

[Journal](#)
[home page](#)
[General](#)
[information](#)
[Submit an](#)
[item](#)
[Download](#)
[style files](#)
[Copyright](#)
[Contact us](#)

THE PracTeX Journal



Issue Table of Contents

2007, Number 1

[Published 2007-02-20]

Notices

[From the Editor:](#) In this issue; Next issue: TeX for Editors; A Practical Approach to Graphics
Yuri Robbers

[Feedback](#)

From the Readers

[News from Around:](#)

Videos of the Grand Wizard
Kern Shirt
LaTeXpedia
LaTeX Tutorial Days
Recent Release: The 2007 TeX Live and *TeX Collection*
Recent Release: PCTeX 6
The Editors

[Whole Issue PDF for PracTeX Journal 2007-1](#)

The Editors

Articles

[Graphics in LaTeX](#)

Claudio Beccari

[Graphics with PGF and TikZ](#)

Andrew Mertz and William Slough

[Square concepts](#)

Dirk Baechle

[Creating book covers with PSTricks](#)

Yuri Robbers and Annemarie Skjold

[Commutative Diagrams with XY-pic II. Frames and matrices](#)

Paul Blaga

[Create Trees and Figures in Graph Theory with PSTricks](#)

Manjusha S. Joshi

[LaTeXpedia: the future of LaTeX documentation](#)

Lapo Filippo Mori

[Tables in LaTeX2e: Packages and Methods](#)

Lapo Filippo Mori

[Babel, how to enjoy writing in different languages](#)

Enrico Gregorio

[Hacking DVI files](#)

Jin-Hwan Cho

[Writing and checking complete proofs in TeX](#)

Bob Neveln and Bob Alps

Columns

[Travels in TeX Land: The post-typesetting phase of producing a book](#)

David Walden

[The "hacking for learning" paradigm in LaTeX — Some thoughts by a long-time LaTeX user](#)

S. Parthasarathy

[Ask Nelly:](#)

How do I create European style spacing within numbers?

The Editors

[Distractions — Some chess problems created in LaTeX](#)

The Editors

[Journal](#)
[home page](#)
[General](#)
[information](#)
[Submit an](#)
[item](#)
[Download](#)
[style files](#)
[Copyright](#)
[Contact us](#)

THE PracTeX Journal



From the Editor: In this issue; Next issue: TeX for Editors; A Practical Approach to Graphics

Yuri Robbers

- [Comment on this paper](#)
- [Send submission idea to editor](#)

[In this issue](#)

[Next issue: TeX for Editors](#)

[Thanks](#)

[Editorial: A practical approach to graphics](#)

In this issue

This 2007-1 issue marks the beginning of the Journal's third year. It also marks my debut as guest editor-in-chief, an opportunity Lance Carnes gracefully offered me, and for which I am grateful. My role for this issue has been the catalyst in discovering new tricks in LaTeX and HTML as well as in getting to know and working with several inspiring people.

The theme for this issue is "Graphics in TeX". Even though TeX was not originally designed for graphical work, it has acquired a lot of functionality in this area over the years. ConTeXt is especially noteworthy in this respect since it offers seamless integration of TeX and Metafont.

In addition to Metafont many other graphical add-on packages exist that extend the capabilities of the TeX-engine, such as PSTricks and PGF with TikZ. This issue introduces various of these packages and shows some of their many applications.

The papers in this issue are both individual submissions and selections from the presentations at [last summer's PracTeX06 conference](#).

The articles at the top of the table of contents have something to do with the theme, and the ones further down deal with other subjects. Within both sets the first articles are for a general audience and beginning/intermediate users. As you progress towards the bottom of each

category there are articles on more advanced topics. If you are an expert user or a beginning/intermediate user who would like a challenge, try reading about and possibly doing the projects described by these authors.

The PracTeX Journal now offers an RSS feed. If you are not familiar with RSS (an easy way to "subscribe" to a web site) see [this introduction](#). The RSS link is an orange "XML" icon on the home page.

Next issue

Theme: **TeX for editors**

Editor: Paul Blaga

Most people use TeX, LaTeX or ConTeXt mainly for writing their own scientific articles. They don't need skills for fine tuning the visual aspect of their work, because this is what the editor of the journal does. The same goes for the books, if the publisher has an in-house TeX expert. However, sometimes the user needs more knowledge, for instance if he has to edit a volume or has to provide a camera-ready version of his book that has to meet certain standards. This issue of PracTeX is devoted to aspects of LaTeX that may be of help to these users. We have in mind, for instance, packages as geometry, caption, fancyhdr, but also explanations of the way somebody can change the aspects of chapters heads and others of the same kind, without becoming, really, a TeX wizard.

Of course we will also welcome any contributions that do not fit within the theme. Issue 2007-2 is scheduled to appear around May 20th. Please [contact me](#) if you have a topic for a "TeX for editors" article or other article. — Paul Blaga

Thanks

This issue is a big one, with about 15 articles. Many thanks to the authors who wrote and typeset the pieces that appear in this issue.

A large part of the organizational work of assembling this issue was done by production editors **Joe Hogg, Lance Carnes, Paul Blaga, Francisco Reinaldo, Will Robertson, and myself**. These editors spent many hours coordinating with authors, reviewers, and proofreaders to get each article into finished form. My special thanks to Paul Blaga who has done a lot of work, editing three articles, authoring one, helping out with production editing and getting the job done even if it is sprung on him at extremely short notice.

Many thanks also to the reviewers and proofreaders who checked the articles and sent comments and corrections.

Dave Walden has solved one or two problems I have encountered while keeping the journal's web site on track, even though Dave officially resigned as the Journal's web master in August 2006. Many thanks, Dave.

Editorial: A practical approach to graphics

The truly practical approach to any given job does not necessarily include using the best tool in existence for this job. Sometimes it is far more practical to use a tool you know well rather than investing time in learning a new tool. Conceivably you could even research the various tools in existence before deciding which tool you should learn to use. By then we have definitely left the realm of practical approaches.

This raises the question of when it is worthwhile to learn a new tool and when it is not. To answer that question I would like to explore the world of Graphics in TeX (any flavour) a bit deeper.

Suppose you are asked to give a presentation on a subject rather dear to your heart — let's say the question of whether or not fish sleep. Further suppose that you are a scientist and used to working with LaTeX, but having no prior experience with Microsoft PowerPoint. The logical choice would be to use LaTeX for this presentation, since learning the Beamer class is probably going to be relatively easy, and it is much easier to reuse LaTeX material that one has used before in publications with Beamer than it would be to reuse this same material within PowerPoint.

Creating the presentation turns out to be a breeze, but a nice background image, perhaps with some fish in faded colours that do not interfere with the text, is still lacking. Which tool should you use? If you have no prior experience creating graphics in LaTeX, but you are a little bit of a Photoshop wizard, my recommendation would be: use Photoshop. Don't spend time searching CTAN or the web, printing the PGF and TikZ manuals, learning the graphics language and applying it to this problem, since that is an awful lot of work for creating one picture. In Photoshop this background graphic can probably be produced in fifteen minutes, rather than the many hours needed to familiarize oneself with the intricacies and oddities of a completely new package, even if this package has been specifically designed to work well within a LaTeX environment.

Should you wish to give presentations often, or create a lot of graphics, then learning a new tool might become worthwhile. Suppose this lecture is widely acclaimed and you receive a prestigious grant to write the definitive standard work on sleep in fish. This book is obviously going to be hundreds of pages, and contain hundreds or even thousands of graphs. These graphs are based on work of other scientists, on your own experiments, on computer simulations, and on the output of various programs for statistical analysis. In other words: all these graphs are different. They use different fonts, different scales, different ways of indicating points of interest, etc.

In this case you cannot escape redrawing all graphs in a standardized way. Of course you could refer to your trusty statistical software of choice, but in this case I would not. TeX in most variations has a strong emphasis on separating content and form. Most users consider this to be one of its main advantages. In a document that has hundreds or thousands of graphs, you could truly benefit from this separation.

It is, for example, often considered best to have the fonts in the figures match the text fonts. But what if, near the end of the project, you discover that the publisher requires the text font to be Minion Pro rather than Palatino, because that is what the other volumes in the series use. If you have used statistical software such as [R](#) for all graphs, you would need to either redraw all

graphs, or end up with a book that uses Minion Pro for the main text and the immensely different Palatino in all figures. If, however, you have made the more auspicious choice of investing some time in learning MetaPost, PSTricks, PGF or almost any other TeX-friendly graphics language, then the change is a matter of changing the font definition for all graphics along with the definition for all text, and Bob's your uncle (as the British so eloquently put it).

Another example, this time from the viewpoint of publishers of self-published authors. Suppose one writes a book, and needs to design a cover. It is possible to use any TeX-friendly language, such as PSTricks, or MetaPost. If you are a ConTeXt user, the "built-in" MetaPost integration called MetaFun would be the obvious choice. If this, however, is a new endeavour, using an already familiar program such as Photoshop might again be a lot easier. But if you intend to publish a whole series of books, with covers that are somewhat standardized, then the benefits of using MetaPost, PSTricks or some such language become obvious.

A new book in the series will have the same size, except perhaps width of the spine. If in the original design one has defined a new length variable, spine width, for this purpose then the change is easily made. Similarly, the title, author and publisher on front cover and spine will at least have the same font at the same size, and some of these are even likely to remain the same. With a cover designed in a TeX-like language, the necessary changes are easily made, and the new cover will match the old cover perfectly in all the necessary aspects while in other aspects it can easily be adapted exactly as you desire.

So far I have pretended that you should only use one tool. In practice most people will use a combination of tools for any given task. You could, for example, design a book cover in PSTricks, but use a graphic that has been created in Photoshop. You could use statistical graphs in a book that have been created in R, but with labels and other bits of text that are being typeset and positioned using AMS-LaTeX. Nevertheless, the principles outlined in this editorial apply equally well in such a hybrid environment. Any project can potentially benefit from some thought on which tools to use. Sometimes learning a new tool will easily be worthwhile, but at other times you would be better advised to stick with what you are already familiar with. Also, even though I myself often learn new things just for the fun of it, in critical projects you should probably refrain from using new tools just for the fun of it. As former US government official Bert Lance and famous proponent of the "Practical Approach" put it: "If it ain't broke, don't fix it!"

Yuri Robbers
Editor

[Journal home page](#)
[General information](#)
[Submit an item](#)
[Download style files](#)
[Copyright](#)
[Contact us](#)

THE PracTeX Journal



Feedback

From the Readers

- [Comment on this paper](#)
- [Send submission idea to editor](#)

TPJ

The Mahajan paper in issue 2006-4, <http://tug.org/pracjournal/2006-4/mahajan/> was an outstanding article. I too am a Ph.D. student and T.A. who has been using LaTeX to write my Thesis, Dissertation, various reports, and project papers. This was the perfect article for me. I am going to start learning ConTeXt

Keith Jones

[Aditya Mahajan along with editor Joe Hogg did a great job on that article. Good to hear it was useful to you. — The Editors.]

TPJ

Please,

In the abstract look at the contact email address for yuri.robbers. As a piece of typesetting, it is not very attractive! Surely something is wrong here and I am not an expert. But I love Latex.

Joe Mc Cool

[Dear Joe, it seems you are referring to the abstract on the journal webpage, since there is no email address in the abstract of the PDF file. This email address is actually a graphics file that is included in the text. It sure looks ugly, but it is one of only very few ways to thwart programs that scavenge websites for e-mail addresses that can be used for spam. So sadly enough this ugliness seems necessary. As a consolation it only shows up in some cases, depending on the width of your web browser window. — The Editors]

TPJ

In the article by Singh et. al, <http://tug.org/pracjourn/2006-4/singh/>. parts of the three reports have almost identical words. This should have been caught by the editor.

I am a LaTeX fan of long standing and always read new-user reports with great interest. I've had my own battles trying to get it accepted over M\$ Void, so I know the drill.

Charles L. Hethcoat III

[Dear Charles, you are, of course, right. The similarity in the reports was noticed by the editor, but we decided to print the piece as is. — The Editors]

TPJ

What I have found of xy-pic is that it is a wonderful program with utterly opaque documentation. So everything I can learn about it is helpful and I learned several new things from Paul Blaga's article <http://tug.org/pracjourn/2006-4/blaga/>. I will download and print out his articles and doubtless make much use of them.

I would like to call his attention to diagxy that provides an alternate syntax for most commutative diagrams. Just to see what the diagxy code would look, I programmed the diagram at the end of the article whose code got omitted (while the code for the penultimate diagram was repeated). I cannot tell whether it is familiarity or not, but I find my code considerably easier to generate and read. At any rate the code follows.

Michael Barr

```

$$\bfig
\node 1a(300,1700)[\Sigma^L]
\node 1b(1300,1700)[\Sigma^R]
\node 2a(0,1300)[L]
\node 2b(500,1300)[L_r]
\node 2c(1000,1300)[R]
\node 3a(0,700)[L_m]
\node 3b(500,700)[K_{r,m}]
\node 3c(1000,700)[K_{m^*}]
\node 4a(300,400)[\Sigma^G]
\node 4b(1300,400)[\Sigma^H]
\node 5a(0,0)[G]
\node 5b(500,0)[G_{r^*}]
\node 5c(1000,0)[H]
\arrow[1a1b;\phi^r]
\arrow|r|/ @{>} | (.31){\hole} | (.77){\hole} / [1a4a;\phi^m]
\arrow|r| [1b4b;\phi^{m^*}]
\arrow|l| [2a1a;\lambda^L]

```

```

\arrow|a|/ >->/[2b2a;i_1]
\arrow[2b2c;r]
\arrow|l|/ >->/[3a2a;i_2]
\arrow/ >->/[3b3a;i_3]
\arrow[3b3c;r]
\arrow|r|/ >->/[3b2b;i_4]
\arrow|r|[5a4a;\lambda_G]
\arrow|l|[3a5a;m]
\arrow|l|[3b5b;m]
\arrow|l|[3c5c;m^*]
\arrow|a|/@{>}|(.2){\hole}|(.7){\hole}/[4a4b;]
\arrow|r|[5c4b;\lambda_H]
\arrow|b|/ >->/[5b5a;i_5]
\arrow|b|[5b5c;r^*]
\arrow|l|[2c1b;\lambda_R]
\arrow|r|/ >->/[3c2c;i_6]
\efig$$

```

[We are glad this article has been useful to you. You might be delighted to see that this issue has a further article by Paul Blaga on XY-pic <http://tug.org/pracjourn/2007-1/blaga/>. Thank you for your alternative code. As you say, the documentation for XY-pic is rather opaque, and doubtlessly readers will benefit from your alternate solution. — The Editors]

TPJ

Dear PracTeX Journal,

I just read Walter Schmidt's short guide to fonts <http://tug.org/pracjourn/2006-4/schmidt/>. and this is just what I was searching. I use LaTeX for about four years and fonts were always a mystery for me.

Now with your guide and the psnfss manual I see much clearer.

Thanks a lot.

Best regards and keep on TeXing

Kai Wassermann

TPJ

Madsen's paper <http://tug.org/pracjourn/2006-4/madsen/> is a useful article. I've been trying to convince people where I work to use amslatex's align environment over eqnarray for several year, so it will be nice to be able to point them to your article. It will also be useful to point it out to the producers of some of the newer latex editors to try and convince them to include the align environment as the default for aligning equations.

Glenn

Glenn Fulford

g.fulford at qut.edu.au 3864 4295

<http://www.maths.qut.edu.au/~fulford>

TPJ

As I've only be reading PracTeX Journal for a day or two, I don't have much to say except it is amazing to see such a source of information available here. Many thanks.

Regarding the discussion of eqnarray by Lars Madsen <http://tug.org/pracjourn/2006-4/madsen/>, I notice an error on page 9, last line, where "but when using eqnarray" should probably state "but incorrect when using eqnarray".

This may sound trite, but I hope the author appreciates the correction. I certainly appreciated the effort he has put into his explanation of the evils of , and alternatives to, eqnarray.

Best regards,
Gernot Hassenpflug

[Dear Gernot, we appreciate user feedback: corrections such as yours from our readers mean we can improve upon the quality of our journal. Any improvement, however small, is worth aiming for! — The Editors]

TPJ

[Journal](#)
[home page](#)
[General](#)
[information](#)
[Submit an](#)
[item](#)
[Download](#)
[style files](#)
[Copyright](#)
[Contact us](#)

THE PracTeX Journal



News from Around:
Videos of the Grand Wizard
Kern Shirt
LaTeXpedia
LaTeX Tutorial Days
Recent Release: The 2007 TeX Live and *TeX Collection*
Recent Release: PCTeX 6

The Editors

- [Comment on this paper](#)
- [Send submission idea to editor](#)

Videos of the Grand Wizard

Here are some [videos](#) from the early 1980's, showing Don Knuth teaching and explaining TeX. Those of you who were TeX pioneers may remember seeing Don demonstrate TeX. His fingers flew over the keyboard and beautiful documents would appear. For some of us, though, our fingers turned to lead while trying to type TeX commands and it took some time before anything useful appeared. The TeX system that existed before 1982 is no longer used. The "TeX82" system referred to in these pages was an old name for the TeX program we use today. The LaTeX macro package first appeared in 1982.

Kern shirt

Do you have difficulty explaining to your family and friends what you do with TeX and LaTeX? Here's a [nifty aid](#) which doubles as a fashion statement.

(forwarded by Peter Flynn)

LaTeXpedia

An exciting project has been proposed by Lapo Mori: the [LaTeXpedia project](#). This project will be sponsored by the Italian TeX users (GuIT) and will be wiki-based. Over the years several good resources have been developed, and it is always possible to use a search engine to find information about a LaTeX command or package. However, there is no one unified, online LaTeX reference. The LaTeXpedia will hopefully fill this gap.

Poll: LaTeX tutorial days

Last fall the United Kingdom TeX Users (UKTUG) sponsored a successful [Day of LaTeX](#). Would you be interested in participating in a similar event if it were held near you? What topics would you like to see presented? Can you help organize? Contact the [editors](#) or post your ideas on the [PracTeX Journal forum](#) (or [join the forum](#)).

Recent release: The 2007 TeX Live and *TeX Collection*

TeX Live is an easy way to get up and running with TeX. It provides a comprehensive TeX system with binaries for most flavors of Unix, including GNU/Linux, and also Windows. It includes all the major TeX-related programs, macro packages, and fonts that are free software, including support for many languages around the world. For details on how to download the installation or to order CD/DVD's see [The TeX Collection](#), page.

Recent release: PCTeX 6

PCTeX 6 was released in February. This new release offers several new LaTeX productivity tools, such as a graphics inclusion wizard, wizards for configuring document and package options, easy access to documentation, and more. For more information see the [PCTeX web site](#).

[Journal](#)
[home page](#)
[General](#)
[information](#)
[Submit an](#)
[item](#)
[Download](#)
[style files](#)
[Copyright](#)
[Contact us](#)

THE PracTeX Journal



Graphics in LaTeX

Claudio Beccari

Abstract

This tutorial describes some facilities offered by LaTeX and its extension packages for producing line art graphics directly in the source document. Some of these facilities are standalone, in the sense that they do not require functionalities of external programs, while others rely on external programs.

Claudio Beccari is a long time LaTeX user and in 1991 wrote a book in Italian with the title "LaTeX — Guida a un sistema di editoria elettronica". Since then he is considered one of the gurus of the Italian TeX Users, even if this fame is totally undeserved. He has contributed several papers to TUGboat; he produced the hyphenation patterns for Italian and Latin, still in use today; he designed the default Greek fonts for use with Babel and supplied the Greek hyphenation patterns (luckily enough the Greek Users produced better patterns and these replaced Claudio's). Presently the Italian language definition file of the Babel package, although under the full control of Johannes Braams, is regularly updated and enriched by Claudio. He has participated in various TeX conferences and is a member of TUG GuIT, the official Group of the Italian TeX Users. He can be reached at claudio.beccari@polito.it

- [PDF version of paper](#)
- [Comment on this paper](#)
- [Send submission idea to editor](#)

Page generated March 14, 2007 ; [TUG home page](#); [search](#); [contact webmaster](#).

Graphics in L^AT_EX

Claudio Beccari

Email claudio.beccari@polito.it

Address Politecnico di Torino, Turin, Italy

Abstract This tutorial describes some facilities offered by L^AT_EX and its extension packages for producing line art graphics directly in the source document. Some of these facilities are standalone, in the sense that they do not require functionalities of external programs, some, on the opposite, rely on external programs.

1 Introduction

When a book is produced it is very unlikely that it does not contain some graphic material, be it pictures, diagrams, line art, and the like.

In this paper I will not discuss how to insert graphics that are already available in files produced by external equipment (for example photo cameras) or external programs, such as any of the commercial or freeware drawing programs available outside the T_EX-system distributions, or even METAPOST, that comes with the distribution.

The standard package `graphics` and its “extension” `graphicx` already available with the system, offer simple macros with which such inclusion may be performed with a number of optional actions available, such as rotation, scaling, trimming, clipping, et cetera.

Maybe these extension packages assume that the user is well aware that if L^AT_EX is being used the only graphic formats that can be included are those belonging to the PostScript family (including the METAPOST output), while if pdfL^AT_EX is being used, the only formats admissible are PDF, JPEG, PNG, and METAPOST output, provided in the latter that its file extension is changed to `mps`. It is true that some other formats may be included if a separate file containing the bounding box information is available, but as far as I know this possibility is seldom used because it requires some hacker’s ability.

At the same time, any or most T_EX-system distributions contain such little programs as `ps2pdf` in order to transform (encapsulated) PostScript file into PDF ones, or `jpegtops` to transform JPEG files to PostScript format. A variety of such programs is available so that a transformation from one format to another allows to use any image with both “plain” L^AT_EX or pdfL^AT_EX, or ConT_EXt, or...

Here I focus my attention on native and imported graphic functionality that is available with (pdf)L^AT_EX and ConT_EXt. With that I mean the standard graphics macros and the extended macros available with extension packages that conform with the L^AT_EX “language”; METAPOST will be left out from this tutorial, even if it belongs to the T_EX-system bundle, because its macros require the user to learn another language. The interested reader may look up the METAPOST documentation, [?]; be aware, though, that ConT_EXt MetaFun package, [?], allows to exploit modern T_EX-system implementations in order to simultaneously run the T_EX and METAPOST engines and to produce line art graphics directly from one source file.

2 The standard L^AT_EX `picture` environment

L^AT_EX was born in 1984 with a native `picture` environment that allowed one to draw lines, vectors, circles and ‘ovals’ with some limitations; special fonts were used to draw all graphic objects. This was a severe limitation, because at L^AT_EX birth all fonts usable with L^AT_EX could contain only 128 glyphs; this was true with text fonts as well as with L^AT_EX special drawing fonts.

This implied that there were only two thicknesses available for all graphic objects, but, worst of all, the line and vector slopes and the circle diameters were available in only a limited number. Line slope parameters could only be integer relatively prime numbers in the range $-6 \text{ --- } +6$, while for vectors they could range only within $-4 \text{ --- } +4$. This means that the only lines and vectors that could be drawn are represented (with just positive slopes) in figure ?? . Filled circles (disks) were limited to diameters from 1pt to 15pt, while unfilled circles were limited to diameters from 1pt to 40pt; circles of diameter larger than 15pt were drawn as four quarter circles, each one of them being available also for the rounded corners of ‘ovals’.

This environment allowed to set text by means of extensions of the `\makebox` macros, and framed text by means of `\framebox` ; fine tuning of the position of the

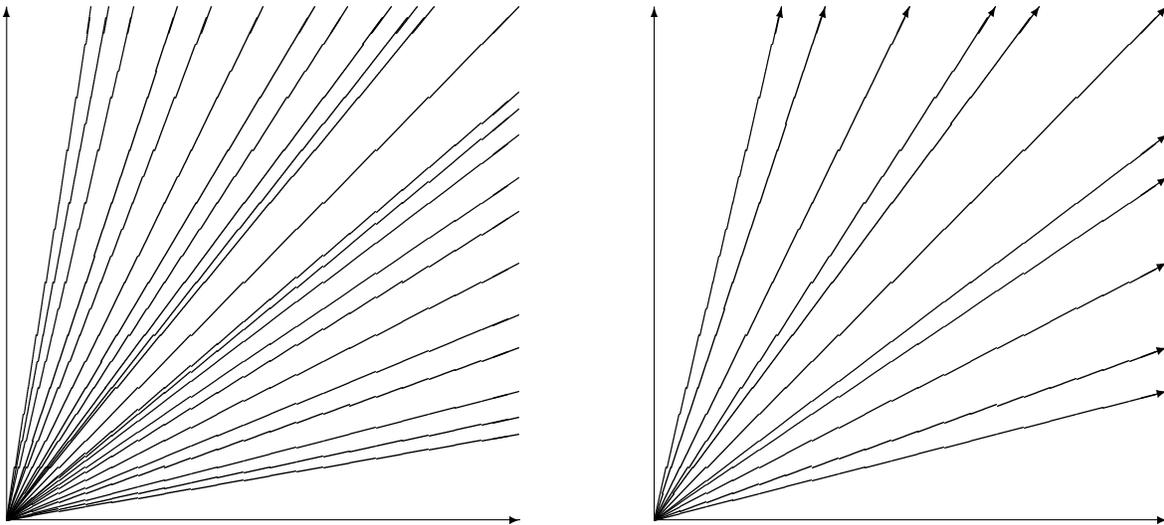


Figure 1: Possible positive slopes for lines and vectors in the standard \LaTeX environment

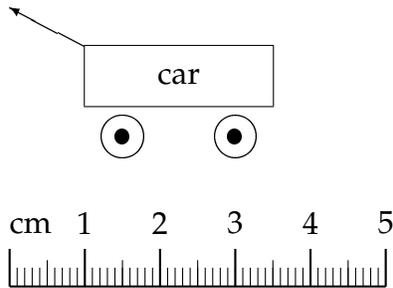


Figure 2: A \LaTeX `picture` environment simple drawing

text allowed to put any text in the precise required position and, most important of all, the fonts used were the same ones used within the whole document. Under the typographic point of view this feature was and remains the most valuable one of this built-in environment. Figure ?? displays the same drawing Lamport used in his handbook to describe his `picture` environment.

3 PiCT_EX

The first graphic extension available on the market was PiCT_EX[?]. It was designed to work with plain T_EX, in an initial stage of T_EX's "history"; after L^AT_EX became available, some macros were set up in order to let L^AT_EX import those PiCT_EX macros and use them at almost full power. Those were the times when T_EX and L^AT_EX were run on mainframes and PCs were in their infancy; The T_EX-system was already available for those small PCs, but the memory limitations were so strong that on such machines it was quite normal to get the dreadful message that there had been a memory saturation and processing aborted.

Now-a-days the PC memory limitations have substantially vanished, and a medium price PC has enough read-only memory, virtual memory and disk space to do more than the mainframes available in the eighties. As a matter of fact a new version of PiCT_EX, named m-PiCT_EX, [?], is part of the ConT_EXt system as a normal accessory.

All the functionalities of PiCT_EX, or most of them, were transferred to more modern extensions, but I often dig into its code in order to learn the exceptionally witty tricks implemented by Michael Wichura in order to have the T_EX-engine perform fractional decimal number computations, from logarithms to trigonometric functions, and the like.

The problem with PiCT_EX (and also with more recent macro sets) is that arbitrary lines are drawn by setting a myriad of dots partially superimposed one to another so as to simulate continuous line drawing; this multitude of dots implies such a heavy memory usage that remains locked until the picture is shipped out into the dvi file; a single picture may saturate the available memory during its processing; let's not speak of what happens when several figures containing PiCT_EX graphics are queued by L^AT_EX before shipping them out.

In spite of these drawbacks, PiCT_EX is very good at drawing line art of arbitrary complexity, included that form of graphical mathematical objects known as commutative diagrams. It is true that 2-dimensional commutative diagrams may be typeset with the `cmd.sty` package made available by the American Mathematical Society; but PiCT_EX can draw 3-dimensional ones; a very attractive commutative diagram decorates the first page of the PiCT_EX manual, which is not free, so I can't reproduce it; a partial imitation is shown in figure ??, where I used the standard macros available within the L^AT_EX `picture` environment.

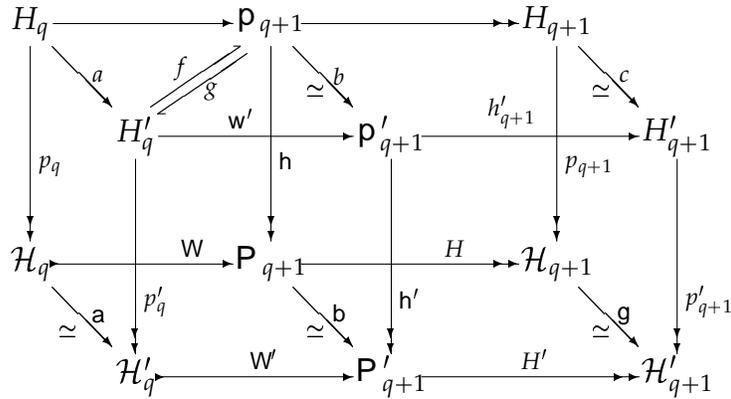


Figure 3: A three dimensional commutative diagram

4 The epic and eepic picture extensions

In 1986 Sunil Podar published his extension package `epic` [?], to the standard `LATEX picture` environment; in 1988 Conrad Kwok published the further enhancement `eepic` [?].

Both were conceived so as to relieve the strong limitations of the standard `LATEX picture` environment, in particular the limitation on the line and vector slopes and the limited range of circle radii.

Both programs were conceived for use under the ‘old’ `LATEX`, today known as `LATEX 209`, in contrast with the new one `LATEX 2#` (which is not ‘new’ anymore, since it is about twelve years old). In particular `eepic` was conceived in order to translate picture drawn with the `xg` program into a set of macros that `epic` could use for drawing complex drawings. `xg` is a drawing program original of the UNIX platforms that can output its contents into a variety of formats, including `LATEX picture` commands. Nevertheless such commands bear the limitations described above, so that the `fig` format is preferred, but such a format must be translated into commands `LATEX` can actually execute by itself or with the intermediary of a macro package. This is exactly what `eepic` is for.

It must be said that most of the limitations of the old `LATEX picture` environment are overcome by the set of macros available with the `pict2e` extension package, which I will speak about section ???. Therefore these extensions `epic` and `eepic`, although not obsolete, are sort of outdated.

5 The curves package

The ctan archives contain the package prepared by Ian Maclaine-Cross initially in 1991, then revised for L^AT_EX2#. The latest release in the year 2000 is the current one, [?].

The package `curves` accepts a number of options in order to simplify the output file by letting several graphic objects to be drawn by the output driver. Without these specifications, the curves drawn by the package are made up by means of the superposition of a multitude of black disks; if possible it is better to use the specific driver option, but the `dvips` is available while the `pdfTEX` one is not. If a PDF output format is desired, it is necessary to obtain it through the lengthy path `LATEX → dvips → pstopdf`.

In any case the main package `curves` contains many new features compared with the standard situation; in fact all the graphic objects can be drawn with ‘arbitrary’ thick lines, lines may have arbitrary slopes, curves may be drawn by means of second degree Bèzier splines (parabolas) where it is necessary to specify only the interpolation nodes, since control points are determined by the macros themselves. If the user is inclined to specify few nodes, the macros try to do their best, but more often than not curves end up having spurious loops, especially if inflection points are implied. But if the user does not spare the interpolation nodes, the curves turn out to be very nice and smooth, even if inflection points are involved. Let’s remember that second order Bèzier splines are those used to describe the contours of the TrueType fonts, while the third order splines are used to describe fonts in the METAFONT output format and in PostScript Type 1 format.

This package allows one to draw objects, save them and redraw them with ‘arbitrary’ transformations (rotation, x -scaling and y -scaling) so that fine assembling of such objects is possible. Figure ?? represents a prospective view of a square washer drawn with `curves`.

6 PStricks

The previous package `curves`, although capable of working as a standalone L^AT_EX macro package, is the first example of a drawing package where some of the functionality is provided by an external program, in this particular case to `dvips`

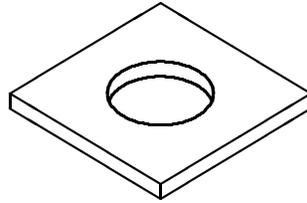


Figure 4: A square washer drawn with curves

and a few other similar PostScript based drivers.

The set of packages that are collectively known as `PSTricks` do the same, in the sense that they exploit to a full extent the PostScript language by writing to the output file, a `dvi` one in our case, all the `\special` commands that introduce raw PostScript code necessary to draw all of the required objects.

The PostScript language is a very powerful programming language specifically designed to describe a typeset page and all the objects it contains; of course images and line art are at home in a typeset page, so that graphics handling is an integral part of the PostScript language.

Basically this language can handle fonts and their glyphs and can draw lines and curves and fill the paths they enclose with colors and/or patterns; but the most important feature is that PostScript can define macros more or less in the same way as the \TeX -engine does. These macros contain a full set of control statements, so that PostScript is capable of doing actions that one might reserve only to higher programming languages. The difficult part of the PostScript language is that it uses the Polish notation, to which we humans are not used so much; the code appears cryptic to anybody who is not a PostScript programmer, at least it is for me; but evidently Timothy van Zandt, who wrote the interface, had the same idea and produced a large set of \TeX macros that allow the user to write with the usual \LaTeX style, while these macros translate the \LaTeX information into PostScript code, doing, besides other things, the necessary calculation for transferring all the internal quantities in typesetter's points (72.27pt to 1in) into PostScript points (72bp to 1in); the difference might appear small and may be negligible, but after many calculations the precise positioning of graphic objects is decreased because this rounding is not so negligible. In any case the user does not have to learn the PostScript language.

It would be too lengthy to describe the available commands; suffice to say that

the bundle of the PSTricks packages covers almost any possible graphics situation; from structured graphs, to electric circuits, from optics setups to mechanical springs, from box diagrams to 2- and 3-dimensional graphics.

The point is that L^AT_EX can process files where one or more PSTricks packages have been used to draw something; the raw PostScript code written in the output dvi file by means of the `\special` commands may be processed only by further programs that understand and execute PostScript code; typically this program is `dvips` and the result of this process is a PostScript file. This is good for many applications, but more often than not a PDF file is required, so that the PS file must be ‘distilled’ by any of the various programs that convert a PS file into a PDF one. The normal process, therefore, is L^AT_EX \longrightarrow `dvips` \longrightarrow `pstopdf`; many T_EX-system distributed editors already contain a ‘button’ to click that execute the complete triple process, so that the user should not worry too much.

But all this implies also that the user cannot employ pdfL^AT_EX as the engine to process an input L^AT_EX file, because the pdfL^AT_EX engine is not capable of understanding and executing the raw PostScript code, in spite of the fact that the uncompressed PDF page description language is a subset of PostScript.

This incompatibility reflects itself on other side points, such for example the inclusion of external graphic material; since L^AT_EX is the only program that can process PSTricks macros and their `\special` commands, this implies that all external graphic material must be in encapsulated PostScript format, besides the METAPOST output; if the user wants to include pictures coming from a digital photo camera, s/he must first transform the picture format, probably JPEG, into an encapsulated PostScript.

7 X_Y-pic

Another very powerful graphics package that relies on using an external software for actually rendering the graphic objects is X_Y-pic, [?].

This package is designed to work with L^AT_EX, plain T_EX, and AMS-T_EX. It can draw diagrams of any ‘mathematical’ kind, from commutative diagrams to those used in category theory, automata theory, algebra, neural networks, and database theory.

The feature that makes it so versatile is that mathematicians were the first ones to use plain T_EX, and, apparently keep preferring this incarnation of the T_EX-

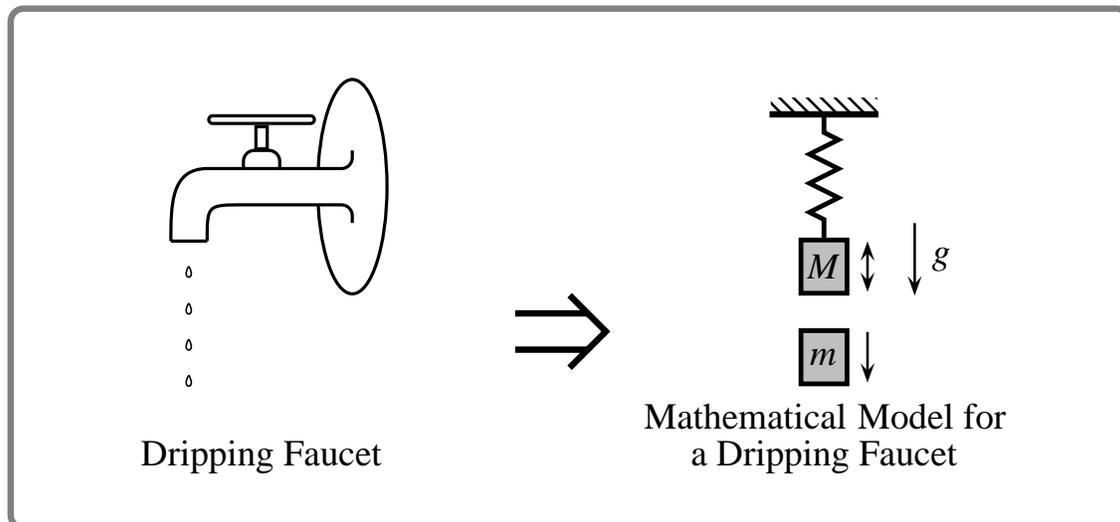


Figure 5: A PostScript line art figure obtained from the use of PSTricks; it appears into the cover page of the PSTricks manual

system rather than using more user friendly packages; plain $\text{T}_{\text{E}}\text{X}$ and $\text{AMST}_{\text{E}}\text{X}$ are sort of similar and refrain from using those bells and whistles that make $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ so widespread. Nevertheless the developers adapted it also to $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$, but the users must cope with what the authors consider a ‘logical composition of visual components’. The idea is brilliant, but in practice I find the syntax of the various object descriptions a little too cryptic, or may be I am too old to learn new sophisticated graphic description languages.

I can read the manual, see the examples and agree that the power of $\text{X}_{\text{Y}}\text{-pic}$ is highly underestimated; I am guilty on my own for not daring to sit down and try hard to get the best out of it. In any case I show a simple figure taken from the manual, figure ??; the commutative diagram is simple, but it contains objects such as the curved and dotted arrows that are not to be easily found in other packages.

What is amazing with $\text{X}_{\text{Y}}\text{-pic}$ is the fact that its syntax is extremely compact; the whole commutative diagram of figure ?? is described in just six lines of code in the two-column formatted $\text{X}_{\text{Y}}\text{-pic}$ Guide.

One little point of warning: $\text{X}_{\text{Y}}\text{-pic}$ uses several characters as special purpose ones; among these characters there is the double quote " , which acts also as an ac-

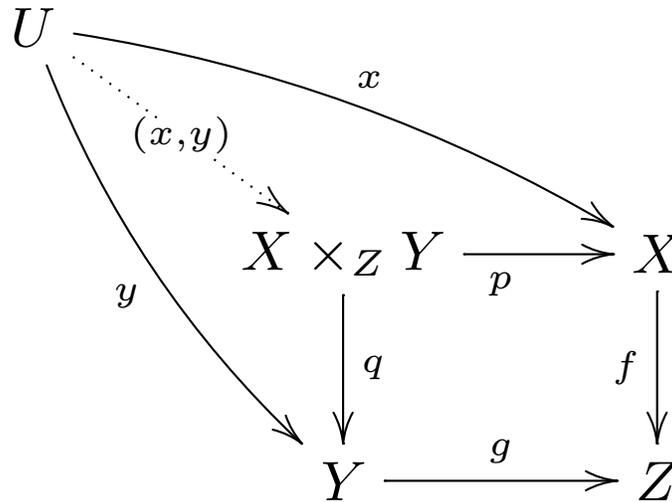


Figure 6: A simple commutative diagram drawn with \mathcal{X}_Y -pic

tive character in most `babel` language description files, perhaps all languages but English. In order to use \mathcal{X}_Y -pic, it is necessary to turn off the shortcuts associated with the double quote by means of the `babel` command `\shorthandoff{"}`; this can be done just upon entering the `xypic` environment, and without this action \mathcal{X}_Y -pic is virtually unusable when the current language option is different from english !

8 The `pict2e` and `curve2e` extensions to the standard \LaTeX `picture` environment

Leslie Lamport, in his second edition of the \LaTeX handbook [?], fixed the syntax of a new extended `picture` environment, where most if not all limitations of the standard implementation could be overcome: unlimited slopes of lines and vectors, any circle radius, arbitrary line thickness also for curved lines, real third order Bèzier curves, et cetera.

Eventually in 2003 H.Gäßlein and R.Niepraschk published an implementation of Lamport's extensions with their package `pict2e` [?], which implements these extensions by means of the driver capabilities.

Figure ?? shows some of the new possibilities: vectors of any slope and thick-

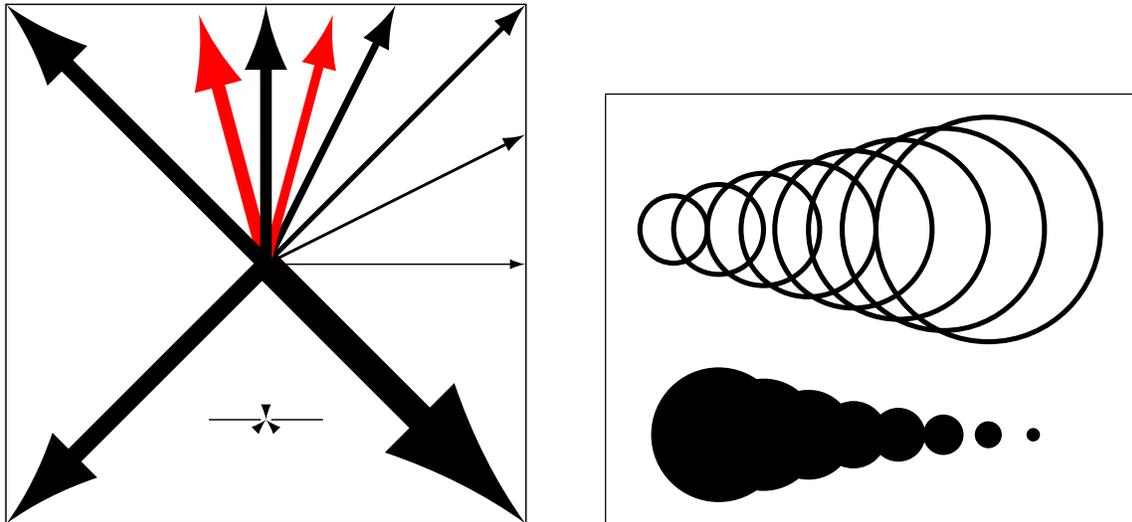


Figure 7: Some examples of `pict2e` capabilities

ness and circles and disks of any diameter. Of course that is a simple example, but it gives a good idea of the progress obtained with the `pict2e` extension.

I did some personal extensions to the already available extensions offered by `pict2e` [?]. My goal was to prove the point that `pict2e` could be freed from the remaining constraints of integer slope parameters imposed by Lamport to the new package syntax; it is true that the new slope parameters are constrained to be any signed three digit integer, but this does not improve the performance of tracing lines and vectors.

