 77	N 78	O 79	"40
’120	P 80	Q 81	R 82	S 83	T 84	U 85	V 86	W 87	X 88	Y 89	Z 90	[91	“ 92	] 93	ˆ 94	˙ 95	"50
’140	‘ 96	a 97	b 98	c 99	d 100	e 101	f 102	g 103	h 104	i 105	j 106	k 107	l 108	m 109	n 110	o 111	"60
’160	p 112	q 113	r 114	s 115	t 116	u 117	v 118	w 119	x 120	y 121	z 122	– 123	— 124	˝ 125	˜ 126	¨ 127	"70
	"00	"01	"02	"03	"04	"05	"06	"07	"08	"09	"0A	"0B	"0C	"0D	"0E	"0F	

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz

Parametri

inclinazione	0,00000	x-height	4,30554 pt
spazio interparola	3,33332 pt	larghezza del quadrato	10,19999 pt
allungamento interparola	1,66665 pt	spazio extra	1,11111 pt
accorciamento interparola	1,11111 pt	corpo nominale	10,00000 pt
alfabeto maiuscolo	186,18093 pt	alfabeto minuscolo	127,58369 pt

Tabella 1.2 Font Computer Modern teletype monospaziato in corpo 10

	’00	’01	’02	’03	’04	’05	’06	’07	’10	’11	’12	’13	’14	’15	’16	’17	
’000	Γ 0	Δ 1	Θ 2	Λ 3	Ξ 4	Π 5	Σ 6	Υ 7	Φ 8	Ψ 9	Ω 10	↑ 11	↓ 12	' 13	¡ 14	¿ 15	"00
’020	ı 16	ȷ 17	` 18	´ 19	˘ 20	ˇ 21	¯ 22	° 23	˛ 24	ß 25	æ 26	œ 27	ø 28	Æ 29	Œ 30	Ø 31	"10
’040	␣ 32	! 33	" 34	# 35	$ 36	% 37	& 38	’ 39	(40	) 41	* 42	+ 43	, 44	- 45	. 46	/ 47	"20
’060	0 48	1 49	2 50	3 51	4 52	5 53	6 54	7 55	8 56	9 57	: 58	; 59	< 60	= 61	> 62	? 63	"30
’100	@ 64	A 65	B 66	C 67	D 68	E 69	F 70	G 71	H 72	I 73	J 74	K 75	L 76	M 77	N 78	O 79	"40
’120	P 80	Q 81	R 82	S 83	T 84	U 85	V 86	W 87	X 88	Y 89	Z 90	[91	\ 92	] 93	ˆ 94	_ 95	"50
’140	‘ 96	a 97	b 98	c 99	d 100	e 101	f 102	g 103	h 104	i 105	j 106	k 107	l 108	m 109	n 110	o 111	"60
’160	p 112	q 113	r 114	s 115	t 116	u 117	v 118	w 119	x 120	y 121	z 122	{ 123	| 124	} 125	˜ 126	¨ 127	"70
	"00	"01	"02	"03	"04	"05	"06	"07	"08	"09	"0A	"0B	"0C	"0D	"0E	"0F	

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz

Parametri

inclinazione	0,00000	x-height	4,30554 pt
spazio interparola	5,24995 pt	larghezza del quadrato	10,49989 pt
allungamento interparola	0,00000 pt	spazio extra	5,24995 pt
accorciamento interparola	0,00000 pt	corpo nominale	10,00000 pt
alfabeto maiuscolo	136,49880 pt	alfabeto minuscolo	136,49880 pt

killer application, nel senso che contribuì ad una affermazione improvvisa del sistema TEX in tutto il mondo. In particolar modo in Europa, nonostante gli inconvenienti del sottostante programma tex che poteva gestire solo font a 128 caratteri e una lingua sola alla volta, senza parlare della cesura impedita dai segni diacritici. Ricordo che nella prima conferenza europea del TUG (tenutasi in Italia nel 1985) fu presentato un lavoro del francese Jacques Desarmenien[8] che presentò un suo lavoro per comporre il francese con font CM modificati così da togliere alcuni glifi meno usati e sostituirli con i glifi delle vocali minuscole francesi accentate, oltre alla 'ç'; in questo modo, ci voleva un diverso file di formato e una diversa tabella di cesura, ma riusciva a comporre in francese eseguendo la cesura correttamente nonostante la moltitudine di accenti presenti in quella lingua. Certamente il suo contributo ha avuto una notevole importanza in Francia e i pattern di cesura francesi oggi disponibili sono nati originariamente anche con la sua collaborazione. Ma quel lavoro apriva la via all'uso di font già muniti di lettere disegnate con i loro diacritici; questo fatto si realizzò dopo pochi anni, dopo la conferenza internazionale del TUG a Cork nel 1991.

Fino al 1991, circa, i programmi di composizione del sistema TEX usavano solo font a 128 caratteri; in realtà METAFONT-2 non aveva questa limitazione, ma di fatto fino a quel momento non erano stati sviluppati font con più di 128 caratteri. In quell'anno Knuth pubblicò e rese disponibile la versione 3 di TEX che fece due importanti balzi in avanti: (*a*) fu in grado di gestire font di 256 caratteri e (*b*) poté gestire contemporaneamente più di una lingua. Entrambi questi progressi furono importantissimi per gli utenti europei e di tutti quei paesi dove si scrive con l'alfabeto latino. Nel frattempo anche nel mondo slavo erano stati sviluppati i font cirillici con gli stessi parametri stilistici dei font CM; anche i font greci a 128 caratteri furono resi disponibili in quegli anni. Cominciò ad essere realizzato in quegli anni il pacchetto `babel` per gestire le lingue; esso fece progressi considerevoli e furono disponibili quasi immediatamente le regole di cesura per alcune decine di lingue.

Il programma dvips risale anch'esso alla seconda metà degli anni '80 e consentì di poter incorporare direttamente le figure in formato `.eps`;

[8] Nella rivista TUGboat del 1984 (Desarmenien, 1984) apparve un altro articolo di Desarmenien che citava il suo articolo presentato per la conferenza europea; quest'ultimo sembra introvabile.

precedentemente le figure venivano semplicemente incollate a mano nei fogli stampati nei quali erano stati lasciati gli spazi per questa operazione. Non solo ma anche i font PostScript erano diventati disponibili per una quantità di programmi e sistemi operativi; quindi anche i font CM furono trasformati in font vettoriali, cosicché mediante dvips fu possibile servirsi di questi "nuovi" font vettoriali.

Ma nella conferenza internazionale del TUG (TEX Users Group) tenutasi a Cork nel 1991, i partecipanti definirono una nuova codifica dei font a 256 caratteri da usarsi con il sistema TEX; la vecchia codifica, più o meno definita e mostrata nelle tabelle 1.1 della pagina 20 e 1.2 della pagina 20, fu chiamata *OT1* e la nuova codifica per i caratteri latini a 256 caratteri fu chiamata *T1*; essa è mostrata nella tabella 1.3 a fronte. Poco dopo furono disponibili i font DC (Double Cork encoding), diventati in forma definitiva i font EC (European Computer Modern); furono creati con METAFONT, ma non furono mai resi disponibili in forma vettoriale con le distribuzioni TEX-Live e MiKTeX[9]. Fu invece resa disponibile la collezione dei font CM-super (completamente vettoriali e contenenti anche i caratteri cirillici; i caratteri latini e i caratteri cirillici sono gestibili separatamente tramite appositi font virtuali, di cui si parlerà più avanti). Per maggiori informazioni sulle codifiche si veda Beccari e Gordini (2018).

Contemporaneamente fu definita la codifica *TS1* per rendere disponibile un supplemento di un centinaio di glifi che non trovano posto nella codifica a 256 caratteri; tabella 1.4 della pagina 24.

I font Latin Modern vengono distribuiti con diverse posizioni dei glifi negli indirizzi dal 128 al 255. Costituiscono codifiche diverse; per esempio si consideri la tabella 1.5 della pagina 25 che contiene più o meno gli stessi caratteri della tabella 1.3 della prossima pagina, ma i caratteri sono disposti diversamente; in particolare contengono il simbolo dell'euro.

Gli stessi font Latin Modern sono anche disponibili nella codifica *T5*, tabella 1.6 della pagina 26, e contengono tutti i caratteri latini con diacritici particolari per poter comporre in vietnamita. Capisco che comporre in

[9] Queste due distribuzioni contengono i file sorgente per generare i font EC bitmapped; chi installasse una versione base del sistema TEX, disporrebbe solo dei font EC bitmapped, e la loro visualizzazione con i programmi adatti al formato PDF è o scadente o mal definita. Ora nelle versioni più recenti di Adobe Reader un filtro digitale bidimensionale aggiusta bene la visualizzazione, ma almeno fino alla versione 8 era praticamente impossibile leggere un documento PDF composto con i font EC bitmapped.

Tabella 1.3 Font Latin Modern tondo con grazie di serie media in corpo 10

	’00	’01	’02	’03	’04	’05	’06	’07	’10	’11	’12	’13	’14	’15	’16	’17	
’000	` 0	´ 1	ˆ 2	˜ 3	¨ 4	˝ 5	° 6	ˇ 7	˘ 8	¯ 9	˙ 10	˛ 11	¸ 12	’ 13	‹ 14	› 15	"00
’020	“ 16	” 17	” 18	« 19	» 20	– 21	— 22	23	0 24	1 25	J 26	ff 27	fi 28	fl 29	ffi 30	ffl 31	"10
’040	␣ 32	! 33	" 34	# 35	$ 36	% 37	& 38	’ 39	(40	) 41	* 42	+ 43	, 44	- 45	. 46	/ 47	"20
’060	0 48	1 49	2 50	3 51	4 52	5 53	6 54	7 55	8 56	9 57	: 58	; 59	< 60	= 61	> 62	? 63	"30
’100	@ 64	A 65	B 66	C 67	D 68	E 69	F 70	G 71	H 72	I 73	J 74	K 75	L 76	M 77	N 78	O 79	"40
’120	P 80	Q 81	R 82	S 83	T 84	U 85	V 86	W 87	X 88	Y 89	Z 90	[91	\ 92	] 93	ˆ 94	— 95	"50
’140	‘ 96	a 97	b 98	c 99	d 100	e 101	f 102	g 103	h 104	i 105	j 106	k 107	l 108	m 109	n 110	o 111	"60
’160	p 112	q 113	r 114	s 115	t 116	u 117	v 118	w 119	x 120	y 121	z 122	{ 123	| 124	} 125	˜ 126	- 127	"70
’200	Ă 128	Ą 129	Ć 130	Č 131	Ď 132	Ě 133	Ę 134	Ğ 135	Ĺ 136	Ľ 137	Ł 138	Ń 139	Ň 140	Ŋ 141	Ő 142	Ŕ 143	"80
’220	Ř 144	Ś 145	Š 146	Ş 147	Ť 148	Ţ 149	Ű 150	Ů 151	Ÿ 152	Ź 153	Ž 154	Ż 155	IJ 156	İ 157	đ 158	§ 159	"90
’240	ă 160	ą 161	ć 162	č 163	ď 164	ě 165	ę 166	ğ 167	ĺ 168	ľ 169	ł 170	ń 171	ň 172	ŋ 173	ő 174	ŕ 175	"A0
’260	ř 176	ś 177	š 178	ş 179	ť 180	ţ 181	ű 182	ů 183	ÿ 184	ź 185	ž 186	ż 187	ij 188	¡ 189	¿ 190	£ 191	"B0
’300	À 192	Á 193	Â 194	Ã 195	Ä 196	Å 197	Æ 198	Ç 199	È 200	É 201	Ê 202	Ë 203	Ì 204	Í 205	Î 206	Ï 207	"C0
’320	Ð 208	Ñ 209	Ò 210	Ó 211	Ô 212	Õ 213	Ö 214	Œ 215	Ø 216	Ù 217	Ú 218	Û 219	Ü 220	Ý 221	Þ 222	SS 223	"D0
’340	à 224	á 225	â 226	ã 227	ä 228	å 229	æ 230	ç 231	è 232	é 233	ê 234	ë 235	ì 236	í 237	î 238	ï 239	"E0
’360	ð 240	ñ 241	ò 242	ó 243	ô 244	õ 245	ö 246	œ 247	ø 248	ù 249	ú 250	û 251	ü 252	ý 253	þ 254	ß 255	"F0
	"00	"01	"02	"03	"04	"05	"06	"07	"08	"09	"0A	"0B	"0C	"0D	"0E	"0F	

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz

Parametri

inclinazione	0,00000	x-height	4,30549 pt
spazio interparola	3,33332 pt	larghezza del quadrato	10,00000 pt
allungamento interparola	1,66665 pt	spazio extra	1,11111 pt
accorciamento interparola	1,11111 pt	corpo nominale	10,00000 pt
alfabeto maiuscolo	186,18060 pt	alfabeto minuscolo	127,58364 pt

vietnamita non è molto frequente fra gli utenti italiani; ma questo esempio mostra un’altra disposizione dei caratteri con un’ulteriore codifica adatta per mettere a disposizione del compositore dei glifi che non sono presenti con la codifica *T1*, quella normalmente usata per comporre nelle lingue europee occidentali. Per i font Latin Modern esiste anche una codifica *L7X* per comporre in lituano che usa ulteriori glifi oltre a quelli esposti nella tabella 1.3.

Per altri alfabeti come il cirillico, le tabelle dei font possono apparire

Tabella 1.4 Font Text Companion tondo con grazie di serie media in corpo 10

	'00	'01	'02	'03	'04	'05	'06	'07	'10	'11	'12	'13	'14	'15	'16	'17	
'000	` 0	´ 1	ˆ 2	˜ 3	¨ 4	˝ 5	° 6	ˇ 7	˘ 8	¯ 9	˙ 10	˛ 11	¸ 12	' 13	14	15	"00
'020	16	17	" 18	19	20	¯ 21	¯ 22	23	← 24	→ 25	⌒ 26	⌒ 27	⌒ 28	⌒ 29	30	31	"10
'040	b 32	33	34	35	$ 36	37	38	' 39	40	41	* 42	43	, 44	= 45	. 46	/ 47	"20
'060	0 48	1 49	2 50	3 51	4 52	5 53	6 54	7 55	8 56	9 57	58	59	⟨ 60	− 61	⟩ 62	63	"30
'100	64	65	66	67	68	69	70	71	72	73	74	75	76	℧ 77	78	○ 79	"40
'120	80	Ȯ 81	82	83	84	85	86	Ω 87	88	89	90	⟦ 91	92	⟧ 93	↑ 94	↓ 95	"50
'140	` 96	97	⋆ 98	○|○ 99	† 100	101	102	103	104	105	106	107	108	⚭ 109	♪ 110	111	"60
'160	112	ȯ 113	114	f 115	116	117	118	119	120	121	122	123	124	125	~ 126	= 127	"70
'200	˘ 128	˘ 129	˝ 130	" 131	† 132	‡ 133	‖ 134	‰ 135	• 136	℃ 137	$ 138	¢ 139	f 140	₡ 141	₩ 142	₦ 143	"80
'220	₲ 144	₱ 145	£ 146	₨ 147	⁋ 148	¡ 149	₫ 150	™ 151	‰ 152	¶ 153	₪ 154	№ 155	⅍ 156	℮ 157	° 158	SM 159	"90
'240	{ 160	} 161	¢ 162	£ 163	¤ 164	¥ 165	¦ 166	§ 167	¨ 168	© 169	ª 170	⓪ 171	¬ 172	Ⓟ 173	® 174	¯ 175	"A0
'260	° 176	± 177	² 178	³ 179	´ 180	µ 181	¶ 182	· 183	※ 184	¹ 185	º 186	√ 187	¼ 188	½ 189	¾ 190	€ 191	"B0
'300	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	"C0
'320	208	209	210	211	212	213	× 214	215	216	217	218	219	220	221	222	223	"D0
'340	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	"E0
'360	240	241	242	243	244	245	÷ 246	247	248	249	250	251	252	253	254	255	"F0
	"00	"01	"02	"03	"04	"05	"06	"07	"08	"09	"0A	"0B	"0C	"0D	"0E	"0F	

Parametri

inclinazione	0,00000	x-height	4,30549 pt
spazio interparola	3,33332 pt	larghezza del quadrato	10,00000 pt
allungamento interparola	1,66665 pt	spazio extra	1,11111 pt
accorciamento interparola	1,11111 pt	corpo nominale	10,00000 pt

come nella tabella 1.7 della pagina 27; vi si nota che la prima metà della tabella contiene in realtà l'alfabeto latino e la seconda metà l'alfabeto cirillico; questa situazione rispecchia un poco quella che si presenta nei paesi dove si usano alfabeti diversi, dove la codifica dei font deve contenere anche l'alfabeto latino affinché gli utenti possano scrivere indipendentemente con l'uno o con l'altro alfabeto. La tastiera dei calcolatori in uso in quei paesi contiene un comodo tasto per passare immediatamente dall'uno all'altro alfabeto. Di fatto di alfabeti cirillici se ne usano diversi, nel senso che alcuni contengono segni speciali che non sono presenti negli altri; esistono quindi codifiche diverse per le varianti di ciascuno di questi alfabeti, *T2A*, *T2B*, *T2C*; per i font cirillici esiste anche la codifica *X2* che non contiene i

Tabella 1.5 Font Latin Modern tondo con grazie di serie media in corpo 10 con codifica LY1

	’00	’01	’02	’03	’04	’05	’06	’07	’10	’11	’12	’13	’14	’15	’16	’17	
’000	$_{0}$	€ $_{1}$	$_{2}$	$_{3}$	/ $_{4}$	˙ $_{5}$	˝ $_{6}$	˛ $_{7}$	fl $_{8}$	$_{9}$	$_{10}$	ff $_{11}$	fi $_{12}$	$_{13}$	ffi $_{14}$	ffl $_{15}$	"00
’020	ı $_{16}$	J $_{17}$	` $_{18}$	´ $_{19}$	ˇ $_{20}$	˘ $_{21}$	¯ $_{22}$	° $_{23}$	¸ $_{24}$	ß $_{25}$	æ $_{26}$	œ $_{27}$	ø $_{28}$	Æ $_{29}$	Œ $_{30}$	Ø $_{31}$	"10
’040	$_{32}$	! $_{33}$	" $_{34}$	# $_{35}$	$ $_{36}$	% $_{37}$	& $_{38}$	’ $_{39}$	($_{40}$	) $_{41}$	* $_{42}$	+ $_{43}$	, $_{44}$	- $_{45}$	. $_{46}$	/ $_{47}$	"20
’060	0 $_{48}$	1 $_{49}$	2 $_{50}$	3 $_{51}$	4 $_{52}$	5 $_{53}$	6 $_{54}$	7 $_{55}$	8 $_{56}$	9 $_{57}$	: $_{58}$	; $_{59}$	< $_{60}$	= $_{61}$	> $_{62}$	? $_{63}$	"30
’100	@ $_{64}$	A $_{65}$	B $_{66}$	C $_{67}$	D $_{68}$	E $_{69}$	F $_{70}$	G $_{71}$	H $_{72}$	I $_{73}$	J $_{74}$	K $_{75}$	L $_{76}$	M $_{77}$	N $_{78}$	O $_{79}$	"40
’120	P $_{80}$	Q $_{81}$	R $_{82}$	S $_{83}$	T $_{84}$	U $_{85}$	V $_{86}$	W $_{87}$	X $_{88}$	Y $_{89}$	Z $_{90}$	[$_{91}$	\ $_{92}$	] $_{93}$	ˆ $_{94}$	— $_{95}$	"50
’140	‘ $_{96}$	a $_{97}$	b $_{98}$	c $_{99}$	d $_{100}$	e $_{101}$	f $_{102}$	g $_{103}$	h $_{104}$	i $_{105}$	j $_{106}$	k $_{107}$	l $_{108}$	m $_{109}$	n $_{110}$	o $_{111}$	"60
’160	p $_{112}$	q $_{113}$	r $_{114}$	s $_{115}$	t $_{116}$	u $_{117}$	v $_{118}$	w $_{119}$	x $_{120}$	y $_{121}$	z $_{122}$	{ $_{123}$	| $_{124}$	} $_{125}$	˜ $_{126}$	¨ $_{127}$	"70
’200	Ł $_{128}$	' $_{129}$	’ $_{130}$	f $_{131}$	" $_{132}$	··· $_{133}$	† $_{134}$	‡ $_{135}$	ˆ $_{136}$	‰ $_{137}$	Š $_{138}$	‹ $_{139}$	Œ $_{140}$	Ž $_{141}$	ˆ $_{142}$	— $_{143}$	"80
’220	ł $_{144}$	‘ $_{145}$	’ $_{146}$	“ $_{147}$	” $_{148}$	• $_{149}$	– $_{150}$	— $_{151}$	˜ $_{152}$	TM $_{153}$	š $_{154}$	› $_{155}$	œ $_{156}$	ž $_{157}$	˜ $_{158}$	Ÿ $_{159}$	"90
’240	$_{160}$	¡ $_{161}$	¢ $_{162}$	£ $_{163}$	¤ $_{164}$	¥ $_{165}$	¦ $_{166}$	§ $_{167}$	¨ $_{168}$	© $_{169}$	ª $_{170}$	« $_{171}$	¬ $_{172}$	- $_{173}$	® $_{174}$	¯ $_{175}$	"A0
’260	° $_{176}$	± $_{177}$	² $_{178}$	³ $_{179}$	´ $_{180}$	µ $_{181}$	¶ $_{182}$	· $_{183}$	¸ $_{184}$	¹ $_{185}$	º $_{186}$	» $_{187}$	¼ $_{188}$	½ $_{189}$	¾ $_{190}$	¿ $_{191}$	"B0
’300	À $_{192}$	Á $_{193}$	Â $_{194}$	Ã $_{195}$	Ä $_{196}$	Å $_{197}$	Æ $_{198}$	Ç $_{199}$	È $_{200}$	É $_{201}$	Ê $_{202}$	Ë $_{203}$	Ì $_{204}$	Í $_{205}$	Î $_{206}$	Ï $_{207}$	"C0
’320	Ð $_{208}$	Ñ $_{209}$	Ò $_{210}$	Ó $_{211}$	Ô $_{212}$	Õ $_{213}$	Ö $_{214}$	× $_{215}$	Ø $_{216}$	Ù $_{217}$	Ú $_{218}$	Û $_{219}$	Ü $_{220}$	Ý $_{221}$	Þ $_{222}$	ß $_{223}$	"D0
’340	à $_{224}$	á $_{225}$	â $_{226}$	ã $_{227}$	ä $_{228}$	å $_{229}$	æ $_{230}$	ç $_{231}$	è $_{232}$	é $_{233}$	ê $_{234}$	ë $_{235}$	ì $_{236}$	í $_{237}$	î $_{238}$	ï $_{239}$	"E0
’360	ð $_{240}$	ñ $_{241}$	ò $_{242}$	ó $_{243}$	ô $_{244}$	õ $_{245}$	ö $_{246}$	÷ $_{247}$	ø $_{248}$	ù $_{249}$	ú $_{250}$	û $_{251}$	ü $_{252}$	ý $_{253}$	þ $_{254}$	ÿ $_{255}$	"F0
	"00	"01	"02	"03	"04	"05	"06	"07	"08	"09	"0A	"0B	"0C	"0D	"0E	"0F	

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz

Parametri

inclinazione	0,00000	x-height	4,30549 pt
spazio interparola	3,33332 pt	larghezza del quadrato	10,00000 pt
allungamento interparola	1,66665 pt	spazio extra	1,11111 pt
accorciamento interparola	1,11111 pt	corpo nominale	10,00000 pt
alfabeto maiuscolo	186,18060 pt	alfabeto minuscolo	127,58364 pt

caratteri latini, ma che vale per tutte le varianti del cirillico.

