

Introduzione al L^AT_EX per il pigro

Davide Poggiali

18 novembre 2019

Indice

Avvertenze: il presente documento è uno schema indicativo del relativo corso di 6 ore nell'ambito dell'iniziativa g2gformation. Le istruzioni qui descritte sono frutto dell'esperienza personale maturata in anni e sono presentate in modo poco ortodosso e scherzoso in modo che l'utenza abbia accesso ad un corso vasto, ma anche introduttivo e leggero. Ciò vuol dire anche che non ci assumiamo alcuna responsabilità¹.

Ciao, io sono il pigro ! Il mio compito è infastidire l'autore del testo con domande idiote!

1 Modulo 1

1.1 Introduzione e cenni storici

TEX, scritto anche *T_EX* e pronunciato come *tech*, è un programma di tipografia digitale, o *Typesetting*, adatto alla stesura di testi matematici e scientifici, creato da Donald Knuth nel 1978, contemporaneamente a *METAFONT* ed ai tipi di carattere *Computer Modern* per permettergli di scrivere il libro *The Art of Computer Programming*. È distribuito con una licenza di software libero e gode di ampia popolarità in campo universitario, specialmente nell'ambito della matematica, della fisica e della scienza dell'informazione.

Il numero di versione di TEX converge a π . La versione attuale è la 3.1415926.

¹in parole povere chi fa boiate, affari suoi hehe

L^AT_EX, il derivato più popolare di T_EX, è essenzialmente un programma per la videoscrittura di tipo WYMIWYG, estremamente diverso dal punto di vista concettuale dai programmi tradizionali WYSIWYG come Word, OpenOffice Word Processor, etc.. particolarmente adatto per la redazione di testi scientifici e molto elegante. Con L^AT_EX l'utente si può sbizzarrire e creare testi per le più svariate esigenze, dall'annuncio economico al libro.

Sì vabbeh, ma mi sembra che Word sia molto più comodo...cosa ci guadagno a scrivere con questo tuo programma?

Ecco alcuni vantaggi derivanti dall'utilizzo di questo strumento:

- una volta presa confidenza con il programma, l'utente può generare un documento di qualità superiore con poco sforzo
- mi dispiace per gli juventini presenti, ma questo programma (a differenza delle partite) non lo si può rubare: è open source²!
- proprio per questo si possono trovare o creare molti pacchetti aggiuntivi per aumentare le possibilità di utilizzo di questo software
- il programma è largamente utilizzato, quindi su internet si trovano tonnellate di guide, trucchi, soluzioni a problemi, etc..
- non si finisce mai di imparare, c'è sempre qualcosa da scoprire, e le soddisfazioni sono tante.

1.2 Installazione

I sistemi linux hanno latex nei repository ufficiali, MacTeX per Mac Os, MikTeX per Windows. Oppure basta trovare un editor web-based come latexlab, disponibile all'indirizzo <http://latexlab.org/>.

1.3 Editors

Il file sorgente .tex è un semplice file di testo che il programma utilizza per generare i files .pdf, .dvi. o .ps contenenti l'output. Il comando base è del tipo:

```
$ latex nomefile.tex
$ dvi2pdf nomefile.dvi
```

²e, fra l'altro, forza viola!

oppure:

```
$ pdflatex nomefile.tex
```

Già qui mi sembra complicato....

Eh, tranquillo! Ci sono svariati editors che aiutano l'utente nella redazione del file sorgente .tex e nella sua compilazione (comandi: latex, pdflatex). A volte bisogna indicare all'editor dove si trovano i files eseguibili di cui necessita, nel caso basta digitare da shell il seguente comando

```
$ whereis latex
```

Ci sono diversi editor per ogni s.o. utilizzato, ad esempio:

- Kile
- GNU TeXmacs
- Texmaker
- TeXnicCenter
- TeXnicle

Solitamente un buon editor presenta dei pulsanti per la compilazione rapida e per la visualizzazione del risultato e una barra laterale contenente la struttura del testo (capitoli, sezioni), un elenco delle immagini e/o delle equazioni inserite, e soprattutto un elenco di simboli T_EX suggeriti. Eventualmente uno strumento di correzione ortografica³.

1.4 Struttura generale di un file .tex

Vediamo ora come è fatto all'interno un file sorgente .tex: Il testo è simile a quello di un file .txt, ma all'interno troviamo anche comandi, ambienti e formule.

