

TeXnica Ars

Rivista italiana
di $\text{\TeX}$, $\text{\LaTeX}$ e
tipografia digitale

35 • 2024

Editoriale

Guida allo studio di $\text{\LaTeX}$

Macros in language $\text{\LaTeX}3$

Alla scoperta di LuaCAS

$\text{\LaTeX}$ per la composizione di prove d'esame e di verifica

Composizione di atti giudiziari utilizzando $\text{\LaTeX}$

Framed material for humanists

Some macros to produce twin figures

I polinomi di Bernstein e le funzioni intere e fratte di Bézier

Concorso: Elaborazione di dati per determinare e disegnare la retta di regressione

Concorso: *La spirale delle radici quadrate*



TeXnica Ars

G_UIT – Gruppo Utilizzatori Italiani di T_EX

Direttore

Claudio Beccari

Comitato Scientifico

Renato Battistin, Francesco Biccari,
Agostino De Marco, Roberto Giacomelli,
Tommaso Gordini, Enrico Gregorio,
Guido Milanese, Ivan Valbusa

Redazione

Giulia Atanasio, Riccardo Campana, Massimo Caschili,
Gustavo Cevolani, Massimiliano Dominici,
Andrea Fedeli, Carlo Marmo, Silvia Maschio,
Federica Panarotto, Gianluca Pignalberi,
Antonello Pilu, Ottavio Rizzo,
Gianpaolo Ruocco, Emmanuele Somma,
Enrico Spinielli, Emiliano Vavassori

ArsT_EXnica è la pubblicazione ufficiale del G_UIT

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

Le uscite hanno cadenza semestrale e sono pubblicate nei mesi di Aprile e Ottobre. In particolare, la seconda uscita dell'anno contiene gli Atti del Convegno Annuale (G_UITmeeting).

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

arstexnica@guitex.org.

Pubblicare su *ArsT_EXnica*

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

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

Di volta in volta verrà fissato, e reso pubblico sulla pagina web, un termine di scadenza per la presentazione degli articoli da pubblicare nel numero in preparazione della rivista. Tuttavia gli articoli potranno essere inviati in qualsiasi momento e troveranno collocazione, eventualmente, nei numeri seguenti.

Nota sul Copyright

Il presente documento e il suo contenuto è distribuito con licenza  Creative Commons 4.0 di tipo "Attribuzione — Non commerciale — Non opere derivate". È possibile, riprodurre, distribuire, comunicare al pubblico, esporre al pubblico, rappresentare, eseguire o recitare il presente documento alle seguenti condizioni:

-  **Attribuzione:** devi riconoscere il contributo dell'autore originario.
-  **Non commerciale:** non puoi usare quest'opera per scopi commerciali.
-  **Non opere derivate:** non puoi alterare, trasformare o sviluppare quest'opera.

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

Per maggiori informazioni:

<http://www.creativecommons.org>

Associarsi a G_UIT

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

Indirizzi

Gruppo Utilizzatori Italiani di T_EX
c/o Università degli Studi di Napoli Federico II
Dipartimento di Ingegneria Industriale
Via Claudio 21
80125 Napoli – Italia
<http://www.guitex.org>
guit@guitex.org

Redazione *ArsT_EXnica*:
<http://www.guitex.org/arstexnica/>
arstexnica@guitex.org

ISSN 1828-2350 (Stampa)
ISSN 1828-2369 (Online)

15 Ottobre 2024

ArsT_EXnica

Rivista italiana di T_EX e L^AT_EX

Numero 35, Aprile 2024

- 3 **Editoriale**
Claudio Beccari
- 5 **Guida allo studio di L^AT_EX**
Maurizio Molinaro
- 23 **Macros in language L^AT_EX3**
Claudio Beccari
- 31 **Alla scoperta di LuaCAS**
Roberto Giacomelli
- 41 **L^AT_EX per la composizione di prove d'esame e di verifica**
Grazia Messineo, Salvatore Vassallo
- 65 **Composizione di atti giudiziari utilizzando L^AT_EX**
Federico Gallo
- 73 **Framed material for humanists**
Claudio Beccari
- 79 **Some macros to produce twin figures**
Claudio Beccari
- 87 **I polinomi di Bernstein e le funzioni intere e fratte di Bézier**
Claudio Beccari
- 103 **Concorso: Elaborazione di dati per determinare e disegnare la retta di regressione**
G_UT_EX Members
- 111 **Concorso: La spirale delle radici quadrate**
Il G_UT_EX

Gruppo Utilizzatori Italiani di T_EX

Editoriale

Claudio Beccari

Per diversi motivi il Meeting del 2024 non si è potuto tenere né a maggio né a ottobre, ma il Consiglio Direttivo ha deciso di pubblicare in rete questa edizione di $\text{\textit{4sTeXnica}}$, che ormai è pronta da diverso tempo.

Ritengo, insieme ad altri appassionati sostenitori dell'uso di $\text{\textit{L\TeX}}$, che questo programma sia particolarmente indicato anche per comporre testi relativi ad ogni disciplina che si insegna nelle scuole primarie e secondarie, da quando la comunicazione con gli allievi si appoggia fortemente su Internet e la maggior parte di loro hanno accesso agli strumenti necessari per comunicare con i loro docenti in modo “informatico”.

In questa edizione vengono presentati diversi lavori focalizzati principalmente sull'uso di $\text{\textit{L\TeX}}$ in ambito umanistico.

Certo il sistema $\text{\textit{T\TeX}}$, che mette liberamente a disposizione molti strumenti per comporre documenti di qualsiasi genere, specialmente con $\text{\textit{L\TeX}}$, si basa su linguaggi di programmazione che ad alcuni potrebbero sembrare ostici; ma le piccole difficoltà di questi linguaggi permettono di produrre documenti ben composti e attraenti per lettori giovani e adulti.

I lavori presentati in questa sede riguardano molti argomenti interessanti che vanno dall'uso della matematica per disegnare grafici molto belli adatti alle lezioni di geometria, alla creazione di ambienti per gestire le figure per testi di storia dell'arte; per mostrare come le formule matematiche, composte bene, usate in un testo possano produrre i risultati numerici attesi quando vengono usate in pratica; come un testo di scienze possa essere corredato di figure grandi e piccole in modo ergonomicamente valido per il lettore giovane e adulto. Come documentarsi facilmente per usare efficacemente il numero sterminato di strumenti che il sistema $\text{\textit{T\TeX}}$ mette a disposizione dell'utente.

Spero che i lettori si troveranno arricchiti di nuove conoscenze che permetteranno loro di usare $\text{\textit{L\TeX}}$ ancora meglio.

Claudio Beccari

PROFESSORE EMERITO DEL POLITECNICO DI TORINO

Guida allo studio di $\LaTeX$

Maurizio Molinaro

Sommario In questo articolo si delinea un progetto di sviluppo di una guida allo studio e alle applicazioni di $\LaTeX$. Questa guida potrebbe rappresentare un nuovo sistema di documentazione di $\LaTeX$. Si definiscono gli elementi costituenti (argomenti, fonti, formato delle schede) del progetto, attualmente ancora in uno stadio preliminare, e si riportano esempi di schede sperimentali. Infine si segnalano alcune potenziali difficoltà connesse con il progetto.

Abstract In this article a project for the development of a guide to the study and the applications of $\LaTeX$ is introduced. This guide could represent a new system for $\LaTeX$ documentation. The constitutive elements (topics, sources, sheet format) of the project, still at a preliminary stage at this time, are defined and examples of experimental sheets are presented. Finally, some potential difficulties related to the project are pointed out.

Knowledge is of two kinds. We know a subject ourselves, or we know where we can find information upon it. When we enquire into any subject, the first thing we have to do is to know what books have treated of it.

Samuel Johnson

1. Motivazioni e obiettivi del progetto

Scopo di questo articolo è di descrivere le caratteristiche essenziali di un progetto per la documentazione di $\LaTeX$.¹

Il progetto consiste nella costruzione di quella che si può chiamare una “Guida allo studio e alle applicazioni di $\LaTeX$ ”. Questa guida potrebbe rappresentare un’innovazione significativa nel panorama degli strumenti a disposizione degli utenti di $\LaTeX$.

Punto di partenza

L’idea del progetto deriva da un’esperienza personale, di utente di $\LaTeX$ che ha superato la fase iniziale di apprendimento e intende affrontare argomenti via via più complessi. Spesso si scopre che le informazioni utili sono disperse in una grande quantità di fonti eterogenee e non coordinate; spesso si fa fatica a trovare subito la soluzione migliore tra le molte (o moltissime) esistenti; spesso ci si smarrisce davanti alla varietà degli strumenti disponibili. Ne nasce una sensazione di disagio, se non addirittura di sfiducia nelle proprie capacità di fare progressi reali. È molto probabile che questa esperienza sia condivisa da altri utenti.

1. Nel seguito dell’articolo si fa riferimento, per brevità, soltanto a $\LaTeX$, ma naturalmente sono compresi anche predecessori e successori ($\TeX$, $\text{PDF}\LaTeX$, $\text{Xe}\LaTeX$, $\text{Lua}\LaTeX$).

Documentazione esistente

La citazione che segue (C. Beccari, Prefazione a *L^AT_EX*pedia, gennaio 2017) riassume in modo molto efficace le difficoltà inerenti allo studio di *L^AT_EX*.

Il linguaggio *L^AT_EX* è di per sé molto vasto. Le migliaia di pacchetti esistenti lo estendono ancora di più. È impossibile descrivere compiutamente il numero sterminato di funzioni disponibili. Perfino il testo “ufficiale” *L^AT_EX* Companion, che ha più di mille pagine, non è e non può essere completo. Non è facile scrivere di *L^AT_EX*, poiché il sistema viene aggiornato continuamente e ogni settimana vengono modificati o aggiunti una cinquantina di pacchetti. Oggi un’installazione completa di questo programma comprende più di tremila pacchetti. Quindi per scrivere una guida su *L^AT_EX* o si resta sul generale o la si aggiorna spesso.

A grandi linee, i materiali di studio riguardanti *L^AT_EX* comprendono: *manuali generali*, non necessariamente pubblicati come libri; *manuali particolari*, dedicati a uno o a pochi argomenti specifici, quasi sempre resi disponibili in un sito web; *articoli*, generalmente pubblicati in riviste specializzate.

A questa quantità di fonti di informazioni, già di per sé enorme, si devono aggiungere i *manuali dei pacchetti*. I pacchetti fondamentali, o almeno di grande utilità, sono sicuramente in numero superiore a 300; alcuni di essi, per di più, sono corredati di manuali aventi dimensioni straordinariamente grandi (per esempio, il manuale di *pgf / tikz* attualmente ha la bellezza di 1321 pagine!).

Tuttavia i manuali generali non sempre sono sufficienti per un approfondimento adeguato; i manuali specifici e gli articoli coprono solo uno o pochi argomenti; i manuali dei pacchetti sono eterogenei e non sempre di facile lettura, e ovviamente non offrono un quadro di assieme.

In molti casi, se si vuole risolvere un problema un po’ particolare, la via migliore è cercare informazioni nel forum del *C_TAN* o in un sito come TeX-LaTeX StackExchange. Così si riesce abbastanza rapidamente a capire dove si possono trovare le risposte nei manuali dei pacchetti, mentre la consultazione diretta dei manuali spesso equivale a tentare l’avventura. Inoltre è possibile accedere a soluzioni valide ideate *ad hoc* dai moderatori del forum o da altri utenti. Purtroppo però una ricerca tra queste “discussioni” è quasi sempre estremamente faticosa, a causa dell’assoluta mancanza di organizzazione.

Le considerazioni sulle fonti di informazioni sono riprese e sviluppate nel §3.

Caratteristiche della guida

La guida è costituita da un insieme di “schede tematiche”, una per ciascuno degli argomenti che sono stati definiti (attualmente 50, vedi §2). Ogni scheda contiene una breve introduzione all’argomento trattato e alcuni elenchi di riferimenti, aggiornati e accessibili. Ogni scheda, pertanto, è una specie di mappa per orientarsi nell’argomento e al tempo stesso fornisce spunti per gli approfondimenti successivi. Non c’è una progressione da argomenti basilari ad argomenti avanzati, ma si mantiene un livello medio uniforme.

Le schede hanno dimensioni variabili secondo la complessità dell’argomento, per lo più comprese tra le 2 e le 5 pagine. Una valutazione preliminare molto sommaria porta a prevedere che la guida comprenda tra le 150 e le 250 pagine: un ingombro totale non eccessivo.

Le dimensioni della guida sono contenute proprio perché il testo è fortemente ridotto e non si pretende di spiegare (o rispiegare) gli argomenti.

Questa guida si propone dunque come un'opera di consultazione: non sostituisce lo studio accurato dei vari argomenti, in particolare dei manuali dei pacchetti, ma può avere una funzione propedeutica.

Essa dovrebbe riuscire utile per una fascia di utenti che si può presumere abbia una certa consistenza: utenti intermedi tra neofiti ed esperti, desiderosi di trovare soluzioni valide a problemi non proprio elementari, possibilmente in tempi ragionevoli e senza finire su un binario morto.

2. Argomenti di studio

Il primo passo nella costruzione della guida è la definizione degli argomenti da trattare nelle schede.

La tabella 1 riporta l'elenco degli argomenti attualmente definiti, raggruppati in sezioni.²

Tabella 1. Argomenti di studio.

ARGOMENTI GENERALI

1. Informazioni generali su $\text{\TeX}$, $\text{\LaTeX}$, ecc.
2. Organizzazione dei documenti
3. Formati e strutture dei file sorgenti

STRUMENTI DI SVILUPPO

4. Ambienti di sviluppo
5. Programmi di supporto

PROGRAMMAZIONE

6. Definizione / ridefinizione di comandi / ambienti
7. Registri (contatori, lunghezze, ecc.)
8. Strutture di controllo (condizionali, cicli, ecc.)
9. Calcoli
10. Costruzione di pacchetti e di classi
11. Altri elementi di programmazione

TIPI DI TESTO

12. Libri
 13. Articoli e affini
 14. Tesi di laurea
 15. Relazioni tecniche
-

Continua nella pagina seguente

2. Per ragioni di chiarezza alcuni titoli della tabella sono più lunghi di quelli che presumibilmente appariranno nella scheda corrispondente.

Continua dalla pagina precedente

- 16. Presentazioni
- 17. Altri testi (lettere, CV, documenti legali, ecc.)

COMPONENTI STRUTTURALI

- 18. Frontespizio e retro-frontespizio
- 19. Indice generale
- 20. Indice delle figure, delle tabelle e altri indici
- 21. Bibliografie
- 22. Glossari
- 23. Indici analitici

ELEMENTI DEL TESTO

- 24. Geometria e struttura della pagina
- 25. Selezione e uso delle lingue
- 26. Composizione del testo in prosa
- 27. Composizione del testo in poesia
- 28. Oggetti mobili
- 29. Inserti vari nel testo
- 30. Elenchi
- 31. Note
- 32. Citazioni
- 33. Espressioni matematiche e di altre scienze
- 34. Figure esterne (importate)
- 35. Diagrammi
- 36. Grafici
- 37. Disegni tecnico-scientifici
- 38. Tabelle
- 39. Listati e algoritmi
- 40. Riferimenti incrociati

MISCELLANEA

- 41. Selezione e uso dei caratteri
- 42. Caratteri alfanumerici, punteggiatura, spazi, altri segni grafici
- 43. Simboli scientifici e tecnici; altri simboli
- 44. Selezione e uso dei colori
- 45. Caselle di testo, contorni, filigrane

APPLICAZIONI

Continua nella pagina seguente

Continua dalla pagina precedente

-
- 46. Testi filologici (edizioni critiche)
 - 47. Testi religiosi
 - 48. Dizionari
 - 49. Testi scacchistici e di altri giochi
 - 50. Testi musicali
-

Il numero consistente di argomenti permette di alleggerire le singole schede e di rendere più comoda la loro consultazione.

Scorrendo l'elenco si vede subito che vi sono forti differenze, quanto a complessità, tra i vari argomenti, e questo si riflette inevitabilmente sulla mole delle schede. Pertanto non è da escludere che in seguito risulti opportuno distribuire gli argomenti più complessi su più schede, mentre è meno probabile l'accorpamento di qualche scheda.

Non è nemmeno da escludere l'aggiunta di argomenti in qualche sezione; sono meno probabili, invece, l'aggiunta di nuove sezioni o lo spostamento di qualche argomento da una sezione a un'altra.

La definizione degli argomenti e il loro ordinamento obbedisce a una certa logica, ma, come spesso avviene, ciò che sembra logico a chi scrive non è affatto detto che sembri altrettanto logico a chi legge. Per questo motivo è fondamentale il ruolo dei collegamenti tra una scheda e l'altra (vedi §4).

3. Fonti di informazioni

Nel §1 si è già avuto modo di osservare che le fonti di informazioni su $\text{\LaTeX}$ sono quasi illimitate; pertanto un problema tutt'altro che trascurabile è come individuare le fonti più adeguate per gli obiettivi della guida. In un certo senso, il censimento e la classificazione delle fonti potrebbero già essere riguardati come un risultato positivo di questo lavoro.

Sono state individuate cinque categorie di fonti:

- Manuali generali
- Manuali particolari
- Pacchetti e classi
- Articoli e affini
- Contributi in rete

Per tutte le fonti i due principali criteri di validità dovrebbero essere: fonte significativa e fonte accessibile.

Nel seguito si svolgono alcune considerazioni su ciascuna categoria, con l'attenzione già rivolta al trasferimento delle informazioni nelle schede.

Manuali generali

I manuali generali possono situarsi su vari livelli, dal *tutorial* al testo avanzato. Possono essere pubblicati in forma di libro tradizionale o essere resi disponibili in un sito web per l'accesso gratuito. Il numero di manuali pubblicati è esorbitante, come si può appurare, per esempio,

cercando in uno dei siti che ne curano la vendita. La maggioranza dei manuali è in inglese; se si estende l'orizzonte ad altre lingue (*in primis* tedesco, spagnolo, francese), il loro numero cresce ancor più.

Per non portare l'utente fuori strada, il livello del manuale dev'essere chiaramente specificato e dev'essere preferibilmente medio-avanzato. Inoltre i manuali non devono essere troppo datati, con l'eccezione delle pietre miliari come *The L^AT_EX Companion*.

In genere, sembra consigliabile non eccedere nel numero di manuali generali a cui si fa riferimento nelle schede: lo stesso argomento può essere trattato più o meno con la stessa profondità, e quindi c'è il rischio che ogni fonte successiva aggiunga poco o nulla alle precedenti.

Inoltre si deve usare la massima cautela e distinguere tra i manuali effettivamente consultati e quelli di cui si dispone soltanto dell'indice generale.

La tabella 2 a fronte contiene un elenco di manuali generali "candidati". Si tratta di un elenco puramente indicativo, o meglio provvisorio, con dati bibliografici incompleti. La distinzione tra i gruppi non è in realtà così rigida, perché alcuni manuali pubblicati come libri si possono trovare anche in rete e alcuni manuali originariamente in rete sono stati pubblicati come libro. È da notare che di alcuni manuali esistono, accanto all'originale, traduzioni in varie lingue. Un esempio è *The Not So Short Introduction to L^AT_EX 2_ε*, di cui merita segnalare la versione italiana, anche più ricca dell'originale.

Manuali particolari

I manuali particolari costituiscono un insieme piuttosto eterogeneo. Vi si possono annoverare:

- le guide tematiche del G_UT, come "Comporre tabelle e matrici" di C. Beccari o "Comporre la bibliografia in L^AT_EX" di F. Vomiero;
- i manuali dedicati alla descrizione e alle applicazioni di uno o più pacchetti, come "Introduzione all'uso di TikZ in ingegneria" di C. Fiandrino o "Breve guida ai pacchetti di uso più comune" di E. Gregorio, anch'essi reperibili nel sito del G_UT;
- altri manuali dedicati ad argomenti specifici, come "The Comprehensive L^AT_EX Symbol List" di S. Pakin o "Math mode" di H. Voß;
- i manuali dedicati alla programmazione, come "Appunti di programmazione T_EX – L^AT_EX" di E. Gregorio o "Guida al linguaggio Lua per LuaT_EX", di R. Giacomelli, anch'essi reperibili nel sito del G_UT.

I manuali particolari sono quasi tutti in formato elettronico; sono più rari quelli pubblicati come libri (un esempio è "More Math into L^AT_EX" di G. Grätzer).

Pacchetti e classi

Per i pacchetti e le classi il punto di partenza è la documentazione che li accompagna, leggibile con il comando di sistema `texdoc`.

Accanto al manuale "ufficiale" alcuni pacchetti offrono anche documenti contenenti esempi applicativi, che a tutti gli effetti possono essere considerati manuali particolari; altri pacchetti, invece, riportano gli esempi applicativi all'interno del manuale ufficiale.

È facile prevedere che la selezione dei pacchetti da elencare nelle schede presenti una notevole complessità. Il numero di pacchetti riguardanti un argomento può essere molto grande; vi possono essere sovrapposizioni, diramazioni, estensioni, restrizioni; vi possono

Tabella 2. Manuali generali.

MANUALI PUBBLICATI COME LIBRI		
D. E. Knuth	The $\text{\TeX}$ Book	ed. 1984
L. Lamport	$\text{\LaTeX}$: A document preparation system	ed. 1994
H. Kopka, P. W. Daly	Guide to $\text{\LaTeX}$	ed. 2003
F. Mittelbach (...)	The $\text{\LaTeX}$ Companion	ed. 2004
M. Goossens (...)	The $\text{\LaTeX}$ Graphics Companion	ed. 2008
V. Eijkhout	$\text{\TeX}$ by Topic	ed. 2013
G. Grätzer	Practical $\text{\LaTeX}$	ed. 2014
S. Kottwitz	$\text{\LaTeX}$ Beginner's Guide	ed. 2021
MANUALI DISPONIBILI IN INTERNET (IN ITALIANO)		
C. Beccari	Il $\text{\LaTeX}$ Reference Manual commentato	ed. 1/2019 (v.2.9.10)
C. Beccari (coord.)	Guida $\text{\G}\text{\I}\text{\R}$	ed. 4/2021 (v.0.99.50)
L. Pantieri	$\text{\LaTeX}$ pedia	ed. 11/2021
MANUALI DISPONIBILI IN INTERNET (IN INGLESE)		
N. Talbot	$\text{\LaTeX}$ for Complete Novices	ed. 9/2012
WikiBook		ed. 6/2016
P. Flynn	An introduction to typesetting with $\text{\LaTeX}$	ed. 6/2017
T. Oetiker (...)	The Not So Short Introduction to $\text{\LaTeX}$ 2 ϵ	ed. 3/2021 (v.6.4)
K. Berry (coord.)	Unofficial Reference Manual	ed. 9/2023

essere pacchetti obsoleti o non raccomandabili per nuovi lavori o di applicabilità limitata (p.es. per una sola lingua).

Una discussione critica, anche se appena accennata, dei pregi e dei difetti di ciascun pacchetto non sarebbe compatibile con il formato della scheda: è importante, perciò, che l'elenco dei pacchetti sia studiato con cura e che le descrizioni siano efficaci.

Articoli e affini

Gli articoli che interessano sono quelli pubblicati nelle riviste specializzate, quali:

- Ars $\text{\TeX}$ nica
- TUGboat
- Die $\text{\TeX}$ nische Komödie (DTK)
- Cahiers GUTenberg / Lettre GUTenberg
- The Prac $\text{\TeX}$ Journal (fino al 2012)

A questi articoli si possono affiancare:

- le presentazioni a convegni come G_UIT Meeting, TUG, BachoT_EX, Tagung DANTE, Journée GUTenberg (spesso derivanti da articoli e spesso non facilmente reperibili);
- gli scritti aventi l'estensione di un articolo e resi disponibili in un sito web, come “Creating a L^AT_EX Minimal Example” di N. Talbot.³

Contributi in rete

Come accennato nel §1, tra i contributi in rete vi sono, in primo luogo, il Forum del GuIT (<https://www.guitex.org/home/it/forum/index>) e TeX-LaTeX StackExchange (<https://tex.stackexchange.com/>), ma non sono gli unici.

Questi siti sono una miniera di materiali interessantissimi, ma purtroppo la selezione dei contributi più significativi non è affatto agevole. Molte “discussioni” sono prolisse ed estremamente frammentate; manca un’organizzazione che permetta di individuare velocemente l’importanza o la novità del contributo; manca un’organizzazione temporale, in quanto sono collocati sullo stesso piano contributi recenti e contributi che risalgono anche a 10-15 anni fa.

Aggiornamento delle fonti

È evidente che questa guida ha valore se le fonti sono aggiornate, ma la difficoltà nel tenerla aggiornata è più o meno grande secondo le fonti stesse.

Molti manuali, specialmente quelli dei pacchetti, sono aggiornati anche a distanza di pochi mesi, cioè a un ritmo ben superiore a quello della guida (ragionevolmente sei mesi – un anno). È vero, tuttavia, che le differenze tra una versione e l’altra dei manuali raramente sono così macroscopiche da incidere sul livello di informazione che può offrire la guida. È dunque necessario che la versione del manuale utilizzata nella guida sia indicata chiaramente, come si vede nella tabella 2.

La questione dell’evoluzione continua non sussiste, naturalmente, per i libri pubblicati in forma tradizionale e per gli articoli.

Per altro, non è detto che un libro o un articolo pubblicati 10-15 anni fa siano per forza di scarsa utilità: si è già accennato alle pietre miliari come *The L^AT_EX Companion*.

4. Formato e contenuto delle schede

Dopo aver individuato le fonti più adeguate per gli obiettivi della guida, si deve risolvere il problema di come trasferire le informazioni nelle schede.

Come criterio generale, le informazioni devono essere compattate per sfruttare al massimo lo spazio disponibile nelle schede, che è necessariamente limitato.

Ne scaturiscono alcune esigenze contrastanti: i contenuti delle schede devono essere sintetici, ma comprensibili senza troppo sforzo; le schede devono essere il più possibile autonome, ma evitare il più possibile ripetizioni che fanno consumare spazio prezioso.

Sezioni

Ogni scheda è divisa in 7 sezioni, che corrispondono in massima parte alle fonti di informazioni discusse nel §3.

3. Alcuni di questi articoli sono in realtà un incrocio tra un articolo e un manuale: sono manuali in veste di articolo.

- Introduzione
- Manuali generali
- Manuali particolari
- Articoli e affini
- Pacchetti e classi
- Siti web (Forum ecc.)
- Collegamenti con altre schede

Si può ritenere che pacchetti e classi di fatto siano disgiunti; le classi dovrebbero essere presenti solo nelle schede che trattano i tipi di testo.

Formato e grafica

Il formato di pagina prescelto è A4, preferibile a B5 soprattutto perché rende disponibile uno spazio maggiore.

La definizione del formato grafico è un punto cruciale; può essere richiesto non poco tempo prima di arrivare a un risultato soddisfacente. Si prevede di studiare almeno due soluzioni; per ciascuna di esse è bene provare tutte le sezioni, in modo da accertare che l'inserimento e l'aggiornamento dei dati siano sufficientemente rapidi. Non bisogna anteporre l'estetica alla praticità, in quanto le schede sono piuttosto diverse una dall'altra come dimensioni e le revisioni sono piuttosto frequenti.

I titoli delle sezioni devono essere dotati di uno sfondo colorato per favorirne l'identificazione. I colori devono essere scelti in modo da non rendere illeggibile il titolo della sezione, qualora la scheda sia stampata in bianco e nero.

Per aumentare la quantità di informazioni nella scheda, i testi delle sezioni potrebbero essere composti in corpo 10, anziché 11.

Non si può escludere la presenza di sezioni senza materiale, soprattutto per i manuali particolari e per gli articoli. In tal caso, è preferibile lasciare vuota la sezione, anziché ometterla, perché così le schede sono più uniformi e si dà all'utente una percezione migliore dello stato della documentazione.

Introduzione

Il testo introduttivo assolve diverse funzioni: sottolineare l'importanza dell'argomento, chiarire qualche aspetto meno evidente, tracciare le linee di evoluzione dal nucleo di $\text{\LaTeX}$ ai pacchetti, ecc.

Riferimenti ai manuali e agli articoli

I riferimenti ai manuali devono essere precisi (paragrafo o pagina), altrimenti l'utente è costretto a cercare i luoghi interessati direttamente nel manuale, il che farebbe scendere drasticamente l'utilità delle schede. D'altra parte, quanto più precisi sono i riferimenti, tanto più la manutenzione della guida diventa onerosa, a causa dell'evoluzione continua di molte fonti (vedi §3).

Un altro motivo che rende difficile trasferire nelle schede i riferimenti in modo sintetico è la forte dispersione degli argomenti propria di molti manuali generali. Per esempio, nella Guida $\text{\GjT}$ (ma anche nella $\text{\LaTeX}$ pedia o nell'*Unofficial Reference Manual*) può accadere che lo stesso argomento sia trattato in 5-6 luoghi diversi: se si vogliono riportare tutti, si ha subito un

notevole appesantimento della scheda corrispondente. Occorre fare delle scelte, talvolta non così immediate. Fortunatamente molti di questi manuali sono dotati di ricchi indici analitici, che consentono di integrare i riferimenti che per ragioni di spazio non possono essere inclusi nelle schede.

Per i manuali generali conviene utilizzare delle sigle, anziché un formato “bibliografico” (non importa con quale stile), che ridurrebbe lo spazio disponibile nella scheda per i riferimenti o per i commenti, senza contare la fastidiosa ripetizione da una scheda all’altra.

Ecco un possibile schema di codificazione di alcuni dei manuali generali contenuti nella tabella 2.

LPE	<i>LT_Xpedia</i>
URM	<i>Unofficial Reference Manual</i>
GGU	<i>Guida G_UT_X</i>
WIK	<i>WikiBook</i>

L’elenco completo dei manuali in formato “bibliografico” e delle sigle corrispondenti dev’essere riportato in un’apposita tabella collocata all’inizio della guida.

Per i manuali particolari e per gli articoli, invece, conviene fornire nelle singole schede i dati completi, anche a costo di ripetizioni, perché per queste fonti le “informazioni specifiche” sono più ridotte rispetto ai manuali generali, mentre i codici potrebbero riuscire incomprensibili. A questo scopo bisogna lavorare su file bibliografici ben strutturati e scegliere opportunamente lo stile bibliografico. Come ordinamento sembra preferibile l’ordine cronologico inverso.

Riferimenti ai pacchetti

Una delle sezioni più estese è indubbiamente quella dei pacchetti, mentre meno critica appare quella delle classi (per le schede che la contengono). Con ogni probabilità la sezione dei pacchetti è anche la più irregolare come dimensioni, in quanto si può stimare che per alcuni argomenti si possa arrivare a elencare fino a 10-15 pacchetti.

Conviene affiancare la data dell’ultimo aggiornamento e la versione del pacchetto, anche se, per quest’ultima, il risultato è disomogeneo perché non esistono regole fisse per la numerazione.

La data dell’ultimo aggiornamento permette di avere una prima idea dell’utilità del pacchetto. Per altro, anche se un pacchetto è stato aggiornato da poco tempo, ciò non significa che si siano avute revisioni sostanziali nelle sue funzioni: molto spesso si è trattato di correzioni di errori o di aggiunte secondarie.

L’ordinamento dei pacchetti dev’essere adattato all’argomento: in ordine alfabetico o di importanza (stimata) o per sotto-argomento.

Riferimenti ai siti web

Per la sezione dei siti web, una volta superata la difficoltà della selezione dei contributi più interessanti (vedi §3), bisogna decidere in che ordine presentarli: in ordine alfabetico o in ordine temporale o in ordine di importanza (stimata) o per sotto-argomento. I titoli delle “discussioni” spesso sono tutt’altro che illuminanti.

Per dare all’utente una percezione di quanto il contributo possa essere attuale è opportuna una forma (anche minima) di datazione: potrebbero essere sufficienti il mese e l’anno in cui la discussione ha avuto inizio. (Incidentalmente, si è notato che alcune discussioni nascono in un certo momento e sono poi riprese anche a parecchi anni di distanza.)

Commenti

In quasi tutte le sezioni è opportuno che i riferimenti siano accompagnati da brevi commenti che possano aiutare l'utente. Il formato e l'estensione dei commenti è assai variabile da una sezione all'altra. Per i manuali e per gli articoli la difficoltà consiste nel concentrare le informazioni nel poco spazio lasciato libero dai riferimenti.

Collegamenti ipertestuali

Per rendere più comoda la consultazione della guida, che si può supporre sia in formato elettronico, conviene fare ampio ricorso a collegamenti ipertestuali. Essi sono fondamentali, per esempio, nella sezione “Collegamento con altre schede”. Per alcune sezioni i collegamenti ipertestuali possono essere introdotti abbastanza facilmente, per altre molto meno.

Considerazioni finali

Non è detto che l'organizzazione in schede sia la migliore per tutti gli argomenti: alcuni si prestano meglio di altri, anche in dipendenza dalla loro complessità.

Parimenti, non è detto che lo stesso formato si adatti bene a tutte le schede.

5. Esempi di schede

Come esempi di schede si propongono i seguenti argomenti:

- Indici analitici (scheda 23)
- Note (scheda 31)
- Tabelle (scheda 38)

Le schede mostrate nelle pagine seguenti sono del tutto sperimentali; si intende unicamente dare un'idea di massima della loro struttura. Il formato di presentazione è il primo dei due previsti, quello che è stato portato più avanti nello studio.

23	COMPONENTI STRUTTURALI INDICI ANALITICI	Rev.24/3/2024 1/2
----	--	-----------------------------

Gli indici analitici sono particolarmente utili nei manuali e nei testi didattici, destinati anche alla consultazione. Purtroppo la loro composizione richiede spesso uno sforzo non proporzionato al risultato atteso, anche se si fa ricorso a strumenti che diano una certa assistenza all'autore. I pacchetti che estendono le funzioni di $\text{\LaTeX}$ standard si propongono come obiettivi principali: permettere di creare indici analitici multipli per lo stesso testo; generare l'indice analitico con una sola compilazione del sorgente; usufruire di una maggiore varietà nella disposizione e nella formattazione delle voci dell'indice.