Per il greco il sistema TEX usa la collezione dei miei font CBfonts (Beccari, 2008b) mostrati nella tabella 1.8 della pagina 28; la codifica si chiama *LGR* (Local GReek encoding) diventata la codifica standard per il greco nel sistema TEX. Questa codifica ha una particolare corrispondenza con i caratteri latini, in modo che scrivendo in greco si può usare senza difficoltà una tastiera latina e non è necessario usare una tastiera greca politonica per accedere a tutti i glifi presenti nella codifica; per altro questa

Tabella 1.6 Font Latin Modern tondo con grazie di serie media in corpo 10 con codifica T5

	’00	’01	’02	’03	’04	’05	’06	’07	’10	’11	’12	’13	’14	’15	’16	’17	
’000	` 0	´ 1	ˆ 2	˜ 3	¨ 4	˙ 5	˚ 6	ˇ 7	˘ 8	¯ 9	̇ 10	̊ 11	̊ 12	’ 13	‹ 14	› 15	"00
’020	“ 16	” 17	” 18	« 19	» 20	– 21	— 22	23	₀ 24	₁ 25	Y̊ 26	ẙ 27	Ẏ 28	ẏ 29	Đ 30	đ 31	"10
’040	␣ 32	! 33	" 34	# 35	$ 36	% 37	& 38	’ 39	(40	) 41	* 42	+ 43	, 44	- 45	. 46	/ 47	"20
’060	0 48	1 49	2 50	3 51	4 52	5 53	6 54	7 55	8 56	9 57	: 58	; 59	< 60	= 61	> 62	? 63	"30
’100	@ 64	A 65	B 66	C 67	D 68	E 69	F 70	G 71	H 72	I 73	J 74	K 75	L 76	M 77	N 78	O 79	"40
’120	P 80	Q 81	R 82	S 83	T 84	U 85	V 86	W 87	X 88	Y 89	Z 90	[91	\ 92	] 93	ˆ 94	— 95	"50
’140	‘ 96	a 97	b 98	c 99	d 100	e 101	f 102	g 103	h 104	i 105	j 106	k 107	l 108	m 109	n 110	o 111	"60
’160	p 112	q 113	r 114	s 115	t 116	u 117	v 118	w 119	x 120	y 121	z 122	{ 123	| 124	} 125	˜ 126	- 127	"70
’200	À 128	Á 129	Ã 130	Å 131	Ą 132	Â 133	Ǎ 134	Ǻ 135	Ã 136	Ả 137	Ą 138	Ă 139	Ǎ 140	Ǻ 141	Ã 142	Ả 143	"80
’220	Ă 144	È 145	É 146	Ẽ 147	Ẻ 148	Ę 149	Ê 150	Ề 151	É 152	Ẽ 153	Ę 154	Ê 155	Ì 156	Í 157	Ĩ 158	Ỉ 159	"90
’240	à 160	á 161	ã 162	å 163	ą 164	â 165	ǎ 166	ǻ 167	ã 168	ả 169	ą 170	ă 171	ǎ 172	ǻ 173	ã 174	ả 175	"A0
’260	ă 176	è 177	é 178	ẽ 179	ẻ 180	ę 181	ê 182	ề 183	é 184	ẽ 185	ę 186	ê 187	ì 188	í 189	ĩ 190	ỉ 191	"B0
’300	Ị 192	Ò 193	Ó 194	Õ 195	Ỏ 196	Ọ 197	Ô 198	Ồ 199	Ó 200	Õ 201	Ọ 202	Ô 203	Ọ 204	Ò 205	Ó 206	Õ 207	"C0
’320	Ỏ 208	Ọ 209	Ù 210	Ú 211	Ũ 212	Ủ 213	Ụ 214	Ư 215	Ù 216	Ứ 217	Ũ 218	Ủ 219	Ụ 220	Ỳ 221	Ý 222	Ỹ 223	"D0
’340	ị 224	ò 225	ó 226	õ 227	ỏ 228	ọ 229	ô 230	ồ 231	ó 232	õ 233	ỏ 234	ô 235	ọ 236	ò 237	ó 238	õ 239	"E0
’360	ỏ 240	ọ 241	ù 242	ú 243	ũ 244	ủ 245	ụ 246	ư 247	ù 248	ứ 249	ũ 250	ủ 251	ụ 252	ỳ 253	ý 254	ỹ 255	"F0
	"00	"01	"02	"03	"04	"05	"06	"07	"08	"09	"0A	"0B	"0C	"0D	"0E	"0F	

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz

Parametri

inclinazione	0,00000	x-height	4,30549 pt
spazio interparola	3,33332 pt	larghezza del quadrato	10,00000 pt
allungamento interparola	1,66665 pt	spazio extra	1,11111 pt
accorciamento interparola	1,11111 pt	corpo nominale	10,00000 pt
alfabeto maiuscolo	186,18060 pt	alfabeto minuscolo	127,58364 pt

codifica viene selezionata automaticamente quando si invoca la lingua greca con `babel`, quindi l’utente non deve preoccuparsi di come sia la codifica.

Con le tabelle per i vari font che si sono esposte, si è anche mostrato che cosa si intende per *codifica dei font*; l’unica codifica che non si è mostrata è quella che viene definita UNICODE. Questa codifica, abbastanza recente ma già molto diffusa, viene usata essenzialmente per i font OpenType che contengono miriadi di caratteri; sarebbe quindi impossibile mostrare una tabella che stia in una pagina e che li contenga tutti.

Tabella 1.7 Font cirillico tondo con grazie di serie media in corpo 10

	’00	’01	’02	’03	’04	’05	’06	’07	’10	’11	’12	’13	’14	’15	’16	’17	
’000	Γ 0	Δ 1	Θ 2	Λ 3	Ξ 4	Π 5	Σ 6	Υ 7	Φ 8	Ψ 9	Ω 10	ff 11	fi 12	fl 13	ffi 14	ffl 15	"00
’020	ı 16	ȷ 17	` 18	´ 19	˘ 20	ˇ 21	¯ 22	˚ 23	¸ 24	ß 25	æ 26	œ 27	ø 28	Æ 29	Œ 30	Ø 31	"10
’040	˽ 32	! 33	” 34	# 35	$ 36	% 37	& 38	’ 39	(40	) 41	* 42	+ 43	, 44	- 45	. 46	/ 47	"20
’060	0 48	1 49	2 50	3 51	4 52	5 53	6 54	7 55	8 56	9 57	: 58	; 59	¡ 60	= 61	¿ 62	? 63	"30
’100	@ 64	A 65	B 66	C 67	D 68	E 69	F 70	G 71	H 72	I 73	J 74	K 75	L 76	M 77	N 78	O 79	"40
’120	P 80	Q 81	R 82	S 83	T 84	U 85	V 86	W 87	X 88	Y 89	Z 90	[91	“ 92	] 93	ˆ 94	˙ 95	"50
’140	‘ 96	a 97	b 98	c 99	d 100	e 101	f 102	g 103	h 104	i 105	j 106	k 107	l 108	m 109	n 110	o 111	"60
’160	p 112	q 113	r 114	s 115	t 116	u 117	v 118	w 119	x 120	y 121	z 122	– 123	— 124	˝ 125	˜ 126	¨ 127	"70
’200	А 128	Б 129	В 130	Г 131	Д 132	Е 133	Ж 134	З 135	И 136	Й 137	К 138	Л 139	М 140	Н 141	О 142	П 143	"80
’220	Р 144	С 145	Т 146	У 147	Ф 148	Х 149	Ц 150	Ч 151	Ш 152	Щ 153	Ъ 154	Ы 155	Ь 156	Э 157	Ю 158	Я 159	"90
’240	а 160	б 161	в 162	г 163	д 164	е 165	ж 166	з 167	и 168	й 169	к 170	л 171	м 172	н 173	о 174	п 175	"A0
’260	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	"B0
’300	192	193	194	195	¯ 196	¤ 197	198	199	200	201	202	203	204	205	206	207	"C0
’320	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	"D0
’340	р 224	с 225	т 226	у 227	ф 228	х 229	ц 230	ч 231	ш 232	щ 233	ъ 234	ы 235	ь 236	э 237	ю 238	я 239	"E0
’360	Ё 240	ё 241	Ґ 242	ґ 243	Є 244	є 245	І 246	і 247	Ї 248	ї 249	Ў 250	ў 251	№ 252	« 253	» 254	” 255	"F0
	"00	"01	"02	"03	"04	"05	"06	"07	"08	"09	"0A	"0B	"0C	"0D	"0E	"0F	

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz

Parametri

inclinazione	0,00000	x-height	4,30554 pt
spazio interparola	3,33332 pt	larghezza del quadrato	10,19999 pt
allungamento interparola	1,66665 pt	spazio extra	1,11111 pt
accorciamento interparola	1,11111 pt	corpo nominale	10,00000 pt
alfabeto maiuscolo	186,18093 pt	alfabeto minuscolo	127,58369 pt

Il concetto alla base della codifica UNICODE è che gli indirizzi dei font possono essere espressi in notazione esadecimale con 6 cifre esadecimali; il massimo numero di glifi che la tabella UNICODE può contenere può arrivare ad alcuni milioni. Non conosco nessun font che li contenga tutti; ma spessissimo i font OpentType contengono gli alfabeti latino, cirillico, greco, copto, ebraico, arabo e il cosiddetto alfabeto IPA (International Phonetic Alphabet) che contiene segni speciali per rappresentare i fonemi di una lingua, non più, quindi, i suoi grafemi. Spessissimo questi font OpenType contengono insiemi più o meno vasti di ideogrammi cinesi, giapponesi e

Tabella 1.8 Font greco tondo con grazie di serie media in corpo 10

	’00	’01	’02	’03	’04	’05	’06	’07	’10	’11	’12	’13	’14	’15	’16	’17	
’000	– 0	‿ 1	[illegible] 2	[illegible] 3	[illegible] 4	[illegible] 5	[illegible] 6	[illegible] 7	ı 8	Aı 9	Hı 10	Ωı 11	A 12	Ϋ 13	α 14	ϋ 15	"00
’020	′ 16	‵ 17	˩ 18	ˀ 19	˘ 20	ˀ 21	⊓ 22	ʎ 23	€ 24	‰ 25	ə 26	ˀ 27	‘ 28	’ 29	˘ 30	– 31	"10
’040	˝ 32	! 33	’ 34	ˊ 35	ˋ 36	% 37	˙ 38	′ 39	(40	) 41	* 42	+ 43	’ 44	- 45	· 46	/ 47	"20
’060	0 48	1 49	2 50	3 51	4 52	5 53	6 54	7 55	8 56	9 57	: 58	· 59	˚ 60	= 61	˚ 62	; 63	"30
’100	˝ 64	A 65	B 66	ˆ 67	Δ 68	E 69	Φ 70	Γ 71	H 72	I 73	Θ 74	K 75	Λ 76	M 77	N 78	O 79	"40
’120	Π 80	X 81	P 82	Σ 83	T 84	Υ 85	˝ 86	Ω 87	Ξ 88	Ψ 89	Z 90	[91	˝ 92	] 93	˝ 94	ˆ 95	"50
’140	` 96	α 97	β 98	ς 99	δ 100	ε 101	φ 102	γ 103	η 104	ι 105	ϑ 106	κ 107	λ 108	μ 109	ν 110	ο 111	"60
’160	π 112	χ 113	ρ 114	σ 115	τ 116	υ 117	118	ω 119	ξ 120	ψ 121	ζ 122	« 123	‘ 124	» 125	˜ 126	— 127	"70
’200	ὰ 128	ά 129	ᾶ 130	ᾶ 131	ὰ 132	ᾶ 133	ᾶ 134	ᾶ 135	ά 136	ᾶ 137	ᾶ 138	ᾶ 139	ᾶ 140	ᾶ 141	ᾶ 142	ᾶ 143	"80
’220	ᾶ 144	ᾶ 145	ᾶ 146	ᾶ 147	ᾶ 148	ᾶ 149	ᾶ 150	˘ 151	ὴ 152	ή 153	ῆ 154	155	ῆ 156	ῆ 157	ῆ 158	159	"90
’240	ή 160	ῆ 161	ῆ 162	ῆ 163	ή 164	ῆ 165	ῆ 166	ῆ 167	ῆ 168	ῆ 169	ῆ 170	ῆ 171	ῆ 172	ῆ 173	ῆ 174	ῆ 175	"A0
’260	ὼ 176	ώ 177	ὼ 178	ὼ 179	ὼ 180	ὼ 181	ὼ 182	ὼ 183	ώ 184	ὼ 185	ὼ 186	ὼ 187	ὼ 188	ὼ 189	ὼ 190	ὼ 191	"B0
’300	ῶ 192	ῶ 193	ῶ 194	F 195	ῶ 196	ῶ 197	ῶ 198	199	ὶ 200	ί 201	ὶ 202	ῖ 203	ὺ 204	ύ 205	ὺ 206	ῦ 207	"C0
’320	ί 208	ῖ 209	ῖ 210	ῖ 211	ύ 212	ῦ 213	ῦ 214	ῦ 215	ῖ 216	ῖ 217	ῖ 218	Ϊ 219	ῦ 220	ῦ 221	ῦ 222	Ϋ 223	"D0
’340	ὲ 224	έ 225	ἒ 226	ἒ 227	ὸ 228	ὸ 229	ὸ 230	ὸ 231	έ 232	ἔ 233	ἔ 234	ἒ 235	ό 236	ὸ 237	ὸ 238	ὸ 239	"E0
’360	ϋ 240	ῗ 241	΄ 242	ῗ 243	ϋ 244	ῦ 245	ύ 246	ῦ 247	α 248	η 249	ω 250	ῥ 251	ῥ 252	253	΄ 254	΄ 255	"F0
	"00	"01	"02	"03	"04	"05	"06	"07	"08	"09	"0A	"0B	"0C	"0D	"0E	"0F	

ΑΒΓΔΕΖΗΘΙΚΛΜΝΞΟΠΡΣΤΥΦΧΨΩ

αβγδεζηθικλμνξοπρσςτυφχψω

Parametri

inclinazione	0,00000	x-height	4,30450 pt
spazio interparola	3,33251 pt	larghezza del quadrato	9,99755 pt
allungamento interparola	1,66625 pt	spazio extra	1,11083 pt
accorciamento interparola	1,11083 pt	corpo nominale	10,00000 pt
alfabeto maiuscolo	171,90246 pt	alfabeto minuscolo	126,13586 pt

coreani[10]; inoltre possono contenere i glifi per le scritture indiane, tailandesi e di altre lingue più o meno note. Possono contenere anche i segni matematici e diversi segni analfabetici di vario genere. Si capisce bene che il numero di segni può davvero arrivare a parecchie decine di migliaia anche in un font non completo.

[10] I segni grafici del coreano assomigliano ad ideogrammi; sono invece delle sillabe intere i cui i piccoli segni componenti rappresentano suoni elementari e sono quindi delle lettere né più né meno che le lettere latine.

La tabella UNICODE è quindi divisa in pagine; la pagina 'zero' contiene segni solo nella prima metà e sono i segni della tabella ASCII; questa pagina è obbligatoriamente presente in ogni font; le pagine successive sono più o meno logicamente distinte per collezione di segni. Per usare questi font bisogna disporre di tastiere apposite, oppure bisogna riferirsi agli indirizzi dei singoli segni specificando i lunghi indirizzi esadecimali che individuano ogni segno.

Per semplificare questo fatto è stata creata una transcodifica chiamata UTF-8, un algoritmo che permette di riferirsi agli indirizzi UNICODE con un numero inferiore di cifre esadecimali almeno per i segni che si usano più spesso. Questa viene specificata con l'opzione *utf8* al pacchetto `inputenc`, ma a stretto rigore riguarda l'immissione dei caratteri nei file sorgente `.tex` più che una caratteristica dei font stessi. Infatti esistono pacchetti che consentono di indirizzare i glifi dei vari font greco, cirillico, eccetera, anche se questi font non hanno la codifica UNICODE; disponendo della tastiera giusta o adeguatamente transcodificata, non è difficile scrivere direttamente in greco nel file sorgente e comporre il testo con i font CBfonts con la codifica *LGR*. Come fare è descritto in altre guide, per esempio (Beccari e Gordini, 2016), quindi qui non se ne parla più.

1.4 Le famiglie dei font

Come si è già detto, ferma restando la codifica di un dato font, questo forma una collezione di caratteri distinti per famiglia, serie e forma; ogni famiglia è distinta da particolarità nel disegno dei font; per esempio i font CM, come gli EC e gli LM, sono divisi in tre o quattro famiglie per comporre il testo e in diverse altre famiglie per comporre la matematica.

1.4.1 Le famiglie dei font per comporre il testo

Le tre famiglie principali per comporre il testo sono quella dei font con grazie, o graziati; quella dei font senza grazie, o sgraziati, o lineari, o bastoncino, o *sans serif*; quella dei font teletype monospaziati. Alcune collezioni comprendono anche la famiglia dei font teletype proporzionali, dei font dunhill (con gli ascendenti particolarmente alti che assomigliano ai caratteri usati per il marchio dei famosi tabacchi omonimi), i funny font (font creati con parametri insoliti, in particolare inclinati all'indietro),

i *concrete font* (usati in particolare da Knuth per comporre il suo testo *Concrete Mathematics* (Knuth *et al.*, 1994)); eccetera.

I font con grazie

I font con grazie sono quelli usati in questo testo; le lettere maiuscole sono caratterizzate dalle *grazie* presenti alla fine di ogni asta diritta o curva che forma la lettera; nelle minuscole le grazie sono presenti essenzialmente alle estremità delle aste diritte, mentre i tratti curvi sono generalmente terminati mediante delle *gocce* o altre decorazioni sobrie, ma ben distinte; ogni carattere occupa in orizzontale uno spazio diverso in modo che i loro tratti non siano mai troppo ravvicinati al punto da costituire una macchia quasi nera.

I font senza grazie

I font senza grazie vengono usati in questo capoverso; le aste e i tratti delle lettere maiuscole e minuscole terminano senza nessuna visibile decorazione; alcuni font senza grazie possono avere le terminazioni leggermente svasate rispetto al tratto principale, ma generalmente questo tipo di decorazione non li fa passare nella categoria dei font graziati.

I font graziati sono più comodi da leggere per una lettura continua; i font senza grazie sono più facili da leggere per brevi tratti o per parole isolate; talvolta sono più facili da leggere in corpi molto piccoli.

