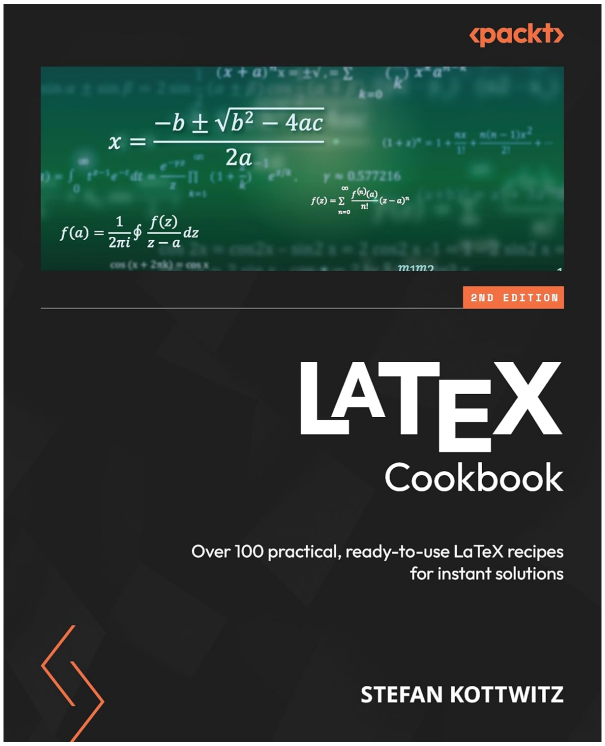
Book review: *LaTeX Cookbook*, second edition, by Stefan Kottwitz

Uwe Ziegenhagen

Stefan Kottwitz, *LaTeX Cookbook*, February 2024, 424pp., ISBN 9781835080320. Available in print and electronic formats. www.packtpub.com/en-us/product/latex-cookbook-9781835080320


‹packt›
2ND EDITION
LATEX
Cookbook
Over 100 practical, ready-to-use LaTeX recipes for instant solutions
STEFAN KOTTWITZ

1 The author

Stefan Kottwitz is well known, foremost but not only in the German-speaking TeX world. After his mathematics studies in Jena and Hamburg he has worked as a network and IT security engineer in the maritime industry, equipping cruise ships with network equipment. Besides this full-time job he manages several websites on TeX and LaTeX, among them LaTeX.org, goLaTeX.de, TeXwelt.de, and tikz.net.

As if this was not enough already, he has written several books on LaTeX-centered topics, authoring the *LaTeX Beginner's Guide* (in two editions), a book on TikZ (LaTeX Graphics with TikZ; a Japanese version of it coming in November 2024) and the *LaTeX Cookbook* second edition which we will review today. I have known Stefan for more than fifteen years now, so I happily accepted the role of the reviewer.

2 The book

In February 2024 the second edition of Stefan Kottwitz's LaTeX Cookbook was published. The second edition of the LaTeX Cookbook features 100 LaTeX recipes on more than 400 pages, so in comparison with the first edition the amount of text has grown. It is also printed in color while the first edition's images had been printed in grayscale. The book is divided into 13 chapters:

1: Exploring Various Document Classes
The first chapter introduces the basic concepts of LaTeX: What "document classes" and "packages" are, and how one can write basic documents such as a small article, thesis or book, or more specialised ones such as a CV, leaflet or poster.

2: Tuning the text
This chapter contains various recipes that deal with the text: its shape, its hyphenation as well as the visualisation of the layout or the typesetting in a grid.

3: Adjusting Fonts
Chapter 3 introduces basic font commands, how to switch a font locally and globally, the output of font tables and the handling of ligatures.

4: Creating Tables
Tables, tables, tables, . . . From the design of a table to the positioning to the individual merging of rows and columns, this chapter contains these recipes.

5: Working with Images
This chapter shows how to get the optimal quality, the customization of images, the various shapes and the arrangement of several graphics next to each other.

6: Creating Graphics
This chapter is a small extract of Stefan's book on TikZ. It does not introduce the general concepts of TikZ or PGF, but focuses more on practical diagram types: flowcharts, trees, bar and pie charts, etc.