MANUALI GENERALI

LPE capitolo 12, p.273-277; §14.3, p.297
URM §25.2, p.235-241
GGU capitolo 12, p.287-296; §19.8.2, p.485-489; §29.13.1, p.762-763
WIK capitolo 34, p.421-427

MANUALI PARTICOLARI

P. Chen (...) *Index Preparation and Processing*, 2014

ARTICOLI

G. Milanese Forindex: computer-aided indexing, *ArsTeXnica* (22), ottobre 2016, p.64-68
AMS T. Group *TeXnical Tips for Producing a "Clean" Index*, v.0.95, 14/12/2015

23	COMPONENTI STRUTTURALI INDICI ANALITICI	Rev.24/3/2024 2/2
PACCHETTI		
imakeidx	(v.1.3e, 18/10/2016) Enables the user to produce and typeset one or more indexes simultaneously with a document. Works in LaTeX documents processed with pdfLaTeX, XeLaTeX and LuaLaTeX. If makeindex is used for processing the index entries, no particular setting up is needed. Using xindy or other programs it is necessary to enable shell escape.	
indextools	(v.1.7.0, 14/10/2023) Enables the user to produce and typeset one or more indices simultaneously. Works in LaTeX documents processed with pdfLaTeX, XeLaTeX and LuaLaTeX. This is a fork of imakeidx, with new features and fixed bugs.	
repeatindex	(v.0.01, 20/7/2016) Repeats item of an index if a page or column break occurs within a list of subitems. This helps to find out to which main item a subitem belongs.	
SITI WEB		
Forum $\LaTeX$ Personalizzare indice analitico (novembre 2022) Indice analitico (giugno 2019)		
TeX StackExchange How to create a new index entry using imakeidx package (febbraio 2024) Makeidx, Subentries and dashes (novembre 2014) Footnote numbers in index (maggio 2012)		
COLLEGAMENTI CON ALTRE SCHEDE		
Scheda 5	(Programmi di supporto)	

31	ELEMENTI DEL TESTO NOTE	Rev.24/3/2024 1/2
----	----------------------------	----------------------

L'importanza del trattamento corretto delle note non dev'essere sottovalutata, non solo per la frequenza con cui esse si incontrano in tutti i tipi di testo, ma anche per la possibile presenza di una grande quantità di errori e di incongruenze.

Le note possono essere collocate a piè di pagina, oppure nel margine, oppure possono essere raggruppate in una raccolta in fondo a ogni capitolo o al testo completo. I pacchetti che estendono le funzioni di $\LaTeX$ standard si propongono come obiettivi principali: offrire una varietà maggiore di soluzioni per le prime due collocazioni e permettere di coprire anche la terza.

Nelle edizioni critiche dei testi letterari le note possono avere una complessità straordinaria.

MANUALI GENERALI	
LPE	§4.6, p.87-88; (§14, p.295) – Note a margine e a piè di pagina
URM	Ch. 11, p.104-108; §13.1, p.131 – Note a piè di pagina §15.4, p.144 – Note a margine
GGU	§7.8; §§29.2-3-4; §29.4.3; §29.8.1; §29.10.1-2 – Note a margine e a piè di pagina
WIK	Ch. 19, p.251-256; §24.3, p.296 – Note a margine e a piè di pagina

MANUALI PARTICOLARI

ARTICOLI

31	ELEMENTI DEL TESTO NOTE	Rev.24/3/2024 2/2
----	----------------------------	----------------------

PACCHETTI	
footmisc	(v.6.0f, 17/7/2023) A collection of ways to change the typesetting of footnotes. Some of the functions of the package are overlapped with the functionality of other packages.
sidenotes	(v.1.00a, 25/3/2020) Allows typesetting of texts with notes, figures, citations, captions and tables in the margin. This is common (for example) in science textbooks.
enotez	(v.0.10d, 9/1/2022) Allows nested endnotes, supports hyperref and provides means for easy customization of the list of notes. Requires the expl3 bundle and packages from the $\text{\LaTeX}$ 3 package set.

SITI WEB
Forum $\text{\LaTeX}$ Note a piè di pagina "miste" (aprile 2021) Formattazione note a margine in classicthesis (aprile 2020)
TeX StackExchange Custom margin notes (gennaio 2022) Why does not cross-referencing of end nodes work with 'enotez' and 'cleveref'? (agosto 2018) How to use two kinds of notes separated at the bottom and in the end of text (ottobre 2015)

COLLEGAMENTI CON ALTRE SCHEDE	
Scheda 40	(Riferimenti incrociati)
Scheda 46	(Testi filologici)

38	ELEMENTI DEL TESTO TABELLE	Rev.24/3/2024 1/2
----	-------------------------------	----------------------

Le tabelle sono particolarmente frequenti nei testi tecnico-scientifici, ma si trovano in abbondanza anche nei testi didattici del settore umanistico. Esse consentono di rappresentare sinteticamente ed efficacemente ciò che sarebbe troppo complicato descrivere a parole.
I pacchetti che estendono le funzioni di $\LaTeX$ standard si propongono come obiettivi principali: avere maggiore flessibilità e varietà nella formattazione della tabella nel suo complesso e delle singole righe e colonne; rendere possibile la composizione di tabelle di dimensioni eccezionali, sia in lunghezza, sia in larghezza.

MANUALI GENERALI	
LPE	§§8.1-8.3, p.145-168; §8.5, p.172-176
URM	§§8.21-8.23, p.79-87; §11.5, p.106-108
GGU	capitolo 8, p.195-216; §29.10.1, p.747-753; §29.11, p.754-758
WIK	capitolo 14, p.151-185; §§18.1-18.3, p.231-238

MANUALI PARTICOLARI	
C. Beccari	<i>Comporre tabelle e matrici</i> , Guida tematica GuIT, v.1.1.02, 22/9/2021

ARTICOLI	
L. F. Mori	Tabelle su $\LaTeX 2_{\epsilon}$, <i>ArsTeXnica</i> (2), ottobre 2006, p.31-47

38	ELEMENTI DEL TESTO TABELLE	Rev.24/3/2024 2/2
PACCHETTI		
array	(v.2.5g, 16/10/2023) An extended implementation of the LaTeX array and tabular environments. The special merits of this implementation are further options to format columns and the fact that fragile LaTeX commands don't have to be protected any more within those environments.	
booktabs	(v.1.618..., 15/1/2020) Enhances the quality of tables in LaTeX, providing extra commands as well as behind-the-scenes optimisation. Guidelines are given as to what constitutes a good table in this context. From version 1.61, the package offers longtable compatibility.	
longtable	(v.4.19, 1/11/2023) Defines a new environment, longtable, which has most of the features of the tabular environment, but produces tables which may be broken by TeX's standard page-breaking algorithm.	
tabularray	(v.2024A, 16/2/2024) Typesets tabulars and arrays with LaTeX3. Directly uses LaTeX3 functions to parse the table, and then typeset the entire table. Under the premise of being compatible with the basic syntax of LaTeX2 tables, separates the content and style of the table, and the style of the table can be completely set in keyval way.	
tabularx	(v.2.11c, 8/7/2023) A new environment, tabularx, is defined, which takes the same arguments as tabular*, but modifies the widths of certain columns, rather than the inter column space, to set a table with the requested total width.	
widetable	(v.2.1, 16/11/2020) Defines a new environment that, unlike tabularx, typesets a table of specified width by working on the inter-column glue; the tabular cells will all be stretched (or shrunk) according to need. The e-TeX arithmetic extensions are used, if available.	
SITI WEB		
Forum $\LaTeX$ Consiglio di template per un formulario (settembre 2023) Calcolo dei sequenti in LaTeX (aprile 2023)		
TeX StackExchange How to generate a complicated table (febbraio 2024) How to use multirow and multicolumn in formatting a comparison table (novembre 2023) My table doesn't fit; what are my options? (ottobre 2016)		
COLLEGAMENTI CON ALTRE SCHEDE		
Scheda 20	(Indice delle figure, delle tabelle e altri indici)	
Scheda 28	(Oggetti mobili)	
Scheda 33	(Espressioni matematiche)	
Scheda 40	(Riferimenti incrociati)	

6. Conclusioni

Ovviamente le conclusioni che si espongono in questo articolo sono provvisorie: poco più di un bilancio dei pro e dei contro riscontrati in seguito alle esperienze fatte finora.

Si riprendono e si riassumono, anzi tutto, alcune considerazioni svolte nei paragrafi precedenti.

- La suddivisione degli argomenti in sezioni e l'indice degli argomenti sono abbastanza assestati.
- La selezione delle fonti più adeguate per gli obiettivi della guida presenta ancora difficoltà non trascurabili.
- Il formato delle schede dev'essere completato e valutato su altri esempi.
- Le sezioni di cui si compone ciascuna scheda sono abbastanza assestate.
- Il contenuto delle varie sezioni dev'essere studiato più accuratamente, in particolare i commenti, di natura diversa da una sezione all'altra.

Oltre alle difficoltà specifiche per le varie componenti, sono emerse alcune difficoltà connesse con il progetto nel suo complesso.

Il progetto non si giustificerebbe se coprisse soltanto una parte minima della documentazione esistente. Al tempo stesso, non dev'essere troppo ambizioso: non avrebbe senso mettersi in competizione con le centinaia di libri, di manuali, di articoli, di pagine web esistenti.

La peculiarità di questa guida, anzi la ragione della sua creazione, è proprio l'estrema sintesi, il fatto di non pretendere di (ri)spiegare gli argomenti. Pertanto si deve trovare il giusto compromesso, affinché la guida sia snella, ma non superficiale, e diventi uno strumento di lavoro agile ed efficace per il pubblico interessato alla sua fruizione.

Lo sviluppo del progetto deve avvenire in una forma che sia il più possibile collaborativa, almeno nella fase di revisione, se non già nella fase di redazione. Le schede sono inevitabilmente soggette ad aggiunte, cancellazioni, correzioni; operazioni che una persona sola non è in grado di portare a termine su vasta scala senza il confronto e l'apporto di altre persone. Anche il formato grafico dev'essere valutato da più persone. Tuttavia, com'è ben noto, un lavoro collettivo richiede un coordinatore molto abile ed energico.

In definitiva, il progetto è ancora nello stadio sperimentale, ed è quindi aperto a critiche e suggerimenti, auspicabilmente stimolati anche dalla lettura di questo articolo.

Maurizio Molinaro
TORINO
maurizio.molinaro@gmail.com

Macros in language $\LaTeX$ 3

Claudio Beccari

Abstract Depending on the document contents, users might need some macros and/or environments suited for their typesetting. Several $\TeX$ modules are available, but sometimes users are more comfortable with macros they created directly. Here we show how to do when the functionalities of the $\LaTeX$ 3 language are required.

Sommario A seconda del contenuto del documento che si sta componendo, potrebbe essere necessario disporre di macro e/o di ambienti adatti alla composizione del testo. Sono disponibili molti pacchetti di estensione, ma talvolta gli autori si trovano più a loro agio con macro scritte da loro stessi. Qui si mostra come fare se sono necessarie le funzionalità del linguaggio $\LaTeX$ 3.

1. Introduction

We needed special functionalities to create macros and/or environments suited to do certain types of composition that were impossible or more complicated to do with $\LaTeX$ 2 ϵ .

The powerful functionalities of language $\LaTeX$ 3 (L3 in the sequel) were supposed to be sufficient to solve our problems. The documentation for such language is contained in the `interface3.pdf` file already distributed with any $\TeX$ system installation. Any user can read it by using the terminal command `texdoc interface3`.

The problem is that this is a very large document, totalling 359 pages, written in a very essential way, so that the “normal” $\LaTeX$ user is afraid to open the document and read its contents. Some experience is required in order to scan the table of contents, that is very detailed, but of course the part and section titles are very short and some ingenuity is required to find what supposedly is necessary.

Moreover the “normal” $\LaTeX$ user is used to write macros that do not contain special characters, therefore it is necessary to create an L3 to $\LaTeX$ conversion, so that users can maintain their habits in writing the source files.

What follows is suited to human science applications as well as to technical ones.

2. Some macros already available for $\LaTeX$ 2 ϵ

A couple of packages that help in using the new L3 functionalities are already available. They are `xparse` and `xfp`. The functionalities of both packages are contained almost completely into any $\LaTeX$ based format, be it for `pdflatex`, `lualatex`, or `xelatex`. This implies that no package is required to use them, although it might be necessary to load them if unusual functionalities are required. For example the `xparse` package allows to define *optional* arguments delimited with *braces*, `{...}`, but this functionality was considered a deprecated one and was not transferred to the $\LaTeX$ kernel; according to our personal opinion, it is not to be deprecated and we

used it. Few other features were not transferred to the $\text{\LaTeX}$ kernel because they were still experimental and possibly unstable.

2.1. The `xparse` package

Among the `xparse` package functionalities there are the definitions of robust commands in order to create macros/commands and environment; such `xparse` command names have prefixes such as `New`, `Renew`, `Provide`, `Declare` that apply to both defined commands and defined environments, plus the strings `NewExpandable`, `RenewExpandable`, `ProvideExpandable`, `DeclareExpandable` applicable only to commands; they can prefix the name roots `DocumentCommand` and `DocumentEnvironment`. For example, here we use `\ProvideExpandableDocumentCommand`.

All the defined commands and environments have a first, possibly empty mandatory argument that contains the argument descriptors, and a second argument (commands) or two arguments (environments) that contain the actual definitions, similarly to the $\text{\LaTeX 2}_\epsilon$ analogous definition macros.

Of course, the four prefixes applied to command definitions exist with different names also in $\text{\LaTeX 2}_\epsilon$, while the latter does not use prefixes similar to the last two ones for environments. Furthermore environment arguments can be used also in the environment closing commands, while this feature is not available with $\text{\LaTeX 2}_\epsilon$ environments.

But the real progress consists in the *argument descriptors*; they define mandatory, optional, boolean, token arguments, the first two possibly delimited with arbitrary delimiters, and optional arguments may receive a preset default value. All this is really new compared to $\text{\LaTeX 2}_\epsilon$, that may receive just the number of arguments the first of which may be an optional one with or without a bracketed default value.

The reader, even if no package loading is generally necessary, may read the documentation by using the terminal command `texdoc xparse`; it contains a lot of useful information.

2.2. The `xfp` package

Again the `xfp` package functionalities are included in the $\text{\LaTeX}$ kernel, therefore no package loading is required.

The name `xfp` stands for “eXtended Floating Point”, but the commands are not limited to decimal floating point arithmetics: the defined commands are `\fpeval`, `\inteval`, `\dimeval`, `\skipeval`¹. Unfortunately the `xfp` documentation is essential, and the integer calculations are just listed without any further comment.

In practice `\inteval` can use integers expressions that contain only the usual sums, subtractions, multiplications and divisions even grouped with round parentheses to express precedences; the division quotient is rounded to the nearest integer n ; for “nearest” it is meant that if the floating point quotient q were $n - 0.5 \leq q < n + 0.5$, the quotient is rounded to n ; $\text{\LaTeX 2}_\epsilon$ division, on the contrary, rounds q to n when $n - 0.5 < q \leq n + 0.5$.

The `\fpeval` can execute expressions containing round parentheses for precedence, round parentheses to contain functional operator arguments, the usual arithmetic operators, the square root, the exponentiation and logarithms, some special single or multiple operand func-

1. The last two ones are not documented in `xfp.pdf`, but have been announced in the $\text{\LaTeX}$ News letter, number 35, page 3, that may be read with the terminal command `texdoc ltnews35`.

tional operators, trigonometric functions working with both radians and degrees, hyperbolic functions, and some other less frequent operations. Normally the decimal computations are done with 15 digit decimal fractional parts, but results can be rounded in different ways and with a specified number of digits; on the opposite it is possible to specify to work with a larger number of digits. If dimensions or skips values are inserted among the operands, the used value is their decimal representation in typographical points ($72.27 \text{ pt} = 1 \text{ in}$); the dimension units of measure can be used as constants: for example `\fpeval{in/mm}` yields 25.4. The `atan` and `acot` functions may receive one single value or two comma separated values; in the first case the computed arc(co)tangent value is within the interval $(-\pi/2, \pi/2]$; in the second case the value lays in the interval $(-\pi, \pi]$. In this way it is possible to correctly compute the anomaly of a complex number in the four quadrants. With complex numbers $x + iy$ the pair of values to enter into the argument of `\atand2` function is y, x ; for example:

$x + iy$	$1 + i0$	$1 + i1$	$0 + i1$	$-1 + i1$	$-1, +i0$	$-1, -i1$	$0, -i1$	$1, -i1$
<code>\atand(y, x)</code>	0°	45°	90°	135°	180°	-135°	-90°	-45°
<code>\acotd(x, y)</code>	0°	45°	90°	135°	180°	-135°	-90°	-45°

Here we added the degree sign just to stress the point, but the `atand` function yields just the numerical value. If users dislike to invert the components of a complex number in the argument of the arctangent, they can use the arccotangent without any inversion, as it is shown in the third line of the above table

3. L3 functions

Macros in L3 are called “functions”; actually functions are more versatile than macros and their syntax may be more similar to mathematical functions, than to standard $\text{\LaTeX}$ 2 _{ϵ} macros.

In the previous section dedicated to numerical computations we have already seen the `xfp` macros at work with small expressions to work on.

Without using the `\dimeval` macro we can compute the outer grid rounded width in millimetres:

```
\fpeval{round{(\textwidth+\marginparwidth+\marginparsep)*pt/mm,2}}\,mm
```

that yields 148.85 mm. If we used the `\dimeval` command, we could not directly compute the width in millimetres.

Here we would like to show some new macros the definitions of which use directly some L3 functions.

The myriad L3 functions are listed and described in the `interface3` documentation. Here we use a few of them in order to define some $\text{\LaTeX}$ macros that we use very frequently.

They are to be taken just as examples to create in a similar way other macros for different uses.

3.1. A robust testing macro

$\text{\TeX}$ tests are fragile; several tricks were developed in the past to render them robust. But if a boolean expression is to be tested these tricks require powerful packages; among such packages

2. Trigonometric function names ending with the letter ‘d’ work with degrees.

we can cite `ifthen`, the documentation of which may be read by using `texdoc ifthen`, `xifthen` (documentation with `texdoc xifthen`), `etoolbox` (documentation with `texdoc etoolbox`); the last one has a very wide $\TeX$ powerful service macros collection that should always be taken into account when writing $\TeX$ code.

Nevertheless we needed a powerful macro that could accept a simple logical operand notation and we found it among the myriad functions of `interface3`. Here is the definition³

```
\ExplSyntaxOn%
\AtBeginDocument{%
\ProvideExpandableDocumentCommand\fpctest{m m m}{%
  \fp_compare:nTF{#1}{#2}{#3}}
}
\ExplSyntaxOff
```

the syntax of which turns out to be

```
\fpctest{<boolean expression>}{<true code>}{<false code>}
```

The *<boolean expression>* is the magic of this macro: any operand may be any true or false expression, and the boolean operators are `&&` (AND), `||` (OR) and `!` (NOT). Operators involving math quantity relations, such as `=`, `>`, `<`, may be used as single ones, or may be paired, or may be associated all together: `<=>` means the math terms to be compared are “comparable”; yes, because `\fpeval` can use also “not comparable” entities named `NaN`, meaning “not a number”, that can be obtained during computations in place of, say, ∞ , when a division by zero takes place; a `NaN` cannot be compared with anything, not even with another `NaN`. Furthermore groups of mathematical relation operators may be negated; therefore we have `<`, `=`, `>`, `<=`, `>=`, `<>`, `<=>`, `!<`, `!>`, `!=`, ... It is evident that a comparison such as `>=` is equivalent to `!<`.

Here is a small example:

```
\ProvideDocumentCommand\goodpagebreak{0{4}}{%
\fpctest{\pagegoal-\pagetotal!>#1\baselineskip}{\newpage}{\relax}}
```

As it can be seen the above command `\goodpagebreak` inserts a `\newpage` command if the remaining space on the page is less than four times the normal text line height. This produces a slightly short page but may avoid widows and orphans, even before a section title.

The number of lines may be optionally changed by the user who may enter a code such as `\goodpagebreak[6]` to allow a page break if the space between paragraphs is enlarged too much because the page does not have enough space to receive, for example, a system of equations. The argument descriptor `0{4}` describes a bracket enclosed optional argument the default value of which is preset to 4. The `Provide` prefix avoids to define anything if the class or the previously loaded packages already defined that command.

Another definition that may be useful is

```
\ExplSyntaxOn
\ProvideDocumentCommand\IfVertical{m m}{%
  \mode_if_vertical:TF{#1}{#2}}
\ExplSyntaxOff
```

3. Here and in the sequel we enclose each definition within the delimiters `\ExplSyntaxOn` and `\ExplSyntaxOff` but users can wrap a whole set of definitions within a single pair of such delimiters in their preamble.

the syntax of which is

```
\IfVertical{<true code>}{<false code>}
```

Therefore the `\goodpagebreak` may be used within another command:

```
\ProvideDocumentCommand\goodbreakwithinaparagraph{0{4}}{%
  \IfVertical{\goodpagebreak[#1]}{\vadjust{\newpage}}}%
}
```

In this way command `\goodbreakwithinaparagraph`, inserted between two paragraphs, inserts a page break if the remaining space on the page is shorter than that prescribed number of text lines (four by default), but it can be set within a paragraph so that at the end of the line where the command is inserted, it produces a page break.

In two column mode command `\newpage` shifts to the next column, therefore if the `\good...break...` commands are given while in the first column, this column is broken and typesetting continues in the second column; on the opposite, if the commands are issued in the second column typesetting continues on the following page.

3.2. Other testing macros

We defined and used also the following macros:

```
\ExplSyntaxOn
\ProvideExpandableDocumentCommand\IfOdd{m m m}%
  {\int_if_odd:nTF{#1}{#2}{#3}}
\ProvideExpandableDocumentCommand\IfEqual{m m m m}%
  {\str_if_eq:eeTF{#1}{#2}{#3}{#4}}
\ExplSyntaxOff
```

Again you see inside the definition the internal L3 functions `int+if_odd:nTF` and `\str_if_eq:eeTF` found in `interface3`.

The syntax of the first macro is the following

```
\IfOdd{<integer number>}{<true code>}{<false code>}
```

The robust `\IfOdd` macro replaces the native test construct `\ifodd...\else...\fi` that is fragile.

The second test macro `\IfEqual` compares two strings and is true if and only if the strings are absolutely identical; they may contain even alphabetic glyphs. This macro can test the identity of two explicit strings, but nobody would test two explicit identical strings; users would test couples of strings of which at least one is the replacement of an expanded macro. Its syntax is the following:

```
\IfEqual{<string1>}{<string2>}{<true code>}{<false code>}
```

Notice that `<string1>` and `<string2>` may be either explicit strings, or macros containing strings; as already noticed, it is wasted time to compare two identical explicit strings.

Therefore if the following macros are defined as such:

```
\def\A{abc} \def\B{abc } \def\C{abc} \def\D{xyz}
```

tested by using `\IfEqual` with the explicit string “abc” would yield:

```
String \A is equal to “abc”.
String \B is not equal to “abc”.
String \C is equal to \A.
String \D is not equal to “abc”.
```

Implicit string comparisons are possible with native commands through the `\ifx` test that compares the definitions of two macros but this test is fragile; it can be done in a robust way with other packages, especially with `etoolbox`. Also some internal $\TeX$ 2 ϵ test macros are robust, but their names are protected with the `@` character and are not available to the end user.

3.3. Cyclic repetition of code

$\TeX$ 2 ϵ has a native command with syntax

```
\while{<test>}do{<code>}
```

That can repeat the `<code>` while `<test>` is true; generally `<test>` contains a simple relationship between the contents of two numeric registers, or a counter register versus an integer constant, and the numeric counter is modified at each cycle within the `<code>`. The construct is fragile. In any case if the counter is not initially assigned a value before executing the above cyclic macro, and/or the counter is not modified within the `<code>`, the risk is an infinite loop that might be difficult to abort or that causes the saturation of the compiler engine working memory.

Using L3 functions the above risks remain, and user should pay attention to the correct handling of the counter.

But with L3 functions it is possible to avoid counters, and/or to use tests that are more flexible than those of the native commands.

We defined two macros that control the cyclic repetition of code:

```
\ExplSyntaxOn
\ProvideExpandableDocumentCommand\fpdowhile{m m}{%
  \fp_do_while:nn{#1}{#2}}
\ProvideExpandableDocumentCommand\fpwhiledo{m m}{%
  \fp_while_do:nn{#1}{#2}}
\ExplSyntaxOff
```

Their syntax is the following:

```
\fpdowhile{<test>}{<code>}
\fpwhiledo{<test>}{<code>}
```

They appear to be very similar, but they are not; during each cycle `\fpwhiledo` first checks the `<test>` then executes the `<code>`, while `\fpdowhile` first executes the `<code>` then checks the `<test>`.

Before entering the cyclic operations the `<test>` must be ‘true’; while repeating the cycles some element involved in the `<test>` must eventually change so that such `<test>` becomes ‘false’. But the test may be any boolean expression or any testing macro as those described in the previous subsection.

See the next section for an example.

3.4. Modular arithmetics

$\text{\LaTeX}$ 2_ε defines two macros to describe two modular operations: `\bmod` and `\pmod`. They can be used to typeset mathematics, but they do not execute any computation; it is possible to typeset $m \bmod n$ and $m \pmod n$.

But the integer remainder of the integer division m/n has to be computed in some way. The solution is:

```
\ExplSyntaxOn
\ProvideExpandableDocumentCommand\Mod{m m}%
  {\interval{\int_mod:nn{#1}{#2}}}
\ExplSyntaxOff
```

where the first argument is the dividend m and the second argument is the divisor n that in this case defines the “modulo”.

With this macro it is possible to write that 140 hours equals 5 days plus 20 hours:

$$140 \text{ h} = 6 \text{ d plus } 20 \text{ h}$$

produced with

```
\[140\text{\,h} = \interval{140/24}\text{\,d plus }\Mod{140}{24}\text{\,h}\]
```

A simple example might be the following: it is desired to append zero or more pages to the end of a document so that the total number becomes an integer multiple of an n -folio signature.

The code would be the following:

```
\clearpage\pagestyle{empty}
\addtocounter{page}{0}% correction for initial pages
\fpdowhile{\Mod{\value{page}}{n}!=1}{\null\newpage}
```

The `\clearpage` command ensures that the extra pages are effectively added after all the book material has already been sent to the output file. The `\pagestyle{empty}` ensures that the possibly added pages are completely empty, even without the folio. In this example the page number has a certain value before starting the cycling code; the `\newpage` command outputs a new page and the page counter is stepped up. We are therefore sure that no infinite loop can take place, even if no explicit counter command is seen in the code to be executed. The most puzzling command that appears in the above example is the fact that the cycling process stops when the page number to be tested equals 1 modulo n ; this is due to the fact that the page counter points to the page that has not yet been output, therefore in order to stop when the last output page folio was 0 modulo n , it is necessary to stop the cycling process when the page folio is one unit higher.

In any case, if the main matter is preceded by pages numbered in a different way, for example with roman numerals, and numbering restarts from one at the beginning of the main matter, a correction is necessary because the extra blank pages number must be computed considering the total number of pages. The correction implies a single statement such as `\addtocounter{page}{\langle page number correction \rangle}`. In the above case the value 0 is specified for

no correction; in this case the statement may be deleted or commented out; otherwise that zero should be replaced with the initial pages total number.

Users are invited to obtain the same results by using the native $\text{\LaTeX 2}_{\epsilon}$ commands and they will appreciate the simplicity of the above code.

4. Conclusion

We could go on with other examples; but we stop here because we have shown the way to tackle such unusual problems, that are very simple to solve with L3 functions. $\text{\LaTeX 3}$ code is not simple at all; reading `interface3.pdf` is difficult, but after a while one gets used to certain terms and understanding the documentation becomes easier.

Claudio Beccari
RIVOLI, TO, ITALY
`claudio(dot)beccari(at)gmail(dot)com`

Alla scoperta di LuaCAS

Roberto Giacomelli

Sommario In questo articolo esploreremo il pacchetto `luacas`, una libreria scritta in Lua per il calcolo simbolico, dal punto di vista dell'utente con l'obiettivo principale di valutarne l'espressività del codice.

Tra le applicazioni d'esempio il calcolo della radice quadrata secondo il metodo noto agli antichi babilonesi senza approssimazione delle frazioni risultato di ogni iterazione.

Abstract In this article we will explore the `luacas` package — a Lua Computer Algebra System library — from the user point of view. The main goal is the assessment of code expressiveness.

The calculation of the square root using closed form rational numbers with the ancient Babylonian Method is presented.

1. Motivazione

Il pacchetto `luacas` è stato recentemente aggiunto agli archivi CTAN ed è quindi disponibile per tutti gli utenti. Esso racchiude alcune funzioni per il calcolo simbolico — CAS sta per *Computer Algebra System* — ed è scritto in Lua perciò può essere eseguito solamente dai motori di composizione più recenti.

Poter effettuare calcoli simbolici permette all'utente di predisporre documenti per la matematica con un grado di automazione senza precedenti. Per esempio, potremo far risolvere equazioni di secondo grado in forma chiusa, cioè con radici simboliche, per comporre un eserciziario con meno fatica e privo di errori.

Un software CAS è normalmente in grado di valutare interi di grandezza arbitraria, generare il numero voluto di cifre decimali di numeri irrazionali, o completare il calcolo di funzioni derivate o integrali. Tra i software CAS open source possiamo citare il pacchetto SymPy (SYMPY DEVELOPMENT TEAM, 2023), scritto interamente in Python.

Dato l'interesse applicativo di un pacchetto nativo per Lua $\TeX$ ho pensato di presentare `luacas` ai lettori di *4 $\TeX$ nica* valutando attentamente più che le funzionalità simboliche l'esperienza di scrittura del codice.

Il linguaggio informatico esprime entità e concetti in forma testuale. Esso determina perciò quanto per l'utente sia naturale o intuitivo costruire il codice, impararne la logica in poco tempo, evitare errori e programmare facilmente estensioni in caso se ne presenti la necessità.

2. Requisiti e avvertenze

Per la comprensione del codice degli esempi è indispensabile conoscere le basi del linguaggio Lua. Rimando perciò al libro di Roberto Ierusalimschy (IERUSALIMSKY, 2016) principale autore di Lua, e alla guida tematica (GIACOMELLI, 2021) predisposta per l'utente $\TeX$ per facilitargli l'accesso alla programmazione Lua.

Sempre importante da consultare è la documentazione di Lua $\TeX$ (LUA $\TeX$ DEVELOPMENT TEAM, 2023) dove le funzioni interne che utilizzeremo nei listati sono presentate nello stile di un manuale di riferimento, come per esempio la funzione `tex.print()`.

Il codice Lua risulta spesso semplice da leggere tuttavia, come per tutti i linguaggi di programmazione, ci sono dettagli che non possono essere semplicemente ipotizzati.

Quanto al pacchetto stesso, la documentazione di `luacas` si trova su CTAN oppure sul sito `texdoc.net` digitandone il nome nella barra di ricerca. Se il pacchetto è installato sul vostro computer, sarà possibile visualizzarla a schermo con l'utility da terminale `texdoc`.

Infine segnalo che con le future versioni del pacchetto `luacas` potrebbe essere necessario aggiornare i listati illustrati in questo articolo scritti per la versione v1.0.2 del maggio 2023.

3. Primi sorgenti

In questa sezione scriveremo due sorgenti introduttivi all'uso del pacchetto `luacas`, uno per il compositore `lua $\LaTeX$` e uno per l'interprete Lua. Scopo dei sorgenti sarà quello di risolvere l'equazione $x^2 - 4x - 2 = 0$ con le sue due radici irrazionali.

3.1. Utilizzo con Lua $\TeX$

Con Lua $\TeX$ è sufficiente caricare il pacchetto `luacas` nel preambolo del documento e scrivere il codice Lua nell'ambiente `CAS`. Per l'obiettivo fissato, apriamone uno per definire il simbolo x , l'espressione del trinomio e infine per ottenere i valori delle radici:

```

1 % !TeX program = LuaLaTeX
2 \documentclass{article}
3 \usepackage{luacas}
4 \begin{document}
5 \begin{CAS}
6   vars("x")
7   f = x^2 -4*x -2
8   r = roots(f)
9 \end{CAS}
10
11 Le radici dell'equazione \(\ \print{f} = 0 \) sono
12 \(\ \left\{ \print{r[1]}, \quad \print{r[2]} \right\} \) \).
13 \end{document}
```

Compilando questo sorgente minimo otterremo:

Le radici dell'equazione $x^2 - 4x - 2 = 0$ sono $\{2 + \sqrt{6}, \quad 2 - \sqrt{6}\}$.

Nell'ambiente `CAS` va inserito solamente codice Lua facilitato sia perché i nomi delle funzioni sono tutti globali e perciò più semplici da digitare e leggere, sia perché alcuni dati come gli interi vengono trasformati automaticamente negli oggetti simbolici corrispondenti *prima* che il codice Lua venga effettivamente eseguito.

Queste scelte hanno pro e contro. Osservando il brevissimo codice del listato precedente, notiamo che la funzione `vars()` accetta una o più stringhe per convertirle in oggetti riferiti da

variabili globali con lo stesso nome (riga 6). L'intento è semplificare il linguaggio ma comporta le conseguenze dovute alla modifica diretta della memoria globale che esamineremo.

3.2. *Locale o globale?*

Normalmente la funzione `vars()` avrebbe restituito il riferimento all'oggetto corrispondente al simbolo matematico consentendo all'utente di scegliere non solo un nome appropriato per la variabile anche differente dal simbolo stesso, ma soprattutto consentendogli di scegliere se costruire una variabile locale oppure una globale.

Una variabile *locale* viene distrutta al termine del blocco di codice in cui è stata creata, mentre una *globale* rimane disponibile anche per i blocchi successivi.

Lo spazio globale è condiviso da *tutti* gli oggetti Lua creati durante la compilazione dell'intero documento. Si considera un buon stile di programmazione il dichiarare come locali gli oggetti temporanei il cui *significato* è logicamente attinente a una singola procedura.

Non si tratta solo del vantaggio che mette al riparo da errori di programmazione o dal rischio di collisione dei nomi nel namespace globale, ma di rendere il linguaggio maggiormente espressivo nella semantica degli oggetti.

Durante la compilazione di un documento magari molto complesso con decine di pacchetti caricati, il rischio di collisione dei nomi è un problema anche per le macro tradizionali. Per limitarlo è buona regola usare il carattere `@` nei nomi per far controllare al compositore stesso l'accesso alle macro private. Ciò è vero anche per il nucleo di $\text{\LaTeX}$.