I font a spaziatura fissa

`I font teletype hanno usi molto particolari; vengono usati specialmente per la scrittura dei termini informatici e, fra questi, per indicare gli indirizzi di rete o gli indirizzi di posta elettronica; nei libri tecnici sono usati per comporre il codice di programmi che vengono descritti e commentati. La particolarità di questi font a spaziatura fissa è che con questi font è impedita la cesura; le parole di un programma non possono essere divise in sillabe; se lo fossero il segno di cesura potrebbe essere frainteso per un operatore matematico.`

```
Con i font standard del sistema TEX, il font teletype è
particolarmente ben riuscito, ma non bisogna usarlo altro che
per gli scopi per i quali è stato previsto.
```

1.4.2 Le serie dei font

All'interno di ogni famiglia ogni font può essere disponibile in diverse serie; quelle normalmente presenti sono la serie (di nerezza) media e la serie nera; questa spessissimo è anche estesa, nel senso che ingrossando le aste per rendere più scuro ogni glifo, il disegnatore ha pensato bene di allargare ogni segno in modo che resti ancora una quantità discreta di spazio bianco fra le varie aste che formano un segno. Tuttavia questa estensione non è sempre presente e talvolta si può scegliere fra glifi di serie nera o di serie nera estesa. Non è sempre presente la serie nera, semplice o estesa, per ogni font di ogni famiglia: per esempio la famiglia dei font teletype monospaziati in generale non ha la serie nera. Lo stesso succede per alcune forme.

1.4.3 Le forme dei font

Ogni famiglia può contenere diverse forme di font; per esempio la famiglia dei font graziati contiene in generale il tondo e il corsivo, ma può contenere anche altre forme, spesso il maiuscoletto, talvolta il tondo inclinato, raramente il corsivo diritto; non tutte queste forme sono dotate di tutte le serie disponibili per il tondo e il corsivo. Per esempio, il maiuscoletto è una forma adatta ad evidenziare certe parole; anche la serie nera è adatta ad evidenziare certe parole; usare il maiuscoletto nero sembrerebbe un modo troppo forte per evidenziare certe parole; perciò il maiuscoletto generalmente è privo della serie nera, così come è privo della forma inclinata. Bene inteso esistono famiglie in cui queste combinazioni di forma e serie sono disponibili, ma non vi si può fare sempre affidamento; per documentarsi su queste disponibilità bisogna consultare la documentazione relativa ad ogni collezione di font; in mancanza bisogna aprire e leggere i file di descrizione dei font di cui si parlerà più avanti.

1.4.4 I corpi dei font

Che siano dotati di corpi ottici o siano disponibili in un solo corpo i font possono sempre essere scelti in diversi corpi; se il font dispone di corpi

ottici, i pedici e gli apici vengono composti bene, e i titoli come quelli dei frontespizi vengono composti in modo equilibrato.

Se il font è disponibile in un solo corpo il suo ingrandimento come il suo rimpicciolimento possono produrre difetti molto visibili quando i fattori di scala sono molto grandi o molto piccoli.

1.4.5 Le famiglie di font per comporre la matematica

La composizione della matematica richiede delle famiglie diverse di font che vengono usati in un modo completamente diverso da come vengono usati i font in modo testuale. Nel gergo di LaTeX 2_ε queste famiglie sono definite anche *math group* ma il nome famiglia sembra più adeguato (anche se può confondere il lettore) in quanto il comando primitivo del programma `tex` che serve per identificarle si chiama `\fam`, l'abbreviazione di *family*. Io preferirei usare il termine *gruppo* sottintendendo matematico

La caratteristica di ogni gruppo è che in matematica è costituita da tre corpi; uno uguale al corpo del testo serve per comporre la parte principale delle espressioni matematiche; un secondo, di solito pari al 70% del primo, serve per comporre gli apici e i pedici di primo livello; un terzo corpo di solito pari alla metà del primo, quindi al 70% del secondo, serve per comporre gli apici e i pedici di secondo livello, come in questa espressione:

$$V_{R_\mathrm{e}} = I_\mathrm{e} R_\mathrm{e}$$

Inoltre ogni gruppo matematica contiene sia la serie media sia la serie nera, anche se per comporre la matematica con la serie nera bisogna adottare "le maniere forti" cioè dichiarare di volere la serie nera prima di cominciare a scrivere l'espressione matematica; oppure, per comporre solo certi simboli con la serie nera, si può usare il pacchetto `bm` e il suo comando `\bm`.

Tuttavia i gruppi matematici sono diversi dalle famiglie testuali; un gruppo è quello degli operatori, per i quali si usano font in codifica *OT1*, di default i font CM tondi con grazie; un altro è quello per le lettere, che contiene gli alfabeti latino e greco corsivi con caratteristiche diverse dai corrispondenti caratteri testuali (tabella 1.9 della pagina 34): infatti il corsivo matematico è privo di legature, non riconosce lo spazio interparola, e gli accenti matematici vanno collocati mediante appositi comandi; un altro gruppo è quello per i simboli (tabella 1.10 della pagina 34) che mette a disposizione del compositore un centinaio di simboli e segni di operazione

che non si trovano nei font testuali e che comunque vanno trattati in modo matematico; infine un quarto gruppo di font per i grandi operatori e per i delimitatori estensibili (tabella 1.11 della pagina 35). Ognuno di questi gruppi è caratterizzato da una codifica diversa: come già detto il gruppo degli operatori ha la codifica *OT1*; invece quello delle lettere ha la codifica *OML*, quello per i simboli *OMS* e quello per i grandi operatori *OMX*. Se si usano i pacchetti `amsmath` e `amssymb` si accede anche ai font messi a disposizione dall'American Mathematical Society che costituiscono altri due gruppi con codifica *U*. Per le esigenze della matematica possono essere usati i caratteri maiuscoli calligrafici (contenuti nel file dei simboli); i font con calligrafia edwardiana o inglese; i fonti gotici; i font detti *neretto da lavagna* (in inglese *blackboard bold*); i font che contengono le lettere greche non inclinate per soddisfare le esigenze della tipografia francese; eccetera.

Quando si usano collezioni di font diverse da quelle standard, i pacchetti che ne permettono l'uso ridefiniscono i vari gruppi matematici, ma conservano le codifiche indicate sopra.

Va notato che i programmi di composizione del sistema TEX permettono di usare solo 16 famiglie matematiche o gruppi matematici, numerati da 0 a 15; il gruppo degli operatori è il gruppo zero, quello delle lettere il gruppo 1, quello dei simboli il gruppo 2, quello dei grandi operatori il gruppo 3; se si usano anche le famiglie dei simboli della American Mathematical Society (tabelle 1.12 della pagina 36 e 1.13 della pagina 36) si aggiungono altri due gruppi; se si desidera usare la collezione dei font Euler, si aggiungono altri gruppi. Tuttavia sembra impossibile superare il numero di 16; eppure a me è successo; quindi è necessario caricare i font matematici con grande oculatezza in modo da non caricare font inutili o doppioni (per esempio, non bisogna mai caricare i font speciali di LATEX209 con il pacchetto `latexsym` se si invoca il pacchetto `amssymb` che contiene già i simboli messi a disposizione dall'altro pacchetto) o altri font simili; queste sono piccole disattenzioni che potrebbero impedire un uso corretto dei gruppi matematici.

Personalmente ritengo che i font Euler siano molto belli, ma non siano adatti alla matematica, sebbene siano stati creati dal famoso font designer Hermann Zapf per l'uso che Knuth ne fece nel suo libro (di matematica) *Concrete Mathematics* (Knuth *et al.*, 1994). Le norme ISO prescrivono l'uso del corsivo per le variabili e il corsivo (matematico) è convenientemente inclinato rispetto al tondo e le varie lettere sono disegnate in modo

Tabella 1.9 Il font per le lettere matematiche in corpo 10

	'00	'01	'02	'03	'04	'05	'06	'07	'10	'11	'12	'13	'14	'15	'16	'17	
'000	Γ_{0}	Δ_{1}	Θ_{2}	Λ_{3}	Ξ_{4}	Π_{5}	Σ_{6}	Υ_{7}	Φ_{8}	Ψ_{9}	Ω_{10}	α_{11}	β_{12}	γ_{13}	δ_{14}	ϵ_{15}	"00
'020	ζ_{16}	η_{17}	θ_{18}	ι_{19}	κ_{20}	λ_{21}	μ_{22}	ν_{23}	ξ_{24}	π_{25}	ρ_{26}	σ_{27}	τ_{28}	υ_{29}	ϕ_{30}	χ_{31}	"10
'040	ψ_{32}	ω_{33}	ε_{34}	ϑ_{35}	ϖ_{36}	ϱ_{37}	ς_{38}	φ_{39}	$\leftharpoonup_{40}$	$\leftharpoondown_{41}$	$\rightharpoonup_{42}$	$\rightharpoondown_{43}$	$\smallfrown_{44}$	$\smallsmile_{45}$	$\triangleright_{46}$	$\triangleleft_{47}$	"20
'060	0_{48}	1_{49}	2_{50}	3_{51}	4_{52}	5_{53}	6_{54}	7_{55}	8_{56}	9_{57}	$.\,_{58}$	$,\,_{59}$	$<_{60}$	$/_{61}$	$>_{62}$	$\star_{63}$	"30
'100	∂_{64}	A_{65}	B_{66}	C_{67}	D_{68}	E_{69}	F_{70}	G_{71}	H_{72}	I_{73}	J_{74}	K_{75}	L_{76}	M_{77}	N_{78}	O_{79}	"40
'120	P_{80}	Q_{81}	R_{82}	S_{83}	T_{84}	U_{85}	V_{86}	W_{87}	X_{88}	Y_{89}	Z_{90}	$\flat_{91}$	$\natural_{92}$	$\sharp_{93}$	$\smile_{94}$	$\frown_{95}$	"50
'140	ℓ_{96}	a_{97}	b_{98}	c_{99}	d_{100}	e_{101}	f_{102}	g_{103}	h_{104}	i_{105}	j_{106}	k_{107}	l_{108}	m_{109}	n_{110}	o_{111}	"60
'160	p_{112}	q_{113}	r_{114}	s_{115}	t_{116}	u_{117}	v_{118}	w_{119}	x_{120}	y_{121}	z_{122}	$\imath_{123}$	$\jmath_{124}$	$\wp_{125}$	$\vec{}_{126}$	$\hat{}_{127}$	"70
	"00	"01	"02	"03	"04	"05	"06	"07	"08	"09	"0A	"0B	"0C	"0D	"0E	"0F	

Parametri

inclinazione	0,25000	x-height	4,30554 pt
spazio interparola	0,00000 pt	larghezza del quadrato	10,19999 pt
allungamento interparola	0,00000 pt	spazio extra	0,00000 pt
accorciamento interparola	0,00000 pt	corpo nominale	10,00000 pt

Tabella 1.10 Il font per i simboli matematici in corpo 10

	'00	'01	'02	'03	'04	'05	'06	'07	'10	'11	'12	'13	'14	'15	'16	'17	
'000	$-_{0}$	$\cdot_{1}$	$\times_{2}$	$*_{3}$	$\div_{4}$	$\diamond_{5}$	$\pm_{6}$	$\mp_{7}$	$\oplus_{8}$	$\ominus_{9}$	$\otimes_{10}$	$\oslash_{11}$	$\odot_{12}$	$\bigcirc_{13}$	$\circ_{14}$	$\bullet_{15}$	"00
'020	$\asymp_{16}$	$\equiv_{17}$	$\subseteq_{18}$	$\supseteq_{19}$	$\leq_{20}$	$\geq_{21}$	$\preceq_{22}$	$\succeq_{23}$	$\sim_{24}$	$\approx_{25}$	$\subset_{26}$	$\supset_{27}$	$\ll_{28}$	$\gg_{29}$	$\prec_{30}$	$\succ_{31}$	"10
'040	$\leftarrow_{32}$	$\rightarrow_{33}$	$\uparrow_{34}$	$\downarrow_{35}$	$\leftrightarrow_{36}$	$\nearrow_{37}$	$\searrow_{38}$	$\simeq_{39}$	$\Leftarrow_{40}$	$\Rightarrow_{41}$	$\Uparrow_{42}$	$\Downarrow_{43}$	$\Leftrightarrow_{44}$	$\nwarrow_{45}$	$\swarrow_{46}$	$\propto_{47}$	"20
'060	$\prime_{48}$	∞_{49}	$\in_{50}$	$\ni_{51}$	$\triangle_{52}$	∇_{53}	$/_{54}$	$'_{55}$	$\forall_{56}$	$\exists_{57}$	$\neg_{58}$	$\emptyset_{59}$	$\Re_{60}$	$\Im_{61}$	$\top_{62}$	$\bot_{63}$	"30
'100	$\aleph_{64}$	$\mathcal{A}_{65}$	$\mathcal{B}_{66}$	$\mathcal{C}_{67}$	$\mathcal{D}_{68}$	$\mathcal{E}_{69}$	$\mathcal{F}_{70}$	$\mathcal{G}_{71}$	$\mathcal{H}_{72}$	$\mathcal{I}_{73}$	$\mathcal{J}_{74}$	$\mathcal{K}_{75}$	$\mathcal{L}_{76}$	$\mathcal{M}_{77}$	$\mathcal{N}_{78}$	$\mathcal{O}_{79}$	"40
'120	$\mathcal{P}_{80}$	$\mathcal{Q}_{81}$	$\mathcal{R}_{82}$	$\mathcal{S}_{83}$	$\mathcal{T}_{84}$	$\mathcal{U}_{85}$	$\mathcal{V}_{86}$	$\mathcal{W}_{87}$	$\mathcal{X}_{88}$	$\mathcal{Y}_{89}$	$\mathcal{Z}_{90}$	$\cup_{91}$	$\cap_{92}$	$\uplus_{93}$	$\wedge_{94}$	$\vee_{95}$	"50
'140	$\vdash_{96}$	$\dashv_{97}$	$\lfloor_{98}$	$\rfloor_{99}$	$\lceil_{100}$	$\rceil_{101}$	$\{_{102}$	$\}_{103}$	$\langle_{104}$	$\rangle_{105}$	$\vert_{106}$	$\Vert_{107}$	$\updownarrow_{108}$	$\Updownarrow_{109}$	$\backslash_{110}$	$\wr_{111}$	"60
'160	$\surd_{112}$	$\amalg_{113}$	∇_{114}	$\int_{115}$	$\sqcup_{116}$	$\sqcap_{117}$	$\sqsubseteq_{118}$	$\exists_{119}$	$\S_{120}$	$\dagger_{121}$	$\ddagger_{122}$	$\P_{123}$	$\clubsuit_{124}$	$\diamondsuit_{125}$	$\heartsuit_{126}$	$\spadesuit_{127}$	"70
	"00	"01	"02	"03	"04	"05	"06	"07	"08	"09	"0A	"0B	"0C	"0D	"0E	"0F	

Parametri

inclinazione	0,25000	x-height	4,30554 pt
spazio interparola	0,00000 pt	larghezza del quadrato	10,19999 pt
allungamento interparola	0,00000 pt	spazio extra	0,00000 pt
accorciamento interparola	0,00000 pt	corpo nominale	10,00000 pt

Tabella 1.11 Il font per i grandi operatori e per i delimitatori estensibili in corpo 10

	'00	'01	'02	'03	'04	'05	'06	'07	'10	'11	'12	'13	'14	'15	'16	'17	
'000	(0	) 1	[2	] 3	⌊ 4	⌋ 5	⌈ 6	⌉ 7	{ 8	} 9	⟨ 10	⟩ 11	| 12	‖ 13	/ 14	\ 15	"00
'020	(16	) 17	(18	) 19	[20	] 21	⌊ 22	⌋ 23	⌈ 24	⌉ 25	{ 26	} 27	⟨ 28	⟩ 29	/ 30	\ 31	"10
'040	(32	) 33	[34	] 35	⌊ 36	⌋ 37	⌈ 38	⌉ 39	{ 40	} 41	⟨ 42	⟩ 43	/ 44	\ 45	/ 46	\ 47	"20
'060	(48	) 49	⌈ 50	⌉ 51	⌊ 52	⌋ 53	| 54	| 55	(56	) 57	⟨ 58	⟩ 59	{ 60	} 61	' 62	| 63	"30
'100	(64	) 65	| 66	| 67	⟨ 68	⟩ 69	⊔ 70	⊔ 71	∮ 72	∮ 73	⊙ 74	⊙ 75	⊕ 76	⊕ 77	⊗ 78	⊗ 79	"40
'120	∑ 80	∏ 81	∫ 82	∪ 83	∩ 84	⊎ 85	∧ 86	∨ 87	∑ 88	∏ 89	∫ 90	∪ 91	∩ 92	⊎ 93	∧ 94	∨ 95	"50
'140	∐ 96	∐ 97	⌢ 98	⌢ 99	⌢ 100	~ 101	~ 102	~ 103	⌈ 104	⌉ 105	⌊ 106	⌋ 107	⌈ 108	⌉ 109	{ 110	} 111	"60
'160	√ 112	√ 113	√ 114	√ 115	√ 116	| 117	‖ 118	‖ 119	↑ 120	↓ 121	↗ 122	↘ 123	↘ 124	↙ 125	⇑ 126	⇓ 127	"70
	"00	"01	"02	"03	"04	"05	"06	"07	"08	"09	"0A	"0B	"0C	"0D	"0E	"0F	

Parametri

inclinazione	0,00000	x-height	4,30554 pt
spazio interparola	0,00000 pt	larghezza del quadrato	10,19999 pt
allungamento interparola	0,00000 pt	spazio extra	0,00000 pt
accorciamento interparola	0,00000 pt	corpo nominale	10,00000 pt

sufficientemente diverso per distinguerle dal tondo prescritto, invece, per gli operatori e per le unità di misura. È vero che le norme ISO (ISO, 2009) si riferiscono alla matematica delle scienze (quelle sperimentali che usano grandezze misurabili) e della tecnologia; quindi a stretto rigore non si applicano agli scritti di matematica pura. Tuttavia, secondo me, è importantissimo distinguere i vari segni anche dalla loro forma, visto che la matematica si serve di un tipo di scrittura privo di ridondanza; secondo me i font Euler non sono sufficientemente inclinati e di disegno sufficientemente distinto da quello dei font tondi. Ma queste sono considerazioni (o raccomandazioni) che non hanno nulla a che vedere con i metodi con i quali i programmi

Tabella 1.12 Il font per la prima serie di simboli matematici dell'American Mathematical Society in corpo 10

	'00	'01	'02	'03	'04	'05	'06	'07	'10	'11	'12	'13	'14	'15	'16	'17	
'000	⊡ 0	⊞ 1	⊠ 2	□ 3	■ 4	▪ 5	◊ 6	⧫ 7	⟲ 8	⟳ 9	⥽ 10	⥾ 11	⊟ 12	⊩ 13	⊪ 14	⊨ 15	"00
'020	⇸ 16	⇷ 17	⇇ 18	⇉ 19	⇈ 20	⇊ 21	↾ 22	↿ 23	↿ 24	↿ 25	⥊ 26	⥋ 27	⥇ 28	⥉ 29	↰ 30	↱ 31	"10
'040	⇝ 32	⇜ 33	↫ 34	↬ 35	≐ 36	≻ 37	≳ 38	≳ 39	⊸ 40	∴ 41	∵ 42	÷ 43	≜ 44	≺ 45	≲ 46	≲ 47	"20
'060	≼ 48	≽ 49	≺ 50	≻ 51	≺ 52	≦ 53	≤ 54	≦ 55	∖ 56	- 57	≑ 58	≒ 59	≥ 60	≧ 61	≥ 62	≧ 63	"30
'100	⊏ 64	⊐ 65	⊳ 66	◁ 67	⊵ 68	⊴ 69	★ 70	◊ 71	▼ 72	▶ 73	◀ 74	→ 75	← 76	△ 77	▲ 78	▽ 79	"40
'120	≡ 80	≤ 81	≥ 82	≦ 83	≧ 84	¥ 85	⇒ 86	⇐ 87	✓ 88	≤ 89	⊼ 90	≡ 91	∠ 92	∡ 93	◁ 94	∝ 95	"50
'140	∪ 96	∩ 97	⋐ 98	⋑ 99	⋓ 100	⋒ 101	⋏ 102	⋎ 103	⋊ 104	⋉ 105	⊆ 106	⊇ 107	≏ 108	≎ 109	⋘ 110	⋙ 111	"60
'160	⌜ 112	⌝ 113	® 114	Ⓢ 115	⋔ 116	⊹ 117	∽ 118	≌ 119	⌞ 120	⌟ 121	✠ 122	∁ 123	⊤ 124	⊙ 125	⊛ 126	⊖ 127	"70
	"00	"01	"02	"03	"04	"05	"06	"07	"08	"09	"0A	"0B	"0C	"0D	"0E	"0F	

Parametri

inclinazione	0,00000	x-height	4,30554 pt
spazio interparola	0,00000 pt	larghezza del quadrato	10,19999 pt
allungamento interparola	0,00000 pt	spazio extra	0,00000 pt
accorciamento interparola	0,00000 pt	corpo nominale	10,00000 pt

Tabella 1.13 Il font per la seconda serie di simboli matematici dell'American Mathematical Society in corpo 10

	'00	'01	'02	'03	'04	'05	'06	'07	'10	'11	'12	'13	'14	'15	'16	'17	
'000	≨ 0	≩ 1	≰ 2	≱ 3	≮ 4	≯ 5	≮ 6	≯ 7	≨ 8	≩ 9	≰ 10	≱ 11	≤ 12	≥ 13	≠ 14	≠ 15	"00
'020	≺ 16	≻ 17	≺ 18	≻ 19	≠ 20	≠ 21	≠ 22	≠ 23	≺ 24	≻ 25	≺ 26	≻ 27	∼ 28	≠ 29	/ 30	\ 31	"10
'040	⊊ 32	⊋ 33	⊈ 34	⊉ 35	⊂ 36	⊃ 37	⊊ 38	⊋ 39	⊂ 40	⊃ 41	⊄ 42	⊅ 43	∦ 44	↑ 45	↓ 46	↕ 47	"20
'060	⊬ 48	⊮ 49	⊬ 50	⊮ 51	⊯ 52	⊯ 53	⋪ 54	⋫ 55	↚ 56	↛ 57	⇍ 58	⇏ 59	⇔ 60	↔ 61	∗ 62	∅ 63	"30
'100	∄ 64	𝔸 65	𝔹 66	ℂ 67	𝔻 68	𝔼 69	𝔽 70	𝔾 71	ℍ 72	𝕀 73	𝕁 74	𝕂 75	𝕃 76	𝕄 77	ℕ 78	ℚ 79	"40
'120	ℙ 80	ℚ 81	ℝ 82	𝕊 83	𝕋 84	𝕌 85	𝕍 86	𝕎 87	𝕏 88	𝕐 89	ℤ 90	⌒ 91	⌒ 92	⌒ 93	⌒ 94	95	"50
'140	∃ 96	∋ 97	98	99	100	101	℧ 102	∂ 103	≈ 104	⊐ 105	⊐ 106	⌉ 107	≺ 108	≻ 109	⋉ 110	⋊ 111	"60
'160	∣ 112	∥ 113	∖ 114	∼ 115	≈ 116	≈ 117	≻ 118	≺ 119	↶ 120	↷ 121	ⅎ 122	⅋ 123	𝕜 124	ℏ 125	ℏ 126	∍ 127	"70
	"00	"01	"02	"03	"04	"05	"06	"07	"08	"09	"0A	"0B	"0C	"0D	"0E	"0F	

Parametri

inclinazione	0,25000	x-height	4,62963 pt
spazio interparola	3,30000 pt	larghezza del quadrato	10,00000 pt
allungamento interparola	1,49993 pt	spazio extra	1,60000 pt
accorciamento interparola	1,60000 pt	corpo nominale	10,00000 pt

del sistema TEX accedono ai vari font. Se i font Euler piacciono e non si è vincolati dalle norme ISO, li si usi tranquillamente.