As any $\text{T}_\text{E}\text{X}$ -system user knows, the $\text{T}_\text{E}\text{X}$ -engine is not capable of making computations with decimal fractional numbers; the only ones it deals with are the scale factors for dimensions. It is not difficult to create macros for simulating addition, subtraction and multiplication, but division is another thing; let's not speak of square roots or, even worse, trigonometric and other transcendental functions. Many of the described graphic packages went around these limitations with very clever algorithms, but apparently Lamport did not want to extend his new `picture` environment so far.

But with a suitable division routine even the line and vector tracing algorithms can exploit any fractional slope parameter; and this opens the door for 'turtle graphics'. My package `curve2e` does exactly that, allowing also for circular

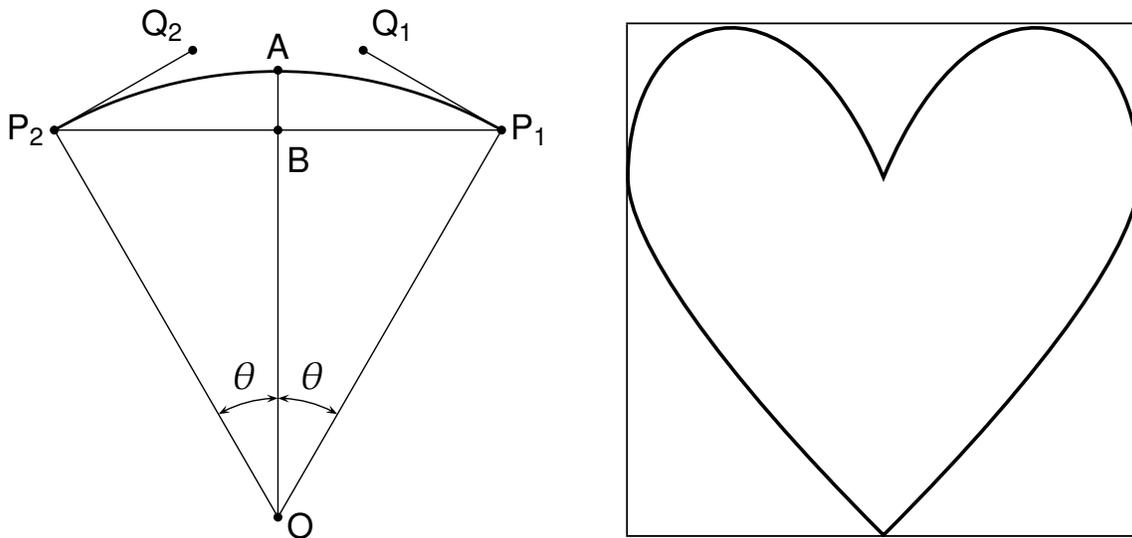


Figure 8: Some examples with package `curve2e`

arcs, circular vectors with the arrow tip at one or at both ends, to generic curves obtained by specifying just the interpolation nodes and the tangent directions in such nodes. The simple examples in figure ?? show some functionalities of this package.

9 The `pgf` and `xcolor` packages

The acronym PGF stands for ‘Portable Graphics Format’; the package `pgf` [?], implements or re-implements all commands of the standard `LATEX picture` environment and those of the `graphics` package so as to create a coherent set of commands capable of doing most of the graphic operations in `LATEX` and in `pdfLATEX`. Many commands are added to that fundamental set so as to extend the `LATEX` graphic capabilities almost to the level of `PSTricks`; its capabilities are further extended by means of the `xcolor` package that on its own extends the possibilities of color handling offered by the standard `color` package.

This package automatically examines the default configuration files and inserts in the output files the `\special` commands suited for the output driver; if `LATEX` is being used (and therefore `dvips` or some of its kin is used) the suit-

able PostScript code is inserted in the output file. If pdfL^AT_EX is being used, the `\specials` contains PDF code that is therefore immediately active within the output PDF file.

Since PDF code at the moment is not as powerful as the PostScript code, when pdfL^AT_EX acts as the typesetting engine it is not possible to exploit the full power of the PSTricks package, but Till Tantau is working hard to extend this excellent graphic package so that the full functionality may be achieved.

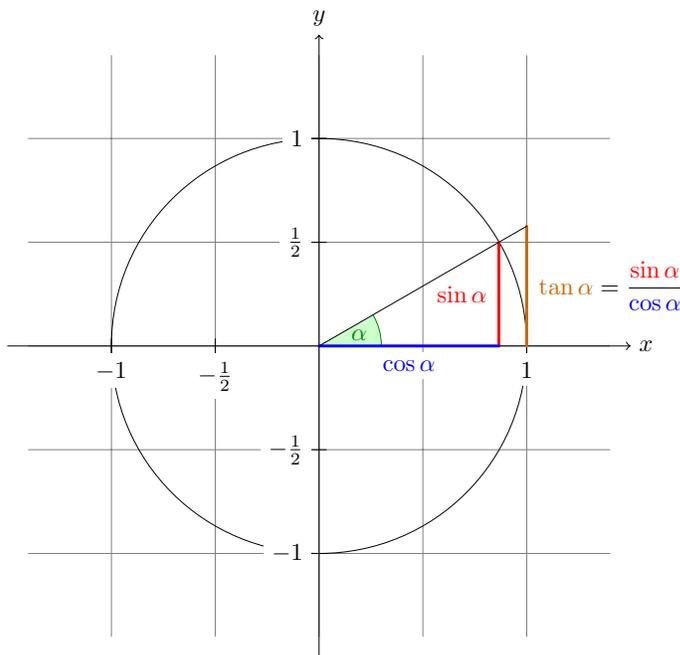
A personal experience: in September 2006 I bought myself a new laptop; I installed all the packages I am used to, among which, of course, the whole T_EX distribution, the latest version available at that time. I had to prepare my presentation to the Marrakech TUG2006 Conference, so I furbished what I had already prepared using my previous laptop. I did not realize that the PGF package had changed so much; I used the facilities I was used to, but I was not aware that a major revision had added a lot of new features to that package. If I had known, I would have withdrawn my contribution to that conference. Fortunately enough in my presentation I was praising the PGF bundle, but I was not describing it, so that my presentation was still “presentable”...

What I discovered while preparing this paper was that a completely new interface had been created so that PGF graphics could be drawn in a much simpler way.

The new PGF bundle, version 1.10, contained a new package and its new graphic environment that is called TikZ; this acronym stands for “TikZ ist kein Zeichenprogramm” (TikZ is not a drawing program), but this is an understatement set forth by Till Tantau in order to avoid bragging.

Yes, dear readers, PGF and its new package tikz is a very nice drawing interface that allows to draw almost anything; it has not the full power of PSTricks, but it goes a very good length in that direction; the manual explains in good detail what can’t be done with the PostScript interface, but, what’s best, it explains that the PGF format and program were specifically imagined to be used with pdfL^AT_EX, where it performs best.

I had little time to get familiar with and to master TikZ, but after my first experiments and real drawings I can definitely state that I will stick to this software for a long time from now. Of course the first thing I have to upgrade is to transform the circuit drawing environment I spoke about in my earlier paper [?] in order to exploit the best of TikZ. But for other drawings I’ll use TikZ directly.



The angle α is 30° in the example ($\pi/6$ in radians). The sine of α , which is the height of the red line, is

$$\sin \alpha = 1/2.$$

By the Theorem of Pythagoras we have $\cos^2 \alpha + \sin^2 \alpha = 1$. Thus the length of the blue line, which is the cosine of α , must be

$$\cos \alpha = \sqrt{1 - 1/4} = \frac{1}{2}\sqrt{3}.$$

This shows that $\tan \alpha$, which is the height of the orange line, is

$$\tan \alpha = \frac{\sin \alpha}{\cos \alpha} = 1/\sqrt{3}.$$

Figure 9: An example taken from Tantau's manual [?]

I have already been using the PGF bundle for handling pictures when using the presentation software `beamer` (another achievement of Till Tantau's), but now almost everything can be done with it.

Tantau says that `TikZ` is intended to offer the user a simplified and uniform interface so as to program its drawings with the least possible fuss; in his manual he wrote an introductory small teaching demo so as to produce figure ???. I will not copy the code here, because anybody can find it on the manual [?], that in any case has to be used intensively (it's a about 370 pages long and it is very well done and well organized) because of the multitude of drawing commands available for any possible situation.

As you can see in figure ??? the program allows for black and white, and color graphics, and the text uses the same fonts as the default ones in the PDF document; this is one of the features that is missing from most external drawing packages and forces the user to circumvent this vacancy with a number of tricks. `TikZ` frees the user from using any trick for this purpose because it is fully integrated with `(pdf)LaTeX`.

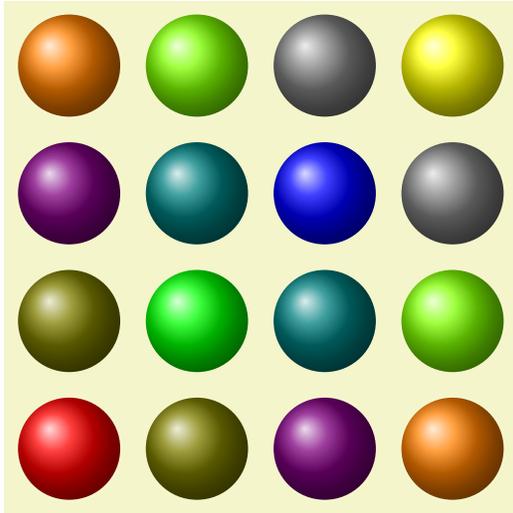


Figure 10: Color gradients and mixtures from Tantau’s manual [?]

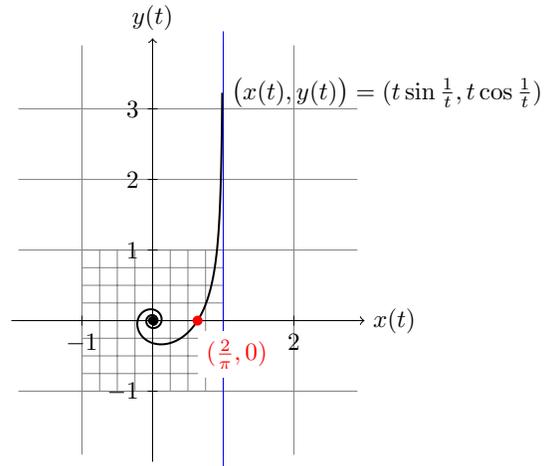


Figure 11: A spiral plotted by TikZ the coordinates of which are computed by GnuPlot; from Tantau’s manual [?]

Color is dealt with advanced commands; in particular it is possible to do gradients and to mix colors; figure ?? displays an array of balls with circular gradients and mixes the four colors that appear in the matrix secondary diagonal.

The PGF bundle contains a lot of libraries of additional commands; the standard drawing objects are available also without resorting to these libraries, but the latter add a lot of options and functionalities. It is possible to color the inside of closed paths, to make node drawings, automata diagrams, there is an enormous variety of “arrow” tips, the choice of backgrounds, including the page background over which the regular pages are typeset, entity-relationship diagrams, mindmap diagrams (very attractive), a choice of background patterns for technical drawings, Petri-net drawings, and so on.

The bundle handles the drawing of plots; the latter may be defined as a series of two or three coordinate pairs or triplets so that 3D drawings may be executed, but it can interact with GnuPlot, a freeware external program that can compute and store the pairs or triplets of coordinates for subsequent use by pdfL^AT_EX¹;

1. This feature implies the activation of the `\write18` (pdf)T_EX facility; the latter is disabled by default, but it may be enabled for particular tasks, so that its use is not so dangerous as it might be; the default enabling, in facts, would allow the user of the ever present malware to launch

actually it is not necessary to launch GnuPlot from within pdfL^AT_EX; it can be run in deferred mode by the user himself and in a second run of pdfL^AT_EX retrieves the files produced by GnuPlot and the plotting facility of TikZ executes the plotting; see figure ?? for another example taken from Tantau’s manual [?]. This plotting facility is completed by the possibility of marking the plotting nodes (the experimental points) with different symbols so that plots presenting different lines can be better distinguished even if colors are not used.

The “snake” library allows to change shape to most drawing objects, specifically straight or curved lines; a couple of special shapes I appreciate are the curved arrows, that allow to connect different nodes with non intersecting and not blurring arrows or lines, and the helical shape that is used so frequently in technical drawings.

The “shape” library contains the definitions of a lot of typical shapes, that can be colored inside and that contain text or symbols used in a lot of schematic block diagrams; a useful one is the “forbidden sign”, that can be used in a variety of situations; the example on Tantau’s manual contains the text “Smoking” so that the strong meaning “It’s forbidden to smoke” emerges with due emphasis.

The “to path” library allows one to draw curved lines with third order Bèzier splines without the need of specifying the node tangent directions. Such Bèzier third order splines may be used freely even without the use of this library provided the node tangents are specified; but this library renders the curve drawing task much easier.

The “tree” library allows to draw trees of nodes with parents and children so that most of the logical visual drawings can be performed with no difficulty.

The TikZ syntax is rather simple; its statements start with a command, continue with options and coordinates and finish with a semicolon; they remind of the METAFONT or METAPOST commands and syntax, but TikZ is not a replacement of METAPOST or METAFONT. The options and the object names remind of some PSTricks commands, but TikZ is not a replacement for PSTricks.

A true novelty is the possibility of making simple inline drawings, such as this:  without opening an environment. The intuitive code for drawing the above shaded button is the following:

[...] such as this: `\tikz \shade[ball color=red] (2ex,1ex) circle(1ex);`

destructive programs, without the user being aware of what’s happening.

without opening [...]

A final comment: *TikZ* may be used also with plain $\text{T}_{\text{E}}\text{X}$, $\text{pdfT}_{\text{E}}\text{X}$, and $\text{ConT}_{\text{E}}\text{Xt}$; it is thought as a modern package usable in different situations, and typesetting engines; it recognizes by itself what engine is being used and changes its performances, syntax and interface accordingly; it eventually issues the necessary `\special` commands, which are transparent to the user and match the specific driver that is necessary to use for rendering the document in human readable form; possibly to the user remains the task of running the necessary filters, such as `dvips`, but the $\text{pdfL}_{\text{A}}\text{T}_{\text{E}}\text{X}$ users do not have to do anything in order to display their work, because $\text{pdfL}_{\text{A}}\text{T}_{\text{E}}\text{X}$ outputs directly the readable document.

10 Conclusion

This tutorial scans a variety of packages for drawing simple or complicated drawings with a variety of programming interfaces and different levels of sophistication in the complexity of the final output.

Certainly the basic standard $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ `picture` environments pales in front of the performances of `PSTricks` or those of the `PGF` bundle, even if the standard environment is upgraded with `pict2e` and `curve2e`, but its simple drawings may be drawn by anybody with a learning time reduced to nothing. `Xy-pic`, `PSTricks` and *TikZ* definitely have more performances but require a more or less steep learning curve.

With *TikZ* I am personally experiencing the satisfaction of drawing technical line art that I never succeeded to make with other $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ drawing interfaces; the results are worth the little strain of learning a new interface language.

Even if this tutorial does not show virtually any programming code, the reader is addressed to the required handbooks and manuals, all fetchable in the international `ctan` archives and generally uploaded with the various bundles, most of which are already included in most modern distributions of the $\text{T}_{\text{E}}\text{X}$ system, be it for a Windows, a UNIX/Linux or a Macintosh platform.

If you explore the `ctan` archives you can certainly find other graphic drawing interfaces; either they are very specific or they are overridden by the packages I described in this tutorial. Nevertheless if you need to draw something very special (as I have to, for example, with my electronics circuits) you'd better explore

ctan , where most likely the `graphics/` folder contains the specific package for your needs.²

I set forth some little warnings deriving from my experience; to say the truth with some of these packages I have not such a great experience yet, nevertheless I collected some information from colleagues that use them; the warning about disabling the double quote active character when using `Xy-pic` derives from a small article written on the web site of the Italian TUG, written by a user who was very disappointed when the Italian language description file was upgraded to include some “double quote” driven shorthands. When I introduced them into the language description file, I had the impression that all other language description files, but the English one, contained such shorthands, so why not in Italian, where the keyboard deficiency is remarkable³; there are other circumstances where some shorthands are necessary and the double quote stands out as the only character that Knuth did not use for something else (well, the hexadecimal numbers, but one can use the octal or the decimal ones instead). I am surprised that no such complaints were raised from users writing in other languages.

I encourage (pdf)L^AT_EX users to experiment extensively with the PGF package; after a while they will become addicted. . .

References

- [1] John Hobby, *METAPOST*, [CTAN:graphics/metapost/](#)
- [2] Hans Hagen, *MetaFun*, [CTAN:manuals/metafun-p.pdf](#)
- [3] Michael Wichura, *PiCT_EX*, [CTAN:graphics/pictex/](#)
- [4] Michael Wichura *m-PiCT_EX*, [CTAN:help/Catalogue/entries/m-pictex.html](#)
- [5] Sunil Podar, *The epic package*, [CTAN:macros/latex/contrib/epic/](#)

2. Why, then, am I going to rewrite my electronics circuit drawing package, if there is one already there in `ctan` ? Because even if that package contains a larger variety of components, they are drawn according the North American standards which are not standard in Europe.

3. The Italian standard keyboard does not contain the ever present T_EX characters such as both curly braces (!), the tilde and the grave accent; the latter is the only accent to be used on any Italian vowel, but the ‘e’ that can take the acute one instead.

- [6] Conrad Kwok, *The eepic package*, [CTAN:macros/latex/contrib/eepic/](#)
- [7] Ian Maclaine-Cross, *The curves and curveslm packages*, [CTAN:macros/latex/contrib/curves/](#)
- [8] Timothy van Zandt, *The PSTricks bundle*, [CTAN:graphics/pstricks/](#) ; the documentation is in [CTAN:applications/PSTricks/](#)
- [9] Kristofer Høgsbro Rose and Ross Moore, *The Xy-pic bundle*, [CTAN:applications/Xy-pic/Xy-pic.html](#)
- [10] Leslie Lamport, *A document preparation system — L^AT_EX — User's guide and reference manual*, Addison Wesley Publ. Co., Reading, mass., 2nd ed., 1994.
- [11] Hubert Gäßlein and Rolf Niepraschk, *The pict2e package*, [CTAN:macros/latex/contrib/pict2e/](#)
- [12] Claudio Beccari, *The curve2e package*, [CTAN:macros/latex/contrib/curve2e/](#)
- [13] Till Tantau, *TikZ and PGF*, [CTAN:graphics/pgf/](#)

[Journal](#)
[home page](#)
[General](#)
[information](#)
[Submit an](#)
[item](#)
[Download](#)
[style files](#)
[Copyright](#)
[Contact us](#)

THE PracTeX Journal



Graphics with PGF and TikZ

Andrew Mertz and William Slough

Abstract

Beautiful and expressive documents often require beautiful and expressive graphics. PGF and its front-end TKZ walk a thin line between power, portability and usability, giving a TeX-like approach to graphics. While PGF and TKZ are extensively documented, first-time users may prefer learning about these packages using a collection of graduated examples. The examples presented here cover a wide spectrum of use and provide a starting point for exploration.

Andrew Mertz and William Slough are faculty members in the Mathematics and Computer Science Department at Eastern Illinois University. They are enthusiastic fans of LaTeX, having a combined experience of 30 years with the typesetting system. They can be contacted at aemertz@eiu.edu and waslough@eiu.edu, respectively.

- [PDF version of paper](#)
- [Comment on this paper](#)
- [Send submission idea to editor](#)

Page generated March 14, 2007 ; [TUG home page](#); [search](#); [contact webmaster](#).

Graphics with TikZ

Andrew Mertz and William Slough

Email aemertz@eiu.edu, waslough@eiu.edu
Address Department of Mathematics and Computer Science
Eastern Illinois University
600 Lincoln Avenue
Charleston, IL 61920-3099

Abstract Beautiful and expressive documents often require beautiful and expressive graphics. PGF and its front-end TikZ walk a thin line between power, portability and usability, giving a $\text{T}_{\text{E}}\text{X}$ -like approach to graphics. While PGF and TikZ are extensively documented, first-time users may prefer learning about these packages using a collection of graduated examples. The examples presented here cover a wide spectrum of use and provide a starting point for exploration.

1 Introduction

Users of $\text{T}_{\text{E}}\text{X}$ and $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ intending to create and use graphics within their documents have a multitude of choices. For example, the UK $\text{T}_{\text{E}}\text{X}$ FAQ [1] lists a half dozen systems in its response to “Drawing with $\text{T}_{\text{E}}\text{X}$.” One of these systems is PGF and its associated front-end, TikZ [4].

All of these systems have similar goals: namely, to provide a language-based approach which allows for the creation of graphics which blend well with $\text{T}_{\text{E}}\text{X}$ and $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ documents. This approach stands in contrast to the use of an external drawing program, whose output is subsequently included in the document using the technique of graphics inclusion.

PGF provides a collection of low-level graphics primitives whereas TikZ is a high-level user interface. Our intent is to provide an overview of the capabilities of TikZ and to convey a sense of both its power and relative simplicity. The examples used here have been developed with Version 1.0 of TikZ .

2 The name of the game

Users of $\text{T}_\text{E}\text{X}$ are accustomed to acronyms; both PGF and TikZ follow in this tradition. PGF refers to **P**ortable **G**raphics **F**ormat. In a tip of the hat to the recursive acronym GNU (i.e., GNU's not UNIX), TikZ stands for “**T**ik**Z** ist kein **Z**eichenprogramm,” a reminder that TikZ is not an interactive drawing program.

3 Getting started

TikZ supports both plain $\text{T}_\text{E}\text{X}$ and $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ input formats and is capable of producing PDF, PostScript, and SVG outputs. However, we limit our discussion to one choice: $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ input, with PDF output, processed by pdf $\text{L}_\text{A}\text{T}_\text{E}\text{X}$.

TikZ provides a one-step approach to adding graphics to a $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ document. TikZ commands which describe the desired graphics are simply intermingled with the text. Processing the input source yields the PDF output.

Figure 1 illustrates the layout required for a document which contains TikZ-generated graphics. Of central interest is the `tikzpicture` environment, which is used to specify one graphic. Within the preamble, the `tikz` package must be specified, along with optional PGF-based libraries. Exactly which additional libraries are needed will depend on the type of graphics being produced. The two PGF libraries shown here allow for a variety of arrowheads and “snakes,” a class of wavy lines.

Commands which describe the desired graphic appear within a `tikzpicture` environment. In the simplest case, these commands describe paths consisting of straight line segments joining points in the plane. For more complex graphics, other primitive graphics objects can appear; e.g., rectangles, circles, arcs, text, grids, and so forth.

Figures 2 and 3 illustrate how a diamond can be obtained, using the `draw` command to cause a “pen” to form a closed path joining the four points $(1,0)$, $(0,1)$, $(-1,0)$, and $(0,-1)$, specified with familiar Cartesian coordinates. The syntax used to specify this path is very similar to that used by MetaPost [2]. Unlike MetaPost, TikZ uses one centimeter as the default unit of measure, so the four points used in this example lie on the x and y axes, one centimeter from the origin.

```

\documentclass[11pt]{article}
...
\usepackage{tikz}
% Optional PGF libraries
\usepackage{pgflibraryarrows}
\usepackage{pgflibrarysnakes}
...
\begin{document}
...
  \begin{tikzpicture}
    ...
  \end{tikzpicture}
...
\end{document}

```

Figure 1: Layout of a TikZ-based document.

```

\begin{tikzpicture}
  \draw (1,0) -- (0,1) -- (-1,0) -- (0,-1) -- cycle;
\end{tikzpicture}

```

Figure 2: Drawing a diamond with a closed path.

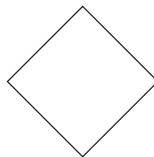


Figure 3: Results produced by Figure 2.

```

\begin{tikzpicture}
  \draw[step=0.25cm,color=gray] (-1,-1) grid (1,1);
  \draw (1,0) -- (0,1) -- (-1,0) -- (0,-1) -- cycle;
\end{tikzpicture}

```

Figure 4: Adding a grid.

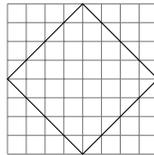


Figure 5: Results produced by Figure 4.

In the process of developing and “debugging” graphics, it can be helpful to include a background grid. Figure 4 expands on the example of Figure 2 by adding a draw command to cause a grid to appear:

```

\draw[step=0.25cm,color=gray] (-1,-1) grid (1,1);

```

Here, the grid is specified by providing two diagonally opposing points: $(-1, -1)$ and $(1, 1)$. The two options supplied give a step size for the grid lines and a specification for the color of the grid lines, using the `xcolor` package [3].

4 Specifying points and paths in TikZ

Two key ideas used in TikZ are *points* and *paths*. Both of these ideas were used in the diamond examples. Much more is possible, however. For example, points can be specified in any of the following ways:

- Cartesian coordinates
- Polar coordinates
- Named points
- Relative points

As previously noted, the Cartesian coordinate (a, b) refers to the point a centimeters in the x -direction and b centimeters in the y -direction.

A point in polar coordinates requires an angle α , in degrees, and distance from the origin, r . Unlike Cartesian coordinates, the distance does not have a default dimensional unit, so one must be supplied. The syntax for a point specified in polar coordinates is $(\alpha : r \textit{ dim})$, where *dim* is a dimensional unit such as `cm`, `pt`, `in`, or any other $\text{T}_{\text{E}}\text{X}$ -based unit. Other than syntax and the required dimensional unit, this follows usual mathematical usage. See Figure 6.

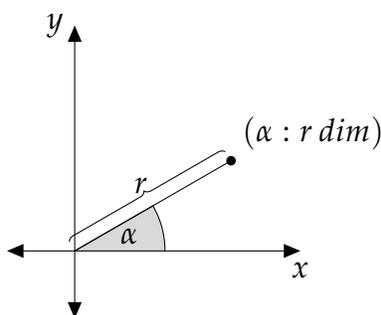


Figure 6: Polar coordinates in TikZ.

It is sometimes convenient to refer to a point by name, especially when this point occurs in multiple `\draw` commands. The command:

```
\path (a,b) coordinate (P);
```

assigns to P the Cartesian coordinate (a, b) . In a similar way,

```
\path (\alpha:r dim) coordinate (Q);
```

assigns to Q the polar coordinate with angle α and radius r .

Figure 7 illustrates the use of named coordinates and several other interesting capabilities of TikZ. First, infix-style arithmetic is used to help define the points of the pentagon by using multiples of 72 degrees. This feature is made possible by the `calc` package [5], which is automatically included by TikZ. Second, the `\draw` command specifies five line segments, demonstrating how the drawing pen can be moved by omitting the `--` operator.

```

\begin{tikzpicture}
  % Define the points of a regular pentagon
  \path (0,0) coordinate (origin);
  \path (0:1cm) coordinate (P0);
  \path (1*72:1cm) coordinate (P1);
  \path (2*72:1cm) coordinate (P2);
  \path (3*72:1cm) coordinate (P3);
  \path (4*72:1cm) coordinate (P4);

  % Draw the edges of the pentagon
  \draw (P0) -- (P1) -- (P2) -- (P3) -- (P4) -- cycle;

  % Add "spokes"
  \draw (origin) -- (P0) (origin) -- (P1) (origin) -- (P2)
        (origin) -- (P3) (origin) -- (P4);
\end{tikzpicture}

```

Figure 7: Using named coordinates.

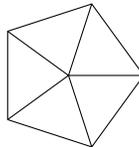


Figure 8: Results produced by Figure 7.

The concept of the *current point* plays an important role when multiple actions are involved. For example, suppose two line segments are drawn joining points P and Q along with Q and R :

```
\draw (P) -- (Q) -- (R);
```

Viewed as a sequence of actions, the drawing pen begins at P , is moved to Q , drawing a first line segment, and from there is moved to R , yielding a second line segment. As the pen moves through these two segments, the current point changes: it is initially at P , then becomes Q and finally becomes R .

A relative point may be defined by providing offsets in each of the horizontal and vertical directions. If P is a given point and Δx and Δy are two offsets, a new point Q may be defined using a `++` prefix, as follows:

```
\path (P) ++(\Delta x,\Delta y) coordinate (Q);
```

Alternately, the offset may be specified with polar coordinates. For example, given angle α and radius r , with an appropriate dimensional unit *dim*, the command:

```
\path (P) ++(\alpha:r dim) coordinate (Q);
```

specifies a new point Q . See Figure 9.

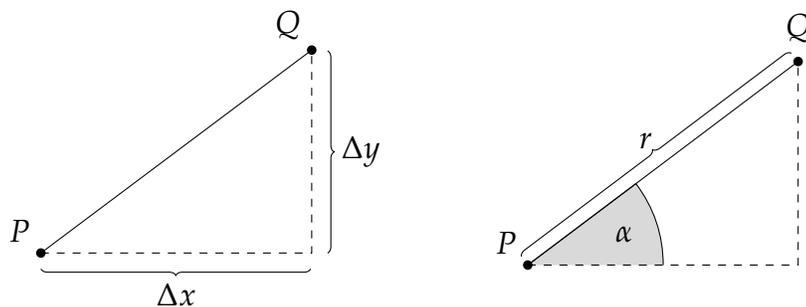


Figure 9: A relative point, Q , determined with Cartesian or polar offsets.

There are two forms of relative points — one which updates the current point and one which does not. The `++` prefix updates the current point while the `+` prefix does not.

Consider line segments drawn between points defined in a relative manner, as in the example of Figure 10. The path is specified by offsets: the drawing pen starts at the origin and is adjusted first by the offset $(1,0)$, followed by the offset $(1,1)$, and finally by the offset $(1,-1)$.

```
\begin{tikzpicture}
  \draw (0,0) -- ++(1,0) -- ++(1,1) -- ++(1,-1);
\end{tikzpicture}
```

Figure 10: Drawing a path using relative offsets.



Figure 11: Output produced by Figure 10.

By contrast, Figure 12 shows the effect of using the $+$ prefix. Since the current point is not updated in this variation, every offset which appears is performed relative to the initial point, $(0,0)$.

```
\begin{tikzpicture}
  \draw (0,0) -- +(1,0) -- +(0,-1) -- +(-1,0) -- +(0,1);
\end{tikzpicture}
```

Figure 12: Drawing a path using relative offsets without updating the current point.

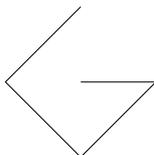


Figure 13: Output produced by Figure 12.

Beyond line segments

In addition to points and line segments, there are a number of other graphic primitives available. These include:

- Grids and rectangles
- Circles and ellipses
- Arcs
- Bézier curves

As previously discussed, a grid is specified by providing two diagonally opposing points and other options which affect such things as the color and spacing of the grid lines. A rectangle can be viewed as a simplified grid — all that is needed are two diagonally opposing points of the rectangle. The syntax

```
\draw (P) rectangle (Q);
```

draws the rectangle specified by the two “bounding box” points P and Q . It is worth noting that the current point is updated to Q , a fact which plays a role if the `\draw` command involves more than one drawing action. Figure 14 provides an example where three rectangles are drawn in succession. Each rectangle operation updates the current point, which then serves as one of the bounding box points for the following rectangle.

```
\begin{tikzpicture}
  \draw (0,0) rectangle (1,1)
         rectangle (3,2)
         rectangle (4,3);
\end{tikzpicture}
```

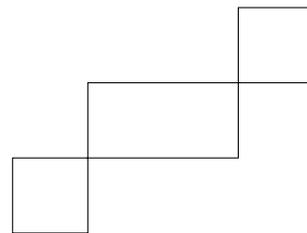


Figure 14: Drawing rectangles.

A circle is specified by providing its center point and the desired radius. The command:

```
\draw (a,b) circle (r dim);
```

```

\begin{tikzpicture}
  \draw (0,0) circle (1cm)
         circle (0.6cm)
         circle (0.2cm);
\end{tikzpicture}

```

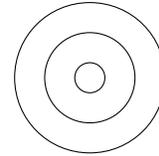


Figure 15: Drawing circles — one draw command with multiple actions.

```

\begin{tikzpicture}
  \draw (0,0) circle (1cm);
  \draw (0.5,0) circle (0.5cm);
  \draw (0,0.5) circle (0.5cm);
  \draw (-0.5,0) circle (0.5cm);
  \draw (0,-0.5) circle (0.5cm);
\end{tikzpicture}

```

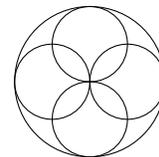


Figure 16: Drawing circles — a sequence of draw commands.

causes the circle with radius r , with an appropriate dimensional unit, and center point (a, b) to be drawn. The current point is not updated as a result. Figures 15 and 16 provide examples.

The situation for an ellipse is similar, though two radii are needed, one for each axis. The syntax:

```

\draw (a,b) ellipse (r1 dim and r2 dim);

```

causes the ellipse centered at (a, b) with semi-axes r_1 and r_2 to be drawn. See Figure 17.

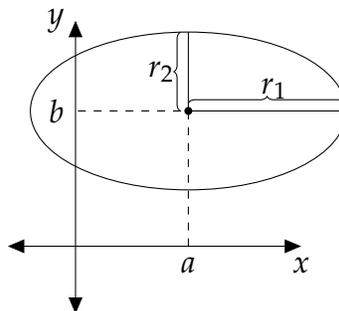


Figure 17: An ellipse in TikZ.

Like `circle`, the `ellipse` command does not change the current point, so multiple ellipses which share the same center point can be drawn with a single draw command, as Figure 18 shows.

```
\begin{tikzpicture}
  \draw (0,0) ellipse (2cm and 1cm)
    ellipse (0.5cm and 1 cm)
    ellipse (0.5cm and 0.25cm);
\end{tikzpicture}
```

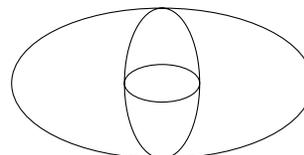


Figure 18: Three ellipses produced with a single draw command.

Arcs may also be specified in TikZ. For a circular arc, what is required is an initial point on the circle, the radius of the circle and an indication of how much of the circle to be swept out. In more detail, the syntax

```
\draw (P) arc ( $\alpha$ : $\beta$ : $r$  dim);
```

draws the arc shown in Figure 19. At first glance it might seem unusual to use the point P and not the center point of the circle. However, when one realizes that the arc might be just one of several components of a draw command, it is very natural to use the point P , as it will be the current point.

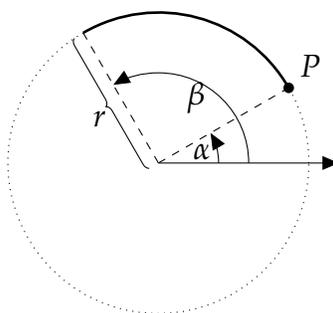


Figure 19: An arc in TikZ.

For example, Figure 20 shows how to draw a portion of an annulus by drawing two arcs and two line segments. This particular figure is drawn by directing

```

\begin{tikzpicture}
  \draw (0:1cm) -- (0:2cm)
        arc (0:60:2cm) -- (60:1cm)
        arc (60:0:1cm) -- cycle;
\end{tikzpicture}

```

Figure 20: Combining arcs and line segments.

the pen in a counter-clockwise fashion—the horizontal line segment, the outer circular arc, a line segment, and finally the inner arc.

TikZ also provides the ability to produce Bézier curves. The command

```
\draw (P) .. controls (C) and (D) .. (Q);
```

draws the curve shown in Figure 21. Four points are needed: an initial point P , a final point Q , and two control points. The location of the control points controls the extent of the curve and the slope of the curve at the initial and final points.

Bézier curves provide for a wealth of variety as Figure 22 indicates.

An alternate syntax for Bézier curves allows for a more convenient specification of the curvature at the starting and ending points. Using polar coordinates with respect to these two points provides this capability. The syntax is as follows:

```
\draw (P) .. controls +(\alpha:r_1 dim) and +(\beta:r_2 dim) .. (Q);
```

See Figure 23.

5 From coordinates to nodes

A *node* is a generalization of the coordinate primitive. Two characteristics of a node are its *shape* and its *text*. A node allows for arbitrary T_EX text to appear within a diagram. The command

```
\path (0,0) node[draw,shape=circle] (v0) {\$v_0\$};
```

defines a node named v_0 , centered at the origin, with a circular shape and text component $\$v_0\$$. The draw option causes the associated shape (in this case, a circle) to be drawn. Figure 24 illustrates how nodes can be used to draw an undirected graph. Notice how line segments which join nodes stop at the boundary

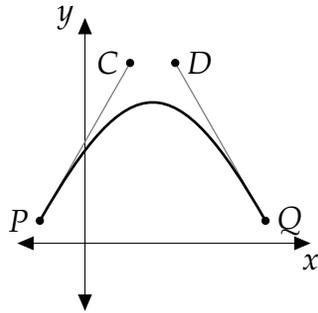


Figure 21: A Bézier curve.

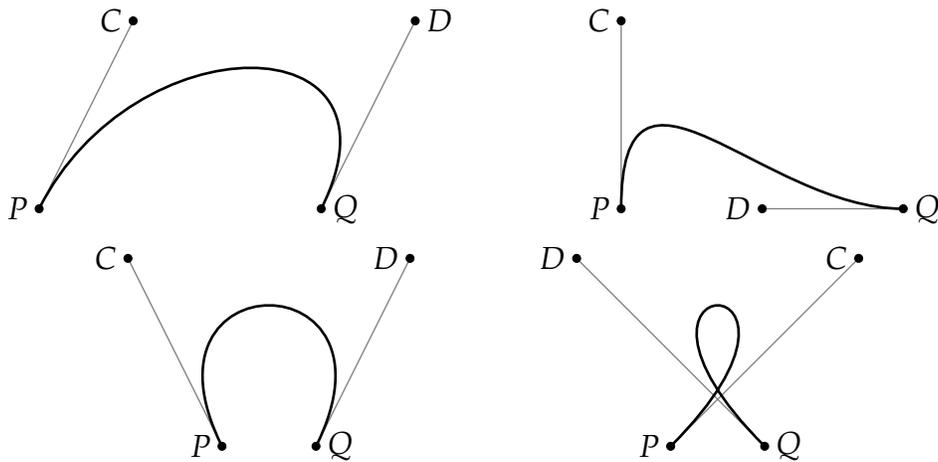


Figure 22: Various Bézier curves.

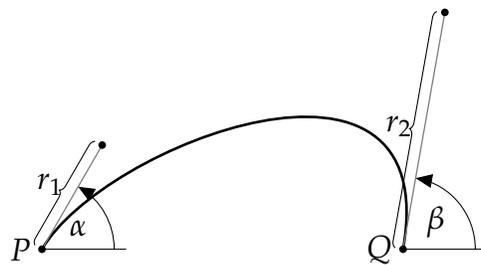


Figure 23: A Bézier curve specified with relative coordinates.

of the shape rather than protruding into the center point of the node. In this example, we have made use of the `tikzstyle` command to factor out code that would otherwise be repeated in each of the node commands.

Additionally, this example illustrates the use of the option `[scale=2.5]`, which indicates the result is to be scaled by a factor of 2.5. Using scale factors allows the picture to be designed in convenient units, then resized as desired. However, scaling a TikZ picture does not scale the font size in use.

```
\begin{tikzpicture}[scale=2.5]
  \tikzstyle{every node}=[draw,shape=circle];
  \path (0:0cm) node (v0) {$v_0$};
  \path (0:1cm) node (v1) {$v_1$};
  \path (72:1cm) node (v2) {$v_2$};
  \path (2*72:1cm) node (v3) {$v_3$};
  \path (3*72:1cm) node (v4) {$v_4$};
  \path (4*72:1cm) node (v5) {$v_5$};

  \draw (v0) -- (v1)
        (v0) -- (v2)
        (v0) -- (v3)
        (v0) -- (v4)
        (v0) -- (v5);
\end{tikzpicture}
```

Figure 24: Drawing an undirected graph with nodes.

There are various features within TikZ which provide fine control over nodes. Many of these are related to how line segments or curves connect a pair of nodes. For example, one can provide specific locations on the node's shape where connections should touch, whether or not to shorten the connection, how and where to annotate the connection with text, and so forth.

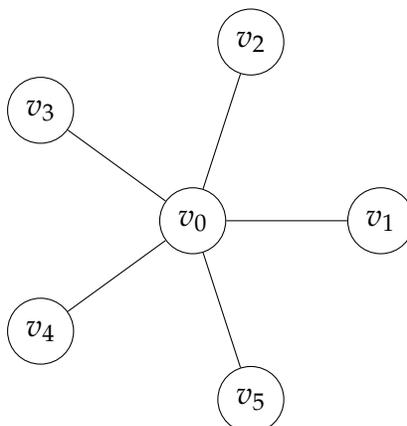


Figure 25: Output produced by Figure 24.

6 Loops

TikZ provides a loop structure which can simplify the creation of certain types of graphics. The basic loop syntax is as follows:

```

\foreach \var in {iteration list}
{
  loop body
}

```

The loop variable, \var , takes on the values given in the iteration list. In the simplest case, this list can be a fixed list of values, such as $\{1, 2, 3, 4\}$ or as an implied list of values, such as $\{1, \dots, 4\}$.

Consider the loop in Figure 26. Four coordinates, x_1 through x_4 are introduced at $(1, 0)$, $(2, 0)$, $(3, 0)$, and $(4, 0)$, respectively. In addition, a small filled circle is drawn at each coordinate.

Figure 27 shows how to extend this idea to yield a bipartite graph. As one might expect, `foreach` loops can be nested, a feature utilized here to specify all the edges in the graph.

Iteration lists need not consist of consecutive integers. An implicit step size is obtained by providing the first two values of the list in addition to the final value.

```

\foreach \i in {1,...,4}
{
  \path (\i,0) coordinate (X\i);
  \fill (X\i) circle (1pt);
}

```

Figure 26: A loop to create four named coordinates

```

\begin{tikzpicture}[scale=2]
  \foreach \i in {1,...,4}
  {
    \path (\i,0) coordinate (X\i);
    \fill (X\i) circle (1pt);
  }
  \foreach \j in {1,...,3}
  {
    \path (\j,1) coordinate (Y\j);
    \fill (Y\j) circle (1pt);
  }
  \foreach \i in {1,...,4}
  {
    \foreach \j in {1,...,3}
    {
      \draw (X\i) -- (Y\j);
    }
  }
\end{tikzpicture}

```

Figure 27: A bipartite graph drawn using loops.

For example,

```
\foreach \angle in {0,60,...,300}
{
  loop body
}
```

causes `\angle` to take on values of the form $60k$, where $0 \leq k \leq 5$.

Specifying pairs of values in an iteration list provides simultaneous iteration over these values. For example,

```
\foreach \angle / \c in {0/red,120/green,240/blue}
{
  loop body
}
```

produces three iterations of the loop body, successively assigning the pairs (0, red), (120, green), and (240, blue) to the variables `\angle` and `\c`.

7 Plotting

A list of points can be plotted using the TikZ `plot` command. Lists can be generated on-the-fly by `gnuplot`[6], read from a file, or specified within a `plot` itself. These three approaches are supported by the following commands:

```
\draw plot function{gnuplot formula};
\draw plot file{filename};
\draw plot coordinates{point sequence};
```

Using other TikZ commands, these graphs can be enhanced with symbols or other desired annotations.