I comandi sono del tipo

```
\comando[opzioni]{nome del comando}
```

e gli ambienti (environnement)

³e installare un dizionario italiano non è così difficile !

```
\begin[opzioni]{ambiente}
.....
\end{ambiente}
```

A volte un comando o un ambiente contiene più di un input; in tal caso bisogna inserire i diversi input dentro le rispettive parentesi graffe concatenate

```
\comando[opzioni]{comando1}{comando2}{comando3}...
```

Un file .tex è diviso in due parti. La prima è il *preambolo* contenente le definizioni, fra cui:

- `\documentclass[opzioni]{tipo documento}`
dove si specifica il tipo di documento e che si vuole ottenere (fra le graffe) e delle informazioni aggiuntive sul tipo di pagina e/o la dimensione dei caratteri (fra le quadre)
- `\usepackage[]{ }`
pacchetti aggiuntivi utilizzati e i relativi parametri
- `\newcommand{\comando}{definizione}`
nuovi comandi definiti dall'utente.
- `\newtheorem{tipo di teorema}{nome visualizzato}`
parti speciali di testo, da dedicare ad esempio ai teoremi o alle definizioni.
- `\author{ } \date{ } \title{ }`

Si, mi sono già divertito! Ma a che cosa servono tutte queste cose nel preambolo?

Lo capirai più tardi! Ora proseguiamo con il nostro tour.

La seconda parte è relativa al testo effettivamente da 'stampare' sul file di output. Essa è compresa fra i due comandi:

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

Qualsiasi cosa scritta dopo il secondo comando non verrà presa in considerazione dal compilatore.

Eventualmente all'inizio del documento posso inserire titolo e indice:

```
\maketitle
\tableofcontents
```

Il testo può essere suddiviso in capitoli (solo per le documentclass=book e report), sezioni, sottosezioni e addirittura sotto-sottosezioni e paragrafi con i comandi

```
\chapter{titolo}
\section{ }
\subsection{ }
\subsubsection{ }
\paragraph{ }
```

2 Modulo 2

2.1 Il testo

Il testo è in effetti un normale testo come nei file .txt con alcune particolari modifiche. Innanzitutto gli accenti e i caratteri speciali non sono contemplati, per ottenerli bisogna utilizzare dei comandi, come

```
\‘{ } \’{ }
```

per gli accenti gravi e acuti rispettivamente. Come spesso accade in T_EX, se l'accento va su una lettera sola le parentesi graffe si omettono.

Altra particolarità sono gli spazi e il comando ‘a capo’: un doppio spazio viene contato come uno, mentre la fine di una riga non viene riportata. Occorre quindi utilizzare il comando

```
\hspace{(lunghezza)cm} \vspace{(lunghezza)cm}
```

per lasciare spazio vuoto orizzontale e verticale rispettivamente e il comando

```
\\
```

per terminare una riga di testo e andare a capo. Infine il comando % indica un commento ossia una riga che il compilatore non considera.

Inoltre il testo può essere messo in *italico usando il comando textit*, **grassetto con textbf**, sottolineato con underline⁴.

⁴le note, poco usate in ambito scientifico ma adorate dall'autore del testo si ottengono con il comando *footnote*.

2.2 Formule

Le formule sono scritte mediante un comando apposito inserito nel testo. Ad esempio il seguente codice

```
$e^{i \pi}+1=0$
```

produce come risultato $e^{i\pi} + 1 = 0$, mentre il codice

```
$$e^{i x}+1=\cos x + i \sin x$$
```

produce come risultato

$$e^{ix} + 1 = \cos x + i \sin x$$

In alternativa si può usare l'ambiente *equation*.

```
\begin{equation}
e^{i x}+1=\cos x + i \sin x
\label{equaz}
\end{equation}
```

più pratico per la gestione dei riferimenti, come vedremo più tardi.

$$e^{ix} + 1 = \cos x + i \sin x \tag{1}$$

Per scoprire altre formule matematiche basta consultare i prontuari disponibili online, come ad esempio: Formule matematiche \TeX oppure Prontuario \TeX .

2.3 Tabelle

Un altro ambiente di largo utilizzo è *table*, che riguarda le tabelle. All'interno dell'ambiente viene inserito il tabular, la tabella vera e propria, che come secondo argomento ha la disposizione nelle colonne (c=centrato,l=sx, r=dx) e le relative righe in interlinea indicate con la barra verticale |.

Nella tabella gli elementi sono separati dal simbolo & mentre le righe orizzontali in interlinea sono date dal comando *hline*. Un semplice esempio il codice:

```
\begin{table}[h] % h \ 'e la posizione, here.
\centering
\begin{tabular}{| c | | c | c | c |}
\hline
$\times$ & 1 & 2 & 3 \\ \hline \hline
```

```

1& 1& 2 & 3 \\ \hline
2& 2& 4 & 6 \\ \hline
3& 3& 6 & 9 \\ \hline
4& 4& 8 & 12 \\ \hline
\end{tabular}
\caption{Tavola delle moltiplicazioni.}
\label{tabella}
\end{table}

```

produce questo output:

×	1	2	3
1	1	2	3
2	2	4	6
3	3	6	9
4	4	8	12

Tabella 1: Tavola delle moltiplicazioni.