L'accesso ai font con LaTeX 2ε 2

Ma con tutti questi font, come fa ogni programma di composizione del sistema TeX ad accedere e a caricare i font necessari alla composizione? Ricordo anche che i programmi latex e pdflatex possono accedere a centinaia di font bitmapped o, meglio ancora, migliaia di font vettoriali distribuiti con l'installazione di TeXLive o di MiKTeX, e che xelatex e lualatex possono accedere anche alle centinaia di font OpenType presenti sul particolare calcolatore in uso, oltre che alla moltitudine di font OpenType installati con le distribuzioni TeXLive o MiKTeX.

Le modalità di accesso sono diverse per i font Type 1 e TrueType a 128 o a 256 caratteri con le codifiche descritte nel precedente capitolo, rispetto ai font OpenType testuali e matematici con la codifica UNICODE.

Caricare i caratteri OpenType con xelatex e lualatex è in un certo senso molto più facile che caricare i font più semplici di tipo Type 1 o TrueType a 128 o 256 caratteri; basta infatti usare le funzionalità del pacchetto `fontspec` per definire ogni font delle tre famiglie principali (con grazie, senza grazie e monospaziati); se esiste una difficoltà essa consiste nel fatto di caricare ogni famiglia di font specificando correttamente le caratteristiche del font che si desidera usare; i font OpenType, infatti, hanno di solito diverse varianti i cui glifi sono già registrati nello specifico file `.otf`; possono avere legature particolari, possono avere diverse serie di cifre arabe, da quelle per le tabelle (ugualmente spaziate per facilitare l'incolonnamento) a quelle "maiuscole" (a spaziatura individuale, non adatte all'incolonnamento), a quelle minuscole che si adattano in modo particolarmente elegante alla composizione di numeri nell'ambito del testo.

Spesso lo stesso font può essere usato anche per lingue che si scrivono con alfabeti diversi; il pacchetto `polyglossia`, che gestisce la composizione delle lingue, permette di definire il nome di una famiglia di font il cui

nome cominci con il nome della lingua, e automaticamente `polyglossia` usa quel font quando compone in quella lingua. Siccome con xelatex e lualatex tutto è codificato in ingresso e in uscita in UNICODE (o nella sua transcodifica UTF-8) non esistono problemi ad usare i font OpenType; ripeto: se c'è una difficoltà, è quella di conoscere quali siano le particolarità di un font specifico e come attivarle; la ricca e dettagliata documentazione del pacchetto `fontspec` permette di districarsi in queste difficoltà che nascono dall'enorme versatilità dei font OpenType.

2.1 L'accesso ai font TrueType

L'accesso ai font TrueType da parte di pdflatex è indiretto, nel senso che prima bisogna estrarre le informazioni metriche che in questo tipo di font non sono registrate in file separati. L'uso di questi font può implicare anche l'uso dei font virtuali. Nessun problema. Il programma ttf2tfm (Lemberg e Loyer, 2002) permette di generare sia un font virtuale di 256 caratteri, sia il corrispondente font metrico di cui abbisogna pdflatex; alternativamente si può usare il programma ttf2afm per generare un font virtuale e il corrispondente font Adobe font metric file (`.afm`) e poi usare il programma afm2tfm per ottenere il file metrico per i programmi di composizione del sistema TEX. Il font TrueType deve essere elencato nel file `pdftex.map` o `psfonts.map` o `dvipdfmx.map` a seconda che il font debba essere usato da pdflatex o da dvips o da dvipdfmx.

2.2 L'accesso ai font Type 1

Con la maggior parte dei font Type 1 le cose sono più semplici, perché a differenza dei font TrueType questi spesso contengono solo 256 glifi; al massimo sono disposti in un altro ordine nei file che ne contengono la descrizione grafica e i loro nomi sono associati a indirizzi nei file metrici `.afm`, che ne contengono le proprietà metriche, diversi da quelli che si aspettano i programmi del sistema TEX; per questo avendo per le mani un nuovo file Type 1 da installare sul proprio elaboratore per poterlo usare con il sistema TEX è necessario elaborare il file `.afm` per creare il corrispondente file `.tfm` ed eventualmente per creare un font virtuale i cui glifi siano gli stessi del font reale, ma i cui indirizzi siano disposti secondo l'ordine numerico che i programmi del sistema TEX si aspettano.

Ciò fatto bisogna aggiungere l'informazione nelle mappe dei font, come già indicato per i font TrueType, in modo che i programmi che ne fanno uso sappiano quale file `.tfm` corrisponda a quale font Type 1 e se ci sia necessità di usare un file di (ri)codifica per accedere ai glifi giusti nel font reale che ne contiene la descrizione grafica.

Per fortuna quasi tutti i font Type 1 liberi sono già inclusi nelle distribuzioni TeXLive e MiKTeX, quindi le operazioni indicate sopra – molto sommariamente, ma descritte un po' meglio nel prossimo capitolo – sono da eseguire molto raramente, se non mai. Si tenga presente che il file di mappa per il programma `pdftex`, `pdftex.map`, contiene (nel mio calcolatore) 12 445 righe delle quali pochissime sono righe di commento; direi che nella mia macchina, dove c'è una installazione completa di TeXLive, posso usare circa 12 000 diversi font Type 1. Sarà mai possibile che ne abbia bisogno di altri? Per la verità, sì: me ne sono disegnati alcuni che, a differenza dei CBfonts e di alcuni altri[1] caricati su CTAN, e quindi disponibili con ogni distribuzione completa del sistema TeX; ho creato alcuni altri font che non ho mai reso di pubblico dominio e quindi esistono solo e soltanto nel disco rigido del mio calcolatore. Con questo non voglio farmi passare per un esperto creatore di font; voglio solo dire che mi sono fatto una certa esperienza per renderli usabili come il sistema TeX richiede.

2.3 I font virtuali

Nel 1990 Knuth pubblicò un articolo sui font virtuali, (Knuth, 1990), indicandolo come un nuovo gioco per gli appassionati di TeX, anzi per i *grand wizard* del sistema TeX; le distribuzioni del sistema TeX furono contestualmente arricchite da una manciata di programmi necessari per giocare con i font virtuali. Sebbene non fossi affatto un *grand wizard*, ricordo di averci giocato anch'io quando non erano disponibili i bei pacchetti che sono disponibili oggi per usare font vettoriali in ogni lavoro di composizione contenente anche la composizione della matematica avanzata.

[1] Fra questi la collezione dei font LX, i font per le slide, derivati da quelli originariamente creati da Knuth, ma estesi completandoli anche con le famiglie matematiche, incluse quelle dell'American Mathematical Society. Io ora uso `beamer` per le mie presentazioni, ma anche con `beamer` uso sempre questi miei font LX (pacchetto `lxfonts`), (Beccari, 2008a).

Ero riuscito a creare i font virtuali per usare i font Times per il testo e per la matematica creando virtualmente tutti i file necessari per simulare tutte le famiglie, serie, forme per il testo e, per la matematica, usando i font Symbols virtualizzati. Il mio lavoro non era riuscito male, anzi; questi font virtuali furono usati per comporre e inviare articoli in formato *camera ready* a riviste scientifiche americane da parte mia e di diversi colleghi del nostro dipartimento, appassionati utenti di LaTeX; tutti usavamo già LaTeX209 da tempo e, successivamente, diventammo utenti di LaTeX 2_ε appena questo fu reso disponibile.

Un font virtuale è sostanzialmente un font dotato del suo bravo file metrico `.tfm` che però fa riferimento non a un file che contiene la descrizione grafica di ogni glifo, ma ad un file intermedio, (con lo stesso nome e l'estensione `.vf`), che verrà poi elaborato dal driver di visualizzazione per file in formato DVI, oppure dai programmi di trasformazione in formato `.ps` (dvips) o `.pdf` (dvipdfmx), perché sappiano che cosa debbano fare con i glifi virtuali indicati dagli indirizzi del file `.tfm`; questi font virtuali servono per lo stesso scopo anche quando si usi pdflatex che produce il file PDF direttamente senza trasformazioni intermedie.

In buona sostanza il font virtuale corrisponde ad un file dove per ogni indirizzo di glifo ci sono scritte in codice cose di questo genere: "per comporre questo glifo, prendi il glifo che si trova all'indirizzo x del file reale A"; oppure: "per comporre questo glifo prendi il glifo all'indirizzo x del font reale A, sollevalo di z punti, e mettici sotto quest'altro glifo preso all'indirizzo y del font reale B".

Sembra complicatissimo; in effetti scrivere correttamente queste cose a mano nel file `.vf` è fonte di uno stress non indifferente[2]; per fortuna oggi queste cose sono per lo più eseguite da programmi automatici, quindi i vari mal di pancia che derivano dalla creazione dei font virtuali se li sono già presi *una tantum* colui o coloro che hanno scritto i programmi. I mal di pancia restano se e solo se uno vuol crearsi un font virtuale molto strano per il quale non esistono programmi generali già pronti all'uso.

[2]In realtà il file `.vf` è scritto in linguaggio macchina; il file che si può costruire, editare e modificare a mano ha estensione `.vpl` ed è scritto in caratteri ASCII, quindi è direttamente leggibile dall'uomo; il linguaggio usato nei file `.vpl` ricorda un po' la programmazione in Assembly language, rispetto alla programmazione in linguaggio macchina, ma il vantaggio è minimo; completatolo, lo si trasforma in binario mediate il programma vptovf.

Tuttavia è importante sapere non come creare, ma che cosa siano i font virtuali e che i file che descrivono i font virtuali già esistenti sono raccolti nella cartella `.../fonts/vf/` dell'albero principale della propria distribuzione del sistema TeX.

2.4 Il file di descrizione dei font

Tutto ciò è necessario ma in un certo senso poco importante agli occhi dell'utente medio, specialmente se i file `.tfm`, `.vf`, `.pfb`, `.ttf`, eccetera, sono già presenti e installati nelle apposite cartelle. Mancano ancora i file di descrizione dei font, i file delle mappe generali e, volendo, i file di estensione `.sty` che agevolano l'uso dei font.

I rari font bitmapped hanno bisogno dei file `.fd` ma non richiedono l'uso delle mappe; richiedono solo la disponibilità dei file metrici `.tfm` e dei file `pk`; questi sono file il cui nome è generalmente costituito dal nome vero del font (per esempio *cmr10*), con una estensione formata agglutinando il numero, che indica la densità di pixel al pollice per i quali sono stati creati, con il suffisso `pk`; nell'esempio, il font *cmr10* "rasterizzato" (elaborato per creare la matrice di pixel per ogni glifo) per lavorare su una stampante da 600 dpi, avrà il nome `cmr10.600pk`. L'utente non deve preoccuparsi di questo file e di dove metterlo; se, come sempre accade, il font può essere creato con METAFONT, quando il file rasterizzato del font non fosse già registrato sul disco rigido al posto giusto, ci pensa il programma stesso di composizione, per esempio latex, a lanciare METAFONT e i programmi accessori e a registrare il file rasterizzato con il nome giusto nella cartella giusta. Il tutto è automatico e viene svolto *una tantum*, quindi una volta che il file del font rasterizzato è stato creato e registrato sul disco rigido, l'utente non noterà più nessun rallentamento nell'elaborazione dei suoi documenti.

Può succedere invece che si voglia usare un font Type 1 e che si sbagli a scriverne il nome; oppure che si voglia usare un font Type 1 il cui nome non compaia ancora nel file delle mappe. In questi casi si vede nella console del programma di composizione un messaggio relativo al fatto che il programma di composizione ha lanciato METAFONT, ma che questo non ha trovato i file sorgente con estensione `.mf` per generare il font rasterizzato, e che quindi non può proseguire la composizione; o meglio, il programma di composizione prosegue sostituendo il font richiesto con un altro font, ma

scrive il nome del file mancante nel file accessorio `missfont.log` nella stessa cartella dove risiede il file principale del proprio documento. Se dovesse succedere questo "infortunio" si può aprire il file `missfont.log` e vedere quale o quali font il programma non è riuscito a creare, così da scoprire l'errore, che sia di ortografia oppure di mancato aggiornamento dei file delle mappe; per aggiornare i file generali delle mappe si veda più avanti il paragrafo 3.3.4 della pagina 70.

2.5 Accesso ai font tramite i loro file di descrizione

Dicevo all'inizio che i paragrafi precedenti erano necessari ma non importanti all'occhio dell'utente medio; diventano importanti quando si debba cercare di scoprire certi errori. In effetti anche se quei passi preliminari sono tutti stati svolti correttamente, i file veramente importanti per accedere con una data codifica a certe famiglie, alle loro serie, alle loro forme e ai loro corpi sono i file di descrizione dei font con estensione `.fd`.

Questi sono file il cui nome, in tutte lettere minuscole, è costituito agglutinando il nome della codifica del font con il nome della famiglia e aggiungendoci l'estensione `.fd`. Semplice, sì, ma che cosa contengono? Contengono per ogni forma e serie i corpi che possono essere usati e i nomi dei font reali o virtuali da cui trarre le informazioni geometriche per il loro disegno. Merita esaminare un file, per esempio il file `ot1cmr.fd` relativo alla famiglia CM di font con grazie con la codifica *OT1*.

```
%%
%% This is file ‘ot1cmr.fd’,
%% generated with the docstrip utility.
%%
%% The original source files were:
%%
%% cmfonts.fdd  (with options: ‘OT1cmr’)
%%
%% This is a generated file.
%%
%% The source is maintained by the LaTeX Project team and bug
%% reports for it can be opened at https://latex-project.org/bugs.html
%% (but please observe conditions on bug reports sent to that address!)
%%
%%
%% Copyright (C) 1993-2020
```

```
%% The LaTeX3 Project and any individual authors listed elsewhere
%% in this file.
%%
%% This file was generated from file(s) of the LaTeX base system.
%% ------------------------------------------------------------
%%
%% It may be distributed and/or modified under the
%% conditions of the LaTeX Project Public License, either version 1.3c
%% of this license or (at your option) any later version.
%% The latest version of this license is in
%%    https://www.latex-project.org/lppl.txt
%% and version 1.3c or later is part of all distributions of LaTeX
%% version 2008 or later.
%%
%% This file may only be distributed together with a copy of the LaTeX
%% base system. You may however distribute the LaTeX base system without
%% such generated files.
%%
%% The list of all files belonging to the LaTeX base distribution is
%% given in the file `manifest.txt'. See also `legal.txt' for additional
%% information.
%%
%% In particular, permission is granted to customize the declarations in
%% this file to serve the needs of your installation.
%%
%% However, a modified version of this file under its original name
%% should not be distributed as part of a standard LaTeX distribution, as
%% the modification will be nontransparent for the user of that
%% distribution (and thus violating clause 6a of the LPPL license),
%% making successful document exchange impossible.
%%
\ProvidesFile{ot1cmr.fd}
        [2019/12/16 v2.5j Standard LaTeX font definitions]
\DeclareFontFamily{OT1}{cmr}{\hyphenchar\font45 }
\DeclareFontShape{OT1}{cmr}{m}{n}%
     {<5><6><7><8><9><10><12>gen*cmr%
      <10.95>cmr10%
      <14.4>cmr12%
      <17.28><20.74><24.88>cmr17}{}
\DeclareFontShape{OT1}{cmr}{m}{sl}%
     {%
      <5><6><7>cmsl8%
      <8><9>gen*cmsl%
      <10><10.95>cmsl10%
      <12><14.4><17.28><20.74><24.88>cmsl12%
```

```
      }{}
\DeclareFontShape{OT1}{cmr}{m}{it}%
     {%
      <5><6><7>cmti7%
      <8>cmti8%
      <9>cmti9%
      <10><10.95>cmti10%
      <12><14.4><17.28><20.74><24.88>cmti12%
      }{}
\DeclareFontShape{OT1}{cmr}{m}{sc}%
     {%
      <5><6><7><8><9><10><10.95><12>%
      <14.4><17.28><20.74><24.88>cmcsc10%
      }{}
% Warning: please note that the upright shape below is
%          used for the \pounds symbol of LaTeX. So this
%          font definition shouldn't be removed.
%
\DeclareFontShape{OT1}{cmr}{m}{ui}
   {
      <5><6><7><8><9><10><10.95><12>%
      <14.4><17.28><20.74><24.88>cmu10%
      }{}
%%%%%%% bold series
\DeclareFontShape{OT1}{cmr}{b}{n}
     {%
      <5><6><7><8><9><10><10.95><12>%
      <14.4><17.28><20.74><24.88>cmb10%
      }{}
%%%%%%%% bold extended series
\DeclareFontShape{OT1}{cmr}{bx}{n}
   {%
      <5><6><7><8><9>gen*cmbx%
      <10><10.95>cmbx10%
      <12><14.4><17.28><20.74><24.88>cmbx12%
      }{}
\DeclareFontShape{OT1}{cmr}{bx}{sl}
      {%
      <5><6><7><8><9>%
      <10><10.95><12><14.4><17.28><20.74><24.88>cmbxsl10%
      }{}
\DeclareFontShape{OT1}{cmr}{bx}{it}
      {%
      <5><6><7><8><9>%
      <10><10.95><12><14.4><17.28><20.74><24.88>cmbxti10%
```

```
107        }{}
108 % Again this is necessary for a correct \pounds symbol in
109 % the cmr fonts Hopefully the dc/ec font layout will take
110 % over soon.
111 %
112 \DeclareFontShape{OT1}{cmr}{bx}{ui}
113        {<->sub*cmr/m/ui}{}
114 \endinput
115 %%
116 %% End of file 'ot1cmr.fd'.
```

Le prime 39 righe, di cui una, quella con la lista degli anni coperti da copyright, esce nel margine[3], sono solo commenti iniziali. Le righe 40 e 41 dicono di quale file si tratti con la solita sintassi raccomandata per tutti i file di ogni genere, anche per i file sorgente del proprio documento.

Le linee 45 e 46 specificano per il file in oggetto la data, il numero di versione e una brevissima descrizione.

La riga 47 dichiara la codifica e il nome della famiglia, aggiungendo l'informazione supplementare che per la cesura si deve usare il carattere con indirizzo 45 al fine di rappresentare il trattino di "a capo"; se l'indirizzo fosse -1 vorrebbe dire che nessun glifo è adatto a indicare il punto di cesura, e che quindi non si può eseguire la cesura; è questo il valore che viene impostato nel corrispondente file per la famiglia dei font teletype per impedirne la cesura.

Le righe da 48 a 52 dicono quali corpi sono disponibili per la forma tonda con grazie diritta e di serie media; la scrittura è un po' oscura, ma vuol dire questo:

- la riga 49, `{<5><6><7><8><9><10><12>gen*cmr%`, dice sostanzialmente: prendi il disegno dei glifi per i vari corpi dai file il cui nome si ottiene agglutinando il prefisso `cmr` al valore del corpo racchiuso fra parentesi uncinate: il corpo 5 dal file `cmr5` fino al corpo 12 dal file `cmr12` (si tratta dei file metrici, quindi l'estensione sottintesa è `.tfm`);
- la riga 50, `<10.95>cmr10%`, dice: disegna il corpo 11 (in realtà il corpo 10,95) ingrandendo di quel che occorre il font di corpo 10 descritto nel file `cmr10`;

[3] Trattandosi di un file di sistema, non ho voluto editarlo per rendere le righe adatte a questa composizione.