Figure 29 provides an example of one such plot, the graph of $y = \sin(2x)e^{-x/4}$. The curve itself is generated with the command:

```
\draw[smooth,domain=0:6.5] plot function{\sin(2*x)*exp(-x/4)};
```

This command causes `gnuplot`[†] to generate points of the graph, saving them in a file, which is subsequently processed by `TikZ`. The `smooth` option joins these points with a curve, in contrast to line segments. Although not used in this example, the `samples` option can be used to control the number of generated points. The `domain` option specifies the desired range of x values. Everything else which appears in this graph, including axes, tick marks, and multiples of $\pi/2$ have been added with additional `TikZ` commands.

A list of points can be used to create a bar chart, as illustrated in Figure 30. Each of the bars is drawn by command:

```
\draw[ycomb,color=gray,line width=0.5cm]
  plot coordinates{(1,1) (2,2) (3,3)};
```

The `ycomb` option specifies vertical bars are to be drawn and `line width` establishes the width of the bars.

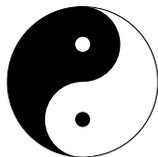
8 Clipping and scope

It is sometimes useful to be able to specify regions of a graphic where drawing is allowed to take place — any drawing which falls outside this defined region is “clipped” and is not visible.

This feature is made available by the `\clip` command, which defines the clipping region. For example,

```
\clip (-0.5,0) circle (1cm);
```

specifies that all future drawing should take place relative to the clipping area consisting of the circle centered at $(-0.5,0)$ with radius 1 cm. Figure 31 shows how to fill a semicircle with clipping. The yin-yang symbol, a popular example, can be easily obtained by superimposing four filled circles on this filled semicircle:



†. To generate points with `gnuplot`, `TEX` must be configured to allow external programs to be invoked. For `TEX Live`, this can be accomplished by adjusting `texmf.cnf` to allow a shell escape.

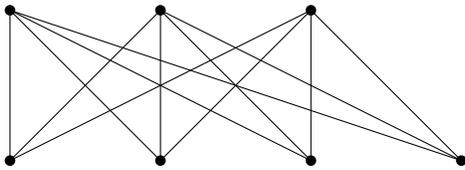


Figure 28: Results produced by Figure 27.

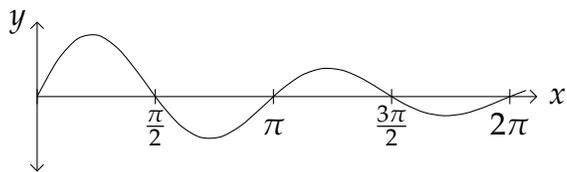


Figure 29: The graph of a function, with tick marks and annotations.

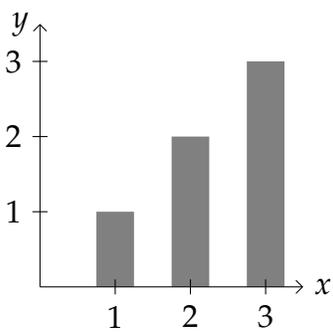


Figure 30: A graph that includes a bar chart.

When multiple `\clip` commands appear, the effective clipping region is the intersection of all specified regions. For example,

```
\clip (-0.5,0) circle (1cm);
\clip (0.5,0) circle (1cm);
```

defines a clipping area corresponding to the intersection of the two indicated circles. All subsequent commands which cause drawing to occur are clipped with respect to this region.

A scoping mechanism allows a clipping region to be defined for a specified number of commands. This is achieved with a scope environment. Any commands inside this environment respect the clipping region; commands which fall outside behave as usual. For example,

```
\begin{scope}
  \clip (-0.5,0) circle (1cm);
  \clip (0.5,0) circle (1cm);
  \fill (-2,1.5) rectangle (2,-1.5);
\end{scope}
```

shades the intersection of two overlapping circles, since the filled rectangle is clipped to this region. Commands which follow this scope environment are not subject to this clipping region. Figure 32 shows a complete example which makes use of `\clip` and scoping.

The scoping mechanism may also be used to apply options to a group of actions, as illustrated in Figure 33. In this example, options to control color and line width are applied to each of three successive `\draw` commands, yielding the top row of the figure. At the conclusion of the scope environment, the remaining `\draw` commands revert to the TikZ defaults, yielding the lower row of the figure.

```
\begin{tikzpicture}
  \draw (0,0) circle (1cm);
  \clip (0,0) circle (1cm);
  \fill[black] (0cm,1cm) rectangle (-1cm,-1cm);
\end{tikzpicture}
```

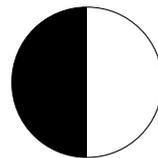


Figure 31: An example of clipping.

```

\begin{tikzpicture}
  \draw (-2,1.5) rectangle (2,-1.5);
  \begin{scope}
    \clip (-0.5,0) circle (1cm);
    \clip (0.5,0) circle (1cm);
    \fill[color=gray] (-2,1.5)
      rectangle (2,-1.5);
  \end{scope}
  \draw (-0.5,0) circle (1cm);
  \draw (0.5,0) circle (1cm);
\end{tikzpicture}

```

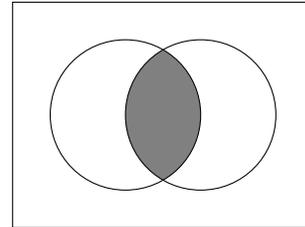


Figure 32: Using clipping and scope to show set intersection.

```

\begin{tikzpicture}
  \begin{scope}[color=gray,line width=4pt]
    \draw (0,0) -- (2,2);
    \draw (2,0) -- (0,2);
    \draw (-1,1) circle (1cm);
  \end{scope}
  \draw (0,0) -- (-2,-2);
  \draw (0,-2) -- (-2,0);
  \draw (1,-1) circle (1cm);
\end{tikzpicture}

```

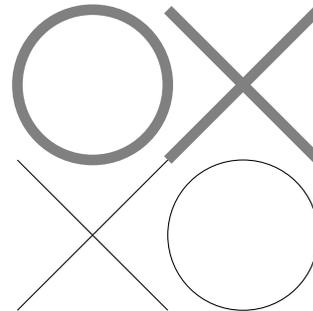


Figure 33: Using scope to apply options.

9 Summary

TikZ, a high-level interface to PGF, is a language-based tool for specifying graphics. It uses familiar graphics-related concepts, such as point, line, and circle and has a concise and natural syntax. It meshes well with pdf \LaTeX in the sense that no additional processing steps are needed. Another positive aspect of *TikZ* is its ability to blend \TeX fonts, symbols, and mathematics within the generated graphics.

We are especially indebted to Till Tantau for developing *TikZ* and for contributing it to the \TeX community.

References

- [1] Robin Fairbairns, ed. *The UK \TeX FAQ*. Available from <ftp://cam.ctan.org/tex-archive/help/uk-tex-faq/letterfaq.pdf>.
- [2] John Hobby. *Introduction to MetaPost*. Available from http://cm.bell-labs.com/who/hobby/92_2-21.pdf.
- [3] Uwe Kern. *Extending \LaTeX 's color facilities: the xcolor package*. Available from <http://www.ctan.org/tex-archive/macros/latex/contrib/xcolor>.
- [4] Till Tantau. *TikZ and PGF, Version 1.01*. Available from <http://sourceforge.net/projects/pgf/>.
- [5] Kresten Krab Thorup, Frank Jensen, and Chris Rowley. *The calc package: infix arithmetic in \LaTeX* . Available from <ftp://tug.ctan.org/pub/tex-archive/macros/latex/required/tools/calc.pdf>.
- [6] Thomas Williams and Colin Kelley. *gnuplot*. Available from <http://www.gnuplot.info/>.

[Journal](#)
[home page](#)
[General](#)
[information](#)
[Submit an](#)
[item](#)
[Download](#)
[style files](#)
[Copyright](#)
[Contact us](#)

THE PracTeX Journal



Square concepts

Dirk Baechle

Abstract

The following text, available in [HTML](#) as well as the regular [PDF](#) version, contains some basic chess concepts and advice, presented in the form of tasks. It was prepared for the *Distractions* column of the TeX online PracTeX Journal, using the program ChessTask, and is aimed at hobby players of any strength. The given positions are all taken from my own games for the chess club SK Soltau, so they are not constructed examples but actually happened. While the tasks themselves are kept rather terse, I try to elaborate things a bit further when I unveil the correct moves at the end of the article. So even if you did not find the solution in first place, you can hopefully learn from the additional information.

Enjoy!

Dirk Baechle

- [PDF version of paper](#)
- [Comment on this paper](#)
- [Send submission idea to editor](#)

Page generated March 14, 2007 ; [TUG home page](#); [search](#); [contact webmaster](#).

Square concepts

Dirk Baechle

Email dl9obn@darcd.de
Website <http://chesstask.sourceforge.net/>
Address Hamburg, Germany

Abstract The following text contains some basic chess concepts and advices, presented in the form of tasks. It was prepared for the *Distractions* column of the T_EXonline journal PracT_EX, using the program ChessTask, and is aimed at hobby players of any strength. The given positions are all taken from my own games for the chess club SK Soltau, so they are no constructed examples but actually happened. While the tasks themselves are kept rather terse, I try to elaborate things a bit further when I unveil the correct moves at the end of the article. So even if you did not find the solution in first place, you can hopefully learn from the additional information.

Enjoy!

Square concepts

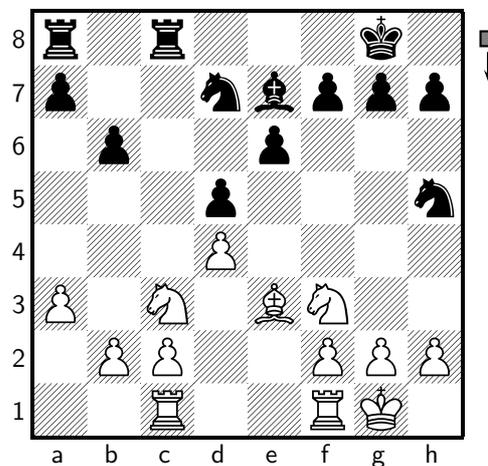
Trivial tactics

At the beginner level, chess is all about tactics. Among the first things for a newcomer to learn are,

1. The rules (of course),
2. the values of the single pieces relative to each other and
3. how to compute series of captures.

It is vital to calculate capture variants up to the end to ensure that the pieces still even afterwards. If an unbalanced stock of material results from a tactical combination this means one side has gained an advantage. And if you are the one that is behind, it most often means that you are about to lose.

Task A



So let us start with a simple tactical problem, for warming up a little. Above you see a game which is still in the opening phase. Both players have developed nicely and positioned their pieces such that each controls a lot of squares—always a good thing to do, by the way. Whites last move was ♖a1–c1. How can Black win a pawn immediately?

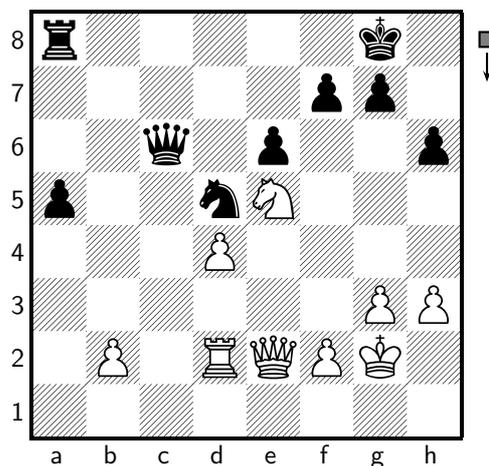
Time is the enemy

If you are a hobby player, you might never have played with a chess clock yet. These little gadgets really add another level of complexity to your fight against an opponent. The additional dimension of time is always ready to break your neck in a complicated situation: you either spend all your time, which means you lose immediately, or you play too quickly and make a severe fault.

The latter happens a lot, even if no clock is involved and you enjoy a nice game of chess against a friend. So here is my advice for this section beforehand: Take your time, even if you are directly attacked by your opponent and think that you know exactly what to do. If you are in check and see only one escape, try to look over the whole board and fathom your possibilities. Often a second alternative shows up that gives you more counterplay, or even the initiative sometimes.

In the next two games that are presented, I fail to see a better continuation because I do not have enough time. In both situations, only a few seconds are left to make the 40th move for the first time control. I was forced to draw quickly... you are not, so try to sit back and think twice!

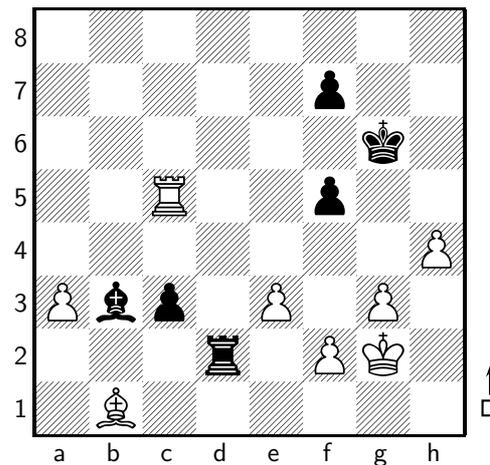
Task B



Whites last move was Nf3-e5 which is a blunder. It looks as if the black queen has to move because it is under attack by the knight. I never fail to tell my chess kids how important it is to care about threatened pieces first and retreat them in such a situation, instead of simply "threatening back". This, combined with the ticking clock, led me to a quick Qd6 and I passed the time control. How could I

have ended the game in two moves?

Task C



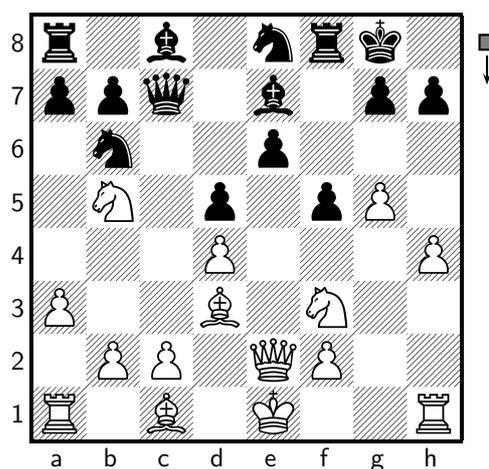
In this position I am two pawns ahead, but Black threatens to promote his advanced pawn on c3 soon. His last move was ♔e6-b3 . I quickly play the tempting ♙xc3 but with some more time on my clock I might have noticed that there is a better move. Try to find the combination that leads to a, more or less, clear win for White.

Plan your escape

When you try to come up with your next move, always think about your retreat. Well, if you plan to sacrifice your figure or want to exchange it anyway, you certainly do not have to care. But if you spot an alluring square deep within the enemy's territory, ask yourself: "What happens if I get attacked? Is there a safe way back?"

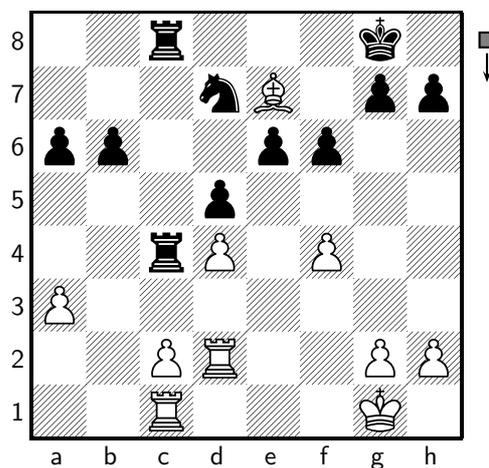
The following two tasks demonstrate how not to do it...

Task D



In this diagram my, admittedly questionable, idea was to get the black ♞b6 to d6 via c4. For this to work, I had already placed my queen on c7. After White's last move ♞c3-b5 I should have changed my mind and retreated. Instead, I stuck to my plan and opted for the weak ♞c6 . What is now the best move for White?

Task E

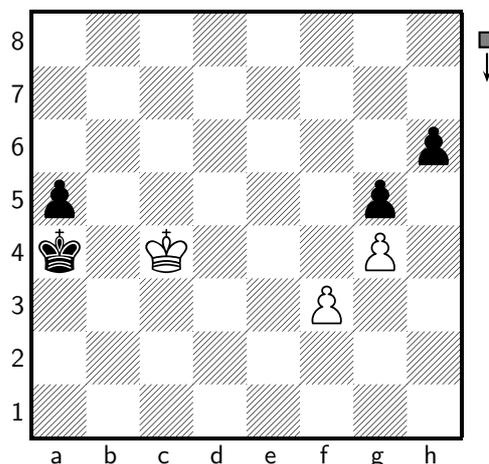


This time I am on the lucky side again, having the black pieces. White's last move was f2-f4. Why is this a blunder?

Keep your opponent busy

The two last tasks are definitely for the more advanced players. They do not deal with concrete moves and captures but share a basic theme. In this first part I will lay out the concept itself directly, while showing how it can be employed in a more complicated situation afterwards. Hence, I suggest that you ponder over task F a bit, then take a peek into the solution before progressing to part two of this section.

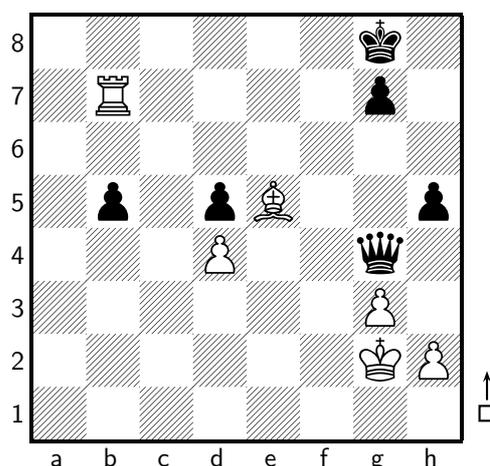
Task F



The pawns on the right side are blocked, they can not move forward without getting lost. Black is a pawn ahead, but it looks as if he can not do very much to prevent a draw. On ♔a3, White plays ♔c3 and after another ♔a4 the obvious ♕c4 follows. White encircles the black king in front of the pawn, such that no advance is possible.

So how can you get the pawn through and win the game? Try to come up with a plan for this, which does not mean that you should calculate endless variations. Instead, imagine what the position should look like to give you a clear advantage.

Task G



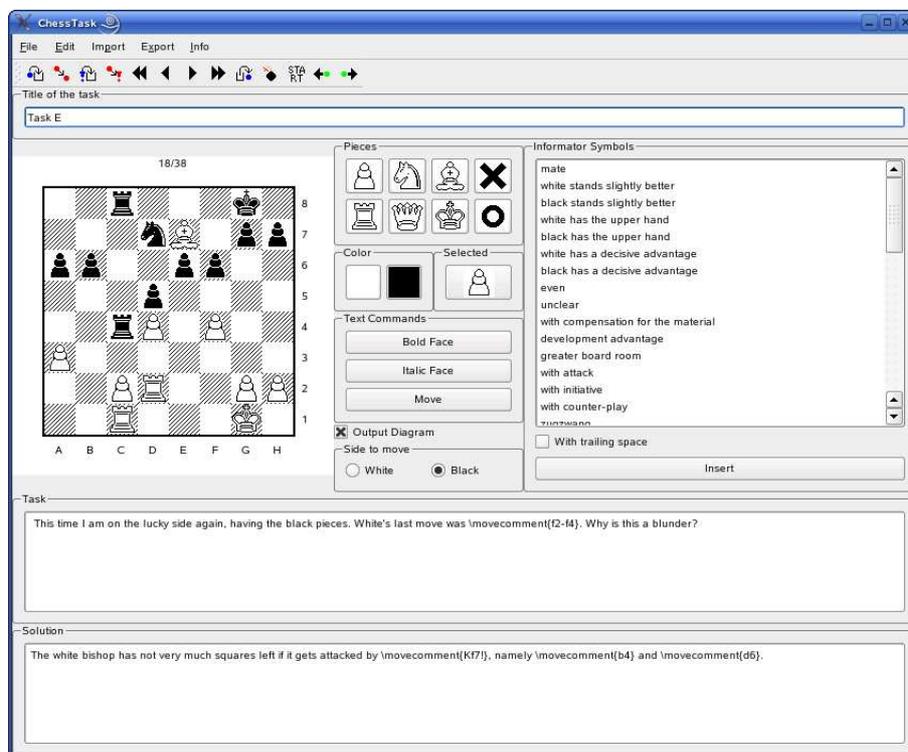
This asymmetric position—rook and bishop against a single queen—needs even more foresight than the previous one. Try to visualize the board with only the pawns and kings remaining. Is this a draw or can one side win this fight? What are the main ideas to follow and how does the basic concept of “keeping the opponent busy” fit in? Is it, for example, a good idea to play ♖×b5 in order to even the material?

About ChessTask

I created the first draft of ChessTask when I started tutoring some of the kids in our local chess club. I wanted to prepare sheets with tasks on them, such that they could do some homework during the week. All I found for \LaTeX at that time was the style `chess` by Piet Tutelaers ([CTAN:fonts/chess/chess](https://ctan.org/ctan/packages/fonts/chess/chess)). It is OK for replaying and commenting complete games but setting up an arbitrary position is not really user-friendly. I needed some sort of graphical interface for editing diagrams more quickly.

Another problem that I tried to solve was the separation between the actual tasks and their solutions. I wanted to edit each single task in a whole but produce output with just the tasks or both, without having to reedit my input.

Up until today, this is how ChessTask mainly works. It manages a list of task “chunks”, where each entry consists of a diagram, a task and a solution. You give it all the information you have and during the export to a \LaTeX file you can specify what you would like to see in the final PDF.



The figure shows the main screen of ChessTask, which is what you will work with most of the time. You have the diagram at the top and can enter a text for the task and, if required, give a solution. Your text is formed as a sequence of these “task sets”. It’s as simple as that.

During the years, a lot of features were introduced that helped me in my work. One of the first things that I added was the output to HTML files. Other output formats and options like the export of whole directories and the special “ChessCards” format followed. The import functions offer to read in whole sets of FEN or EPD lines from a file. While converting a single position, ChessTask even recognizes EPD commands like “am” (avoid move) and adds an appropriate comment to the text of the task.

For a more complete overview, visit <http://chesstask.sourceforge.net> and read the online manual, please. Even better, try out the program itself and check whether it fits your needs.

Currently, a new version of ChessTask is under development which will support things like:

- Chess diagrams in solutions,
- customizable L^AT_EX and HTML headers/trailers,
- support for the “side to move” flag in diagrams,
- and much more.

It will take some months for the next release to appear, so be patient please and take a peek at the homepage and download area from time to time.

Finally, I want to direct your attention to a few packages for typesetting chess with L^AT_EX:

- SKAK ([CTAN:fonts/chess/skak](#))

This excellent package is what ChessTask relies on for creating all the diagrams, informant symbols and move stuff in general. It is a great replacement for the CHESS style and helps a lot when writing texts about chess. The moves of a game can be input in PGN (Portable Game Notation) and single board positions are set up by FEN (Forsyth Edwards Notation) strings. You can highlight single squares, draw move arrows, hide pieces. . . the list goes on and on. The latest version 1.4 has been revised by its author Torben Hoffmann and Ulrike Fischer for a better support of different chess fonts in notation text and the diagrams. It is now based on the CHESSFSS package (see below), making it possible to draw boards in any size you like, for example.

- SKAKNEW ([CTAN:fonts/chess/skaknew](#))

Ulrich Dirr took care of the original SKAK font sources and converted them from METAFONT to Type1. He added a lot of corrections to the figures and informant symbols and the result looks great, as you can see in this article. Highly recommended if you use the latest version of SKAK.

- CHESSFSS ([CTAN:macros/latex/contrib/chessfss](#))

This package, aiming at writers of chess styles, offers commands to use and switch between chess fonts. It is based on the L^AT_EX font selection scheme (NFSS) and was written by Ulrike Fischer.

- CHESSBOARD ([CTAN:macros/latex/contrib/chessboard](#))

Another nice add-on by Ulrike Fischer for the extended display of board diagrams. Thanks to her effort you can crank the look of your chess boards

to eleven by using bended move arrows or exotic pieces like for fairy chess, highlighting squares, colouring the background, colouring the figures, colouring... whatever.

– ENPASSANT ([CTAN:fonts/chess/enpassant](http://ctan.org/tex-archive/fonts/chess/enpassant))

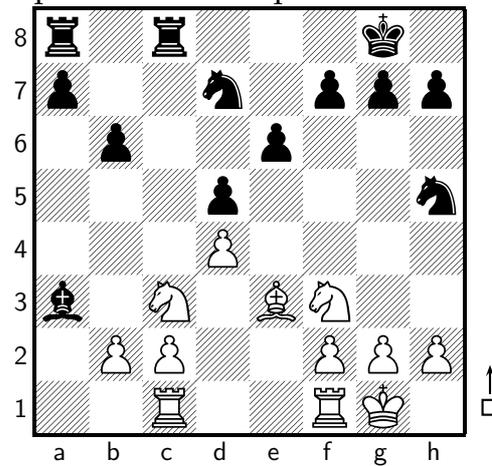
A collection of Type1 fonts, ready for use with the CHESSFSS package. It contains most of the TrueType chess fonts available at <http://www.enpassant.dk> and was compiled by Ulrike Fischer.

All together, they can be very powerful and offer a lot of options. So if the output of this article convinces you and you plan to write about chess: Try them out today!

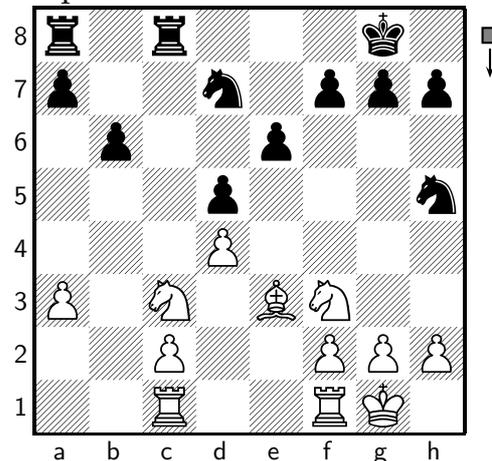
Solutions for: Square concepts

Task A

$\text{♞}\times\text{a3}$ uncovers the weak point in White's position.



The pawn b2 is (or better: was) responsible for covering the pawn at a3 and the $\text{♞}\text{c3}$, which is attacked by the $\text{♞}\text{c8}$. He can not do both at the same time, so after taking the black bishop with $\text{b2}\times\text{a3}$...



... Black captures the knight with $\text{♞}\times\text{c3}$, is a pawn ahead and additionally attacks the white pawn on a3.

This all happened because White tried to develop his rook on a1. While moving it to c1 he did not realize that it had a task to attend to: covering the pawn a3. Instead of sharing work with his buddy, it said: "Goodbye, I'm off." Not really a

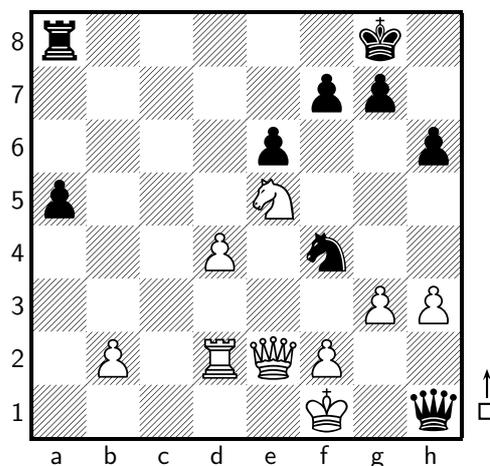
nice thing to do, is it?

My advice in general is: before making a move, check that the piece does not have a concrete task. This can be covering an own piece, but also attacking an important square such that your opponent can not easily take it under his control. In the same way you can ask the following questions after each move of your contrahent: "Did he leave one of his pieces uncovered?" "Is a square unoccupied now, such that I can safely place one of my figures on it?"

Try to find out what has changed on the board and how you can exploit the new situation!

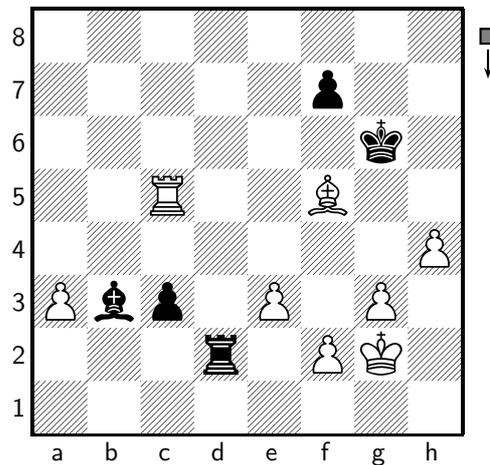
Task B

It would have been better to take a closer look and see the "mate in two" with ♞f4+ . This powerful double-check leaves no escape for White. On ♔f1 the simple ♚h1\# follows...

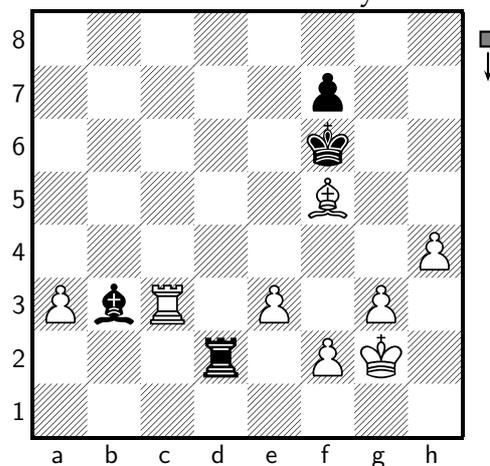


... while ♔h2 is answered by ♚g2\# .

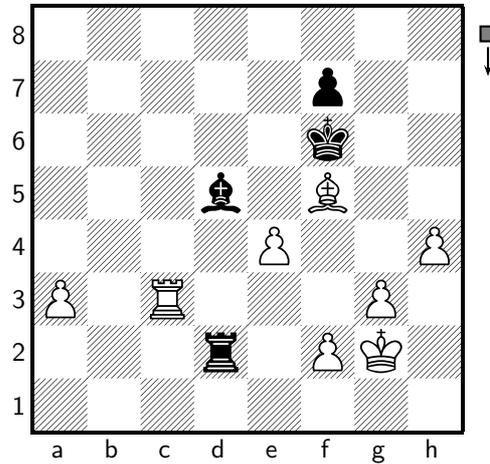
have saved my day.



The king has to move, let us assume that Black plays ♔f6 , and now the rook captures the next pawn ♖xc3 and simultaneously attacks the bishop on b3.



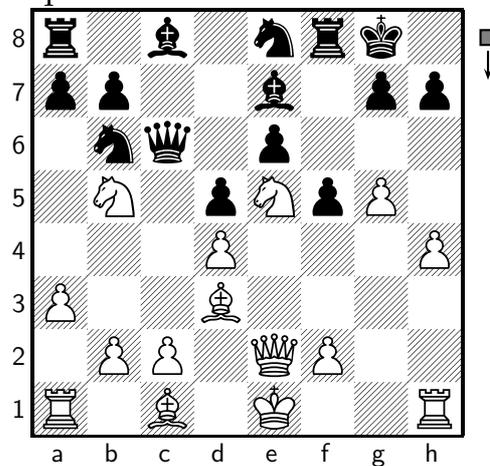
If Black captures the bishop ♜xf5 , White answers ♖xb3 . Another try would be ♞d5+ but then the simple e4 follows.



The white bishop is covered, Black's is attacked and White is now four pawns ahead. That is really what I call a decent alternative!

Task D

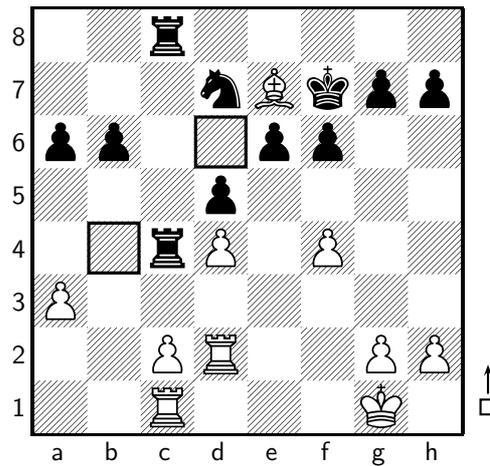
White plays ♞e5 and the queen is lost!



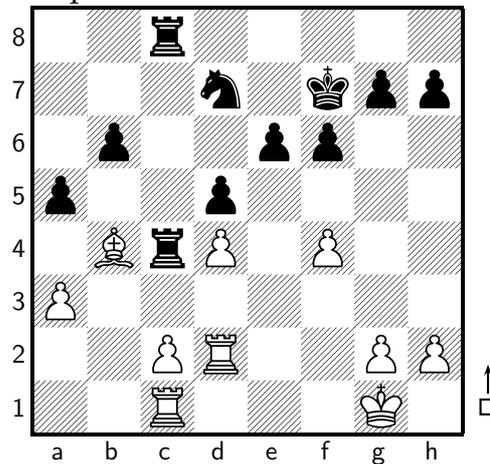
Quite embarrassing... and it is not even near the "enemy's territory". Just like when this happened to me, I am a bit speechless, so without further comments on this little disaster my advice: "Do better than me in your own games and try to keep your eyes open!"

Task E

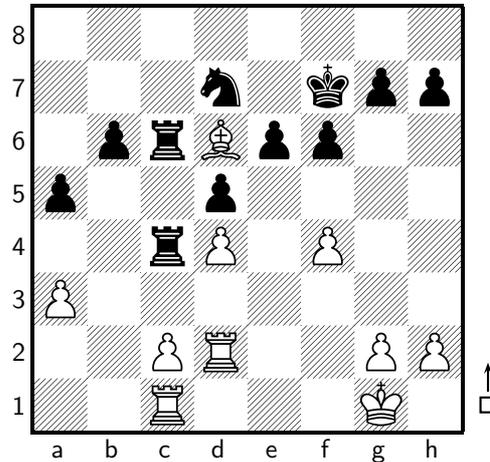
The white bishop has not very much squares left if it gets attacked by ♔f7!, namely b4 and d6.



With f4 his own buddy took him the last chance to escape via the diagonal d6–g3. Regard that ♗b4 is only a temporary resort, because after the immediate answer a5 there is only one place available,



and that is ♗d6. Now follows the lethal ♖8c6 and the bishop is lost.



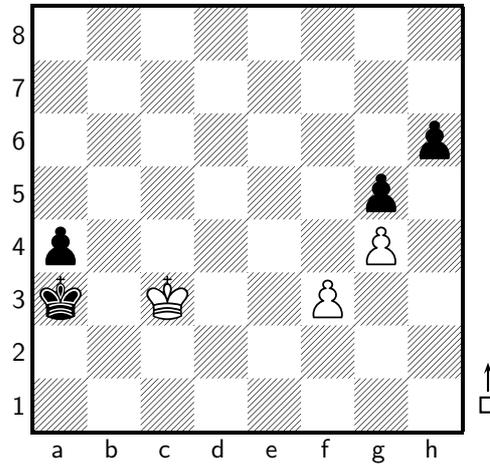
Playing it the other way round with 1. ♔f7 ♕d6 2. ♖8c6 ♘b4 3. a5 does not help either.

Task F

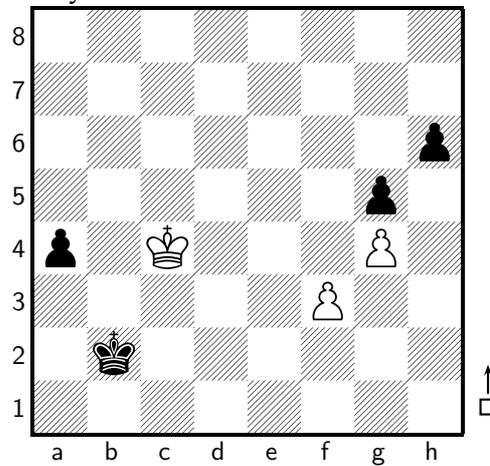
The answer is: "You can not advance the pawn to the last rank, but you can win the game!". If the blocked pawns to the right would not be there, it would indeed be a draw. But having them on the board lets you tackle your contrahent down.

All you have to do is: "Let go of the advanced pawn." Leave it alone, and start off to attack the remaining pawns of White. Since the white king has to capture a4 first, else it promotes, you will arrive ahead of time and win both pieces. This should be your plan for this constellation: "Keeping White busy by forcing him to capture your advanced pawn and decoying him further away from the center of the board."

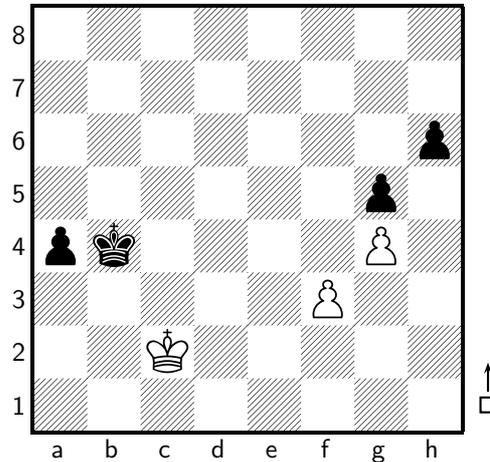
Now let us look at this from a more technical point of view and watch a concrete series of moves. At first we have to get out of the corner, so after the moves 1. ... ♔a3 2. ♔c3 Black plays 2. ... a4



Notice how things are turning here. Now Black, instead of White, holds what is called the “opposition”. He can wait for the move of White and then decide where to go next, which is often an advantage in endings like this one. However, White does not have a lot of choices... he has to play 3. ♔c2. On all other moves, like 3. ♔c4, Black can win by 3... ♚b2

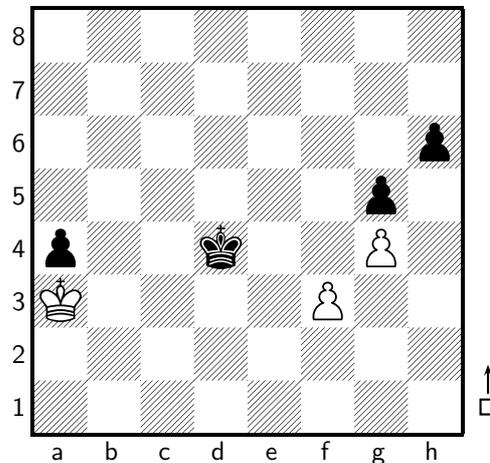


which guards the a-pawn on his last three steps to the promotion. But after the mandatory 3. ♔c2 Black escapes his prison with 3... ♚b4.



Again, White has no choice but to further attack your isolated pawn with 4. ♖b2. If he tries something fancy like 4. ♔d2 you can always strain at his leash by advancing the a-pawn. Keep him busy, he has to come back!

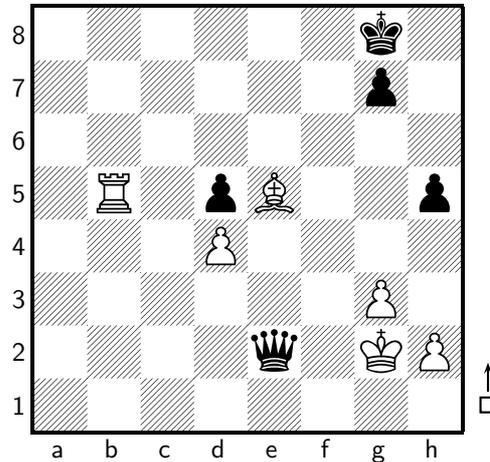
Assuming that White gives in by 4. ♖b2 ♔c4 5. ♖a3 ♔d4, the following position is reached:



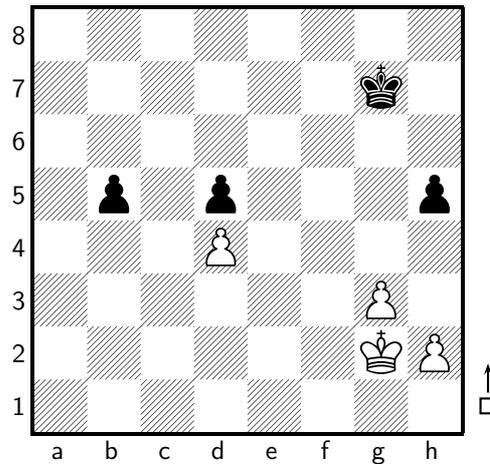
The advantage for Black is evident now: White captures the a-pawn but will have to stop two connected pawns on the g- and h-file with his bare hands, so to speak.

Task G

The greedy ♖xb5 is definitely the wrong road to take. After Black's reply ♗e2+ the rook is lost in the next move.



The best thing White can do in the starting position is to exchange rook and bishop for the queen and a pawn with 1. ♖×g7+ ♚×g7 2. ♙×g7 ♜×g7. The resulting diagram...

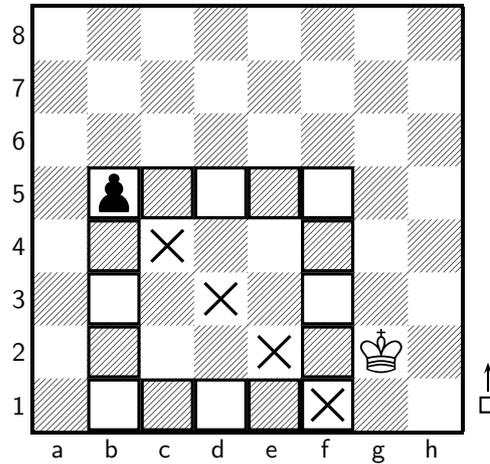


... is now examined further to show that it is, in fact, a draw!

We try to decompose the problem into smaller "chunks" by blending out the pieces that do not affect the current part we look at. This is not always possible, but here we can subdivide the board into three main areas of interest:

1. We have a passed pawn on the b-file (good for Black),
2. two blocked pawns on the d-file and
3. two against one on the g- and h-file (good for White).

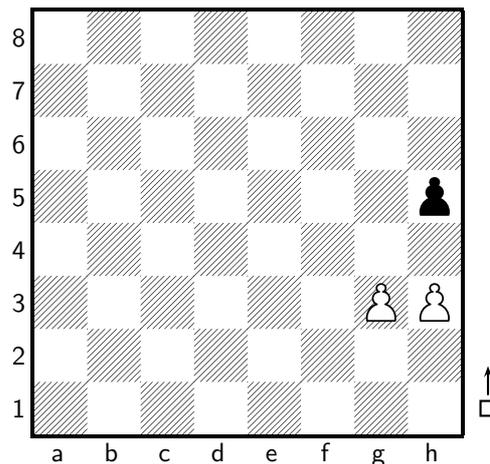
Let us neglect the block on the d-file and see if we can still intercept the b-pawn.



If you try to catch a single pawn with your king, there is a simple rule that goes as follows: “Starting at the pawn, the diagonal to the last rank spans a square. If your king gets into this square with his next move you can get the pawn.”

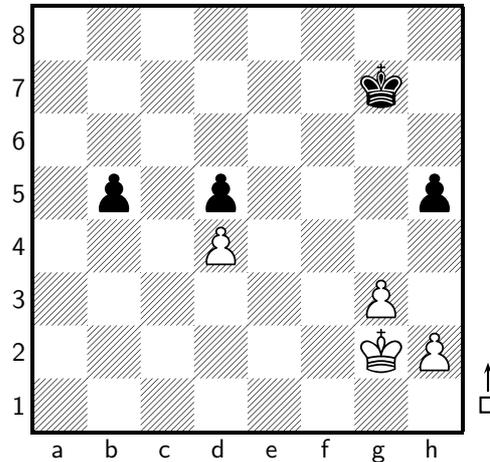
Here, the square shows that White’s next move has to be something like 3. ♔f3.

