

Consigli per la scrittura di una tesi di matematica in $\text{\LaTeX}$

Luca Spada

10 marzo 2026

Sommario

Scrivere una tesi di matematica non significa soltanto mettere in fila definizioni, teoremi e dimostrazioni: significa anche imparare a comunicare idee complesse in modo chiaro, preciso e leggibile. Una tesi ben scritta non nasce per magia, e nemmeno da una lunga lotta contro $\text{\LaTeX}$. Nasce piuttosto da tante piccole attenzioni: notazione coerente, frasi chiare, formule leggibili, riferimenti precisi e qualche buona abitudine tipografica. Queste pagine raccolgono alcuni consigli pratici per evitare errori frequenti e rendere il testo più ordinato ed efficace. Leggetele con attenzione, ma non consideratele definitive: se vi accorgete che manca qualche suggerimento utile, segnalatelo. Anche questo testo, come una buona tesi, può sempre essere migliorato.

1 Quattro consigli fondamentali per la scrittura di una tesi in matematica

Un testo matematico è, per sua natura, molto denso. Le definizioni e la notazione permettono di condensare idee molto complesse in poche parole o simboli. Per questo motivo, *lo stile dovrebbe essere il più semplice possibile*; è inutile appesantire il testo con perifrasi superflue o frasi troppo complicate. Inoltre, le abbreviazioni come t.c. o i simboli di quantificatore andrebbero usate solo nelle formule, non nel testo.

Quando si scrive un testo matematico, bisognerebbe cercare di mantenerlo il più possibile completo, fornendo al lettore tutte le definizioni e i lemmi necessari alla comprensione dei risultati. Tuttavia, non è possibile dimostrare tutto o introdurre tutte le nozioni necessarie. La domanda naturale “Quali nozioni posso ritenere che il lettore conosca?” non ha una risposta semplice. Una buona approssimazione potrebbe essere: *Tutte le definizioni e i teoremi visti durante il corso di laurea possono essere dati per scontati*. Ovviamente, se una nozione è particolarmente cruciale, allora può avere senso ricordare la sua definizione o le sue principali proprietà. In altri casi, si possono rivedere brevemente nozioni comuni, soprattutto allo scopo di fissare la notazione, che non è sempre universalmente stabilita.

Un altro aspetto intrinseco dei testi matematici è che sono molto strutturati, poiché l’impalcatura logica richiede che tutto sia ben definito a partire da nozioni comuni possedute da ogni matematico (o almeno dai presunti lettori del testo). È importante, quindi, *giustificare tutto ciò che si scrive, fornendo riferimenti bibliografici per ciò che non si approfondisce e dando motivazioni chiare per il resto*. Ad esempio, è sempre meglio evitare frasi del tipo “è facile vedere che...”, che possono essere sostituite, senza allungare troppo il testo, da motivazioni più chiare, come ad esempio “usando la commutatività e l’equazione (2) otteniamo facilmente...”.

Un’altra cosa importante da tenere in mente è che una buona tesi matematica non è solo formalmente corretta: deve anche aiutare il lettore a orientarsi. Ad esempio, dopo una definizione importante, conviene spesso inserire almeno un esempio, o spiegare in una frase “che cosa dice davvero” la definizione. Analogamente, prima di una dimostrazione lunga, è utile anticiparne l’idea. Ancora, può essere utile pensare alla tesi come un “racconto matematico”. Inizialmente conviene scrivere i capitoli solo come liste di definizioni e teoremi. Ma verso la fine della stesura bisogna dare alla tesi una linea narrativa. Ad esempio,

- all’inizio di ogni capitolo: dire che cosa si farà;
- alla fine: riassumere che cosa si è ottenuto;
- prima di un risultato tecnico: spiegare perché serve.

Spesso bastano un paio di righe per migliorare molto la leggibilità.

Infine, è importante usare una notazione coerente. Una volta scelta una notazione, bisogna fare attenzione a mantenerla coerente in tutta la tesi. Esempi tipici:

- non alternare $\subset$ e $\subseteq$ senza criterio;
- non chiamare lo stesso oggetto prima G , poi Γ , poi X ;
- non cambiare notazione per identità, formule, strutture, ecc.