In Lua si preferisce creare una tabella contenitore globale per memorizzarne al suo interno un numero arbitrariamente grande di oggetti. Per aderire a questa prassi modifichiamo il listato precedente creando la tabella contenitore chiamata `dati`:

```

1 % !TeX program = LuaLaTeX
2 \documentclass{article}
3 \usepackage{luacas}
4 \begin{document}
5 \begin{CAS}
6     dati = {}
7     vars("x")
8     dati.eq = x^2 -4*x -2
9     dati.roots = roots(f)
10 \end{CAS}
11 Le radici dell'equazione \(\ \print{dati.eq} = 0 \) sono \(\ \left\{
12 \print{dati.roots[1]}, \quad \print{dati.roots[2]} \right\} \).
13 \end{document}
```

Per evitare che anche l'oggetto del simbolo `x` sia globale anziché usare la funzione `vars()` occorre usare il costruttore della classe `SymbolExpression` come nei prossimi listati.

Alla sezione 4 ci occuperemo invece di una delle particolarità più importanti del pacchetto `luacas`, almeno per gli scopi di questo lavoro, ovvero quella di poter scrivere espressioni di fatto sommando, sottraendo o moltiplicando oggetti come se fossero gli effettivi simboli matematici come alla riga 8 del precedente listato.

3.3. Utilizzo con texlua

La libreria `luacas` si può utilizzare anche all'esterno di un sorgente `Lua4TeX` nella modalità *scripting*. Non tutti sono a conoscenza che all'interno delle moderne distribuzioni `TeX` è presente anche un interprete Lua. Il suo nome deriva dall'inversione delle sillabe di quello del motore di composizione `luatex` ed è `texlua`.

Con l'interprete `texlua` è possibile eseguire uno script memorizzato in un file di testo solitamente di estensione `.lua` digitandone il nome dopo quello dell'interprete stesso alla riga di comando.

Per caricare la libreria `luacas` è sufficiente impostare il ramo `lua4latex` della distribuzione `TeX` in cui si trovano i file sorgenti attraverso l'utilità `kpse` solamente presente in `texlua`, ed eseguire la funzione `require()` con il nome del file di inizializzazione previsto dagli sviluppatori del pacchetto.

Ottenuto il riferimento principale alla libreria `luacas` possono essere caricati a sua volta i moduli ritenuti necessari, nel nostro caso `core` e `algebra`:

```
1 kpse.set_program_name("lua4latex")    -- path to the luacas files
2 local luacas = require("luacas_init") -- load the library
3 luacas:initglobalmodule("core")       -- then the main module
4 luacas:initglobalmodule("algebra")    -- and a useful second module
```

Per la stampa in console delle radici del trinomio si crea l'espressione corrispondente e su di essa si esegue il metodo `roots()`. Il codice Lua è più dettagliato di quello richiesto per l'ambiente `CAS`:

```
5 local x = SymbolExpression("x")
6 local f = x^Integer(2) -Integer(4)*x -Integer(2)
7 local ok
8 f, ok = f
9     :autosimplify()
10    :topolynomial()
11 assert(ok)
12 local re = f:roots()
13 print(re[1]:tolatex(), re[2]:tolatex())
```

Il risultato dell'esecuzione è quello atteso: la stampa in console di

$2 + \sqrt{6}$ $2 - \sqrt{6}$

I nomi `SymbolExpression` e `Integer` corrispondono a classi `luacas` nel namespace globale. Devono essere inseriti poiché, a differenza dell'ambiente `CAS`, questa volta non vi è alcuna elaborazione del codice prima dell'esecuzione. Anzi possiamo dire che sostanzialmente il lavoro effettuato dall'ambiente è proprio quello di inserire quei nomi al posto dell'utente.

4. La struttura interna di luacas

Ciascun modulo `luacas` immette nel namespace globale vari oggetti utili ai calcoli simbolici, come `SymbolExpression` che crea nuovi simboli letterali. Ognuno di questi oggetti è program-

mato secondo il paradigma a oggetti sulla base di una gerarchia che li relaziona uno con l'altro allo stesso modo delle relazioni tra i vari concetti matematici che rappresentano.

Per esempio una funzione esponenziale farà parte della famiglia delle funzioni, mentre una numero razionale farà parte del gruppo dei numeri.

Nel codice Lua per sommare o moltiplicare questi oggetti utilizzando i normali simboli aritmetici occorre implementarne i *metamethodi* in una tabella detta *metatabella*. Si tratta di normali tipi funzione e tabella che l'interprete ricerca ed esegue se nel codice gli oggetti condividono una stessa metatabella e fanno parte di espressioni.

Si tratta di una tecnica molto semplice per creare delle *aritmetiche* su gruppi di oggetti per cui ha senso definire delle operazioni. Per esempio se l'oggetto rappresentasse un contenitore di dati, alla somma potrebbe essere dato il significato di unione tra insiemi.

In luacas una tale tecnologia è perfetta per rendere il codice compatto e intuitivo a beneficio degli utenti tanto che converrà mostrare un esempio per valutare l'effettiva semplicità e naturalezza: la somma e il prodotto di due polinomi:

```

1 \begin{CAS}
2   a = PolynomialRing({1, 2, 0, 3}, "t")
3   b = PolynomialRing({3, -2}, "t")
4   % sommo i due polinomi per ottenerne un terzo:
5   p_sum = a + b
6   % moltiplico i due polinomi
7   p_mul = a * b
8 \end{CAS}
9 % stampo tutto in ambiente matematico
10 \[ (\print{a}) + (\print{b}) = \print{p_sum} \]
11 \[ (\print{a})(\print{b}) = \print{p_mul} \]
```

L'esecuzione di questo codice in questo stesso documento produce:

$$(3t^3 + 2t + 1) + (-2t + 3) = 3t^3 + 4$$

$$(3t^3 + 2t + 1)(-2t + 3) = -6t^4 + 9t^3 - 4t^2 + 4t + 3$$

Da notare che `\print`, una macro definita dal pacchetto `luacas`, non fa altro che chiamare il metodo `tolatex()` sull'argomento e inviare il risultato allo stream del compositore.

5. Il metodo babilonese

Si rimane ogni volta increduli davanti ai progressi matematici delle civiltà antiche, di come fossero sofisticate le strutture geometriche dei greci o di quanto avanzate fossero le conoscenze dei numeri nella millenaria cultura indiana.

E anche il metodo babilonese per il calcolo della radice quadrata sorprende poiché consiste in un processo iterativo, ovvero un'idea niente affatto banale. Il punto di partenza è geometrico: il quadrato di area α ha lati di lunghezza $\sqrt{\alpha}$, perciò la ricerca della radice equivale alla costruzione del quadrato di area nota.

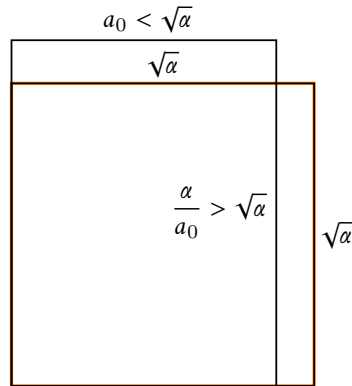


Figura 1. L'idea geometrica che sta alla base del metodo babilonese sta nell'approssimare il quadrato di area pari al radicando con un rettangolo di uguale area. I lati del rettangolo sono uno più grande e uno più piccolo rispetto al lato del quadrato che rappresenta la radice cercata. Da qui l'idea di una procedura iterativa che genera nuovi rettangoli dal precedente sempre più simili al quadrato.

Consideriamo a questo scopo un rettangolo di area α . L'osservazione importante è che i lati di questo rettangolo sono uno più corto e uno più lungo del lato del quadrato. Per dimostrarlo è sufficiente osservare la figura 1.

Se un lato del rettangolo vale a_0 di conseguenza l'altro lato dovrà valere α/a_0 . E se questi due valori sono uno più piccolo e uno più grande di $\sqrt{\alpha}$ una sua migliore approssimazione è la *media* dei due lati:

$$a_1 = \frac{a_0 + \alpha/a_0}{2}$$

Assumendo la media a_1 come lato di un secondo rettangolo sempre di area α , saremo ancora nella stessa situazione iniziale. Ripetere la costruzione ci darà una sequenza di rettangoli sempre più vicini al quadrato.

L'iterazione non termina. In nessuno dei passi della sequenza otterremo il quadrato esatto¹ ma ogni volta ci avvineremo ad esso sempre più con un processo asintotico. Un'idea affascinante nella sua semplicità e nell'impossibilità di raggiungere il risultato esatto.

Il metodo babilonese consiste dunque nel fissare un primo valore per la radice cercata e nel calcolare le successive approssimazioni fino a ottenere la precisione desiderata. Il lato del rettangolo successivo si ottiene dal precedente con la formula iterativa:

$$a_{n+1} = \frac{1}{2} \left(a_n + \frac{\alpha}{a_n} \right)$$

Per lo studio delle proprietà del metodo e per gli approfondimenti, rimando il lettore a testi specifici di argomento matematico. Un primo punto di partenza è la pagina di Wikipedia (WIKIPEDIA, 2023).

1. Se il radicando è un quadrato perfetto e se il primo valore è la radice esatta allora *tutti* i rettangoli finiscono per essere la stessa figura ovvero lo stesso quadrato esatto.

n	a_n	$\frac{\alpha}{a_n}$	$a_{n+1} = \frac{1}{2} \left(a_n + \frac{\alpha}{a_n} \right)$	Valore decimale di a_{n+1}	Cifre esatte
0	1	2	$\frac{3}{2}$	1,5	1
1	$\frac{3}{2}$	$\frac{4}{3}$	$\frac{17}{12}$	1,416 666 666 666 7	3
2	$\frac{17}{12}$	$\frac{24}{17}$	$\frac{577}{408}$	1,414 215 686 274 5	6
3	$\frac{577}{408}$	$\frac{816}{577}$	$\frac{665857}{470832}$	1,414 213 562 374 7	12
4	$\frac{665857}{470832}$	$\frac{941664}{665857}$	$\frac{88673108897}{627013566048}$	1,414 213 562 373 1	24

Tabella 1. Le prime cinque iterazioni per il calcolo di $\sqrt{2}$ con il metodo babilonese conseguenza del valore iniziale $a_0 = 1$. Nell'ultima riga il numero di cifre esatte di 24 è corretto ma non corrisponde al numero decimale arrotondato riportato nella colonna del valore decimale della frazione che è limitato dal calcolo in virgola mobile in doppia precisione. Si noti come a ogni iterazione il numero di cifre esatte raddoppi.

5.1. Un esempio di applicazione del metodo

Se il radicando e il valore iniziale sono entrambe numeri razionali ne segue osservando la formula dell'iterazione, che la sequenza convergente sarà anch'essa composta da numeri razionali.

Se si fanno i conti a mano ci si accorge ben presto che numeratore e denominatore crescono in modo molto rapido come dimostra la tabella 1 che mostra la procedura di ricerca di $\sqrt{2}$.

Il pacchetto luacas ci viene in aiuto nel gestire tali frazioni. All'interno di un documento Lua $\LaTeX$ potremo definire una funzione molto semplice che calcola il termine successivo a_{n+1} a partire dalla frazione corrente a_n :

```

1 \begin{CAS}
2   % contenitore di libreria
3   sqrtlib = {}
4   function sqrtlib.next(alpha, a)
5     local a_next = (a + alpha/a)/2
6     return a_next
7   end
8 \end{CAS}

```

Ho riscontrato che l'ambiente CAS al momento non gradisce linee vuote al suo interno. Tuttavia esso semplifica effettivamente il codice dell'espressione non essendo necessario esprimere esplicitamente le costanti come oggetti Integer.

Nel successivo listato la funzione next() viene chiamata il numero di volte previsto all'interno di un secondo ambiente CAS che si trova a sua volta in modo matematico:

```

1  \[\left\{
2  \begin{CAS}
3      local alpha = 2
4      local a = 1
5      tex.print(a:tolatex())
6      for i in range(2, 7) do
7          a = sqrtlib.next(alpha, a)
8          tex.print('..'a:tolatex())
9      end
10 \end{CAS}
11 \right\}\]
```

Il risultato è la stampa delle prime sette iterazioni del metodo con $\alpha = 2$ e $a_0 = 1$:

$$\left\{ 1, \frac{3}{2}, \frac{17}{12}, \frac{577}{408}, \frac{665857}{470832}, \frac{886731088897}{627013566048}, \frac{1572584048032918633353217}{1111984844349868137938112} \right\}$$

Il metodo `tolatex()` restituisce la stringa di testo corrispondente al codice $\text{\TeX}$ della frazione che con la funzione `tex.print()`, a sua volta viene inviato al compositore. Si tratta di una modalità molto semplice e diretta di scambio dati da Lua verso il motore $\text{\TeX}$ che li elabora come se fossero stati letti dal file sorgente.

Le particolarità dell'ambiente **CAS** non consentono l'utilizzo del ciclo `for` tradizionale perché i valori numerici interi sono tutti convertiti in oggetti `Integer` prima che il codice sia effettivamente eseguito da Lua. Per questo è prevista la speciale funzione `range()`.

Tutto sommato non è difficile convertire il codice per l'esecuzione diretta in Lua senza la mediazione dell'ambiente **CAS**:

```

1  \[\left\{\directlua{
2      local alpha = Integer(2)
3      local a = Integer(1)
4      tex.print(a:tolatex())
5      for i = 2, 7 do
6          a = sqrtlib.next(alpha, a)
7          tex.print('..'a:tolatex())
8      end
9  }\right\}\]
```

Confrontando il codice, oltre all'uso della primitiva `\directlua` nativa del compositore $\text{\TeX}$, la differenza consiste solamente nel dover generare gli oggetti `Integer` esplicitamente e nel ricorso al ciclo iterativo con contatore intero.

Lascio al lettore convertire questo codice in un comando più generale. La macro potrebbe chiamarsi `\babilon` e accettare tre argomenti: in ordine il radicante α , il valore iniziale della sequenza a_0 e il numero di iterazioni da calcolare.

L'effetto del comando `\babilon{5}{2}{6}` è la stampa delle prime sei frazioni con valore iniziale 2 che il metodo babilonese genera per $\sqrt{5}$, ovvero:

$$\left\{ 2, \frac{9}{4}, \frac{161}{72}, \frac{51841}{23184}, \frac{5374978561}{2403763488}, \frac{57780789062419261441}{25840354427429161536} \right\}$$

Una nota finale, per comporre la tabella 1 ho utilizzato rispetto al codice precedente anche il metodo `asnumber()` della classe `Rational` che restituisce la frazione come valore decimale con la precisione del calcolo in virgola mobile.

6. Conclusioni

Il pacchetto `luacas` offre funzionalità di calcolo simbolico per $\text{Lua}\TeX$ già di completezza accettabile rivelandosi molto utile in special modo per produrre documenti dedicati alla matematica in ambito didattico.

Speriamo che gli Autori perfezionino ed estendano il codice sia nelle funzionalità sia nell'architettura generale e che per questo ricevano il contributo della comunità.

Sarebbe molto utile che lo sviluppo basato su Lua venisse condotto in modo coordinato. Un linguaggio utente ben studiato per coerenza e intuitività aiuta a migliorare il delicato e faticoso processo di traduzione delle idee in concrete composizioni sulla pagina.

Riferimenti bibliografici

GIACOMELLI, Roberto (2021). *Guida al linguaggio Lua per $\text{Lua}\TeX$* . GuIT, Gruppo Utilizzatori Italiani di $\TeX$. <https://www.guitex.org/home/images/doc/GuideGuIT/guidalua.pdf>.

IERUSALIMSKY, Roberto (2016). *Programming in Lua, Fourth Edition*. Lua.Org.

LUA $\TeX$ DEVELOPMENT TEAM (2023). *Lua $\TeX$ Reference Manual*. [Consultabile a schermo con il comando `$ texdoc luatex`].

SYMPY DEVELOPMENT TEAM (2023). «Official web site». <https://www.sympy.org>. [Online; in data 21 novembre 2023].

WIKIPEDIA (2023). «Metodi per il calcolo della radice quadrata — wikipedia, l'enciclopedia libera». http://it.wikipedia.org/w/index.php?title=Metodi_per_il_calcolo_della_radice_quadrata&oldid=136277502. [Online; in data 21 novembre 2023].

Roberto Giacomelli

ISTITUTO DI ISTRUZIONE SUPERIORE ZACCAGNA GALILEI — CARRARA
giaconet.mailbox@gmail.com

L^AT_EX per la composizione di prove d'esame e di verifica

Grazia Messineo e Salvatore Vassallo

Sommario A distanza di più di 15 anni, rifacciamo un'analisi dettagliata dei pacchetti disponibili per la creazione di verifiche e prove d'esame. Dal 2007 ad oggi i pacchetti di questo tipo sono notevolmente aumentati e molti di essi hanno caratteristiche interessanti che illustriamo in questo articolo.

Abstract Over 15 years later, we make another detailed analysis of the available packages for creating assessments and exams. From 2007 to present days, these types of packages have significantly increased, and many of them have interesting features that we will highlight in this article.

1. Introduzione

Quando nel 2007 abbiamo sentito, in Università Cattolica, l'esigenza di comporre prove d'esame con un elevato numero di varianti (da 20 a 40), senza usare software commerciali, abbiamo esplorato i pacchetti L^AT_EX allora disponibili, che ci consentissero di realizzare quanto richiesto, senza trovarne nessuno che soddisfacesse tutte le nostre esigenze.

I pacchetti più vicini a quello che volevamo, ovvero *eqexam*, *exam* e *examdesign*, non avevano tutte le caratteristiche che ci occorreavano. In particolare, non gestivano i parametri casuali, non sempre consentivano il mescolamento delle domande, delle risposte e non tutti consentivano di produrre in automatico le soluzioni. Inoltre, nessuno di essi consentiva di gestire un numero di varianti così elevato. Questo ci ha portato a scrivere il nostro primo pacchetto, *esami* (si vedano MESSINEO e VASSALLO (2023) e MESSINEO e VASSALLO (2012)).

Oggi il panorama è decisamente cambiato: una rapida ricerca sul CTAN mostra che i pacchetti che ricadono a vario titolo nel topic “exam” sono circa una sessantina e circa una trentina nel topic “exercise”.

Confrontandoci con un collega (di scuola e del G_JIT) abbiamo capito che esigenze che noi davamo per scontate e sentite da tutti in realtà non lo sono, mentre si possono manifestare, nella composizione di verifiche o esami, necessità che noi non abbiamo preso in considerazione.

Pertanto, abbiamo deciso di esaminare alcuni pacchetti oggi a disposizione per la composizione di prove d'esame e di verifica, illustrandone le caratteristiche principali. In questa disamina, non abbiamo preso in considerazione, il pacchetto *esami*, poiché già oggetto di interventi in altri G_JIT *meeting*.

Premettiamo alla nostra analisi un'osservazione generale. Ci sono molte tipologie di esercizi che possono essere inseriti in una verifica o in un esame:

- Domande a scelta multipla¹: cioè domande a risposta chiusa in cui una sola risposta è corretta.

1. A volte useremo l'acronimo MCQ.

- Domande a risposta multipla (acronimo MAQ): cioè domande a risposta chiusa in cui ci possono essere più risposte corrette (o anche nessuna). Osserviamo che, a nostro parere, con questa tipologia di domande è necessario prevedere un punteggio negativo per le risposte sbagliate, in quanto, in caso contrario, uno studente che segna tutte le risposte ottiene sempre il massimo dei punti; questo porta a qualche complicazione nella programmazione per i pacchetti che prevedono la correzione automatica.
 - Esercizi “a riempimento”, come, ad esempio, “Giuseppe _____ guidò la spedizione dei Mille che iniziò nell’anno _____”.
 - Esercizi “a risposta breve” come “La capitale della Spagna è ” o “ $2 + 2 = \text{$ ”.
- Noi indicheremo sia questa tipologia di esercizi che la precedente come esercizi *fillin* poiché sono molto simili se non identici da un punto di vista del codice usato per scriverli. C’è un unico pacchetto che li differenzia, il pacchetto *moodle* (si veda il Paragrafo 4.3) che serve a preparare esercizi per la piattaforma MOODLE, in quanto tale piattaforma li considera diversi.
- Esercizi Vero/Falso che possono rientrare sia nella tipologia *fillin* che tra le MCQ a seconda di come vengono presentati.
 - Esercizi *matching* in cui bisogna far corrispondere agli elementi di un elenco (ad esempio, nomi di nazioni) gli elementi di un altro (ad esempio nomi di capitali).
 - Esercizi a risposta aperta “lunga” che possono essere suddivisi in parti. Spesso, in Matematica vengono chiamati “problemi”.

2. I pacchetti per le prove di verifica

In questa sezione parleremo di due pacchetti specifici per la produzione di verifiche per le scuole secondarie. Entrambi i pacchetti sono di autore italiano.

La nostra scelta di questi due pacchetti fra i tanti presenti sul CTAN deriva dal fatto che molti pacchetti sono specifici per le prove in una nazione (ce ne sono molti dedicati alle scuole cinesi, con documentazione solo in cinese) o non presentano tutte le tipologie di esercizio a cui siamo interessati, ad esempio ce ne sono molti dedicati alla MCQ, alcuni solo ad esercizi a risposta aperta o a domande *fillin*.

2.1. Verifica

Verifica è una classe di Riccardo Dossena creata per comporre verifiche con lo stile della scuola superiore italiana. Può essere usata, con alcuni adattamenti, anche per comporre temi d’esame universitari (DOSSENA, 2020).

L’instestazione della prova può essere ottenuta con tre stili diversi (standard, semplice e su righe), ognuno dei quali è personalizzabile con comandi opportuni (per esempio si può specificare il tipo di prova, il tempo a disposizione, la classe, la data, il nome dello studente, ecc.).

La classe mette a disposizione alcuni ambienti o comandi per esercizi, problemi e test:

- L’ambiente *esercizi*, che consente di creare elenchi di domande (l’elenco è gestito dal pacchetto *enumitem*). Ogni domanda principale può essere divisa in sotto-domande e ci sono varie opzioni di personalizzazione degli elenchi;

- Gli ambienti `test`, `test-orizz` e `test-orizz-newline` per test a risposta multipla. Gli ambienti per test vanno inseriti all'interno dell'ambiente `esercizi` e producono, rispettivamente, un test con risposte allineate in verticale, in linea o in orizzontale su nuova linea. Le box per le risposte possono anch'esse essere personalizzate ampiamente.
- il comando `\vf` per scrivere le caselle "Vero" e "Falso" alla fine di una domanda all'interno di un ambiente `esercizi` (il comando traduce automaticamente le label V ed F nell'inglese T ed F se la lingua selezionata è l'inglese). I test vero - falso possono essere scritti anche usando l'ambiente `test-verofalso`. All'interno di questo ambiente, sostituendo il comando `\item` delle liste usuali con `\vfitem`, il testo viene automaticamente allineato in modo da non sovrascrivere le box vero e falso.

Il pacchetto mette inoltre a disposizione i comandi per generare le righe per le risposte (`\riga(numero righe)`) e i comandi per generare gli spazi per gli esercizi a completamento (comandi `\dotword` e `\dotrule`).

Prevede inoltre dei comandi per inserire il punteggio di ogni singolo esercizio (comando `\punti[<testo aggiuntivo>]{<punteggio>}`) e una tabella con il totale dei punti (comando `\totpunti`).

Il codice che segue, tratto da uno dei file di esempio del pacchetto, ne mostra le possibilità principali.

```
\begin{esercizi}
\item La soluzione dell'equazione  $x+1=0$  è:
\begin{test-orizz}
\item  $1$ ;
\item  $-1$ ;
\item  $0$ ;
\item  $+\infty$ .
\end{test-orizz}
\punti{5}
\item Quale fra le seguenti è la derivata di  $3x^2$ ? \punti{5}
\begin{test-orizz-newline}
\item  $3x$ ;
\item  $3$ ;
\item  $3x^2$ ;
\item  $6x$ .
\end{test-orizz-newline}
\medskip
{\setlength\columnsep{2.5cm}} % lo spazio tra le colonne si può modificare
\begin{multicols}{2}
\item Indicate con  $p$  e  $q$  due generiche proposizioni, quattro delle seguenti affermazioni sono tra loro logicamente equivalenti, mentre una non lo è con le altre. Quale?
\begin{test}
\item  $p$  implica  $q$ 
\item  $q$  è condizione necessaria per  $p$ 
\item  $p$  segue dal verificarsi di  $q$ 
\item  $p$  solo se  $q$ 
\item  $p$  è condizione sufficiente per  $q$ 
\end{test}
\columnbreak
\item Quale delle seguenti frasi è equivalente a
\emph{non è vero che Mario studia e ascolta la radio}
\begin{test}
\item Mario studia e ascolta la radio
\item Mario studia o ascolta la radio
\item Mario non studia né ascolta la radio
\item Mario non studia o non ascolta la radio
\end{test}
\end{multicols}
\item Vero o falso? \punti[per ognuna]{2}
```

```

{\setlength\columnsep{1.3cm} % lo spazio tra le colonne si può modificare
\begin{multicols}{3}
\begin{test-verofalso}[fattorevf=.6]
\vfitem{$1+1=2$}
\vfitem{$3+0=0$}
\vfitem{$3\times 3=9$}
\vfitem{$0:3=1$}
\vfitem{$\lim_{x\to 2}x^2=4$}
\vfitem{$\sum_{n=0}^1n^2=1$}
\end{test-verofalso}
\end{multicols}
\item Enuncia il teorema di Weierstrass. \punti{8}
\riga{2}
\item \emph{Completa.} \doublespacing In un triangolo \dotword{rettangolo}
il quadrato costruito sull' \dotword{ipotenusa} è uguale alla
\dotword{somma} dei \dotword{quadrati} costruiti sui
\dotword{cateti}. \punti{20} \par\singlespacing
\item Stabilisci se le seguenti affermazioni sono vere o
false. \punti[per ognuna]{10}
\begin{test-verofalso}[fattorevf=.85]
\vfitem{Il coefficiente angolare di una retta si deve \emph{sempre} indicare con la lettera $m$ e
l'ordinata all'origine \emph{sempre} con la lettera $q$}
\vfitem{Se una funzione derivabile in un intervallo ha derivata positiva, allora la funzione è
crescente in tale intervallo.}
\end{test-verofalso}
\end{esercizi}
\totpunti[/10+2]

```

Il pacchetto non prevede al momento la possibilità di produrre un file delle soluzioni da usare come traccia per la correzione o da fornire agli studenti a scopo didattico.

2.2. Verifiche

Il pacchetto *verifiche* di Francesco Raccanello è un altro pacchetto per la composizione di verifiche per le scuole superiori, anch'esso orientato principalmente al sistema scolastico italiano (RACCANELLO, 2024).

Il pacchetto mette a disposizione diverse opzioni, già in fase di caricamento, per stampare o no il nome della scuola, il nome del candidato, il tipo di prova, ecc.

L'opzione più interessante tra quelle iniziali è l'opzione *solutions*, che consente di ottenere, se specificata, la soluzione del test proposto e, se non specificata, la prova da somministrare agli studenti.

Gli ambienti fondamentali per produrre esercizi sono

- **esercizio**: è un ambiente che consente di scrivere gli esercizi della prova.

```

\begin{esercizio}[ label =<label>, diff =<number>, pt =<number> , partialpt]
...
\end{esercizio}

```

Le opzioni interessanti sono quelle per attribuire un punteggio (*pt* per attribuire i punti, *partialpt* per sommare i punteggi parziali attribuiti ad ogni domanda mediante l'omonimo comando *\partialpt*) e soprattutto l'opzione *diff* che consente di esprimere, mediante un numero di pallini decisi dal docente, il livello di difficoltà dell'esercizio. Di default gli esercizi sono numerati. Se si desidera averli non numerati si può usare la versione *esercizio**.

La soluzione dell'esercizio può essere scritta nell'ambiente `soluzione` o nel comando `\inlinesol{soluzione}`, il cui contenuto viene stampato solo se nel file sorgente è specificata l'opzione `solutions`.

- `\truefalse`: questo comando produce test del tipo vero-falso. Il parametro obbligatorio (V o F) consente di mettere un segno di spunta sulla risposta corretta se l'opzione `solutions` è specificata:

Vero o Falso? <code>\truefalse {V}</code>

- `closedquestions` e `closedquestion` consentono di creare quesiti a risposta multipla. Ogni risposta è introdotta dal comando `\item` e l'opzione `checked` consente di specificare la risposta corretta, segnalata nelle soluzioni dalla spunta. La differenza tra i due ambienti è l'allineamento verticale o orizzontale delle risposte.
- `\finderror` crea un esercizio del tipo “trova l'errore”. I due argomenti obbligatori sono il testo sbagliato e il testo corretto. Nella soluzione stampa il testo sbagliato coperto di linee di cancellazione e sopra il testo corretto.

Esistono poi due comandi di “supporto”:

- `\openquestion`, che consente di creare un supporto a righe (`lines`) o a quadretti (`squares`) per la risposta;
- `\textandimage`, che consente di affiancare il testo di un esercizio ad una immagine.

Il pacchetto consente inoltre di produrre un numero n di verifiche, definito dall'utente, mediante l'ambiente `multitest`, usando la possibilità di randomizzazione di `pgf` e `tikz`.

In alternativa, la documentazione del pacchetto suggerisce l'uso di `datatool` per poter operare con esercizi inseriti in un database e creare un numero a piacere di verifiche diverse.

3. I pacchetti per esami universitari

In questa sezione parleremo dei pacchetti che sono stati creati avendo come scopo primario la creazione di esami universitari. Ovviamente tutti o quasi possono essere usati anche per produrre verifiche per le scuole secondarie.

Anche in questo caso la nostra scelta è caduta su pacchetti che permettano l'uso di più tipologie di esercizio e che non siano specifici per un'area geografica (anche qui molti sono solo in cinese).

Tra i pacchetti che non abbiamo trattato è comunque di sicuro degno di menzione `probsoln` di Nicola Talbot (TALBOT, 2017) a cui siamo debitori per il nostro pacchetto `esami`.

Il pacchetto permette la scelta di esercizi da un file che ne contiene molti e l'inserimento degli esercizi scelti in un file di problemi da sottoporre agli studenti.

3.1. Exam

`Exam` è una classe $\LaTeX$ che consente di comporre esami (o verifiche scolastiche), mettendo a disposizione una serie di comandi che consentono di gestire in modo automatico molti aspetti della composizione:

- Consente di creare domande (anche divise in parti), con numerazione automatica delle domande e delle loro parti.
- Consente di attribuire un punteggio sia alla domanda principale che alle sue parti, con una macro per sommare in automatico i punti.
- Mette a disposizione una tabella riassuntiva per la valutazione del compito o verifica.
- Mette a disposizione una serie di comandi per personalizzare *header*, *footer*, margini e per creare una pagina di copertina, laddove necessario.
- Mette a disposizione un ambiente per scrivere le soluzioni degli esercizi, da includere in un file per la correzione.

La classe ha un manuale ben dettagliato (HIRSCHHORN, 2023) ed è stata descritta in dettaglio in PILU (2011), quindi ne descriveremo sommariamente le caratteristiche principali, rimandando all'articolo di Antonello Pilu e al manuale per maggiori dettagli.

3.1.1. *L'intestazione e la parte iniziale dell'esame*

Il pacchetto consente di creare per l'esame o per la verifica un'intestazione, le istruzioni e la tabella per la valutazione. Tutto questo materiale può essere contenuto all'inizio della prova oppure in una pagina di copertina.

Il pacchetto non è localizzato in italiano, quindi all'inizio del documento vanno tradotte le parole che compaiono nei vari ambienti. Ad esempio, il codice

```
\pagestyle{head}
\firstpageheader{}{}{}
\runningheader{\class{\examnum} - Pag. \thepage di \numpages}{\examdate}
\runningheadrule
\pointpoints{Punto}{Punti}
\pagestyle{headandfoot}
\firstpageheader{\bf\large Esame di \class}{\bf\large \examdate}
\runningheader{\bf\large Esame di \class}{\bf\large \examdate}
\headrule
\footer{}{Pag. \thepage di \numpages}{\textit{\iflastpage{Fine del compito}{segue \dots}}}
```

consente di ottenere intestazioni e piè di pagina con le scritte in italiano.

Il codice della Tabella 1 mostra come creare un'intestazione personalizzata. Il codice della Tabella 2 mostra come creare una tabella per la valutazione, in orizzontale, con la scritta in italiano.

3.1.2. *Le tipologie di domanda*

L'ambiente di base per le domande è l'ambiente **questions**. Questo ambiente consente di generare domande (numerate) che si "autoconcludono", domande divise in parti, domande a risposta multipla, domande a risposta breve o lunga, ecc.

La Tabella 3 mostra il codice per creare una domanda in una sola parte o in più parti: ogni domanda prevede che possa essere indicato un punteggio tra parentesi quadre. I punteggi delle domande divise in parti vengono automaticamente sommati.

Il codice della Tabella 4 mostra come creare domande a scelta multipla, con una sola risposta corretta. Le risposte possono essere allineate verticalmente, orizzontalmente o su più colonne. È inoltre possibile creare domande a risposta multipla, con caselle di controllo da spuntare, come mostra il codice della Tabella 4. Le caselle sono normalmente tonde, se si

```
\newcommand{\class}{Matematica Generale}
\newcommand{\term}{9 gennaio 2024}
\newcommand{\examnum}{Esame 1}
\newcommand{\examdate}{9/1/2024}
\newcommand{\timelimit}{60 Minuti}
\begin{document} \noindent
\begin{tabular*}{\textwidth}
{l @{\extracolsep{\fill}} r
@{\extracolsep{6pt}} l}
\textbf{\class} & \textbf{Nome:} & \\
& \makebox[2in]{\hrulefill} & \\
\textbf{\term} & & \\
\textbf{\examnum} & & \\
\textbf{\examdate} & & \\
\textbf{Tempo concesso: \timelimit} & & \\
& \textbf{Docente} & \makebox[2in]{\hrulefill} \\
\end{tabular*} \\
\rule[2ex]{\textwidth}{2pt}
Questo esame contiene \numpages\ pagine
e \numquestions\ domande. \\
I punti totali sono \numpoints.
Scrivere le risposte su questo foglio,
negli spazi assegnati. Le risposte alle
domande aperte devono essere adeguatamente
giustificate.
\end{document}
```

Esame di Matematica Generale9/1/2024

Matematica GeneraleNome: _____

Esame 1

9/1/2024

Tempo concesso: 60 MinutiDocente _____

Questo esame contiene 4 pagine e 11 domande.
I punti totali sono 103.
Scrivere le risposte su questo foglio, negli spazi assegnati. Le risposte alle domande aperte
devono essere adeguatamente giustificate.