3 Modulo 3

3.1 Figure

Che noia “sto testo!..non è che potrei vivacizzarlo un po’, tipo con delle figure?”

Altro ambiente di fondamentale importanza è *figure*, che serve a inserire le immagini in formato raster come .jpg, .bmp, .png ma anche e soprattutto le immagini vettoriali in formato .pdf, .ps, .eps. Ovviamente l’immagine dichiarata deve essere nella stessa cartella del sorgente .tex o in un’apposita directory dichiarata nel preambolo.

Ecco un esempio di codice con il relativo output:

```

\begin{figure}[h]
\centering
\includegraphics[scale=.8]{img/chuck.png}
\caption{G2G Formation rocks}
\label{norris}
\end{figure}

```



Figura 1: G2G Formation rocks

Per grafici e figure generate dall'utente si consiglia di utilizzare immagini di alta qualità o meglio immagini vettoriali. Succede spesso infatti che una figura risulti “sgranata” e la divisione in pixel risulti evidente...proviamo con un'immagine più dettagliata (Figura ??).

Dei pacchetti aggiuntivi quali PsTriks o Tikz, che generano l'immagine da un codice fornito dall'utente. Una valida alternativa per i principianti è l'editor latexdraw, un comodo tool scritto in java che permette di esportare il codice PsTricks o un'immagine in .pdf da un disegno eseguito dall'utente. La comodità di questo editor sta nella possibilità di inserire formule all'interno delle immagini.

Ma cosa mi importa delle immagini vettoriali?

Il \LaTeX è pensato in modo vettoriale, pensa al vantaggio di poter zoomare all'infinito una figura. La qualità della pagina stampata poi è molto migliore rispetto all'immagine raster!



Figura 2: Quando Facundo

3.2 Teoremi

Theorem è un ambiente utile per i matematici, ma personalizzabile, per chiunque voglia creare all'interno del testo un'area separata con la struttura del tipo:

Titolo # numerotitolo *Intestazione o spiegazioni, anche lunghe e dettagliate, che richiedono più di una riga.* In modo da rendere maggiormente visibile una parte importante del testo o un concetto fondamentale. Le definizioni vanno nel preambolo

```
\newtheorem{theorem}{Teorema}[section]
```

Vediamo un esempio:

```
\begin{theorem}
Chi si loda si sbroda.
\end{theorem}
```

`\begin{proof}`

Chi parla troppo bene di se' stesso effettivamente pu\`o finire per sovrastimare le proprie reali capacit\`a...certo che questo non dimostra niente, non \`e vero in generale!

`\end{proof}`

Teorema 3.1. *Chi si loda si sbroda.*

Dimostrazione. Chi parla troppo bene di se' stesso effettivamente può finire per sovrastimare le proprie reali capacità...certo che questo non dimostra niente, non è vero in generale! **Dimostrazione fallita!** \square

Questo fa pensare alle differenze fra il pensiero comune e il pensiero matematico...

3.3 Riferimenti incrociati

Ma a che servivano, scusa tutti quegli href? una formula magica??

Dato che la gestione del posizionamento nel testo degli ambienti *table* e *figure* è decisa dal compilatore, spesso ci troviamo nel bisogno di fare riferimento a un'immagine, a un teorema, a un ambiente la cui posizione nel testo è lontana, magari in qualche capitolo precedente. Per far fronte al problema dei riferimenti abbiamo messo delle etichette o *label* all'interno degli oggetti precedenti, che quindi richiamiamo quando ci fa comodo con il comando

`\ref{nome}`, `\eqref{nome equaz}`

le equazioni possono anche essere chiamate con `ref`, ma `eqref` le indica come sono scritte, fra parentesi. Questo può sembrare una pignoleria, ma è di fatto più comodo a chi legge un testo scientifico avere chiaro da subito la natura del riferimento scritto in testo.

Per esempio possiamo parlare di quanto ci è utile l'equazione (??), spiegata intuitivamente in Figura ??

non è vero, che c'entra??

 e come si può dedurre dalla Tabella ??.

Si, poi? se uno mi deduce l'identità di Euler da una tabellina pitagorica fatta male è davvero un mostro!!

Ti ringrazio per lo spirito critico, ma era solo un esempio...