Next, we take a look at the right side of the board. White’s goal here is to reach the following pawn structure:

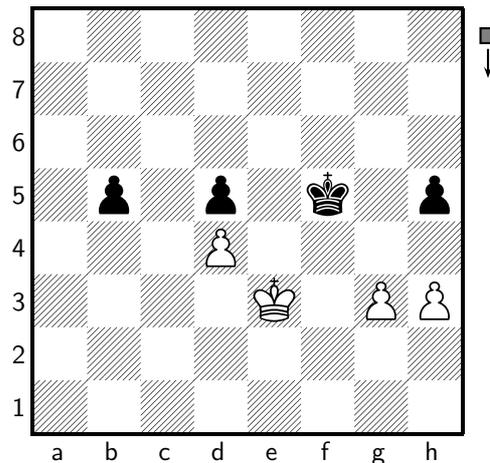


It ensures that we can create a passed pawn at any time with g4 and this is what “keeps Black busy” and guarantees the draw.

Now we blend in all pieces again and look at the actual moves that have been played.



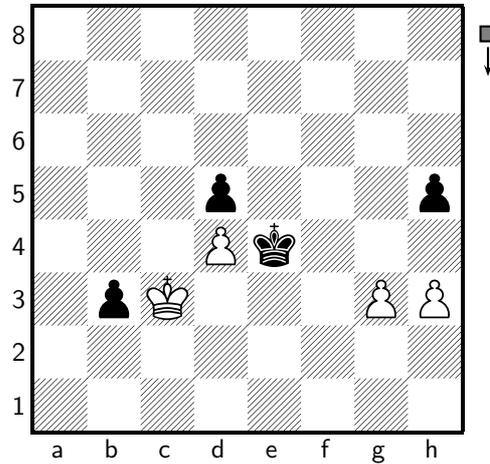
3. ♔f3 ♕f6 4. ♖e3 ♗f5 5. h3 and the white pawns to the right are ready to give Black something to think about.



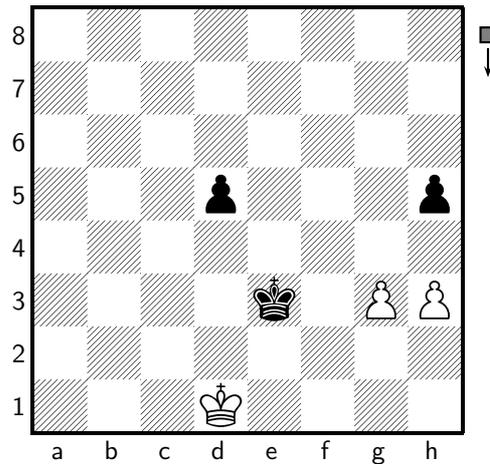
Now what is Black to do? He can not get through on the right flank and when he tries to support his b-pawn

1. he will give White enough time to block it and
2. is moving further away from the threat g4.

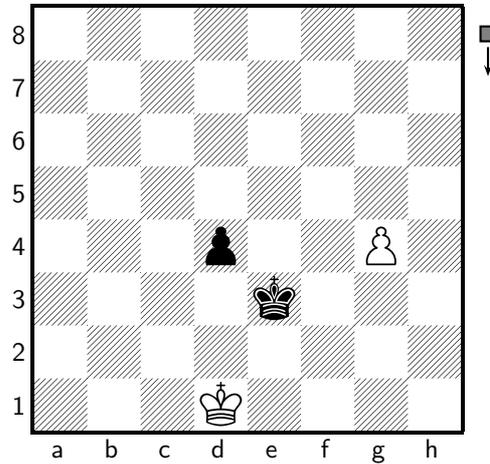
Feeling the pressure, he decides to make for a change and wants to give White some work with 5. ... b4 6. ♖d3 b3 7. ♗c3 ♖e4.



Both kings are ready to capture a pawn, but can White stop the d-pawn once it is free? Yes, after 8. ♔×b3 ♕×d4 9. ♕c2 ♕e3 10. ♔d1



his way is blocked. Now the “keep busy” pattern shows up again and with 10. ... d4 .g4 h×g4 12. h×g4 the game ends with the final position



which is a draw. Do not worry too much about all the moves that have been played or could have been played. Try to focus on patterns and memorize how the decomposed pawn structures blend in with the “keep busy” theme. That’s all this task is about in first place.

Finally, a short word to all you advanced players out there. I did not go into very much depth for this task, so there may be better sidelines and deep and thoughtful variants around somewhere. . . not to speak of faults and errors.

On purpose, I tried to pick out a single certain aspect—like in the tasks before—in order to present what I designed ChessTask for: as an aid in teaching basic chess concepts to beginners.

[Journal](#)
[home page](#)
[General](#)
[information](#)
[Submit an](#)
[item](#)
[Download](#)
[style files](#)
[Copyright](#)
[Contact us](#)

THE PracTeX Journal



Creating book covers with PSTricks

Yuri Robbers and Annemarie Skjold

Abstract

The title and the cover of the book are the very first impressions a potential reader is likely to get from a book. It is therefore of utmost importance to make sure these impressions are good ones. This paper will give some general notes on cover design, and some specific notes on implementing such designs using PSTricks.

Yuri Robbers holds a degree in Animal Behaviour and is a teacher, a researcher and a published author. He's always had a keen interest in typography, possibly because his father is a professional typographer. Ever since he discovered LaTeX in 1995, he's always done his best to avoid using word processors, and he has embarked on a quest to learn as much as possible about TeX and its derivatives and to apply this knowledge whenever possible. Contact him at yuri.robbers@gmail.com

Danish author Annemarie Skjold has a background in architecture, set design, and journalism. Currently residing in Holland, she is co-director of a small independent publishing company, where she is chief designer. Skjold writes freelance articles, short stories and has published two novels. She illustrates her own work. Contact her at askjold@english-expression.com

- [PDF version of paper](#)
- [Comment on this paper](#)
- [Send submission idea to editor](#)

Creating Book Covers using PStricks

Yuri Robbers and Annemarie Skjold

Email yuri.robbers@gmail.com

Abstract The cover of a book presents the first impression for any potential readers or book buyers. Therefore it is of vital importance this impression is a good one. This paper offers general notes on cover design, and specific notes on implementing such designs using PStricks.

1 Introduction

It is said one shouldn't judge a book by its cover, however, book buyers are certainly influenced by it. As a writer, you naturally want your book to be as attractive to the buyer as possible. When people step into a book shop with the intention of buying a book, they usually look around, see what is on display and browse the shelves. Some books will jump out at them, some will not. It is usually the cover (including the spine), title, or a specific author, that attracts a buyer to pick up certain books. Even when browsing online bookshops, it is the cover that first draws the shopper's attention.

This paper provides step by step information for the writer who wants to create his or her own book cover using PStricks¹ (Voß, 2007). In the first section, I will discuss some very general aspects of cover design. In the second section, I will explain how to implement these ideas and I will be providing examples of everything I do as we go along.

2 Deciding to design your cover

You have written your book, edited it, settled on the title, made it perfect, and now comes the final icing on the cake. The cover. Larger publishing houses provide cover design in their in-house art and graphics department, and it is

1. <http://www.pstricks.de>

decided well in advance often without the writer's input at all. But for the do it yourself freelance writer and print on demand self-publisher, the cover is often the last thing to be tackled, and can suddenly present quite a quandary. Some authors may prefer to find a reasonably priced freelance graphics service, or a template download, and there are plenty available online. For instance, a good place to start is the Lulu website². This website also has some excellent articles on the subject.

But for those writers who are keen to try it for themselves, it can be very satisfying to design and create one's own cover. Designing an attractive cover that is going to sell its book is definitely a knack, and graphic arts students take years learning about complicated issues such as 'balance of form' and the use of colour, and still do not always get it right. One reason for this however, could be a lack of connection to the book itself, and it could be argued that the writer knows the book intimately and therefore who better to present it?

3 Where to start

A good general principle to keep in mind is clarity. Does the cover send a clear message? Is the title easily readable? Does the image instantly suggest what the book is about?

Before you begin a design project of any sort, it is always recommended to undertake a little background research. Go to a bookstore and browse the shelves. Familiarize yourself with the different genres, especially the one your book would be in, and how they are presented. Take a note of which books stand out from the rest and why you think they do. Are they books you would want to buy? Another general principle of good design is, that if you, as the designer, are happy with it, others will like it too. Any background research would not be complete without also browsing the websites of established book cover designers. For instance, well known designer Will Harris explains online his own approach to some of his excellent cover designs³.

As a basic design start point, three rules to keep in mind are:

2. <http://www.lulu.com>

3. http://www.will-harris.com/covers_0.html

1. keep it simple;
2. illustrate the title;
3. define the essence of the book.

3.1 On keeping it simple

Clarity is vital, and with a book cover, less really is more. Your cover and its message should be easily seen and understood from across a crowded bookstore. The title should be easy to read on both cover and spine. Minimize the use of colour to say, three or maybe four shades. Decide at the outset whether you want to use garish contrasts to make your point, or to harmonise, blending colours that are gentle and stunningly attractive.

Do resist the temptation to enhance your cover with a generous display of symbols. Just one or possibly two floating grails, framed portraits or blood soaked daggers are quite enough for any browsing book buyer to take in at a glance. Clutter is off-putting and extremely detrimental to the successful book cover design.

3.2 On illustrating the title

For a truly cohesive, clear message, it helps if the title and the cover are shouting the same thing. During your background research, you may have found — even subliminally — that the most successful covers are the ones in which the image backed up the title. Deeply abstract and bizarre expressions of life's tangled webs in tumultuous colour and profusion beneath the estranged letters 'Red sails in the Sunset' will puzzle the reader at best, or at worst send them straight on past to the cover with the indigo ocean, wide golden sky and a single red sail on the horizon.

It just makes more sense. Readers generally do not like to work too hard. They want to feel safe that they will be entertained or informed well and easily by your book. A strong link between image and title instantly gives them the idea they will be able to understand all of your work.

3.3 On defining essence

Take some time to sum up the theme and mood of your book in one short sentence — just two or three words. What emotions, colours and symbols do the words evoke? Can you also sum up this imagery into a main colour, a central image or symbol, a single defining object or character? You are seeking the essence of your text, in a visual expression. The exercise you underwent in deciding the title of your book should help you in deciding the essential images for your cover.

A further guide to defining your theme is to decide what is appropriate for the genre of your book, and to choose images within that framework.

4 A survey of genres and style

I will now give some more specific guidelines, all based on research in bookstores both physical and online, and — most importantly — conversations with the staff of the local bookshops. Of course I cannot give a fail safe recipe for creative design, but I can give some general guidelines that will hopefully help you avoid the more common pitfalls, and maybe even inspire you.

4.1 Scientific works

There are several basic options when covering scientific books. There is a simple way, which one would not really get away with any other type of book: using a plain cover with just one colour and some text in a different colour. One sees this often, especially in so-called “student editions” of text books. The main reason for choosing this option is that it is much cheaper to print. To enhance this simple layout one could opt to include a graph from the book or a graph that pertains to the book. This gives a proper “scientific” look to your book that will attract most scientists and scare away most laypeople. Then there is the option of choosing a photograph or artwork, either relevant to the contents of the book or just pretty. The more abstract the topic of the book, the more difficult it is to come up with relevant photographs. Such images, whether they are photographs or objects d’art can fill the entire page or sit artistically within a frame. In the latter case text may run entirely outside, entirely inside or across the picture. Two examples (Kortmulder, 1998; Kortmulder & Robbers, 2005) are shown in figure 1. *Play and*

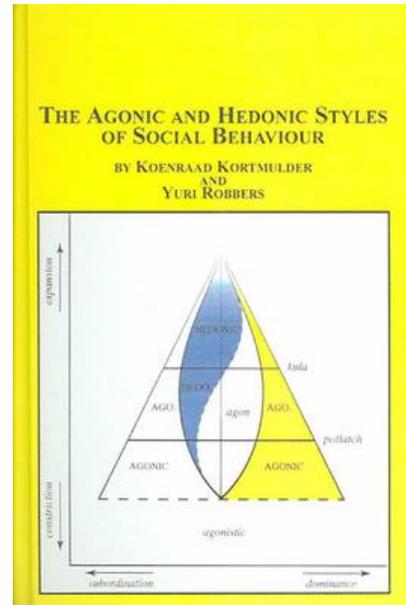
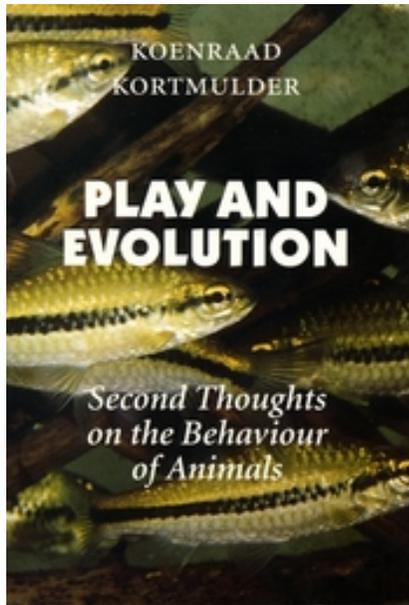


Figure 1: The front covers of two scientific books. The one on the left is far more attractive to the average layperson or an undergraduate student than the one on the right. The one on the right, however, is far more in line with the general look of scientific books, and the cover does seem to match the overly complicated title.

Evolution (Kortmulder, 1998) has an attractive photograph that matches the title, yet manages to intrigue. Scientists as well as lay readers may be tempted to read this book. The cover for *Agonic and Hedonic Styles of Social Behaviour* (Kortmulder & Robbers, 2005), on the other hand, is far more complicated. The graph is a complete summary of the book, but not many people will understand it without reading the book. The complex cover matches the complex title, but a lay reader is unlikely to look at any further than the front cover of this book. Scientists, however may well be tempted to explore this book a bit further though.

4.2 Popular non-fiction

Popular non-fiction really does need a more elaborate cover with good graphics or, at the very least, interesting font effects, or a stylish colour effect in the background. A regular photograph or some artwork might do nicely, but to re-

ally stand out may require some advanced-looking image manipulation. On the bright side, as long as it *looks* advanced, it does not really have to actually *be* advanced. This is an area where one might well want to use dedicated image manipulation software in addition to PSTricks.

4.3 Novels

There are many sub-genres to the Novel: traditional and serious fiction; romances; sci-fi; chick-lit and pulp fiction all have styles of their own. From the William Morris wallpaper with New Gothic font, to garish pink with curly turquoise type, or high-tech space scenes with busty avengers and fierce dragons, you will find the modern paperbacks do follow fashion conventions of their own.

As I mentioned earlier, an important part of defining the essence of your book is to decide the best genre framework for your design. Figure 2 and 3 are two cover designs for the same novel (Skjold, 2006a). Figure 2 suggests serious fiction, and a rather complicated book. Figure 3 gives the impression of a more upbeat and flamboyant paperback, possibly a thriller or good who-dunnit to take on holiday.

Although the image used in figure 2 illustrates the main character's predicament well, the cover design in figure 3 illustrates the title perfectly. It is simple in both colour and composition, and the strange, hybrid passion-magnolia personifies an essential theme of the story, so it scores on all three of my basic design starting points. The vibrant colour makes the book subliminally more inviting, and the image is so closely linked to the title that anyone picking it up can immediately understand it and this creates a positive expectation from the book. This is therefore, the cover that the author chose to use when publishing her mystery-thriller novel.

A good cover should *fit* its book. Creating a cover that does not live up to its promises is guaranteed to leave customers disappointed and have them steer clear of any other books you might want to sell them.

4.4 Poetry

Poetry books often have a very plain and simple cover: usually just the title, author and publisher in an easily palatable font. The reason is that in general

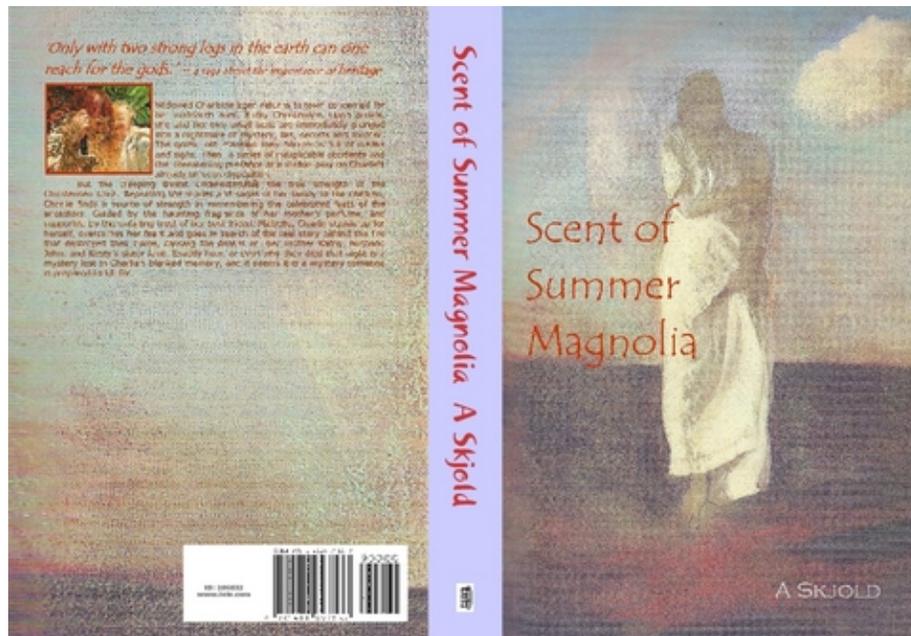


Figure 2: The complex version of the cover for “Scent of Summer Magnolia”.

buyers of poetry are more text-oriented; they are looking for poems that will evoke strong images, and not for strong images that are given to the brain in bite-sized chunks. An understated cover is often best when it comes to poetry.

4.5 Children’s books

Although people often assume it is a myth, research has shown that children really do like bright primary colours when they are young. One could use this on book covers. And all kids, young, older, teenagers and kids of other ages do generally like books with nice drawings on the cover. A children’s book with just text will appeal a lot less — if at all — to the kids as well as to the grandparents who, by far, are the ones who purchase the most children’s books. In addition to attractive illustrations, a book aimed at children should have a font that is easy to read.

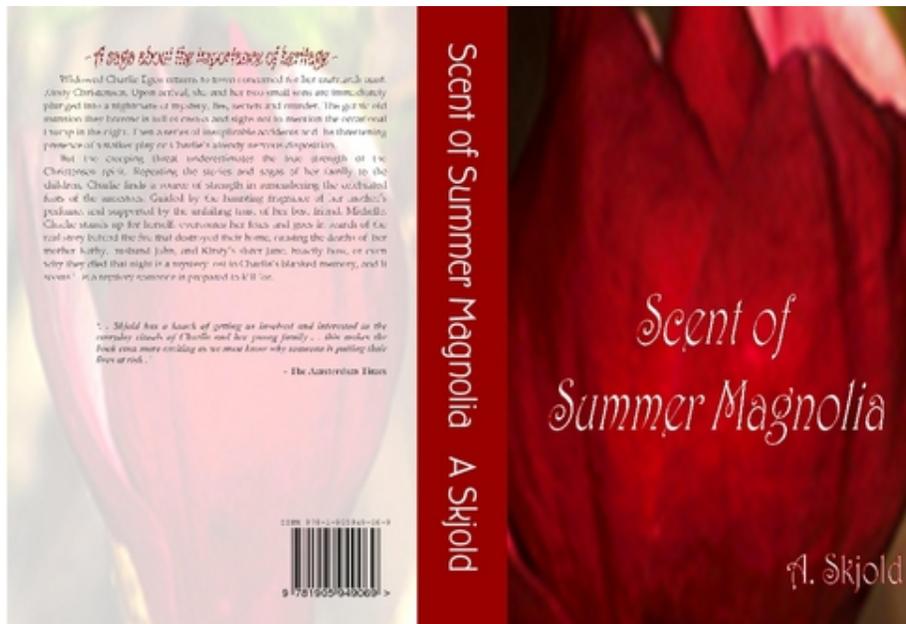


Figure 3: The more upbeat version of the cover for “Scent of Summer Magnolia”.

4.6 Fantasy, Science Fiction and Horror

This is a genre with a highly developed sense of fashion. High quality computer graphics, strong colours, body-beautiful heroes and heroines and strange and wonderful beasts denote the section you are now browsing. Just occasionally an example of abstract art, or coolly classic design can be found covering the more serious league sci-fi writer.

The choice of font here, is extremely important. Fonts generally considered to be less easily readable, such as Gothic scripts, half uncials, or futuristic letter forms often convey exactly the right atmosphere for a book in this genre. Choose your font well.

5 The back of the book

You may think that once you have settled on a design and have legible text on the spine that the matter is finished with, but that is not the case. Once attracted

to your cover, the first thing a buyer does is turn the book over to see what it is about. This is the second line of attraction to reel your reader in. Once again, simplicity and clarity is the goal.

Your secondary header should be concise and provocative, the blurb short and to the point. If you want to put a symbol or photo on the back cover, a small square at top left or right of the blurb is the generally acceptable spot. Unless of course you are a celebrity and your face sells your book, then the back cover is dedicated to you and the blurb goes on the cover flap.

But however you treat the back cover, remember to keep a space at the bottom, usually the bottom right, of the area free for the bar code and publisher's logo.

The back of the book is informative rather than pictorial, so keep the colours and objects to a minimum. If you wish to wrap the front cover illustration around to the back, a good method is to finish it at one third into the back and pick up a soft background colour for the rest. A similar idea using a picture juxtaposed to the one on the front cover is demonstrated in figure 4, a revised edition of [Skjold \(2003\)](#) to be published later this year.

If there have already been reviews from good papers or comments by well known authors, place two or three excerpts on your back cover in a good legible font. I will address this in more detail in the technical section of the paper.

6 Conventional bookstores versus online bookstores

According to many, the advice given above is only important when books are sold in conventional bookshops. People walk in, browse, pick up the occasional book and are either seduced by its cover, title and possibly author to take a closer look, or they are not. The cover needs to be sufficiently alluring to warrant that closer look. Online bookstores are a different beast altogether though: people tend to search by keyword or browse by category. Either way, the first impression they will get of any book will be the title, the author (often in a smaller font already) and a diminutive icon of its cover ([Shepard, 2007](#)). This is all one has at one's disposal to make a book stand out from a whole web page full of similar information about all the other books that match the search criteria or fall into the same category.

Some say the design of the cover is less important as a first impression for online sales of books. What attracts people at first is usually limited to how

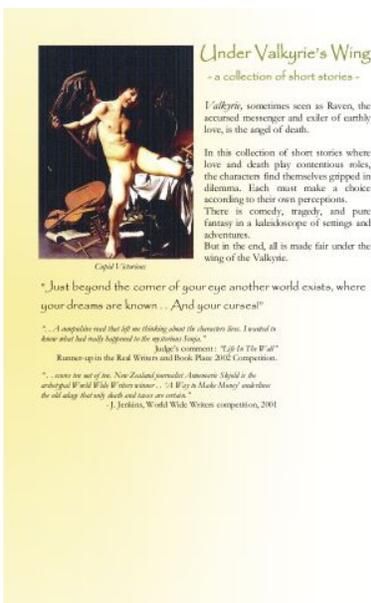


Figure 4: The back cover of *Under Valkyrie's Wing* showing an illustration in juxtaposition to the front cover illustration on the back. Also note the excerpts from favourable reviews and the space at the bottom for the ISBN bar code and the publisher's address.

high up on the page a book appears, its title and author, and perhaps the overall impression of colour the cover conveys. Authors who do not intend to sell books in regular bookstores but focus their attention solely on online stores sometimes even choose a plain monochrome or colour gradient background for their cover with either a very simple caricaturesque drawing or diagram, or — more often — even just the title of the book in as large a font as will fit the cover, and the name of the author squeezed into the remaining empty space, again using a font as large as the title allows. Sans serif fonts seem to be the norm in such cases. One such author is Aaron Shepard, who talks more about designing covers for online sales in his book *Aiming at Amazon* (Shepard, 2007).

Of course when people are attracted by your book in the search result page, they will click on it and get a page with details about the book, including an enlarged picture of the cover. Although many authors, including the aforemen-

tioned Aaron Shepard, just focus on first impressions and design their cover for first impressions only, I am of the opinion that this second impression — with a more detailed cover — is just as important, and this is where a design that is too simple will fail you. Of course I am basing this solely on my own online book buying habits and the impressions of a few friends, so do not take this as gospel. Yet, my own subconscious guideline when browsing books seems to be that a good book deserves a good cover, and when authors obviously do not care enough about their book to create a good cover, chances are they did not care enough to create a good book either.

In some cases it might even be possible or desirable to have different editions for selling books online and selling them in the old fashioned bookstores.

7 Legal matters

Lastly, but certainly not of least importance, there are rights and copyrights to be considered when using artwork on your book cover. When choosing a picture or work of art for use on a book cover, be sure to check that either it is free of rights or that one obtains permission to use it from those who hold the rights. Generally these rights expire 50 years after the death of the artist, but one should always check each and every individual case.

8 T_EXnical Matters: P_STricks

Implementing a good cover requires first of all the use of a style or package that allows the paper size to be adjusted easily, such as — among many others — Memoir ([Wilson, 2004](#)) or geometry ([Umeki, 2000](#)). For a paperback book the page height needs to be equal to the page height as used in the book, while the page width should be twice the page width as used in the book, plus twice the binding correction (*i.e.* the amount of paper lost in the spine of the book in the binding process) plus the width of the spine.

In almost all cases the printer will require extra space around the cover that will be cut away within a certain margin of error. This is called *bleed*. When designing your cover, make sure you allow for an extra 1cm around all edges. It should be coloured in similarly to the rest of your cover, so that if it is not

completely cut away, the final result will still look good. Otherwise, white lines may show up along the trimmed edges of your book. One should, of course, make sure that no crucial parts of the design, such as bits of text or logos, extend into the bleed or even come too close to it, because they may well be cut off. In general, the bleed is exactly 1cm (0.375"), but it is always best to ask your printer for specifics.

Often printers or binders cannot be completely accurate when folding the cover around the book, which will lead to part of the spine showing on the front cover, and part of the back cover on the spine, or the other way round. Even though this is never really more than one eighth of an inch or 3mm, and often less than that, it is still best to make sure the colours do not contrast too much.

In this section I will be considering the example of a popular scientific book on the evolution of sexual reproduction in plants. The book is called *Secrets of the Stamen* and is aimed at a very large audience including laymen, students as well as scientists. This book will be published as a thick, large paperback, 5.5 inches wide, 8 inches high and 1.24 inches thick. My printer has, upon request, informed me that bleed sizes for his equipment are half an inch in each direction. This gives me a cover width of bleed + back + spine + front + bleed, which equals 13.24 inches, and a cover height of bleed + cover + bleed, which equals 9 inches.

Please be aware that by using PSTricks one needs to use the $\text{T}_{\text{E}}\text{X} \rightarrow \text{DVI} \rightarrow \text{POSTSCRIPT} (\rightarrow \text{PDF})$ way. One cannot use PDF $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ directly, unless using the $\text{V}_{\text{T}}\text{E}_{\text{X}}$ distribution.

When converting DVI files into POSTSCRIPT files, please be aware that many programs do not automatically detect that a non-standard page size has been used. For the often-used DVIPS by [Rokicki \(2005\)](#), which comes standard with most $\text{T}_{\text{E}}\text{X}$ distributions, one needs to use the `-T` parameter followed by the width and height of the page. So in case of our example book, one would need to use

```
dvips -T 13.24in,9in -o cover.ps cover.dvi
```

in order to generate the correct POSTSCRIPT file named `cover.ps` from the DVI file `cover.dvi`.

PSTricks can be used with plain $\text{T}_{\text{E}}\text{X}$, $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ ($\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}2_{\epsilon}$) as well as $\text{C}_{\text{O}}\text{N}_{\text{T}}\text{E}_{\text{X}}\text{T}$, but to avoid confusion by trying to cater for all, I will base my examples on $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}2_{\epsilon}$, since that is what I use most often myself. The adaptations needed for plain $\text{T}_{\text{E}}\text{X}$ and $\text{C}_{\text{O}}\text{N}_{\text{T}}\text{E}_{\text{X}}\text{T}$ are minimal, and are explained in the documentation ([Voß, 2007](#)) as

well as on the PSTricks website⁴.

8.1 Setting the Sizes

There are various ways of setting the size of your output. I will demonstrate how to do this using the geometry package (Umeki, 2000). We first set up a file, just with the front cover. Since our book is 5.5×8 inches, we want the width and height of the cover to be set to 5.5 and 8 inches respectively. We use the command

```
\usepackage[paperheight=8in,paperwidth=5.5in,margin=0in]{geometry}
```

to set up the size. We set the margins to zero, so that we have the whole page to play with, and not just part of the page.

Of course what we really need is a complete cover. In order to set that up we will need to simply add twice the bleed to our paper height, and calculate our paper width by doubling the paper width including bleed (once for the front and once for the back, and adding the spine width. The spine width can only be calculated only when you know the number of pages as well as the type of paper your printer uses. In our case that would be $2 \times (5.5 + 0.5) + 1.24 = 13.24$ inches, so

```
\usepackage[paperheight=9in,paperwidth=13.24in,margin=0in]{geometry}
```

will do the trick quite nicely.

8.2 Use of Background Graphics

Before we can use any PSTricks commands, we shall of course need to load the PSTricks package⁵. For including graphics, we will need to load the graphicx package. We also need to suppress page numbers on our cover, which we do with the `\thispagestyle{empty}` command. Our document preamble will now look something like this:

```
\documentclass[12pt]{article}
\usepackage[paperheight=9in,paperwidth=13.24in,margin=0in]{geometry}
```

4. <http://www.pstricks.de>

5. Presumably with some of its add-on packages, but we will get to that in the following paragraphs.

```

\usepackage{pstricks}
\usepackage{graphicx}
\begin{document}
\thispagestyle{empty}

\end{document}

```

In order to easily manipulate our cover, we will now create a single `pspicture` environment covering the entire page:

```

\psset{unit=1in}
\begin{pspicture}(13.24in,9in)

\end{pspicture}

```

This allows us to put anything we want anywhere we want it, using the `PSTricks` `\rput` command

```

\rput{<reference point>}{<angle of rotation>}(x,y){<Object>}
\rput*{<reference point>}{<angle of rotation>}(x,y){<Object>}

```

where the reference point and angle of rotation are optional parameters. The reference point, if specified, allows setting a particular corner of the box we are placing to be put at the coordinates specified. To set the bottom left corner of this box at point $(1,1)$, use `bl` as the reference point, etc. The angle of rotation allows one to rotate the object. The coordinate pair (x,y) determines the position, x to the right and y to the top of the origin, at which the Object will be set. Using the starred version rather than the non-starred version makes the whole box containing the object cover what is underneath.

Please note that even though regular $\text{T}_{\text{E}}\text{X}$ and derivatives use the top left hand corner as the origin, counting positive offsets down and to the right of that origin, `PSTricks` follows the tradition of `POSTSCRIPT` and mathematicians of taking the lower left hand corner to be the origin. Positive coordinates go up and to the right from the origin. This system is officially known as *Cartesian coordinates*⁶.

6. See http://en.wikipedia.org/wiki/Cartesian_coordinates. It is possible to change the coordinate system to, for example, polar coordinates (http://en.wikipedia.org/wiki/Polar_coordinates). This is beyond the scope of this paper, and we refer you to the `PSTricks` documentation (Voß, 2007)

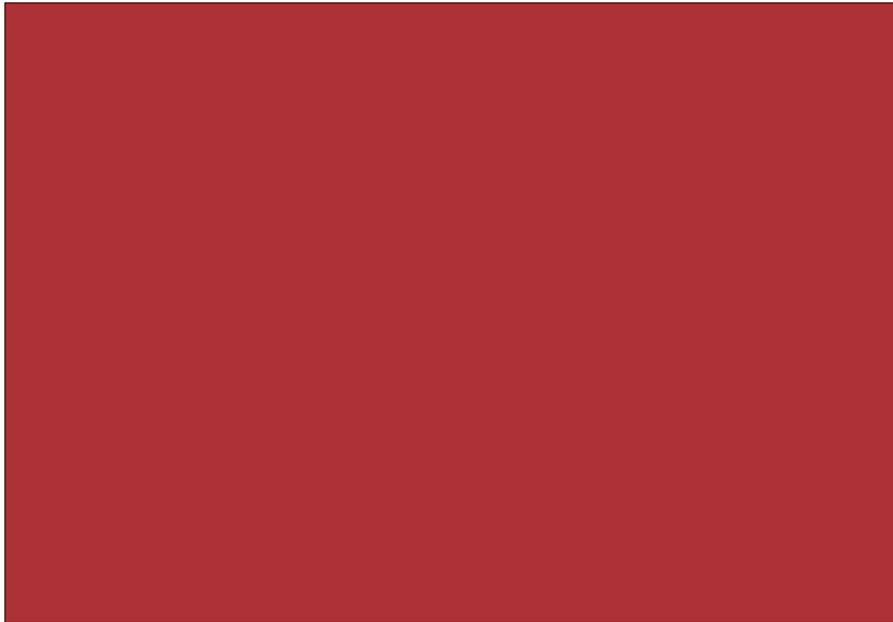


Figure 5: Our cover, at the correct size and entirely maroon.

Often one will want a different background colour than white for the book cover. This can easily be done by creating a rectangle the size of the page in the desired colour. For this we use the `\psframe` command

```
\psframe[<Optional parameters>](x1,y1)(x2,y2)
```

where (x_1, y_1) denotes the bottom left corner, in our example $(0, 0)$, and (x_2, y_2) the upper right corner, in our example $(13.24, 9)$. We use the optional parameters to set a solid fill style, and a fill color of our choice. We could, for example give the command

```
\psframe[fillstyle=solid,fillcolor=Maroon](0,0)(13.24,9)
```

to give the entire background a maroon colour. See figure 5.

It also possible to use a gradient rather than just one colour. We can achieve this with the add-on packages `pst-grad`, `pst-ghsb` and `pst-slpe`, which should be included in the document preamble as follows:

```
\usepackage{pst-grad}
```

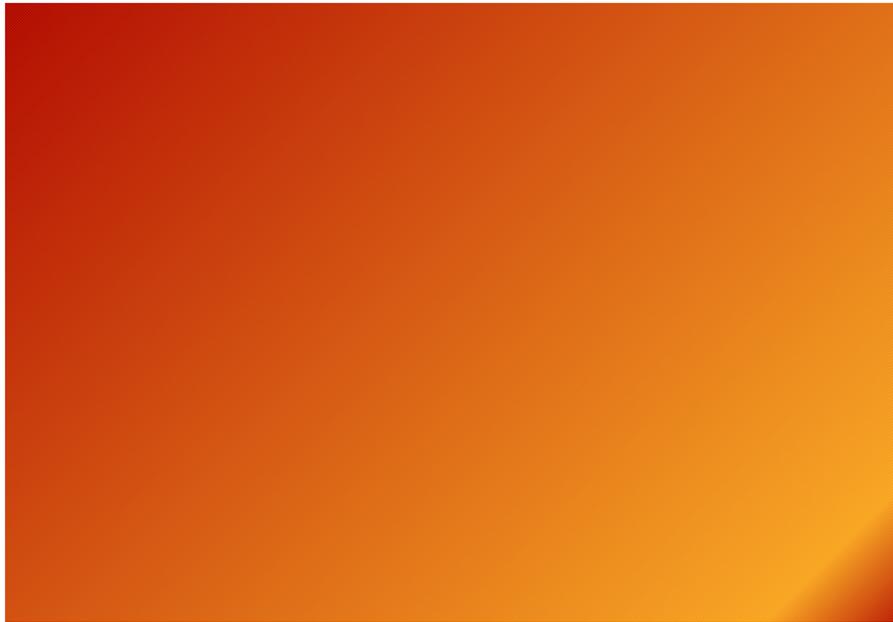


Figure 6: Our cover, at the correct size and coloured with a gradient.

```
\usepackage{pst-ghsb}  
\usepackage{pst-slpe}
```

These packages need to be included *after* the PSTricks package is included. To have more than the eight basic colours available, please include the pstricks package with the dvipsnames prologue and table options enabled. This is done by replacing the `\usepackage` command mentioned above by

```
\usepackage[dvipsnames,prologue,table]{pstricks}
```

A simple example of gradients (using these extra colours) should suffice. Consult the PSTricks-documentation (Voß, 2007) for more details.

```
\psset{fillstyle=gradient,linestyle=none,gradbegin=Maroon,  
gradend=Dandelion,gradangle=45,gradlines=2048}  
\psframe(0,0)(13.24,9)
```

See figure 6.

Now we will include a picture, using the `\includegraphics` command, properly packaged in an `\rput` command in order to get a picture at the correct po-

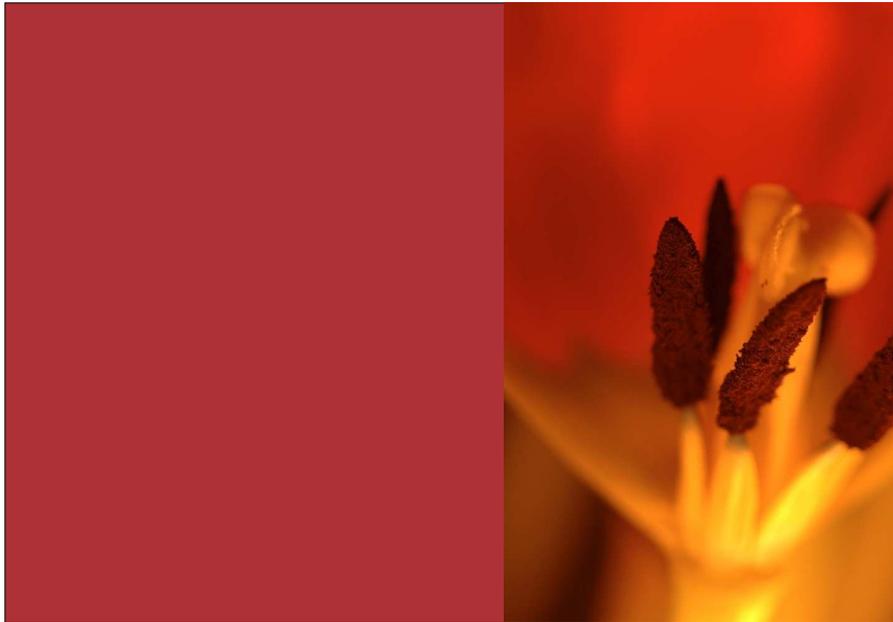


Figure 7: Our cover, with the cover picture included.

sition. Our picture is in the file `flower.eps` and we want to stretch it to cover the entire front cover. The front cover is the right half of our cover, so we need to set our picture 7.24 inches to the right, at the bottom of the page. We need to stretch or shrink the relevant part of our picture to fill the entire front cover including bleed on three sides. This adds up to 6 inches wide and 9 inches high. Unfortunately we need to create a special box that holds our picture for things to work out properly. We can do this before we even give our `\pspicture` command

```
\newsavebox\IBox  
\sbox\IBox{\includegraphics[height=9in]{flower.eps}}
```

Now we can `\rput` our picture where we want it with a command like

```
\rput [1b] (7.24,0){\usebox\IBox}
```

See figure 7.

Please note that stretching or shrinking an image this way can lead to distortions when height and width are not stretched equally. These distortions may or may not be desirable. It is often best to play around a bit with the stretching and

clipping options that either the `graphicx` package or your photo editing software provide to see what works best.

8.3 Text Effects

Almost every book cover has text on it⁷. We can easily put text on our cover wherever we like using the `\rput` command, which has been introduced before. The command

```
\rput(8.24,7){\sffamily\bfseries\Huge  
  \color{white}{Secrets of the Stamen}}
```

would put the title of this hypothetical book, “Secrets of the Stamen” with huge white boldfaced sans serif letters on the front cover, 7 inches from the bottom and 1 inch off of the left hand side. See figure 8.

This is nice of course, but often one wants something a bit more than this. A different font... a different size... text along a curved line... In other words, one wants text effects. PSTricks offers two add-on packages called `pst-text` and `pst-char` that allow one to easily create such effects.

First of all the font needs to be set. It is possible to use the standard $\text{\LaTeX}2_{\epsilon}$ PSNFSS commands (Schmidt, 2004) to select fonts, but PSTricks offers easier manipulation of sizes when using its own `\DeclareFixedFont` command. This command does still use the PSNFSS names of fonts. To create, for example, a title font with the name `\PT` that makes available the italic Palatino font, boldfaced, scaled to 0.5 inch, we use:

```
\DeclareFixedFont{\PT}{T1}{ppl}{b}{it}{0.5in}
```

where `\PT` is the font command we create, `T1` is the encoding, `ppl` the PSNFSS font name for Palatino, `b` the code for the bold weight, `it` the code for the italics series, and `0.5in` the height of the font. We can apply this newly declared font command by changing the code for our title to:

```
\rput(8.24,7){\PT \color{white}{Secrets of the Stamen}}
```

See figure 9.

7. The only one I know that does not is a 19th century volume of poetry by Aleister Crowley called “White Stains”. On the front of its black cover it has nothing more than a few white stains, and the back as well as the spine of the cover are completely blank. Even with this book, however, the dust jacket that folds around the cover does have text on it.



Figure 8: Our cover, with the title in place.

It is possible to have the text follow a set path rather than the normal straight line. This is done with the `\pstextpath` command. To use text and character paths please include the relevant packages.

```
\usepackage{pst-text}  
\usepackage{pst-char}
```

Basically any shape can be used as the path to set text along. I will give a simple example here, setting our title along a circle

```
\psset{linestyle=none}  
\rput[c](10.24,7){\pstextpath[r](0,0){\pscircle{2}}%  
  {\PT \color{white}{Secrets of the Stamen}}}
```

The command `\psset{linestyle=none}` makes sure that the circle we use as a text path is not actually drawn. The `\rput` command sets the text path on the page where we want it to be. The `\pstextpath` command creates the actual text path, where the optional command `[r]` sets the text flush left (l), centered (c) or flush right (r) on the path, the coordinate pair `(0,0)` determine the offset of



Figure 9: Our cover, with Palatino bold italic as the title font.

characters off of the defined path, the `\pscircle{2}` creates the text path itself, in our example a circle with a 2 inch radius, and in the final pair of braces we find the actual text that is set, our title set in the predefined Palatino font, in white. See figure 10.

Fancy effects can also be achieved by outlining the letters with a very thin yellow line and filling them with a gradient, for example. Again, a simple example should suffice.

```
\psset{linecolor=yellow,fillstyle=gradient,%
      gradbegin=white,gradend=Dandelion}
\rput[1b](7.74,7){\pscharpath{\Chancery Secrets}}
\rput[1b](8.24,5){\pscharpath{\Chancery of the}}
\rput[1b](7.74,4){\pscharpath{\Chancery Stamen}}
```

See figure 11.