- la riga 51, `<14.4>cmr12%`, dice: disegna il corpo 14,4 ingrandendo di quel che occorre il font di corpo 12 descritto dal file `cmr12`;
- la riga 52, `<17.28><20.74><24.88>cmr17}{}`, dice: disegna i corpi 17,28, 20,74 e 24,88 ingrandendo di quel che occorre il font di corpo 17,28 descritto nel file `cmr17`.

Informazioni analoghe sono indicate per le altre forme della famiglia `cmr`. Merita rilevare che le letterine semplici o accoppiate che compaiono negli argomenti del comando `\DeclareFontShape` sono esattamente le stringhe di lettere da usare nei comandi di basso livello che servono per selezionare un font con il New Font Selection Scheme (NFSS), non più tanto nuovo, che fu introdotto nel 1994 nell'allora nuovo LaTeX 2_ε: `\fontencoding` (dove la codifica va espressa in lettere maiuscole come indicato nel file); `\fontfamily` dove per tutte le forme di questa famiglia va indicato *cmr*; `\fontseries` dove le serie disponibili sono *m*, *b* e *bx*, la serie media, la serie nera e la serie nera estesa; `\fontshape` dove le forme disponibili sono *n* (tondo normale o diritto), *sl* (tondo inclinato), *it* (corsivo), *sc* (maiuscoletto), *ui* (corsivo diritto; secondo il commento servirebbe solo per comporre il simbolo `\pounds` per produrre il segno £).

Si notino infine le righe 109 e 110 dove per la serie nera estesa del corsivo diritto si esegue una sostituzione silenziosa (`ssub`) per tutti i corpi; tutti vengono sostituiti con i corrispondenti corpi della serie media.

Per un font nella codifica *T1* come il Latin Modern, ci troviamo invece in una situazione assai diversa:

```
% This file belongs to the Latin Modern package. The work is released
% under the GUST Font License. See the MANIFEST-Latin-Modern.txt and
% README-Latin-Modern.txt files for the details. For the most recent version of
% this license see http://www.gust.org.pl/fonts/licenses/GUST-FONT-LICENSE.txt
% or http://tug.org/fonts/licenses/GUST-FONT-LICENSE.txt

\ProvidesFile{t1lmr.fd}[2009/10/30 v1.6 Font defs for Latin Modern]
\DeclareFontFamily{T1}{lmr}{}
\DeclareFontShape{T1}{lmr}{m}{n}%
     {<-5.5>    ec-lmr5     <5.5-6.5> ec-lmr6
      <6.5-7.5> ec-lmr7     <7.5-8.5> ec-lmr8
      <8.5-9.5> ec-lmr9     <9.5-11>  ec-lmr10
      <11-15>   ec-lmr12
      <15-> ec-lmr17
      }{}
\DeclareFontShape{T1}{lmr}{m}{sl}%
```

```
17      {<-8.5>    ec-lmro8    <8.5-9.5> ec-lmro9
18       <9.5-11>  ec-lmro10   <11-15>   ec-lmro12
19       <15-> ec-lmro17
20       }{}
21 \DeclareFontShape{T1}{lmr}{m}{it}%
22      {<-7.5>    ec-lmri7
23       <7.5-8.5> ec-lmri8    <8.5-9.5> ec-lmri9
24       <9.5-11>  ec-lmri10   <11->   ec-lmri12
25       }{}
26 \DeclareFontShape{T1}{lmr}{m}{sc}%
27      {<-> ec-lmcsc10}{}
28 \DeclareFontShape{T1}{lmr}{m}{ui}%
29      {<-> ec-lmu10}{}
30 %
31 % Is this the right 'shape'?:
32 \DeclareFontShape{T1}{lmr}{m}{scsl}%
33      {<-> ec-lmcsco10}{}
34 %%%%%%% bold series
35 \DeclareFontShape{T1}{lmr}{b}{n}
36      {<-> ec-lmb10}{}
37 \DeclareFontShape{T1}{lmr}{b}{sl}
38      {<-> ec-lmbo10}{}
39 %%%%%%% bold extended series
40 \DeclareFontShape{T1}{lmr}{bx}{n}
41      {<-5.5>   ec-lmbx5      <5.5-6.5> ec-lmbx6
42       <6.5-7.5> ec-lmbx7      <7.5-8.5> ec-lmbx8
43       <8.5-9.5> ec-lmbx9      <9.5-11>  ec-lmbx10
44       <11->   ec-lmbx12
45       }{}
46 \DeclareFontShape{T1}{lmr}{bx}{it}
47      {<-> ec-lmbxi10}{}
48 \DeclareFontShape{T1}{lmr}{bx}{sl}
49      {<-> ec-lmbxo10}{}
50 %%%%%%% Font/shape undefined, therefore substituted
51 \DeclareFontShape{T1}{lmr}{b}{it}
52      {<->sub * lmr/b/sl}{}
53 \endinput
54 %%
55 %% End of file `t1lmr.fd'.
```

Le righe da 1 a 6 sono i soliti commenti iniziali, mentre la riga 7 è la solita riga con la quale si identifica il file, con la sua data, la versione e la breve descrizione; la riga 8 dichiara la famiglia e le righe successive dichiarano i corpi disponibili per ogni forma e serie; si noti la diversa

sintassi, per esempio per la forma tonda diritta della serie media:

```
\DeclareFontShape{T1}{lmr}{m}{n}%
     {<-5.5>    ec-lmr5     <5.5-6.5> ec-lmr6
      <6.5-7.5> ec-lmr7     <7.5-8.5> ec-lmr8
      <8.5-9.5> ec-lmr9     <9.5-11>  ec-lmr10
      <11-15>   ec-lmr12
      <15-> ec-lmr17
     }{}
```

Vi compaiono degli intervalli di corpo: `<-5.5>` indica l'intervallo di corpo da "piccolissimo" (idealmente zero) fino al corpo 5,5 escluso; `<5.5-6.5>` indica l'intervallo di corpo da 5,5 incluso a 6,5 escluso; e via di questo passo fino a `<15->` che indica l'intervallo di corpo da 15 incluso in su. In sostanza tutti i corpi possono essere rappresentati, anche il corpo 3,2 se interessasse un corpo così piccolo, o 61,7 se interessasse un corpo così grande. Ogni corpo che non corrisponda al corpo di disegno del file indicato viene ottenuto con un (generalmente piccolo) ingrandimento o rimpicciolimento di un font reale esistente.

Nelle righe 26 e 27, relative al maiuscoletto, è indicato l'intervallo `<->`; questo intervallo senza estremi indica "per qualsiasi corpo"; questa notazione è ripetuta per il corsivo diritto, per il maiuscoletto inclinato, per la serie nera non estesa, per il corsivo nero esteso, per il tondo inclinato nero esteso; per il corsivo nero non esteso viene richiesta la sostituzione con il tondo inclinato nero non esteso. La sostituzione eseguita con il comando `sub` non è silenziosa, per cui verrà indicata questa operazione sia nel file `.log` sia nella console.

In sostanza i font Latin Modern sono dotati di corpi ottici ma sono trattabili con una possibilità di selezione di corpo continua da valori piccolissimi a valori grandissimi, non con una legge lineare, bensì con una legge lineare a tratti che permette di usare i corpi ottici con ingrandimenti o rimpicciolimenti (generalmente) piccoli così da non rovinarne le caratteristiche di leggibilità.

Si consideri la figura 2.1 della prossima pagina; a sinistra c'è la descrizione della relazione discreta fra i corpi desiderati e quelli effettivamente ottenuti con le definizioni del file di descrizione `ot1cmr.fd`; se si specificasse con i comandi di base un corpo diverso da quelli ottenibili, l'algoritmo

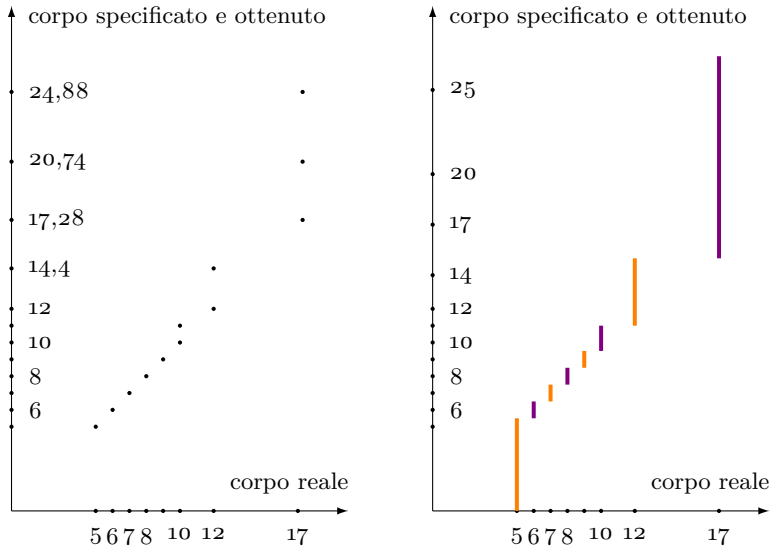


Figura 2.1 Corpi specificati e ottenuti; a sinistra secondo le definizioni presenti nel file `ot1cmr.fd`, a destra secondo quanto specificato nel file `t1lmr.fd`.

metterebbe a disposizione solo il corpo di valore più vicino a quello desiderato. Invece con le definizioni del file di descrizione `t1lmr.fd` qualunque corpo si desideri lo si ottiene, perché la relazione fra i corpi desiderati e quelli ottenibili è continua. L'unica differenza, messa in evidenza con i colori alternati delle barre, è che il font fisico usato mediante ingrandimento o rimpicciolimento è diverso per ogni banda di colore, per cui il disegno degli stessi glifi non è proporzionalmente uguale. I corpi ottici si fanno vedere ancora, ma si possono ottenere i font di qualsiasi corpo; il buon senso raccomanda di non richiedere mai corpi microscopici, né corpi giganteschi; il tutto si basa sul fatto che i corpi ottici possono venire ingranditi o rimpicciolti di poco, altrimenti si rovinano le loro caratteristiche.

La linearità a tratti si manifesta anche ad occhio nudo se si confrontano due corpi molto vicini, per esempio 14,999 e 15 a cavallo di uno degli estremi degli intervalli:

M a confronto con M

La discontinuità nello spessore delle aste e nella larghezza del glifo è visibile ad occhio nudo.

Secondo me, la descrizione dei font Latin Modern, che esistono solo nella forma vettoriale, è la forma migliore per sfruttare i corpi ottici, nello stesso tempo sfruttando la possibilità di una selezione praticamente continua del corpo dei font da usare.

Merita sottolineare che l'esistenza di font dotati di corpi ottici è un fatto raro; i font nati per il sistema TEX, i CM, i CM-super e gli LM sono fra i pochi font esistenti in generale dotati di corpi ottici e in numero così elevato per ciascuna serie e forma. La quasi totalità dei 12 000 font presenti nella mia installazione di TEXLive per Mac, sia Type 1, sia TrueType, sia OpenType sono privi di corpi ottici e ottengono i font di corpo maggiore o quelli di corpo minore ingrandendo o rimpicciolendo i glifi di un solo corpo; il risultato è quello discusso in questo paragrafo: i corpi minori di 10 pt diventano via via più esili al diminuire del corpo e in mancanza di *hint* i tratti più sottili possono svanire nella stampa; i corpi maggiori diventano via via più scuri all'aumentare del corpo. Usando il solo corpo 10 pt e lavorando solo con corpi nell'intervallo 8 pt–12 pt gli inconvenienti sono minimi, ma fuori di questo intervallo la resa lascia a desiderare. Questa osservazione è particolarmente importante per i corpi minori che vengono usati molto in matematica.

Come fa il programma di composizione a conoscere l'esistenza dei file di descrizione dei font? Semplicemente li legge quando è necessario generandone il nome mediante l'agglutinazione della codifica (in lettere minuscole), il nome della famiglia e l'estensione `.fd`. Supponiamo che in questo momento io voglia usare il font Type 1 Helvetica, e supponiamo che io conosca il nome dalla famiglia `phv` a cui esso appartiene; in questo momento sto componendo con la codifica T1; quindi se io specificassi `\fontfamily{phv}\selectfont` il programma riconoscerebbe di non sapere nulla del font che voglio usare; allora il programma genererebbe il nome del file `t1phv.fd` e ne ordinerebbe la lettura; a quel punto saprebbe come usare il font Helvetica e comporrebbe quello che occorre con il font selezionato: infatti questo breve brano di testo è composto con il font Helvetica. Incidentalmente si nota subito che non è il caso di mescolare i font a casaccio; qui il font Helvetica dimostra di essere più scuro del testo circostante e, a pari corpo, di avere un occhio[4]

[4] L'*occhio* è l'altezza delle lettere minuscole senza ascendenti.

Tabella 2.1 La stessa frase composta con vari font. Si noti che sono tutte composte in corpo 11, ma i diversi font presentano occhi di diverse altezze e ogni frase è più o meno lunga a seconda della larghezza dell'alfabeto. Si osservino inoltre le differenze nella forma delle grazie, nell'inclinazione degli accenti e nella loro posizione fra font CM e LM. Gli esempi composti con le collezioni TeX Gyre Sono identificati dai loro nomi preceduti da TG; essi sono collocati nella tabella sotto ai font di cui sono cloni ampliati.

Font	Esempio
Avantgarde	La città è più pulita, perciò è così bella!
TG Adventor	La città è più pulita, perciò è così bella!
Bookman	La città è più pulita, perciò è così bella!
TG Bonum	La città è più pulita, perciò è così bella!
Chancery	*La città è più pulita, perciò è così bella!*
TG Chorus	*La città è più pulita, perciò è così bella!*
Charter	La città è più pulita, perciò è così bella!
Courier	La città è più pulita, perciò è così bella!
TG Cursor	La città è più pulita, perciò è così bella!
Helvetica	La città è più pulita, perciò è così bella!
TG Heros	La città è più pulita, perciò è così bella!
Garamond (EB)	La città è più pulita, perciò è così bella!
Garamond (URW)	La città è più pulita, perciò è così bella!
CM roman	La città è più pulita, perciò è così bella!
LM roman	La città è più pulita, perciò è così bella!
LM italic	*La città è più pulita, perciò è così bella!*
LM sans serif	La città è più pulita, perciò è così bella!
LM tt-mono	La città è più pulita, perciò è così bella!
LM tt-prop	La città è più pulita, perciò è così bella!
New Century	La città è più pulita, perciò è così bella!
TG Schola	La città è più pulita, perciò è così bella!
Palatino	La città è più pulita, perciò è così bella!
TG Pagella	La città è più pulita, perciò è così bella!
Palladio	La città è più pulita, perciò è così bella!
Times	La città è più pulita, perciò è così bella!
TG Termes	La città è più pulita, perciò è così bella!
Utopia	La città è più pulita, perciò è così bella!

visibilmente più grande. Vedi la tabella 2.1 nella pagina precedente.

Si notano in particolare i cloni della collezione TeX Gyre, marcati con il prefisso TG. Sono quasi identici agli originali, composti, per ciascuno, nella riga immediatamente superiore; se ne consiglia l'uso al posto degli originali perché cono stati creati con il sistema TEX in mente dal gruppo di lavoro polacco GUST; sono esplicitamente creati per il sistema TEX e sono disponibili nell'installazione di TEXLive sia come font Type 1 (per pdflatex), sia come font OpenType (per lualatex e xelatex) e sono corredati anche dei corrispondenti font matematici; in sostanza sono completi.

In questa tabella si vedono anche due diverse interpretazioni del font Garamond; esistono infatti font con lo stesso nome che, pur assomigliandosi, hanno caratteristiche ben diverse. Si osservino anche gli esempi creati con i font delle collezioni TeX Gyre; questi font vettoriali sono stati creati dal gruppo polacco degli utenti di TEX; imitano molto bene le caratteristiche di noti font commerciali, ma sono creati apposta leggermente diversi non per eludere i diritti di copyright, ai quali i gruppi di utenti sono sempre molto attenti, quanto per adattarli alle funzionalità del sistema TEX, in particolare alle varie codifiche che pdflatex conosce e gestisce.

La scelta dei font da usare simultaneamente in un dato testo deve quindi seguire regole estetiche ben precise in modo da evitare situazioni come quella esposta sopra. Ci sarebbero diversi testi da consultare; tra questi c'è il breve manuale di Wilson (2014) scritto da un tipografo che è anche un utente di LATEX. Ci si può istruire sui font in generale leggendo il breve testo storico divulgativo (Garfield, 2012). Una formazione molto più profonda si può avere leggendo il testo di Robert Bringhurst (Bringhurst, 2004) del quale esiste anche una versione in italiano[5].

2.6 Scelta dei font matematici

I font matematici sono difficili da selezionare e da configurare; i font matematici, così ricchi di glifi e di caratterisitche, sono una specialità del sistema TEX; oserei dire che nessun altro software di tipocomposizione abbia la capacità di gestire correttamente una collezione così vasta di segni dalle

[5]Sembra che sia difficile da trovare perché viene ricreata on demand; tuttavia non la consiglio, perché è una traduzione della prima edizione del Bringhurst (2004); l'indicazione della bibliografia di questa guida si riferisce ad una edizione molto più recente

caratteristiche così raffinate; sul TeXbook (Knuth, 1984) sono illustrate le funzionalità dei font matematici e i quasi 30 parametri che li caratterizzano per la collocazione allineata dei vari simboli sia a livello dell'espressione principale, sia a livello delle piccole espressioni che possono trovarsi in posizione di esponente o di deponente, in entrambi i casi di primo o di secondo livello, sia quando si compone la matematica in linea con il testo, sia quando la si compone in display.

Solo lualatex e xelatex

Con xelatex e lualatex la varietà di font matematici è sufficiente, ma non grandissima; la collezione di segni più grande è quella dei font XITS Math (XITS, 2011), che si accoppiano bene con i font stretti come il Times. Per i font di default con xelatex e lualatex, che sono i font Latin Modern nella loro forma OpenType, i corrispondenti font matematici sono i font Latin Modern Math. Sono disponibili anche i font OpenType Asana Math che si accoppiano bene con i font con un grande occhio come i Palatino. Con il sistema TeX vengono distribuiti altri font OpenType matematici tra i quali i TeX Gyre Termes Math (TeX Gyre, 2012a) e TeX Gyre Pagella Math (TeX Gyre, 2012b) che si accoppiano benissimo rispettivamente con i TeX Gyre Termes e i TeX Gyre Pagella testuali (ovviamente); più recentemente sono stati resi disponibili i TeX Gyre Bonum Math (TeX Gyre, 2016a), i TeX Gyre DejaVu Math (solo sulle macchine Linux), e i TeX Gyre Schola Math (TeX Gyre, 2016b) che si accoppiano benissimo con i corrispondenti font testuali. Ma non ce ne sono molti altri.

Al momento di aggiornare questa guida tematica, la mia installazione di MacTeX su un portatile Mac dotato del sistema operativo OS X denominato "Big Sur", mi permette di usare i font matematici seguenti: *Asana Math*, *Cambria Math*, *Latin Modern Math*, *Libertinus Math*, *New Computer Modern Math*, *STIX Math*, *STIXIntegrals*, *TeX Gyre Bonum Math*, *TeX Gyre Pagella Math*, *TeX Gyre Schola Math*, *TeX Gyre Termes Math*, *XITS Math*. I font matematici il cui nome comincia con TeX Gyre sono tutti accoppiabili con i corrispondenti font testuali OpenType con lo stesso nome ma senza l'"aggettivo" Math.

Per quel che riguarda il font *Cambria Math*, va detto che esso risulta installato insieme alla versione almeno del 2003 della suite Microsoft Office. Non è chiaro dalla licenza se quel font sia usabile da parte dell'utente finale solo con con quella suite; tuttavia è presente sul disco fisso e i programmi xelatex e lualatex potrebbero farne uso.

I segni dei font XITS Math sono i più numerosi e sono fatti tutti molto

bene; si noti la particolarità: con i font OpenType non è più necessario fare riferimento ai gruppi matematici: infatti tutti i segni disponibili con ogni foggia possibile sono già contenuti in un unico font OpenType. L'importante è avere e usare il pacchetto `unicode-math` (Robertson *et al.*, 2015), già facente parte di ogni distribuzione completa del sistema TEX, che permette di accedere ai vari segni con semplici e comprensibili comandi in stile LATEX; l'elencazione di tutti questi comandi richiede circa 80 pagine di testo, e quindi ritengo che qui sia fuori luogo. Tuttavia ci sono alcuni testi introduttivi, come quello di Enrico Gregorio *introxelalatex.pdf* (Gregorio, 2011), che ritengo sia molto utile a tutti, come lo è stato per me. La gestione dei font matematici OpenType descritta in quel testo introduttivo vale anche se si usa lualatex.

Solo pdflatex

Per usare altri font matematici con pdflatex ci sono molti pacchetti; è generalmente raccomandato l'uso di `mathdesign` (Pichaureau, 2013), ma suggerirei di prendere in considerazione anche altri pacchetti completi per l'impostazione sia dei font testuali sia di quelli matematici; per esempio i pacchetti `fourier` (Bovani, 2005), `kpfonts` (Caignaert, 2010), (Voss, 2020) con il suo accessorio (Sharpe, 2018); quest'ultima collezione di font è molto interessante, perché è l'unica, di cui io sia al corrente, che usa font matematici di 256 glifi (da cui la sigla T1 nel nome del font matematico) e quindi consente di risparmiare molto sul numero di gruppi matematici, visto che pdfLaTeX può usarne solo1 6.