In conclusione i riferimenti possono alleviare le sofferenze di chi legge e di chi scrive...inoltre il pacchetto *href* permette di linkare tutti i riferimenti...Non ci credi? [??]

3.4 Bibliografia

Uno dei modi più semplici per ottenere una bibliografia è l'utilizzo dell'ambiente *thebibliography*. Se nel testo vogliamo evidenziare o discutere il contributo letterario di B.B. Cluster, in particolare la sua opera principale “Lettere di Carta”, invece che scrivere il riferimento al momento, possiamo evitare di appesantire il testo usando il comando:

```
\cite{codicelibro}
```

A un certo punto del testo inseriamo la bibliografia

```
\begin{thebibliography}{biblio}
\bibitem{codicelibro} B.B. Cluster, \emph{Lettere di Carta}, ed. Lasesta, 1997
\bibitem{codicelibro2}.....
\end{thebibliography}
```

E riferirsi al testo con una semplice formula quale:

Come scritto in [?], Cap 1 p.132,.. per dimostrare l'identità di Euler (??) basta una tavola pitagorica srausa e sbilenca mostrata in Tabella ??.

4 Modulo 4

4.1 Beamer

Un altro tipo di documento di largo utilizzo è **beamer**, usato per creare presentazioni. Un beamer è un semplice documento .tex con nel preambolo il comando

```
\documentclass{beamer}
\usetheme{ }
```

dove il comando *usetheme* contiene la scelta del tema per l'output. Di temi ce ne sono a dozzine per la rete, nel nostro esempio useremo il tema di un dottorando padovano⁵, disponibile sulla sua pagina web

<http://www.math.unipd.it/~burattin/other/tema-latex-beamer-padova/>.

Come si può vedere nell'esempio un file di questo tipo è molto simile ad un normale file L^AT_EX con qualche differenza. Ad esempio, spesso si rinuncia agli ambienti *equation*, *figure* e *table* in quanto in una presentazione poco spesso si ha il bisogno di inserire riferimenti incrociati.

⁵che ovviamente ringraziamo!

Il testo è sempre diviso in sezioni e sottosezioni, talvolta riportate in una barra superiore (dipende dal tema scelto). Ogni slide è contenuta nel proprio ambiente *slide*, all'interno del quale si inseriscono i contenuti. Le parti più importanti possono apparire dentro dei blocchi, generati dagli ambienti *block*, *alertblock*, *exampleblock*, ad esempio teoremi o avvertenze o esempi. Volendo si può inserire del testo nella slide progressivamente con il comando *pause*.

4.2 Utilizzo di pacchetti aggiuntivi

L'estensione di \LaTeX può essere potenzialmente infinita, mediante pacchetti aggiuntivi. Il sito di riferimento ufficiale per tali pacchetti è <http://www.ctan.org/>. Per installare un pacchetto bisogna scompattarlo nella cartella di riferimento `texmf/tex/latex`, avendo privilegi da amministratore.

Si, e il povero mortale che deve scrivere nel laboratorio informatico, deve stare a rompere agli amministratori per ogni pidocchioso pacchettino che voglio provare?

Ok, modo alternativo! nella home ti crei la cartella apposita così

```
$ mkdir -p ~/texmf/tex/latex/
```

dentro la quale scompatterai il pacchetto appena scaricato, poi aggiornerai il programma sull'esistenza di questa cartella utile con il comando

```
$ texhash ~/texmf
```

In seguito basta seguire gli esempi, e con un po' di creatività i pacchetti aggiuntivi possono aiutarti a generare simboli, testo con font e ambienti nuovi oppure tipi di documento per le più svariate esigenze.

4.3 Come trovare o creare simboli

Ho in mente un simbolo un po' strano da usare, ma non lo trovo nei prontuari..

Prova a disegnarlo su questo sito <http://detexify.kirelabs.org/classify.html> è un tool interessante, ti indica anche il pacchetto da usare per avere quel simbolo. Se non lo trovi devi usare un po' di inventiva, come in questo esempio:

La notazione usata dagli egiziani per indicare $\$1/n\$$ era `\$ \accentset{\begin{sideways}}{()} \end{sideways}}{n} \$`

La notazione usata dagli egiziani per indicare $1/n$ era \overline{n} .
L'alternativa meno allettante è quella di impararsi ad usare metafont⁶...

4.4 Idee per un futuro migliore

Siamo arrivati alla fine del nostro percorso guidato nel mondo del L^AT_EX e volevo inserire un paio di provocazioni per nerd e sviluppatori hardcore:

1. servirebbe integrazione fra beamer e impress.js per un nuovo stile di presentazione, che fonda le features del latex con un'impostazione più moderna e che includa facilmente filmati e gif animate (vera spina nel fianco di beamer).
2. non farebbe male, di questi tempi, un programma che compili il sorgente latex in ebook mobi/epub in modo efficiente e automatico, magari usando mathml...in questo modo ogni utente potrebbe crearsi il proprio ebook direttamente del sorgente .tex!