Of course, when everything has been decided and properly placed, it is time to put the name of the author in. Then, of course, on most books one would want

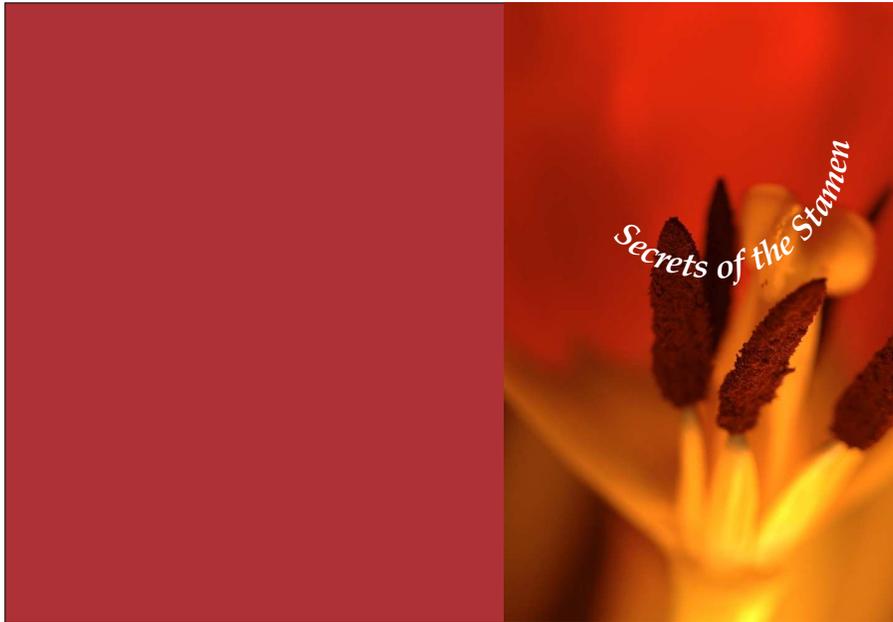


Figure 10: Our cover, with the text typeset along a circular path. Personally I do not think it works for this particular cover, but with a different design something like this can work wonders.

to include the name and/or the logo of the publisher somewhere on the front cover.

When that is done, we shift our attention to the spine. Usually it is best to just put the name of the author and the title down the spine, with the logo of the publisher. Interestingly, most spines are printed so that they can be properly read when the book is lying down, with the front cover facing up. German books, however, tend to be printed the other way round. Their spine can be properly read when the book is lying down with the back cover facing up. I have no idea why this is, but I have only come across very few exceptions in my life as a bibliophile.

Then of course text, called the blurb, can be added onto the back of the book. This text could be a short summary of the book, a titillating excerpt or quotes from favourable reviews. There are other options of course, but these tend to be the main contenders. It is usually good to include at least a short summary



Figure 11: Our cover, with the title in 2 inch Zapf Chancery with a white to dandelion gradient, and a yellow hairline around it.

or excerpt of the book, and a few lines of biographical text about the author. A picture of the author is also a good idea. Of course the author biography and picture can also be inside the book.

An excerpt from a favourable review, or even a quote from an enthusiastic reader, can work wonders for sales. If prospective buyers see quotation marks, they tend to take a closer look. If the comments are from an independent observer and are favourable, this is a strong incentive to take a closer look inside the book.

We will add some text to our sample cover, which shall be nonsensical text, since there is no actual book. For generating this nonsensical text, I use the lipsum package (Happel, 2005). Since a lot of additions have been made by now, I will give the complete code, as it now stands:

```
\documentclass[12pt]{article}

% load the necessary packages
```

```

\usepackage[paperheight=9in,paperwidth=13.24in,margin=0in]{geometry}
\usepackage[dvipsnames,prologue,table]{pstricks}
\usepackage{pst-text}
\usepackage{pst-char}
\usepackage{pst-grad}
\usepackage{graphicx}
\usepackage{lipsum}

% begin the document and suppress page numbers

\begin{document}
\pagestyle{empty}

% create the box with the front cover picture

\newsavebox\IBox
\sbox\IBox{\includegraphics[height=9in]{flower.eps}}

% set up the picture environment

\psset{unit=1in}
\begin{pspicture}(13.24in,9in)

% set up the fonts we use

\DeclareFixedFont{\PT}{T1}{ppl}{b}{it}{0.5in}
\DeclareFixedFont{\PTsmall}{T1}{ppl}{b}{it}{0.4in}
\DeclareFixedFont{\PTsmallest}{T1}{ppl}{b}{it}{0.3in}
\DeclareFixedFont{\PTtext}{T1}{ppl}{b}{it}{11pt}
\DeclareFixedFont{\Logo}{T1}{pbk}{m}{n}{0.3in}

% create a maroon background

\psframe[fillstyle=solid,fillcolor=Maroon](0,0)(13.24,9)

```

```

% place the front cover picture

\rput[lb](7.24,0){\usebox\IBox}

% put the text on the front cover

\rput[lb](7.74,7){\PT \color{white}{Secrets of the Stamen}}
\rput[lb](8.94,6){\PTsmall \color{white}{Yuri Robbers}}
\rput[lb](9.04,0.8){\PTsmallest \color{white}{Lughdunum Press}}

% put the text on the spine (note the rotation over 270 degrees)

\rput[l]{270}(6.62,8.5){\PTsmall%
  \color{white}{Yuri Robbers --- Secrets of the Stamen}}

% put the publisher's logo on the spine

\rput[b](6.62,0.75){\color{white}{\fbox{\Logo L}}}

% Create a Box containing the text for the back cover

\newsavebox\Blurbbox
\sbbox\Blurbbox{\begin{minipage}{4.5in}
\textcolor{white}{\lipsum[1]}
\end{minipage}}

% And position the box

\rput[tl](1,8){\usebox\Blurbbox}

% Then we close all open environments

\end{pspicture}
\end{document}

```

And the result so far can be seen in figure 12.

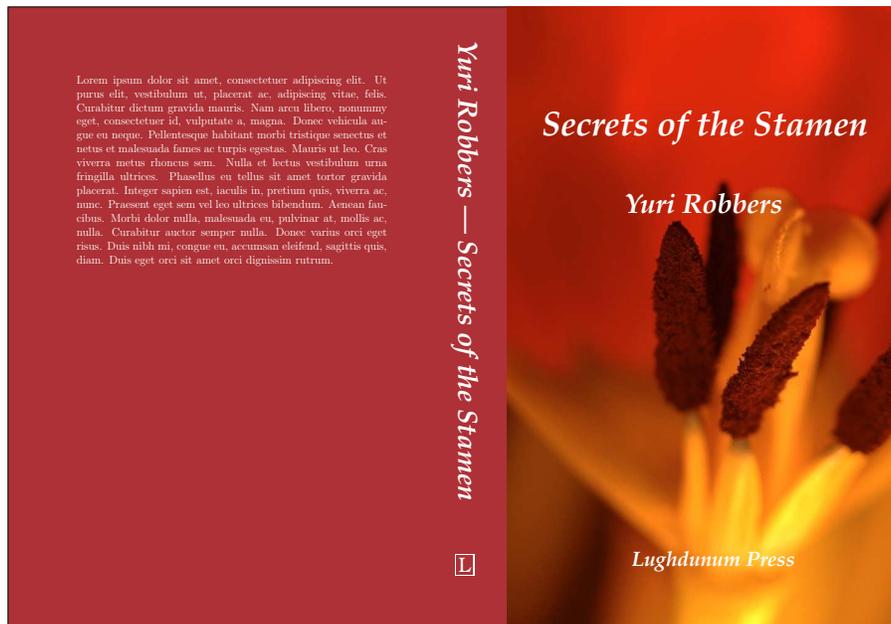


Figure 12: Our cover, with almost all text in place, including the spine and the blurb on the back cover.

8.4 Framing Graphics

It is of course also possible to use graphics without having them cover the entire page. In that case one can place them anywhere on the cover, or it might be nice to put a frame around them. The front cover of two books like that, [Skjold \(2003, 2006b\)](#), can be seen in figure 13. *Beneath Valkyrie's Wing* has an irregularly shaped figure, placed on a white cover so that it looks best. Part of the angelic figure, taken from Giovanni Baglioni's *Heavenly and Earthly Love*, cannot even be seen, but this just serves to improve the dramatic impact of the cover as a whole, exactly as it did in Baglioni's painting. *To Move a Mountain* has two included graphics. First of all there is a big one covering most of the page. The title (set in the *Trebuchet* typeface) stands across the image. Then there is the symbol of the horse: this is rotated, and has a thicker border. The cover would also work quite well without this badge: the whole design would be "cleaner" for those who would prefer that.

In order to show how to frame pictures, we will include a picture of the au-

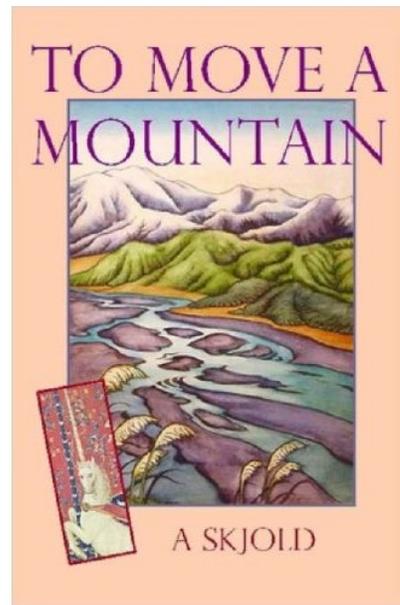
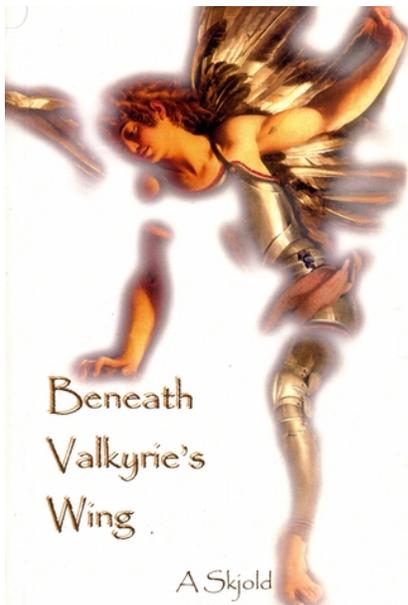


Figure 13: The front cover of Annemarie Skjold’s *Beneath Valkyrie’s Wing* and *To Move a Mountain*. On *Beneath Valkyrie’s Wing*, the picture is irregularly shaped, and a bit larger than the page. It is positioned where it looks best, and the remainder is cut off. The picture for *To Move a Mountain* does not cover the entire front page, and the text flows over it. The badge with the symbol is much smaller, rotated and has a thicker frame.

thor on our example book “Secrets of the Stamen”, along with a short biography (courtesy of the `PracTEX` website). We will frame this picture in a reasonably thick white frame, and a very thin black line.

```
% create a savebox for the picture

\sbox\Authorbox{\includegraphics[width=0.5in]{yuri}}

% create a solid white rectangle of the correct size at the correct
% spot by default this is framed with a thin black line. Use
% linestyle=None to avoid this
```

```

\psframe[fillstyle=solid,fillcolor=white](1,4)(1.7,4.95)

% now place the picture

\rput[lb](1.1,4.1){\usebox\Authorbox}

% create a savebx for the biography. The width has been adjusted so
% that the right margin matches with that of the book blurb

\newsavebox\Biobox
\sbox\Biobox{\begin{minipage}{3.6in}
\textcolor{white}{Yuri Robbers holds a degree in Animal Behaviour
and is a teacher, a researcher and a published author. He's always
had a keen interest in typography, possibly because his father is a
professional typographer. Ever since he discovered \LaTeX{} in 1995,
he has always done his best to avoid using word processors, and he
has embarked on a quest to learn as much as possible about \TeX{}
and its derivatives and to apply this knowledge whenever possible.
Contact him at yuri.robbers@gmail.com}
\end{minipage}}

% and put it where it belongs

\rput[tl](1.9,4.95){\usebox\Biobox}

```

And see the result in figure 14.

8.5 The Finishing Touches: the ISBN and the publisher's address

Now all that needs to be done is adding the publisher's address and the bar code with the ISBN number on the back of the book. Luckily, PSTricks also has a standard add-on package for creating bar codes, `pst-barcode`, which can also create bar codes for ISBN numbers. Include it with a `\usepackage` command.

If you have a publisher, only your publisher can take care of obtaining an ISBN number, Of course if you self-publish, you need to get your own ISBN number for your book. There are national authorities in charge of administering

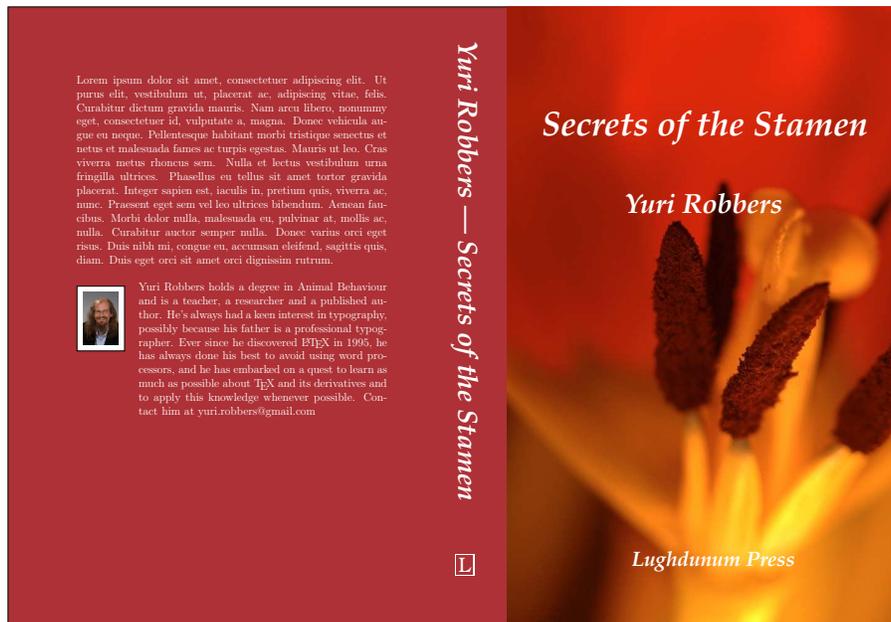


Figure 14: Our cover with the picture of the author and the author’s biography in place.

and purveying ISBN numbers, and generally one can find this authority easily by typing “ISBN” and one’s country name into the Google search engine⁸. Prices differ markedly from one country to another. An eighty-fold price difference is no exception! If you have co-authors residing in other countries, it may therefore be worthwhile to compare the prices for each author’s country of residence.

Since the ISBN bar code needs to be machine-readable, it is usually a good idea to use a solid white background. If the background colour is not too dark, and there are no strong contrasts, you may prefer to leave off the white background for aesthetic reasons, as has been done in *Scent of Summer Magnolia* (Skjold, 2006a), as can be seen in figure 3.

Since our example book has a maroon back cover, this will most certainly not be an option: we need to create a white rectangle for it. We do this the same way we did with the author picture.

Sometimes one also sees the price of the book printed next to the ISBN bar

8. <http://www.google.com>

code, sometimes even prices in various currencies for various countries. When you have a publisher, the publisher will usually have a standard solution for this. When self publishing, I advise against putting a price on, since it is usually a highly costly procedure to have your printer switch from using one PDF file for your cover to using your new updated PDF cover. This almost always requires paying a new setup fee for your entire book. It is a miracle to me why this should be necessary, but unfortunately it is how things usually work.

Back to the technical matters. All we need to do is include a nice white rectangle, and put the bar code on top. The bottom of the back cover is the customary location for the bar code, and the right hand corner or the centre seem to be more common than the left.

Here is the code for the bar code

```
\psframe[fillstyle=solid,fillcolor=white](3.9,0.8)(5.8,2.2)
\rput(4.2,1){\psbarcode{1-58880-149}{includetext guardwhitespace}%
  {isbn}}
```

and the code for the publisher's address

```
\rput[lb](1,1.2){\PTsmallest \color{white}{Lughdunum Press}}
\rput[lb](1,1){\PTtext \color{white}{Leiden --- The Netherlands}}
\rput[lb](1,0.8){\PTtext \color{white}{http://www.lughdunumpress.nl}}
```

And the final result can be seen in figure 15.

9 Concluding remarks

As you can see, creating a good book cover using PSTricks is not at all difficult. Almost everything one might want to do has been thought of already, and commands or packages exist. Doing something new may involve some browsing through the documentation, but both the published documentation (Voß, 2007) and the online documentation at <http://www.pstricks.de> are excellent.

Should one, for example, wish to create a graph as is used on the front cover of *Agonic and Hedonic Styles of Social Behaviour* (Kortmulder & Robbers, 2005), as is displayed on the right hand side in figure 1 then this can easily be done using PSTricks as well.

Experiment, play around, make mistakes and fix them, and soon you will be making excellent book covers (or other graphics) using PSTricks!

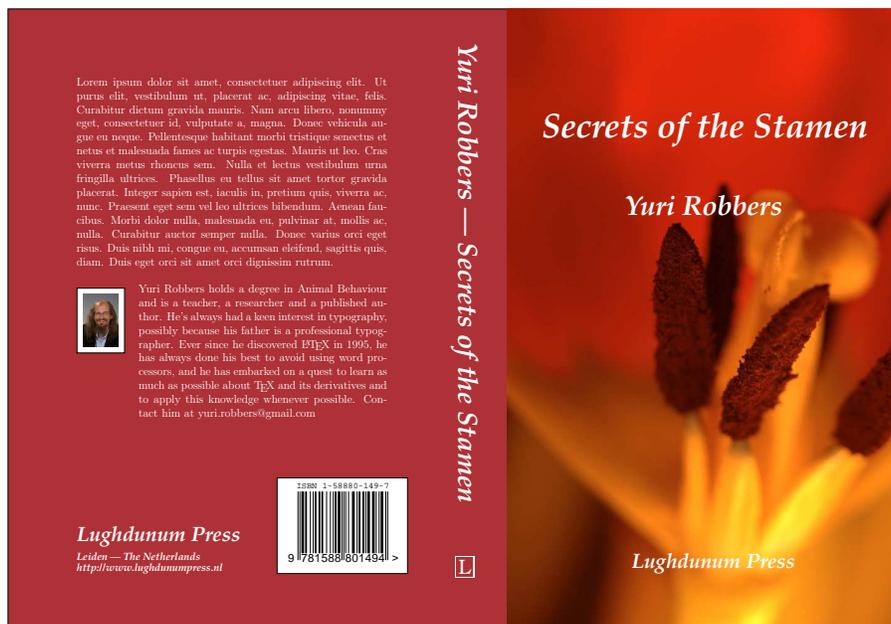


Figure 15: Our cover, the final version including the finishing touches.

10 Acknowledgments

We thank Paul Blaga and the anonymous referees for their helpful comments. Staff members of several local bookshops have freely shared their experiences regarding the role of book covers in sales. Paul Blaga’s and Lance Carnes’s suggestions have helped us reduce the size of this PDF from 64 to 1.1 Megabytes.

References

- HAPPEL, P. (2005). *lipsum* — access to 150 paragraphs of *lorem ipsum* dummy text. CTAN://macros/latex/contrib/lipsum/.
- KORTMULDER, K. (1998). *Play and Evolution. Second Thoughts on the Behaviour of Animals*. Utrecht, the Netherlands: International Books.
- KORTMULDER, K. & ROBBERS, Y. (2005). *Agonic and Hedonic Styles of Social Behaviour*. Lampeter, Wales, UK: The Edwin Mellen Press.

- ROKICKI, T. (2005). Dvips: A DVI-to-PostScript translator. <http://www.radicaleye.com/dvips>.
- SCHMIDT, W. (2004). PSNFSS2e: Using common postscript fonts with L^AT_EX2_ε. CTAN://macros/latex/required/psnfss.
- SHEPARD, A. (2007). *Aiming at Amazon*. Olympia, WA, USA: Shepard Press.
- SKJOLD, A. (2003). *Beneath Valkyrie's Wing*. West Yorkshire, UK: central Publishing.
- SKJOLD, A. (2006a). *Scent of Summer Magnolia*. Eastbourne, UK: Ivory Moon.
- SKJOLD, A. (2006b). *To Move a Mountain*. Spain: Lulu.
- UMEKI, H. (2000). The *geometry* package. CTAN://macros/latex/contrib/geometry/.
- Voss, H. (2007). *PSTricks. Grafik mit PostScript für T_EX und L^AT_EX, 4th Edition*. Heidelberg: Dante.
- WILSON, P. (2004). The memoir class for configurable typesetting. CTAN://macros/latex/contrib/memoir/.

[Journal](#)
[home page](#)
[General](#)
[information](#)
[Submit an](#)
[item](#)
[Download](#)
[style files](#)
[Copyright](#)
[Contact us](#)

THE PracTeX Journal



Commutative Diagrams with XY-pic II. Frames and matrices

Paul Blaga

Abstract

This is the second article dedicated, essentially, to the use of Xy-pic for constructing commutative diagrams. By using the same kind of approach as in the first part, we focus, now, on frames, matrices and other extensions of the kernel language.

Paul Blaga is an Assistant Professor of Geometry at the Faculty of Mathematics and Computer Sciences of the "Babes-Bolyai" University in Cluj-Napoca, Romania. He started using LaTeX in the early 1990's. Initially, his interest was just for writing scientific papers, but afterwards he became more and more involved and gradually turned into a great LaTeX fan.

He has served several years as an editor of the mathematical journal *Studia Universitatis Napocensis, Mathematica* and was responsible for switching it to LaTeX in the late 1990's. He is also the co-author of a book on LaTeX.

His main interests in LaTeX are related, especially, to the graphical capabilities -- not only commutative diagrams, but also more complicated drawings, made, for instance, with PSTricks.

You can reach Paul at pablaga@cs.ubbcluj.ro

- [PDF version of paper](#)
- [Comment on this paper](#)
- [Send submission idea to editor](#)

Commutative Diagrams with X_Y-pic II. Frames and Matrices

Paul A. Blaga

Email pablaga@cs.ubbcluj.ro
Website <http://www.cs.ubbcluj.ro/~pablaga>
Address “Babeş-Bolyai” University of Cluj Napoca,
Faculty of Mathematics and Computer Science
1, Kogălniceanu Street,
400609 Cluj Napoca,
Romania

Abstract This is the second article dedicated, essentially, to the use of X_Y-pic for constructing commutative diagrams. By using the same kind of approach as in the first part, we focus, now, on frames, matrices and other extensions of the kernel language.

1 Introduction

We continue (and end!) here our incursion into the realm of X_Y-pic ([7]) we begun in [1]. Our exposition, in this part of the paper, continues to rest, especially, on [8], [9] (hereafter cited as “Reference Manual”), [5] and [4]. The reader wanting to make an impression about the possibilities of other packages for producing commutative diagrams, can have a look at the very nice survey of Valiente-Feruglio [10]. I would like to express my gratitude to Lance Carnes for support and for pressing me to finish the work.

2 Frames

2.1 Introduction

One of the nice features of X_Y-pic is being able to put *frames* around objects. To do this you use the option *frames* when you load the *xy* package. Before going

directly to the description of the various framing commands, we shall return for a while to *objects*, to better understand their basic characteristics.

2.2 Objects, again!

An *object* in \mathcal{X}_Y -pic is like a T_EX box, with the essential difference that it may have several *shapes*. The kernel of \mathcal{X}_Y -pic provides three basic shapes:

- point;
- rectangular;
- elliptic.

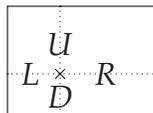
The default shape of an object is rectangular. This means that if we drop an object at some position, by using the $\ast\{\}$ command, the respective object is included in a rectangular box. For instance, a command like $(0,0)\ast\{A\otimes B\}$ will produce:

$$\boxed{A \otimes B}$$

(the frame will not actually be produced, and we put it to emphasize the edges of the rectangular box of the object; we shall see below how to produce the frame). To ensure that the object has a rectangular shape, we should use, in principle, the option $[\square]$: $(0,0)\ast[\square]\{A\otimes B\}$. As this is the default shape, we don't have to use it, actually.

To get an object which has the shape of a point, we have to use the option $[.]$. Finally, to get the circular shape, we have to use the option $[o]$ (the letter small *o*, not zero). We shall see more details in a moment.

To each object of rectangular shape there are associated four numbers, L, R, U, D , identifying the position of the *reference point* (the "center") of the encapsulating box:

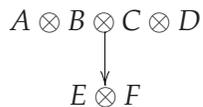


By default, the two horizontal and the two vertical dimensions are equal and the reference point is actually the center of the smallest rectangular box containing the object. We notice that an arrow between two objects has as a support line the line connecting the reference points of the two objects. Therefore, if we want

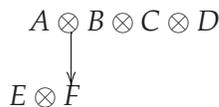
to change slightly the direction of the arrow it is enough to move the reference points. If we change the shape of an object from rectangular to elliptic, the edge of the object will be an ellipse of axes $(L + R)/2$ and $(U + D)/2$, respectively. Again, by default, the reference point of an elliptic object is at the center of the ellipse.

It is important, when constructing a diagram, to be able to move the reference point of the object, because, in this way, we can modify the direction of the arrow.

This is possible by using the modifier `!<vector>`, where `<vector>` can be specified as a pair of TeX lengths. The modifier simply translates the reference point of the object with the corresponding vector. Compare the following examples:



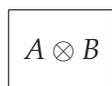
```
\begin{xy}
(0,20)*{A\otimes B\otimes C\otimes D}
(0,10)*{E\otimes F}
{\ar "a";"b"}
\end{xy}
```



```
\begin{xy}
(0,20)*!<-20pt,0pt>{A\otimes B\otimes C\otimes D}
(0,10)*!<10pt,0pt>{E\otimes F}
{\ar "a";"b"}
\end{xy}
```

Other important modifiers that can be added to an object are related to the *dimension* and *color*. To modify the color, the `color` option has to be loaded and you have to use a driver that supports the use of colors (such as dvips).

The size modifier actually modifies the dimensions of the bounding box of the object. It is possible either to set the size of the object, with the modifier `=<size>`, where the size is a pair of *positive length* which are the two sides of the rectangle that contains the object:



```
\begin{xy}
(0,0)*=<40pt,30pt>[F]{A\otimes B}
\end{xy}
```

Here [F] creates the frame around the object. The details will be given bellow.
 The color modifier has as effect the coloring of the contents of the object in a prescribed color:

$A \otimes B$

```
\begin{xy}
(0,0)*[magenta]{A\otimes B}.
\end{xy}
```

The usual colors (red, blue, black, green, yellow, cyan, magenta,...) are defined by the program, but there is the possibility to define more colors, as well(see the Reference Manual).

2.3 Adding frames

The *frames* are Xy-pic objects themselves and they are produced by a command of the form

`\frm{type}`

The frames can be added to the picture both by using the drop operator `*{...}` and the connecting operator `**{...}`. When we use the drop operator, we obtain a frame around the current object (the last one dropped):

$A \otimes B$

```
\begin{xy}
(0,0)**{A\otimes B}*\frm{-}
\end{xy}
```

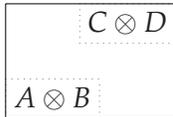
We shouldn't add a semicolon between the current object and the framing command, otherwise the object wouldn't be framed. If, instead, we use the connecting operator, we obtain a frame around the last two added objects:

$A \otimes B$
 $C \otimes D$

```
\begin{xy}
(0,0)**{A\otimes B};%
(10,10)**{C\otimes D} **\frm{-}
\end{xy}
```

Please notice that we have to add a semicolon between the two objects added (this operator actually turns the last added object into the current object).

We may even construct something like:



```
\begin{xy}
(0,0)*+{A\otimes B}*\frm{.};%
(10,10)*+{C\otimes D}*\frm{.}%
**\frm{-}
\end{xy}
```

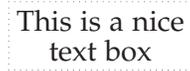
Although they are produced by the same kind of commands, the frames are actually of several kinds, that will be discussed separately:

- proper frames (they really produce some frames around objects);
- brackets – they provide some stretchable vertical and horizontal brackets;
- filled regions – instead of producing frames, they fill the bounding box of the objects with a colour.

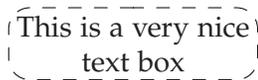
Let us, now, describe, first, the proper frames.

- (i) The dummy frame, produced by `\frm{}` doesn't actually produce any frame at all. It is only provided that because some of the \Xy-pic take a frame as an argument and sometime we don't want any frame to be produced.
- (ii) The *rectangular* frames are produced by one the framing commands: `\frm{.}`, `\frm{-}`, `\frm{--}`, `\frm{=}`, `\frm{==}` and `\frm{o-}`. All these six commands produce rectangular frames around an object. The object is supposed to be rectangular and the frame is, essentially, nothing but an outline of the edge of the object. The arguments of the framing command indicate, for the first three types of frames, what kind of line it is used to draw the frame: a dotted line, a solid line, a double solid line, a dashed line, a double dashed line, respectively. The last framing command produce a dashed rectangular frame with rounded corners, the radius of the corners being a constant, equal to $5pt$. This radius is actually, equal to the size of the dashes and can be modified by modifying this size, using the command `\xydashfont`.

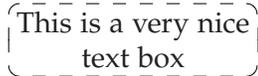
The first three framing commands from this category admit an optional argument *corner radius* (a \TeX length), rounding the corners with quarter circles of the prescribed radius. Here are some examples:



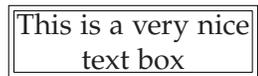
```
\begin{xy}
*+{\txt{This is a nice \\%
text box}}*\frm{.}
\end{xy}
```



```
\begin{xy}
*+{\txt{This is a very nice \\%
text box}}*\frm<8pt>{--}
\end{xy}
```

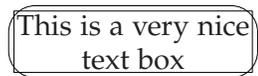


```
\begin{xy}
*+{\txt{This is a very nice \\%
text box}}*\frm{o-}
\end{xy}
```



```
\begin{xy}
*+{\txt{This is a very nice \\%
text box}}*\frm{=}
\end{xy}
```

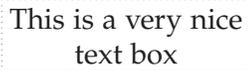
The authors of \Xy -pic claim that the double frames can also be rounded. Anyway, although we don't get any error, the commands don't seem to produce the expected result:



```
\begin{xy}
*+{\txt{This is a very nice \\%
text box}}*\frm<10pt>{=}
\end{xy}
```

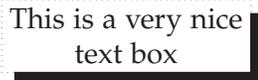
As you can see, only the *outer* frame is actually rounded.

- (iii) An alternative rectangular frame is produced by the command `\frm{,}`. This command puts a shade beneath the rectangular object, creating the illusion of three dimensions:



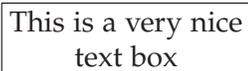
```
\begin{xy}
*+{\txt{This is a very nice \\%
text box}}*\frm{,}*\frm{.}
\end{xy}
```

As you can see, this command actually adds a frame only on two sides of the objects (down and left). We added a second framing command to indicate the actual limits of the objects. This command also accepts an optional length argument. This time the optional argument specifies the depth of the shadow:



```
\begin{xy}
*+{\txt{This is a very nice \\%
text box}}*\frm<5pt>{,}*\frm{.}
\end{xy}
```

\Xypic also provides a combined framing command, `\frm{-,}`, which produces the shade, but also puts frames all around the object:



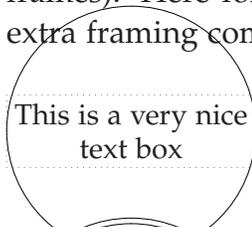
```
\begin{xy}
*+{\txt{This is a very nice \\%
text box}}*\frm{-,}
\end{xy}
```

Unfortunately, with this command you have no control on the depth of the shade. You can put the optional argument, if you like, but it has no effect. All is not lost, however, because, if you want a deeper shade, you can use, in conjunction, the commands `\frm{,}` and `\frm{-}`:

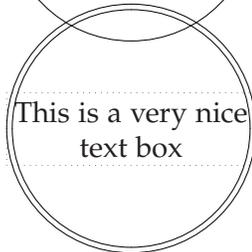
This is a very nice
text box

```
\begin{xy}
*+{\txt{This is a very nice \\%
text box}}*\frm<5pt>{,}*\frm{-}
\end{xy}
```

- (iv) We can produce circular frames using the commands `\frm{o}` (simple frames) and `\frm{oo}` (double frames). If the current object has the horizontal dimensions R and L , respectively, then the radius of the circle will be $(R + L)/2$, (in the case of simple frames; for double frames, this is the radius of the outer circle). Both commands take an optional length argument, specifying the radius of the circle (or outer circle, in the case of double frames). Here follows some examples where, again, we outlined, with an extra framing command, the edges of the document:

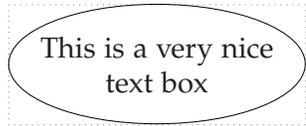


```
\begin{xy}
*+{\txt{This is a very nice \\%
text box}}*\frm{o}*\frm{.}
\end{xy}
```

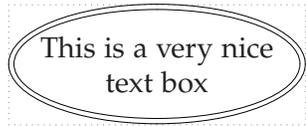


```
\begin{xy}
*+{\txt{This is a very nice \\%
text box}}*\frm{.}*\frm{oo}
\end{xy}
```

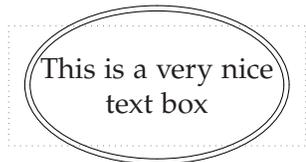
- (v) We can have, finally, *elliptic* frames, simple or double. They are produced with the commands `\frm{e}` (the simple frames) and `\frm{ee}` (the double frames). For a rectangular object of dimensions R, L, U, V , the axis of the ellipse are $(R + L)/2$ and $(U + D)/2$, respectively (for a double frame, these dimensions refer to the outer ellipse). The two commands admit an optional argument, a pair of lengths, allowing you to prescribe the dimensions of the two axes:



```
\begin{xy}
*++++{\txt{This is a very nice \\%
text box}}*\frm{.}*\frm{e}
\end{xy}
```

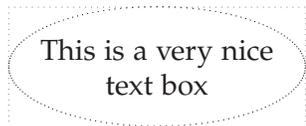


```
\begin{xy}
*++++{\txt{This is a very nice \\%
text box}}*\frm{.}*\frm{ee}
\end{xy}
```



```
\begin{xy}
*++++{\txt{This is a very nice \\%
text box}}*\frm{.}%
*\frm<50pt,30pt>{ee}
\end{xy}
```

The commands `\frm{o}` and `\frm{e}` have also the variants `\frm{.o}`, `\frm{-o}` and `\frm{.e}`, `\frm{-e}`, respectively, producing dotted or dashed frames:



```
\begin{xy}
*++++{\txt{This is a very nice \\%
text box}}*\frm{.}*\frm{.e}
\end{xy}
```

These variants don't work for double frames.

A final word should be added regarding curved (circular or elliptic) frames. They work properly only in one of the following two circumstances:

- (i) We use the Postscript driver `dvips` (meaning we use the `Xy-pic` option `dvips`). Beware that this means that the circles and ellipses are produced by Postscript `\special` commands which are not recognized by all the visualisation and printing drivers. In particular, we can't use `pdflatex` to produce directly a pdf file.

- (ii) We use the option `curve` of `Xy-pic`. However, this is not enough, we also have to use the command

```
\UseCurveFrames
```

before we use the commands for producing curved frames. The command respects the `TEX` grouping and can be annihilated by the command

```
\UseFontFrames
```

which restores the usual frames. You should be aware, also, of two facts:

1. in the Reference Manual the command is misspelled as
2. there is, also, another misspelling, this time in the input file `xy-frame.tex`, where, on the line 521, the authors typed

```
\UseCurvedframes@
```

instead of

```
\UseCurvedFrames@.
```

You should correct this before attempting to use curved frames.

If none of the two circumstances is met, the commands for curved frames will produce just circles (even if we expect to get ellipses) and, even in the case of circular frames, the radius is usually not the one we would expect it to be.

The second kind of frames are the *brackets*. They are, essentially, `TEX` braces or (round) parentheses, scaled and arranged in a certain way. They are produced with one of the following eight framing commands:

```
\frm{_\}, \frm{^_\}, \frm{\{\}, \frm{\}\},  
\frm{_\}, \frm{^}\}, \frm{(\}, \frm{)\}
```

In each group the first two commands produce horizontal braces (parentheses), while the other two – vertical braces (parentheses). Here are some examples:

This is a very nice
text box

```
\begin{xy}
(0,0)*+{\txt{This is a very nice \\%
text box}}*\frm{.}\frm{\}}
\end{xy}
```

This is a very nice
text box

```
\begin{xy}
(0,0)*+{\txt{This is a very nice \\%
text box}}*\frm{.}\frm{)}
\end{xy}
```

This is a very nice
text box

```
\begin{xy}
(0,0)*+{\txt{This is a very nice \\%
text box}}*\frm{.}\frm{)}
\end{xy}
```

A B

```
\begin{xy}
(0,0) *+++{A}; (10,7) *+++{B}%
**\frm{.}\frm{^}\};**\frm{\}}
\end{xy}
```

If you look at the last example, when we use the brackets as connectors, you will notice that the nibs of the brackets are not centered but are aligned with the reference point of one of the objects. It is essential to use the semicolon between the two brackets commands. This aligns the second bracket with the second object. Otherwise, both brackets would be aligned with the first object:

A B

```
\begin{xy}
(0,0) *+++{A}; (10,7) *+++{B}%
**\frm{.}\frm{^}\} **\frm{\}}
\end{xy}
```

To finish this paragraph, let us say a few words about *filled regions*. These are produced with the commands `\frm{*}` and `\frm{**}` and the result is that the inside of the current object is filled with ink. Moreover, the command `\frm{**}` also draws a very thin black line along the edges of the object. By default then

the program uses black ink. If a different color is desired, this color should be used as a modifier for the frame itself:



```
\begin{xy}
(0,0)****{\txt{Nice frame}}%
*[red]\frm{*}
\end{xy}
```



```
\begin{xy}
(0,0)****[o]{\txt{Nice frame}}%
*[red]\frm{**}
\end{xy}
```

Moreover, if the object is rectangular, then the framing commands for filled regions can also be used with an optional length parameter, producing a rounded rectangular frame:



```
\begin{xy}
(0,0)****{\txt{Nice frame}}*[red]%
\frm<5pt>{*}
\end{xy}
```

If you want the text to appear in a different color, add this after the framing command. Don't put a semicolon after the frame:



```
\begin{xy}
(0,0)****{\txt{Nice frame}}*[red]%
\frm<5pt>{**}%
****[white]{\txt{Nice frame}}
\end{xy}
```

2.4 Framing as object modifiers

Frames can be added to the objects also using a different approach, i.e. they can be introduced as *modifiers of objects*. In this case, the syntax we shall use is

[F<frame>]. If all we want to add is the ordinary solid frame of the same shape as the shape of the object, the syntax is, simply, [F] If, moreover, we want to apply a modifier to the frame itself, the syntax is:

[F<frame>:modifier]

where the modifier after the colon refers to the frame, not to the object. The <frames> that can be used are exactly those described above. If we intend to use an optional length argument for the frame, it should be placed, also, after the colon. Of course, we take from the framing command only the argument between the brackets:



```
\begin{xy}
(0,0)*+[F]{A\otimes B}
\end{xy}
```



```
\begin{xy}
(0,0)*+[o][F]{A\otimes B}
\end{xy}
```



```
\begin{xy}
(0,0)*+[F-o]{A\otimes B}
\end{xy}
```



```
\begin{xy}
(0,0)*+[F]{A\otimes B}
\end{xy}
```

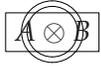
We can use several frames, if we like:



```
\begin{xy}
(0,0)*+[F]--[Foo]{A\otimes B}
\end{xy}
```

Beware: if several framings are used, they are executed in reverse order (in other words, from right to left). Also, the order of the options should not be

changed, otherwise we would get a different result. The spacing command + and - should also be used with care. Thus, to get the same result as above, changing the orders of the frames, we must use the code:



```
\begin{xy}
(0,0)*-[Foo]++[F]{A\otimes B}
\end{xy}
```

Also, if we use a modifier that changes the shape of the object, it has to be placed *before* any framing modifier.

3 The matrix extension

3.1 First steps

The `matrix` extension of `Xy-pic` provides the command `\xymatrix{...}` which allows the construction of a diagram in a way which is similar to the construction of a matrix structure in `TEX` or `LATEX`. The entries of the matrix (which are separated by usual alignment characters & on the same row, while the rows are separated by `\\`) are `Xy-pic` objects and they will play the roles of origin and target for the arrows of the diagram. In the simplest form (without arrows), `\xymatrix` produces just an ordinary matrix:

A	B	<code>\xymatrix{%</code>
		<code>A&B\\</code>
		<code>%</code>
C	D	<code>C&D }</code>

The matrix can be part of an `xy` environment or not. All the entries are in mathematical mode. It is advisable, though, to put the `\xymatrix` itself in a mathematical mode. Also, exactly as in the case of a `TEX` matrix, it is not necessary to put the same number of entries in each row. Thus, we can have matrices of the form:

A	B	<code>\begin{equation*}</code>
		<code>\xymatrix{%</code>
		<code>A&B\\</code>
C		<code>%</code>
		<code>C& }</code>
		<code>\end{equation*}</code>

or

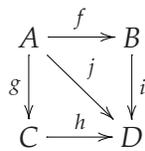
A	B	<code>\begin{equation*}</code>
		<code>\xymatrix{%</code>
		<code>A&B\\</code>
C		<code>%</code>
		<code>C }</code>
		<code>\end{equation*}</code>

As one can see from the examples, the entries of the matrices, which are, as we said, Xy-pic objects, are not introduced through the drop operator `*{}` as before. Nevertheless, we can use this operator, as well, if we need more control over the object, as we shall see a little bit later.

The syntax of the commands for arrows is, essentially, the same as in the case of the `xy` environment. There is, however, an important difference: the target of the arrow is indicated by specifying the position occupied in the matrix by the target entry. There are several ways of indicating this position, depending whether we use *absolute* or *relative* positions. For now, we shall describe the most used way, a relative one, and below we shall explain, also, some other ways of specifying the targets. Thus, the general syntax of an arrow command will be:

`\ar...[direction]label`

where the dots indicate any modifier we may use, specifying the characteristics of the arrow, while the direction is indicated through one or more of the characters `u,d,l,r` (up, down, left, arrows), depending on how many position upwards, downwards, leftwards or rightwards is the target of the arrow as compared to the origin. If no modifier is present, the standard arrow is used (which will correspond, in fact, to the modifier `@{->}`). Here follows a very simple example:

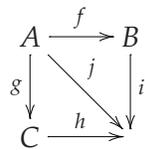


```

\begin{equation*}
\xymatrix{%
A\ar[r]^f\ar[d]_g&\ar[dr]^j&B\ar[d]^i\\
C\ar[r]^h & & D}
\end{equation*}

```

Notice that not any direction can be used as an argument of a given arrow. There is no point directing an arrow towards an absent position. One should not confuse *absent position* with *absent entry*. Thus, for instance, in the previous diagram, there is no position, say, upstairs from A , therefore we cannot draw an arrow with the origin at A with the direction given by $[u]$. We can delete, instead, the entry from the lower right corner of the diagram, without, however, deleting the preceding alignment mark and everything still works, i.e. we can still draw arrows rightwards from C and downwards from B , although the result is by no means meaningful:

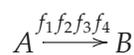


```

\begin{equation*}
\xymatrix{%
A\ar[r]^f\ar[d]_g&\ar[dr]^j&B\ar[d]^i\\
C\ar[r]^h & & }
\end{equation*}

```

Before going any further, we have to mention that, exactly as it happens in the case of the arrows used without a matrix environment, the arrows *are not* extendible. They do not lengthen automatically to accommodate some long labels. Here follows an ugly example:



```

\begin{equation*}
\xymatrix{A\ar[r]^{f_1 \ * \ f_2 \ * \ %
f_3 \ * \ f_4}&B}
\end{equation*}

```

Unfortunately, there is not much we can do to correct this, except for a *global* modification of columns or rows separation, as we shall see later.