Molto interessanti sono i relativamente nuovi font TX e PX descritti nella documentazione raccolta nei documenti `newtxdoc.pdf`, (Sharpe, 2016b), e `newpxdoc.pdf`, (Sharpe, 2016a), con i loro due pacchetti `newtxtext` e `newtxmath`, e, rispettivamente, `newpxtext` e `newpxmath`; usando la matematica di questi pacchetti bisogna stare attenti a quali opzioni si usano o si impostano; c'è il pericolo di caricare fino a quattordici gruppi matematici e quindi di arrivare molto vicino al limite di 16 gruppi, oltre il quale è impossibile andare quando si usa pdflatex.

Queste sono le vie migliori per non impegolarsi con la delicatezza delle impostazioni dei font matematici; chi volesse comunque documentarsi a fondo potrebbe aprire con texdoc il file `fntguide.pdf` (LATEX 3 Project Team, 2005) che spiega tutto quello che c'è da sapere sulla gestione dei font con LATEX 2_ε. Basta aprire un terminale o prompt dei comandi (rispettivamente macchine UNIX oppure Windows) e dare il comando:

```
texdoc fntguide
```

Tutte le distribuzioni dispongono del comando da terminale texdoc e quasi tutte possono leggere la documentazione di qualunque pacchetto o di qualunque aspetto del sistema TeX perché questo comando espone i file già residenti sul proprio disco rigido; non è quasi mai necessario andare a reperire l'informazione nella "Rete".

2.7 Ricapitolazione

Alla fine della lettura di questo capitolo il lettore dovrebbe aver chiarito che cosa si intende per codifica di un font; che cosa significhi il nome "famiglia" quando si parli di font e sappia distinguerne il significato fra le famiglie dei font testuali e quelle dei font matematici che ho sistematicamente chiamato "gruppi matematici" proprio per marcare la distinzione; che cosa significhino le parole serie e forma riferite ai font; infine dovrebbe aver imparato bene che cosa sono i corpi ottici e aver capito la loro utilità.

Per completare, ora dovrebbe essergli chiaro che cosa sia necessario per accedere alla miriade di font vettoriali specialmente di tipo Type 1 e OpenType; con i primi dovrebbe essergli chiara la funzione dei file di descrizione dei font come intermediaria fra la disponibilità di font e le necessità dell'utente.

Dovrebbe forse essersi reso conto che davanti ad una scelta sterminata di font come quella disponibile con i programmi pdflatex, xelatex e lualatex, la scelta di una collezione di font, dove le famiglie, le forme e le serie si adattino l'una con l'altra in modo armonioso, è un'operazione molto delicata che non coinvolge solo gli aspetti tecnici ma specialmente gli aspetti estetici, che non possono venire trascurati quando si vuole tipocomporre testi di qualità.

L'interfaccia per l'utente 3

Abbiamo visto come i programmi di tipocomposizione del sistema TEX accedono ai font. Si tratta di una parte molto tecnica che si svolge dietro le quinte; l'utente non è obbligato a conoscerne tutti i dettagli, ma se li conosce è meglio, perché capisce anche che cosa succede davvero quando il suo programma preferito non fa quello che l'utente si aspetta.

"Ma come! Ho specificato di voler usare il maiuscoletto nero, ma mi usa il maiuscoletto medio". Già, di chi è la colpa? Del programma incapace di usare il font richiesto dall'utente, o dell'utente che pretende l'impossibile?

Come abbiamo visto l'errore è umano, come spesso succede; non è che i programmi siano privi di errori, ma fanno esattamente quello che il programmatore ha indicato loro di fare. Se non fanno quello che l'utente vuole è molto probabile che sia l'utente che non si è documentato abbastanza. Nei capitoli precedenti ho mostrato i meccanismi che soggiaciono al funzionamento dei programmi del sistema TEX per quel che riguarda i font, e negli esempi che ho riportato si vede chiaramente che i font mancanti vengono sostituiti da altri font simili. Il programma fa esattamente quello che gli è stato detto di fare: "sostituisci il maiuscoletto nero con il maiuscoletto medio". Niente di più, niente di meno.

Che cosa deve fare l'utente in queste circostanze? Deve leggere il file `.log` per andare a cercare tutti i warning di sostituzione dei font. Il file `.log` non è certo una lettura piacevole, ma è una fonte inesauribile di informazioni circa il decorso del processo di composizione da parte dei programmi. Lì c'è scritto tutto. Basta capire quello che c'è scritto; questa non è una operazione eccessivamente difficile, anche se i messaggi sono spesso scritti in inglese; si tratta quasi sempre di una o due righe di facile interpretazione.

Tuttavia ci sono molte altre cose che riguardano i font per le quali ogni

distribuzione del sistema TEX mette a disposizione gli strumenti adatti per evitare di dover fare a mano operazioni delicate e difficili.

3.1 Selezione di font alternativi

Una cosa che gli utenti vorrebbero fare più spesso di quanto non facciano sarebbe di selezionare dei font diversi da quelli preimpostati. Come ho già detto i font usabili sono diverse migliaia quando si dispone di una distribuzione completa del sistema TEX. Metto l'accento su questa completezza, perché gli utenti, specialmente quelli che lavorano su macchine Windows, spesso installano la versione base di MiKTeX; siccome impostano il loro sistema in modo che carichi al volo i pacchetti mancanti quando vengono richiesti, pensano di risparmiare tempo (e spazio su disco) installando solo quello che serve loro e niente di più. Già, ma se al momento del bisogno di un certo pacchetto non sono connessi alla rete? Già, ma come fanno a sapere quali font sono disponibili se non li hanno a bordo del loro calcolatore? Con una versione completa di TEXLive o di MiKTeX questi problemi non si pongono, anche se è molto probabile che la versione completa contenga anche cose che l'utente non userà mai.

Dunque: si installi sempre e soltanto il sistema TEX completo. Con i dischi rigidi di oggi, un gigabyte in più o in meno di occupazione del disco non costituisce una differenza sensibile.

Poi bisogna sapere dove sono i font disponibili. Con la distribuzione TEXLive il sistema di alberi e di cartelle è conforme alle specifiche TDS (TEX Directory System). Queste specifiche sono descritte nel manuale di TEXLive (Berry, 2016). A quanto pare, i sistemi di alberi e cartelle della distribuzione MiKTeX non sono conformi alle specifiche TDS; non molto difformi, comunque non conformi. Mi riferirò perciò alla struttura TDS usando come separatore delle cartelle il segno `/` tipico dei sistemi UNIX; per Windows si sostituisca quel segno con \.

Ogni albero del sistema TEX *a partire dalla sua radice, non dalla radice del disco,* contiene un insieme di rami uno dei quali si chiama `/fonts`; questo a sua volta è suddiviso in rami parziali che hanno nomi abbastanza chiari; per esempio il ramo `/fonts/type1/public/lm/` contiene i file di descrizione geometrica dei font vettoriali Type 1 Latin Modern di cui ho parlato nei capitoli precedenti; il ramo `/fonts/tfm/public/lm/` contiene i file metrici per gli stessi font Latin Modern. Ripartendo dalla radice

dell'albero, il ramo `/tex/latex/lm/` contiene i pacchetti di estensione per usare i font Latin Modern nelle varie codifiche e i file `.fd` di descrizione di questi font.

Se si esaminano i "rametti" innestati nel ramo `/doc/fonts/` si scopre l'esistenza di una documentazione sterminata, spesso troppo essenziale ma, altrettanto spesso, illuminante.

La maggior parte di questi rami si trova in `$BASE/texlive/`⟨*anno*⟩`/`, dove `$BASE` ha un valore diverso a seconda del sistema operativo usato: potrà essere `/usr/local` per i sistemi UNIX, oppure `C:\Programmi` per i sistemi Windows. Fra gli alberi ci può essere anche l'albero personale che comincia con `~/texmf` nei sistemi Linux, con `~/Library/texmf` nei sistemi Mac OS X, oppure `C:\Users\`⟨*username*⟩`\texmf` in alcuni sistemi Windows (maggiori dettagli si possono trovare in Pantieri e Gordini (2017)).

La cosa migliore da fare per scegliere font alternativi è quella di usare un pacchetto che faccia tutto quel che serve, senza che l'utente debba scervellarsi con comandi di basso livello. Per esempio se si vogliono usare i font Latin Modern, basta caricare il pacchetto `lmodern` e il gioco è fatto.

Lo stesso si può dire per usare i font Utopia: basta caricare il pacchetto `fourier`, che fornisce tutto l'occorrente per le varie famiglie, serie e forme sia per il testo sia per la matematica. Lo stesso si può fare per usare i font KP, una elaborazione dei font Palladio: basta usare il pacchetto `kpfonts`; entrambi questi pacchetti offrono soluzioni molto belle e complete sia per il testo sia per la matematica.

I font Palatino completi per la matematica si potevano (e si possono ancora) usare invocando il pacchetto `pxfonts`; siccome questo pacchetto è un "fratello" di `txfonts`, utile per usare i font Times, ne condivide pregi e difetti; entrambi sono criticati per avere alcuni piccoli difetti di accostamento quando si compone la matematica. Per ovviare a questi inconvenienti è disponile la collezione dei "Nuovi font Times estesi" (`newtx`) da usare tramite due pacchetti distinti, uno per i font testuali, `newtxtext`, e un altro per i font matematici, `newtxmath`; entrambi questi pacchetti accettano numerose opzioni, per cui la lettura della documentazione è ancora più necessaria che non in altri casi. Lo stesso si può dire per i "Nuovi font Palatino estesi" con i pacchetti `newpxtext` e `newpxmath`; lo si può ripetere anche per i font `Libertinus` con i suoi compagni `LibertinusT1math`

Il pacchetto `mathdesign` permette di abbinare certi font testuali, scelti mediante opzioni nella chiamata del pacchetto, con corrispondenti font matematici in modo da fornire un insieme di font completo; i font testuali a cui sono abbinati i font matematici sono Adobe Utopia, URW Garamond e Bitstream Charter. I Garamond testuali a stretto rigore non sono font liberi e quindi vanno acquistati o comunque acquisiti separatamente dal sistema TEXe, anche se sono gratuiti, esistono delle limitazioni circa il loro uso; recentemente però sono stati aggiunti alle distribuzioni del sistema TEX i font EBgaramond, per ora soltanto nella serie media; ma il progetto sta avanzando per cui non dovrebbe essere lontano il giorno in cui questi font saranno corredati della serie nera.

Il pacchetto `mathpazo` permette di usare per il testo il font Adobe Palatino; il pacchetto `mathptmx` permette di usare invece il font Times; entrambi i pacchetti contengono tutti i gruppi matematici necessari e con alcune opzioni possono produrre glifi non presenti normalmente nelle collezioni standard CM, CM-super e LM. Entrambi i pacchetti non si occupano di caricare i font senza grazie e i font teletype, per cui questi rimangono quelli di default del sistema TEX; questa, anzi, è una delle differenze rispetto ai corrispondenti pacchetti `txfonts`, `pxfonts` e `newtxtext`, `newpxtext`.

I pacchetti indicati sono solo un piccolo insieme dei pacchetti che si possono usare; nella cartella `tex/latex/psnfss/` sono disponibili pacchetti per caricare i font PostScript più comuni: Avantgarde, Bookman, Chancery, Charter, Courier, Helvetica, New Century School Book, Utopia. Nella tabella 2.1 della pagina 52 essi sono mostrati insieme ad alcune forme della collezione Latin Modern. Si è già fatto notare la differenza della grandezza delle lettere minuscole, mentre la lunghezza dell'alfabeto, mostrata indirettamente dalla stessa frase, è del tutto evidente, senza bisogno di sottolinearlo.

Con i programmi xelatex e lualatex la scelta dei font testuali è relativamente più facile; nella cartella `/fonts/opentype/` compaiono le cartelle contenenti tutti i file dei font OpenType distribuiti con il sistema TEX; c'è solo l'imbarazzo della scelta, tenuto conto che ogni font viene caricato semplicemente con i comandi resi disponibili con il pacchetto `fontspec`; bisogna solo ricordarsi di specificare le opzioni che si desiderano attivare fra le molteplici scelte che i font OpenType offrono. Inoltre sono accessibili tutti i font distribuiti insieme al sistema operativo; talvolta, licenza permettendo, anche i font che accompagnano certe suite di programmi di produttività personale.

Come ho già detto, con la matematica il problema è un po' più difficile, perché la scelta non è così abbondante; tuttavia esiste ormai una certa disponibilità di font OpenType matematici e l'unico imbarazzo può essere quello di scegliere una accoppiata valida sia per nerezza sia per l'occhio del font matematico con le corrispondenti caratteristiche del font testuale.

3.2 I comandi di basso livello

La guida ufficiale per maneggiare i font con pdflatex è formata dal testo LaTeX 3 Project Team (2005); questo descrive sia i comandi di basso livello per i font testuali sia per quelli della matematica.

Con l'accesso ai font eseguito con il NFSS (New Font Selection Scheme) ogni font testuale è caratterizzato da sei informazioni: la *codifica*, la *famiglia*, la *serie*, la *forma*, il *corpo*, e l'*scartamento* (chiamato anche *avanzamento di riga* e, erroneamente, *interlinea*); esistono comandi di basso livello per specificare ciascuna di queste caratteristiche:

```
\fontencoding{⟨codifica⟩}
\fontfamily{⟨famiglia⟩}
\fontseries{⟨serie⟩}
\fontshape{⟨forma⟩}
\fontsize{⟨corpo⟩}{⟨scartamento⟩}
\selectfont
\usefont{⟨codifica⟩}{⟨famiglia⟩}{⟨serie⟩}{⟨forma⟩}
```

I comandi di alto livello sono quelli che si usano normalmente per il mark up di LaTeX: le dichiarazioni `\rmfamily`, `\sffamily`, `\ttfamily` permettono di scegliere le famiglie: `\mdseries` e `\bxseries` permettono di scegliere la serie media o la serie nera estesa; `\upshape`, `\slshape`, `\itshape`, `\scshape` permettono di scegliere le forme diritta, inclinata, corsiva e maiuscoletta; le dichiarazioni da `\tiny` fino a `\Huge` permettono di scegliere i corpi e i rispettivi scartamenti come predefiniti dall'opzione della classe quando viene specificata l'opzione per il corpo normale.

Ricordo che le precedenti dichiarazioni e i corrispondenti comandi cambiano una caratteristica alla volta senza cambiare le altre cinque; in altre parole se il font corrente ha la codifica ⟨*T1*⟩, la famiglia ⟨*lmr*⟩, la serie ⟨*m*⟩, la forma ⟨*n*⟩, il corpo ⟨*10.95*⟩ e lo scartamento ⟨*13.2*⟩, i seguenti comandi

`{\fontfamily{phv}\fontshape{it}\selectfont...}`,
`{\fontfamily{phv}\itshape...}`,
`{\fontfamily{phv}\textit{...}}`,
eccetera,

producono lo stesso effetto sul testo che sostituisce i puntini, che verrà composto con gli stessi parametri di codifica, serie, corpo e scartamento, che non sono stati modificati, ma con la famiglia ⟨*phv*⟩ e la forma ⟨*it*⟩ che sono le sole caratteristiche modificate dai comandi o dalle dichiarazioni usate negli esempi.

L'unica caratteristica che non si può cambiare con un comando di alto livello è la codifica; ci sono molte ragioni per cui non esiste un comando di alto livello: una è che quando si compone e si è specificata l'opzione della codifica dei font in uso al pacchetto `fontenc` è raro, per non dire impossibile, che l'utente cambi codifica, a meno che non cambi alfabeto per comporre del testo in una lingua diversa da quella principale; se ciò succedesse, generalmente è il pacchetto invocato o l'opzione a `babel` che provvede a cambiare codifica dietro le quinte, senza che l'utente debba occuparsene. Un'altra ragione è che le codifiche sono così numerose che sarebbe difficile prevederle tutte.

Tuttavia, come mostrato nella sintassi di questi comandi esiste il comando `\usefont` col quale si può specificare tutto, compresa la codifica, ma non il corpo o lo scartamento; personalmente ne faccio un certo uso quando scrivo file di classe o moduli di estensione, ma è probabile che un utente "normale" non ne faccia mai uso; conviene però sapere che esiste anche questa possibilità.

I primi comandi di basso livello non fanno altro che impostare le stringhe adatte per la ricerca dei font negli appositi file `.fd` ma il loro uso non cambia nulla finché non si specifica il comando `\selectfont` in modo esplicito o implicito (è implicitamente eseguito quando si esegue un comando di alto livello); se si devono cambiare diverse specifiche contemporaneamente si può usare il comando `\usefont` con tutti i suoi argomenti senza bisogno di specificare `\selectfont` perché ci pensa lui stesso ad eseguirlo. Ho già fatto un esempio d'uso dei comandi di basso livello nella pagina 51.

Che si usino i comandi di alto o basso livello, che cosa succede dietro le quinte? Lo spiego con un esempio: in questo momento sto componendo con i font Latin Modern (tramite il pacchetto `cfr-lm` che definisce delle famiglie

di font virtuali diverse dalle famiglie che vengono usate quando si invochi il pacchetto `lmodern`) con la codifica *T1*, la famiglia *clm2j*, la serie media *m*, la forma normale *n* e il corpo 11 (in realtà 10.95). I comandi di alto o di basso livello hanno caricato il font giusto nel corpo giusto concludendo con la definizione e l'esecuzione della macro `\T1/clm2j/m/n/10.95`; posso farmi scrivere qui da pdflatex stesso il significato della macro, ed è questo: "`select font clmr2j8t10 at 10.95pt`". Vi compaiono dei comandi primitivi (tutti i comandi primitivi del NFSS sono privi di backslash) che dicono "seleziona ed usa il font (virtuale) clm2j8t10 ingrandendolo a 10,95 punti tipografici". Queste sono sostanzialmente le informazioni che erano scritte ne file `.fd`; ma da dove è saltato fuori quel nome di macro? Per giunta un nome di macro che non contiene solo lettere, come dice qualunque guida su LaTeX. Be', questa è una domanda impropria; è chiaro che in ogni guida c'è la specificazione pressante di usare solo lettere nei nomi delle macro, oppure un solo simbolo non letterale, ma questa specificazione vale solo per gli utenti finali, non per il programma pdflatex che deve poter eseguire delle operazioni complicate come, per esempio, formare dei nomi di macro a partire da generiche stringhe di caratteri; non entro nei dettagli, ma pdflatex può usare anche nomi di macro che contengono cifre e segni analfabetici.

Perciò quando ha cominciato a scrivere con questo font, il programma ha esaminato la codifica (*T1*), la famiglia (*clm2j*), la serie (*m*), la forma (*n*) e il corpo (*10.95*) ed è andato a vedere fra le definizioni di `\DeclareFontShape` nel file `t1clm2j.ld` quella che conteneva le stringhe 'm' ed 'n" fra i suoi parametri; trovatala, è andato a vedere che cosa deve fare per il corpo 10.95 e vede che deve prendere il debito font (virtuale, in questo caso: *clm2j8t10*) di corpo 10, ma lo deve ingrandire fino a portarlo alla dimensione richiesta; ecco dunque che queste ultime operazioni le ha messe nel comando interno di selezione del font, cosicché, quando deve usarlo sa esattamente che cosa fare, perché ha già stivato le informazioni metriche e il fattore di ingrandimento in memoria, e non deve più ripetere quella ricerca sul file di descrizione dei font.

Per i gruppi matematici i comandi di basso livello sono decisamente più delicati da usare; sarebbe meglio che l'utente non si occupasse di queste cose e che lasciasse fare ai pacchetti già disponibili. Se però ha bisogno di usare come matematico un font particolare, deve definire un alfabeto matematico nei modi e con i comandi che trova esposti nella guida LaTeX 3 Project Team (2005); deve stare molto attento a seguire con estrema

attenzione quanto vi è scritto; non deve confondere `\DeclareMathAlphabet` con `\SetMathAlphabet`, capendone la differenza nei minimi dettagli. In fondo i font matematici formano gruppi un po' più complessi da usare dei font testuali, quindi è logico che la loro gestione sia più complessa; ma con un po' di pratica si scopre che la cosa non è poi così difficile.

3.3 Installazione di nuovi font

3.3.1 I font OpenType da usare con lualatex e xelatex

Solo lualatex e xelatex

Per installare nuovi font OpenType da usare con xelatex e lualatex basta metterne i file `.otf` nella cartella del proprio calcolatore dove sono raccolti i font di quel tipo; meglio ancora: per tenerli ben separati da quelli del sistema operativo, secondo me, conviene mantenerli in una cartella personale (allo stesso modo come i font OpenType distribuiti con il sistema TEX si trovano in una cartella di uno degli alberi del sistema TEX non in una cartella del sistema operativo) e mettere nella cartella dei font del sistema operativo o dell'utente solo un *symbolic link* che punti alla cartella personale.