Tabella 1. Intestazione in exam

```
{\begin{center}
\hword{Esercizio}
\hword{Punti}
\bhpword{Punti Bonus}
\htword{\textbf{Tot}}
\vwword{Domanda}
\hword{Domanda}
\vpword{Punti}
\vsword{Punteggio}
\hsword{Punteggio}
\vtword{Totale}
Tabella di valutazione (solo per il docente) \\
\addpoints
\gradetable[h][questions]
\end{center}}
```

Tabella di valutazione (solo per il docente)												
Domanda	1	2	3	4	5	6	7	8	9	10	11	Tot
Punti	1	20	2	2	3	3	2	10	20	20	20	103
Punteggio												

Tabella 2. Tabella del punteggio in exam

vogliono avere, ad esempio, quadrate, si possono cambiare localmente inserendo, prima di introdurre l’ambiente checkboxes, il comando `\checkboxchar{$\Box$}` (come mostrato nella seconda domanda della figura della Tabella 4). Osserviamo che il pacchetto non prevede di base la possibilità di randomizzare né gli esercizi, né le risposte delle domande a scelta multipla o a risposta multipla.

Infine, è possibile creare domande con risposta breve o lunga, contenuta in uno spazio delimitato, con un numero di righe assegnato (linee o punti), come mostra il codice che segue

```
\question[10]
Spiega, in non più di un paragrafo, perché\dotso %esempio di domanda con risposta breve (1 paragrafo)
\makeemptybox{5cm}
```

La box vuota creata con il comando del codice appena visto può essere sostituita da

```

\begin{questions}
\question[1] Quanto fa  $2+2$ ?
\addpoints
\question[20] Considera la funzione
\[f(x)=3x^3+2x^2+x+1\]
\noaddpoints % serve a non conteggiare due
volte i punti
\begin{parts}
\part[10] Calcola  $f'(x)$ .
\part[10] Calcola  $f''(x)$ .
\end{parts}
\addpoints
...
\end{questions}

```

1. (1 Punto) Quanto fa $2+2$?
2. (20 Punti) Considera la funzione

$$f(x) = 3x^3 + 2x^2 + x + 1$$

- (a) (10 Punti) Calcola $f'(x)$.
- (b) (10 Punti) Calcola $f''(x)$.

Tabella 3. Domande semplici o in parti in exam

```

\question[2] Esempio di test a risposta
multipla allineato verticalmente
\begin{choices}
\choice Risposta 1
\choice Risposta 2
\choice Risposta 3
\choice Risposta 4
\choice Risposta 5
\end{choices}
\question[2] Esempio di test a risposta
multipla allineato orizzontalmente
\begin{oneparchoices}
\choice Risposta 1
\choice Risposta 2
\choice Risposta 3
\choice Risposta 4
\choice Risposta 5
\end{oneparchoices}
{\renewcommand*\thechoice
{\arabic{choice}}}
\renewcommand*\choicelabel
{\thechoice}
\question[2] Questa è una domanda con
risposte su più colonne
\begin{multicols}{2}
\begin{choices}
\choice Risposta 1
\choice Risposta 2
\choice Risposta 3
\choice Risposta 4
\choice Risposta 5
\choice Risposta 6
\choice Risposta 7
\choice Risposta 8
\end{choices}
\end{multicols}

```

3. (2 Punti) Esempio di test a risposta multipla allineato verticalmente
 - A. Risposta 1
 - B. Risposta 2
 - C. Risposta 3
 - D. Risposta 4
 - E. Risposta 5
4. (2 Punti) Esempio di test a risposta multipla allineato orizzontalmente
 - A. Risposta 1
 - B. Risposta 2
 - C. Risposta 3
 - D. Risposta 4
 - E. Risposta 5
5. (2 Punti) Questa è una domanda con risposte su più colonne

1) Risposta 1	5) Risposta 5
2) Risposta 2	6) Risposta 6
3) Risposta 3	7) Risposta 7
4) Risposta 4	8) Risposta 8

Tabella 4. Domande a scelta multipla in exam

```

\question[3] Metti un segno di spunta sulla casella se
la risposta è corretta.
\addpoints
\begin{checkboxes}
\choice  $2+2=4$ 
\choice  $\frac{d}{dx}(x^2+1) = 2x+1$ 
\choice  $2x+1 = 2x+1$ 
\choice  $1$  non è un numero naturale.
\end{checkboxes}

```

6. (3 Punti) Metti un segno di spunta sulla casella se la risposta è corretta.
 - ☐ $2+2=4$
 - ☐ $\frac{d}{dx}(x^2+1) = 2x+1$
 - ☐ 1 non è un numero naturale.
7. (3 Punti) Metti un segno di spunta sulla casella se la risposta è corretta.
 - ☐ $2+2=4$
 - ☐ $\frac{d}{dx}(x^2+1) = 2x+1$
 - ☐ 1 non è un numero naturale.

Tabella 5. Domande a risposta multipla in exam

```

\question[20]
Spiega, limitando la lunghezza della risposta
alle sole righe assegnate, perché \dots
%esempio di domanda con righe
per la risposta
\fillwithlines{\fill}
\begin{solutionorlines}[\fill]
    Questa è la soluzione
\end{solutionorlines}

```

10. (20 Punti) Spiega, limitando la lunghezza della risposta alle sole righe assegnate, perché...

Soluzione:
Questa è la soluzione

Tabella 6. Soluzioni alle domande aperte in exam

- una box che riempia la pagina: `\makeemptybox{\fill};`
- delle linee: `\fillwithlines{\fill}`
- delle linee punteggiate: `\fillwithdottedlines{8em}`

3.1.3. Le soluzioni

Il pacchetto `exam` prevede anche la possibilità di creare in automatico le soluzioni per la prova che si sta preparando.

Le soluzioni delle domande aperte si ottengono con gli ambienti `solution`, `solutionorbox`, `solutionorlines`, `solutionordottedlines`, `solutionorgrid`.

In tutti questi ambienti è possibile scrivere la soluzione dell'esercizio, che viene stampata solo se la classe `exam` viene caricata con l'opzione `answers` oppure se si usa il comando `\printanswers`. Se la classe viene invece caricata senza l'opzione `answers` oppure se si usa il comando `\noprintanswers`, viene stampato rispettivamente (se viene specificata una misura come parametro opzionale) uno spazio vuoto, uno spazio delimitato da riquadro, delle righe, delle linee punteggiate, una griglia.

Vediamo un esempio nella Tabella 6.

Per indicare invece la risposta o le risposte corrette nelle domande a risposta multipla o a scelta multipla, è possibile sostituire, per le risposte corrette, il comando `\choice` che introduce ciascuna alternativa con il comando `\CorrectChoice`. In questo caso, nelle soluzioni la risposta corretta sarà stampata in grassetto.

Vediamo anche in questo caso un esempio nella Tabella 7. Anche le soluzioni hanno ampie possibilità di personalizzazione dello stile, si rimanda alla documentazione del pacchetto (HIRSCHHORN, 2023) o all'articolo (PILU, 2011) per ulteriori dettagli.

3.2. Exam-randomizechoices

Il pacchetto `exam-randomizechoices` (OP DEN BROUW, 2021) estende le funzionalità del pacchetto `exam`, definendo dei nuovi ambienti per le domande a risposta multipla, mediante i quali le scelte possono essere riordinate in modo casuale.

Gli ambienti nuovi sono:

- `randomizechoices`
- `randomizeoneparchoices`
- `randomizecheckboxes`
- `randomizeoneparcheckboxes`

```

\question[2] Esempio di test a risposta
multipla allineato orizzontalmente
\begin{oneparchoices}
\choice Risposta 1
\CorrectChoice Risposta 2
\choice Risposta 3
\choice Risposta 4
\choice Risposta 5
\end{oneparchoices}
\question[3] Metti un segno di spunta sulla
casella se la risposta è corretta.
\addpoints
\begin{checkboxes}
\CorrectChoice 2+2=4
\choice $\frac{d}{dx}$ $(x^2+1)= 2x+1$
\choice $1$ non è un numero naturale.
\end{checkboxes}

```

5. (2 Punti) Esempio di test a risposta multipla allineato orizzontalmente
 A. Risposta 1 **B. Risposta 2** C. Risposta 3 D. Risposta 4 E. Risposta 5
6. (3 Punti) Metti un segno di spunta sulla casella se la risposta è corretta.
☒ $2+2=4$
☐ $\frac{d}{dx}(x^2+1) = 2x+1$
☐ 1 non è un numero naturale.

Tabella 7. Soluzioni alle domande a scelta multipla in exam

```

\question[2] Esempio di test a risposta multipla
allineato verticalmente (primo)
\begin{randomizechoices}
\CorrectChoice Risposta 1
\choice Risposta 2
\choice Risposta 3
\choice Risposta 4
\choice Risposta 5
\end{randomizechoices}
\question[2] Esempio di test a risposta multipla
allineato verticalmente (secondo)
\begin{randomizechoices}
\CorrectChoice Risposta 1
\choice Risposta 2
\choice Risposta 3
\choice Risposta 4
\choice Risposta 5
\end{randomizechoices}

```

1. (2 Punti) Esempio di test a risposta multipla allineato verticalmente (primo)
 A. Risposta 1
 B. Risposta 4
 C. Risposta 3
 D. Risposta 5
 E. Risposta 2
2. (2 Punti) Esempio di test a risposta multipla allineato verticalmente (secondo)
 A. Risposta 3
 B. Risposta 4
 C. Risposta 1
 D. Risposta 5
 E. Risposta 2
1. (2 Punti) Esempio di test a risposta multipla allineato verticalmente (primo)
A. Risposta 1
 B. Risposta 4
 C. Risposta 3
 D. Risposta 5
 E. Risposta 2
2. (2 Punti) Esempio di test a risposta multipla allineato verticalmente (secondo)
 A. Risposta 3
 B. Risposta 4
C. Risposta 1
 D. Risposta 5
 E. Risposta 2

Tabella 8. Testo e soluzioni alle domande chiuse in examrandomizechoices

che sostituiscono i corrispondenti ambienti del pacchetto exam qualora si vogliano le risposte permutate in ordine casuale.

Nella Tabella 8 è riportato un esempio di codice (lo stesso inserito due volte per mostrare il risultato) con l'output, sia come testo che come soluzione.

Il pacchetto ha alcune opzioni interessanti, ne citiamo solo alcune:

- Opzioni che possono essere date sia agli ambienti che globalmente al pacchetto quando lo si carica:
 - norandomize e randomize per disattivare o attivare la randomizzazione. Di default, se non si specifica nulla, con i nuovi ambienti le risposte vengono permutate in ordine casuale;
 - nokeepplast e keepplast per non fissare (default) o fissare l'ultima risposta;
- Opzioni utilizzabili solo in fase di caricamento del pacchetto:
 - overload e nooverload, per sovrascrivere o non sovrascrivere gli ambienti del pacchetto exam originario con i nuovi ambienti che permutano le risposte (di

<pre>\keylistquestionname{Domanda} \keylistkeyname{Risposta} \ifprintanswers \printkeytable \fi</pre>	1. (2 Punti) Esempio di test a risposta multipla allineato verticalmente (primo) A. Risposta 4 B. Risposta 5 C. Risposta 3 D. Risposta 1 E. Risposta 2 2. (2 Punti) Esempio di test a risposta multipla allineato verticalmente (secondo) A. Risposta 5 B. Risposta 3 C. Risposta 2 D. Risposta 1 E. Risposta 4 <table><tr><th>Domanda</th><th>Risposta</th></tr><tr><td>1</td><td>D</td></tr><tr><td>2</td><td>D</td></tr></table>	Domanda	Risposta	1	D	2	D
Domanda	Risposta						
1	D						
2	D						

Tabella 9. Tabella con la lista delle risposte esatte exam-randomizechoices

- default non vengono sovrascritti);
- debug, per ottenere una serie di messaggi di debug.

È possibile definire il seme casuale mediante il comando `\setrandomizerseed`.
È inoltre possibile stampare alla fine della prova una tabella con la lista delle risposte esatte. Questa tabella si ottiene (in italiano) con il codice della Tabella 9. Anche per questo comando ci sono alcune opzioni, la più interessante delle quali si ottiene con l’argomento opzionale del comando `\printkeytable[pag. iniziale - pag. finale]`, che consente di specificare le domande per le quali si vuole la chiave delle risposte. Si osservi che, se gli ambienti di default non sono sovrascritti, nella tabella delle risposte in corrispondenza di quelle domande appare il segno dell’etichetta non definita.

3.3. Examz

La classe examz (McCLAIN, 2023b) nasce anch’essa per estendere le funzionalità di exam e permettere di creare esami con versioni multiple, elementi casuali ed esercizi contenuti in file esterni (banche dati).

L’idea della classe è quella di creare, sulla base del pacchetto counterz (McCLAIN, 2023a), una serie di istruzioni che consentano di creare versioni multiple di una prova a partire dagli stessi esercizi o da esercizi con versioni simili.

L’ambiente di base per la creazione di queste versioni è versions, all’interno del quale va inserito tutto il “materiale” che compone la prova:

```
\begin{versions}[numero di varianti]
  contenuto della prova
\end{versions}
```

Gli esercizi dai quali deve essere composta la prova vengono scritti su file esterni, un file per ogni tipologia di esercizio.

Il file degli esercizi (questionbank nella denominazione della classe) deve contenere obbligatoriamente all’inizio il numero di domande (varianti) di cui il file è composto:

```
\setnumberofquestions{1}%
```

Le domande vanno tutte inserite nell’ambiente questionbank ed ognuna di esse è introdotta dal comando `\qbitem{numero progressivo esercizio}`. Ciascun esercizio può essere diviso

in parti oppure no. Gli elementi casuali (parametri) sono introdotti usando opportuni comandi del pacchetto `counterz`, mediante i quali è anche possibile eseguire semplici operazioni sui parametri:

```
\begin{questionbank}
  \begin{qbitem}{1}
    \begin{parts}
      \part
        \xrandprovidecounternz{ua}{-9}{9}%
        \xrandprovidecounternz{ud}{-9}{9}%
        \xprovidecounter{uad}%
        \xsetcounter{uad}{\xvalue{ua}*\xvalue{ud}}%
        Risolvi il seguente sistema:
        \[testo del sistema\]
        \begin{qbsolution}
          Soluzione
        \end{qbsolution}
      \part
        Un altro pezzo di esercizio
    \end{parts}
  \end{qbitem}
  \begin{qbitem}{2}
    altro esercizio
  \end{qbitem}
\end{questionbank}
```

Una volta scritti gli esercizi sui file esterni, è possibile includerli nella prova che si vuole comporre usando il comando `\questionfrombank` all'interno dell'ambiente `versions`:

```
\begin{versions}
  \printcover
  \setquestionpath{percorso dei file con gli esercizi/}
  \begin{questions}
    \questionfrombank[punti]{File esercizi 1}
    \questionfrombank[punti]{File esercizi 2}
    \questionfrombank[punti]{File esercizi 3}
  \end{questions}
  \printworkspace
\end{versions}
```

Il comando opzionale `\printworkspace` consente di stampare dello spazio aggiuntivo per i conti (solo nella versione testo, non nelle soluzioni).

Per il resto, la classe eredita tutti gli ambienti per gli esercizi dalla classe `exam`, che viene automaticamente caricata.

3.4. Examdesign

Il pacchetto `examdesign` (ALEXANDER, 2006) è uno dei più completi e consolidati pacchetti per la generazione di prove d'esame. Poiché non è localizzato in italiano e non è semplice intervenire per introdurre la lingua italiana, lo presenteremo molto brevemente.

Ha a disposizione tutte le classiche tipologie di esercizio:

- Vero/Falso,
- Riempimento,
- Risposta breve,
- Scelta multipla,

- Matching,
- Problemi.

È possibile generare più versioni dello stesso esame (al massimo ventisei) con riordinamento casuale dell'ordine degli esercizi all'interno di una sezione e degli elementi nelle domande di tipo *matching*, mentre l'ordine delle risposte nelle MCQ rimane fissato. È anche possibile fissare il seme per la generazione delle sequenze aleatorie così che successive compilazioni diano lo stesso risultato.

L'esame può essere stampato su due colonne.

Il pacchetto permette anche di creare i fogli di soluzioni sia in forma “estesa” (con tutte le domande e le risposte, anche dei quesiti a risposta multipla) che in forma “ridotta” (in cui appaiono solo le risposte esatte).

Il codice per la generazione delle prove è semplice ed intuitivo, ma, come detto, a quanto ci consta, supporta solo la lingua inglese.

3.5. *Exam-n*

Purtroppo anche il pacchetto *exam-n* (GRAY, 2022) non è localizzato in italiano e per poter avere un compito in italiano bisogna riscrivere molte parti di codice della classe: in effetti il pacchetto è in un certo senso specifico per le università anglosassoni e richiede come dati necessari codici e nomi di corsi che in Italia non vengono generalmente usati.

Citiamo il pacchetto perché è uno dei pochi che ha come obiettivo quello di permettere una produzione “collaborativa” della prova d'esame: infatti è prevista (teoricamente) una struttura di questo genere

- l'organizzatore crea su un server una cartella con un file master e alcune sottocartelle, ognuna contenente un template e le istruzioni, dove verranno caricate le domande;
- ognuna delle sottocartelle è resa visibile al responsabile della rispettiva domanda, che provvede a scriverla e a caricarla sul server; il file che contiene una domanda è

```
\documentclass[compose]{exam-n}
... caricamento pacchetti necessari ...
\begin{document}
  \begin{question}
    ...
  \end{question}
\end{document}
```

- l'organizzatore scarica l'intera cartella e, quando ritiene che l'esame sia definitivo, lo genera.

Il pacchetto ha opzioni apposite per ognuno dei passaggi precedenti: con l'opzione *compose*, da usare nella prima fase, vengono mostrate le soluzioni, non ci sono controlli sul numero di esercizi, ogni esercizio viene mostrato su una pagina, con l'opzione *draft*, da usare quando il compito viene assemblato, vengono mostrate le soluzioni e viene lasciato uno spazio per i commenti, infine l'opzione *final* produce il compito vero e proprio ed è possibile avere o no le soluzioni.

Il pacchetto supporta solo domande a risposta multipla e domande aperte e non permette il rimescolamento casuale degli esercizi.

3.6. *Randexam*

Il pacchetto *randexam* (LYU, 2024) ha supporto nativo solo per il cinese e l'inglese, ma il manuale spiega come utilizzare in modo estremamente semplice altre lingue: in effetti è sufficiente scrivere in un file le traduzioni dei termini usati e poi caricare il file e richiedere l'utilizzo delle stringhe tradotte.

Il pacchetto supporta i seguenti tipi di esercizi:

- *fillin*,
- domande a scelta multipla (con quattro possibili scelte),
- Vero/Falso,
- domande a risposta aperta lunga.

È possibile indicare il punteggio non solo per ogni domanda, ma anche per ogni passaggio della soluzione.

Il pacchetto permette di generare una tabella riassuntiva in cui indicare il punteggio raggiunto in ogni parte e una "tabella delle risposte" in cui lo studente deve indicare le risposte date alle domande Vero/Falso e a risposta multipla.

L'intestazione e altre parti dell'esame sono facilmente modificabili dall'utente ed è possibile generare l'esame su due colonne.

L'autore suggerisce di generare un esame con due file: uno, diciamo *risposte.tex*, che contiene tutto il testo e le risposte, compilando questo file si avrà il file delle soluzioni, l'altro, diciamo *bianco.tex* che contiene semplicemente queste righe di codice

```
\PassOptionsToClass{noanswer}{randexam}
\input{risposte}
```

compilando il quale si ottiene il file dell'esame, senza risposte.

Una volta generato un esame (diciamo il file *A-risposte.tex*) è possibile generare un'altra versione con l'ordine delle domande in ogni parte e l'ordine delle risposte nelle MCQ modificato semplicemente scrivendo un file *B-risposte.tex* con queste righe di codice

```
\PassOptionsToClass{random}{randexam}
\input{A-risposte}
```

Nella Tabella 10 vediamo un esempio di domande di tipo *fillin* e le tabelle riepilogative dei punteggi: come si vede le domande vengono scritte in modo molto intuitivo e lo stesso accade con le altre tipologie. Si noti la differenza tra il comando `\fillin` e il comando `\fillout` che contengono entrambi la risposta esatta. Nella Tabella 11 vediamo un esempio di domande Vero/Falso e di domande a risposta multipla: la differenza tra la lettera maiuscola o minuscola nella risposta delle domande Vero/Falso è nell'apparenza delle soluzioni (lettera o simbolo, rispettivamente), mentre la differenza tra `\tickin` e `\tickout` e tra `\pickin` e `\pickout` è nella posizione dello spazio per la risposta (tutti questi comandi hanno come argomento la risposta esatta). Nelle domande a risposta multipla l'incolonnamento su due o una colonna avviene automaticamente.

```
\examtitle{name=Prova,date=\today}
\gradetable[total=4]
\exampart{Risposte brevi.}
\answertable[total=3,column=3]
\begin{question}
La capitale d'Italia è \fillout{Roma}.
\end{question}
\bigskip
\begin{question}
$2+2=$ \fillout{$4$}.
\end{question}
\bigskip
\begin{question}
La terza domanda \fillout{$corretto$}.
\end{question}
\newpage
```

Tabella 10. Domande *fillin* in randexam

Prima verifica

Nome: _____

Parte	I	II	III	IV	Totale
Punteggio					

Parte I: Risposte brevi. (3 domande; 1 punto ognuna; 3 punti in totale.)

Attenzione: si DEVONO scrivere le risposte nella seguente tabella.

Numero	1	2	3
Risposta			

1. La capitale d'Italia è _____.

2. $2+2=$ _____.

3. La terza domanda _____.

```
\exampart{Vero/Falso}[3 punti ognuna.]
\begin{question}
La prima domanda \tickout{V}
\end{question}
\begin{question}
La seconda domanda. \tickout{f}
\end{question}
\exampart{Scegliere la risposta corretta}
\begin{question}
La prima domanda \pickout{C}
\begin{abcd}
\item uno
\item due
\item tre
\item quattro
\end{abcd}
\end{question}
\begin{question}
La seconda domanda \pickin{B}
\begin{abcd}
\item questa prima risposta è molto lunga
\item 12
\item 13
\item 14
\end{abcd}
\end{question}
```

Tabella 11. Domande Vero/Falso e MCQ in randexam

Parte II: Vero/Falso (3 punti ognuna.)

1. La prima domanda[]

2. La seconda domanda. []

Parte III: Scegliere la risposta corretta

1. La prima domanda ()

(A) uno (B) due (C) tre (D) quattro

2. La seconda domanda ()

(A) questa prima risposta è molto lunga

(B) 12

(C) 13

(D) 14

3.7. *Eqexam*

Il pacchetto *eqexam* (STORY, 2021c), pur facendo parte del bundle *AcroT_EX* può essere usato in modo autonomo dagli altri pacchetti.

Anche questo pacchetto non è nativamente localizzato in italiano, ma può essere adattato alla lingua italiana scrivendo un semplice file di testo o modificando uno dei file di configurazione con le traduzioni dei termini usati e il procedimento viene spiegato nel manuale. Ad esempio per avere il nome dello studente, le istruzioni e i nomi dei punti in lingua italiana sarà sufficiente scrivere

```
\examNameLabel{Nome:}
\defaultInstructions{Istruzioni.}
\pointLabel{punto} % singular form
\pointsLabel{punti} % plural form
```

Il pacchetto è estremamente adattabile alle esigenze dell'utente e utilizza le principali tipologie di esercizi:

- domande a risposta multipla,
- domande a scelta multipla,
- Vero/Falso,
- *matching*,
- *fillin*,
- domande a risposta aperta breve,
- domande a risposta aperta lunga in una o più parti.

Le domande chiuse e quelle di tipo *matching* possono avere gli elementi in ordine casuale ed è anche possibile fissare alcuni o tutti gli elementi così che il loro ordine non venga modificato.

È possibile avere fino a 26 varianti di una prova e ci sono comandi per avere del testo alternativo in ognuna delle versioni.

Il pacchetto permette di avere un quadro riepilogativo delle risposte suddiviso per parti o per pagina (solo se viene creata una “copertina” per l'esame), un foglio delle soluzioni, di stampare solo le soluzioni, di stampare l'esame su due colonne, di avere una formattazione adatta a un libro (soluzioni dei problemi dispari alla fine del libro se in edizione per studenti, tutte le soluzioni in edizione per docenti).

Purtroppo questo pacchetto, come tutto il bundle *AcroT_EX* (si veda il paragrafo 4.1), è orfano per la scomparsa del professor Story e alcune indicazioni del manuale che fanno riferimento al sito di *AcroT_EX* che non è più raggiungibile sono recuperabili solo tramite Internet Archive (<https://web.archive.org/web/20221016233643/http://www.acrotex.net/>).

Nella Figura 1 vediamo l'intestazione con la tabella riassuntiva suddivisa per parti.

Vediamo ora come si scrivono alcuni esercizi. Nella Tabella 12 ci sono un esempio di esercizio a risposta lunga e uno di tipo *fillin*: nello spazio destinato alla risposta oltre alla riga continua possono essere usati i punti o uno spazio bianco. Lo spazio verticale per la soluzione può essere inserito o meno. Il punteggio totale di una Parte viene calcolato automaticamente.

Nella Tabella 13 possiamo vedere come appaiono le domande a risposta multipla e a scelta multipla. Il comando `\sqForms` serve a introdurre le caselle per la risposta. Si noti all'inizio del codice l'ambiente *instructions* vuoto: serve a far stampare il numero della parte

Nome: _____	
Matricola: _____	
Prof. Pico de' Paperis, 4 Aprile 2024	
Punteggio dell'Esame	
Parte 1 _____	/ 33 punti
Parte 2 _____	/ 15 punti
Totale: _____	/ 48 punti
Voto: _____	

Figura 1. La tabella riassuntiva del punteggio in eqexam

<pre>\begin{exam}[Parte I.]{Parte 1} \begin{instructions}[Parte I.] Risolvi i seguenti esercizi. Non ci sono penalità per le risposte errate. \end{instructions} \begin{problem}[5] Questo è un esempio di problema. La soluzione appare o meno a seconda dell'opzione scelta. Lo spazio per la soluzione è opzionale. \begin{solution}[3cm] La soluzione al problema. \end{solution} \end{problem} \begin{problem}[5] Un esempio di domanda \emph{fillin}:\n È noto che \fillin{4cm}{Newton} e \fillin{4cm}{Leibniz} sono ritenuti i fondatori del moderno calcolo infinitesimale. \end{problem}</pre>	Parte 1. (33 punti) Risolvi i seguenti esercizi. Non ci sono penalità per le risposte errate. (5pt) 1. Questo è un esempio di problema. La soluzione appare o meno a seconda dell'opzione. Può essere lasciato uno spazio per la soluzione. (5pt) 2. Un esempio di domanda <i>fillin</i>: È noto che _____ e _____ sono ritenuti i fondatori del moderno calcolo infinitesimale.
---	--

Tabella 12. Un esercizio e una domanda di tipo *fillin* in eqexam

e il suo punteggio. L'opzione auto nel primo problema (diviso in parti) serve a far calcolare automaticamente il punteggio.

Infine nella Tabella 14 vediamo un esempio di domanda di tipo *matching* e di domanda Vero/Falso. Si noti che il codice per la domanda *matching* è più complesso rispetto a quello di altri pacchetti

4. I pacchetti con supporto per esami online

In questa sezione esamineremo alcuni pacchetti che permettono di sottoporre agli studenti, oltre che le usuali prove cartacee, versioni online degli esercizi.

4.1. Il bundle Acrotex

Acrotex di D.P. Story è un bundle di vari pacchetti per la produzione di esami ed esercizi, sia interattivi che “statici”, con numerose “variazioni sul tema”. Il file che viene prodotto è sempre un file pdf (si veda STORY (2021a), STORY (2021d)) che può essere somministrato

```

\begin{exam}[Parte II.]{Parte 2}
\begin{instructions}[Parte II.]
\end{instructions}
\begin{problem*}[\auto]
Scegliere la risposta corretta.
\begin{parts}
\item\PTS{6} In che anno Colombo
partì alla volta delle Indie?
\begin{answers}{2} %tabular
\begin{choices}
\Ans0 1490\Ans
\Ans1 1492\Ans
\Ans0 1493\Ans
\end{choices}
\end{answers}
\sqForms % cambia formato alle risposte
\item\PTS{4} In che anno Colombo
partì alla volta delle Indie?
\begin{answers}{1} % list
\begin{choices}
\Ans0 1490\Ans
\Ans0 1491\Ans
\Ans1 1492\Ans
\Ans0 1493\Ans
\end{choices}
\end{answers}
\end{parts}
\end{problem*}
\begin{problem}[5]
Quali sono i colori primari?
\begin{answers}{6}
\begin{choices}
\Ans1 Blu\Ans
\Ans0 Verde\Ans
\Ans1 Giallo\Ans
\Ans0 Arancio\Ans
\Ans1 Rosso\Ans
\end{choices}
\end{answers}
\end{problem}
\end{exam}

```

Parte II. (15 punti)

1. Scegliere la risposta corretta.
 (10^{pt}) (a) (6 pt) In che anno Colombo partì alla volta delle Indie?
 (A) 1490 (B) 1491
 (C) 1492 (D) 1493

(b) (4 pt) In che anno Colombo partì alla volta delle Indie?
☐ 1490
☐ 1491
☐ 1492
☐ 1493

(5^{pt}) 2. Quali sono i colori primari?
 (a) Blu (b) Verde (c) Giallo (d) Arancio (e) Rosso

Tabella 13. Domande a risposta e a scelta multipla in **eqexam**

online oppure offline in modalità interattiva oppure ancora in forma cartacea (anche se per la somministrazione di prove in tale modalità viene indicato il pacchetto **eqexam**, si veda il paragrafo 3.7).

I pacchetti del bundle sono

- **web** che è essenzialmente un pacchetto di definizione di layout del file prodotto,
- **exerquiz** che è il pacchetto fondamentale per definire gli esercizi che possono essere sia online che cartacei, sia statici che interattivi,
- **eForms** che è un pacchetto di supporto alle *forms* dei file pdf (STORY, 2021b),
- **insdljs** che permette l’inserimento di codice javascript nel file pdf rendendolo pertanto “interattivo” (STORY, 2021b),
- **eq2db** che funziona da interfaccia per poter usufruire degli esercizi online (STORY, 2016).

Ci sono poi altri pacchetti e librerie che rientrano in qualche modo nel bundle, come, ad esempio, il già citato **eqexam**.

```

\begin{problem*}[\auto]
Associa a ogni nazione la sua capitale
\begin{multicols}{2}
\begin{parts}
\item\PTS*{2}\fillin{1cm}{\ref{par}}Francia
\item\PTS*{2}\fillin{1cm}{\ref{mad}}Spagna
\item\PTS*{2}\fillin{1cm}{\ref{ate}}Grecia
\columnbreak
\end{parts}
\end{multicols}
\end{problem*}

\begin{problem*}[2ea]
\textit{Vero} o \textit{Falso}. Non è
necessario giustificare le risposte.
\begin{parts}
\item[h] \TF{T} Parigi è la capitale della Francia.
\item[H] \TF{F}  $1+1=3$ .
\end{parts}
\end{problem*}

```

(6^{mi}) 3. Associa a ogni nazione la sua capitale

(a) _____ Francia	A. Atene
(b) _____ Spagna	B. Londra
(c) _____ Grecia	C. Parigi
	D. Vienna
	E. Madrid

(2^{mi}) 4. Vero o Falso. Non è necessario giustificare le risposte.

(a) _____ Parigi è la capitale della Francia.

(b) _____ $1 + 1 = 3$.

Tabella 14. Un esempio di domanda *matching* e di domanda Vero/Falso in *eqexam*

Gli esercizi possibili con il pacchetto *exerquiz* sono:

- esercizi a scelta multipla,
- esercizi a risposta multipla,
- esercizi *fillin* cioè con una risposta breve numerica o testuale,
- problemi in una o più parti.

Gli esercizi sono inseriti in un ambiente *quiz* o *shortquiz* che hanno comportamenti diversi soprattutto nell'interattività, oppure, nel caso di problemi, in un ambiente *exercise*. Le risposte agli esercizi a risposta multipla possono essere mescolate in modo aleatorio, ma non gli esercizi all'interno del *quiz*.

Una delle differenze degli ambienti *quiz* e *shortquiz* è che l'ambiente *quiz* permette di assegnare un punteggio agli esercizi.

L'unicità di *acrotex* è appunto la possibilità di avere esercizi interattivi, cioè la possibilità di avere immediatamente un riscontro alle risposte date, il punteggio ottenuto, la risposta corretta.

Osserviamo che il pacchetto è localizzato in italiano ed è comunque semplice adattarlo alla lingua italiana.

Usando *acrotex*, grazie a javascript, è possibile anche avere interattività ponendo domande a risposta aperta breve: nella versione interattiva allo studente viene presentata una casella in cui scrivere la risposta e il codice provvede al controllo.

Se la domanda prevede come risposta un numero il docente può dare un grado di approssimazione, ma la risposta può essere anche una stringa di testo o una formula. Il codice è in grado di stabilire la correttezza della risposta anche se formalmente diversa: ad esempio riconosce come "uguali" le risposte (testuali) "Dante Alighieri" e "dante alighieri" oppure, in ambito matematico $2x-y$ e $-y+x*2$ o 18 e $36/2$.

Un altro punto interessante è quello di poter dare punteggi negativi alle risposte errate, di mostrare o meno la risposta corretta e se si se mostrarla alla fine del quiz o a ogni domanda, la possibilità di avere un link allo svolgimento completo dell'esercizio, di "contare" il numero di tentativi fatti, di avere un timer e una tabella finale con un riepilogo delle domande a cui si è risposto e delle risposte date.

Ovviamente non vi è interattività per i problemi o "esercizi lunghi".

Oltre a produrre file pdf contenenti esercizi interattivi è possibile mediante il pacchetto `eq2db` avere un file che invia le risposte a un server. Nella versione originale il server doveva essere Microsoft IIS 6, ma è possibile adattare il pacchetto (e gli script che devono risiedere sul server) in modo che possa funzionare con altri linguaggi: lo stesso D.P. Story ha lavorato in tale senso, ma non c'è una documentazione.

Nonostante le enormi potenzialità del pacchetto, il suo utilizzo non è difficile ed è alla portata di tutti.