A final point we would like to mention in this fast introduction is that the diagrams are not necessary “displayed” objects. They can be included in the text, if the matrix that includes them only has one row. These are called “one-line” diagrams and are produced by something like:

This is a very, very, very nice one-line diagram: $A \xrightarrow{f} B$. Do you like it?

This is a very, very, very nice one-line diagram:
`\xymatrix@1{A\ar[r]^f&B}`.
 Do you like it?

The only thing that makes this diagram special is, actually, the option @1 that improves the spacing of the diagram with respect to the surrounding space. Without this option, the diagram would look like this:

This is a very, very, very nice one-line diagram: $A \xrightarrow{f} B$. Do you like it?

This is a very, very, very nice one-line diagram:
`\xymatrix{A\ar[r]^f&B}`. Do you like it?

You will notice that, in the case of the absence of the option @1, the horizontal space before and after the diagram is too big (is bigger than one would actually expect to be *inside the text*).

I would like to stress, again, the importance of putting the diagram in mathematical mode. Look what would happen otherwise:

This is a very, very, very nice one-line diagram: $A \xrightarrow{f} B$. Do you like it?

This is a very, very, very nice one-line diagram:
`\xymatrix@1{A\ar[r]^f&B}`.
 Do you like it?

As you can see, in this case the *vertical* spacing of the diagram is not correct: the diagram is too low.

You might think that for simple diagrams, as the one above, it is not worth

using $\Xy-pic$ since you can get a similar result by using just standard \LaTeX command. Well, is not really like that, as you can see for yourself:

This is a very, very, very nice one-line diagram: $A \xrightarrow{f} B$. Do you like it?

This is a very, very, very nice one-line diagram:
 $\$A\overset{f}{\longrightarrow} B\$$.
 Do you like it?

You would probably agree that in this case the arrow is a bit too short and it doesn't look as good as the one produced with $\Xy-pic$.

3.2 More about arrows and labels

It was explained in some detail in the first part of this paper how different kinds of arrows can be produced, and this will not be repeated here. All the constructions made with the command \ar work as well in a $xymatrix$ environment. The essential difference is that in this environment the arrow is not constructed indicating explicitly the origin and the target, but, as previously mentioned, the arrow command is placed immediately after the origin (i.e. the corresponding entry of the matrix) and the target is indicated through the direction in which the arrow is pointing.

Also, everything we said about labels is still true in the new context.

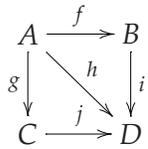
3.2.1 Other ways of specifying targets

We shall describe now, as promised, some other ways of specifying the target of an arrow, besides the relative one, described in the previous paragraph.

There are, actually, three more ways of indicating the target of an arrow:

- (a) a relative way, by using a pair of integers $[r, c]$ to indicate the entry lying r rows below and c columns to the right of the current entry. The current entry corresponds to $[0, 0]$. r and c will be negative if the target lies above or to the left of the current entry. Of course, this way of describing it is the same as the previous one. Thus, for instance, the target $[-2, 1]$ is equivalent to the target $[\text{uur}]$, while the target $[2, 1]$ is equivalent to the target $[\text{ddr}]$. The same restriction applies: the position indicated must exist already, otherwise we get an error message.

- (b) an absolute way, indicated by a pair "r, c" of *strictly positive* integer numbers, indicating the row and the column of the target. The top left entry corresponds to "1, 1". Here is an example:

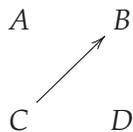


```
\begin{equation*}
\xymatrix{%
A\ar "1,2"~f \ar "2,1" _g%
\ar "2,2" ^h & B\ar "2,2"~i \\
%
C\ar "2,2"~j&D }
\end{equation*}
```

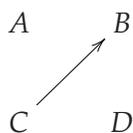
- (c) In all the examples given so far, an arrow with the origin at a given entry of the matrix had to be posted, necessarily, after the given entry (before the next alignment mark or the next new line command). Xy-pic also has a feature to insert, at a given entry of the diagram, an arrow that connects two *other* entries. This is done by indicating, instead of a single target, a pair of targets, separated by a semicolon. Any of the three ways of indicating targets described so far can be used, they can be even combined. Please remember: if you use *relative* ways of indicating the targets, they should always refer to the *current* entry of the diagram (the one where you put the arrow command). Thus, a command of the form

```
\ar [ul]; [rd]
```

will connect an entry situated one row above and one column to the left with respect to the current entry, with the entry situated one row below and one column to the right of the current entry. Below we shall give some examples, in which the same diagram is constructed by using different ways of indicating the pair of targets:



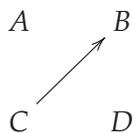
```
\begin{equation*}
\xymatrix{%
A \ar [d]; "1,2" & B \\
%
C&D }
\end{equation*}
```



```

\begin{equation*}
\xymatrix{%
A \ar [d]; [0,1] & B \\
%
C&D }
\end{equation*}

```



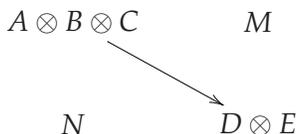
```

\begin{equation*}
\xymatrix{%
A \ar [1,0]; "1,2" & B \\
%
C&D }
\end{equation*}

```

3.2.2 Changing the targets of arrows

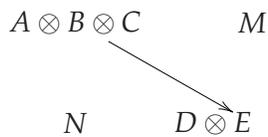
In many situations, the arrows don't point exactly in the direction we want. This may be, for instance, because of the entries at the target positions which may have big dimensions or, for instance, because in one entry enters many arrows and we want more space between their tips. Xy-pic allows changing the target at will. One way of doing this is to apply modifiers to the target objects, for instance modifying its reference point. Compare, for instance, the following two examples:



```

\begin{equation*}
\xymatrix{%
A\otimes B\otimes C & M \\
N & D\otimes E \\
}
\end{equation*}

```

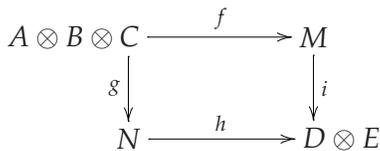


```

\begin{equation*}
\xymatrix{%
A \otimes B \otimes C \ar[dr] & M \\
N & D \otimes E
}
\end{equation*}

```

The disadvantage of this approach is that the target object itself moves. This may be desirable sometimes; it happens in the following example:



```

\begin{equation*}
\xymatrix{%
**[1] A \otimes B \otimes C \\
\ar[r]^f \ar[d]_g & M \ar[d]_i \\
N \ar[r]^h & D \otimes E
}
\end{equation*}

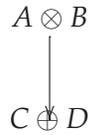
```

which you probably agree looks pretty nice. However, in most situations, especially when the entries of the diagram have approximately the same dimensions, it is preferable not to change their position. The alternative is to change the direction of the arrow. This may be done by one of the following two constructions:

1. $\pm vector$, which changes the target in the following way. It adds or subtracts $vector$ at the reference point of the target vector, places a zero-sized object at that position and then changes the target to be this new object. Notice, however, that no object is actually typeset at the new position and the original target object doesn't change position. Thus, all that happens is that the tip of the arrow changes the position.
2. $!vector$, which simply moves the center of the target by the vector.

We shall see in a moment how to describe the vectors. Before that, we would like to emphasize the difference between the two approaches. In the first approach, everything works as if we would move the object with the opposite of the vector and then set the size of the object to zero. The same thing happens in the second case, except that now we don't modify the size of the object. Therefore, the arrows

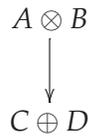
we get in the two approaches have the same direction, but the one obtained by using the first approach is longer. The simplest way to see that is to use a zero vector and compare the results:



```

\begin{equation*}
\xymatrix{%
A\otimes B\ar[d]<0pt,0pt>\!
C\oplus D}
\end{equation*}

```



```

\begin{equation*}
\xymatrix{%
A\otimes B\ar[d]!<0pt,0pt>\!
C\oplus D}
\end{equation*}

```

Of course, what we get in the second example is just the default, i.e. the diagram we would get without any modifier.

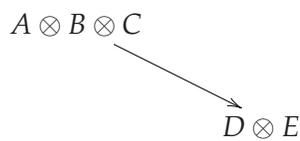
Now, the details about the vectors. They can be indicated in one of the following ways:

- a) as a pair of T_EX lengths, between angular brackets: $\langle D_x, D_y \rangle$, representing a vector in the plane with the respective coordinates.
- b) as one of the following letters or combinations of letters (in upper case!): L, R, U, D, UL, DL, UR, DR, corresponding to the vectors that would move the reference point of the target object to one of the corners of its bounding box (the combinations of letters), or to one of edges of the object (the single letters).
- c) 0 – the zero vector. As we mentioned above, only ± 0 has a real effect.
- d) in the form $/d\ length/$. This is the vector in the plane going in the direction given by d , and having the given $length$. The direction d is given either by one of the directions from the figure 1, by either one of the following:
 - (i) $va(\alpha)$ – representing the absolute angle (i.e. the angle made by the vector with the positive direction of the x -axis);

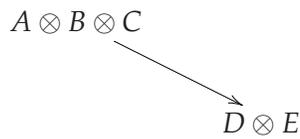
- (ii) $d : a(\alpha)$ – representing the relative angle (i.e. the angle made by the current arrow with the vector);
- (iii) $d : (x, y)$ – the relative vector, given by its components (just coordinates, not lengths!);
- (iv) \hat{d} or d_- , which are shorthand for $d : a(90)$ and $d : a(-90)$.

Here the absolute angle is given in radians, while the relative angle is given in degrees. d may be absent if the direction of the vector is the direction of the current arrow. In this case, the only effect is that the arrow modifies its lengths

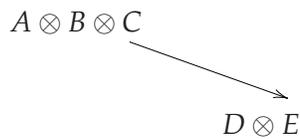
Here follows some examples, to illustrate what we just said:



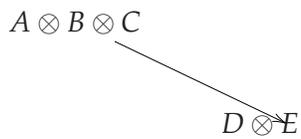
```
\begin{equation*}
\xymatrix{%
A\otimes B\otimes C\ar[dr]!<d:a(45) -10pt/&\&D\otimes E
}
\end{equation*}
```



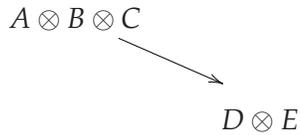
```
\begin{equation*}
\xymatrix{%
A\otimes B\otimes C\ar[dr]!<d:a(45) -10pt/&\&D\otimes E
}
\end{equation*}
```



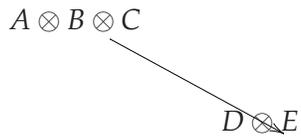
```
\begin{equation*}
\xymatrix{%
A\otimes B\otimes C\ar[dr]!<10pt,10pt>&\&D\otimes E
}
\end{equation*}
```



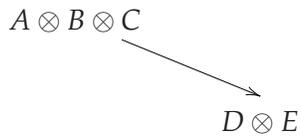
```
\begin{equation*}
\xymatrix{%
A\otimes B\otimes C\ar[dr]%
+/\va(2) 10pt/\&\&D\otimes E
}
\end{equation*}
```



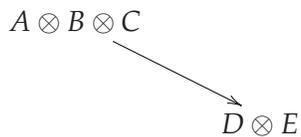
```
\begin{equation*}
\xymatrix{%
A\otimes B\otimes C\ar[dr]%
-/d:a(45) 20pt/\&\&D\otimes E
}
\end{equation*}
```



```
\begin{equation*}
\xymatrix{%
A\otimes B\otimes C\ar[dr]%
+/d:(1,2) 10pt/\&\&D\otimes E
}
\end{equation*}
```



```
\begin{equation*}
\xymatrix{%
A\otimes B\otimes C\ar[dr]%
+/10pt/\&\&D\otimes E
}
\end{equation*}
```



```
\begin{equation*}
\xymatrix{%
A\otimes B\otimes C\ar[dr]%
+/\ul 10pt/\&\&D\otimes E
}
\end{equation*}
```

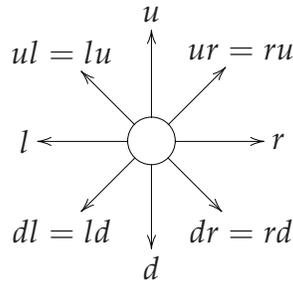


Figure 1: Directions

Notice that the modifier should always be placed immediately after the characters indicating the targets of the arrow (before the label, if there is one).

3.2.3 More ways of curving arrows

We saw in the first part of this article how it is possible to construct a curved arrow between two objects. This method works, as well, in the case of matrix diagrams:

$$A \overset{f}{\rightarrow} B$$

```

\begin{equation*}
\mathrm{xymatrix}{%
A \ar@/^/[r] |f& B }
\end{equation*}

```

In the case of matrix diagrams, however, we have more ways of constructing curved arrows. One of them is to use a pair (in, out) of directions to specify in which direction the arrow leaves the source and from which direction it enters the target. The directions are indicated in the figure 1. We still have to specify the target, of course:

$$A \overset{f}{\curvearrowright} B$$

```

\begin{equation*}
\mathrm{xymatrix}{%
A \ar@(ur,ul) [rr] |f&& B }
\end{equation*}

```

This construction is particularly useful where we want to have an arrow for

which the source and the target coincide, as one can see in the following example, taken from the User's guide:



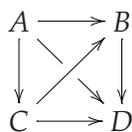
```
\begin{equation*}
\xymatrix{%
x \ar@(ul,dl) [] |{\text{id}}\ar@/^/[rr] |f
%
&&f(x)\ar@/^/[ll] |{f^{-1}}
%
}
\end{equation*}
```

3.2.4 "3D" diagrams

In the first part of this article, we saw how it is possible to "put a hole" in an arrow, in order to suggest a 3D look for a diagram. We did this by hand, trying to find the place where the two arrows meet. In the case of the matrix diagrams, there is a simpler way of doing this and the program takes care of finding the intersection. The idea is to place a hole in the current arrow at the position where the current arrow intersects the arrow connecting targets t and t' . The syntax of the command is

```
!!{t;t'}\hole
```

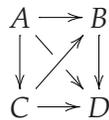
Here is a very simple example:



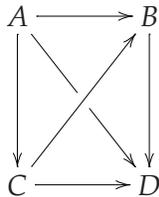
```
\begin{equation*}
\xymatrix{A\ar[r]\ar[d]
\ar[dr] !!{"2,1";"1,2"}\hole&B\ar[d]
%
C\ar[r]\ar[ur]&D}
\end{equation*}
```

3.3 Some fine tuning

We already saw, when we discussed about one-line diagrams, that the command `\xymatrix` takes optional arguments, and we encountered one of them. It is possible, also, to specify, for instance, the spacing of rows or columns, by using arguments of the form `@R<dim>` or `@C<dim>`, respectively. They can be used together, of course, as in the following examples:



```
\begin{equation*}
\xymatrix@R20pt@C15pt{A\ar[r]\ar[d]\ar[dr]
&B\ar[d] \\
C\ar[r]\ar[ur]&D}
\end{equation*}
```



```
\begin{equation*}
\xymatrix@R50pt@C35pt{A\ar[r]\ar[d]\ar[dr]
&B\ar[d] \\
C\ar[r]\ar[ur]&D}
\end{equation*}
```

If we desire uniform spacing, this is done by the option `@!`. Similarly, if we only want the columns or the rows to be uniformly spaced, we can use the commands `@R!` and `@C!`.

The default spacing of a diagram is dictated by the dimensions of the entries. We can modify these dimensions by using commands of the form `@R+<dim>`, where, of course, `R` can be replaced with `C` or can be dropped if we wish to modify both the spacing of the rows and columns. Below is another example. You can find the entire discussion in the Reference Manual.

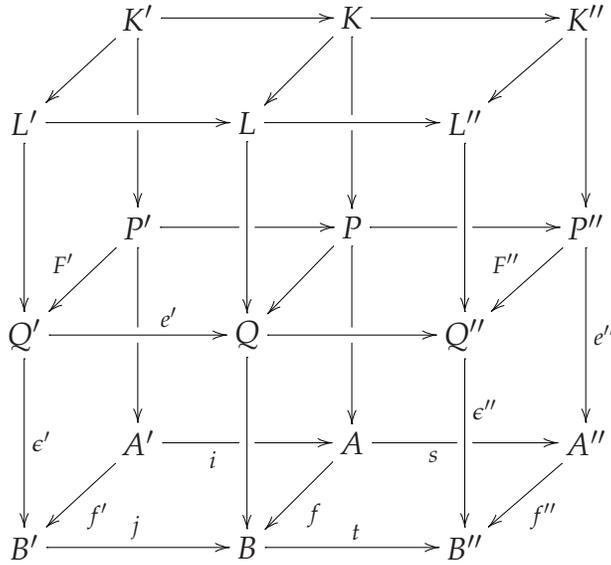
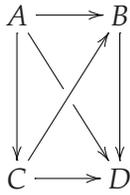


Figure 2: First diagram



```

\begin{equation*}
\mathrm{xymatrix@R+2pc{A\ar[r]\ar[d]\%
\ar[dr]||!"2,1";"1,2"}\hole&B\ar[d]\%
%
C\ar[r]\ar[ur]&D}
\end{equation*}

```

4 A little bit of technique ...

Now that you know how to draw simple commutative diagrams, let us try our hand on something more difficult. Suppose we have to draw the diagram in the figure 2. At first sight, it seems very difficult and the beginner may want to give up trying. But, as the title of this section suggests, we'll get by with a little bit of technique. By inspecting the diagram, we notice immediately that it corresponds to a 6×6 matrix. So, let us start by constructing only the matrix, without any arrows, and put a smile “ \smile ” in the positions where there are no objects in our diagram. We get the construction from the figure 3. One of the advantages of

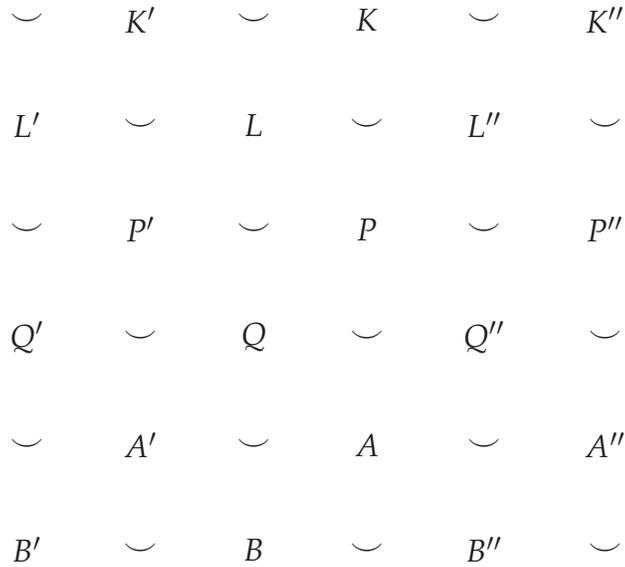


Figure 3: First diagram: a draft

putting the vertices of the diagram first is that now you can draw arrows in any direction you want, as long as they point towards one of the vertices and, thus, you can process the file from time to time to see how things are working. However, the main advantage is that you can easily identify the direction of each arrow. Thus, for instance, an arrow from K' to L' should go down one position and left one position. In other words, it will be produced by something like `\ar@{->}[d1]`. If we check, we will get the second draft from the figure 4. Now that we can see that it works, we can add, step by step, all the arrows, without caring about how to treat the intersections and then delete all the smiles. We get the diagram 5 Handling the intersections, in order to obtain a 3D illusion, is easy. It is enough to put holes in some of the arrows at well chosen positions. This is done by a command of the form `!!{t;t'}\hole`. This means, you will recall, that we put a hole in the current arrow at the point where this arrow intersects with the arrows that connect the targets t and t' . These targets are described either in an absolute way or a relative way, with respect to the current entry of the diagram. Thus, to put a hole in the arrow that connects K' to P' , and at the point of intersection with the arrow connecting L' to L , we can use, for instance, the command:

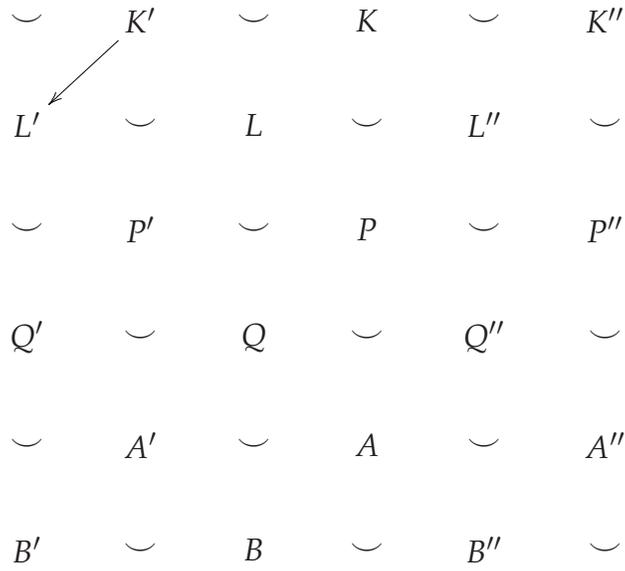


Figure 4: First diagram: second draft

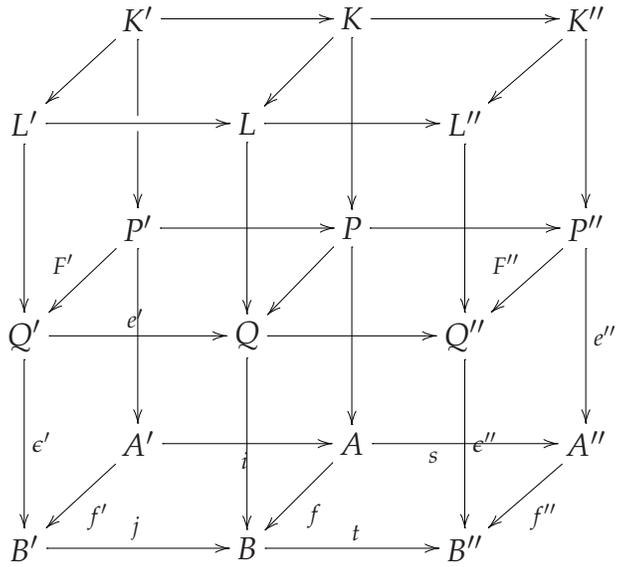


Figure 5: First diagram: third draft

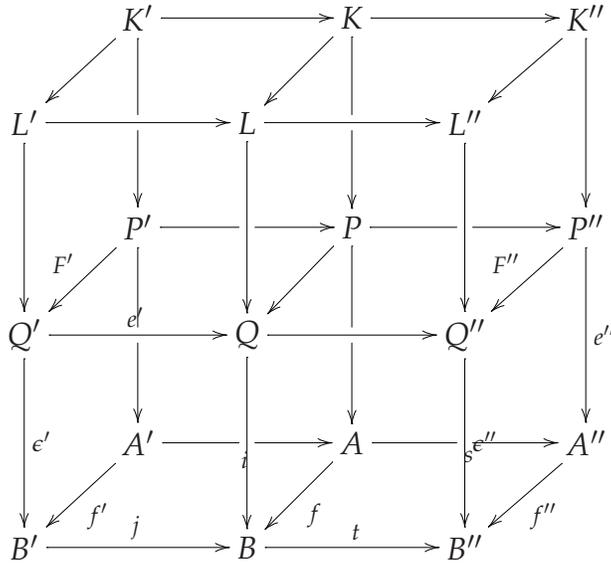


Figure 6: First diagram: last draft

`!!{[d1];[dr]}\hole`

(because L' is one row below and one column to the left of K' , while L is one row below and one column right with respect to the same entry) or

`!!{"2,1";"2,3"}\hole`

(because L' and L , respectively, are at the positions "2,1" and "2,3", respectively, of the diagram). After adding all the holes, we get the diagram 6. Now, final tuning has to be done. Clearly, for some of the labels the default placing doesn't work, so we have to "slide" them along the arrow a little bit. After doing this, we get the desired diagram, which has been produced by the code:

```

\begin{equation*}
\xymatrix{%
&K'\ar@{->}[rr]\ar@{->}[d1]\ar@{->}[dd]!!{[d];[d]}\hole&&K %
\ar@{->}[rr]\ar@{->}[d1] \ar@{->}[dd]!!{[d];[d]}\hole &&
K''\ar@{->}[d1]\ar@{->}[dd]
\\
%

```

```

L'\ar@{->}[rr]\ar@{->}[dd]&&L\ar@{->}[rr]\ar@{->}[dd]&&L''
\ar@{->}[dd]\
%
&P'\ar@{->}[rr]||\{[r];[r]\hole\ar@{->}[dl]_{F'} %
\ar@{->}[dd]||\{[d];[d]\hole&&P \ar@{->}[rr]||\{[r];[r]\hole
\ar@{->}[dl]\ar@{->}[dd]||\{[d];[d]\hole
&&P''\ar@{->}[dl]_{F''}\ar@{->}[dd]^{\{e''\}}\
%
Q'\ar@{->}^{(.65)\{e'\}}[rr]\ar@{->}[dd]^{\{\epsilon'\}}&&Q\ar@{->}[dd]
\ar@{->}[rr]&&Q''\ar@{->}[dd]^{(.35)\{\epsilon''\}}\
%
&A'\ar@{->}[rr]_{(.35)i}||\{[r];[r]\hole\ar@{->}[dl]^{\{f'\}}&&A %
\ar@{->}[rr]_{(.35)\{s\}}||\{[r];[r]\hole\ar@{->}[dl]^{\{f''\}}&&A''
\ar@{->}[dl]^{\{f''\}}\
%
B'\ar@{->}[rr]^j&&B\ar@{->}[rr]^t&&B''
%
}
\end{equation*}

```

5 Two final examples

To finish, we shall make two other real life examples, taken from category theory. The first one (see figure 7) is taken from [2], pag. 86 and it was produced by the code

```

\begin{equation*}
{\SelectTips{cm}{}}
\xymatrix@+1.5pc{%
C\ar[r]^{\gamma}\ar@<.6ex>[d]^k\ar@<-.6ex>[d]_h
\ar'1[dd]'[dd][dd]&A\ar@<.6ex>[d]^g\ar@<-.6ex>[d]_f\
D\ar[r]_{\beta}\ar[d]_p&B\ar[d]^q\
P\ar[r]_{\alpha}&Q
%
} }

```

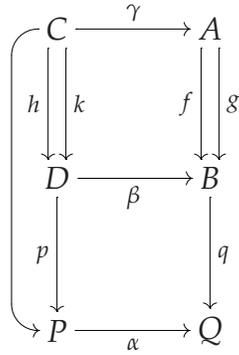


Figure 7: A first diagram from category theory

`\end{equation*}`

The second one (see Figure 8), instead, was taken from [3], pag. 31 and it was

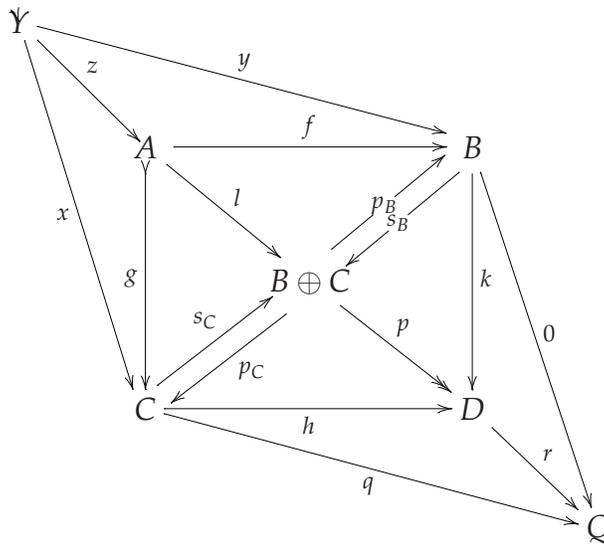


Figure 8: A second diagram from category theory

produced by the code:

`\begin{equation*}`
`\xymatrix@+0.5pc{%`

```

Y \ar "2,2"+/d:(1,1) -3pt/^z \ar "2,4"!/d:a(45) -5pt/ ^y%
\ar "4,2"!/d:a(45) -5pt/_x\ar&&&&\%
&*++[o]{A}\ar"2,4"~f\ar@{>->} "4,2"_g\ar"3,3"~1&&*++[o]{B}%
\ar@<.75ex>"3,3" | {s_B}
\ar@<.75ex>"3,3"; "2,4" | {p_B}\ar"4,4"~k\ar"5,5"~0&\%
&&*++[o]{B\oplus C}\ar@{->>} "4,4"~p&&\%
&C\ar@<.75ex>"3,3"~{s_C}\ar@<.75ex>"3,3"; "4,2"~{p_C}%
\ar"4,4"_h\ar"5,5"_q&&D\ar"5,5"~r&\%
&&&&Q
%
}
\end{equation*}

```

References

- [1] Blaga, P.: *Commutative Diagrams with X_y-pic I. Using the Kernel Functions*, PracTeX Journal, 4/2006
- [2] Borceux, F.: *Handbook of Categorical Algebra 1: Basic Category Theory*, Cambridge University Press, 1994
- [3] Borceux, F.: *Handbook of Categorical Algebra 2: Categories and Structures*, Cambridge University Press, 1994
- [4] Campani, C.A.P.: *Tutorial de X_y-pic* (in portuguese), 2006, available from CTAN
- [5] Goosens, M., Rahtz, S., Mittelbach, F.: *The L^AT_EX Graphics Companion*, Addison-Wesley, 1997
- [6] Rotman, J.: *An Introduction to Homological Algebra*, Academic Press, 1979
- [7] Rose, K.H.: *How to typeset pretty diagram arrows with T_EX– design decisions used in X_y-pic*, in Jiri Zlatuska, editor, *EuroT_EX '92 – Proceedings of the 7th European T_EX Conference*, pages 183-190, Prague, Czechoslovakia, September 1992. Czechoslovak T_EX Users Group.
- [8] Rose, K.H.: *X_y-pic User's Guide*, version 3.7, 1999, available from CTAN

- [9] Rose, K.H., Moore, R.: *Xy-pic Reference Manual*, version 3.7, 1999, available from CTAN
- [10] Valiente Feruglio, G.: *Typesetting Commutative Diagrams*, TUGboat, **15**(4), 1994, 466-484.

[Journal](#)
[home page](#)
[General](#)
[information](#)
[Submit an](#)
[item](#)
[Download](#)
[style files](#)
[Copyright](#)
[Contact us](#)

THE PracTeX Journal



Create Trees and Figures in Graph Theory with PSTricks

Manjusha S. Joshi

Abstract

Drawing trees and figures in the mathematical area of graph theory is a requirement for researchers and teachers. This includes loops, arcs, nodes, and weights for edges. This article aims to get started with PSTricks by keeping two commands in mind, viz. `pstree` and `psmatrix`. Using the most useful options of these commands the reader can draw tree diagrams, loops, node labels, and add weights to edges. Once the diagrams are completed they can be added to a TeX file. With a little working knowledge about drawing figures in graph theory the reader can then produce his or her own.

Manjusha Joshi has an M.Sc. in Mathematics and M.Phil in 'Bezier Curves and its applications' from the University of Pune.

She has been working at Bhaskaracharya Pratishthana (www.bprim.org) for 9 years, as editorial assistant for 'Bona Mathematica', Journal for Mathematics, using LaTeX for typesetting. Recently, she began a position as a visiting faculty member.

She has attended two TUG meetings and is member of the Indian TeX User Group (TUG), Pune LUG.

Manjusha is a free software enthusiast and strongly believes that free mathematical software is useful in popularizing mathematics among students. She has a special interest in Computational Mathematics, Computer Graphics, and the promotion of free software.

Contact her at msj@bprim.org.

- [PDF version of paper](#)
- [Comment on this paper](#)
- [Send submission idea to editor](#)

Create Trees and Figures in Graph Theory with PSTricks

Manjusha S. Joshi

Bhaskaracharya Institute in Mathematics, Pune, India

Abstract Drawing trees and figures in the subject Graph Theory (Mathematics) is a requirement for researchers and teachers. This include loops, arcs, nodes, weights for edges. This article aims to get started with PSTricks by keeping two commands in mind viz. `pstree` and `psmatrix` with their most useful options with the help of which a reader can draw neat tree diagrams, loops, can label nodes, add weights to edges. After completing the diagram, one can add these diagrams to T_EX file. So working knowledge about drawing figures in graph theory can be gained.

Contents

1 Preliminaries	2
2 <code>psmatrix</code>	2
2.1 Types of edges: lines, arcs and loops	4
2.2 Directions and Weights to Edges	6
2.3 Additional PSTricks	7
3 <code>pstree</code>	9
4 Include figures	12
5 Most common mistakes and some tips	13
A More figures	14
References	19

1 Preliminaries

Prerequisites: Basic knowledge of L^AT_EX.

Expectations: After going through this article newbie to PSTricks can also create his/her own figure files and will be able to insert it in his/her T_EX document.

System requirement: L^AT_EX with PSTricks packages

Web site: You can get the required package and information from <http://tug.org/PSTricks/>

Preamble should include packages: pstricks, pst-node, pst-tree

2 psmatrix

In the command `\psmatrix` the word ‘matrix’ is involved. Let us have a look at matrix. We write a matrix as $A_{3 \times 4}$, which indicates that the matrix A has 3 rows and 4 column. These are called as dimensions of the matrix.

$$\begin{matrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{matrix}$$

Each element in the matrix can be identified with, *which row?* and *which column?* e.g. a_{23} indicates that the element is in the 2nd row and 3rd column. So the order is row number followed by column number.

Now, let us have a look at the environment `\psmatrix`

```
 $\psmatrix
```

```
 \endpsmatrix
```

```
 $
```

Though, this environment doesn’t start with `\begin`, it ends with `\end` command. Note this difference with other T_EX environments.

To use these special PSTricks commands one requires to include `\usepackage{pstricks,pst-node,pst-tree}`. Here is the minimum code that includes the necessary preamble.

```

\documentclass{article}
\usepackage{graphics,graphicx}
\usepackage{pstricks,pst-node,pst-tree}
\begin{document}
$
\psmatrix
u&v&z\\
x&y
\endpsmatrix
$
\end{document}

```

The out put will look exactly like this:

$$\begin{matrix}
 u & v & z \\
 \\
 x & y &
 \end{matrix}$$

Note that the vertical and horizontal distances are by default. We can control them by adding optional argument to `psmatrix` command with `\psmatrix[colsep=2.5cm,rowsep=1.5cm]` thus we ask to keep column separation as 2.5cm and row separation as 1.5cm.

Also, frequently we like to denote these points with small circles around the letters like . This can be obtained by `\psmatrix[mnode=circle]`, which means choose node of type circle. So after adding these two things our code will be:

```

\documentclass{article}
\usepackage{graphics,graphicx}
\usepackage{pstricks,pst-node,pst-tree}
\begin{document}
$
\psmatrix[colsep=2.5cm,rowsep=1.5cm,mnode=circle]
&u&v&z\\
&x&y&
\endpsmatrix
$
\end{document}

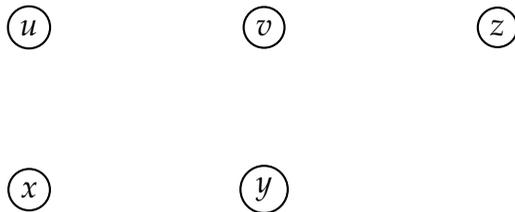
```

```

&x&y
\endpsmatrix
$
\end{document}

```

The output of the code looks like:



Such an output can be used to draw the vertices of a graph.

2.1 Types of edges: lines, arcs and loops

Now we would like to join vertices by edges which can be of types: Lines, arc or loop. For joining the vertices remember the matrix way to locate an element. The same way will be followed to refer to an element.

Now we want to join elements u and v in the above figure. What is the place of u ? It is in the first row and the first column, so refer to it by $(1,1)$, now since v is in the first row and the second column, it is referred to by $(1,2)$; remember the order (row,column). Finally use the command `\ncline{1,1}{1,2}` to draw line segment between these two nodes. Many times just the line is not sufficient, we are require to denote the direction also. For this, add `{<->}`, from whom towards whom? If $v \rightarrow u$ is the direction then order should change to `{1,2}{1,1}`.

```

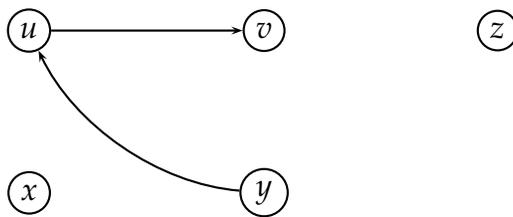
\documentclass{article}
\usepackage{graphics,graphicx}
\usepackage{pstricks,pst-node,pst-tree}
\begin{document}
$
\psmatrix[colsep=2cm,rowsep=1cm,mnode=circle]
u&v\\

```

```

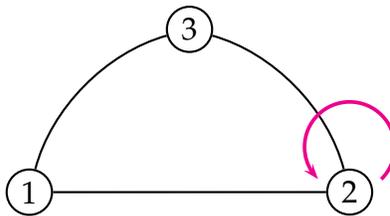
z\\
x&y
\ncline{<->}{1,2}{1,3}
\ncarc[arcangle=-30]{<->}{1,1}{2,2}
\endpsmatrix
$
\end{document}

```



`\ncarc[arcangle=-30]{<->}{1,1}{2,2}` this command is useful to draw a circular edge between the element in first row and first column to the element in second row second column.

To obtain a loop for a node `\nccircle{->}{1,1}{.5cm}` this command is used. If you want to show the direction, you can add bracket `{->}` followed by node (referred by its row,column) followed by radius.



```

$
\psmatrix[colsep=1.5cm,rowsep=1.5cm,mnode=circle]
&3\\
1&&2
\ncline{2,1}{2,3}

```

```

\ncarc[arcangle=-30 ]{2,3}{1,2}
\ncarc[arcangle=-30]{1,2}{2,1}
\ncircle[nodesep=4pt]{->}{2,3}{.6cm}
\endpsmatrix
$

```

2.2 Directions and Weights to Edges

Directions can be shown with the options `{->}`, `{<-}`, `{<->}`. All the times these arrows appears at the end points of the edge. This can be overcome by the package `pstricks-add` discussed in the section 2.3. Also size of the arrow is fixed but if you want to increase it, see section 2.3.

The distance between arrows and nodes is by default zero. If some separation is required then option `[nodesep=1pt]` is available.

You can add weights to edges. `\ncline{1,1}{1,2}^{\{1\}}` will print weight as '1' upside of the edge joining nodes u and v . These options are available:

- < left side of the edge
- > right side of the edge
- ^ upside of the edge
- _ down side of the edge

```

\documentclass[a4paper,12pt]{article}
\usepackage{pstricks,pst-node,pst-tree}
\begin{document}
$
\psmatrix[colsep=2.5cm,rowsep=1.5cm,mnode=circle]
&u&v\
z\
&x&y
\ncline[nodesep=5pt]{<->}{1,2}{1,3}^{\{1\}}
\ncline{<->}{3,2}{1,2}>\{3\}
\ncline{<->}{3,2}{3,3}_\{2\}
\ncarc[arcangle=-30]{<->}{3,3}{1,3}<\{6\}
\ncarc[arcangle=-30]{<->}{1,2}{3,2}>\{2\}
\ncline{<->}{2,1}{1,2}<\{10\}
\ncline{->}{2,1}{3,2}_\{5\}

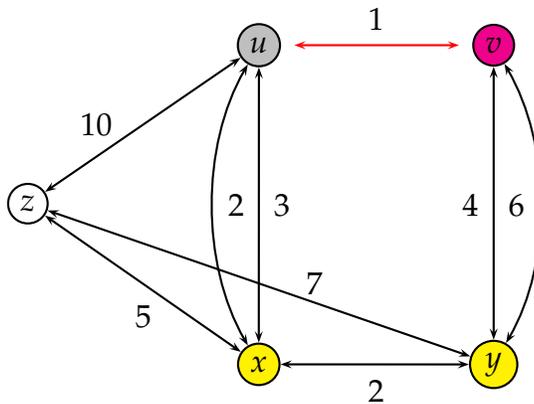
```

```

\ncline{<->}{3,3}{2,1}>{7}
\ncline{<->}{1,3}{3,3}<{4}
\endpsmatrix
$
\end{document}

```

With `\ncarc` option 'arcangle' is available. Angle is between 0 and 180 if you go anticlockwise and between 0 and -180 if you go clockwise. If you don't want arrows you can skip the argument `{<->}`.



Look at the red line between u and v . It is separated from nodes. This is the effect of node separation.

2.3 Additional PSTricks

In the above figure observe that arrows are small and at the end of the edge. Some times it may required that arrows should lie in between instead of end of the line. Also, big size arrows, can highlight direction of the edge.

With `pstricks-add` package, we can obtain arrows in between the edge instead of at the end points. Also, one can increase the size of arrows.

Preamble should include package: `pstricks-add`

Make sure that the package is a part of your system, otherwise inclusion of it in the code will cause errors. Download the package from PSTricks site. Include it

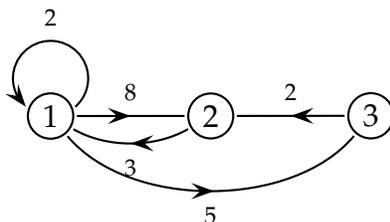
in your working directory and check that your code is working properly. Then try to save it in the appropriate directory.

```

\documentclass[a4paper,12pt]{article}
\usepackage{pstricks,pst-node,pst-tree,pstricks-add}
\begin{document}
\pagestyle{empty}
$
\psmatrix[colsep=1.5cm,rowsep=1.5cm,mnode=circle]
1&2&3
\psset{arrowscale=2}
\psset{ArrowInside=->,nodesep=1pt}
\everypsbox{\scriptstyle}
\ncline{1,3}{1,2}^2
\ncline{1,1}{1,2}^8
\ncarc[arcangle=30]{1,2}{1,1}_{3}
\ncircle{->}{1,1}{.5cm}^2
\ncarc[arcangle=-50]{1,1}{1,3}_{5}
\endpsmatrix
$
\end{document}

```

With PSTricks-add we have option `arrowscale=2`. By default `arrowscale=1`. When we want to change it we can give values like 2, 3 as per our requirement. Normally `arrowscale=2` is enough. One more interesting command with `pstricks-add` is `ArrowInside=->` this will set arrows inside the edges instead of endpoints. That way they are more visible.



```

$
\psmatrix[colsep=1.5cm,rowsep=1.5cm,mnode=circle]

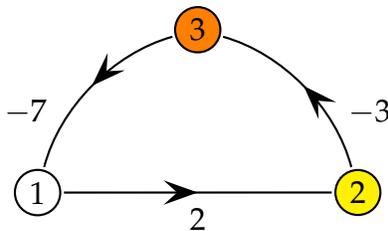
```

```

&[fillstyle=solid,fillcolor=orange]3\\
1&&[fillstyle=solid,fillcolor=yellow]2
\psset{arrowscale=3}
\psset{ArrowInside=->,nodesep=1pt}
\ncline{2,1}{2,3}_{2}
\ncarc[arcangle=-30 ]{2,3}{1,2}>{-3}
\ncarc[arcangle=-30]{1,2}{2,1}<{-7}
\endpsmatrix
$

```

Observe more options with node `[fillstyle=solid, fillcolor=orange]`. This makes the node full of color orange.