Riferimenti bibliografici

- [1] B.B. Cluster, *Lettere di Carta*, ed. Lasesta, 1997

⁶non sono così avanti, mi spiace!

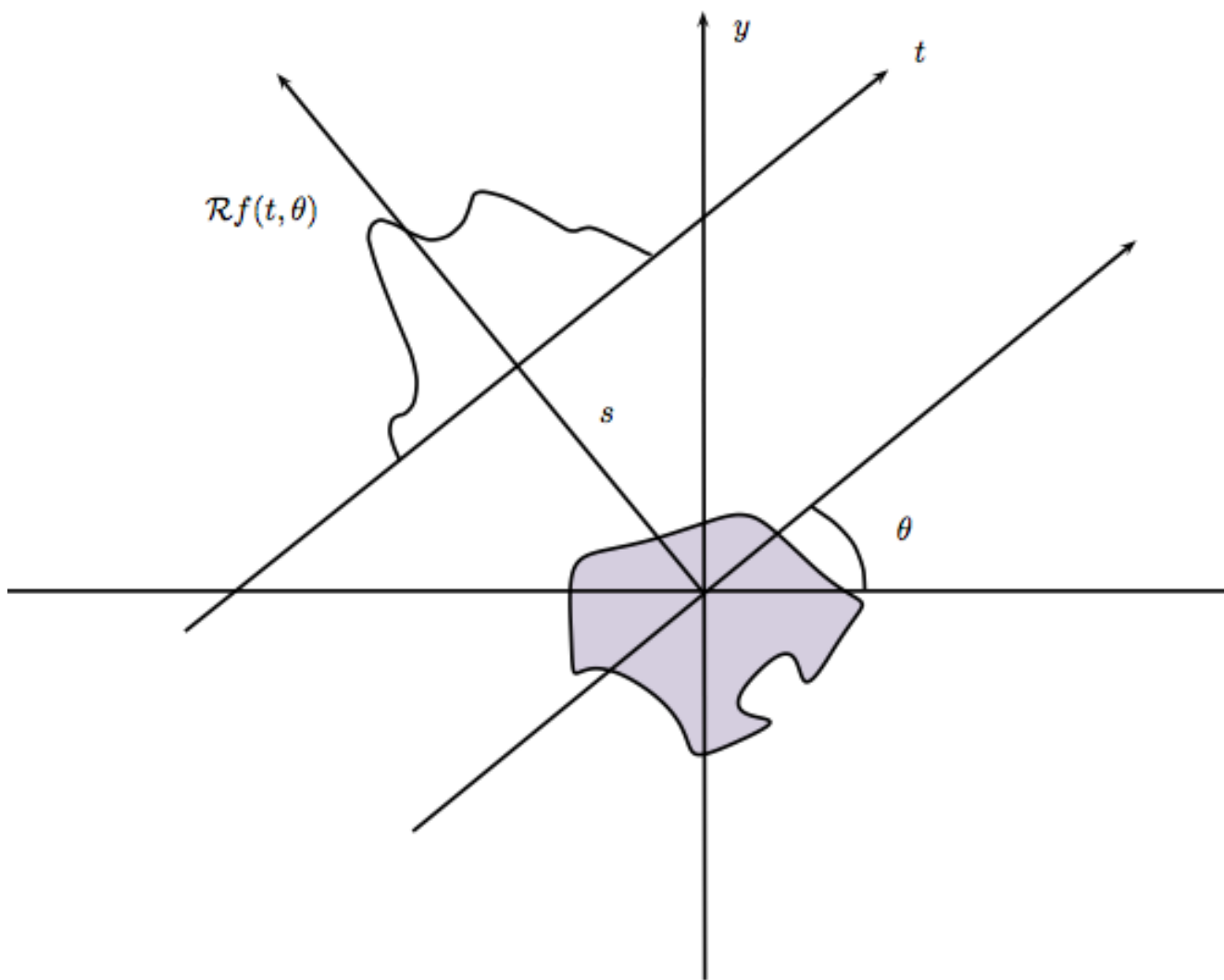


Figura 3: Esempio di figura in latexdraw: la trasformata di Radon: $\mathcal{R}f(t, \theta) = \int_{\mathbb{R}} f(t \cos \theta - s \sin \theta, t \sin \theta + s \cos \theta) ds$