Un symbolic link è un oggetto che a tutti gli effetti si comporta come un file o una cartella di file, ma rappresenta solo un puntatore al file o alla cartella di file. Nei sistemi UNIX il concetto è consolidato "da sempre"; nei sistemi Windows esso è presente solo a partire dalla versione Windows Vista.

Se i font OpenType personali sono in una cartella `~/mieifont`, per esempio, in un sistema UNIX la creazione del symbolic link si esegue da terminale, dopo essersi posti nella cartella di destinazione, dando il comando:

```
ln -s ~/mieifont MieiFontOT
```

cosicché nella cartella di destinazione, visualizzata attraverso l'interfaccia grafica, appare una nuova voce `MieiFontOT` affiancata da una icona di cartella; cliccandoci sopra si vede in realtà la lista dei file contenuti nella cartella `~/mieifont`, ma che appaiono come se fossero in una cartella reale dentro la cartella di destinazione.

In una macchina Windows (Vista, 7, 8, 10) i font contenuti nella cartella `C:\Users\`⟨*user name*⟩`\mieifont` possono apparire nella cartella dei font

del sistema operativo, aprendo il prompt dei comandi, (Giacomelli, 2013), andando nella cartella dei font del sistema operativo e dando il comando:

`mklink MieiFontOT C:\Users\`⟨*user name*⟩`\mieifont`

quello che succede è la stessa cosa che avviene nelle macchine UNIX.

Talvolta, a seconda del sistema operativo, è necessario eseguire quelle operazioni con i privilegi dell'amministratore e può essere necessario aggiornare la *cache* dei font. Queste sono cose di amministrazione "ordinaria" del proprio calcolatore, anche se non vengono eseguite molto spesso; tuttavia la documentazione del sistema operativo dice come eseguirle.

Per l'uso di questi font con xelatex e lualatex non c'è altro da fare; basta usarli, invocandoli attraverso i comandi specifici del pacchetto `fontspec`.

3.3.2 I font vettoriali da usare con pdflatex

Solo pdflatex

Per usare nuovi font Type 1, TrueType o OpenType con pdflatex le cose sono molto più complicate; come già detto in precedenza, bisogna creare i file metrici `.tfm`, bisogna creare, se necessario, i font virtuali, bisogna creare i file di mappa specifici per i nuovi font, bisogna spostare tutti i file in cartelle opportune, bisogna configurare il sistema TEX in modo che possa usare questi font.

Esistono diversi modi per fare queste cose ma, sebbene io non abbia ancora avuto modo di provarlo, esiste un recente programma autoinst del pacchetto `fonttools` (Penninga, 2013a,b). Dando da terminale il comando `texdoc fontools` viene aperto un file `README` testuale, molto scarno; ma dando da terminale il comando `texdoc autoinst` si apre una bella documentazione che spiega bene la funzione e l'uso del programma autoinst.

In sostanza questo pacchetto può essere usato per installare in modo quasi automatico i font OpenType sul proprio calcolatore generando tutti i file necessari, compresi i file di estensione `.sty` e di descrizione dei font `.fd` necessari per l'uso con pdflatex; se il lavoro viene svolto su una macchina che abbia una distribuzione TEXLive con i suoi alberi conformi alle specifiche TDS, provvede anche a spostare i vari file nelle opportune cartelle, generandone di nuove se fosse necessario.

Per i font TrueType il programma ttf2afm estrae le informazioni metriche scritte dentro i rispettivi file `.ttf` e le trascrive in file `.afm`; in modo

che possano essere trattati alla stessa maniera dei font Type 1. Alternativamente si può usare il programma ttf2tfm che produce direttamente anche i file virtuali necessari; vedi meglio qui di seguito.

I font Type 1 sono dotati di un file `.afm` con le informazioni metriche dei vari glifi nel formato richiesto dalla Adobe e, generalmente, di un file `.pfb` che contiene la descrizione PostScript dei glifi, con un codice compresso in forma binaria (`pfb` è l'acronimo che si ottiene da Postscript Font in Binary form); talvolta sono dotati invece di un file `.pfa` (`pfa` è l'acronimo che si ottiene da Postscript Font in Ascii form). Il file binario può essere trasformato in file ASCII mediante il programmino pfb2pfa, ma di solito non è necessario farlo.

Poi il programma fontinst (Jeffrey *et al.*, 2009) permette di fare in modo automatico quasi tutto quello che serve per installare i font Type 1 che si vogliono usare con pdflatex. La documentazione fa riferimento anche all'uso del programmino ttf2afm citato sopra per i font TrueType, ma all'interno della documentazione non ne parla più. Sembra che sia utile usare il programmino ttf2tfm (Lemberg e Loyer, 2002) che con un po' di manualità in più provvede anche alla creazione dei font virtuali.

Il disegno che compare nel frontespizio della documentazione di `fontinst`, riportato nella figura 3.1 della prossima pagina, lascerebbe pensare che il programma sia finalizzato all'uso con i file DVI da trasformare in formato PostScript mediante dvips. Non è così: una volta che siano disponibili i file `.tfm`, i file virtuali `.vf`, i file delle mappe dei font `.map` e si siano aggiornate le mappe generali, i font sono usabili anche da pdflatex. In sostanza il programma (che è è scritto in linguaggio TEX!) manipola i file specificati in entrata e genera i file virtuali, ricodificando i font da usare, genera il file o i file `.tfm` che corrispondono all'unico font o ai diversi font virtuali estratti dal file `.afm` e collegati al file `.pfb`, genera le mappe, i file di descrizione dei font `.fd` e quant'altro sia necessario per usare quei font con pdflatex; l'unica cosa che resta da fare è aggiornare le mappe generali, di cui parlerò più avanti nel paragrafo 3.3.4 della pagina 70.

3.3.3 Un caso più semplice per usare un font Type 1 con pdflatex

Un caso più semplice, che a me è capitato sovente, è quello che si ha quando si crea un font con METAFONT, oppure quando di un font si dispone solo

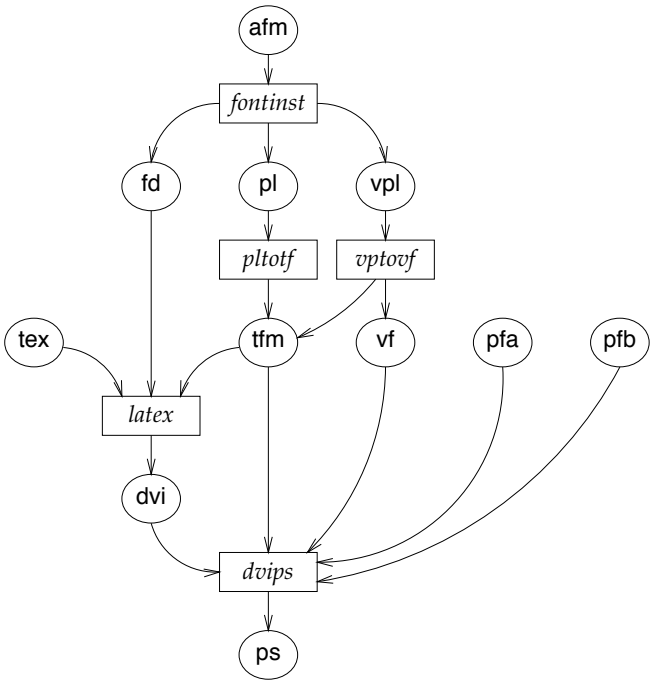


Figura 3.1 Schema di flusso dell'uso di fontinst

dei file sorgente `.mf` per il programma METAFONT. Creare un font partendo da un file unico o principale con estensione `.mf` vuol dire produrre: il file `.tfm` con le informazioni metriche e il file `.600pk` per il font rasterizzato per una stampante, per esempio, da 600 dpi. Quest'ultimo file di solito non serve e, a meno che per qualche remoto motivo non si voglia creare un file DVI, lo si può cestinare. Il file `.tfm`, può essere spostato in una "apposita" cartella nell'albero locale o personale `fonts/tfm/`. Il file `.mf` era presente in una "apposita" cartella dell'albero principale o locale o personale `fonts/source/`; in tutti questi casi è opportuno che l' "apposita" cartella abbia un nome che faccia riferimento allo specifico font.

A questo punto si lancia dal terminale il programma mftrace, (Nienhuys, 2010); se il font si chiama *miofont*, il comando sarà del tipo:

```
mftrace -f PFB –simplify –potrace miofont
```

Può darsi che il programma si lamenti della mancanza della specificazione del file di codifica; se si vuole, lo si può specificare con un'ulteriore opzione, ma non sempre è utile: dipende da quali glifi sono contenuti nel file; io per esempio ho creato dei font con "pittogrammi" (come quello nel margine) che nulla hanno a che vedere con un alfabeto o con glifi analfabetici; questo particolare pittogramma mi serviva per marcare in un mio testo di elettronica gli esercizi che andavano svolti mediante il calcolatore e l'uso del programma PSpice; ecco il significato dello schermo del PC e delle lettere PS al suo interno. Avevo scelto un font vettoriale con un pittogramma, piuttosto che un disegno in formato `.png`, perché il font occupava molto meno spazio sia sul disco (erano tempi in cui dischi rigidi da 100 MiB erano un lusso sfrenato...) sia nel file PDF finale.

PS

Il programma mftrace genera un nuovo font rasterizzato molto grande, traccia i contorni di ogni glifo usando il programma potrace, crea un file in formato vettoriale adatto da dare in pasto al programma FontForge, che a sua volta semplifica i contorni dei glifi così tracciati e genera il file in formato binario `.pfb` come specificato nelle opzioni.

Ovviamente per fare queste operazioni bisogna aver scaricato dalla rete e avere installato i programmi mftrace, potrace, e FontForge (il quale per funzionare completamente ha bisogno di avere a disposizione anche il programma autotrace, ma non è indispensabile per questa particolare operazione).

Alla fine i file ausiliari usati da mftrace e da FontForge vengono cancellati, e nella cartella di lavoro rimane il file finale `miofont.pfb`; questo va spostato nell' "apposita" cartella dell'albero locale o personale sotto a `fonts/type1/`.

Non resta che creare i file ausiliari per usare il nuovo font con pdflatex:

1. il file `.fd` avrà come nome la codifica (U se non si è specificata nessuna codifica a mftrace) agglutinata al nome della famiglia, che nel caso di un unico font potrebbe coincidere con il nome del font. Supponendo di chiamare la famiglia con lo stesso nome del font e di prevedere alcune sostituzioni per serie e forme mancanti, il file sarà dunque:

```
% File umiofont.fd
\ProvidesFile{umiofont.fd}
         [2013/01/05 v.1.0 File di descrizione di miofont]
\DeclareFontFamily{U}{miofont}{}
\DeclareFontShape{U}{miofont}{m}{n}{<-> miofont}{}
```

```
\DeclareFontShape{U}{miofont}{b}{n}{ssub* miofont/m/n}{}
\DeclareFontShape{U}{miofont}{bx}{n}{ssub* miofont/m/n}{}
\DeclareFontShape{U}{miofont}{m}{it}{ssub* miofont/m/n}{}
\DeclareFontShape{U}{miofont}{b}{it}{ssub* miofont/m/n}{}
\DeclareFontShape{U}{miofont}{bx}{it}{ssub* miofont/m/n}{}
...
\endinput
```

2. Il file `.sty` non è strettamente necessario se uno si adatta a usare i comandi di basso livello, ma è utile per risparmiare tempo ed errori:

```
% file miofont.sty
\ProvidesPackage{miofile}
        [2013/01/05 v.1.0 Pacchetto per l'uso di miofont]
\newcommand*{\miofont}
        {\fontencoding{U}\fontfamily{miofont}\selectfont}
\DeclareTextFontCommand{\textmio}{\miofont}
\endinput
```

3. Il file `miofont.map` con le informazioni necessarie per l'uso con pdflatex:

```
miofont MIOFONT <miofont.pfb
```

   Le informazioni un po' criptiche da mettere in una sola riga per ciascun font da "mappare" sono descritte estesamente nella documentazione del programma pdftex, (Thành *et al.*, 2016, §5.1). La prima istanza di `miofont` è il nome, senza estensione, del file metrico che viene letto da pdflatex, la seconda istanza è il nome PostScript del file `miofont.pfb` che contiene le informazioni sul disegno dei glifi; per conoscere con precisione questo nome basta usare FontForge, aprire il file `miofont.pfb`, selezionare la voce del menù Element ⟶ Font Info e, nella finestra che si apre scegliendo questa voce, leggere il valore di "Base Filename" che, però, spesso viene segnalato come identico a "Fontname".
4. I file `.fd` e `.sty` vanno poi messi o spostati nella solita "apposita" cartella nell'albero locale o personale sotto `tex/latex/`; se si mettono nell'albero locale, è necessario aggiornare il database dei nomi dei file con i metodi propri di ogni distribuzione del sistema TEX.
5. Il file `.map` è bene metterlo nella solita "apposita" cartella sotto `fonts/map/` dell'albero locale o personale.

Non resta che creare le mappe generali dei font.

3.3.4 Le mappe generali dei font

Le mappe generali dei font sono `psfonts.map`, `pdftex.map`, `dvipdfmx.map` e `ps2pk.map` che verranno usati rispettivamente dai programmi dvips, pdflatex, dvipdfmx e ps2pk. L'ultimo di questi programmi serve per trasformare in formato raster i font PostScript; il che sembra una pessima idea, ma tant'è: il programma esiste, anche se io in più trent'anni non l'ho mai usato né visto usare. Sono anni che non uso nemmeno dvips, perché riesco ad ottenere tutto quello di cui ho bisogno soltanto usando i programmi di composizione pdflatex, xelatex o lualatex. Il programma dvips, però, non è da passare sotto gamba perché, nonostante i tempi siano cambiati, può ancora essere necessario avere file in formato PostScript, in quanto ci sono ancora alcune case editrici che desiderano avere quel formato. Inoltre quando si deve usare il pacchetto `PSTriks` il passaggio attraverso il formato PS sembra l'unico possibile (in realtà si può aggirare il problema, ma non è questo il luogo per discuterne).

Per creare i suddetti file generali di mappa bisogna ricorrere al programma updmap o updmap-sys di TEXLive oppure attraverso la funzione `update` di initexmf di MiKTeX.

1. Con MiKTeX bisogna aprire il prompt dei comandi e dare i due comandi seguenti:

   ```
   initexmf -edit-config-file updmap
   initexmf -makemaps
   ```

   Il primo comando permette di editare il file `updmap.cfg` per aggiungerci un'unica riga:

   `Map miofont.map`

   Dopo aver salvato il file, al resto provvede i comando successivo.
   Con certi sistemi operativi Windows i due programmi possono essere attivati solo con le prerogative dell'amministratore della macchina.
   Eseguito due volte il programma initexmf con le opzioni e gli argomenti indicati, si può andare a cercare il file `pdftex.map` e controllare che vi sia elencato anche il nome del font *miofont*.
2. Con TEXLive su una macchina con qualunque sistema operativo le cose sembrano più complesse; in realtà sono le stesse, ma vanno eseguite scrivendo direttamente il nome del file di mappa dentro un

file `updmap.cfg` residente nell'albero locale o in quello personale, nella cartella `web2c/`; precisamente bisogna scriverci dentro solamente:

```
Map miofont.map
```

Bisogna ricordare che se il file è nuovo e si trova nell'albero locale, è necessario aggiornare il database dei nomi dei file con il comando mktexlsr con i privilegi dell'amministratore.
Poi, a seconda che si voglia rendere disponibile il font a tutti gli utenti della macchina oppure solo a se stessi, si userà il comando updmap-sys oppure updmap. Di solito se il file `updmap.cfg` si trova nell'albero personale è perché si vuole mantenere l'uso del font strettamente personale; tuttavia se anche il file `.cfg` si trova nell'albero personale, è meglio, secondo me, eseguire l'installazione per tutti gli utenti, anche se questi utenti ammontassero al solo utente che sta eseguendo questa installazione. Perciò, con i privilegi dell'amministratore o del *super user*[1] si darà nel terminale il comando:

```
sudo updmap-sys
```

Dopo qualche istante appariranno nel terminale molti messaggi, ma, salvo errori, alla fine dell'esecuzione le mappe generali saranno state costruite o rigenerate e `pdftex.map` conterrà anche *miofont*, come desiderato e previsto.

A questo punto il programma pdflatex è in grado di usare il font *miofont*. Va segnalato che prima della distribuzione 2012 di TEXLive, era necessario dare il comando

```
tlmgr generate updmap
```

subito prima di eseguire updmap-sys. Ma dal 2012 la procedura è cambiata e il programma updmap-sys esamina in sequenza tutti i file `updmap.cfg` che si trovano nei vari alberi, compreso quello personale, se esiste; "tutti i file che trova" vuol dire tutti i file che il comando `kpsewhich -all updmap.cfg` restituisce come risposta; nella fattispecie della mia installazione questo

[1] L'utente normale sulle macchine UNIX acquista i privilegi dell'amministratore premettendo al comando da eseguire la parola chiave `sudo` (super user esegui); come si lancia il comando, viene chiesta la parola chiave dell'amministratore e, se la parola chiave è corretta, il programma viene eseguito con quei privilegi.

comando (oggi che sto usando la distribuzione di TEXLive del 2017) mi restituisce:

```
/Users/claudio/Library/texmf/web2c/updmap.cfg
/usr/local/texlive/texmf-local/web2c/updmap.cfg
/usr/local/texlive/2017/texmf-dist/web2c/updmap.cfg
```

Infine esegue la generazione dei file generali delle mappe concatenando i contenuti dei vari file `updmap.cfg` che trova. Certo potrebbero esserci dei doppioni e lo stesso file di mappa o lo stesso font potrebbe essere elencato in diversi posti; il programma, però, mette nei file generali di mappa solo la prima istanza che ha trovato; scrive nel file `updmap.log` quanto è accaduto, ma non crea voci multiple nei file di mappa generali.

3.4 Font per scrivere con alfabeti diversi

Comporre in lingue che usano alfabeti diversi può essere una cosa assai delicata. Bisognerebbe che quando si cambia alfabeto e lingua, il comando che esegue tale operazione non solo selezioni le regole di composizione di quella lingua, ma modifichi la selezione dei font in modo da conservare almeno serie, forma e corpo con il corrispondente scartamento, ma possa cambiare sia la codifica dei font, sia la famiglia da cui trarre in glifi.

3.4.1 Alfabeti diversi con lualatex e xelatex

Solo lualatex e xelatex

Con xelatex e lualatex la cosa è particolarmente semplice, grazie all'uso dei font OpenType. Infatti a stretto rigore, font permettendo, non sarebbe nemmeno necessario cambiare famiglia e codifica. Per esempio, se si usassero i font Libertine, che contengono sia l'alfabeto latino sia quello greco completi anche dei segni più rari (non tutti però), basta mettersi a scrivere in greco nel file sorgente (tastiera permettendo) e la composizione con entrambi gli alfabeti avviene senza problemi.

Se invece si usassero i font Latin Modern, che contengono solo l'alfabeto latino, allora bisognerebbe scegliere un altro font per poter comporre le parti in greco; supponiamo di usare a questo scopo la sezione greca dei font Libertine. Basta allora definire una nuova famiglia di font, oltre alle famiglie definite con `\setmainfont`, `\setsansfont` e `\setmonofont`:

```
\newfonfamily{greekfont}{⟨nome del font⟩}[⟨opzioni⟩]
```

dove le ⟨*opzioni*⟩ sono quelle che si desiderano per usare quel font in quella lingua e con il font chiamato con ⟨*nome del font*⟩; con i font *ibertine*[2] il comando diventa

```
\newfontfamily{\greekfont}{Linux Libertine O}
```

È invece importante che il nome della famiglia cominci con il nome della lingua, quello che è stato specificato per la selezione della lingua con i comandi di `polyglossia`; quindi selezionando il font Libertine, per esempio, il nome della famiglia deve chiamarsi `\greekfont`.

In questo modo quando si compone in greco, i comandi che selezionano la lingua in uso, selezionano anche il font che è stato specificato per quella lingua. Le specificità dei font OpenType garantiscono che le altre caratteristiche rimangano invariate.

3.4.2 Alfabeti diversi con pdflatex

Solo pdflatex

Con pdlatex le cose sono più semplici ma più delicate.

I comandi di cambiamento di lingua, attualmente, cambiano alfabeto cambiando solo la codifica; per il greco la codifica predefinita è LGR; di conseguenza la famiglia, la serie, la forma, il corpo e lo scartamento non vengono cambiati. Cose simili avvengono con il cirillico e le opzioni del pacchetto `babel`. Per altre lingue meno comuni per gli utenti italiani, come l'ebraico, l'arabo o le lingue orientali, gli appositi pacchetti o file di descrizione delle lingue permettono di accedere ai font corretti; l'utente può consultare la documentazione di questi pacchetti e di queste lingue. Data la facilità con cui si gestiscono gli alfabeti diversi da quello latino con i font OpenType, il mio suggerimento sarebbe quello di usare direttamente xelatex o lualatex.

[2] Per conoscere il ⟨*nome del font*⟩ da inserire nell'apposito argomento con il Mac basta aprire l'applicazione FontBook.app scegliere e cliccare nella lista dei font il nome Linux Libertine,; nella finestra che si apre si clicca sul dischetto bianco contenente una *i* nella lista di icone in alto a sinistra, e nella metà di destra della schermata appaiono tutte le informazioni relative al font; il ⟨*nome del font*⟩ è quello identificato con la voce Family. Con altri sistemi operativi le informazioni relative ad un dato font si trovano con l'applicativo otfinfo o, sulle macchine Windows, otfinfo.exe.