È necessario sottolineare che l'utilizzo di codice javascript all'interno del file pdf se da un lato permette l'interattività, dall'altro impone che i file vengano necessariamente visualizzati con Adobe Acrobat Reader, che è l'unico lettore in grado di interpretare correttamente tale codice. Mentre quando il pacchetto è stato implementato ciò non era un problema, ora tutti i browser hanno un lettore pdf incorporato e quindi la fruizione degli esercizi online necessita di cambiamenti di configurazione sui pc che vengono usati.

Purtroppo la scomparsa del prof. Story ha lasciato il pacchetto "orfano" e non aggiornato.

4.2. WebQuiz

Il pacchetto `WebQuiz` (MATHAS, 2019) non genera esami o verifiche, ma solo esercizi da utilizzare online. In effetti `WebQuiz` si compone di

- un pacchetto $\LaTeX$,
- un programma in Python3 che usa `TEX4ht` per convertire i file $\LaTeX$ in pagine html
- alcuni file per il web (Javascript, CSS, documentazione).

Poiché `TEX4ht` trasforma quasi tutti i comandi $\LaTeX$ in html non c'è praticamente limite a quello che può essere scritto: disegni, tabelle, grafici, ...

È possibile inserire quattro tipologie di esercizi:

- esercizi a scelta multipla,
- esercizi a risposta multipla (con possibilità di avere risposte parziali),
- esercizi con risposta breve da confrontare con una risposta data (testuale o numerica),
- esercizi a risposta lunga.

Per tutti i tipi di esercizio, tranne l'ultimo, la correzione è automatica, ma a differenza del pacchetto `acrotex` avviene a livello del server. È anche possibile dare allo studente un feedback alla risposta fornita ed è possibile avere gli esercizi in ordine casuale, mentre non è possibile randomizzare l'ordine delle risposte nelle domande a risposta multipla.

Per gli esercizi a risposta breve sono possibili più tipi di confronto con la risposta esatta: numerico intero, numerico con approssimazione, testuale esatto (con maiuscole/minuscole) o no.

La composizione del file $\LaTeX$ è semplicissima con pochi comandi caratteristici del pacchetto e molto intuitivi. Compilando il file con `\pdflatex` si ottiene una versione dove sono

indicate le risposte esatte, compilandolo con il programma Python3 `\webquiz` si ottiene la versione da pubblicare su web.

L'aspetto grafico ottenuto è molto buono, il programma crea un indice degli esercizi e le *breadcrumb*². Il pacchetto è localizzato in italiano.

Non c'è possibilità di assegnare un punteggio agli esercizi e non c'è un timer.

Il programma necessita ovviamente oltre che di una distribuzione $\text{T}_\text{E}\text{X}$, di Python3, di Javascript e di un server web.

Un aspetto interessante che secondo noi va approfondito è la possibilità di poter modificare, rispettando la licenza GPL, il codice Python per ottenere ulteriori caratteristiche.

4.3. Moodle

Il pacchetto *moodle* (GUERQUIN-KERN, 2021) permette di generare file adatti ad essere caricati sulla piattaforma MOODLE³, una piattaforma per l'apprendimento online *open source* e molto diffusa.

In senso stretto, il pacchetto *moodle* non genera quiz: genera banche dati di domande che possono essere importate in MOODLE e il docente deve poi comporre manualmente un quiz dalle banche dati di domande; ci sono però delle funzionalità (i *tag* e le *categorie*) che rendono l'operazione più semplice.

Il pacchetto permette altresì di generare un file pdf annotato con soluzioni per i docenti e un file pdf con il testo dell'esame da sottoporre agli studenti.

Osserviamo che il pacchetto è localizzato in italiano.

In ogni prova d'esame è possibile inserire molte tipologie di esercizio che MOODLE sa gestire⁴:

- Vero/Falso,
- esercizi a scelta multipla,
- esercizi a risposta multipla,
- esercizi a risposta breve (testuale o numerica)
- esercizi *fillin*
- *matching*
- esercizi a risposta lunga

Il pacchetto non supporta nativamente la possibilità di utilizzare formule che vengano poi calcolate al momento della somministrazione del quiz, come, ad esempio, negli esercizi *calculated* in MOODLE, ma utilizzando $\text{Lua}\text{T}_\text{E}\text{X}$ o il pacchetto *python* è possibile estendere le capacità del pacchetto in modo da riprodurre quel tipo di esercizio (si veda ad esempio <https://github.com/avohns/python-latex-moodle-quiz/>).

Per utilizzare il pacchetto al pieno delle sue possibilità è necessario avere a disposizione MOODLE, ma anche i file pdf prodotti possono essere usati, anche se hanno una resa grafica a nostro parere inferiore ad altri pacchetti.

2. La *breadcrumb* è una tecnica di navigazione su siti o programmi che tiene traccia della posizione in cui si trova l'utente. Solitamente è una barra orizzontale posta in alto con i link della "strada" percorsa dall'utente per arrivare alla pagina attuale (si veda <https://it.wikipedia.org/wiki/Breadcrumb>.)

3. Si veda il sito <https://moodle.com/solutions/lms/>

4. Per un elenco completo degli esercizi che si possono usare in MOODLE si può consultare il sito <https://docs.moodle.org/403/en/Questions>

4.4. R/exams

Il pacchetto **R/exams** (ZEILEIS *et al.*, 2022) non è un pacchetto $\LaTeX$, ma un pacchetto del linguaggio di programmazione statistica R (si veda <https://www.r-project.org/>) che utilizza file scritti in $\LaTeX$ o in Markdown per generare esami. Utilizzando il linguaggio R è possibile avere esercizi “dinamici”, cioè contenenti parametri il cui valore cambia ad ogni compilazione. Uno degli aspetti interessanti è che è possibile ottenere molti tipi di output: file pdf attraverso $\LaTeX$, file docx (Word®), file odf (Libre/Openoffice), file html, file per MOODLE, BLACKBOARD, ILIAS, CANVAS ed altri LMS.

I file pdf prodotti possono essere corretti automaticamente mediante la lettura delle prove scannerizzate e una procedura automatizzata.

Ogni esercizio deve essere scritto su un file separato e il pacchetto utilizza R per tutte le routines computazionali: la generazione di valori per i parametri, la scelta aleatoria degli esercizi, la randomizzazione delle risposte nelle domande a scelta o a risposta multipla.

Anche questo pacchetto è localizzato in italiano ed è comunque semplice adattarlo alla lingua italiana, ci sono però alcuni problemi nella correzione automatica usando lingue diverse dall’inglese.

Pur essendo un programma molto potente e interessante non lo tratteremo ulteriormente poiché non è un pacchetto $\LaTeX$.

5. Conclusioni

Rispetto alla nostra analisi di quindici anni fa c’è stato un grande sviluppo di pacchetti per esami e verifiche per scuole superiori, soprattutto per queste ultime, denotando quindi un accresciuto interesse per il mondo $\LaTeX$ da parte degli insegnanti. Inoltre esistono ora parecchi pacchetti che supportano la lingua italiana, sia direttamente che con un lavoro né lungo né difficile e anche questo è un buon segno di interesse che non può che fare piacere al mondo $\LaTeX$ italiano.

Vogliamo anche sottolineare lo sforzo per avere strumenti che permettano l’utilizzo di $\LaTeX$ per somministrare esami online, ovviamente utilizzando anche altri programmi specifici.

Riferimenti bibliografici

ALEXANDER, Jason (2006). «The package examdesign». CTAN:/macros/latex/contrib/examdesign/examdesign.pdf.

DOSSENA, Riccardo (2020). «La classe verifica». CTAN:/macros/latex/contrib/verifica/verifica.pdf.

GRAY, Norman (2022). «exam-n: exam papers». CTAN:/macros/latex/contrib/exam-n/exam-n.pdf.

GUERQUIN-KERN, Matthieu (2021). «The moodle package: generating moodle quizzes via $\LaTeX$ ». CTAN:/macros/latex/contrib/moodle/moodle.pdf.

HIRSCHHORN, Philip (2023). «Using the exam document class». CTAN:/macros/latex/contrib/exam/examdoc.pdf.

- LYU, Jianrui (2024). «The randexam class for $\LaTeX$ ». CTAN: /macros/latex/contrib/randexam/randexam.pdf.
- MATHAS, Andrew (2019). «Webquiz: A latex package for writing online quizzes». CTAN: /macros/latex/contrib/webquiz/doc/webquiz.pdf.
- MCCLAINE, Christopher (2023a). «The counterz package». CTAN: /macros/latex/contrib/counterz/counterz.pdf.
- (2023b). «The examz class». CTAN: /macros/latex/contrib/examz/examz.pdf.
- MESSINEO, Grazia e Salvatore VASSALLO (2012). «Il pacchetto esami per la creazione di prove scritte». *Ars $\TeX$ nica*, (14), pp. 95–103. <http://www.guitex.org/home/numero-14>.
- (2023). «Il pacchetto esami». CTAN: /macros/latex/contrib/esami/doc/esami-doc-it.pdf.
- OP DEN BROUW, Jesse (2021). «The exam-randomizechoices package». CTAN: /macros/latex/contrib/exam-randomizechoices/exam-randomizechoices-doc.pdf.
- PILU, Antonello (2011). «La creazione di una prova d’esame». *Ars $\TeX$ nica*, (11), pp. 5–14. <http://www.guitex.org/home/it/numero-11>.
- RACCANELLO, Francesco (2024). «Il pacchetto verifiche». CTAN: /macros/latex/contrib/verifiche/verifiche.pdf.
- STORY, D. P. (2016). «The eq2db package». CTAN: /macros/latex/contrib/acrotex/doc/aeb_man.pdf.
- (2021a). «Acro $\TeX$ ». <https://www.math.uakron.edu/~dpstory/acrotex.html>.
- (2021b). «Support for AcroForms and Links, and for Document JavaScript and Open Page Events». CTAN: /macros/latex/contrib/acrotex/doc/eformman.pdf.
- (2021c). «The eqexam Package part of the Acro $\TeX$ eDucation Bundle». CTAN: /macros/latex/contrib/acrotex/doc/eqexamman.pdf.
- (2021d). «The Acro $\TeX$ eDucation Bundle (AeB)». CTAN: /macros/latex/contrib/acrotex/doc/aeb_man.pdf.
- TALBOT, Nicola L.C. (2017). «probsoln v3.05: creating problem sheets optionally with solutions». CTAN: /macros/latex/contrib/probsoln/probsoln.pdf.
- ZEILEIS, Achim, Bettina GRÜN, Friedrich LEISCH e Nikolaus UMLAUF (2022). «R/exams». <https://www.r-exams.org/>.

Grazia Messineo

UNIVERSITÀ CATTOLICA MILANO – IIS “FALCONE-RIGHI” CORSICO
graziadotmessineoatunicattdotit

Salvatore Vassallo

UNIVERSITÀ CATTOLICA MILANO
salvatoredotvassalloatunicattdotit

Composizione di atti giudiziari utilizzando $\LaTeX$

Federico Gallo

Sommario L'utilizzo di $\LaTeX$ per la redazione di atti giudiziari non presenta particolari difficoltà e può rivelarsi utile, sia per la resa tipografica, che per il risparmio di tempo che garantisce dopo aver costruito un modello adeguato. Dal 1° settembre 2023 è in vigore il Decreto del Ministero della Giustizia che impone il rispetto di alcune regole tipografiche negli atti del processo civile. Oggi, ci si aspetta che un atto giudiziario contenga riferimenti incrociati, il sommario e link diretti ai documenti esterni che vengono richiamati nel testo. $\LaTeX$, grazie alle sue caratteristiche, garantisce un risultato ottimale in questo senso. Eppure, gli avvocati e i giuristi che utilizzano $\LaTeX$ sono un numero davvero esiguo. In questo articolo propongo un approccio alla redazione degli atti giudiziari che permetta di ottenere un risultato tipografico ottimo, anche a chi non abbia una conoscenza approfondita del programma.

Abstract The use of $\LaTeX$ for drafting legal documents does not present any particular difficulty and can prove useful, both for its typographic output and for the time-saving it ensures after creating an adequate template. Since September 1, 2023, a Decree from the Ministry of Justice has been in effect, mandating adherence to certain typographic rules for lawyers. Today, it is expected that a legal document includes cross-references within the text, an abstract, and direct links to external documents cited in the text. $\LaTeX$, with its capabilities, guarantees an optimal result in this respect. Yet, the number of lawyers and jurists using $\LaTeX$ is remarkably small. In this article, I propose an approach to drafting legal documents that allows for achieving an excellent typographic result, even for those without an in-depth knowledge of the program.

1. Introduzione

Con l'entrata in vigore del Decreto del Ministero della Giustizia in tema di redazione degli atti nel processo civile, oggi anche gli avvocati devono preoccuparsi della tipografia degli atti ¹.

La capacità di redigere un documento che abbia una resa tipografica efficace è diventata un'abilità indispensabile anche per l'avvocato. Inespugnabilmente, durante il percorso universitario, l'aspirante giurista non riceve alcuna formazione tecnica sull'utilizzo dei software per la produzione di testi. Salvo rari casi, dunque, è un autodidatta che utilizza il programma di video scrittura come una macchina da scrivere. Non sorprende, quindi, la poca propensione della categoria ad avvicinarsi a uno strumento informatico che preveda la compilazione di un codice, senza restituire immediatamente il risultato tipografico visibile sullo schermo. Del resto, anche chi utilizza in modo adeguato gli strumenti dei software *wysiwyg*², non trova particolari ragioni per imparare qualcosa che, a prima vista, non garantisce nulla di più, rispetto a quanto sa già fare³.

1. Per la lettura del decreto: GAZZETTA (2023). Per maggiori informazioni sul contenuto del Decreto Ministeriale è possibile consultare i seguenti articoli on line: OCCHIPINTI (2023), BIARELLA (2023).
2. L'acronimo significa "What you see is what you get" che, in italiano, significa "Quello che vedi è ciò che ottieni". I classici programmi che operano in questa modalità sono, ad esempio, MS Word, LibreOffice, OpenOffice e Pages.
3. Senza considerare che oggi esistono anche per $\LaTeX$ strumenti che permettono di vedere in tempo reale la resa delle modifiche al codice, come nel caso del sito Overleaf.

Non si faranno paragoni tra $\LaTeX$ e gli strumenti di scrittura WYSIWYG; esistono già molti articoli che analizzano la questione, giungendo a conclusioni diverse⁴.

A mio avviso, $\LaTeX$ rappresenta una concreta opportunità per ottimizzare la scrittura degli atti processuali, standardizzando la resa tipografica e la loro coerenza. Con questo contributo, vorrei proporre un approccio metodologico che prevede la possibilità di concentrarsi solo sul contenuto dell'atto, che è poi la parte creativa, importante della difesa tecnica, avendo la certezza della resa tipografica finale di grande qualità, grazie ad un lavoro preliminare di configurazione, del quale poi l'autore non deve più preoccuparsi.

2. La visione d'insieme

$\LaTeX$ obbliga chi scrive ad avere una visione d'insieme della struttura dell'atto. Che si tratti di una relazione o di un manuale, l'autore è costretto ad avere ben chiara la struttura del testo che andrà a scrivere. Questa caratteristica è insita in un atto processuale di qualunque tipo, nel quale potremmo prevedere di trovare: un'intestazione, con l'indicazione dell'ufficio processuale, l'indicazione della tipologia di atto (atto di citazione, ricorso, memoria, etc.), l'identificazione dei soggetti coinvolti nella controversia, una parte in "fatto", seguita da una parte in "diritto", poi una *vocatio in ius*, le conclusioni, le istanze istruttorie, i documenti da produrre e le dichiarazioni finali.

L'atto processuale, quindi, si presta perfettamente ad essere inquadrato in un *foglio di stile*, composto da una serie di pacchetti e di comandi che ne determinino a priori la struttura e la resa tipografica. Tale struttura potrà variare a seconda che si tratti di un atto di citazione, rispetto ad una comparsa conclusionale, ma resterà sempre la stessa per gli atti futuri del medesimo tipo.

Cambieranno invece alcune parti variabili (ad esempio, l'autorità giudiziaria competente, il nome delle parti, il valore della controversia, etc.) che saranno indicate in un altro file, a parte. Va considerato, in ogni caso, che queste variabili non cambieranno nel corso del medesimo processo e, dunque, anche questo file andrà completato una sola volta, per poi essere utilizzato in tutti gli atti della medesima controversia giudiziale.

L'unica parte da cambiare realmente è il contenuto creativo dell'atto, rappresentato dalle argomentazioni giuridiche che l'avvocato scriverà di volta in volta e che saranno diverse per ciascun atto processuale che depositerà. Queste argomentazioni saranno scritte in un terzo file, che sarà importato nel foglio di stile.

Prendiamo, ad esempio, il caso di un processo civile da instaurare presso il Tribunale di Roma. In questo caso potremo, ragionevolmente, prevedere la stesura di almeno cinque atti, riferiti alla stessa controversia di primo grado:

- un atto di citazione (oppure una comparsa di costituzione e risposta);
- tre memorie endo-processuali;
- una comparsa conclusionale;
- una memoria conclusionale di replica.

4. On line si trovano molti contributi, come, ad esempio LODE (2022), LUU (2023). Consiglio anche la lettura di PANTIERI e GORDINI (2017)

Utilizzando ℒ_{TeX} si può progettare la scrittura, utilizzando i tre file a cui si è accennato sopra.

1. Un file nominato, ad esempio, “Atto di citazione”, che contiene tutte le informazioni di stile, valide per tutti gli atti citazione dello studio e che, tramite il comando `\input`, inserirà al posto giusto gli altri due file.
 - (a) Un file contenente tutti gli elementi fissi dello specifico processo denominato “Dati”: nome del tribunale, nome delle parti, nome degli avvocati, valore della controversia, eventuali estremi del provvedimento impugnato, etc.
 - (b) Un file contenente il testo dell’atto con le argomentazioni giuridiche, denominato “Testo”.
2. All’interno della stessa cartella saranno inseriti anche i file, riferiti ai documenti da produrre in giudizio (fotografie, PDF, e-mail in formato .eml o .zip), richiamati all’interno dell’atto, ai quali, si potrà prevedere il rinvio tramite un link.

Una volta compilato il file “Atto di citazione”, l’atto sarà completo, con l’indice interattivo, i richiami all’interno dell’atto e i link ai documenti esterni. L’aspetto davvero interessante, a mio avviso, è rappresentato dal fatto che il file “Atto di citazione” è un file di stile che viene preparato una volta soltanto e sarà sempre uguale per tutti gli atti di citazione che si redigeranno negli anni futuri; il file “Dati” sarà redatto solo una volta, all’inizio della causa, con i nomi delle parti, quelli dei loro avvocati, etc. Questo file potrà essere completato con i dati necessari anche dalla segreteria dello studio, dal momento che non necessita di particolari competenze giuridiche o tecniche e potrà essere utilizzato in tutti gli atti scritti per quella medesima causa.

Poiché la resa grafica dell’atto è già predisposta e i dati fissi sono già presenti, l’avvocato si troverà a redigere solo la parte “importante”, costituita dalle ragioni di fatto e di diritto in favore del proprio assistito, le conclusioni nel merito e in via istruttoria, etc., che scriverà nel file “testo”.

Vediamo, dunque, nel dettaglio un esempio di come redigere il file di stile dell’atto di citazione.

3. Il file di stile

Prendiamo come esempio il seguente codice del “foglio di stile” che chiamiamo “Atto di citazione”

```
\documentclass[a4paper,12pt,twoside]{article}
\usepackage[margin=2.5cm]{geometry}
\usepackage[\onehalfspace]{setspace}
```

oltre ai consueti pacchetti:

```
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[italian]{babel}
```

In questa prima parte vengono forniti i parametri coerenti con il Decreto Ministeriale, che prevede un font di grandezza 12pt, tutti i margini della pagina di 2,5 cm e un avanzamento di riga pari a 1,5 volte il corpo del font in uso.

Vanno, poi, inseriti i pacchetti utili a migliorare la resa tipografica del testo. Tra i vari, segnalo:

```
\usepackage{microtype}
\microtypecontext{spacing=italian}
```

Per eliminare le legature tra lettere e rendere l'atto processuale idoneo alla normativa del processo telematico:

```
\disableligatures{}
```

Per il simbolo dell'euro:

```
\usepackage[official]{eurosym}
```

Saranno caricati altri pacchetti utili, ad esempio, per la gestione delle citazioni (`csquotes`), per i riferimenti incrociati (`varioref`), per la codifica ISO 19500-1 PDF/A-1b (`pdfx`) e per la gestione dei collegamenti ipertestuali (`hyperref`).

Nel foglio di stile "Atto di citazione" vengono caricati i pacchetti e vengono definiti i comandi che di fatto preparano l'atto ad essere stampato e, una volta completato, non ci si pensa più. Potrebbero esserci altre parti fisse del documento da inserire in questo primo file: ad esempio, io amo inserire a piè di pagina un link che riporti il lettore all'indice iniziale dell'atto:

```
\pagestyle{fancy}
\fancyhead{}
\renewcommand{\headrulewidth}{0pt}
\fancyfoot[R]{\hyperlink{index}{Torna all'indice}}
```

A questo punto è necessario fornire il comando che importa il file contenente i dati delle variabili dell'atto quali autorità, nome delle parti, etc. (La formulazione `Dati/Dati.tex` = `Cartella/File.tex`)

```
\input{Dati/Dati.tex}
```

Il codice che segue non è altro che il posizionamento delle variabili contenute nel file "Dati" nel punto giusto della struttura dell'atto.

```
\section*{\Large\uline\trib}
\section*{\Large\titolo}
\vspace{0.5cm}
\attorecompl, rappresentata e difesa per procura in calce
al presente atto dall'\avv.\
\begin{center}
\textbf{contro}\
\end{center}
\convcompl
```

Ora inseriamo l'indice.

```
\tableofcontents \label{Indice}
```

Infine l'importazione del file "Testo" contenente il contenuto in fatto e diritto dell'atto e le dichiarazioni finali dell'atto. Faccio notare che anche il valore della causa e i riferimenti dell'avvocato sono inseriti nel file "Dati".

```
\input{Testo/Testo.tex}
\vspace{-0.6cm}
\section[Dichiarazioni finali]{}
Ai sensi dell'art. 9 L. 488/99 e successive modificazioni
ed integrazioni, si dichiara che il valore della
presente controversia \valorecontroversia.\newline
Ai sensi degli art. 133, 134, 170 e 176 c.p.c. il
sottoscritto difensore dichiara di voler ricevere le
comunicazioni relative al presente procedimento a \recap
.\newline
Roma, \makeatletter
\@date.

\hspace{8cm}\@author
\end{document}
%compilare documento due volte per vedere l'indice
correttamente compilato.
```

In questa prima fase, compilando il codice non si otterrebbe nulla, perché mancano gli altri i due file: "Dati", con l'indicazione delle parti fisse del processo e "Testo" con il corpo dell'atto.

Ad esempio:

```
\section*{\Large\uline\trib}
```

prevede che sia stata definita la variabile \trib con il nome del tribunale.

Oppure:

```
\section*{\large\titolo}
```

prevede che sia stata definita la variabile \titolo con il tipo di atto (atto di citazione, ricorso) e così via per tutti i comandi inseriti nel file di stile riferiti alle variabili dichiarate nel file "Dati".

4. Il file "Dati"

Una volta preparata la struttura dell'atto sarà possibile concentrarsi sui due file che lo completano. Il primo file, "Dati", come detto, contiene le variabili riferite al procedimento giudiziario specifico. Anche in questo caso, dunque, prepariamo un modello, che poi completeremo di volta in volta, specificando il contenuto delle variabili. Il file è piuttosto semplice: creiamo le variabili di testo

```
\newcommand{\trib}{TRIBUNALE DI ROMA}
```

```
\newcommand{\titolo}{ATTO DI CITAZIONE}
```

```
\newcommand{\attorecompl}{\textbf{Alfa Srl}, in persona del  
legale rappresentante, con sede legale in \dots, c.f. \  
dots, P.Iva \dots}\label{attorecompleto}
```

```
\newcommand{\attore}{Alfa Srl}\label{attorebreve}
```

La variabile “attorecompl” si distingue dalla variabile “attore” per il fatto che la prima contiene tutti i dati dell’attore, mentre la seconda soltanto la denominazione. Normalmente, la variabile “attorecompl” si utilizza soltanto all’inizio dell’atto, mentre nella parte narrativa si utilizza solo la denominazione. Lo stesso vale per il convenuto.

```
\newcommand{\convcompl}{\textbf{Beta Srl}, in persona del  
legale rappresentante, con sede legale in \dots, c.f. \  
dots, P.Iva \dots}\label{convenutocompleto}
```

```
\newcommand{\conv}{Beta Srl}\label{convenutobreve}
```

Servono poi i dati specifici della controversia, quali la data dell’udienza e il valore della controversia:

```
\newcommand{\dataud}{25.12.2030}\label{dataudienza}
```

```
\newcommand{\valorecontroversia}{pari a ...}
```

Infine i dati dell’avvocato:

```
\newcommand{\avv}{avv. Pinco Pallino, c.f. \dots con studio  
in Atlantide, via Dei Mari, 3, C.F. \dots, fax \dots, P  
.E.C. pinco.pallino@ordineavvocatiatlantide.eu}\label{  
studiolegale}
```

```
\newcommand{\recap}{fax~\dots, P.E.C. pinco.  
pallino@ordineavvocatiatlantide.eu}\label{  
recapitistudio}
```

5. Il file di testo

Ciascuno sarà libero di inserire le variabili utili al proprio lavoro, consapevole del fatto che sia la struttura del file di stile, che quella del testo della difesa saranno agevolate e facilitate. Se, infatti, ogni volta che dovrò citare il nome del convenuto sarà sufficiente scrivere \conv senza preoccuparsi di eventuali errori di battitura, la parte creativa della scrittura ne sarà agevolata. Non solo, ma se avessi atti ripetitivi da redigere, come ad esempio un contratto standard, cambiando il solo file “dati”, il nuovo atto sarà pronto in pochi minuti. Nel file di testo dunque l’avvocato scriverà il contenuto dell’atto diviso in \section, \subsection, etc. Infine, compilando il file di stile “Atto di citazione” apparirà un atto simile a quello illustrato nella figura 1.

Avv. PINCO PALLINO
Tel. 000.1112222 | Fax 000.5334455
Via Dei Mari, 3 - 33555 ATLANTIDE
pincopallino@ordineavvocatiatlantide.it

TRIBUNALE DI ROMA

ATTO DI CITAZIONE

La società **Alfa Srl**, in persona del legale rappresentante, con sede legale in ..., c.f. ..., P.Iva ..., rappresentata e difesa per procura in calce al presente atto dall'avv. Pinco Pallino, c.f. ... con studio in Atlantide, via Dei Mari, 3, C.F. ..., fax ..., P.E.C. pinco.pallino@ordineavvocatiatlantide.eu

control

Beta Srl, in persona del legale rappresentante, con sede legale in ..., c.f. ..., P.Iva ...

Indice

Fatto	1
Diritto	2
Prima contestazione in diritto	2
Seconda contestazione in diritto	2
Terza contestazione in diritto	2
Conclusioni	3
Nel merito:	3
Nel merito in via subordinata:	4
In via istruttoria:	4
Documenti depositati	4
Dichiarazioni finali	

FATTO

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut parus elit, vestibulum ut, placerat ac, adipiscing dolor, vitae. Curabitur dictum gravida mauris. Nam arcu elit, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames a turpis egestas. Mauris ut lac. Cras viverra metus enim sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus ut tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

1

comparire dinanzi al TRIBUNALE DI ROMA, sezione e Giudice istruttore designato, all'udienza del 25.12.2010, invitandolo a costituirsi e a contestare specificamente gli assunti e le domande attore nel termine di settanta giorni prima di tale scadenza, con avvertimento che la mancata costituzione o la costituzione oltre i suddetti termini comporterà la decadenza di cui agli artt. 38 e 167 c.p.c. e che la difesa tecnica mediante avvocato è obbligatoria in tutti i giudizi davanti al Tribunale, fatta eccezione per i casi previsti dall'articolo 86 c.p.c. o da leggi speciali, e che esso convenuto, sussistendone i presupposti di legge, può presentare istanza per l'ammissione al patrocinio a spese dello Stato, per ivi sentire accogliere, anche in sua dichiaranda confutazione, le seguenti

CONCLUSIONI

Nel merito: voglia l'On.le Tribunale adito, previa ogni più utile declaratoria del caso o di legge, ogni diversa e contraria istanza ed eccezione disattesa anche in via istruttoria ed incidentale:

- accertare e dichiarare ...;

- per l'effetto condannare Beta Srial pagamento della somma di €..., oltre interessi legali dalla data ..., fino alla data dell'effettivo soddisfo;
- con vittoria di spese e compensi professionali del presente giudizio, oltre rimborso spese generali, cpa e Iva come per legge.

Nel merito in via subordinata: voglia l'On.le Tribunale adito, previa ogni più utile declaratoria del caso o di legge, ogni diversa e contraria istanza ed eccezione disattesa anche in via istruttoria ed incidentale:

- accertare e dichiarare ...;
- per l'effetto condannare Beta Srial pagamento della somma di €..., oltre interessi legali dalla data ..., fino alla data dell'effettivo soddisfo;

- con vittoria di spese e compensi professionali del presente giudizio, oltre rimborso spese generali, cpa e Iva come per legge.

In via istruttoria:

- si chiede ammettersi prova per testì su seguenti capitoli di prova:

- 1) vero che Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum.

- 2) vero che Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla.

3

[Torna all'indice](#)

DIRITTO

Prima contestazione in diritto

Fuscus maris. Vestibulum lacus nibbi at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu nisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempore ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Seconda contestazione in diritto

Nullam eleifend justo in nisl. In hac habitasse platea dictumst. Morbi nonummy. Aliquam ut felis. In velit leo, dictum vitae, posuere id, vulpate nec, ante. Maecenas vitae pede nec dui dignissim suscipit. Morbi magna. Vestibulum id purus eget velit laoreet laoreet. Praesent sed leo vel nibh convallis blandit. Ut rutrum. Donec nibh. Donec interdum. Fusce sed pede sit amet elit rhoncus ultrices. Nullam at enim vitae pede vehicula laculis.

Terza contestazione in diritto

Etiam suscipit aliquam arcu. Aliquam sit amet est ac purus bibendum congue. Sed in eros. Morbi non orci. Pellentesque mattis lacinia elit. Fusce molestie velit in ligula. Nullam et orci vitae nunc vulputate auctor. Aliquam eget purus. Nulla auctor wisi id ipsum. Morbi porttitor tellus ac enim. Fusce ornare. Proin ipsum enim, tincidunt in, ornare venenatis, molestie a, augue. Donec vel pede in lacus sagittis porta. Sed hendrerit ipsum quis nisi. Suspendisse quis massa ac nibh pretium congue. Sed sodales. Nam eu neque quis pede dignissim ornare. Maecenas eu purus ac urna tincidunt congue.

Alla luce di tutto quanto esposto, la società Alfa Srl, in persona del legale rappresentante pro tempore, *ut supra* rappresentata e difesa,

CITA

la società **Beta Srl**, in persona del legale rappresentante, con sede legale in ..., c.f. ..., P.Iva ... a

2

[Torna all'indice](#)

- 3) vero che Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam.
- 4) vero che Duis eget orci sit amet orci dignissim rutrum.
- 5) vero che Pelentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem.
- Si indicano quali **testi**: Anco Marzio, Tarquinio Prisco.

Si depositano i seguenti documenti

- 1) Documento 1
- 2) Documento 2
- 3) Documento 3

L'attore dichiara che, ai sensi dell'art. 163, co. 3 n. 3-bis, la domanda proposta con il presente atto è soggetta alla condizione di procedibilità prevista dall'art. 5 D.Lgs. n. 28/2010 (ovvero dall'art. 3 del decreto-legge 12 settembre 2014, n. 132) e che tale condizione è stata soddisfatta in quanto”

Ai sensi dell'art. 9 L. 488/99 e successive modificazioni ed integrazioni, si dichiara che il valore della presente controversia è pari a €...

Al sensi degli artt. 133, 134, 170 e 176 c.p.c. il sottoscritto difensore dichiara di voler ricevere le comunicazioni relative al presente procedimento a fax ..., P.E.C. pinco.pallino@avvocatiatl.eu.
Roma, lì

Firma Avvocato

4

[Torna all'indice](#)

Figura 1. Un esempio di “Atto di citazione”.

Riferimenti bibliografici

- BIARELLA, Laura (2023). «Atti giudiziari: il decreto del ministero delinea il format». *Diritto.it*. <https://www.diritto.it/atti-giudiziari-decreto-ministero-delinea-format/>.
- GAZZETTA, Ufficiale (2023). «Regolamento per la definizione dei criteri di redazione, dei limiti e degli schemi informatici degli atti giudiziari con la strutturazione dei campi necessari per l’inserimento delle informazioni nei registri del processo, ai sensi dell’articolo 46 delle disposizioni per l’attuazione del codice di procedura civile.» *Gazzetta Ufficiale*, (187). <https://www.gazzettaufficiale.it/eli/gu/2023/08/11/187/sg/pdf>. Pag. 61.
- LODE, Clemence (2022). «Comparing word and latex». *Lode Publishing*. <https://www.lode.de/blog/comparing-word-and-latex>.
- LUU, Quang Trung (2023). «Latex vs. word: Main differences». *Baeldung*. <https://www.baeldung.com/cs/latex-vs-word-main-differences>.
- OCCHIPINTI, Sara (2023). «Atti giudiziari, dal 26 agosto in vigore il nuovo Dm». *Altalex*. <https://www.altalex.com/documents/news/2023/08/25/atti-giudiziari-dal-26-agosto-in-vigore-il-nuovo-dm>.
- PANTIERI, Lorenzo e Tommaso GORDINI (2017). «L’arte di scrivere con $\LaTeX$ ». http://www.lorenzopantieri.net/LaTeX_files/ArteLaTeX.pdf. Pag. 17 e 18.

Federico Gallo
VIA GERMANIA, 8 — PADOVA
info@federicogallo.it

Framed material for humanists

Claudio Beccari

Abstract Humanists sometimes need to emphasise some small pieces of text or small images so that it is possible to comment them in the proper way; they can use the `wrapfig2` package that has been recently upgraded in order to be compatible with the modifications of the `picture` environment that accept coordinates in absolute dimensions, instead as multipliers of `\unitlength` as it has been normal since $\LaTeX$ was made available in 1984.