3 pstree

In preamble add one more package `\pstree`. It has tree like structure:

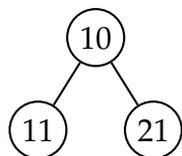
One root and many branches. We can decide what will be the height of the branches. Tree also expands according to the number of branches it has and how healthy branches are i.e. how big is the size of the branch.

With the main command `\pstree{}{}` our tree starts. It's first argument carries root of the tree and second argument is branches if any. Unlike the real tree it grows upside down. The branches of the tree are in pair of braces i.e. in `{ }`. Each branch can be again considered for a new tree to grow. But it should be inside the braces till the whole tree ends.

Heights of the branches are *levels*. It can set by optional argument with `pstree` as `\pstree[levelsep=35pt]`. It has global effect. You can control level locally by `\pstree[thislevelsep=20pt]`. It has a limited effect for the particular tree.

To obtain circular nodes in pstree use `\Tcircle{10}`. The argument in pair of braces is the matter which you want to print inside the node.

```
\documentclass{article}
\usepackage{graphics,graphicx}
\usepackage{pstricks,pst-node,pst-tree}
\begin{document}
\pstree[levelsep=35pt]{\Tcircle{10}}
{
\Tcircle{11} \Tcircle{21}
}
\end{document}
```



Observe :

1. pstree does not require \$ but you can use \$\$ if required.
2. pstree connects nodes i.e. vertices by line segments by default.
3. levelsep controls height of the branches.

You can obtain dashed line, dotted lines or 'noline' between two nodes. There are different types of nodes, like triangular, oval shape etc. To get oval use the command `\Toval{abc}`, to obtain diamond shape of the node use `\Tdia{abc}`, to get triangular shape node `\Ttri{abc}` notice the occurrence of capital **T** followed by small **t**, one may skip other 't' thinking that spelling of triangle contains 'tri' so 'Tri' will be fine, **No!**, **T** is for various types of nodes that can be taken as Tree nodes, so **T** is necessary.

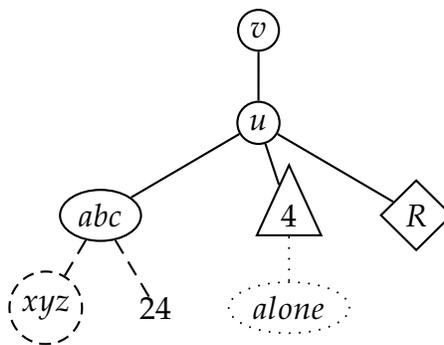
`\Tr{10}` will produce a node without any shape around 10. Following code illustrates use of various nodes with tree.

```

 $\backslash\pstree[levelsep=35pt]{\Tcircle{v}}{\{
\pstree{
\Tcircle{u}}{\{
\psset{linestyle=solid}\pstree{\Toval{abc}}{\{
\psset{linestyle=dashed}\Tcircle{xyz} \Tr{24}
}
\pstree{\Ttri{4}}{\psset{linestyle=dotted}\Toval{alone} }\Tdia{R}
}}
}$ 

```

In the code `\psset` command is used. This allows us to set things related to line style, width, arrows etc. in between the code. With `linestyle` there are four possibilities, none, dashed, dotted, solid. For line thickness use `linewidth=1.5pt`



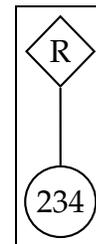
Observe that whatever is line style for the node same is that for the edge joining node from its root. What will happen if we set `linestyle` as none with node type circle? Let us check.

First draw with `linestyle` as solid and then with none :

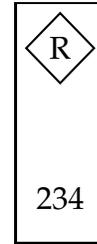
```

 $\backslash\pstree{\Tdia{R}}{\psset{linestyle=solid}\Tcircle{234}}$ 

```



```
\pstree{\Tdia{R}}{\psset{linestyle=none}\Tcircle{234}}
```



As soon as we said `linestyle=none` it removed edge between R and 234 and also drew invisible circle around 234.

4 Include figures

To include figures, one way is to write the code directly in your running text.

```
\pstree{\Tdia{R}}{\psset{linestyle=solid}\Tcircle{234}}
```

 after compiling your .tex file in the usual way the figure also gets compiled automatically and will sit in the output at the corresponding place. Yes! you can add code directly with out saying `\includegraphics{}`.

Also, one can include this code in

```
\begin{figure}  
  \pstree{\Ttri{ABC}}{  
  \end{figure}
```

Which will produce:



Another way is to write the code separately in a file with filename say `figno.tex` then compile it to get .ps file, from that obtain .eps file with some software like

Gimp or Ghostview then include .eps file just like other .eps files: `\includegraphics{figno.eps}`

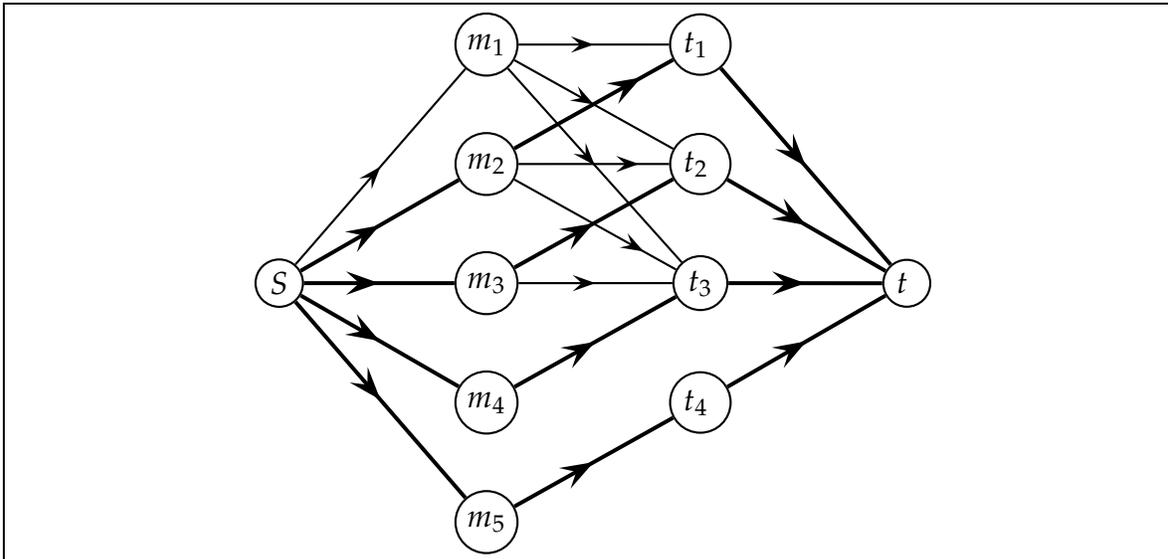
This has some advantage if you want to send your tex file and figures to some one; it is good idea to send it in eps form, so that the receiver gets freedom to adjust the size and place of the figure, and the figure will remain intact. Remember to add `\usepackage{graphics,graphicx}` in the preamble.

Always, convert your file to .ps format. If you want pdf format that can be obtained by `pstopdf` command.

5 Most common mistakes and some tips

- First try some small code of PSTricks in your system to make sure that required packages are already installed in your system.
- Make sure that you have included graphics package in the preamble.
- Use `psmatrix` inside pair of `$`'s.
- In `psmatrix` do not leave blank line; it will cause errors.
- Sometimes we include command `\nc1line{2,2}{2,3}` and in the output expect two nodes to get joined but in output we find that there is no line. Check that these 2 nodes are actually present in your matrix. If one of these is not there, how can a line be drawn between nodes one of which is not present? This happens since we can have blank node in between.
- While using `pstree` environment always write `{ }` braces in pairs and start with `new \pstree` inside these braces. Use some editor like kile (with Linux) which highlights corresponding brackets. It helps a lot.
- Every time write new nested `pstree` on a new line. That will help to structure your code and it will be easy to edit it.
- After every nested `pstree` compile the code. If due to some reason your code is giving error and you are not able to fix it, delete or comment to the latest nested `pstree` and compile again. If the code is error free now. It means the recent `pstree` has some errors to be corrected. Now again add the code for new nested tree.

A More figures



- Observe that some lines are thick in the figure. With `\ncline` optional argument `[linewidth=1.5pt]` has been tried. This way we can make a few paths bold to highlight them among others. `linewidth=1pt` is the default line width.

```

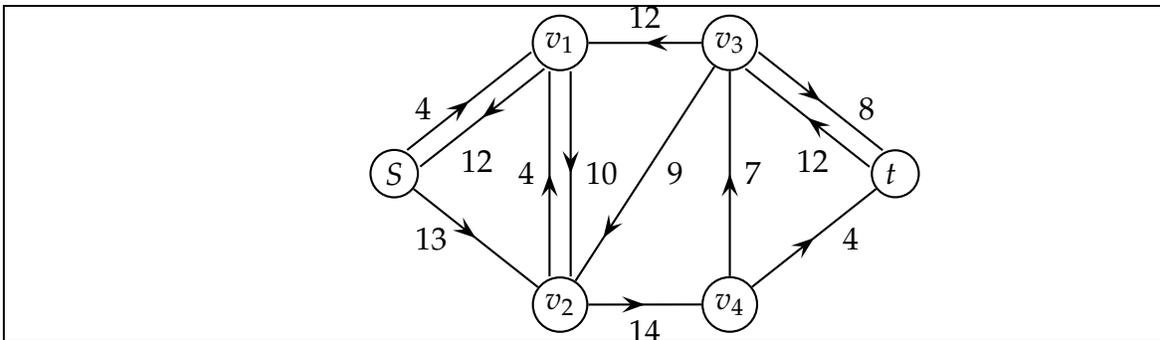
$
\psmatrix[colsep=2cm,rowsep=.7cm,mnode=circle]
&m_1&t_1\;\;
&m_2&t_2\;\;
S&m_3&t_3\;&t\;\;
&m_4&t_4\;\;
&m_5&[mnode=none,name=A]
\psset{ArrowInside=->,arrowscale=2}
\ncline{3,1}{1,2}
\ncline[linewidth=1.5pt]{3,1}{2,2}
\ncline[linewidth=1.5pt]{3,1}{3,2}
\ncline[linewidth=1.5pt]{3,1}{4,2}
\ncline[linewidth=1.5pt]{3,1}{5,2}
\ncline[linewidth=1.5pt]{3,2}{2,3}
\ncline{3,2}{3,3}
\ncline[linewidth=1.5pt]{4,2}{3,3}

```

```

\ncline[ArrowInsidePos=.8]{2,2}{3,3}
\ncline[ArrowInsidePos=.8]{2,2}{2,3}
\ncline[linewidth=1.5pt,ArrowInsidePos=.8]{2,2}{1,3}
\ncline{1,2}{3,3}
\ncline{1,2}{2,3}
\ncline{1,2}{1,3}
\ncline[linewidth=1.5pt]{5,2}{4,3}
\ncline[linewidth=1.5pt]{1,3}{3,4}
\ncline[linewidth=1.5pt]{2,3}{3,4}
\ncline[linewidth=1.5pt]{3,3}{3,4}
\ncline[linewidth=1.5pt]{4,3}{3,4}
\endpsmatrix
$

```



- In this figure directions of arrows are different between the same pair of nodes. That can be highlighted with ArrowInside option. If two lines between same pair of nodes are drawn they may get overlapped and we cannot point out the opposite directions of lines, also weights of the lines, if any. It is better if we can have small separation between two edges. The option offset=4pt is used to achieve the separation.

```

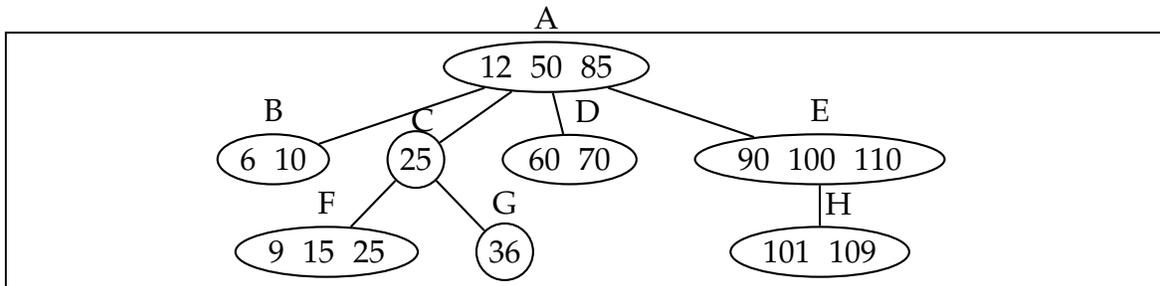
$
\psmatrix[colsep=1.5cm,rowsep=1cm,mnode=circle]
&&v_1&v_3\\
&S &&t\; \\
&&v_2&v_4
\psset{ArrowInside=->,arrowscale=2}
\ncline{1,4}{1,3}^{12}

```

```

\ncline{3,4}{1,4}>{7}
\ncline{3,4}{2,5}>{4}
\ncline{3,3}{3,4}_ {14}
\ncline{2,2}{3,3}<{13}
\ncline[ArrowInsidePos=.8]{1,4}{3,3}>[npos=.7]{9}
\psset{nodesep=0pt,offset=4pt}
\ncline{1,3}{3,3}>{10}
\ncline{3,3}{1,3}<{4}
\ncline{1,4}{2,5}>{8}
\ncline{2,5}{1,4}_ {12}
\ncline{2,2}{1,3}<{4}
\ncline{1,3}{2,2}_ {12}
\endpsmatrix
$

```



- Observe use of `\SpecialCoor` for special coordinates to label object by referring it by its given name. To assign name optional argument with node command can be used. e.g. `\Toval[name=d]{12}`, here 12 will go inside oval shape like  and this node labelled by 'd' for internal use. So reader cannot view it.

```

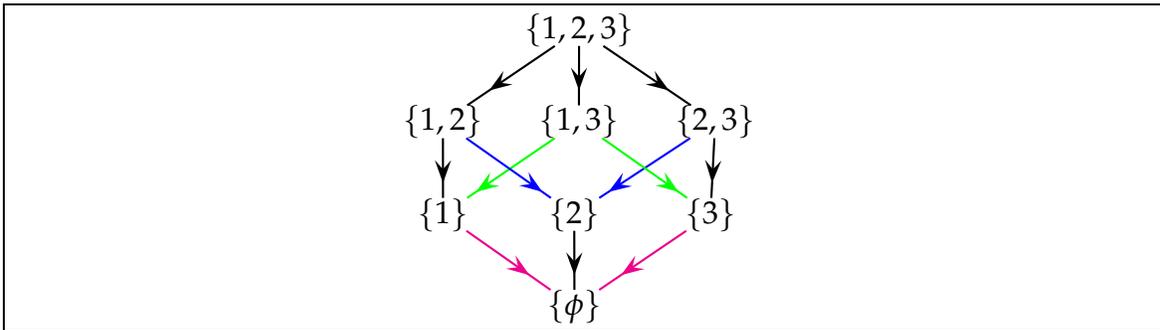
\pagestyle{empty}
\SpecialCoor
\pstree[levelsep=35pt,name=a]{\Toval[name=b]{12~~50~~85}}{
\Toval{6~~10}
\pstree{\Tcircle[name=c]{25}}{\Toval[name=f]{9~~15~~25} \Tcircle[name=g]{36}}
\Toval[name=d]{60~~70}
\pstree{\Toval[name=e]{90~~100~~110}}{\Toval[name=h]{101~~109}}
\uput{.5cm}[90](b){A}
\uput{.5cm}[90](a){B}

```

```

\uput{.37cm}[80](c){C}
\uput{.5cm}[70](d){D}
\uput{.5cm}[90](e){E}
\uput{.5cm}[90](f){F}
\uput{.5cm}[90](g){G}
\uput{.5cm}[70](h){H}

```



- In the above figure black lines are part of the tree and the line between $\{2, 3\}$ and $\{2\}$ is actually line between two nodes from two different *levels*. To identify nodes, 'name' has been assigned like 'TThree', 'Two' etc. By referring those names, line has been drawn with `\ncline{TThree}{Two}`. Colored lines are lines between nodes of different *levels*.

```

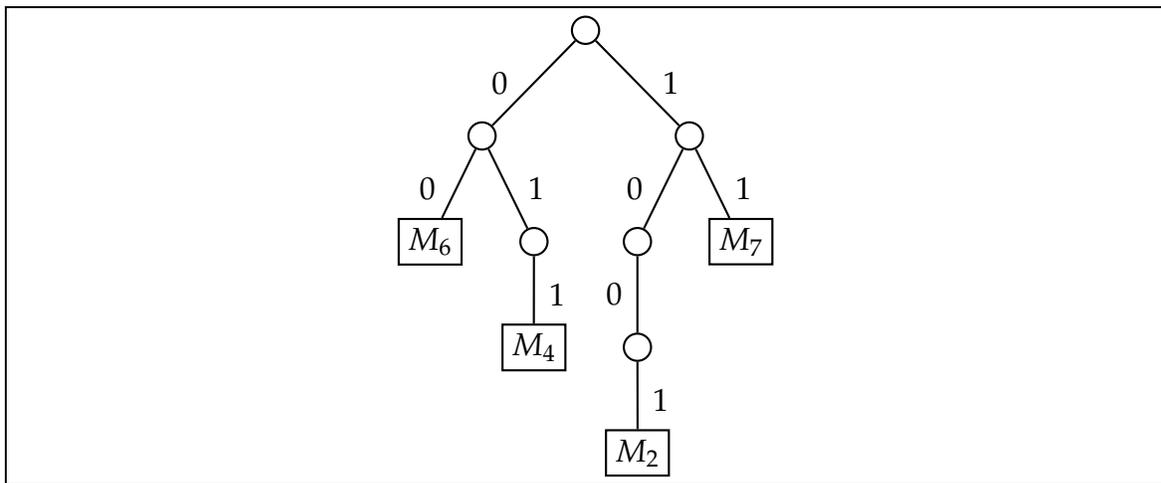
\pstree[levelsep=35pt,ArrowInside=->,ArrowInsidePos=.75,
arrowscale=2,labelsep=14pt]
{\Tr{\$\{1,2,3\}\$}}{
\pstree{\Tr[name=OT]{\$\{1,2\}\$}}{
\Tr[name=One]{\$\{1\}\$}
}
\pstree{\Tr[name=OThree]{\$\{1,3\}\$}}{
\psset{linestyle=none,ArrowInside=none}{
\pstree{\Tr[name=Two]{\$\{2\}\$}}
{\psset{linestyle=solid,ArrowInside=->}\Tr[name=Phi]{\$\{\phi\}\$}}
}
\pstree{\Tr[name=TThree]{\$\{2,3\}\$}}{
\Tr[name=Three]{\$\{3\}\$}
}
\ncline[ArrowInsidePos=.9]{TThree}{Two}

```

```

\ncline[ArrowInsidePos=.9]{0Three}{One}
\ncline[ArrowInsidePos=.9]{0Three}{Three}
\ncline{One}{Phi}
\ncline{Three}{Phi}
\ncline[ArrowInsidePos=.9]{0T}{Two}
}
}

```



- To obtain rectangular or square node one possibility is to use `\psframebox{}` which will form box around the argument enclosed in braces. `\psframebox{Example}` will produce Example in line also.

```

\newcommand{\pbox}[1]{\psframebox{#1}}
$
\pstree[levelsep=40pt]{\Tcircle{~}}{
\pstree[levelsep=40pt]{\Tcircle{~}^0}{
\Tr{\pbox{M_6}}^0\pstree[levelsep=40pt]{\Tcircle{~}_1}{
\Tr{\pbox{M_4}}_1}\pstree[levelsep=40pt]{\Tcircle{~}_1}{
\pstree[levelsep=40pt]{\Tcircle{~}^0}{
\pstree[levelsep=40pt]{\Tcircle{~}^0}{
\Tr{\pbox{M_2}}_1}}\Tr{\pbox{M_7}}_1}
}}
$

```

This is not all. Many commands are not yet discussed, still one can start to get neat figures in Graph Theory with PSTricks. Slowly, as per requirement one can explore new commands through websites. See website [6] for more information on PSTricks. Mailing list of PSTricks is additional place to get good support.

References

- [1] PSTricks: PostScript macros for Generic T_EX, User's Guide, TIMONTHY VAN ZANDT, 1993.
- [2] PsTricks-add: additional Macros for pstricks, HERBERT VOSS, 2005.
- [3] Drawing Trees with PSTricks, DOUG ARNOLD, 2002.
- [4] L^AT_EX Companion, MICHEL GOOSSENS, FRANK MITTELBACH, ALEXANDER SAMARIN, Addison-Wesley, 1994.
- [5] PsTricks mailing list: pstricks@tug.org
- [6] Website <http://tug.org/PSTricks/>

[Journal](#)
[home page](#)
[General](#)
[information](#)
[Submit an](#)
[item](#)
[Download](#)
[style files](#)
[Copyright](#)
[Contact us](#)

THE PracTeX Journal



LaTeXpedia: the future of LaTeX documentation

Lapo Filippo Mori

Abstract

Software documentation is a very important success factor for open source software because it bolsters its diffusion. People who start learning LaTeX and even intermediate users often complain about LaTeX documentation: it is hard to find an updated, complete and well structured resource. This article evaluates advantages and disadvantages of the different sorts of resources for LaTeX documentation available and proposes a new kind of documentation source: a free-content, web-based encyclopedia, LaTeXpedia.

Lapo is a graduate student in Mechanical Engineering at Northwestern University, USA. He started using LaTeX in 2003 while working on his B.S. thesis and has been an enthusiastic user since then. He became a member of GuIT (Italian TUG) in 2003 and an administrative member in 2003. He was among the founders of Ars TeXnica in 2006, the first Italian journal on TeX and LaTeX, and has served as an editor since then. You can reach Lapo at www.lapomori.com.

- [PDF version of paper](#)
- [Comment on this paper](#)
- [Send submission idea to editor](#)

Page generated March 14, 2007 ; [TUG home page](#); [search](#); [contact webmaster](#).

L_AT_EXpedia: the future of L_AT_EX documentation

Lapo Filippo Mori

Email mori@northwestern.edu
Website <http://www.lapomori.com/>
Address Evanston, IL

Abstract Software documentation is a very important success factor for open source software because it bolsters its diffusion. People who start learning L_AT_EX and even intermediate users often complain about L_AT_EX documentation: it is hard to find an updated, complete and well structured resource. This article evaluates advantages and disadvantages of the different sorts of resources for L_AT_EX documentation available and proposes a new kind of documentation source: a free-content, web-based encyclopedia, L_AT_EXpedia.

1 Introduction

Software documentation is very important, especially for open source software. It influences the way software circulates and is further developed, in one word its success. If the best software available has a weak documentation, people are not attracted to it and the number of developers remains small.

I am often asked for the best guide or book to learn L_AT_EX; since I am never able to give a unique answer, there might be a problem about L_AT_EX documentation and, if this is the case, a solution must be found. Being an administrator of the forum of G_UT (Gruppo Utilizzatori Italiani di T_EX) and the author of two guides [33, 34], I matured over the years a personal idea of the weaknesses and the strengths of the information sources available on L_AT_EX but up to now I had never come up with a solution.

In this paper I analyze what is now available and why it does not work to propose a new solution to this relevant problem: L_AT_EXpedia.

2 The problem

2.1 Too many resources

While working on the references of an intermediate guide on writing a thesis with \LaTeX [33, 34], I realized how many resources are available.¹ The beauty of \LaTeX , as other open source softwares, is that many people have written guides about it. For an advanced user this means that, whenever a new problem arises, he can rely on a huge literature and find the solution to the same or a similar problem in a few minutes. For a beginner this means that, besides the difficulties intrinsic in learning a new language, it is also difficult to find out good resources to learn from: an experienced eye can quickly understand if a guide is worth reading but to the beginner they all look the same.

2.2 Obsolescence

Not only are there too many resources, but they are in conflict too. Some of them were written several years ago and the information they provide is obsolete and in contrast with newer guides. \LaTeX , being an open source software, is constantly updated and dozens of new packages are written every year. Since there is little coordination between the people that develop the software, sometimes packages are in conflict between themselves (of course, in this scenario it is very important that guides are regularly updated).

How can a beginner know what the correct answer to a problem is if he cannot identify bad resources?

2.3 The need for a new type of resource

The real problem is that none of the available resources is perfect. To understand what we need and what we must avoid in devising a new kind of resource let's have a look at each type of existing ones.

1. See for example the references reported in par. 2.3.1 and 2.3.2.

2.3.1 Books

Books [3, 11, 18–21, 24, 25, 27–29] are an excellent starting point to learn L^AT_EX because they are meant to be complete. Unfortunately, beginners usually approach the new language reading the guides that are available for free on the net. Books are often bought by intermediate users: after they have learned the basics on a guide and found L^AT_EX suitable for their goals, they decide to study the subject more systematically and buy a book.

A book is usually written by just a few people and, usually being long, requires a big nominal effort (effort per author). As a consequence the time-to-market is at least one year and the number of new releases is usually very limited thus they are all bound to be obsolete before too long.

In theory a single book could become the only resource for L^AT_EX with a solution to all L^AT_EX problems. However, the number of authors must be small in order to avoid coordination problems and hence such a project would require years to be completed and by the time the project reaches the end, most of the material would already be outdated.

Suggestions for L^AT_EXpedia. The new resource type needs:

1. to **be free** so that it will be used by beginners;
2. **a lot of people** working on the project and also to **be released in parts before its completion** (otherwise the time to market is huge and obsolescence will kill most of the work done).

2.3.2 Guides

Among the resources, guides are by far the most numerous. Both generic guides for beginners [2, 16, 23, 37, 41, 43] and guides focused on specific topics are available on the net (tables [6, 8, 15, 35, 45], figures [8, 10, 26, 39, 45], fonts [7, 31], mathematics [1, 44], how to create a PDF file [38, 40], just to cite a few). These are the references that I usually suggest when people ask me; a quick search on the net would, however, provide many more [4, 5, 9, 12, 13, 17, 22, 30, 32, 36, 42, 46, 47].

Common features. As can be easily imagined, being so numerous, guides vary a lot in terms of content, style, length, and level of detail. However, they have some common features. Being shorter than a book, they either cover a lot of topics with few details or cover few topics with greater detail. This means that, unlike a book, it is unlikely that only one guide can cover all the information needed to work with L^AT_EX. On the other hand guides require a smaller nominal time effort than books (this is the main reason why they are so numerous). Like books, however, they have the tendency to become outdated.

Everyone uses L^AT_EX to do different things and it is unlikely that a guide written by a few people can deal with everyone else's problems. When I wrote a guide [33–35], I always emphasized the problems that I had found using L^AT_EX up to that point; some users encounter the same problems but many others have different needs and therefore the bigger the number of people who contribute to the guide, the better. However, it is very hard to manage the writing of many authors: coordination would take more effort than the writing itself.

Suggestions for L^AT_EXpedia. The new resource type needs:

1. to **cover many topics with enough detail** so that it can be used to learn L^AT_EX without the need of other resources.
2. to **be written by many people** so that it is more likely that many topics in different areas are covered.

2.3.3 Forums

Recently, as the use of the internet has increased, online forums have become very popular. As far as regards L^AT_EX, the Usenet's L^AT_EX group², the texhax mailinglist³, and G_UI_T's forum⁴ need to be mentioned.

Bad news. When the number of skilled users is high enough, the answer to a topic is usually very quick and the forum works like a help center. The main problem of forums is that skilled L^AT_EX users usually spend most of their time

2. <http://groups.google.com/group/comp.text.tex>

3. <http://www.tug.org/pipermail/texhax/>

4. <http://guit.sssup.it/phpbb/index.php>

answering to basic questions that often have already been answered in a similar topic. The staff of GJIT's forum has defined some conduct rules (*netiquette*) to change this trend. Despite a good search engine and a database of over two thousand topics, new users continue to ask the same questions and the administrators continue to refer them to existing topics or, even worse, to give the same answers.

When a problem has already been addressed, people are redirected to previous topics. Unfortunately, often the answer is not found in one old topic but is spread over a few of them.

Another problem of forums is that the information can be huge but is not structured. This is in my opinion the major drawback of forums: they cannot be used as a learning tool because the information is so spread that it would take more time to find the topics of interest than to study them.

Good news. On the other hand, a forum is probably the most up-to-date source of information since users can comment on new packages and new features almost in real time: each user can create a new topic in just a few minutes and it is readily available to the whole community.

Another unique feature of forums is that users usually get involved and participate in the production of knowledge. When a user finds a forum helpful, he is likely to help other users in the future. In this way knowledge is not created by a few experienced people but by an ever increasing basis. Of course skilled users, who are usually administrators, make sure that the information provided by the others is correct.

Suggestions for L^AT_EXpedia. The new resource type needs:

1. to **let intermediate users contribute** to the development of the project otherwise the growth is too slow and the experts spend their time in low level activities instead of in coordination.
2. to **be structured** so that it can be used by beginners and intermediate users to learn.

2.3.4 FAQ lists

Numerous FAQ (Frequently Asked Questions) lists are available on the net. They contain a certain number of questions that can arise while using L^AT_EX and the

corresponding solutions. Usually the material is not structured and so it can hardly be used by a beginner who needs to be guided through the new subject. FAQs are usually developed according to the feedback from the users; however the number of people that maintains the list is usually small and so it grows and is updated slowly; the main drawback of a FAQ, as opposed to a forum, is that the user cannot ask directly a question and the feedback process is much slower.

Among \LaTeX FAQs, the “UK List of \TeX Frequently Asked Questions” should be mentioned. It is maintained by the UK TUG and it is available both as HTML⁵ and PDF [14]. It contains a lot of information, it is somewhat structured, and it is updated regularly. Besides these merits, it cannot be used by a beginner and its update process is much slower than a forum.

Suggestions for \LaTeX pedia. The new resource type needs:

1. to **be updated quickly** based on the feedback of its users;
2. to **be well structured** as a book so that it can either be used to find a particular topic (like a FAQ or a forum) or read from the beginning to the end (like a book or a guide);
3. to **be available both in HTML and PDF**.

3 The solution

The solution is an encyclopedia about \LaTeX based on a Wiki engine⁶ to which anyone can contribute; the natural name for the project is then \LaTeX pedia.

The idea of an encyclopedia about \LaTeX is not new. Denis Roegel created “ \LaTeX Navigator: A \LaTeX Encyclopedia”⁷ which is hosted by LORIA.⁸ However here we propose to go much further: we propose to follow the Wikipedia example.

Wikipedia⁹ is a multilingual, Web-based, free-content encyclopedia project that was launched on January 2001 by Larry Sanger and Jimmy Wales. Wikipedia

5. <http://www.tex.ac.uk/cgi-bin/textfaq2html>

6. http://en.wikipedia.org/wiki/Wiki_software

7. <http://tex.loria.fr/index.html>

8. Laboratoire Lorrain de Recherche en Informatique et ses Applications.

9. <http://www.wikipedia.org/>

was thought to be the complement to the expert-written and now defunct Nupedia¹⁰. Nupedia died because it was written only by a few number of experts (many orders of magnitude smaller than the number of participants in the mature stages of Wikipedia) and because the review process of each article was very complicated: these two key aspects made the growth of the encyclopedia be too slow. Wikipedia uses a markup engine¹¹ that allows to write mathematics with the syntax of \LaTeX : this would certainly help authors to provide examples. Possibly this could also be used to automatically convert the whole encyclopedia into a \LaTeX file and from there periodically release a PDF version of \LaTeX pedia.

For sure building an encyclopedia takes a lot of effort, but also the maintenance of a forum requires a lot of work from the administrators. This work would certainly be a better investment if spent in a long lasting, far reaching, and readily available project like \LaTeX pedia.

3.1 Advantages

\LaTeX pedia incorporates the advantages of each resource type and introduces new ones. Let's recall the the ten most interesting ones:

1. **The number of people that contribute is unlimited.** Wikipedia experience teaches us that often users who benefit from the encyclopedia contribute to the project and so the number of people working on it is increasing all the time.
2. **The variety of discussed topics is enormous.** Unlike any other resource (except forums), the number of users contributing to \LaTeX pedia is expected to be huge and each of them can bring up the problems found in his \LaTeX user experience. On the other hand, when the number of authors is small, the content of the resource cannot cover every possible application of \LaTeX .
3. **The work is distributed between users with different skills.** All other resources are developed only by skilled and experienced users, while intermediate and beginners only benefit from their work. With \LaTeX pedia everyone can contribute and hence experts can focus their effort in revising

10. <http://en.wikipedia.org/wiki/Nupedia>

11. [http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style_\(mathematics\)](http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style_(mathematics)) and <http://meta.wikimedia.org/wiki/Help:Formula>

the work done by others and in giving to project a well thought structure. A better distribution of effort makes the project grow much faster.

4. **Access to the project is free.** Despite the fact that L^AT_EXpedia is as complete as a book (or more), it is free and hence it is attractive even for the very beginners.
5. **A powerful search engine helps to find topics.** As in a forum, if a user is looking for a particular topic, he can use a search engine.
6. **The project is well structured.** Like a book or a guide and unlike a forum or a FAQ list, it can be read from the beginning to the end which is particularly useful for beginners. This is also helpful when an intermediate user is looking for the solution to a particular problem: if he does not know the exact topic, he would have hard time to find it with a search engine but the structure of L^AT_EXpedia can help him understand what he is looking for.
7. **The documentation becomes immediately available.** All other resources (except forums) require a certain amount of time between the beginning of the writing and the release of the first edition (from slowest to fastest books, guides, FAQ lists). In this project the material would be immediately available to the public, avoiding the risk of obsolescence. This is possible because the new material is released as soon as is produced.
8. **The project development is continuous.** As opposed to books and guides, L^AT_EXpedia project is supposed to continue as long as L^AT_EX and its packages are developed.
9. **The revision process is quick.** The revision process, done by both expert and intermediate users, is quick because everyone can operate independently whenever he has time to dedicate to the project.
10. **Both HTML and PDF versions of the project are available.** While the HTML version is updated in real time, a PDF version can be released periodically. This follows the successful example of the “UK List of T_EX Frequently Asked Questions”.

3.2 Theory into practice

3.2.1 With a forum

Since many TUGs have their own forums, they could be used to develop L^AT_EXpedia. Let me briefly explain how a forum works and how this could be adapted to develop the encyclopedia.

When a user opens a new topic on a forum and asks a question that has already been answered somewhere on the forum, the administrators or other user give him the link to the previous topic. Often the previous topic partially answers the new question and so new pieces of information are added to the previous topic. On the other hand, if the problem is new, a new discussion begins.

This scheme can be slightly modified to help the creation of an encyclopedia. When a user has a problem, he explains it on the forum. If the solution is available in L^AT_EXpedia, the administrators write the link to the relative page in the encyclopedia and close the topic. On the other hand, if the problem is new or is not completely solved in L^AT_EXpedia, the discussion on the forum goes on. When the user obtains the solution, he creates a page of L^AT_EXpedia with the description of the problem and its solution (or adds the new information to an existing page closely related to the topic). Then the administrators move the topic to a section of the forum with all the solved problems and place there a link to L^AT_EXpedia. In the mean time they also make sure that the page of L^AT_EXpedia contains the correct information and is placed in the right section (i.e. it follows the structure of the project). In this way, the user that obtains help on the forum repays the community by explaining the solution on the encyclopedia and adding examples. L^AT_EXpedia is the only documentation resource to which even beginners could contribute with valuable work.

3.2.2 Without a forum

Although the fact that a forum can speed up the process, L^AT_EXpedia can be developed also without one. When a user cannot find the solution to a problem on the encyclopedia, he creates a new page where he explains the topic. The users who know the solution modify that page by providing an example. The experts verify that the solution is correct and place the page in the right section of L^AT_EXpedia.

3.2.3 Structure of the encyclopedia

Probably the biggest challenge of this project will be to organize the contents in a well ordered structure (item 6 in par. 3.1). To address this issue it is necessary to create a group of editors that will discuss how to organize the topics during the whole life of the project. On this aspect, L^AT_EXpedia is similar to L^AT_EXbooks from which the editors can take inspiration, at least in the early stages. Of course, in the long term, the number of topics covered by the encyclopedia will be by far higher than those covered by any book and the editing process will be even more challenging. However, although the number of topics may be huge, they can be easily subdivided into a few main categories; [14] is a good example on this aspect.

4 Conclusions

One of the appealing features of Wikipedia is its multilingual nature. Articles are continuously translated from one language to another so that the growth rate of pages in each language is always increasing. L^AT_EX is already a multilingual project with hundreds of T_EX User Groups spread all over the world. The development of L^AT_EXpedia would certainly benefit from the collaboration between the TUGs and at the same time the connection between the TUGs would benefit from working on a common project. For all these reasons I would like to finish by inviting the other TUGs to join G_JIT in this new adventure.

Acknowledgments

I would like to thank all the G_JIT members who have contributed to develop this idea and have volunteered to realize it, especially Sandro Allemanni, Daniele Avitabile, Gustavo Cevolani, Massimiliano Dominici, Maurizio Himmelmann, and Emiliano Vavassori.

References

- [1] American Mathematical Society. *AMS-L^AT_EX User's Guide*, 1995. www.dd.chalmers.se/latex/Docs/PDF/amsl.doc.pdf.
- [2] M. Baudoin. *Impara L^AT_EX (...e mettilo da parte)*, 1998. http://www.mat.uniroma1.it/centro-calcolo/manuali/impara_latex.pdf.
- [3] C. Beccari. *L^AT_EX, Guida a un sistema di editoria elettronica*. Hoepli, 1991.
- [4] D. Caon. *L^AT_EX: manuale di base*, 2003. www.guit.sssup.it/downloads/man_base.pdf.
- [5] L. Carusillo. *L^AT_EX*, 2004.
- [6] L. Carusillo. *Le tabelle in L^AT_EX*, 2005. <http://www.webalice.it/lgrsll/pgl/latex/Tabelle.pdf.tar.gz>.
- [7] L. Carusillo. *I caratteri in L^AT_EX*, 2006.
- [8] L. Caucci and M. Spadaccini. *Gestione di Figure e Tabelle con L^AT_EX*, 2005. http://www.guit.sssup.it/downloads/fig_tut.pdf.
- [9] R. Cecchini. *Guida locale all'uso del L^AT_EX*, 1996. www.fi.infn.it/pub/tex/doc/latexu.pdf.
- [10] P. W. Daly. *Graphics and Colour with L^AT_EX*, 1998. <http://tex.loria.fr/graph-pack/grf/grf.pdf>.
- [11] A. Diller. *L^AT_EX Line by Line: Tips and Techniques for Document Processing*. John Wiley & Sons, 1999.
- [12] D. Dowty. *Introductory L^AT_EX Tutorial*, 2002. www.ling.ohio-state.edu/events/lcc/tutorials/intro-latex.pdf.
- [13] Electronic Publishing Unit. *Beginner's L^AT_EX*. UCC Computer Centre, 2001. 158.110.32.35/LATEX/beginlatex.pdf.
- [14] R. Fairbairns. *The UK T_EX FAQ*, 2005. <http://www.ctan.org/tex-archive/help/uk-tex-faq/letterfaq.tex>.

- [15] S. Fear. *Publication quality tables in L^AT_EX*, 2005. <http://www.ctan.org/tex-archive/macros/latex/contrib/booktabs/booktabs.pdf>.
- [16] P. Flynn. *A beginner's introduction to typesetting with L^AT_EX*, 2005. <http://www.tug.org/tex-archive/info/beginlatex/>.
- [17] G. Gonzato. *L^AT_EX for Word Processor Users*, 2003. tug.ctan.org/tex-archive/info/latex4wp/latex4wp.pdf.
- [18] M. Goossens, F. Mittelbach, and A. Samarin. *The L^AT_EX Companion*. Addison-Wesley, 1995.
- [19] M. Goossens, S. Rahtz, and F. Mittelbach. *The L^AT_EX Graphics Companion: Illustrating Documents with T_EX and PostScript*. Addison-Wesley, 1997.
- [20] G. Grätzer. *First Steps in L^AT_EX*. Springer Verlag, 1999.
- [21] G. Grätzer. *Math into L^AT_EX*. Birkhauser, 2000.
- [22] H. J. Greenberg. *A Simplified Introduction to L^AT_EX*, 2006. www.cs.usask.ca/~wew036/latex/simplified-intro/latex.pdf.
- [23] E. Gregorio. *L^AT_EX: Breve guida ai pacchetti di uso più comune*, 2006. profs.sci.univr.it/~gregorio/breveguida.pdf.
- [24] J. Hahn. *L^AT_EX for Everyone: A Reference Guide and Tutorial for Typesetting Documents Using a Computer*. Prentice Hall, 1993.
- [25] D. J. Higham and D. F. Griffiths. *Learning L^AT_EX*. Society for Industrial and Applied Mathematics, 1997.
- [26] I. Hutchinson. *Graphics for Inclusion in Electronic Documents*, 2005. silas.psfc.mit.edu/elec_fig/elec_figures.pdf.
- [27] D. Knuth. *The T_EXbook*. Addison-Wesley, 1992.
- [28] H. Kopka and P. W. Daly. *A Guide to L^AT_EX – Tools and Techniques for Computer Typesetting*. Addison-Wesley, fourth edition, 2003.
- [29] L. Lamport. *L^AT_EX: A Document Preparation System, User's Guide and Reference Manual*. Addison-Wesley, 1994.