Con pdflatex i font di default sono i font della collezione CBfonts, creati apposta per la codifica *LGR*; i nomi delle loro famiglie sono identici ai nomi delle famiglie dei font latini predefiniti. Quindi con la collezione LM, le famiglie standard sono `lmr` per i font proporzionali con grazie, `lmss` per i font proporzionali senza grazie, e `lmtt` per i font teletype. Passare da un alfabeto all'altro vuol dire passare dalla codifica *T1* alla codifica *LGR* o viceversa, senza quasi fare altro, se non per cambiare le impostazioni della lingua, cioè la cesura, le parole predefinite, la composizione della data e poche altre cose.

Per questo motivo, per esempio, le famiglie di font proporzionali con grazie hanno i file di descrizione dei font latini che si chiamano `t1lmr.fd` e `lgrlmr.fd` rispettivamente per i font latini e per i font greci. Le istruzioni interne a questi file di descrizione selezionano, quindi, i font latini o greci veri da usare.

Se si usassero (e di fatto si usano) i font della collezione Computer Modern, le tre famiglie si chiamerebbero `cmr`, `cmss` e cmtt; perciò sono necessari altri tre file di descrizione dei font per usare i font della collezione CBfonts insieme ai Computer Modern.

Questi sei file di descrizione esistono già in ogni distribuzione del sistema TEX.

Ma se si desiderano usare i font CBfonts insieme ad altri font diversi da quelli citati sopra? Oppure se si volessero usare font per il greco diversi da quelli della collezione CBfonts? Oppure si volessero fare le due cose assieme?

A questo punto le cose potrebbero diventare piuttosto complicate. Di soluzioni ce ne sarebbero diverse; forse la più semplice sarebbe quella di modificare la descrizione della lingua greca, in modo che cambiando lingua non si cambiasse solo la codifica dei font, ma si cambiasse anche la famiglia; per lasciare i nomi delle famiglie dei font greci distinte dai nomi delle famiglie dei font latini, bisogna che siano definite diverse macro tali da poter mettere in memoria i nomi correnti, definire i nuovi nomi con quelli delle famiglie dell'altro alfabeto, comporre con l'altro alfabeto e alla fine recuperare dalla memoria i nomi preesistenti, riselezionando i font corrispondentemente. Si potrebbe fare, ma chi cura la gestione e la manutenzione della descrizione della lingua greca non ci ha ancora pensato o forse trova che la cosa sia inutilmente complicata, tenendo conto del fatto

che il suo obbiettivo principiale è quello di rendere accessibile l'input del testo greco direttamente in greco (tastiera permettendo).

Allora? Be', attualmente bisogna affidarsi solo ai pacchetti esistenti per usare font greci diversi in contemporanea con font latini qualsiasi, sempre mantenendo la codifica *LGR* per i font greci, ma usando liberamente famiglie diverse.

I file `.fd` per font greci diversi dai CBfonts, disponibili con il sistema TEX, esistono già, come esistono pacchetti di estensione che permettono di usare quei font greci, ma bisogna ricorrere a trucchi di vario genere per sollevare l'utente dalla necessità si usare (ripetutamente, ogni volta che si cambia alfabeto) i comandi di basso livello per cambiare a mano la famiglia.

Non scendo nei dettagli, perché la cosa diventerebbe troppo lunga, ma per usare i font della collezione CBfonts con font diversi dai CM e LM, ho inserito nel pacchetto `teubner` un comando che genera automaticamente i file `.fd` necessari; una volta creati in modo abbastanza automatico questi file, l'uso dei font CBfonts con font latini diversi da quelli predefiniti, diventa semplice ed automatico; malauguratamente non è una soluzione di carattere generale; inoltre mescolare i CBfonts, con certe caratteristiche grafiche, con font latini con caratteristiche grafiche diverse, non è tipograficamente una cosa corretta. C'è solo da sperare che la sezione *greek* di `babel` venga incontro a queste necessità.

Devo essere sincero: esistono i pacchetti per usare con codifica LGR i font greci messi a disposizione dall'associazione dei tipografi greci GFS; si tratta dei font Artemisia, Baskerville, Bodoni, Complutum, Didot, Neohellenic, Porson e Solomos; i pacchetti per usarli hanno gli stessi nomi in lettere minuscole preceduti dal prefisso "gfs"; è conveniente leggerne la documentazione prima di eseguire scelte a caso. I risultati possono essere buoni o cattivi a seconda delle circostanze, cioè del testo da comporre. Per lavori di tipo filologico direi che sono poco adatti, ma siccome non sono un filologo, lascio ad altri il giudizio.

In ogni caso, se si usano i font della GFS, si può ricorrere al pacchetto `begingreek`, (Beccari, 2015). Con questo pacchetto viene messo a disposizione dell'utente un nuovo ambiente e un nuovo comando che accettano come argomento il nome di una delle collezioni indicate sopra. La sintassi è la seguente:

```
\documentclass[⟨opzioni per la classe⟩]{⟨classe⟩}
...
\usepackage[⟨famiglia⟩]{begingreek}
...
\begin{document}
\greektxt[⟨famiglia⟩]{⟨testo greco⟩}
...
\begin{greek}[⟨famiglia⟩]
⟨testo greco⟩
\end{greek}
...
\end{document}
```

dove ⟨*famiglia*⟩ è il nome di una famiglia di font greci della GFS; l'argomento facoltativo specificato nella invocazione del pacchetto provvede ad una famiglia di default; l'argomento facoltativo specificato al comando o all'ambiente indica la famiglia di font greci da usare solo per comporre il ⟨*testo greco*⟩ del particolare comando o del particolare ambiente. Questo pacchettino rende molto più facile la vita al "tastierista" che compone un documento contenente anche tutto o parte del testo in greco.

As sua volta, lavorando con pdflatex, il ⟨*testo greco*⟩ può essere immesso direttamente in greco (tastiera permettendo), oppure con la traslitterazione latina prevista per l'immissione del testo con i font in codifica LGR.

3.5 Ricapitolazione

È chiaro che alcune delle cose da fare a mano descritte in questi ultimi paragrafi, potrebbero venire automatizzate in modo da ridurle all'esecuzione di un solo script preparato apposta.

Tuttavia si consideri che questo genere di operazioni va fatto molto raramente; a me capita di rigenerare le mappe generali dei font una volta all'anno, mentre molto più raramente installo nuovi font, che non siano già installati nel sistema TEX o che disegno *ex novo* per conto mio.

Quindi le operazioni descritte sopra si possono tenere sott'occhio, semplicemente per eseguire i vari passi nell'ordine giusto, in quei rarissimi casi in cui si debbano installare nuovi font; nell'economia generale dell'uso del sistema TEX, secondo me, non vale la pena di scrivere un script apposta,

anche perché da un anno all'altro le procedure potrebbero cambiare. È più importante leggere la documentazione ad ogni aggiornamento della propria distribuzione del sistema TEX; ma non occorre altro.

La cosa veramente importante da sapere, invece, è di quali file abbia bisogno il programma pdflatex per poter usare i font. Ecco perché mi sono dilungato su queste operazioni molto tecniche. L'ho fatto non tanto per invogliare i lettori a installare altri font, anche se ci sono sufficienti informazioni per farlo quando sia necessario, quanto per spiegare la complessità dell'amministrazione dei font da parte del sistema TEX.

D'altra parte questa è una prerogativa di questo sistema di tipocomposizione, cioè di poter lavorare con qualunque font disponibile; come ho detto sembrerebbe che nel mio calcolatore, dove è installata una versione TEXLive per Mac, io possa usare circa 12 000 diversi font vettoriali; che io sappia nessun word processor ha quest'abbondanza di font a disposizione; non solo l'abbondanza, ma anche la varietà delle tipologie è grandissima, dai font bitmapped ai font Type 1, ai TrueType, agli OpenType. Il sistema TEX è un po' ruspante e poco raffinato nel gestire i font? Non direi: semplicemente è più versatile e questo è uno dei suoi pregi, non uno dei suoi difetti, anche se questa maggiore versatilità si paga un poco con una maggiore complessità.

Bibliografia

LaTeX 3 Project Team (2005). «LaTeX 2_ε font selection». PDF document. Leggibile con `texdoc fntguide` nella distribuzione TeX Live.

— (2006). «LaTeX 2_ε for class and package writers». PDF document. Leggibile con `texdoc clsguide` nella distribuzione TeX Live.

Beccari, C. (2008a). «Revived slide fonts for LaTeX». PDF document. Leggibile con `texdoc lxfonts` nella distribuzione TeX Live.

— (2008b). «The CB Greek fonts». PDF document. Leggibile con `texdoc cbgreek` nella distribuzione TeX Live.

— (2015). «The `begingreek` package». PDF document. Leggibile con `texdoc begingreek` nella distribuzione TeX Live.

Beccari, C. e Gordini, T. (2016). *L'Arte di scrivere in diverse lingue con (Xe)LaTeX*. GuIT. http://www.guitex.org/home/images/doc/ArteLingue.pdf.

— (2018). «Codifiche in TeX e LaTeX – Dal sorgente al PDF – Guida pratica per lavorare con successo». PDF document. Scaricabile da http://www.guitex.org/home/images/doc/GuideGuIT/introcodifiche.pdf.

Berry, K. (2016). «The TeX Live guide – TeX Live 2012». PDF document. Leggibile con `texdoc texlive` nella distribuzione TeX Live.

Bovani, M. (2005). «Fourier-GUT*enberg*». PDF document. Leggibile con `texdoc fourier` nella distribuzione TeX Live.

Bringhurst, R. (2004). *The elements of typographic style*. Hartley & Marks.

Caignaert, C. (2010). «Kp-fonts – The Johannes Kepler project». PDF document. Leggibile con `texdoc kpfonts` nella distribuzione TeX Live.

Charette, F. e Reutenauer, A. (2012). «Polyglossia: a babel replacement for XƎLaTeX». PDF document. Leggibile con `texdoc polyglossia` nella distribuzione TeX Live.

Desarmenien, J. (1984). «How to run TeX in a French environment: hyphenation, fonts, typography». *TUGboat*, anno 5 n° 2, pp. 91–102.

Duffner, G. (2012). «LaTeX support for EB Garamond fonts». PDF document. Leggibile con `texdoc ebgaramond` nella distribuzione TeX Live.

Garfield, S. (2012). *Sei proprio il mio typo – La vita segreta dei font*. Saggi. Adriano Salani Editore, Milano.

Giacomelli, R. (2013). «Guida tematica alla riga di comando». PDF document. Scaricabile da http//http://www.guitex.org/home/images/doc/GuideGuIT/guidaconsole.pdf.

Gregorio, E. (2011). «Introduzione a XƎLaTeX». PDF document. Scaricabile da http//profs.sci.univr.it/~gregorio/introxelatex.pdf.

ISO (2009). «ISO/IEC 80 000». PDF document. Il sito web http://it.wikipedia.org/wiki/ISO/IEC_80000 è stato aggiornato nel novembre 2012; la collezione completa consta di 14 documenti, ognuno dei quali è venduto al prezzo di 146 CHF dalla International Standards Organization, Ginevra, Svizzera.

Jeffrey, A., McDonnell, R. e Hellström, L. (2009). «*fontinst* Font installation software for TeX». PDF document. Leggibile con `texdoc fontinst` nella distribuzione TeX Live.

Knuth, D. E. (1984). *The TeXbook*. Addison Wesley. Nonostante l'anno di pubblicazione nell'arco degli anni sono state stampate molte ristampe rivedute e corrette.

— (1986). *The METAFONTbook*. Addison Wesley Publishing Company, Reading, Massachusetts.

— (1990). «Virtual fonts: More fun for grand wizards». *TUGboat*, anno 11 n° 1, pp. 13–23. Scaricabile dal sito web http//www.tug.org/TUGboat/tb11-1/tb27knut.pdf.

Knuth, D. E., Graham, R. L. e Patashnik, O. (1994). *Concrete Mathematics*. Addison Wesley Publishing Company, Reading, Massachusetts.

Lamport, L. (1986). *LaTeX: A document preparation system – User guide and reference manual*. Addison Wesley, 1ª edizione.

— (1994). *LaTeX: A document preparation system – User guide and reference manual*. Addison Wesley, 2ª edizione.

Lemberg, W. e Loyer, F. (2002). «ttf2tfm – build TeX metric file from a TrueType font». PDF document. Leggibile con `texdoc ttf2tfm` nella distribuzione TeX Live.

Nienhuys, H.-W. (2010). «mftrace». Web document. Scaricabile dal sito http://lilypond.org/mftrace/.

Pantieri, L. e Gordini, T. (2017). «L'arte di scrivere con LaTeX». PDF document. Scaricabile dal sito http://www.lorenzopantieri.net/LaTeX_files/ArteLaTeX.pdf.

Penninga, M. (2013a). «autoinst – wrapper around the LCDF TypeTools for installing OpenType fonts in LaTeX». PDF document. Leggibile con `texdoc autoinst` nella distribuzione TeX Live.

— (2013b). «Fontools». README document. Il file README, leggibile con `texdoc fontools` nella distribuzione TeX Live, è molto scarno, ma dice quali programmi vengono effettivamante usati.

Pichaureau, P. (2013). «The `mathdesign` package». PDF document. Leggibile con `texdoc mathdesign` nella distribuzione TeX Live.

Robertson, W. e Hosny, K. (2016). «The `fontspec` package». PDF document. Leggibile con `texdoc fontspec` nella distribuzione TeX Live.

Robertson, W., Stephani, P. e Hosny, K. (2015). «Experimental Unicode mathematical typesetting: The `unicode-math` package». PDF document. Leggibile con `texdoc unicode-math` nella distribuzione TeX Live.

Ryu, Y. (2000). «The `PX` fonts». PDF document. Leggibile con `texdoc pxfonts` nella distribuzione TeX Live.

Sharpe, M. (2016a). «New PX font package». PDF document. Leggibile con `texdoc newpx` nella distribuzione TeX Live.

— (2016b). «New TX font package». PDF document. Leggibile con `texdoc newtx` nella distribuzione TeX Live.

— (2018). «The Libertinus T1 Math Package». PDF document. Leggibile con `texdoc LibertinusT1Math` nella distribuzione TeX Live.

TeX Gyre (2012a). «`test-xelatex-tg_termes_math.pdf`». PDF document. Leggibile con `texdoc termes_math` nella distribuzione TeX Live.

— (2012b). «`test-xelatex-tg_pagella_math.pdf`». PDF document. Leggibile con `texdoc pagella_math` nella distribuzione TeX Live.

— (2016a). «`test-xelatex-tg_bonum_math.pdf`». PDF document. Leggibile con `texdoc bonum_math` nella distribuzione TeX Live.

— (2016b). «`test-xelatex-tg_schola_math.pdf`». PDF document. Leggibile con `texdoc schola_math` nella distribuzione TeX Live.

Thành, H. T., Ratz, S., Hagen, H., Henkel, H., Jakowski, P. e Schröder, M. (2016). «The pdfTEX user manual». PDF document. Leggibile con `texdoc pdftex` nella distribuzione TeX Live.

Voss, H. (2020). «Type 1 or OpenType Libertinus Font –Type 1 or OpenType». PDF document. Leggibile con `texdoc Libertinus` nella distribuzione TeX Live.

Wikipedia (2012). «Curva di Bézier». Sito Web. Vedi sito web http://it.wikipedia.org/wiki/Curva_di_B%C3%A9zier.

Wilson, G. (2000). «FontForge». Web page. Scaricabile dal sito http://fontforge.org/.

Wilson, P. (2014). «A few notes on book design». PDF document. Leggibile con `texdoc memdesign` nella distribuzione TeX Live.

XITS (2011). «XITS Math». PDF document. Leggibile con `texdoc xits-specimen` nella distribuzione TeX Live.

Indice analitico

B

\bm, 32
\bxseries, 61

D

\DeclareFontShape, 47, 63
\DeclareMathAlphabet, 64
\documentclass, 76

F

\fam, 32
file
- .600pk, 67
- .afm, 39, 65, 66
- .cfg, 71
- .eps, 21
- .fd, 42, 43, 51, 59, 62, 63, 65, 66, 68, 69, 75
- .log, 49, 57
- .map, 66, 69
- .mf, 42, 67
- .otf, 38, 64
- .pdf, 41
- .pfa, 66
- .pfb, 42, 66, 68
- .png, 68
- .ps, 41
- .sty, 42, 65, 69
- .tex, 29
- .tfm, 11, 16, 39–42, 46, 65–67
- .ttf, 42, 65
- .vf, 41, 42, 66
- .vpl, 41
- cmr10.600pk, 42
- dvipdfmx.map, 39, 70
- fntguide.pdf, 55
- lgrlmr.fd, 74
- miofont.map, 69
- miofont.pfb, 68, 69
- missfont.log, 43
- newpxdoc.pdf, 55
- newtxdoc.pdf, 55
- ot1cmr.fd, 43, 49
- pdftex.map, 39, 40, 70, 71
- ps2pk.map, 70
- psfonts.map, 39, 70
- t1clm2j.ld, 63
- t1lmr.fd, 50, 74
- t1phv.fd, 51
- updmap.cfg, 70–72
- updmap.log, 72

font

10.95, 63
Asana Math, 54
Cambria Math, 54
CBfonts, 12
clm2j, 63
clm2j8t10, 63
cmr10, 18, 19, 42
cmtt10, 19
ibertine, 73
Latin Modern Math, 54
Libertinus Math, 54
m, 63
miofont, 67, 70, 71
n, 63
New Computer Modern Math, 54
STIX Math, 54
STIXIntegrals, 54
T1, 63
TeX Gyre Bonum Math, 54
TeX Gyre Pagella Math, 54
TeX Gyre Schola Math, 54
TeX Gyre Termes Math, 54
XITS Math, 54
`\fontencoding`, 47, 61
`\fontfamily`, 47, 61
`\fontseries`, 47, 61
`\fontshape`, 47, 61
`\fontsize`, 61

G

`\greekfont`, 73
`\greektxt`, 76

H

`\Huge`, 61

I

`\itshape`, 61

M

`\mdseries`, 61

N

`\newfonfamily`, 73

O

opzione
greek, 75
L7X, 23
LGR, 25, 29, 74, 75
OML, 33
OMS, 33
OMX, 33
OT1, 22, 32, 33, 43
T1, 22, 23, 47, 63, 74
T2A, 24
T2B, 24
T2C, 24
T5, 22
TS1, 22
U, 33, 68
utf8, 29
X2, 24

P

pacchetto
- amsmath, 33
- amssymb, 33
- babel, 21, 26, 62, 73, 75
- beamer, 40
- begingreek, 75
- bm, 32
- cfr-lm, 62
- fontenc, 62
- fontinst, 66
- fontspec, 38, 39, 60, 65, 80
- fonttools, 65
- fourier, 55, 59
- inputenc, 29
- kpfonts, 55, 59
- latexsym, 33
- Libertinus, 59
- LibertinusT1math, 59
- lmodern, 59, 63
- lxfonts, 40
- mathdesign, 55, 60, 80
- mathpazo, 60
- mathptmx, 60
- newpxmath, 55, 59
- newpxtext, 55, 59, 60
- newtxmath, 55, 59
- newtxtext, 55, 59, 60
- polyglossia, 38, 39, 73
- PSTriks, 70
- pxfonts, 59, 60
- teubner, 75
- txfonts, 59, 60
- unicode-math, 55, 81

\pounds, 47

programma
- afm2tfm, 39
- autoinst, 65, 80
- autotrace, 68
- dvipdfmx, 16, 39, 41, 70
- dvips, 11, 16, 21, 22, 39, 41, 66, 70
- FontBook.app, 73
- FontForge, 14, 68, 69, 82
- fontinst, 6, 66, 67
- initexmf, 70
- latex, 4, 11, 16, 38, 42
- lualatex, 4, 6, 16, 38, 39, 53–56, 60, 64, 65, 70, 72, 73
- mftrace, 14, 67, 68, 80
- mktexlsr, 71
- otfinfo, 73
- otfinfo.exe, 73
- pdfLaTeX, 55
- pdflatex, 4, 6, 11, 16, 38, 39, 41, 53, 55, 56, 61, 63, 65, 66, 68–71, 73, 74, 76, 77
- pdftex, 40, 69
- pdlatex, 73
- pfb2pfa, 66
- potrace, 68
- ps2pk, 70
- PSpice, 68
- pstopdf, 16
- tex, 21, 32
- texdoc, 55, 56
- ttf2afm, 39, 65, 66
- ttf2tfm, 39, 66
- updmap, 70, 71
- updmap-sys, 70, 71
- vptovf, 41

xelatex, 4, 6, 16, 38, 39, 53, 54, 56, 60, 64, 65, 70, 72, 73

R

\rmfamily, 61

S

\scshape, 61
\selectfont, 61, 62
\setmainfont, 72
\SetMathAlphabet, 64
\setmonofont, 72
\setsansfont, 72
\sffamily, 61
\slshape, 61

T

\T1/clm2j/m/n/10.95, 63
\tiny, 61
\ttfamily, 61

U

unità di misura
ex, 12
\upshape, 61
\usefont, 61, 62
\usepackage, 76