Sommario Gli umanisti talvolta necessitano di evidenziare dei brevi brani di testo o piccole immagini in modo che sia possibile commentarli adeguatamente; essi possono usare il pacchetto `wrapfig2`, che recentemente è stato reso compatibile con l'ambiente `picture` che accetta coordinate definite in modo assoluto, invece che come coefficienti di `\unitlength`, come è stato normale fin da quando nel 1984 il linguaggio $\LaTeX$ è diventato disponibile.

1. Introduction

Recently `wrapfig2` has been upgraded in order to avoid the use of package `curve2e` that does not accept the use of coordinates as physical quantities, instead of plain rational numbers, used as coefficients of the `\unitlength` unit of measure. That was normal with the previous versions of the `picture` updated with the extension `pict2e` now updated in 2020 so as to be compatible with the LaTeX Team innovations concerning the official $\LaTeX$ tools.

We think that now humanists can exploit the new `wrapfig2` functionalities for their documents that, more often than not, need to strongly emphasise some phrases or short paragraphs, or small images in a comfortable way.

Here we show how to use the `wraptext` environment and the `\includeframedtext` command.

2. The `wraptext` environment

The `wraptext` environment inserts a small framed text into the normal text lines that are shaped so that the block containing the framed text is conveniently wrapped, even if this wrapping involves more than a single paragraph.

Κα'γὼ σὲ πατάξας διαλύσω τὸ κρανίον

Text 1. A short phrase from Lucianus

`wrapfigure` and `wraptable` environments, while the latter package adds the `wraptext` environment with a similar philosophy. Users can read the `wrapfig2` documentation by means of

This wrapping sometimes is not successful, because of some idiosyncrasies due to the `wrapfig` package by Donald Arseneau; `wrapfig2` is a fork of Arseneau's one. The difference is that the former package defines only the

the terminal command `texdoc wrapfig2`, since this new package is part of any $\TeX$ system installation.

A small sample of use of this package is in the internal page 73 side; it is also possible to insert the block in the external side, as well to let the text protrude in the margin; as a personal taste we prefer to avoid any protrusion. Wrapped text may be declared to be a floating object. More details are available in the package documentation.

The code for the above example 1 is the following.

```
\begin{wraptext}{i}{0.5\linewidth}
\includeframedtext{%
\foreignlanguage{greek}{Κα'γὼ σὲ πατάξας διαλύσω τὸ κρανίον}}
\caption{A short phrase from Lucianus}\label{fig:greek}
\end{wraptext}
```

We call the reader's attention on the command `\includeframedtext` that allows also the use of several options expressed with the *key=value* syntax in order to specify up to nine setting options.

3. The `\includeframedtext` command

The `\includeframedtext` command is the one that actually inserts its content into the output PDF file. This command may be used even outside the *wraptext* environment, and, although it is not necessary, it might be used even within the other two environments *wrapfigure* and *wratable*. But this is the command that may receive the *key=value* settings; in facts its syntax is the following:

```
\includeframedtext{<text to frame>}[<key=value settings>]
```

where the *<key=value settings>* are the following (listed in alphabetical order, but they can be used in any order).

`backgroundcolor` sets the background colour among those defined by the default set provided by package `xcolor`. The default colour is light grey.

`fboxrule` sets the thickness of the frame; a zero value is allowed, otherwise it should not be smaller than 0.4pt; on the opposite it should not be set too large and 1mm appears as a thick enough frame around the wrapped text.

`fboxsep` sets the distance of the frame from the wrapped text; by default it is set to 1ex; also in this case it is better to avoid exaggerations. Notice that the default value depends on the wrapped text font x-height.

`fontstyle` sets any available *declaration*¹ that changes the characteristics of a font: size, series, shape; it is possible to use also the `\usefont` command with all its four arguments, even

1. Remember the difference between *command* and *declaration*; the former either prints a default string, or operates only on its arguments and its effect stops immediately; the latter is a macro that may have arguments, but operates on its possible arguments, and its effects last for a whole group or for the whole document. If `\itshape` is specified, it is a declaration, while `\textit{<text>}` is a command that operates only on its argument *<text>*.

the font encoding. The `wrapfig2` package has available also the `\setfontsize` command that can select any size with any font that has available at least a stepwise continuous size set; for example, the Latin Modern fonts have a stepwise continuous size set, while Computer Modern have available only a discrete size set.

`framecolor` sets the color of the frame; the colours available are those available with package `xcolor` to which no options have been specified; see its documentation and in case load `xcolor` with the desired options before this package `wrapfig2`. The default colour is a very dark grey.

`insertionwidth` sets the insertion width; as it was previously specified, if this width is too small or too large it will be automatically reassigned a value within the allowed range.

`radius` sets the optional radius of the frame rounded “corners”; if it is not specified, such radius is equal to the default value of `\fboxsep`.

`scalefactor` sets the value that establishes a reasonable range for the insertion width; users can specify any value in the range $x y_0 = y_{\min} \leq y \leq y_{\max} = y_0 / x$, where y_0 is the default value, and x is the scaling factor that by default equals 0.8; this means that if y_0 equals half the current measure, the inserted wrapped text produces an indentation of the wrapping lines approximately between 60% and 40% of the current measure; the wrapped text should never have too short a measure and the wrapping indented lines never have too short a measure. If users specify a different value to this key, they might get problems with inter word spacing and with hyphenation.

`textcolor` sets the text colour among those available with the default set provided by package `xcolor`. The default colour is black.

All settable quantities have default values, specified in the above list. The example is in figure 1.

All this means that a single command can do several different things on a given text, depending on the various combinations set up by the user. It may be recalled that the object to be wrapped may be anything, not only text; if instead of a `\text{}` a suitable image is entered, the command provides to set everything necessary to frame everything; by specifying a zero frame thickness, the object is not framed and if the background color is set to white, the object to frame is not emphasised at all.

The code for typesetting figure 1 is the following.

```
\begin{figure}
% first line
\makebox[\textwidth]{\includeframedtext{Text}%
  [insertionwidth=0.45\linewidth]
\hfill
\includeframedtext{Text}%
  [insertionwidth=0.45\linewidth, fboxrule=1mm]}\[2ex]
% Second line
\makebox[\textwidth]{\includeframedtext{Text}%
  [insertionwidth=0.45\linewidth, textcolor=red, fboxrule=2pt]
\hfill
\includeframedtext{Text}%
```

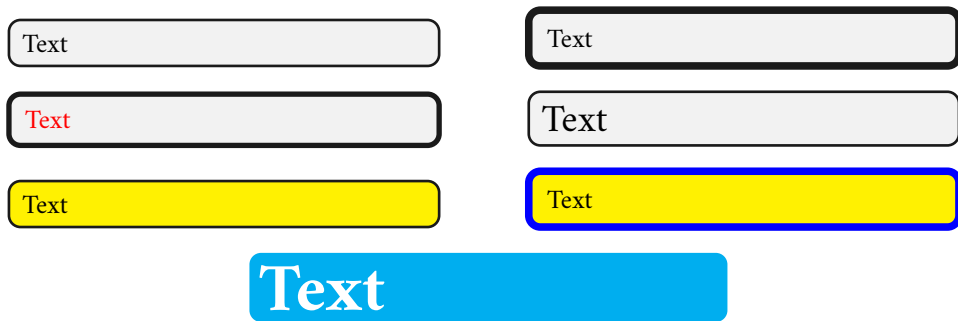


Figure 1. Some framed text boxes with different dimensional parameters, different font size, and different colours.

```
[insertionwidth=0.45\linewidth,fontstyle=\Large]]\[[2ex]
% Third line
\makebox[\textwidth]{\includeframedtext{Text}%
  [insertionwidth=0.45\linewidth,backgroundcolor=yellow]
\hfill
\includeframedtext{Text}%
  [insertionwidth=0.45\linewidth, framecolor=blue,
  backgroundcolor=yellow, fboxrule=1mm]]\[[2ex]
% Fourth line
\makebox[\textwidth]{\includeframedtext{Text}%
  [insertionwidth=0.5\linewidth, fboxrule=0pt,
  backgroundcolor=cyan, textcolor=white,
  fontstyle=\Huge\bfseries]}

\caption{Some framed text boxes with different dimensional
  parameters, different font size, and different colours}\label{fig:framed text}
\end{figure}
```

Notice the all framed texts have a width of `0.45\linewidth`, so that, where there are two framed texts in a line, there is enough space between them to fill an `\hfill` so as to push them at the line extrema.

Notice that the default background color is a very light grey one, the default setting, that discreetly emphasises the wrapped text; nevertheless if the frame thickness was too large, it would become an exaggeration.

In the second line a framed text is coloured in red, while the other frame contains the same text in `\Large` size.

In the third line both frames have a yellow background and both have a blue frame; may be the yellow frame, although being a light colour, is too much, but both contain contrasting colours that match well. If, for example, those colours were a blue frame and a dark brown background, there would not be enough contrast.

The fourth line displays a white text on a cyan background, but there is no frame. The result is fine, but it might be used for other purposes than emphasising a test to be commented, for example, in a critical edition. May be it is good for a title, but certainly not for a wrapped text.



Figure 2. Two framed images. On the left the Todi stela written in Gallic and Latin;Gregorian Etruscan Museum in Rome. On the right a color wheel from Wikipedia.

The curved frame corners have a visible radius change that matches the size of the used font. On the opposite, in figure 2 there is no text, therefore the radius of the corner arcs has been specified.

Speaking of contrasting colours it is possible to choose from the colour wheel,² in figure 2. Contrasting colours are at the end points of each diameter. In any case colours can be selected even by using the “colour expressions” of package `xcolor`.

As an example of framing an object that is not a real *<text>* see figure 2, wich is typeset with the following code.

```
\begin{figure}
\makebox{\textwidth}{%
%
\includeframedtext%
  {\includegraphics[width=\hsize]{stela-todi-small}}%
  [fboxsep=1em, fboxrule=1mm, framecolor=blue,
   backgroundcolor=white,radius=1em,
   insertionwidth=0.45\linewidth]
%
\hfill
%
\includeframedtext}%
  {\includegraphics[\width=\hsize]{ColorWheel}}
[fboxsep=1em, fboxrule=1mm, framecolor=blue,
 backgroundcolor=white,radius=1em,
 insertionwidth=0.45\linewidth]
\caption{Two framed images. On the left the Todi stela written
```

2. From Wikipedia.

```

in Gallic and Latin;Gregorian Etruscan Museum in Rome. On the
right a color wheel from Wikipedia}
\label{txt:todi-stela}\end{figure}

```

4. Conclusion

The possibility of framing small texts, single words, or short phrases, even short paragraphs, may be useful even for humanists; as a matter of fact the whole `wrapfig2` package was assembled explicitly on request of a humanist who wanted to emphasise some short phrases.

The original `wrapfig` package by Donald Arseneau did not contain anything that could frame what was wrapped; Arseneau had already developed another package, `framed`, but his package created just a frame with right angle corners.

On `tex.stackexchange` another user asked for a similar solution and he was pleased to use the framed coloured box of package `tcolorbox`. But the former humanist who asked us a simpler solution, could not use `tcolorbox` that is too heavy for what concerns PC memory requirements; the curious reader who wants to examine the inner workings of `wrapfig2` can see that the macro that draws the frame with curved corners is about a dozen lines long, and there are no problems with any PC working memory.

Certainly these functionalities can be done much better by more experienced “`TEX`pers”; we hope that in the future better solutions can come.

Claudio Beccari
 RIVOLI, TO, ITALY
 claudio(dot)beccari(at)gmail(dot)com

Some macros to produce twin figures

Claudio Beccari

Abstract In humanistic texts it may be necessary to show two images close to one another, so as to have both of them visible at the same time and close to the text that contains suitable comments. The macros that are being proposed in this paper allow to set both images, suitably scaled and with their captions, so that they lay on the same line with the same height.

With some trickery it is possible to insert side by side two drawings made with the $\LaTeX$ drawing packages.

Sommario Nei libri di argomento umanistico può succedere di dover mostrare due immagini da commentare avendo entrambe visibili vicino al testo che le riguarda. Le macro proposte consentono di affiancare due figure adeguatamente scalate in modo che le immagini e le didascalie appaiano allineate e della stessa altezza.

Con qualche semplice artificio è possibile inserire uno accanto all'altro due disegni eseguiti con i pacchetti per disegnare di $\LaTeX$.

1. Introduction

It happens more often than not that images have to be displayed close to one another for several reasons. Humanistic tests may compare images of the same subjects painted by different artists, or scientists may show similar equipments to produce the same results or built in different epochs with different technologies.

The author himself needed to display pictures of paintings existing in the same room of a palace he had access to. The images shown in this paper were made by the author (with his modest photographic devices) so he does not have to specify any copyright licence. The paintings are several centuries old and the painters' names are unknown, so they are not subject to any copyright.

In order to align the images and their captions it is necessary to act with suitable macros.

Some time ago I described an environment, `doublefigure` where two pictures can be aligned without any geometrical relationship among them. The details are described in an Italian commented translation of the C appendix of the standard handbook by Leslie Lamport; the document is freely downloadable from the Web at this URL: <http://www.guitex.org/home/images/doc/GuideGuIT/Latexhandbookcommentato.pdf>; with this environment each image is typeset centred within a fixed width box that occupies approximately half of the line width; similarly each caption is typeset within the same half width boxes and the four of them (two boxes for the images and two ones for their captions) were vertically and horizontally suitably aligned. This environment was suited for similarly sized pictures, but gave poor results with pictures having different height to width ratios.

The author, therefore, devised a new environment, `twinfigures`, where suitable box scaling is used in order that different sized pictures with different height to width ratios could lay one close to the other so that their displayed heights are equal.

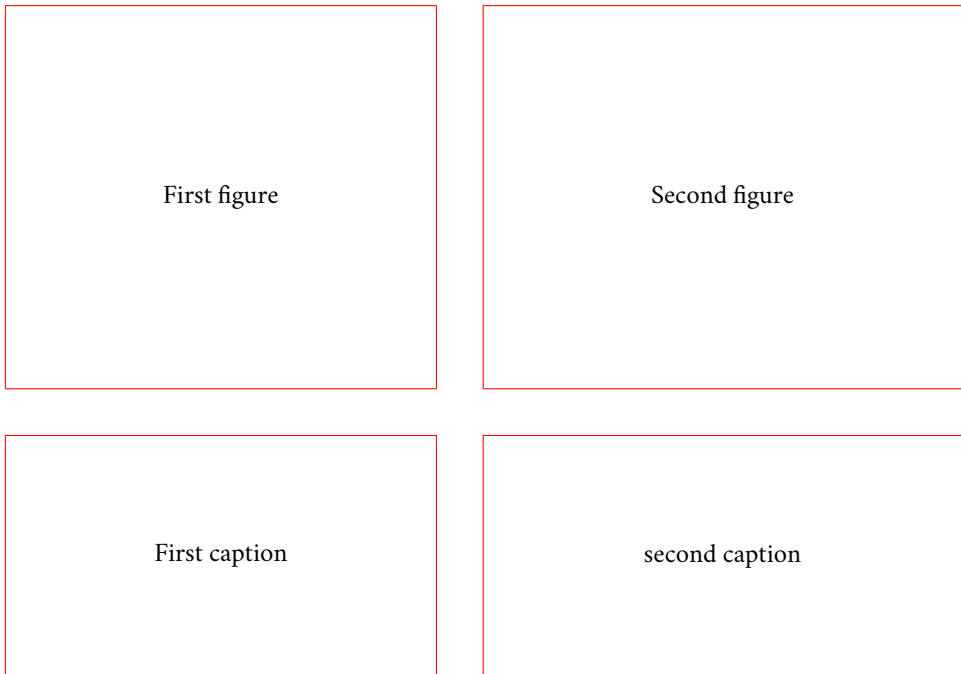


Figure 1. Schematic layout of the `doublefigure` environment.

2. Environment comparison

The former environment `doublefigure` produces a layout such as the one schematically visible in figure 1.

The source code to set two images with their captions is as the following self explanatory one:

```
\begin{doublefigure}
  \firstfigure
    \centering
    \includegraphics[width=\linewidth]{one.jpg}
    \caption{First caption}\label{f:1st fig}
  \secondfigure
    \centering
    \includegraphics[width=\linewidth]{two.jpg}
    \caption{Second caption}\label{f:2nd fig}
\end{doublefigure}
```

The actual figures are centred within the boxes `\linewidth` wide, but actually these line widths are those of the mini pages used to box their contents; of course the user can set a smaller value, if the picture requires it; by playing with the box width, it is possible to find the correct values so as to scale the pictures to the same heights; but the white spaces around them become unequal and the display becomes a poor one.

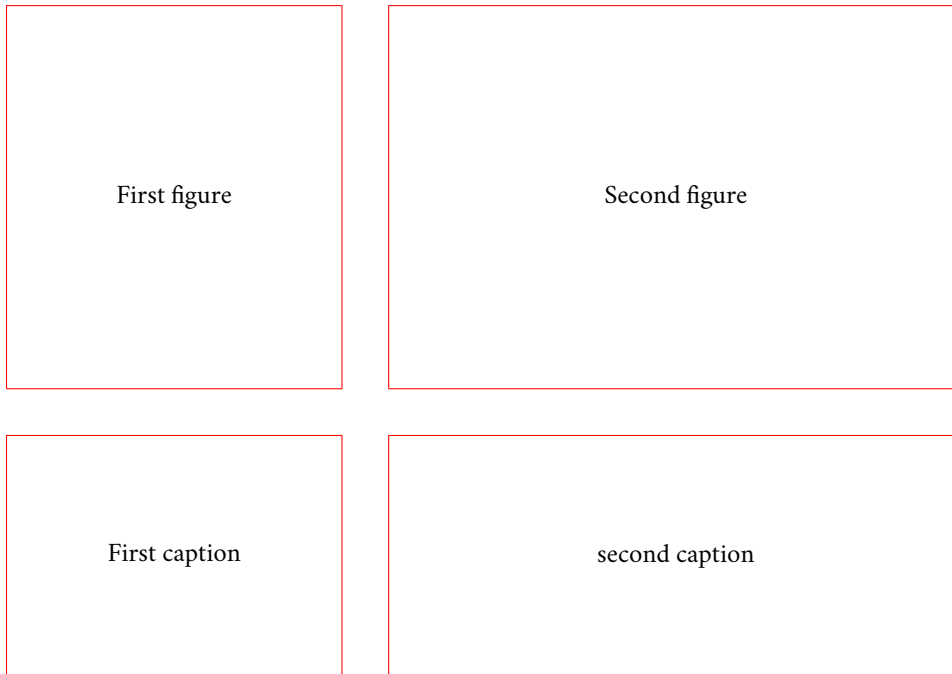


Figure 2. Schematic layout of the `twinfigures` environment.

On the opposite, the `twinfigures` environment scales the heights of both pictures so as to be equal, and determines the width of the enclosing boxes in such a way that each picture fills its box without any white space around.

The schematic layout becomes the one displayed in figure 2.

The code to produce this layout when using the same figures `one.jpg` and `two.jpg` is the following.

```
\begin{twinfigures}{\langle first figure.jpg \rangle}{\langle second figure.jpg \rangle}[\langle position \rangle]<\langle figure separation \rangle>
\firstfigure
\caption{\langle First caption \rangle}\label{\langle fig:1st fig \rangle}
\secondfigure
\caption{\langle Second caption \rangle}\label{\langle fig:2nd fig \rangle}
\end{twinfigures}
```

As it can be seen, the user code is very similar, but the computations behind the scenes are quite different. The third optional argument, preset to `htb`, transfers the optional `\langle position \rangle` to the internal figure environment. Users can specify the `\langle position \rangle` code they like best.

3. The code

The environment source code is the following; users should insert this code either into their main document source file preamble, or within a personal .sty file.

```

1 \usepackage{graphicx}
2 \newlength\Wuno    \newlength\Wdue
3 \newlength\Huno    \newlength\Hdue
4 \newlength\Lwidth  \newlength\FigSep
5
6 \NewDocumentEnvironment{twinfigures}{m m O{htb} D<>{0.05\cs{\textwidth}}}{% opening
7   \setbox0\hbox{\includegraphics{#1}}%
8   \setbox2\hbox{\includegraphics{#2}}%
9   \Wuno=\wd0 \Huno=\ht0 \Wdue=\wd2 \Hdue=\ht2
10  \Lwidth=\dimexpr\linewidth-#4\relax
11  %
12  \edef\Suno{\fpeval{\Lwidth*\Hdue/(\Wuno*\Hdue+\Wdue*\Huno)}}
13  \edef\Sdue{\fpeval{\Lwidth*\Huno/(\Wuno*\Hdue+\Wdue*\Huno)}}
14  \dimen0=\Suno\Wuno \dimen2=\Sdue\Wdue
15  %
16  \def\firstfigure{\begin{minipage}[t]{\dimen0}%
17    \resizebox{\hsize}{!}{\box0}}
18  \def\secondfigure{\end{minipage}\hfill
19    \begin{minipage}[t]{\dimen2}%
20    \resizebox{\hsize}{!}{\box2}}
21  \begin{figure}[#3]
22  }{% closing
23    \end{minipage}%
24    \end{figure}
25 }
```

Some comments are in order.

1. The functionalities of the `graphicx` package are required; remember to load that package unless you already did.
2. Lines 2, 3, and 4 define new length registers that the code will use for its computations.
3. Line 6 is recognised as a $\text{\LaTeX}3$ new statement; actually it is not so new, because the experimental code of packages `xparse` and `xfp` has been almost totally included into the $\text{\LaTeX}$ kernel; therefore their defined user commands are freely usable by all users without the need to load any package; of course this implies that users do not use a $\text{\TeX}$ system vintage version.

Notice that both mandatory arguments to the opening statement contain the names of the image files to work with; such names should contain a directory sequence either with respect to the main .tex file of their document, or a complete file address relative to the Home directory of their operating system; this Home directory may be indicated with the tilde sign, ~, but such sign points to different directories according to the Operating System rules.

The image extension can be omitted; if the extension is missing, the typesetting program can find the file according to a specific search sequence; problems might arise if the image folder contains two or more files with the specified name; such a problem arises only if multiple files with the same name and different extensions are in the same directory.

The third optional argument, as said before, allows users to specify the $\langle position \rangle$ they like best to print the paired images.

The fourth optional argument (enclosed in angle brackets $\langle \rangle$) allows the users to choose a different spacing compared to the default 5% of the line measure.

4. Line 7 defines the spacing between the image boxes; users can set a different value, for example `\FigSep=0.075\linewidth` or `\FigSep=2em`, or whatever value they prefer. This setting should be done before using the `twinfigures` environment. Of course the second optional argument could be used to set this distance.
5. Lines 8 and 9 store the specified image files in some service boxes, that the software will examine in order to compute the scaling factors.
6. Line 10 determines the widths and heights of the image boxes and establishes the overall length available for the image scaled boxes. Let w_1, h_1, w_2, h_2 be the original widths and heights of both boxes before scaling, and l the available length, i.e. the current measure less the spacing between figures.
7. Lines 13 and 14 compute the scaling factors s_1 and s_2 in order to satisfy the requirements specified by the problem at hand, that is: the scaled boxes height should be the same, while the sum of the scaled widths should equal the available length:

$$s_1 w_1 + s_2 w_2 = l \quad \text{and} \quad s_1 h_1 = s_2 h_2$$

The solution of this simple equation system is given by

$$\begin{cases} s_1 = \frac{lh_2}{w_1 h_2 + w_2 h_1} \\ s_2 = \frac{lh_1}{w_1 h_2 + w_2 h_1} \end{cases}$$

The computations are performed by the $\text{\LaTeX}$ 3 user command `\fpeval` that executes floating point decimal computations. When arguments refer to dimensions, the values used by `fpeval` are their fractional decimal values in typographical points.

8. Line 15 assigns the scaled box widths to the local service length registers `\dimen0` and `\dimen2`
9. Lines 17 and 18 define the `firstfigure` macro that resizes the first box containing the first picture; it opens a `minipage` environment with the computed width and the position option set to `t`, so that the reference point is at the base of the included box.
10. lines 19, 20, and 21 define `\secondfigure` that closes the `minipage` opened by the previous command, and does the same things for the second box.
11. Line 22 opens a floating figure environment and is the last statement of the opening commands.
12. Lines 24, 25, and 26 close the last opened `minipage` environment, and close the floating figure environment.



Figure 3. A fresco on the ceiling with Eros.



Figure 4. A Madonna inspired by the fresco Madonna del sacco painted by Beato Angelico.

Therefore when the user opens the environments with its two arguments, all the necessary computations are performed, so that `\firstfigure` provides the output of the scaled first figure, and leaves the minipage open in order to let the user input the figure caption and label; command `\secondfigure` closes the first minipage and opens the second one with the output of the second scaled box, again leaving space for the user to input the second figure caption; the closing `twinfigures` environment statement, besides closing the last minipage, closes the floating environment that saves everything to be output by the typesetting program that finds a suitable place to place it.

4. Example

The following code produces the twin figures displayed in figures 3 and 4. The used code is the following.

```
\begin{twinfigures}{../Angioletto}{../Madonna}
\firstfigure
\caption{A fresco on the ceiling with Eros}\label{fig:angioletto}
\secondfigure
\caption{A Madonna inspired by the Madonna del sacco fresco
  painted by Beato Angelico}\label{fig:Madonna}
\end{twinfigures}
```

The `../` points to the current document folder; it is not necessary, but here it is used as a place holder in case the image files are stored somewhere else.

5. Two drawings side by side

The above code produces the expected results only for pictures stored in graphic files; with pictures there is just the choice of the picture format, .jpg, .png, .pdf, and so on. For drawings made with any drawing program it is not possible to include the specific code in the argument fields of the environment opening statement, therefore it is necessary to resort to some trick such as the one described hereafter.

Simply draw each drawing in a separate $\text{\LaTeX}$ file to be processed with any of the usual $\text{\LaTeX}$ based engines; but use the `standalone` class. By processing such file with `PDF $\text{\LaTeX}$` (or any other $\text{\LaTeX}$ based engine) you get the cropped pdf file containing the drawing.

The small .tex file intended for the above process should have the following structure.

```
%% Drawing Name file
\documentclass{standalone}
<other preamble packages>
\begin{document}
\unitlength=<fraction>\textwidth
\noindent
\begin{picture}(<width>,<height>)(<x displacement>,<y displacement>)
<drawing commands>
\end{picture}
\end{document}
```

The example makes recourse to the `picture` drawing environment, but the situation is very similar whatever other drawing software you use. Personally I like to use the `euclideangeometry` package, and I require it among the `<other preamble packages>`.

For the `<fraction>` value I suggest to use a very small value, for example 0.01.

The `<width>` and `<height>` values express the size for the canvas to draw on; the cropped out file shall have those dimensions `<width>` and `<height>` multiplied by the chosen `\unitlength`; and they will be the dimensions of the original drawing. The `x` and `y` displacements, are the coordinates of the canvas lower left corner relative to the axes origin.

Do the same for both drawings to be set side by side, and use two different Drawing Names for the file names; let us assume that they are named `DrawingOne.tex` and `DrawingTwo.tex`.

It is possible to typeset both files and obtain the cropped images in files `DrawingOne.pdf` and `DrawingTwo.pdf`. These are the names to enter into the two arguments of the opening `twinfigures` statement.

In some cases this is not sufficient; but this inconvenience does not happen because the `twinfigures` environment is failing its purpose.

In facts assume that a critical edition of the Euclides' Elements is being written; the Greek text is very difficult to understand, because the mathematical ancient Greek language is extremely different from the modern one. Very often the names of segments are used as the areas of the squares, or triangles, or polygons the Euclidean statements are related to.

This implies that the Euclidean statement requires two drawings that must be in the same scale. This requires that the `<factor>` be the same as well as the canvas `<width>` and `<height>`, even if such constraints may produce cropped drawings that contain some white space above, or below, or at the sides of the real drawing. See for example the drawings of figures 5 and 6.

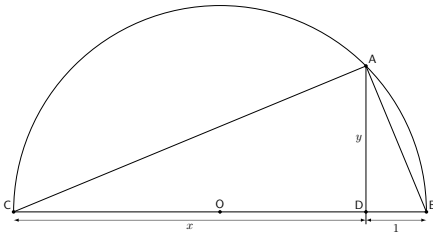


Figure 5. A segment y as long as $\sqrt{x}$.

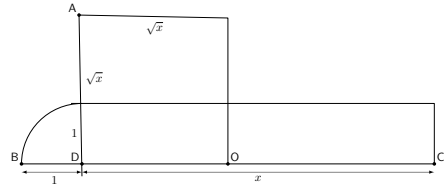


Figure 6. A square equivalent to a rectangle.

As it can be seen, the left figure has the semicircle that is much higher than anything drawn in the right figure, but both AD segments, one labelled y in the left figure, and the other, labelled $\sqrt{x}$ in the right figure, should be equal in both figures. This inconvenience does not happen with pictures, but it can happen in mathematical or technical writings where drawings, schematic diagrams, sketches, and the like, are very frequent. Critical editions of mathematical texts are half way between humanist writings and hard science ones.

6. Conclusion

The code provided to the $\text{\texttt{Ar\TeX nica}}$ readership is very handy; I use it very frequently, except when I write technical material. I hope it may be useful for other $\text{\texttt{L\TeX}}$ users, especially humanists who, I suppose, may find it useful for their documents.

Claudio Beccari
RIVOLI, TO, ITALY
claudio(dot)beccari(at)gmail(dot)com

I polinomi di Bernstein e le funzioni intere e fratte di Bézier

Claudio Beccari

Sommario Il polinomi di Bernstein hanno delle proprietà particolari; fra i tanti usi che hanno nelle discipline matematiche, essi ricordano da vicino le formule che definiscono le spline di Bézier, alla base del disegno automatizzato mediante gli elaboratori elettronici grandi e piccoli.

Qui si esamineranno alcune applicazioni per la grafica, anche quella che si può fare con i comandi nativi del linguaggio $\text{\LaTeX}$ e con le estensioni dei suoi pacchetti.

Abstract Bernstein polynomials have peculiar properties; besides the various mathematical applications, they recall the defining formulas of the Bézier splines, that are at the base of the computerised graphic programs.

Here we examine some graphic applications that are being used by native $\text{\LaTeX}$ macros and by the multitude of its extension packages.

1. Introduzione

Il sistema $\text{\TeX}$ diventò disponibile al pubblico insieme al programma METAFONT necessario per creare i font Computer Modern, dove ogni glifo era prodotto mediante una matrice densissima di piccoli punti neri all'interno di un contorno definito mediante spline di Bézier del terzo ordine. A causa delle limitazioni delle stampanti degli anni '70 del XX secolo, quella era l'unica tecnologia usabile; essi venivano chiamati *bit mapped fonts* e richiedevano proprietà particolari per comporre i testi di cui Donald Knuth aveva bisogno.

Con l'avvento del linguaggio PostScript, i font elettronici divennero *vector fonts* con proprietà molto avanzate rispetto ai precedenti font a matrici di punti. In particolare era la stampante a riempire di macchioline nere il contorno chiuso formato da linee eseguite mediante spline del terzo ordine, il cui disegno era astrattamente definito da queste spline, legate alle coordinate di certi punti guida; cioè in base a informazioni matematiche che prescindevano completamente dalle funzionalità delle stampanti; oggi i font elettronici hanno contorni definiti con spline del secondo o del terzo ordine.

Il linguaggio $\text{\LaTeX}$ nacque con la prima versione nei primi anni '80 del XX secolo; disponeva di un ambiente elementare, *picture*, che permetteva di eseguire disegni semplicissimi che potevano contenere linee e vettori con pendenze formate dai rapporti fra interi, positivi o negativi, primi fra loro e non superiori a 6. Gli archi di cerchio potevano avere raggi di curvatura limitati in numero e non superiori ad un limite di circa una ventina di punti tipografici. $\text{\LaTeX}$, con poche modifiche arrivò fino alla versione 2.09 (e fu chiamato $\text{\LaTeX}209$); nel 1994 nacque $\text{\LaTeX}2\epsilon$, con modifiche sostanziali nella gestione dei font e con tante nuove funzionalità; nel suo testo di descrizione oltre che manuale d'uso, Leslie Lamport descrisse un nuovo ambiente *picture* che non aveva più le limitazioni della precedente versione, ma demandò ad altri la creazione delle macro necessarie per attuare le nuove caratteristiche.

Finalmente nel 2003 fu reso pubblico il pacchetto *pict2e* che rendeva usabili anche i comandi interni del nuovo nato programma di composizione *pdftex* che esegue la composizione usando una versione ridotta delle funzionalità del linguaggio PostScript. In questo modo *pdflatex* (e i successivi *X_YTeX* e *Lua_YTeX*) potevano usare direttamente i font vettoriali, ma anche l'ambiente *picture* poteva usare certi comandi PostScript che aprivano la strada ad una grafica elettronica piuttosto avanzata. Oggi quando ci si riferisce all'ambiente *picture* ci si riferisce alla versione avanzata mediante l'uso del pacchetto *pict2e*.

I polinomi di Bernstein sono “partizioni dell’unità”; le funzioni di t si indicano con B_i^n e indicano un polinomio di grado n che ha in $t = 0$ uno zero di ordine i e hanno un altro zero di ordine $n - i$ per $t = 1$; questo implica che: (a) deve essere $0 < i < n$; ma per $i = 0$ il polinomio vale 1 per $t = 0$ e altrettanto vale 1 per $i = n$; vedi la figura 1. È chiaro che il polinomio di grado zero e indice 0 vale identicamente 1, come si vede nella figura.