2 Alcune note sullo stile

Un punto molto importante e spesso trascurato è che le formule in display fanno parte del periodo, quindi vanno punteggiate se la frase lo richiede. Per esempio, dopo una formula deve spesso esserci una virgola, un punto o un punto e virgola.

3 Consigli generali per scrivere in L^AT_EX

Il linguaggio L^AT_EX è un linguaggio di markup, cioè un insieme di regole che descrivono i meccanismi di rappresentazione (strutturali, semantici, presentazionali) o d'impaginazione di un testo. Questo vuol dire che chi scrive in L^AT_EX non deve preoccuparsi dell'aspetto tipografico del testo, ma solo di quello logico. L'autore inizialmente può occuparsi delle convenzioni da usare, ma una volta fissate, può concentrarsi soltanto sul contenuto del testo. L'impaginazione, l'indice (generale e analitico), e l'inserimento delle figure e delle tabelle saranno curati semi-automaticamente da L^AT_EX. In particolare, *è altamente sconsigliato cercare di forzare L^AT_EX a inserire spazi, a capo o altri strumenti tipografici*. Il testo dovrebbe solo indicare i capitoli, le sezioni, gli ambienti (teoremi, definizioni, etc.) e la divisione in paragrafi (che si ottiene lasciando semplicemente una riga vuota tra un testo e l'altro).

Quando non si è sicuri riguardo a quale notazione usare, potrebbe tornare utile utilizzare una *macro*, cioè un comando personale definito tramite `\newcommand{}`. In questo modo, se si dovesse decidere di cambiare la notazione, basterà solo modificare la definizione del comando. Ad esempio, si potrebbe definire un nuovo comando per la congruenza generata come segue:

```
\newcommand{\congener}[1]{\ensuremath{\mathtt{Cg}}(#1)}.
```

Questo comando può essere usato nel testo scrivendo, ad esempio: `\congener{S}` e ottenendo $\text{Cg}(S)$. Se poi si dovesse decidere di cambiare la notazione in $\langle S \rangle$, basterà solo cambiare la macro all'inizio del file.

3.1 Letture consigliate

- L'arte di scrivere con L^AT_EX:
http://www.lorenzopantieri.net/LaTeX_files/ArteLaTeX.pdf
- Una (mica tanto) breve introduzione a L^AT_EX 2_ε:
https://www.ptep-online.com/files/ctan/lshort_italian.pdf

3.2 Overleaf

Il sito www.overleaf.com permette di scrivere il L^AT_EX direttamente online. La versione gratuita però permette di compilare documenti abbastanza piccoli. Per questo motivo potrebbe essere utile suddividere il sorgente della tesi in file più piccoli (ad esempio, ognuno contenente un capitolo). Ciò può essere fatto usando il comando `\input`, quindi il codice assomiglierebbe al seguente:

```
\begin{document}
:
\input{intro.tex}
\input{preliminari.tex}
\input{capitolo1.tex}
\input{capitolo2.tex}
:
\bibliographystyle{plain}
\bibliography{biblio.bib}
```

In questo modo, anche un testo molto lungo può essere parzialmente compilato su overleaf, commentando le righe che importano i capitoli che in quel momento non interessano.

3.3 Bibliografia

L^AT_EX gestisce in maniera molto efficiente anche la bibliografia. *Si consiglia fortemente di sfruttare questa funzione di L^AT_EX inserendo la bibliografia in un apposito file .bib. Si vedano le pagine di aiuto di Overleaf per informazioni pratiche su come fare.*

3.4 Formato tesi dell'Università di Salerno

Il dipartimento di Matematica ha preparato un template per le tesi. Questo si basa su Regolamento per le Tesi dell'Università di Salerno. Il template è disponibile qui: <https://corsi.unisa.it/uploads/rescue/499/2347/modello-tesi.zip>. Si ricorda inoltre che è necessario inserire un abstract in doppia lingua: italiano e inglese.