7: Creating Beautiful Designs
While LaTeX is mainly used to typeset rather formal standard documents such as articles and books, non-standard "beautiful" designs can be implemented with LaTeX as well. This chapter shows how to add background images, create beautiful ornaments and pretty headings, produce a calendar or build a word cloud.

8: Contents, Indexes, and Bibliographies
This chapter deals with more formal aspects of authoring a document. It shows recipes to tweak the table of contents, create and maintain an

doi.org/10.47397/tb/45-3/tb141reviews-kottwitz-cookbook2

index or use LaTeX's magnificent mechanisms for bibliographies.

9: Optimizing PDF Files
Chapter 9 shows how PDF files can be optimized with respect to hyperlinks, metadata or copyright information. It also shows how PDFs can be tweaked for electronic books.

10: Writing Advanced Mathematics
This chapter not only describes advanced math typesetting features but also introduces the main concepts briefly. This part is followed by recipes on the fine-tuning of mathematical content, the plotting of functions in two and three dimensions and mathematical calculations with LaTeX.

11: Using LaTeX in Science & Technology
Chapter 11 is dedicated to the use of LaTeX in science. It covers typesetting algorithms and source codes, the programming with Lua as well as the drawing of molecules, Feynman diagrams, atom representations and several other applications required in MINT sciences.

12: Getting Support on the Internet
In this chapter Stefan explains how to get help for when LaTeX is not producing what the user wants. It introduces various help forms and shows how to produce minimal working examples to get more effective help.

13: Using AI with LaTeX
Since "Artificial Intelligence" is a hot topic, this book needs to have a dedicated chapter as well :-). Stefan shows how ChatGPT can be used to create example LaTeX code, e.g. for bibliographies or more complex math, or to improve one's own LaTeX documents.

3 Conclusion

While I may not be a member of the main target audience of this book any more — that is likely colleague and university students or other people still in the process of learning LaTeX — even I enjoyed reading and reviewing this book.

It shows best practises for a wide range of LaTeX applications and does so quite well. For more experienced LaTeXnicians, this book presents a good overview of how certain requirements can be handled with LaTeX.

◇ Uwe Ziegenhagen
Escher Str. 221
50739 Cologne
Germany
ziegenhagen (at) gmail dot com
https://www.uweziegenhagen.de

Hints & Tricks

Production notes: PDFs and urls

Karl Berry

Documents commonly include PDF, PNG, and JPEG files as images. It's straightforward to extract the latter two from a composed PDF, using, for instance, `pdfimages` from Poppler[1] (or Xpdf). But I wanted to extract an included PDF back out of a document PDF recently, and was surprised to find that there seemed to be no standard tool for this.

I asked Max Chernoff, fellow *TUGboat* TeXnician. He couldn't find anything either, so he wrote a LuaTeX script for the job. It's invoked like this:

```
./luatex-xobject-tub.tex somedoc.pdf
```

The output is written to `luatex-xobject-tub.pdf`, one page for each included PDF. It's available in the *TUGboat* repository.[2]

Another recurring job with PDFs is to check the live urls that are present. (We like to do this before each issue goes to press, so at least we know the urls are working at that time.) Max again came up with a solution, essentially using:

- `qpdf --qdf` (`github.com/qpdf`) for an ASCII transliteration of the PDF;
- `grep --only-matching` to extract the urls, and
- `wget --spider` to check them.

The full script is `check-pdf-urls-tub` in the same *TUGboat* repository directory. The license on these scripts is "do what you want to". Thanks Max!

That first step, converting a compressed PDF into a human-readable form, is generally needed from time to time — i.e., a `pdftype` program analogous to `dvitype` et al. Some other methods I know of:

- Use LaTeX: `\DocumentMetadata{uncompress}`
- Use pdfTeX:
  ```
  \pdfcompresslevel=0 \pdfobjcompresslevel=0
  \immediate\pdfximage{in.pdf}%
  \pdfrefximage\pdflastximage \end
  ```
- `pdftk in.pdf output out.pdf uncompress`
- `mutool clean -d in.pdf`

Each has its own benefits and drawbacks, and there are surely yet more out there; I'd appreciate further information.

◇ Karl Berry
github.com/TeXUsersGroup

[1] poppler.freedesktop.org
[2] github.com/TeXUsersGroup/tugboat/blob/trunk/misc/

doi.org/10.47397/tb/45-3/tb141prod