I polinomi di Bernstein sono definiti in questo modo

$$b_i^n(t) = t^i(1-t)^{n-i} \quad (1)$$

Nella figura 1 sono rappresentati i polinomi

$$\begin{aligned} b_0^4 &= (1-t)^4 && \text{rosso} \\ b_1^4 &= 4t(1-t)^3 && \text{verde chiaro} \\ b_2^4 &= 6t^2(1-t)^2 && \text{ciano} \\ b_3^4 &= 4t^3(1-t) && \text{blu} \\ b_4^4 &= t^4 && \text{magenta} \end{aligned} \quad (2)$$

La linea arancione rappresenta la somma dei cinque polinomi e giustifica il fatto che vengano chiamati “partizioni dell’unità”. Si notino i coefficienti 1, 4, 6, 4, 1 dei singoli polinomi che sono gli stessi valori che si trovano nella quarta riga del triangolo di Tartaglia; sono sostanzialmente i coefficienti binomiali

$$\binom{4}{i} \quad \forall i = 0, 1, 2, 3, 4$$

2. I polinomi o spline di Bézier

Le spline di Bézier si usano per la grafica al calcolatore. A rigore il nome inglese *spline* indica il curvilinee; chiunque abbia disegnato schemi tecnici o diagrammi scientifici, da parecchi decenni, per non dire secoli, usa curvilinee di legno, o di plastica o di metallo. Da quando

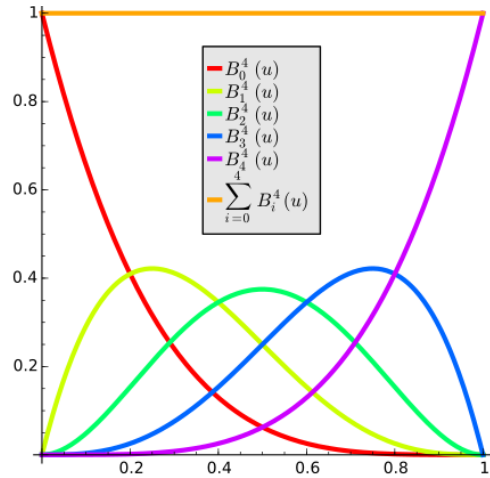


Figura 1. Da Wikipedia

esistono programmi di disegno al calcolatore, le spline (diventata una parola italiana dato il suo uso frequentissimo) le spline di Bézier sono diventate il pane quotidiano. PStricks è forse il programma più efficace fra quelli liberi. Ma il compilatore `pdfTeX`, che usa il linguaggio PDF, un sottoinsieme PostScript, permette di usare alcuni comandi interni che rendono accessibile parte del linguaggio PostScript. Uno degli ambienti nativi di $\text{\LaTeX}$ è l'ambiente *picture* che quando è nato nel 1984 permetteva di fare solo disegni elementari, senza usare spline; nel 1994 il creatore di $\text{\LaTeX}$ ha aggiornato la sua creazione, e ha annunciato una notevole estensione dell'ambiente *picture* in modo da eliminarne i suoi difetti; ci sono voluti 9 anni, ma poi queste estensioni sono state realmente messe a disposizione degli utenti nel 2003 mediante il pacchetto `pict2e`; questo pacchetto è stato ulteriormente esteso e oggi (2024) l'ultimo suo aggiornamento risale al 2020, con la possibilità di usare per le coordinate e le altre dimensioni, anche esplicite, non solo le misure in `\unitlength` di tali quantità; oggi, quindi, è possibile usare, per esempio, `\put(1cm,25mm){\circle{2mm}}` quando prima era impossibile.

Ma fin dai primi anni della sua esistenza `pict2e` ha permesso di usare i comandi interni del linguaggio PDF `\moveto`, `\lineto`, `\curveto`, `\closepath`, `\fillpath`, `\strokepath`, con i quali si comincia una linea, si prosegue con segmenti o curve, si termina la linea complessiva tracciandola o colorandone l'interno se si tratta di una linea chiusa.

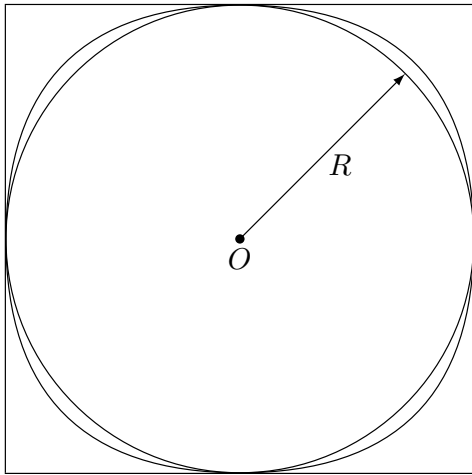


Figura 2. Confronto fra una circonferenza disegnata con spline di Bézier di secondo grado e una circonferenza disegnata con spline di terzo grado

Non è un caso che `\curve` disegni una spline di Bézier; inizialmente erano solo curve (polinomi) di secondo grado; ulteriori estensioni hanno messo a disposizione le spline di terzo grado.

Un polinomio di Bézier è legato ai polinomi di Bernstein da questa espressione

$$B_n(t) = \sum_{i=0}^n P_i b_i^n(t) \quad (3)$$

dove P_i sono le coordinate di punti sul piano (di disegno); la curva comincia in P_0 e termina in P_n ; gli altri punti P_i sono i punti guida. Per $n > 3$ la spezzata chiusa che unisce in sequenza i vari punti chiudendosi nel punto iniziale non ha una interpretazione precisa, ma per $n \leq 3$ questa spezzata chiusa rappresenta il dominio entro il quale si sviluppa la linea definita dal polinomio di Bézier quando

t viene fatto crescere da 0 a 1; la linea parte da P_0 verso P_1 con tangente pari al vettore $P_1 - P_0$ e arriva a P_n arrivando da P_{n-1} con tangente pari al vettore $P_n - P_{n-1}$.

Se $n = 0$ il polinomio si riduce ad una costante, e la linea si riduce al solo punto P_0 ; se $n = 1$ la linea è il segmento che unisce P_0 e P_1 ; se $n = 2$ la linea è una parabola che parte da P_0 con la propria tangente e arriva a P_2 con la propria tangente, sempre rispettando le tangenti iniziale e finale, con $n = 3$ la linea che unisce P_0 e P_3 corrisponde al disegno di un polinomio di terzo grado che nell'intervallo delimitato per la variabile indipendente t può contenere un flesso se i punti P_1 e P_2 giacciono ai lati opposti del segmento che unisce P_0 e P_3 .

Nel pacchetto `curve2e` ci sono ulteriori estensioni di `pict2e`, in particolare sono stati definiti i comandi `\Curve` che è un comando iterativo a cui basta conoscere la sequenza delle coppie $\{P\}<D>$ formata da una sequenza di due o più di queste coppie per tracciare spline di terzo grado fra ogni coppia di punti P della sequenza insieme alla direzione D in ciascun punto; e altri comandi più articolati che permettono di specificare anche la tensione degli archi, se vogliamo un parametro legato alla curvatura della curva in corrispondenza di ciascun punto.

Funzionano abbastanza bene, ma si tratta sempre di approssimazioni. Con queste macro si sono costruite macro per disegnare cerchi, archi circolari, ellissi, parabole e ottime approssimazioni di iperboli. Il pacchetto `curve2e` usa l'aritmetica complessa, per cui le coordinate dei punti, così come i versori delle direzioni possono essere specificati con una sola macro che contiene le informazioni cartesiane oppure polari dell'oggetto a cui si riferiscono. Una variante delle macro di servizio permette di specificare i quattro punti di una spline di terzo grado in modo che il coefficiente complessivo di t^3 sia nullo, cosicché la spline di secondo grado che si ottiene permette di disegnare con precisione una parabola.

Si sono fatti disegni incredibili con queste funzionalità grafiche già implementate, ma resta sempre un residuo di insoddisfazione visto che si tratta di approssimazioni quando si parla di figure geometriche specifiche come il cerchio. Per il cerchio l'approssimazione di ogni quarto mediante una spline di secondo grado presenta un errore radiale di circa il 6% al centro dell'arco, errore relativo che passa inosservato in molte circostanze, ma che si vede benissimo in altre. Nella figura 2 il cerchio interno quasi perfetto è tracciato con spline cubiche, mentre il cerchio esterno è disegnato con spline quadratiche.

3. Funzioni razionali di Bézier

Le spline di Bézier possono essere anche definite mediante funzioni fratte

$$R_n(t) = \frac{\sum_{i=0}^n w_i P_i b_i^n(t)}{\sum_{i=0}^n P_i b_i^n(t)} \quad (4)$$

Queste spline permettono di fare cose che con quelle polinomiali non si possono fare.

Con la spline di secondo grado

$$R_2(t) = \frac{P_0 w_0 b_0^2(t) + P_1 w_1 b_1^2(t) + P_2 w_2 b_2^2(t)}{w_0 b_0^2(t) + w_1 b_1^2(t) + w_2 b_2^2(t)} \quad (5)$$

si possono fare cose molto più complesse di quelle che si possono ottenere con le spline polinomiali.

Innanzitutto ci sono i pesi w_i che possono anche essere nulli (non tutti, però, non perché la funzione non sia definita, ma perché una funzione nulla serve a poco). Essi rappresentano dei pesi (w iniziale di *weight*) che in pratica definiscono un polinomio di polinomi, polinomio che può essere qualsiasi; tuttavia il denominatore in questa doppia trasformazione sembra essere il polinomio unitario in quello spazio di polinomi. Complicare le cose semplici? Non direi; tuttavia fior di matematici hanno usato le funzioni razionali di Bézier per scopi che vanno molto al di là dell'approssimazione di curve.

Di qui in avanti ci si riferirà solo alle funzioni razionali di Bézier di secondo grado; nella documentazione del pacchetto `lapdf` ci sono vari esempi, ma qui si presenta solo una figura con varie curve che corrispondono a diversi pesi. La cosa più interessante è che se la curva data rappresenta un arco di una curva chiusa, cambiando segno ai pesi di ordine dispari si

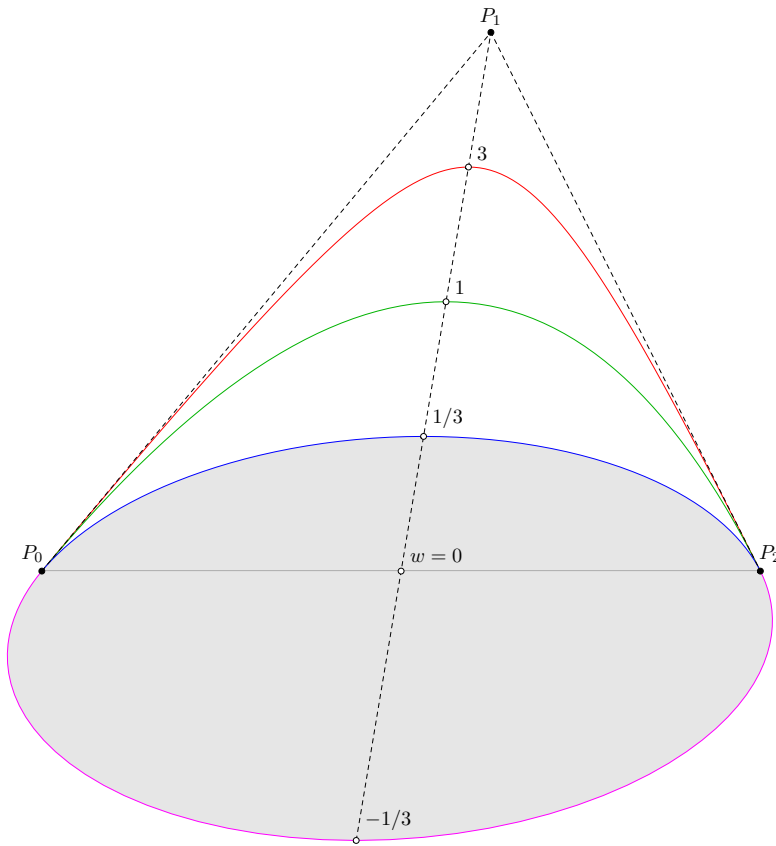


Figura 3. Punti nodali di una curva razionale di Bézier; le linee colorate corrispondono ad una iperbole (rossa), una parabola (verde), un arco di ellisse (blu), il corrispondente arco complementare (magenta).

ottiene l'arco complementare. Inoltre i pesi possono essere scelti in modo che le curve che si ottengono siano delle coniche esatte, cioè delle intersezioni di un (doppio) cono con un piano; è noto che se il piano interseca solo un cono l'intersezione nel piano è un'ellisse o, come caso particolare, un cerchio; se la pendenza del piano uguaglia la pendenza di una retta giacente sulla superficie del cono, l'intersezione è una parabola, mentre se il piano interseca entrambi i coni le intersezioni sono i due rami di un'iperbole.

La figura 6 è tratta dalla documentazione del pacchetto `lapdf` ed è stata leggermente ritoccata per rendere più visibile la linea verde. Ovviamente il codice che accompagna questa figura è scritto con i comandi di quel pacchetto; ma lo si può trovare (integro) nella sezione documentazione della distribuzione del pacchetto `lapdf`. I numeri che sono scritti a fianco della mediana del triangolo $P_0P_1P_2$ dal vertice in P_1 , si riferiscono al modo particolare di comporre i pesi con le macro di quel pacchetto. Tuttavia si vede che con i coefficienti positivi si hanno gli archi di curve contenuti nel dominio di quel triangolo, mentre con il coefficiente negativo si ottiene un arco esterno a quel triangolo. In particolare i due coefficienti $1/3$ e $-1/3$ permettono di disegnare i due archi di ellisse interno ed esterno rispetto alla corda

P_0P_2 ; questa corda non coincide con il diametro dell'ellisse, ma unisce i due punti in cui i due archi ellittici sono tangenti ai due lati del triangolo, in particolare sono tangenti nei punti di partenza e di arrivo di tutte le curve di Bézier disegnate.

3.1. I polinomi di Bernstein di secondo grado

Per costruire le funzioni razionali di Bézier è necessario avere una idea precisa in merito ai polinomi di Bernstein con i quali poi sarà più facile costruire le funzioni razionali. Ci riferiamo alle funzioni di secondo grado:

$$b_i^2(t) = \binom{2}{i} t^i (1-t)^{2-i} \quad (6)$$

cioè

$$\begin{aligned} b_0^2(t) &= (1-t)^2 \\ b_1^2(t) &= 2t(1-t) \\ b_2^2(t) &= t^2 \end{aligned} \quad (7)$$

Qui dunque si mostrano le macro che, con gli strumenti di cui dispone `curve2e` permettono di disegnare le curve dei tre polinomi indicati nell'equazione (7).

```
\def\Bern#1#2#3{%
\ifcase#2
% i=0
\fpval{#1*(1.0-#3/#1)**2}%
\or
% i=1
\fpval{2*#1*(#3/#1)*(1.0-#3/#1)}%
\or
%i=2
\fpval{#1*(#3/#1)**2}%
\fi}
```

Come si vede si è spezzata la definizione nei tre casi mostrati nell'equazione suddetta e abbiamo definito comunque una `\Bern` con tre argomenti che hanno i seguenti significati.

1. Il primo argomento è un fattore di scala che permette di passare dalle coordinate grafiche di un ambiente *picture* alla variabile t che ha il dominio formato dall'intervallo 0–1, e viceversa, perché anche il polinomio di Bernstein ha lo stesso dominio.
2. Il secondo argomento riceve il valore dell'indice i .
3. Il terzo argomento riceve il valore della variabile t .

Con questa macro a disposizione diventa relativamente facile disegnare i grafici mediante delle spezzate formate da segmentini cortissimi ricorrendo alle macro interne del pacchetto `pict2e` richiamato dallo stesso pacchetto `curve2e`.

```

\def\Xbern{<i>}%
\edef\Ybern{\Bern{<scala>}{<i>}{\Xbern}}%
\moveto\Xbern,\Ybern
\fpdowhileMarg\Xbern<<scala>{\edef\Xbern{\fpeval{\Xbern+1}}}%
\edef\Ybern{\Bern{<scala>}{<i>}{\Xbern}}%
\lineto\Xbern,\Ybern}%
\strokepathcs\ignorespaces

```

Il risultato è quello che si vede nella figura 4 eseguito con il codice seguente:

```

\begin{figure}\centering
\unitlength=0.005\linewidth
\begin{picture}(110,110)(-10,-10)
\GraphGrid(100,100)
\VECTOR(0,0)(100,0)\VECTOR(0,0)(0,100)
\multiput(0,0)(10,0){10}{\segment(0,0)(0,-2)}
\multiput(0,0)(0,10){10}{\segment(0,0)(-2,0)}
\xmultiput(10,0)(10,0){9}{\put(0,-3){\makebox(0,0)[t]{0,\arabic{multicnt}}}}
\xmultiput(0,10)(0,10){9}{\put(-3,0){\makebox(0,0)[r]{0,\arabic{multicnt}}}}
\put(-3,0){\makebox(0,0)[r]{0}}
\put(0,-3){\makebox(0,0)[t]{0}}
\put(-3,100){\makebox(0,0)[r]{1,0}}
\put(100,-3){\makebox(0,0)[t]{1,0}}
\put(0,101){\makebox(0,0)[bl]{\$b_i^2\$}}
\put(101,0){\makebox(0,0)[bl]{\$t\$}}
%
\thicklines
%
\color{green!50!black}
\def\Xbern{0}%
\edef\Ybern{\Bern{100}{0}{\Xbern}}%
\moveto(\Xbern,\Ybern)
\fpdowhile{\Xbern<100}{\edef\Xbern{\fpeval{\Xbern+1}}}%
\edef\Ybern{\Bern{100}{0}{\Xbern}}%
\lineto(\Xbern,\Ybern)%
\strokepath\ignorespaces
%
\color{red}%
\def\Xbern{0}%
\edef\Ybern{\Bern{100}{1}{\Xbern}}%
\moveto(\Xbern,\Ybern)
\fpdowhile{\Xbern<100}{\edef\Xbern{\fpeval{\Xbern+1}}}%
\edef\Ybern{\Bern{100}{1}{\Xbern}}%
\lineto(\Xbern,\Ybern)%
\strokepath\ignorespaces
%
\color{blue}%
\def\Xbern{0}%
\edef\Ybern{\Bern{100}{2}{\Xbern}}%
\moveto(\Xbern,\Ybern)

```

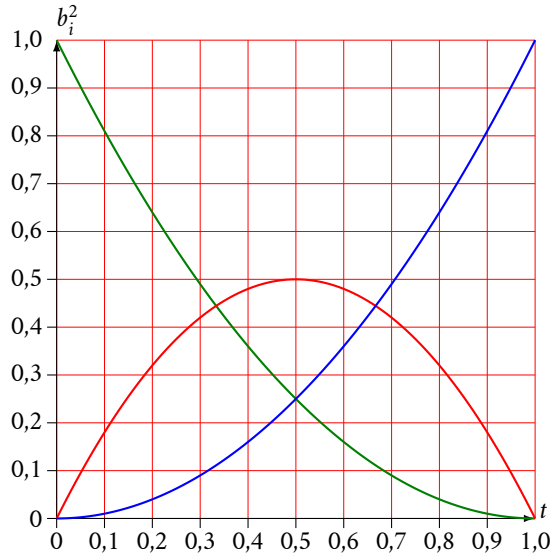


Figura 4. Tre polinomi di Bernstein di secondo grado: verde $i = 0$; rosso $i = 1$; blu $i = 2$.

```
\fpdowhile{\Xbern<100}{\edef\Xbern{\fpeval{\Xbern+1}}%
\edef\Ybern{\Bern{100}{2}{\Xbern}}%
\lineto(\Xbern,\Ybern)}%
\strokepath\ignorespaces
\end{picture}
\caption{Tre polinomi di Bernstein di secondo grado:
verde  $i=0$ ; rosso  $i=1$ ; blu  $i=2$ .\label{fig:3Bern}
\end{figure}
```

Nella figura 4 si può stimare ad occhio con la griglia le cui maglie sono distanziate di un decimo dell'unità sia in orizzontale sia in verticale, che la somma delle tre ordinate per ciascuna ascissa è uguale all'unità. Naturalmente la prova matematica è più convincente, ma dal punto di vista grafico qui ci accontentiamo.

3.2. Le funzioni razionali di Bézier di secondo grado

Avendo visto come disegnare i polinomi di Bezier di secondo grado, dobbiamo vedere come disegnare queste funzioni. Matematicamente esse sono ottenute dalla frazione:

$$B_2(P_0, P_1, P_2; w_0, w_1, w_2; t) = \frac{w_0 P_0 b_0^2(t) + w_1 P_1 b_1^2(t) + w_2 P_2 b_2^2(t)}{P_0 b_0^2(t) + P_1 b_1^2(t) + P_2 b_2^2(t)} \quad (8)$$

Si tenga presente che il pacchetto `curve2e` è in grado di gestire le coordinate dei punti, in particolare dei punti guida P_0, P_1, P_2 come dei numeri complessi sia in coordinate cartesiane sia in coordinate polari. Al suo interno lavora sempre in coordinate cartesiane, ma l'utente con certe cautele può specificare i punti e le direzioni come numeri complessi sia nella forma

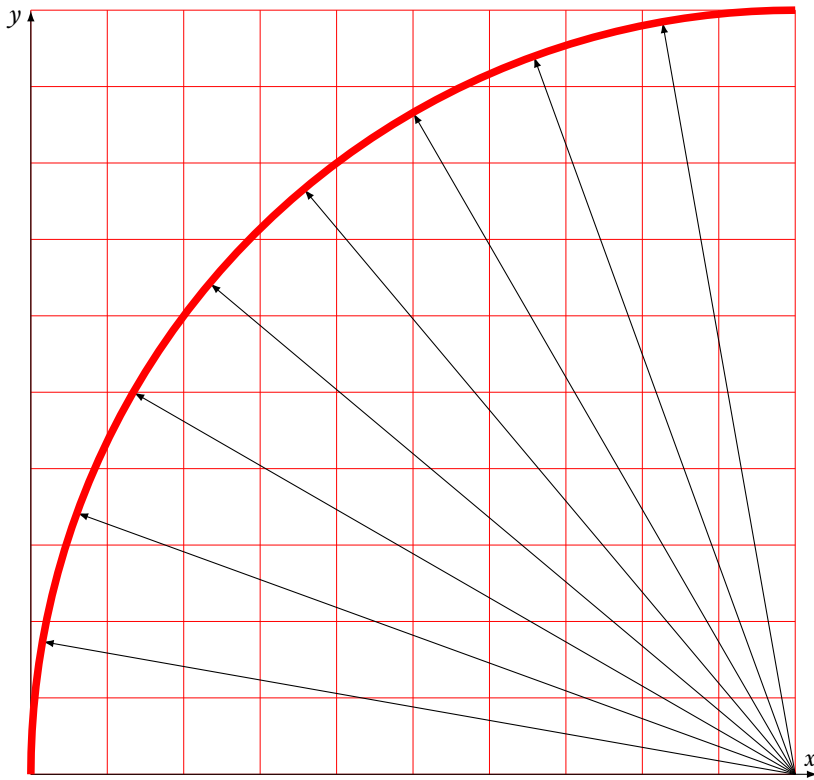


Figura 5. Un quarto di cerchio senza approssimazioni come sarebbe con le curve di Bézier intere

di ascissa e ordinata separate da un virgola, sia nella forma di anomalia e modulo separati da due punti.¹

I pesi w_0, w_1, w_2 , a pari punti guida, fanno cambiare le forme delle curve tracciate. In definitiva la forma frazionale delle spline di Bézier richiede 7 argomenti, di cui tre complessi. La macro `\RCbezier` avrà invece 9 argomenti; ai 7 necessari per la funzione, bisogna aggiungere il fattore di scala e il risultato del calcolo di un punto di quella funzione.

La figura 5 mostra chiaramente un quarto di cerchio descritto con la funzione fratta di Bézier; i vari vettori sottili che si dipartono dall'angolo inferiore sinistro del disegno, sono distanziati di un angolo di 10° l'uno dall'altro e il loro modulo è sempre uguale al raggio dell'arco; le loro punte giacciono sempre nella stessa posizione relativa all'arco rosso.

Si tenga presente che la funzione razionale di Bézier usata ha gli altri tre spigoli del disegno usati come punti guida, ma il peso relativo al punto estremo della diagonale del quadrato vale $w_{1c} = \sqrt{1/2}$; se valesse 1 si otterrebbe un arco di parabola, se fosse maggiore di 1 si

1. Si ricorda che la forma polare dei numeri complessi si esprime come $m e^{i\varphi}$ e l'anomalia φ si esprime con un valore compreso fra $-\pi$ e $+\pi$, quindi non è solo la semplice arcotangente del rapporto fra l'ordinata e l'ascissa. L'anomalia è a quattro quadranti mentre l'arcotangente ha un valore principale solo a due quadranti. I dettagli per usare correttamente i numeri complessi in forma polare, specialmente per quel che riguarda i punti del piano, sono descritti dettagliatamente nella documentazione del pacchetto `curve2e` e nel suo manuale per l'utente.

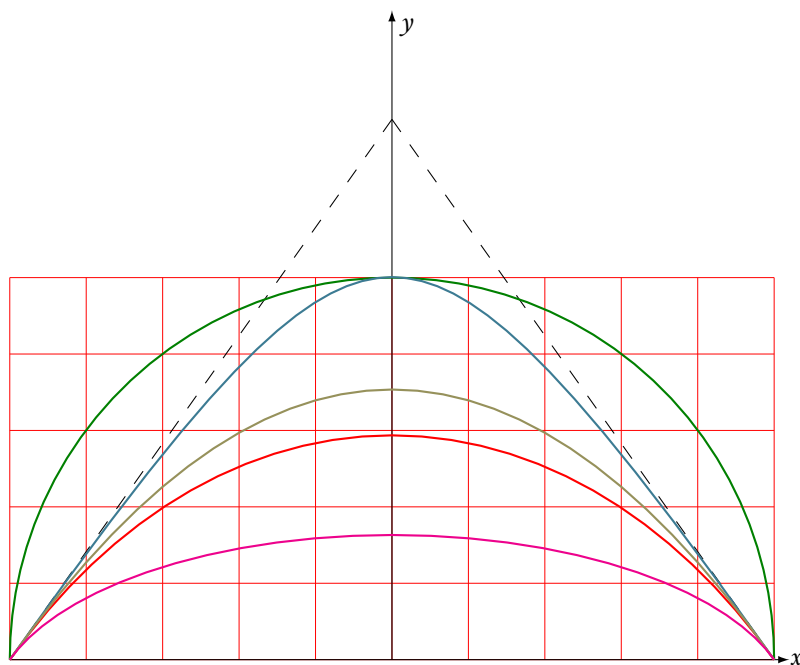


Figura 6. Semicirconfenza e altri archi di coniche. Rosso: circonferenza; blu: iperbole; giallo: parabola; magenta: ellisse; verde: semicirconfenza ottenuta congiungendo due quarti di circonferenza.

otterrebbe un arco di iperbole e se fosse minore di w_{1c} si otterrebbe un arco di ellisse; se quel peso fosse negativo, ma minore in valore assoluto a quello della parabola, si otterrebbero gli archi complementari del cerchio o dell'ellisse.

Nella figura 6 si vede invece una semicirconfenza (verde) completa, dove il punto guida giace fuori del rettangolo della griglia, e dista dalla base esattamente $\sqrt{2}$ volte il raggio della semicirconfenza.

Gli archi di coniche tangenti alle due linee tratteggiate nei due punti estremi passano dall'arco di iperbole (blu) con un peso $w_1 = 2.425$, al valore $w_1 = 1$ per la parabola verde, al valore $w_1 = 0.7071 \approx \sqrt{1/2}$ per l'arco circolare rosso e, infine, col valore $w_1 = -0.3$ per l'arco di ellisse.

Merita ancora vedere che cosa succede con le coniche che formano linee chiuse, cioè circonferenze ed ellissi. Nella figura 7 sono infatti rappresentati gli stessi archi con gli stessi colori della figura 6 solo che è stato disegnato anche l'arco complementare.

Onestamente bisogna ammettere che la cosa funziona come descritto in linea di massima, ma costruendo il disegno della figura 7 le coordinate verticali dei punti guida devono venire modificate un poco, così come il loro pesi. La teoria direbbe che per le curve complementari basta lavorare sul segno del peso della curva di Bernstein con indice dispari, senza cambiare le coordinate del punto guida. Non so se si tratti di un errore della programmazione o se la teoria sia un poco carente, tuttavia non è difficile aggiustare quei parametri numerici e il risultato un poco approssimativo non è male; se si misurano con un righello le lunghezze dei tre vettori sottili che compaiono nella figura 7, si vede che essi sono sostanzialmente uguali, accettabili per un disegno, un po' imprecise da un punto di vista matematico.

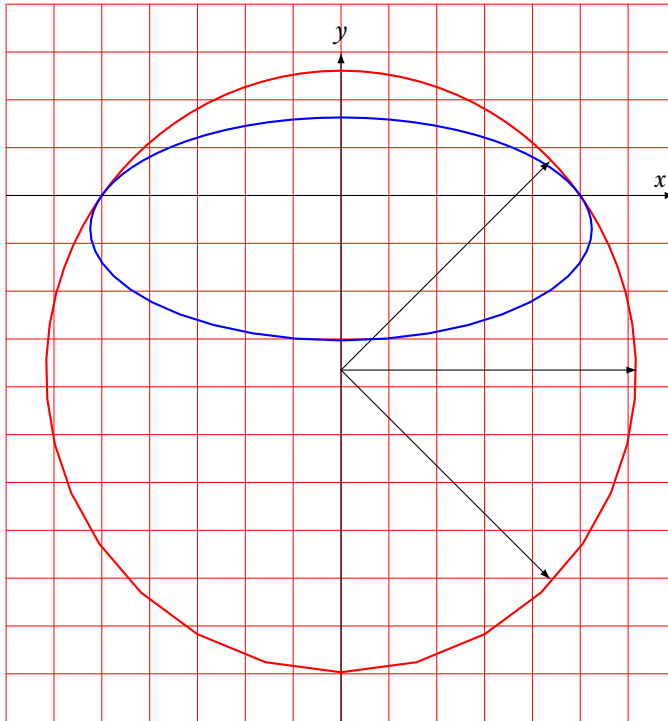


Figura 7. Una circonferenza e un'ellisse disegnate facendo uso delle curve di Bézier fratte complementari rispetto a quelle degli archi più corti

In ogni caso la sintassi di questo comando è la seguente:

```
\RBezier(<P0>)(<P1>)(<P2>)%  
      {<w0>}{<w1>}{<w2>}%  
      {<Pout>}<colore>>{★}
```

dove $P_i \forall i = 0, 1, 2$ sono le coordinate (numeri complessi) dei tre punti guida; $w_i \forall i = 0, 1, 2$; P_{out} sono le coordinate (numero complesso) del risultato dei calcoli; $\langle colore \rangle$ serve per specificare il colore da usare per il disegno; e dove $\langle \star \rangle$ serve per scegliere se partire ex novo o se proseguire dall'ultimo punto dell'arco precedente; se l'asterisco manca, il primo comando `\moveto` viene usato per iniziare il tracciamento della linea di contorno; invece se l'asterisco è presente, il disegno della linea prosegue dall'ultimo punto usato dalla macro precedente. Tocca all'utente sia mettere gli asterischi alla fine della lista di argomenti delle macro, dalla seconda in poi; sia il colore che, invece, deve essere specificato alla fine della prima macro; infine l'utente deve usare esplicitamente il comando `\strokepath` se vuole tracciare la linea di contorno, oppure `\fillpath` se vuole che il contorno sia riempito di colore. La figura 8 mostra diverse combinazioni.

Merita vedere la serie di comandi usati per comporre la figura 8. È bene notare che il codice per disegnare l'arco di parabola riempito di rosso non contiene il comando `\closepath`, perché ci pensa direttamente il comando `\fillpath` a chiudere la linea, altrimenti non saprebbe

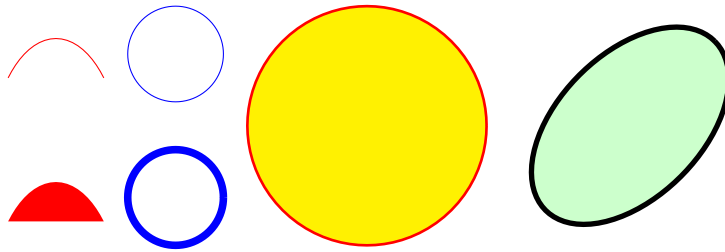


Figura 8. Alcune curve colorate, tracciate e/o riempite.

quale superficie riempire di colore; questa funzionalità fa già parte delle macro interne del pacchetto `pict2e`.

```
\begin{figure}\centering
\unitlength=0,005\linewidth
\begin{picture}(100,50)
\put(0,5){%
\RBezier(0,30)(10,50)(20,30){1}{0.70}{1}[20]<red>\strokepath
\RBezier(0,0)(10,20)(20,0){1}{0.70}{1}[20]<red>\fillpath}
%
\Bcircle(35,40){10}[20]<white><blue>
%
\Bcircle(35,10){10}[20]<white>[1mm]<blue>
%
\Bcircle(75,25){25}[20]<yellow>[1pt]<red>
%
\Bellipse(130,25){25,15}[20]<green!20!white>[2pt][45]
%
\end{picture}
\caption{Alcune curve colorate, tracciate e/o riempite.}
\label{fig:curve colorate}
\end{figure}
```

Le macro `\Bcircle` e `\Bellipse` sono definite una volta per tutte mediante quarti di linea chiusa e vengono ripetute per colorare l'interno o il contorno. Le ellissi possono venire ruotate come si vede nella figura 8.

```
\NewDocumentCommand\Bcircle{r() m O{50} D<>{white}
O{0.4pt} D<>{black}}%
{\put{#1}{%
\linethickness{Opt}
\RBezier(#2,0)(#2,#2)(0,#2){1}{0.7}{1}[#3]<#4>
\RBezier(0,#2)(-#2,#2)(-#2,0){1}{0.7}{1}[#3]*
\RBezier(-#2,0)(-#2,-#2)(0,-#2){1}{0.7}{1}[#3]*
\RBezier(0,-#2)(#2,-#2)(#2,0){1}{0.7}{1}[#3]*\fillpath
%
}}
```

```

\linethickness{#5}
  \RBezier(#2,0)(#2,#2)(0,#2){1}{0.7}{1}[#3]<#6>
  \RBezier(0,#2)(-#2,#2)(-#2,0){1}{0.7}{1}[#3]*
  \RBezier(-#2,0)(-#2,-#2)(0,-#2){1}{0.7}{1}[#3]*
  \RBezier(0,-#2)(#2,-#2)(#2,0){1}{0.7}{1}[#3]*\strokepath
%
}\ignorespaces}

\NewDocumentCommand\Bellipse{r() m O{50} D<>{white}
  O{0.4pt} D<>{black} O{0}}%
{\GetCoord(#2)\BeX\BeY
\put(#1){\rotatebox[origin=c]{#7}{\linethickness{0pt}
  \RBezier(\BeX,0)(\BeX,\BeY)(0,\BeY){1}{0.71}{1}[#3]<#4>
  \RBezier(0,\BeY)(-\BeX,\BeY)(-\BeX,0){1}{0.71}{1}[#3]*
  \RBezier(-\BeX,0)(-\BeX,-\BeY)(0,-\BeY){1}{0.71}{1}[#3]*
  \RBezier(0,-\BeY)(\BeX,-\BeY)(\BeX,0){1}{0.71}{1}[#3]*
  \fillpath
  \linethickness{#5}
  \RBezier(\BeX,0)(\BeX,\BeY)(0,\BeY){1}{0.71}{1}[#3]<#6>
  \RBezier(0,\BeY)(-\BeX,\BeY)(-\BeX,0){1}{0.71}{1}[#3]*
  \RBezier(-\BeX,0)(-\BeX,-\BeY)(0,-\BeY){1}{0.71}{1}[#3]*
  \RBezier(0,-\BeY)(\BeX,-\BeY)(\BeX,0){1}{0.71}{1}[#3]*
  \strokepath
}}\ignorespaces
}

```

Le loro sintassi sono le seguenti.

```

\Bcircle(<coordinate del centro>){<raggio>}[<numero di tratti>]%
  <colore interno>[<spessore del bordo>]<colore del bordo>

```

```

\Bellipse(<coordinate del centro>){<asse-x,asse-y>}[<numero di tratti>]%
  <colore interno>[<spessore del bordo>]<colore del bordo>%
  [<gradi di rotazione antioraria>]

```

Per entrambi le *<coordinate del centro>* sono espresse mediante un numero complesso; per la circonferenza il raggio è un numero reale, per l'ellisse gli assi sono due numeri separati da una virgola; il numero di tratti di default vale 50, ma nella figura si è espressamente indicato il numero 20 e ciò nonostante i contorni delle figure non sembrano formate da spezzate; il colori interno e del bordo di default sono il bianco e il nero rispettivamente, ma si possono specificare i colori che si preferiscono; lo spessore del bordo è predefinito al valore di 0.4pt, ma si può specificare il valore che si preferisce, ma non troppo spesso, come nella figura dove i bordi più spessi sono di 1mm; L'angolo di rotazione antioraria è predefinito al valore di 0°, ma l'utente può specificare il valore che preferisce, compresi valori negativi che corrispondono a rotazioni orarie, e nella figura si è specificato il valore di 45°.