4 Elenco di cose da fare e da non fare

4.1 Cose importanti

1. Usa `\colon` per le funzioni, non `:`. Confronta $f : A \rightarrow B$ con $f : A \rightarrow B$.
2. Usa `\mid` nella descrizione di un insieme invece di `|` o `\colon`.
3. Usa `\cite` per citazioni e `\label+\ref` o `\eqref` per riferimenti interni. Non scrivere a mano i riferimenti.
4. Usa l'ambiente `itemize` o `enumerate` per gli elenchi.
5. Usa `\dots` come default per i puntini di sospensione.
6. Usa l'ambiente `align` o `\[...\]` per le formule. Non usare `$$...$$` né `center`. L'ambiente `align` permette anche di suddividere su più righe le formule o le sequenze di uguaglianze. Non usare `eqnarray`.
7. Non usare `array` o `tabular` per formule, definizioni o matrici.
8. Usa l'ambiente `cases` per definizioni per casi.
9. Non lasciare righe vuote prima di `\end{proof}`.
10. Non inserire spazio verticale con `\vspace`, `\vskip`, ecc.
11. Non modificare lo spazio orizzontale, salvo per casi specifici (quantificatori finali, descrizioni di insiemi).
12. Non usare `~` per spazi forzati, tranne per evitare separazioni indesiderate (es. “in~ A ”).
13. Non forzare interruzioni di riga con `\\` o `\newline`.

4.2 altre cose

1. Usa `\emph{...}` per enfatizzare, non usare i dollari.
2. Usa `\mathrm` per parole nei simboli matematici: meglio S_{def} di S_{def} .
3. $\LaTeX$ permette di inserire virgolette di apertura e di chiusura. Le prime si ottengono con il comando `“`, le seconde con il comando `”`. Notate la differenza tra “questo” e ”questo”.

4.3 Stile

1. Non concludere una frase con il titolo di un teorema. Ad esempio, non scrivere “Così abbiamo dimostrato il” per poi cominciare un `\begin{theorem}`; le frasi devono essere concluse.
2. Non usare abbreviazioni come t.c., c.v.d., Se possibile, evitare anche sse.

5 Pacchetti utili

babel Usato per sillabazione e traduzione dei nomi (ad es. `section` $\rightarrow$ `sezione`)
`\usepackage[italian]{babel}`

amsmath, **amssymb**, **amsthm** Pacchetti AMS per la matematica. `amssymb` fornisce simboli (lettere greche, `\prod`, ecc.), `amsthm` per ambienti di teoremi.

mathtools Aggiunge molte funzioni utili a `amsmath`.

hyperref Per inserire link cliccabili alle citazioni e ai teoremi.

`\usepackage[linkcolor=blue, citecolor=purple, urlcolor=orange, colorlinks=true]{hyperref}`

enumitem Ambiente di enumerazione migliorato: `\usepackage[shortlabels]{enumitem}`

`\begin{enumerate}[\{punto\}a.]`

Elenca con il formato “punto a”, “punto b”, ecc.

makeidx Pacchetto per la creazione dell’indice analitico.

newtxtext, newtxmath Font Times New Roman.

yfonts Font aggiuntivi: gotico, ecc. Permette l’uso di `\textfrac` e `\textswab`.

showkeys Mostra le label degli ambienti nel file PDF.

`\usepackage[notref]{showkeys}`

microtype migliora la resa finale del file pdf.

cleveref sostituisce aggiunge il comando `\cref` che invece di scrivere solo il numero del riferimento capisce automaticamente se si tratta di un lemma, definizione, ecc. Così si ottiene automaticamente “Teorema 2.3”, “Lemma 4.1”, “Equazione (5)”, ecc., e si aggiorna automaticamente se decidiamo di trasformare un lemma in un teorema.

6 Fonts

6.1 Mathematics fonts

- `\mathbf Bold`
- `\mathsf Sans – serif`
- `\mathit Italics`
- `\mathtt Typewriter`
- `\mathrm Roman`
- `\mathfrak Fraktur`
- `\mathcal CALIGRAPHIC` (solo maiuscole)
- `\mathbb Blackboard BOLD` (solo maiuscole)
- `\mathpzc Zapfchancery`
- `\text` Per testo all’interno di una formula, es. $\{x \in \mathbb{R} \mid \text{esiste } y \text{ tale che } x = y^2\}$.

6.2 Text fonts

- `\textit Italics`
- `\textsl Slanted`
- `\textbf Bold`
- `\texttt Typewriter`
- `\textsc SMALL CAPS`
- `\textsf Sans-serif`
- `\emph emphasized`

6.3 Dimensioni fonts (in ordine crescente)

`\tiny, \scriptsize, \footnotesize, \small, \normalsize, \default, \large, \Large, \LARGE, \huge, \Huge.`