4. Estensione dei risultati ottenuti

Bisogna ammettere che le curve di Bézier intere e quelle razionali hanno proprietà molto diverse; secondo chi scrive la teoria delle coniche complementari lascia qualche dubbio, ma i disegni vengono molto bene. Ci si può domandare se le curve che si ottengono sono graficamente più o meno precise di quelle che si ottengono senza appoggiarsi ai polinomi di Bernstein. Nella figura 9 si vedono due ellissi disegnate con le spline di Bézier intere realizzate con le macro interne del pacchetto `pict2e` (richiamato dal pacchetto `curve2e`, che d'altronde viene richiamato anche dalle macro descritte in questo rapporto) e con le funzioni di Bézier descritte mediante i polinomi di Bernstein.

Nella figura 9 ci sono tre ellissi; quella celeste piena è disegnata con quattro archi di ellisse usando la macro `\RBezier`; quella rossa sottile è disegnata con archi di parabola mediante un solo comando `\Qurve` del pacchetto `curve2e`; quella nera sottile, quasi perfettamente sovrapposta al contorno dell'ellisse celeste, è disegnata con un solo comando `\Curve` che usa spline intere cubiche di Bézier. Si vede la differenza con le curve quadratiche, decisamente più semplici, ma con gli archi di parabola che si allargano troppo rispetto alla linea desiderata; prese isolatamente le linee quadratiche sono accettabili, ma a confronto con le linee cubiche o con le linee ottenute con i polinomi di Bernstein sono da scartare. Invece le spline cubiche intere di Bézier o le spline quadratiche razionali producono sostanzialmente gli stessi effetti. Le differenze sostanziali risiedono nel numero totale di calcoli da eseguire.

Una cosa si può affermare con certezza: le curve disegnate con le funzioni intere o razionali di Bézier basate sui polinomi di Bernstein richiedono un tempo sensibile per essere disegnate, perché per disporre di tanti brevissimi segmenti che non diano l'impressione di essere di fronte ad una spezzata, bisogna ripetere molte volte l'uso del comando di `pict2e` `\lineto`, ricalcolando ogni volta le coordinate di ogni estremo. Con i comandi `\curveto` e, raramente, di `\lineto`, bisogna calcolare le coordinate di archi o linee più lunghe, senza bisogno di ripetere molte volte i comandi per descrivere le coordinate di queste componenti grafiche. Per disegnare una circonferenza (o un'ellisse) mediante `\curveto` e con grande precisione bastano otto archi di 45° ciascuno per un totale di 9 coordinate e 9 direzioni; per disegnare con le macro basate sui polinomi di Bernstein un'intera circonferenza con quattro archi di 90° ciascuno, bisogna determinare circa 400 coordinate per usare circa 400 brevi segmenti tali da nascondere la natura di spezzata del disegno.

Se proprio si vogliono usare le macro descritte in questo testo, nonostante i tempi di esecuzione lunghi e la necessità, talvolta, di cercare empiricamente i valori dei pesi o le coordinate dei punti guida, l'utente può definirsi altre macro che richiedano meno argomenti,

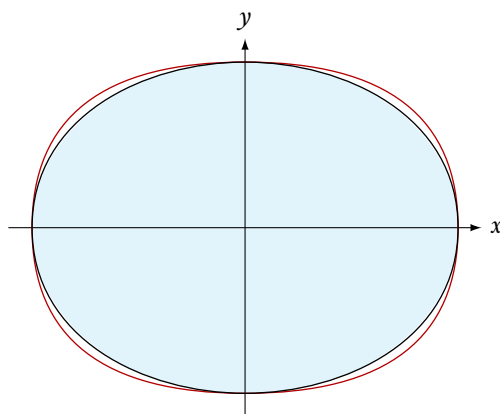


Figura 9. Confronto fra un'ellisse (celeste), disegnata con le funzioni di Bézier basate sui polinomi di Bernstein, con la stessa ellisse (solo contorno) disegnata con le macro di `curve2e`.

ma che contengano le varie funzioni di Bezier fratte e corredate delle necessarie impostazioni, in modo da usare una sintassi più comoda di quella esposta sopra. La cosa è abbastanza semplice per disegnare circonferenze e cerchi colorati internamente; è anche semplice fare una cosa simile per disegnare ellissi, anche con gli assi principali ruotati. Ma secondo l'esperienza di chi scrive, quanto esposto ha un interesse dal punto di vista matematico, ma non è molto comodo per disegnare.

5. Conclusioni

Il pacchetto `lapdf` esegue tutte queste cose direttamente; i risultati sono molto belli, anche se l'autore ha creato questo pacchetto prima che fossero disponibili i pacchetti `xparse` e `xfp`; si è dovuto accontentare del pacchetto `calc` che lavora con l'aritmetica intera nativa del programma $\text{\TeX}$. Per lavorare con numeri fratti, li usa come coefficienti della lunghezza di un punto, che sostanzialmente lavora a virgola fissa, con soli 16 bit destinati alla parte fratta; questi corrispondono a circa 5 o 6 cifre decimali fratte, per cui i troncamenti e gli arrotondamenti fanno calare sensibilmente la precisione dei calcoli.

Nonostante questi inconvenienti l'autore di `lapdf` ha ottenuto dei bellissimi disegni. La cartella che contiene la documentazione di `lapdf`, contiene anche un certo numero di file `.tex` che l'utente può scaricare sul proprio elaboratore e compilare sulla propria macchina; i tempi di esecuzione sembrano minori rispetto a quelli che si sono incontrati nello svolgere questa analisi, ma sono estremamente interessanti anche come suggerimenti per crearsi delle macro personali.

Qui si ringrazia Detlef Reimers, il creatore del pacchetto `lapdf`, che ci ha permesso di entrare in un settore sconosciuto molto bello e interessante.

Claudio Beccari
RIVOLI (TO)
claudio(dot)beccari(at)gmail(dot)com

Concorso: Elaborazione di dati per determinare e disegnare la retta di regressione

GJIT Members

Sommario Ormai dopo diverso tempo, al concorso indetto dai membri del GJIT non ha partecipato nessuno; peccato, perché il problema oggetto del concorso era relativamente semplice, ma richiedeva una certa abilità per essere risolto. Questo articolo mostra una possibile soluzione che ricorre solamente ai comandi del nuovo linguaggio L^AT_EX3.

Abstract After some time since the context was announced, nobody participated. The problem proposed was relatively simple, although it required a certain level of ingenuity to solve it. This article describes a possible solution that uses only some commands of the new language L^AT_EX3.

1. Introduzione

Il concorso richiedeva di determinare la retta di regressione che interpolasse al meglio un insieme di dati sperimentali assegnato mediante una lista che qui si ripete (vedi tabella 1).

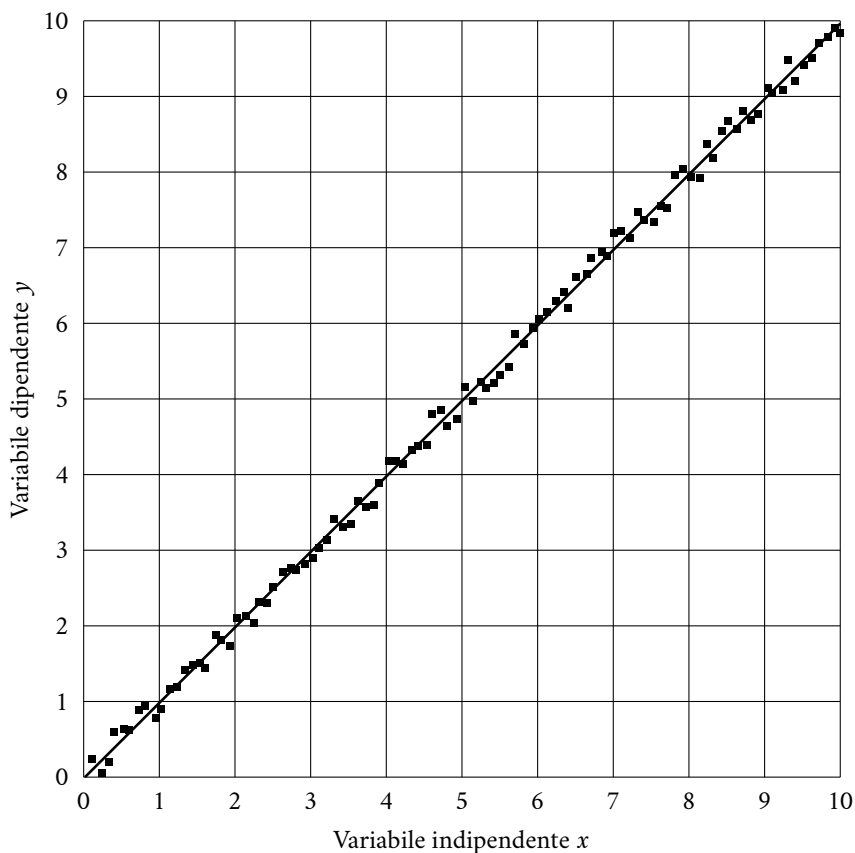
Tabella 1. I dati sperimentali

0.104 0.243,0.237 0.052,0.331 0.201,0.401 0.592,0.537 0.638,0.603 0.617,
0.725 0.885,0.807 0.943,0.950 0.782,1.020 0.905,1.139 1.162,1.232 1.192,
1.343 1.413,1.439 1.486,1.539 1.506,1.608 1.437,1.746 1.884,1.819 1.807,
1.928 1.729,2.029 2.098,2.142 2.135,2.249 2.033,2.320 2.313,2.423 2.305,
2.509 2.512,2.636 2.712,2.734 2.768,2.808 2.740,2.925 2.818,3.034 2.896,
3.109 3.035,3.213 3.136,3.314 3.411,3.434 3.309,3.534 3.344,3.621 3.650,
3.735 3.566,3.831 3.600,3.902 3.887,4.034 4.180,4.133 4.183,4.218 4.145,
4.346 4.329,4.418 4.384,4.537 4.386,4.601 4.797,4.719 4.854,4.803 4.638,
4.938 4.739,5.038 5.155,5.140 4.969,5.249 5.227,5.323 5.145,5.423 5.217,
5.507 5.321,5.616 5.426,5.706 5.855,5.823 5.733,5.933 5.937,6.013 6.057,
6.124 6.146,6.241 6.289,6.344 6.412,6.408 6.209,6.508 6.613,6.648 6.647,
6.706 6.863,6.846 6.945,6.918 6.885,7.007 7.199,7.102 7.227,7.218 7.132,
7.325 7.466,7.408 7.363,7.534 7.344,7.636 7.557,7.705 7.525,7.810 7.955,
7.925 8.035,8.024 7.933,8.146 7.928,8.245 8.374,8.320 8.183,8.442 8.547,
8.516 8.676,8.631 8.563,8.708 8.808,8.820 8.684,8.917 8.766,9.049 9.109,
9.103 9.048,9.237 9.082,9.305 9.482,9.408 9.207,9.520 9.411,9.625 9.506,
9.724 9.713,9.836 9.788,9.927 9.902,10.00 9.833,

I dati non devono necessariamente essere ordinati in nessun modo; in ogni caso le coppie sono separate da una virgola, mentre i due termini di ciascuna coppia sono separati da uno

spazio; il primo termine rappresenta la variabile indipendente normalizzata perché ricada nell'intervallo 0 — 10; il secondo termine rappresenta la variabile dipendente normalizzata nello stesso modo.

Il compito consiste nel tracciare il diagramma della figura 1.



La retta di regressione $y = ax + b$ ha $a = 0.998$ e $b = -0.017$

Figura 1. La retta di regressione $y = ax + b$ ha $a = 0.998$ e $b = -0.017$

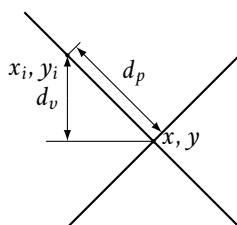


Figura 2. Distanza verticale d_v e distanza pitagorica d_p

2. Soluzione del problema

Il software da preparare deve eseguire diversi compiti.

1. Deve aprire il file dei dati sperimentali; per questo concorso l'utente deve aver copiato i dati esposti nella tabella 1 e deve averli salvati in un file di cui può scegliere arbitrariamente il nome. A titolo di esempio per questo articolo essi sono stati salvati nel file `DatiRegressione.tex`. In generale è necessario che lo sperimentatore abbia raccolto numerosi dati per il suo esperimento; comunque i dati devono essere almeno due, anche se non ha molto senso cercare una retta di regressione per soli due dati.
2. Leggendo le coppie di valori, il cui numero non è noto a priori, in quanto esperimenti diversi potrebbero richiedere un diverso numero di dati sperimentali, deve contare le coppie e possibilmente deve produrre un file ausiliario in cui queste coppie sono ordinate in ordine crescente di valori indipendenti, con ogni coppia di valori racchiusa fra parentesi tonde.
3. Terminata questa operazione e chiuso il processo di scrittura del file ausiliario, deve tracciare una retta attraverso la scia di dati.
4. Via via che si importano questi dati dal file ausiliario, bisogna determinare le distanze dei punti da una ipotetica retta $y = ax + b$ di cui non si conoscono ancora i suoi coefficienti. Invece di calcolare la distanza pitagorica di ciascun punto dalla retta, si può usare la distanza verticale, cioè la differenza delle ordinate fra il punto sperimentale e il punto della retta che si trova all'intersezione della retta incognita con la perpendicolare che passa per il punto sperimentale (vedi la figura 2); se la somma delle distanze verticali è minima, anche la somma delle distanze pitagoriche è minima.
5. La somma (dei quadrati) delle distanze verticali e la somma (dei quadrati) delle distanze pitagoriche sono legate fra di loro. Se si guarda la figura 2, si vede subito che se si minimizza l'altezza del triangolo rettangolo, si minimizza anche l'ipotenusa, visto che la retta $y = ax + b$ e l'ipotenusa sono perpendicolari fra di loro e questa proprietà vale per tutti i punti sperimentali.
6. Minimizzare la somma dei quadrati delle distanze verticali significa trovare dei valori di a e b che azzerano le derivate di quelle somme rispetto, appunto, ad a e b . Vale a dire che bisogna risolvere il sistema

$$\begin{cases} a \sum_i^N x_i^2 + b \sum_i^N x_i = \sum_i^N x_i y_i \\ a \sum_i^N x_i + bN = \sum_i^N y_i \end{cases}$$

7. Quindi via via che si esaminano le coppie di dati si determinano e si sommano i termini delle sommatorie suddette, dopo di che la soluzione del sistema diventa banale.

3. Il codice

Conviene precisare che è conveniente definire un file di macro personali, che per questo articolo si è chiamato `AnalizzaRegressione.sty`, per cui la parte di codice da usare per eseguire

i calcoli suddetti e per disegnare il diagramma si riduce a poche righe:

```
\begin{figure}
\AnalisiRegressione[DatiRegressione.tex]
\centering
\DiagrammaRegressione[10mm]

\bigskip

\caption{...}\label{...}
\end{figure}
```

Perciò qui si commentano alcune parti di codice di questo file .sty.

Si sottolinea il fatto che il codice presentato è solo una delle tante possibili soluzioni. Il vantaggio di questo codice è che ricorre prevalentemente a comandi di $\text{\TeX 2}_{\epsilon}$, e ad alcuni comandi di $\text{\TeX 3}$. Questi ultimi sono particolarmente comodi perché rendono molto più semplice la programmazione. Quasi tutti fanno già parte del nucleo di $\text{\TeX}$, quindi sono accessibili senza usare pacchetti esterni; altre parti sono facili definizioni mediante comandi in stile $\text{\TeX 2}_{\epsilon}$ di “funzioni” $\text{\TeX 3}$, che richiedono un ambiente particolare per accedere ai nuovi comandi, per esempio i seguenti.

Cominciamo con il preambolo del file .sty che specifica la versione del codice $\text{\TeX}$ necessaria per eseguire il codice; inoltre si è inserita anche la dichiarazione delle specificità del pacchetto; cosa raccomandabile per tutti i file, che siano file sorgenti di documenti, file di classe, file di pacchetti, eccetera,

```
% !TEX encoding = UTF-8 Unicode
% !TEX TS-program = pdflatex

\NeedsTeXFormat{LaTeX2e}[2018/01/01]
\ProvidesPackage{AnalizzaRegressione}%
[2023-11-21 v,1.0 Per analizzare dati e disegnare la retta di regressione]
```

I primi due commenti servono per ricordare con quale codifica è stato salvato il file, e con quale programma può essere usato; la successiva riga specifica la data minima della versione del formato di $\text{\TeX}$ nel nostro caso si tratta di un codice in linguaggio $\text{\TeX 2}_{\epsilon}$ e il formato deve essere stato prodotto in data non anteriore a quella indicata, cioè il primo di gennaio 2018. Questo tipo di informazioni dovrebbe essere presente in ogni file da usare con il sistema $\text{\TeX}$, anche i file creati dall’utente finale.

Seguono poi delle definizioni di comandi destinati all’utente finale, che ricorrono direttamente a “funzioni” del linguaggio $\text{\TeX 3}$. Il primo comando `\fptest` esamina lo stato di una espressione booleana indicata nel primo argomento, e se l’espressione è vera esegue il codice indicato come secondo argomento, altrimenti esegue quello contenuto nel terzo argomento.

Il secondo comando `\fpdowhile` serve per eseguire ripetutamente il codice indicato come secondo argomento, finché l’espressione booleana del primo argomento è vera. Ovviamente il codice da ripetere deve contenere anche almeno un comando che modifica qualche informazione usata nell’espressione booleana affinché questa diventi falsa.

Si è usato il comando di definizione `\ProvideExpandableDocumentCommand` perché quel comando potrebbe già essere definito da altri pacchetti, se lo fosse, il comando usato non farebbe niente.

```
\ExplSyntaxOn
\ProvideExpandableDocumentCommand\fpctest{m m m}{%
  \fp_compare:nTF{#1}{#2}{#3}\ignorespaces}
%
\ProvideExpandableDocumentCommand\fpdowhile{m m}{%
  \fp_do_while:nn{#1}{#2}\ignorespaces}
\ExplSyntaxOff
```

Si definisce poi il comando `AnalisiRegressione`:

```
\newread\Idata
%
\NewDocumentCommand\AnalisiRegressione{o}{%
\IfNoValueTF{#1}%
  {\typeout{Manca il nome del file dei dati.}\@@end\end}%
  {\IfFileExists{#1}%
    {\def\ReadFile{#1}}%
    {\typeout{Il file #1 non esiste in questa cartella.}\@@end\end}%
  }%
%
\immediate\openin\Idata\ReadFile
\read\Idata to\PlainData\immediate\closein\Idata
\edef\InputData{}\expandafter\splitplaindata\PlainData!
}%
```

che fa diverse cose; inoltre questo comando è quello che deve usare l'utente finale per informare il programma su come si chiama il file che contiene i dati da analizzare. Questa informazione è un argomento facoltativo; un po' strano? è un trucco che serve per obbligare l'utente finale a ricordarsi che il comando richiede un argomento. Se l'utente finale se ne dimentica, gli viene mostrato un messaggio sulla console che dice che il nome del file manca; se ha sbagliato a scrivere il nome il programma informa l'utente che il file con quel nome non esiste; in entrambi i casi il programma si arresta immediatamente, altrimenti assegna questo nome ad una macro `\ReadFile` che poi legge usando il canale di lettura `Idata`. Questa lettura avviene con i comandi successivi che ricorrono a comandi nativi di $\text{\TeX}$. Una volta letto tutto il file dei dati da elaborare, lo analizza separando le coppie di valori mediante il comando `\splitplaindata` che trasferisce le singole coppie di dati racchiudendole fra parentesi tonde nel contenitore `\InputData` precedentemente inizializzato come contenitore vuoto. `\splitplaindata` è definito come segue ricorrendo ad un comando di definizione nativo di $\text{\TeX}$ che accetta argomenti delimitati.

```
\def\splitplaindata#1 #2,#3!{\edef\InputData{\InputData(#1,#2)}%
\ifblank{#3}%
  {\relax}%
  {\edef\PlainData{#3}\expandafter\splitplaindata\PlainData!}}
```

Il comando `\ifblank` è definito dal pacchetto `etoolbox` molto comodo da usare quando si scrive del codice.

Seguono poi alcuni comandi per estrarre ciascun termine dalle coppie di dati.

```
\def\SplitCoord#1,#2!{\edef\@Tx{#1}\edef\@Ty{#2}}%
\def\GetCoord(#1)#2#3{\expandafter\SplitCoord#1!%
\edef#2{\@Tx}\edef#3{\@Ty}}%
```

Il comando seguente estrae dalla lista di coppie di dati, conservati nella macro `\InputData`, solo la prima coppia e controlla se è seguita da altri dati, per cui ridefinisce se stessa finché ci sono dati da elaborare.

```
\def\SplitCoord#1,#2!{\edef\@Tx{#1}\edef\@Ty{#2}}%
\def\GetCoord(#1)#2#3{%
\expandafter\SplitCoord#1!\edef#2{\@Tx}\edef#3{\@Ty}}%
```

Poi viene davvero l'elaborazione dei dati; il codice che segue definisce il segno per mettere il pallino, o il quadratino (predefinito), o quel che l'utente vuole; poi riceve uno o due dati, ricordando che ogni dato è costituito da una coppia di valori.

```
\ProvideDocumentCommand\Regression{0{\rule{1mm}{1mm}} r() r()}{%
\def\@tA{0}\def\@tB{0}\def\@tC{0}\def\@tD{0}\def\@tE{0}%
\def\@tF{0}%
\def\@segnino{#1}%
\ElaboraDatiUno(#2)\ElaboraDatiUno(#3)\ElaboraDatiDue}%
```

Le funzioni di elaborazione `\ElaboraDatiUno`, `\ElaboraDatiDue` nonché la terza funzione `\ElaboraDatiTre` vengono poi definite dai codici seguenti.

```
\def\ElaboraDatiUno(#1){\GetCoord(#1)\tX\tY
\put(\tX,\tY){\makebox(0,0)[cc]{\@segnino}}%
\edef\@tA{\fpeval{\@tA+\tX^2}}\edef\@tB{\fpeval{\@tB+\tX}}%
\edef\@tC{\@tB} \edef\@tD{\fpeval{\@tD+1}}%
\edef\@tE{\fpeval{\@tE+\tX*\tY}}\edef\@tF{\fpeval{\@tF+\tY}}%
}%
%
\ProvideDocumentCommand\ElaboraDatiDue{d()}{%
\IfValueTF{#1}{\ElaboraDatiUno(#1)\ElaboraDatiDue}{\ElaboraDatiTre}}%
%
\def\ElaboraDatiTre{\edef\Rd{\fpeval{\@tD*\@tA - \@tB^2}}\relax
\xdef\Ra{\fpeval{round((\@tD*\@tE - \@tF*\@tB)/\Rd,3)}}\relax
\xdef\Rb{\fpeval{round((\@tA*\@tF - \@tB*\@tE)/\Rd,3)}}\relax
}%
```

Il primo comando, `\ElaboraDatiUno`, riceve un argomento formato da una coppia di dati racchiusi fra parentesi tonde; li separa e li elabora avendo per ogni dato separato l'ascissa x_i dall'ordinata y_i al fine di accumulare sommandoli i vari termini che servono per risolvere il sistema che consente di determinare i coefficienti a e b della retta di regressione; finita l'elaborazione di una coppia, la macro `\Regression` chiama la coppia di comandi `\ElaboraDatiUno` col

terzo argomento e `\ElaboraDatiDue` senza specificargli nessun argomento. Il comando provvede anche a mettere il segno (quadratin, circoletto, o altro scelto dall'utente) nel diagramma richiesto come risultato finale.

È appunto `\ElaboraDatiDue` che trova il suo argomento racchiuso fra parentesi tonde nel flusso di dati in entrata. Se effettivamente questo argomento esiste, riesegue una coppia di comandi come l'ultima coppia che esisteva prima alla fine del comando `\Regressione`, producendo così un processo iterativo, solo finché `\ElaboraDatiDue` riceve effettivamente un argomento non vuoto; se questo succede, e succede alla fine della lista di dati, allora attiva il comando `\ElaboraDatiTre` che provvede a terminare i calcoli, cioè a determinare i parametri a e b della retta di regressione, conservandone i valori nelle macro `\Ra` e `\Rb`.

L'ultima macro serve per disegnare la retta di regressione evitando di disegnare la parte di retta che cade fuori dal disegno; i test che vi si compiono servono appunto per definire i limiti orizzontali e verticali del disegno e verificare se il punto di partenza e il punto di arrivo della retta sono interni o esterni al diagramma; nel caso, il punto di partenza e/o il punto di arrivo vengono ricalcolati per essere al limite del disegno.

```
\ProvideDocumentCommand\RettaRegressione{%
\edef\Pzx{0}\edef\Pzy{11}\relax\edef\Pux{10}
\edef\Puy{\fpeval{\Ra*\Pux+\Rb}}\relax
\fpptest{\Rb<0}%
{\edef\Pzx{-\fpeval{\Rb/\Ra}}\edef\Pzy{0}}%
{\edef\Pzx{0}\edef\Pzy{\Rb}}\relax
\fpptest{\Pzy<\Puy}%
{\edef\Pux{\fpeval{(\Pzy-\Rb)/\Ra}}\edef\Puy{\Pzy}}%
{\edef\Pux{10}\edef\Puy{\Puy}}\relax
{\linethickness{1pt}\segment(\Pzx,\Pzy)(\Pux,\Puy)}}%
```

Per eseguire per davvero il diagramma, si definisce il secondo comando per l'utente, `\DiagrammaRegressione`, che accetta un argomento facoltativo, cioè il fattore di scala del disegno; sul suo valore preimpostato vale 1cm; siccome le variabili indipendente e dipendente sono normalizzate per restare nell'intervallo da 0 a 10, vuol dire che il disegno viene largo e alto 10 cm. Siccome sotto il disegno, oltre alla graduazione, si scrive anche l'equazione della retta di regressione, bisogna pensare che l'altezza complessiva sia maggiore almeno di 1 cm.

```
\newcount\RgrCount
\newcommand\DiagrammaRegressione[1][1cm]{%
\unitlength=#1\relax
\begin{picture}(10,11)(0,-1)
\multiput(0,0)(1,0){11}{\line(0,1){10}}%
\RgrCount 0\relax
\put(5,-0.7){\makebox(0,0)[tc]{Variabile indipendente $x$}}
\multiput(0,-0.2)(1,0){11}{\makebox(0,0)[tc]{\the\RgrCount}}
\global\advance\RgrCount by1\relax}%
\multiput(0,0)(0,1){11}{\line(1,0){10}}%
\RgrCount=0\relax
\put(-0.65,5){%
\rotatebox{90}{\makebox(0,0)[bc]{Variabile dipendente $y$}}}%
}
```

```

\multiput(-0.2,0)(0,1){11}{\makebox(0,0)[rc]{\the\RegrCount}%
\global\advance\RegrCount by1\relax}%
\expandafter\Regressione\InputData
\RettaRegressione
\end{picture}
%\put(0,-2){
\parbox{\linewidth}{~\\centering\small
  La retta di regressione  $y=ax+b$  ha  $a=\Ra$  e  $b=\Rb$ }%
}

```

I risultati che si ottengono con questo codice sono mostrati nella figura 1 e direi che è valsa la pena creare il pacchetto .sty appena descritto, perché non si tratta di codice banale, e ricostruirne il codice ogni volta che occorre per disegnare un insieme di dati e la retta di regressione che li approssima diventerebbe abbastanza pesante.

4. Conclusioni

Non essendoci stati concorrenti, non ci sono premi da assegnare. Perciò qui si è mostrata solo una possibile soluzione; non tanto semplice, ma nemmeno troppo complicata; sembrerebbe che con i comandi Lua, usabili con Lua $\LaTeX$, si potrebbe fare di meglio.

Certo ci sono gli utenti affezionati a $\LaTeX$ 2 ϵ e $\LaTeX$ 3 che parteggerebbero per la soluzione mostrata, e gli utenti affezionati a Lua $\LaTeX$ parteggerebbero per una soluzione basata su Lua, ma nessuno ha partecipato al concorso. Certo Lua $\LaTeX$ è formidabile, ma è difficile da imparare, per cui al momento gli utenti di questo motore di composizione sono pochi rispetto a coloro che usano $\LaTeX$ 2 ϵ .

G_UIT Members
G_UIT
arstexnica@guitex.org

Concorso: *La spirale delle radici quadrate*

Il G_UIT

Sommario Questo è il bando di “Concorso” fra i lettori di *Ar_sTeX_nica* per creare il miglior codice adatto per disegnare la spirale delle radici quadrate dei numeri naturali.

Abstract This is the announcement of a “Contest” among the *Ar_sTeX_nica* readers to find out the best code to draw the natural numbers Square Root Spiral.

Seguendo una certa tradizione di *Ar_sTeX_nica*, viene bandito un concorso, lo scopo del quale è vedere se i lettori di *Ar_sTeX_nica* sono in grado di usare le capacità grafiche e di calcolo di *TeX*; per “*TeX*” qui si intende il linguaggio, non il motore di compilazione; il linguaggio può essere *TeX*2_ε e/o *TeX*3; il motore può essere pdf*TeX*, *X_YTeX* o Lua*TeX*.

Bisogna creare il disegno della spirale delle radici quadrate dei numeri naturali, formata da una sequenza di punti ottenibile geometricamente riportando in grafico l'estremità del cateto lungo di un triangolo rettangolo con un vertice nell'origine; il cateto corto è di lunghezza costante “unitaria”, e l'ipotenusa di ogni triangolo diventa il cateto lungo del triangolo successivo. Se il cateto lungo del primo triangolo è unitario, la sua ipotenusa vale $\sqrt{2}$, quella del secondo triangolo vale $\sqrt{3}$, e via di questo passo per le radici dei successivi numeri interi.

Su questa spirale di punti hanno lavorato fior di matematici per trovare una curva che passasse per tutti i punti di questa spirale; sono arrivati a trovare un legame fra questi punti e la funzione Γ , ma il concorso non richiede di disegnare anche la spirale basata su questa funzione; il concorso richiede solo il disegno della sequenza di un numero di punti tali da disegnare almeno tre spire della spirale.

Vincerà il concorso il codice più breve; cioè contenente il minor numero di statement; per statement, qualunque sia il linguaggio usato, si intende la singola istruzione con i suoi argomenti; il codice deve essere compilabile con pdf*TeX*, o con *X_YTeX*, o con Lua*TeX*; sono esclusi Plain *TeX*, Con*TeX*t, e gli altri programmi che non usano il linguaggio *TeX* nelle sue varie incarnazioni. Dentro un codice da compilare con Lua*TeX* è consentito usare codice Lua; le definizioni di macro o di comandi Lua, vengono contate come statement singoli o multipli a seconda di quanti punti e virgola contengono. Si possono usare pacchetti per la grafica e per il calcolo fra tutti quelli distribuiti con il sistema *TeX*; le macro definite in questi pacchetti vengono contate per un singolo statement. Evidentemente le linee di commento non vengono contate.

Il Comitato Scientifico del G_UIT valuterà i lavori dei concorrenti, stilerà una graduatoria, e premierà il vincitore con un testo relativo all'uso avanzato del sistema *TeX*.

La scadenza per presentare il proprio codice è fissata 15 giorni prima della scadenza per inviare contributi per il prossimo numero di *Ar_sTeX_nica*. I codici relativi al concorso vanno inviati all'indirizzo arstexnica@guitex.org

Il G_UIT

arstexnica@guitex.org

Progetto grafico di copertina
a cura di Ivan Valbusa

Immagine di copertina:
Typesetting in wood
(Raphael Schaller, 2016)

<https://unsplash.com/photos/GkinCd2enIY>

Questa rivista è stata prodotta
dal Gruppo Utilizzatori Italiani di T_EX
usando esclusivamente software libero.

Versione elettronica per la diffusione via web.



4_{rs}TeX_{nica} – Call for Paper

La rivista è aperta al contributo di tutti coloro che vogliano partecipare con un proprio articolo. Questo dovrà essere inviato alla redazione di 4_{rs}TeX_{nica}, per essere sottoposto alla valutazione di recensori. È necessario che gli autori utilizzino la classe di documento ufficiale della rivista; l'autore troverà raccomandazioni e istruzioni più dettagliate all'interno del file d'esempio (.tex).

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

Di volta in volta verrà fissato, e reso pubblico sulla pagina web <http://www.guitex.org/arstexnica/>, un termine di scadenza per la presentazione degli articoli da pubblicare nel numero in preparazione della rivista. Tuttavia gli articoli potranno essere inviati in qualsiasi momento e troveranno collocazione, eventualmente, nei numeri seguenti.

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