- [30] L. Lamport. *L^AT_EX₂ ϵ : The macro package for T_EX*, 1994.
- [31] P. Lehman. *The Font Installation Guide*, 2004. <http://www.ctan.org/tex-archive/info/Typefonts/fontinstallationguide/fontinstallationguide.pdf>.
- [32] T. Love. *Advanced L^AT_EX*, 1997. people.csail.mit.edu/custo/FTP/latex_tutorial.pdf.
- [33] L. F. Mori. Scrivere la tesi di laurea con L^AT_EX₂ ϵ . In *Proceedings del G_JIrmeeting 2005*, pages 69–92, Pisa, Italy, 22nd October 2005. G_JIr (Italian T_EX Users Group). www.guit.sssup.it/guitmeeting/2005/articoli/mori.pdf.
- [34] L. F. Mori. Scrivere la tesi di laurea con L^AT_EX₂ ϵ . *Ars T_EXnica*, (3), March 2007.
- [35] L. F. Mori. Tables in L^AT_EX₂ ϵ : packages and methods. *The PracT_EX Journal*, (4):?, February 2007.
- [36] R. Murri. *Introduzione a L^AT_EX: Scrivere in L^AT_EX con teT_EX ed Emacs su GNU/Linux*, 2000. <http://poisson.phc.unipi.it/~battiston/murripublichtml/lpratico/lpratico.pdf>.
- [37] T. Oetiker, H. Partl, I. Hyna, and E. Schlegl. *The Not So Short Introduction to L^AT_EX₂ ϵ* , 2006. www.ctan.org/tex-archive/info/lshort/english/lshort.pdf.
- [38] M. Pratesi, M. Latini, and M. Antonecchia. *L^AT_EX–PDF–Howto*, 2003. <http://www.telug.it/marco/LaTeX-PDF-HOWTO/LaTeX-PDF-HOWTO-1.1.1.pdf>.
- [39] K. Reckdahl. *Using Imported Graphics in L^AT_EX₂ ϵ* , 1997. <http://www.ctan.org/tex-archive/info/epslatex.pdf>.
- [40] N. Talbot. *Creating a PDF document using PDFL^AT_EX*, 2004. <http://theoval.cmp.uea.ac.uk/~nlct/latex/pdfdoc/pdfdoc-a4.pdf>.
- [41] The Tutorial Team. *On-line Tutorial on L^AT_EX*. Indian T_EX Users Group, 2000. <http://www.tug.org.in/tutorials.html>.
- [42] E. Tonti. *Il L^AT_EX mediante esempi*, 1998. www.dic.univ.trieste.it/perspage/tonti/MANUALI/latex.pdf.

- [43] M. Trettin and J. Fenn. *An essential guide to L^AT_EX₂ ϵ usage: Obsolete commands and packages*, 2006. <http://www.ctan.org/tex-archive/info/l2tabu/english/l2tabuen.pdf>.
- [44] H. Voß. *Mathmode*, 2003. www.ctan.org/tex-archive/info/math/voss/mathmode/Mathmode.pdf.
- [45] H. Voß. *Rotating Text, Tabulars and Images*, 2003. <http://perce.de/LaTeX/rotating.pdf>.
- [46] J. Warbrick. *Essential L^AT_EX*, 2002. www.sci.usq.edu.au/staff/robertsa/LaTeX/Others/essential.pdf.
- [47] D. R. Wilkins. *Getting Started with L^AT_EX*, 1995. www.copernicus.org/guidelines/GSWLaTeX.pdf.

[Journal
home page](#)
[General
information](#)
[Submit an
item](#)
[Download
style files](#)
[Copyright](#)
[Contact us](#)

THE PracTeX Journal



Tables in LaTeX2e: Packages and Methods

Lapo Filippo Mori

Abstract

This article aims to provide the tools to correctly create tables in LaTeX2e. This objective is pursued analyzing the typical problems that users find creating tables and possible solutions; particular emphasis is on the packages to be used. Examples are given for each case.

Lapo Mori is a graduate student in Mechanical Engineering at Northwestern University, USA. He started using LaTeX in 2003 while working on his B.S. thesis and has been an enthusiastic user since then. He became a member of GuIT (Italian TUG) in 2003 and an administrative member in 2003. He was among the founders of *Ars TeXnica* in 2006, the first Italian journal on TeX and LaTeX, and has served as an editor since then. You can reach Lapo at www.lapomori.com.

- [PDF version of paper](#)
- [Comment on this paper](#)
- [Send submission idea to editor](#)

Page generated March 14, 2007 ; [TUG home page](#); [search](#); [contact webmaster](#).

Tables in L^AT_EX_{2 ϵ} : Packages and Methods

Lapo Filippo Mori

Email mori@northwestern.edu
Website <http://www.lapomori.com/>
Address Department of Mechanical Engineering, Northwestern University, Evanston, IL

Abstract This article aims to provide the tools to correctly create tables in L^AT_EX_{2 ϵ} . This objective is pursued analyzing the typical problems that users find creating tables and possible solutions; particular emphasis is on the packages to be used. Examples are given for each case.

Introduction

Tables are one of the most used objects in scientific publications. Besides the regular L^AT_EX commands, a large number of packages can be used to customize the tables and overcome the limitations that the regular commands have. The documentation on tables is very fragmented because every package modifies just an aspect of the table typeset; hence, authors have to look for information on several guides.

This article explains how to use L^AT_EX_{2 ϵ} and its packages to create tables. Each topic is not studied in depth (the reader can refer to the package manual for this) but the author tries to cover as many aspects as possible highlighting the solutions that he likes most. The choice of the contents comes firstly from the author's experience and then from the many discussions on tables on the forum of G_UI_T (Gruppo Utilizzatori Italiani di T_EX).¹

The text presumes that the reader is familiar with the basics of L^AT_EX_{2 ϵ} , which means that he has read at least one of the many free guides available on the net [8, 21] or a book [1, 5, 10–12, 14–18]. Whenever a package is cited, the article

1. The forum of G_UI_T is at <http://www.guit.sssup.it/forum/>.

Table 1: Table not in agreement of the general typeset rules.

D	P_u	u_u	β	G_f
5 in	269.8 lbs	0.000674 in	1.79	0.04089 psi · in
10 in	421.0 lbs	0.001035 in	3.59	"
20 in	640.2 lbs	0.001565 in	7.18	"

does not give a complete description of how it works, which can be found in the package manual,² but analyzes the main options and suggests their use.

This text is focused on the preparation of tables and does not address problems that are connected to it as for example how to position floating objects or how to modify the caption [3]. For these topics, that are in common with all the other floating objects (e.g. figures), the reader can refer to [20].

1 Basics

1.1 General rules

The typeset of tables should be based on the following rules [7]:

1. never use vertical lines;
2. avoid double lines;
3. place the units in the heading of the table (instead of the body);
4. do not use quotation marks to repeat the content of cells.

To better understand the importance of these simple rules, the reader can compare tab. 1 and 2.

2. The majority of \LaTeX packages has a manual which describes how to use it and usually gives examples. The location of the manual can vary between different distributions but, usually, is under the subfolder `/texmf/doc`.

Table 2: Table in agreement of the general typeset rules.

D (in)	P_u (lbs)	u_u (in)	β	G_f (psi · in)
5	269.8	0.000674	1.79	0.04089
10	421.0	0.001035	3.59	0.04089
20	640.2	0.001565	7.18	0.04089

1.1.1 How to input the code

L^AT_EX does not require the columns (i.e. &) to be aligned, however it is advisable to align them in order to make it easier to modify the code. For example, tab. 3 can be obtained with either this code

```
\begin{table}[tp]%
\caption{Maximum load and nominal tension.}
\label{aggiungi}\centering%
\begin{tabular}{clccc}
\toprule%
D$ & & & & \\
(in)& & & & \\
5 & & & & \\
& test 1 & & & \\
& test 2 & & & \\
& test 3 & & & \\
10 & & & & \\
& test 1 & & & \\
& test 2 & & & \\
& test 3 & & & \\
\end{tabular}
\end{table}
```

or this code

```
\begin{table}[tp]%
\caption{Maximum load and nominal tension.}
\label{aggiungi}\centering%
\begin{tabular}{clccc}
\toprule%
D$&&P_u$&&\sigma_N$\\
(in)&&(lbs)&&(psi)\\
5&test 1&285&&38.00\\
&test 2&287&&38.27\\
&test 3&230&&30.67\\
10&test 1&430&&28.67\\
&test 2&433&&28.87\\
&test 3&431&&28.73\\
\end{tabular}
\end{table}
```

Table 3: Maximum load and nominal tension.

D (in)		P_u (lbs)	σ_N (psi)
5	test 1	285	38.00
	test 2	287	38.27
	test 3	230	30.67
10	test 1	430	28.67
	test 2	433	28.87
	test 3	431	28.73

1.2 Basic commands

1.2.1 How to create a table

The basic tools that L^AT_EX offers to create tables and matrices are the `tabular`, `tabular*`, and `array` environments. The environment `array` can be used only in math mode; all these three environments generate a minipage. The syntax is

```
\begin{array}[pos]{cols}
rows
\end{array}
```

```
\begin{tabular}[pos]{cols}
rows
\end{tabular}
```

```
\begin{tabular*}{width}[pos]{cols}
rows
\end{tabular*}
```

where the meaning of the arguments is [17]:

`pos` Vertical position. It can assume the following values:

`t` the line at the *top* is aligned with the text baseline;

`b` the line at the *bottom* is aligned with the text baseline;

`none` when no option is specified, the table is centered to the text baseline;

`width` Can only be used with `tabular*` and it defines the total width of the table.

When used, the argument `cols` must contain somewhere `@{\extracolsep{\fill}}`.

`cols` Defines the alignment and the borders of each column. It can have the following values:

- `l` the column is aligned to the *left*;
- `r` the column is aligned to the *right*;
- `c` the column is *centered*;
- `p{wth}` the column is justified and its width is `wth` (the text is inserted into a parbox of width `wth`);
- `*{num}{form}` the format `form` is repeated `num` times; for example `*{3}{|1|}` is equal to `|1|1|1|`.

Besides the format of the columns, we can also specify what can be inserted between two columns with the following symbols:

- `|` draws a vertical line (not recommended);
- `||` draws a double vertical line (not recommended);
- `@{text}` insert the text `text` in every line of the table between the columns where it appears. This command eliminates the space that is automatically inserted between the columns. If some horizontal space is needed between `text` and the columns, it can be inserted with the command `\hspace{}`. The command `\extraspaces\fill` in a `tabular*` environment extends the space between the columns where it appears in order to let the table have the width defined by the user (see for example table 12 which is obtained with the code at page 14). In order to eliminate the space that is automatically inserted between two columns it is possible to use the empty command `@{}`.

`rows` represents the content of the cells of the table for each row that is ended by the command `\\`. In each row the content of each column is separated by the symbol `&`; each row has the same number of cells (i.e. same number of `&`)³ which must be equal to that declared in the definition `cols`.

`\hline` can be placed in the first row or at the end of a row `\\` and it draws an horizontal line as wide as the entire table.

`\cline{n-m}` draws an horizontal line from the left of column `n` up to the right of the column `m`.

3. The cells might be empty.

Table 4: Example of the standard L^AT_EX multicolumn and cline commands.

Sample	Roughness R_a (nm)
A ring	385
plate	397
B ring	376
plate	390

`\multicolumn{num}{col}{text}` combines the following num columns into a single cell that has the same width, including the spaces between the columns. The argument col must contain a position symbol l, r o c.

An example of these commands is reported in table 4 which is generated by the following code:

```
\begin{tabular}{llc}
\hline%
\multicolumn{2}{c}{Sample} & Roughness  $R_a$ \\
& & (nm)\hline%
A & ring & 385\\cline{2-3}
& plate & 397\\hline
B & ring & 376\\cline{2-3}
& plate & 390\\hline
\end{tabular}
```

Since the command `@{...}` inserts its argument without any space between the columns, it can be used without any argument if we want to eliminate that space. For example in some cases it is useful to eliminate the space on the left of the first column and on the right of the last column; table 5 is obtained with the following code:

```
\begin{tabular}{@{}>{\bfseries}lp{6cm}@{}}
...
\end{tabular}
```

Compare the result with table 9.

1.2.2 Parameters that control the style of a table

There are some parameters that control the style of a table and to which L^AT_EX assigns a default value. These commands can be modified by the user either

Table 5: Table without horizontal space on the left of the first column and on the right of the last one.

Force	Force is a vector quantity defined as the rate of change of the momentum of the body that would be induced by that force acting alone.
Moment of a force	Moment of a force with respect to an origin is defined as the cross product of the position vector (with respect to the same origin) and the force.

globally in the preamble or locally inside an environment.

`\tabcolsep` is half of the width of the space between the columns of the `tabular` and `tabular*` environments.

`\arraycolsep` is half of the width of the space between the columns of the `array` environments.

`\doublerulesep` is the space between double lines (`\hline\hline`).

The command `\setlength` can be used to modify these parameters [1, 5, 8, 10, 11, 14, 15, 17, 18, 21].

1.2.3 Floating tables

Usually the tables are used as floating objects [7, 17, 18, 20]. In this case the `tabular` environment must be inserted in a `table` environment which:

- allows to generate the list of tables with the command `\listoftables` (similar to `\listoffigures` for the figures).
- allows to create a caption taking care of:
 - assigning the right name to the environment; this depends on the language selected with `babel`: in Italian it is “`tabella`” while in English is “`table`”;

- assigning a number to the table.
- allows to assign a `\label` to the table which can be used for references in the text.

To center a floating table,⁴ the `\centering` command must be used instead of the `center` environment because the latter inserts an undesired additional vertical space [6, 22].

For example table 5 is obtained with the following code:

```
\begin{table}[tp]
\caption{...} \label{...} \centering\small
\begin{tabular}{...}
...
\end{tabular}
\end{table}
```

1.3 Multirow cells

As `\multicolumn` allows to have cells on more than one column, the `\multirow` command allows to have cells on more than one row. This command requires the package `multirow`.

`\multirow` can be used in two different ways:

`\multirow{row}*{text}` creates a cell that contains the text `text` and extends on row `rows` and has an undefined width;

`\multirow{row}{larg}*{testo}` creates a cell that contains the text `text` and extends on row `rows` and has an width equal to `larg`;

Table 6 is an example of a table with `multirow` cells and is generated by the following code:

```
\begin{tabular}{c|cc}
\toprule%
\multicolumn{2}{c}{\mathcal{D}} & P_u & \sigma_N \\
\multicolumn{2}{c}{(in)} & (lbs) & (psi) \\
\multirow{3}{*}{5} & test 1 & 285 & \\
38.00 \\ \cmidrule{1}{2-4}
& test 2 & 287 & 38.27 \\ \cmidrule{1}{2-4}
& test 3 & 230 & 30.67 \\ \midrule
\multirow{3}{*}{10} & test 1 & 430 & \\
28.67 \\ \cmidrule{1}{2-4}
```

4. Note that this is valid for every floating object.

Table 6: Example of the standard L^AT_EX multicolumn and multirow commands.

	D (in)	P_u (lbs)	σ_N (psi)
5	test 1	285	38.00
	test 2	287	38.27
	test 3	230	30.67
10	test 1	430	28.67
	test 2	433	28.87
	test 3	431	28.73

```

& test 2          & 433          & 28.87\\cmidrule(1){2-4}
& test 3          & 431          & 28.73\\bottomrule
\end{tabular}

```

Compare it with table 3 that does not use the `\multirow` command.

1.4 Aesthetic improvements

The tabular environment provided by L^AT_EX gives a not satisfactory typographic result because the vertical space between the horizontal lines (obtained with `\hline`) and the text is too thin.

To solve this problem, the `booktabs` and `ctable` packages offer the `\toprule`, `\midrule`, and `\bottomrule` commands to be used instead of `\hline`. In particular `\toprule` and `\bottomrule` must be used for the first and the last line and are thicker than the others, obtained with `\midrule`. `\cmidrule` is to be used instead of `\cline`. These commands modify the thickness of the lines and the vertical space before and after them; in particular `\toprule` and `\bottomrule` are thick and with a wider space below and above respectively, while `\midrule` is thin and with above and below space of the same width. Compare tables 2 and 7.

The default thickness is 0.08 em for `\toprule` and `\bottomrule` and 0.05 em for `\midrule`. To modify the thickness of a line locally (i.e. on a single line) the user can insert the thickness in square brackets after the line command; for example

Table 7: Table obtained with the standard L^AT_EX lines (`hline`).

D (in)	P_u (lbs)	u_u (in)	β	G_f (psi · in)
5	269.8	0.000674	1.79	0.04089
10	421.0	0.001035	3.59	0.04089
20	640.2	0.001565	7.18	0.04089

the command

```
\midrule[0.08em]
```

modifies locally the thickness of a `\midrule`. To modify the thickness of lines globally (i.e. in the whole document), the user can modify the `\heavyrulewidth` (which controls the thick lines, i.e. `\toprule` and `\bottomrule`) and `\lightrulewidth` (which controls the thin lines, i.e. `\midrule`) lengths. For example in this article the thick lines are 0.1 em thick due to the command

```
\setlength{\heavyrulewidth}{0.1em}
```

In this article the line below the heading of the tables has always thickness equal to `\toprule` and `\bottomrule` but it is vertically centered with respect to the row above and below (which is typical of the lines `\midrule`). To obtain this result, a new line type has been defined with the following command

```
\newcommand{\otoprule}{\midrule[\heavyrulewidth]}
```

2 How to align the text inside a cell

L^AT_EX provides four types of alignment (see paragraph 1.2.1): left alignment (`l`), right alignment (`r`), centering (`c`) and justified with assigned width (`p{wth}`). Specific packages provide more types of horizontal alignment for the text.

2.1 The array package

The array package defines new options for column alignment in the array and tabular environments:

`m{wth}` defines a justified column with width `wth` and cells that are vertically centered (as a `\parbox[c]{wth}`);

`b{wth}` defines a justified column with width `wth` and cells that are aligned to the bottom (as a `\parbox[b]{wth}`);

`>{ins}` can be placed before a command `l`, `r`, `c`, `p`, `m` or `b` and inserts `ins` before the content of the cell;

`<{ins}` can be placed after a command `l`, `r`, `c`, `p`, `m` or `b` and inserts `ins` after the content of the cell.

If the content of the table is only mathematics, it can be handy to define it in the preamble of the table instead of using `$. . . $` in each cell. For example it is possible to define new types of column

```
\newcolumntype{A}{>{$}c<{$}}
\newcolumntype{Q}{>{$}l<{$}}
\newcolumntype{V}{>{$}r<{$}}
```

to have column respectively centered, aligned to the left or to the right. If the cell have to contain mathematics in the `displaystyle` format we can define the columns as

```
\newcolumntype{A}{>{$\displaystyle}c<{$}}
\newcolumntype{Q}{>{$\displaystyle}l<{$}}
\newcolumntype{V}{>{$\displaystyle}r<{$}}
```

For example the code

```
\newcolumntype{Q}{>{$\displaystyle}l<{$}}
\newcolumntype{A}{>{$}c<{$}}
\begin{tabular}{QA}
\toprule%
\int \cos x, dx & & \sin x + c & \\ \midrule
\int e^x dx & & e^x + c & \\ \midrule
\int \sec^2 x, dx & & \tan x + c & \\ \bottomrule
\end{tabular}
\end{table}
```

gives table 8.

Further details on the array package can be found on its manual and [13].

2.1.1 How to format the text of a column

The commands `>{ins}` and `<{ins}` can also be used to format certain columns: it is possible to use L^AT_EX commands `\upshape`, `\itshape`, `\slshape`, `\scshape`, `\mdseries`, `\bfseries`, `\rmfamily`, `\sffamily`, and `\ttfamily` [18].

For example the code

Table 8: Table with mathematics obtained with the array package.

$\int \cos x dx$	$\sin x + c$
$\int e^x dx$	$e^x + c$
$\int \sec^2 x dx$	$\tan x + c$

Table 9: Table with automatic boldface text on one column (array package).

Force	Force is a vector quantity defined as the rate of change of the momentum of the body that would be induced by that force acting alone.
Moment of a force	Moment of a force with respect to an origin is defined as the cross product of the position vector (with respect to the same origin) and the force.

```
\begin{tabular}{>{\bfseries}l p{6cm}}
\toprule
Force & Force is a vector quantity defined as the rate of
change of the momentum of the body that would be induced by that
force acting alone.\\
\midrule
Moment of a force & Moment of a force
with respect to an origin is defined as the cross product of the
position vector (with respect to the same origin) and the
force.\\
\bottomrule
\end{tabular}
```

produces table 9 where the text in the first column is boldface.

Table 10: Use of the array package to format the text of a row.

Quantity	Symbol	Unit	Value
Stiffness in z direction	k_z	N/m	2276
Stiffness in r direction	k_r	N/m	3414
Weight of the body	P	N	35

2.1.2 How to format the text of a row

There are no commands or packages to format rows but it is possible to define a new command [6] which uses the tools of the array package discussed above. The following definitions are added to the preamble

```
\newcolumntype{+}{>{\global\let\currentrowstyle\relax}}
\newcolumntype{^}{>{\currentrowstyle}}
\newcommand{\rowstyle}[1]{\gdef\currentrowstyle{#1}%
#1\ignorespaces
}
```

and then + must be added on the left of the first column and ^ on the right of every other column in the column definition of tabular. This defines a new command

```
\rowstyle{...}
```

which can be inserted at the beginning of a row to format its cells. For example the code

```
\begin{tabular}{+l^c^c}
\toprule\rowstyle{\bfseries}%
Quantity & Symbol & Unit & Value \\ \otoprule%
...
\end{tabular}
```

produces table 10 where the first row is boldface.

2.2 The tabularx package

The tabularx package requires the same arguments of tabular* but, in order to let the table have the width specified by the user, it modifies the width of certain columns instead of the space between columns. The columns that can be stretched are identified by the alignment command X. This package requires the array package. For example the commands

Table 11: Table obtained with the tabularx package.

Force	Force is a vector quantity defined as the rate of change of the momentum of the body that would be induced by that force acting alone.
Moment of a force	Moment of a force with respect to an origin is defined as the cross product of the position vector (with respect to the same origin) and the force.

```
\begin{tabularx}{\textwidth}{>{\bfseries}lX}
\toprule
Force & Force is a vector quantity defined as the rate of change of
the momentum of the body that would be induced by that
force acting alone.\\
\midrule
Moment of a force & Moment of a force with respect to an origin is
defined as the cross product of the position vector (with respect to
the same origin) and the
force.\\
\bottomrule
\end{tabularx}
```

produce table 11; notice that in the similar table 9 the length of the second column has been defined by the user, while in table 11 the user defines the total width of the table (in this case `\textwidth`) and then the program computes automatically the width of the second column. Compare the result with table 12 obtained with the `tabular*` environment: in this case the space between the columns is stretched instead of the width of the column. To obtain the table, the following code was used

```
\begin{tabular*}{\textwidth}[tb]{%
  >{\bfseries}l@{\extracolsep{\fill}}p{6cm}}
...
\end{tabular*}
```

The X column are by default of type `p{wth}` (i.e. justified) and the width `wth` is determined automatically by the program. It is anyway possible to define new column types with a different horizontal alignment with the command `\newcolumntype`. For example

```
\newcolumntype{Y}{>{\raggedright\arraybackslash}X}
```

defines columns aligned to the left,

```
\newcolumntype{W}{>{\raggedleft\arraybackslash}X}
```

Table 12: Table obtained with the `tabular*` environment.

Force	Force is a vector quantity defined as the rate of change of the momentum of the body that would be induced by that force acting alone.
Moment of a force	Moment of a force with respect to an origin is defined as the cross product of the position vector (with respect to the same origin) and the force.

defines columns aligned to the right,

```
\newcolumntype{Z}{>{\centering\arraybackslash}X}
```

defines centered columns. For example the code

```
\begin{tabularx}{\textwidth}{YXZW}
\toprule
Cell with text aligned to the left & 1 & 2 & 3\\midrule
4 & Cell with justified text & 5 & 6\\midrule
7 & 8 & Cell with centered text & 9\\midrule
10 & 11 & 12 & Cell with text aligned to the right\\bottomrule
\end{tabularx}
```

produces table 13 which is as wide as the text (`\textwidth`).

The X column are by default of type `p{wth}` corresponding to a `\parbox[t]` which means that vertically the cells are aligned to the top. It is anyway possible to define new column types with a different vertical alignment with the command `\tabularxcolumn`. For example

```
\renewcommand{\tabularxcolumn}[1]{>\arraybackslash}m{#1}
```

defines columns vertically aligned to the center,

```
\renewcommand{\tabularxcolumn}[1]{>\arraybackslash}b{#1}
```

defines columns vertically aligned to the bottom. The code

```
\renewcommand{\tabularxcolumn}[1]{>\arraybackslash}m{#1}
\begin{tabularx}{\textwidth}{YXZW}
\toprule Cell with text aligned to the left & 1 & 2 & 3\\midrule 4 &
Cell with justified text & 5 & 6\\midrule 7 & 8 & Cell with centered
```

Table 13: Table obtained with the tabularx package and redefining new types of horizontal alignment for the columns.

Cell with text aligned to the left	1	2	3
4	Cell with justified text	5	6
7	8	Cell with centered text	9
10	11	12	Cell with text aligned to the right

```
text & 9\\midrule 10 & 11 & 12 & Cell with text aligned to the
right\\bottomrule
\\end{tabularx}
```

produces table 14(a), while the same code with the command

```
\\renewcommand{\\tabularxcolumn}[1]{>{\\arraybackslash}b{#1}}
```

produces table 14(b).

The tabularx package is also useful when some columns must have the same width but it is not required to define it directly. In fact, if several X columns are used in the same table, their width is the same. For example table 15 is obtained with the following code

```
\\newcolumnstype{K}{>{\\centering\\arraybackslash}X<{}}
\\begin{tabularx}{.7\\textwidth}{*{7}{K}}
...
\\end{tabularx}
```

By default the width of the columns is automatically computed by the program, anyway tabularx allows to assign the width of some columns. This can be done only if the sum of the widths X remains the same. If for example a column is supposed to be half as wide as another one and the table has only two X columns (these two), the first one must be $\frac{2}{3} \cdot x$ wide and the second one $\frac{4}{3} \cdot x$; in this way they are in the right proportion and their sum is equal to $2x$ (which is the width of two X columns); to obtain this result the following command can be used

Table 14: Tables obtained with the `tabularx` package and defining the vertical alignment of the cells in order to have them centered (a) and aligned to the bottom (b).

(a) Vertically centered cells

Cell with text aligned to the left	1	2	3
4	Cell with justified text	5	6
7	8	Cell with centered text	9
10	11	12	Cell with text aligned to the right

(b) Cells aligned to the bottom

Cell with text aligned to the left	1	2	3
4	Cell with justified text	5	6
7	8	Cell with centered text	9
10	11	12	Cell with text aligned to the right

Table 15: Table obtained with the `tabularx` package to have columns with the same width.

α	$\sin \alpha$	$\cos \alpha$	$\tan \alpha$	$\csc \alpha$	$\sec \alpha$	$\cot \alpha$
-2π	0	1	0	∞	1	∞
$-7\pi/4$	$\sqrt{2}/2$	$\sqrt{2}/2$	1	$\sqrt{2}$	$\sqrt{2}$	1
$-3\pi/2$	1	0	∞	1	∞	0
$-5\pi/4$	$\sqrt{2}/2$	$-\sqrt{2}/2$	-1	$\sqrt{2}$	$-\sqrt{2}$	-1
$-\pi$	0	-1	0	∞	-1	∞
$-3\pi/4$	$-\sqrt{2}/2$	$-\sqrt{2}/2$	1	$-\sqrt{2}$	$-\sqrt{2}$	1
$-\pi/2$	-1	0	∞	-1	∞	0
$-\pi/4$	$-\sqrt{2}/2$	$\sqrt{2}/2$	-1	$-\sqrt{2}$	$\sqrt{2}$	-1
0	0	1	0	∞	1	∞

Table 16: Table obtained with the `tabularx` package imposing the proportion on the width of columns: the first column is half as wide as the second one.

Force	Force is a vector quantity defined as the rate of change of the momentum of the body that would be induced by that force acting alone.
Moment of a force	Moment of a force with respect to an origin is defined as the cross product of the position vector (with respect to the same origin) and the force.

```
<{\hspace=0.66\hspace}X>{\hspace=1.34\hspace}X
```

Table 16 is obtained adding this line to the code of table 11.

2.3 The tabulary package

The `tabulary` package provides results similar to those obtained with the `tabularx` package in table 13. Instead of redefining the alignment of the column with the command `\newcolumntype` as shown in par. 2.2, the `tabulary` package provides the commands R (aligned to the right), C (centered), L (aligned to the left) e J (justified).

For example table 13 can be obtained also with the code

```
\begin{tabulary}{\textwidth}{LJCR}
\toprule
Cell with text aligned to the left & 1 & 2 & 3\\midrule
4 & Cell with justified text & 5 & 6\\midrule
7 & 8 & Cell with centered text & 9\\midrule
10 & 11 & 12 & Cell with text aligned to the right\\bottomrule
\end{tabulary}
```

2.4 How to break lines manually

If the content of a cell is too long, it is necessary to break it into more lines. The packages presented above (`tabularx` and `tabulary`) solve this problem automatically and then they are preferred. In the cases in which it is not possible to use them it is still possible to break the lines manually. If, for example, we want to break

Table 17: Table with a column of defined width and alignment to the right.

Force	Force is a vector quantity defined as the rate of change of the momentum of the body that would be induced by that force acting alone.
Moment of a force	Moment of a force with respect to an origin is defined as the cross product of the position vector (with respect to the same origin) and the force.

the lines only in certain cells of a column, it is possible to insert into these cells a `parbox` or a `minipage`.

The use of these commands is also convenient when we want to assign the width of a non justified column.⁵ If for example we want a column 5 cm wide and with text aligned to the right, it is possible to insert all the cells of that column into `minipage` environments with the command

```
<{\bfseries}l>{\begin{minipage}[t]{5cm}%
\raggedleft\arraybackslash}%
l<{\end{minipage}\arraybackslash}}
```

With this command it is possible to obtain table 17 with the code used for table 11.

2.5 How to change the horizontal alignment of certain cells

To modify the alignment of certain cells, it is possible to insert the text into a *box* (for example with the command `\makebox`). For example table 18 is obtained with the following code:

```
\begin{tabular}{lp{3cm}}
\hline
Text & to the left\\\hline
```

5. If the column has to be justified we can just use a `p{wth}`.

Table 18: Table with cell with different alignment in the same column.

Text	to the left
Text	at the center
Text	to the right

```
Text & \makebox[3cm][c]{at the center}\\\hline
Text & \makebox[3cm][r]{to the right}\\\hline
\end{tabular}
```

2.6 How to align the numbers to the decimal point

2.6.1 Standard commands

The command `@{.}` can be used to align the numbers to the decimal point if the integral part of the number is on one column and the fractional part on the next one. For example table 20 can be obtained with the following code:

```
\begin{tabular}{c r @{.} l}
\toprule
Expression & \multicolumn{2}{c}{Value} & \\
\midrule
 $\pi$  & 3 & 1416 & \\
 $\pi^{\pi}$  & 36 & 46 & \\
 $\pi^{\pi^{\pi}}$  & 80662 & 7 & \\
\bottomrule
\end{tabular}
```

2.6.2 The dcolumn package

If we do not want to break the fractional and the integral part in two columns, the dcolumn package provides a new type of column

```
D{sep-in}{sep-out}{before.after}
```

The first argument (`sep-in`) is the symbol used in the `.tex` document to separate the integral and the fractional part (usually the decimal point `.` or the decimal comma `,`), the second argument (`sep-out`) is the symbol that we want in the output, the third is the number of digits on the left (`before`) and on the right (`after`) of this symbol. The numbers are aligned to the decimal point and, in case that the third argument is negative, the decimal point is aligned to the center of the column. If the columns have a heading, it must be inserted into the command

Table 19: Table with alignment to the decimal point.

Expression	Value
π	3.1416
π^π	36.46
π^{π^π}	80662.7

`\multicolumn{1}{c}{...}`. For example table 20 can be obtained also with the following code

```
\begin{tabular}{cD{,}{.}{5.4}}
\toprule
Espressione & \multicolumn{1}{c}{Valore} \\ \otoprule
$\pi$ & 3,1416 \\ \midrule
$\pi^\pi$ & 36,46 \\ \midrule
$\pi^{\pi^\pi}$ & 80662,7 \\ \bottomrule
\end{tabular}
```

In order to avoid inserting all the three parameters for every column, it is possible to define a new type of column, for example with the code

```
\newcolumntype{d}[1]{D{,}{.}{#1}}
```

where `d` has only one argument which defines the number of decimal digits.

Gregorio [13] shows an interesting use of `\newcolumntype` to let the decimal separator be dependent on the language selected with `babel`: comma (,) for the italian option and point (.) for the english option.

2.6.3 The rcol package

`rcol` is a more advanced package that allows to align numbers to the decimal point. This package has the same capabilities of `dcolumn` but in addition can round the numbers and even add zeros in order to have all the cells with the same number of decimal digits. The columns to be aligned to the point are defined with the code

```
R[sep-in][sep-out]{before}{after}
```

where the arguments have the same meaning of those for `dcolumn`. The argument `after` in this case defines the number of digits to keep during the approximation; for example if this parameter is equal to 3, 3.1416 becomes 3.142 and 80662.7

Table 20: Table with alignment to the decimal point and automatic approximation.

Expression	Value
π	3.142
π^π	36.460
π^{π^π}	80662.700

becomes 80662.700. These parameters assign to the number a precision of 10^{-n} (where n is the value of after) and thus it can also assume negative values; for example if this parameter is equal to -1 , 3.1416 becomes 0 and 80662.7 becomes 80660.

For example the code

```
\begin{tabular}{cR[,][.]{5}{3}}
\toprule
Expression          & \multicolumn{1}{c}{Value} & \\ \otoprule
$\pi$                & 3,1416                    & \\ \midrule
$\pi^{\pi}$         & 36,46                     & \\ \midrule
$\pi^{\pi^{\pi}}$   & 80662,7                   & \\ \bottomrule
\end{tabular}
```

produces table 20.

3 Large tables

Sometimes, especially in the appendix, the size of a table is bigger than a page in width, height, or both. In case the table is too long, it is possible to

- reduce the size of the font (see par. 3.1);
- scale it down (see par. 3.3);
- break it over several pages (see par. 3.4).

In case the table is too wide, it is possible to

- reduce the size of the font (see par. 3.1);
- rotate it (see par. 3.2);

Table 21: Table with font size smaller (`footnotesize`) than the rest of the text (`normalsize`).

Force	Force is a vector quantity defined as the rate of change of the momentum of the body that would be induced by that force acting alone.
Moment of a force	Moment of a force with respect to an origin is defined as the cross product of the position vector (with respect to the same origin) and the force.

- scale it down (see par. 3.3);
- break it over several pages (see par. 3.4).

Obviously in each case it is possible to use more solutions at once.

3.1 How to reduce the size of the font

Reducing the size of the font in a table just requires the font size command to be inside the table environment (or similar). For example the code

```
\begin{table}[tp]\footnotesize
\caption{...} \label{...} \centering
\begin{tabular}{...}
...
\end{tabular}
\end{table}
```

produces table 21; compare the result with table 9.

In case the table is non floating (i.e. it is not inside the table environment or similar) the font size command must be outside the tabular environment and then, at the end of the table, the font size used in the rest of the document must be reestablished. On the contrary, when the font size command is inside a table environment (or similar) it affects just the text inside this environment and it is not necessary to reestablish the normal font size at the end.

3.2 Rotating tables

The methods to rotate tables are also valid for every other floating object; [23] is an excellent reference on this topic.

3.2.1 The graphicx package

The graphicx package provides the command

```
\rotatebox[options]{angle}{object}
```

that allows to rotate every object (or box) and can also be used for tables; `options` is an optional argument to modify the origin of the rotation (see [23]), `angle` is the angle of the rotation (for tables is typically 90°) and `object` is the object to be rotated (in this case `tabular`).

3.2.2 The rotating package

The rotating package provides commands that rotate specifically floating objects and can be used also with tables. The first thing to do is to define the positive direction for rotations with the commands

`clockwise` (default) measures the angle clockwise to be compatible with [10], where the positive angles are always clockwise.

`counterclockwise` as the previous but with positive angles counterclockwise.

The rotating package provides the `rotate`, `turn`, `sideways`, and `sidewaystable` environments.⁶

The rotate environment. This environment does not leave vertical space for the rotated object thus it cannot be used for tables because they would overwrite the preceding text.

```
\begin{rotate}{angle}  
...  
\end{rotate}
```

6. The package also provides `sidewaysfigure` which is the analogous of `sidewaystable` for figures.

The turn environment. This environment, unlike `rotate`, adds a vertical space for the rotated object and thus behaves like `\rotatebox` (see par. 3.2.1).

```
\begin{turn}{<degree>}
...
\end{turn}
```

The sideways environment. This environment is a particular turn environment: it rotates the object by 90° and leaves vertical space for it; in this case the user does not have to indicate the angle. Since wide tables must be rotated by 90° , this is the most suitable environment.

```
\begin{sideways}
\begin{tabular}
...
\end{tabular}
\end{sideways}
```

The sidewaysstable environment. This environment does the same rotation sideways does but it must be used with floating tables for which it substitutes the `table` environment. The syntax is

```
\begin{sidewaysstable}
\caption{...}
\begin{tabular}
...
\end{tabular}
\end{sidewaysstable}
```

Unlike the preceding environments, `sidewaysstable` uses a whole page. Since it is a floating environment, if necessary, L^AT_EX fills the preceding page with the text that comes after the table.

For example the code

```
\begin{sidewaysstable}[p]\small
\caption{...} \label{...} \centering
\renewcommand{\tabularxcolumn}[1]{>{\arraybackslash}m{#1}}
\newcolumnntype{W}{>{\centering\arraybackslash}X}
\begin{tabularx}{\textheight}{lccWWWWW}
...
\end{tabularx}
\end{sidewaysstable}
```

produces the table at page 1; in this example, besides `sidewaysstable`, also the `tabularx` package is used to define a `W` type column (horizontally centered text),

Table 1: Comparison of performance improvement of blast-resistant structures.

Quantity	Symbol	Unit	Square honeycomb sandwich			Pyramidal honeycomb sandwich
			i	ii	iii	
Core relative density	ρ_c	%	4.0	4.0	4.0	4.1
Mass per area	m_A	kg/m ²	14.0	14.0	14.0	13.78
Experimental results	\dot{I}	—	1.234	0.931	0.939	0.925
	δ_{max}/L	—	0.391	0.299	0.297	0.299
Given Impulse	$(\delta_{max}/L)_N$	—	0.391	0.278	0.283	0.285
	I_N	%	28.9	27.6	28.7	27.2
Given deflection	I_N	—	1.240	1.217	1.236	1.210
	I_N	%	40.6	38.0	40.2	37.1

9

Figure 1: Example of the sideways stable environment.

the text is vertically centered, and the font size is reduced with the command `\small`.

3.2.3 The lscape package

The lscape package defines the landscape environment which rotates its content by 90°. This environment has been designed to work with the longtable package and so its content can even span over several pages.

3.3 Scaling tables

The `graphicx` package provides two commands that can be used to scale tables: `\scalebox` and `\resizebox`.

The command

```
\scalebox{h-scale}[v-scale]{argument}
```

scales `argument` (in this case a table) by `h-scale` horizontally and `v-scale` vertically; for tables the same scale factor must be used horizontally and vertically and this can be done assigning one parameter between `h-scale` and `v-scale` and assigning `!` to the other one.

The command

```
\resizebox{width}{height}{argument}
```

scales `argument` (in this case a table) in order to have a width equal to `width` and a height equal to `height`; in this case is also possible to use `!` to maintain the same aspect ratio of the table. Typically these commands are used to scale down large tables in order to let them be as wide as the text; this can be done for example with the commands

```
\begin{table}[tb]
\caption{...} \label{...} \centering
\resizebox{\textwidth}{!}{%
\begin{tabular}{...}
...
\end{tabular}}
\end{table}
```

For example tables 14(a) and 14(b) are obtained with the same code of 13 but are scaled with `\resizebox` so that their width is 65% of that of the text:

```
\resizebox{0.65\textwidth}{!}{...}
```

3.4 Tables over several pages

The `tabular` environment must always be contained in a single page: if bigger, the parts that exceed the page boundaries are cut and an `Overfull vbox` error message is displayed. Several packages can be used to overcome this limitation and break tables over several pages.

3.4.1 The supertabular package

The supertabular package provides the homonymic supertabular environment that behaves like the regular tabular but, when compiling, it checks the length of the table at every row: when the length exceeds `textheight` the package automatically inserts the optional argument `tabletail` and the `\end{tabular}` command; then it starts a new table on the next page, after the optional argument `tablehead`. The `tabletail` argument can be used to place at the bottom of every page in which the table is broken “continues on the next page” while `tablehead` for “continues from the previous page”. The supertabular package treats the part of the table that is in the new page independently thus the width of the column can vary across different pages unless the user uses fixed width columns.

3.4.2 The xtab package

The xtab package has functions very similar to those of supertabular but it corrects some defects thus it is recommended. It also allows to have a different caption for the last page (`\tablelasthead`).

3.4.3 The longtable package

The longtable package builds pieces of the table during the first compilation and then uses the information that it wrote in the `.aux` file to decide where to break the table during the following compilations. In this way the columns have the same width in every page. The package requires several compilation (as opposed to supertabular) and it is incompatible with many other packages.

3.4.4 The lscape package

The lscape package is meant to rotate by 90° the supertabular environment for wide and long tables.

The reader is suggested to read the manual for all the packages reported above since they report several examples and describe all the options. As regards longtable, [13] is also suggested.

4 Tables and colors

A colored background can be very useful when trying to highlight a part of a table. The `colortbl` package can be used to color the background of cells, rows or columns of `tabular` environments; it can also be used to color the lines (for example `\hline`) but this topic is not analyzed since scientific document should never use colored lines. The `colortbl` requires the `color` and `array` packages. The `xcolor` [4] package can be used instead of `color` but the examples reported below use only the latter.

4.1 How to color columns

The package provides the command `\columncolor` that must be used only inside the command `>{...}` (see par. 2.1). The syntax is

```
\columncolor[clrmodel]{color}[left overhang][right overhang]
```

The first two arguments are standard commands of the `color` package [2], used as in the command `\color`, while the last two arguments are lengths.

`clrmodel` declares the color model to be used; `rgb`, `cmymk`, `gray`, and `named` are available.

`color` declares the color to be used; this definition depends on the color model chosen. `rgb` (Red Green Blue) requires a list of three numbers between 0 and 1 and separated by commas; each of them gives the corresponding color component (red, green, and blue). `cmymk` (Cyan Magenta Yellow Black) requires a list of four numbers between 0 and 1 and separated by commas each of them gives the corresponding color component (cyan, magenta, yellow, and black). `gray` requires just a number between 0 and 1 of the corresponding gray level; in scientific tables grays are preferred to colors thus this command is very convenient. `named` allows to use colors based on their name; the name, if not already defined by default, must be defined by the user with the command

```
\definecolor{name}{clrmodel}{color}
```

where `name` is the color name assigned by the user, `clrmodel` and `color` have the same meaning they have in the command `\columncolor`.

Table 22: Table with a column with colored background obtained with the colortbl package.

D (in)	P_u (lbs)	u_u (in)	β	G_f (psi·in)
5	269.8	0.000674	1.79	0.04089
10	421.0	0.001035	3.59	0.04089
20	640.2	0.001565	7.18	0.04089

`left overhang` and `right overhang` specifies how much the colored background is separated from the text on the left and on the right respectively. The use of these commands is not recommended since, if they are not equal to zero, the colored background does not fill all the space reserved to `\tabcolsep`.

The `\newcolumntype` command (see par. 2.1) can be used to redefine new column types with the desired background color. For example the command

```
\newcolumntype{K}{\columncolor[gray]{0.8}\raggedright}
```

allows to obtain table 23 with the code

```
\begin{tabular}{ccccU}
\hline%
$D$ & $P_u$ & $u_u$ & $\beta$ & $G_f$ \\
(in) & (lbs) & (in) & & (\text{psi}\cdot\text{in}) \\
5 & 269.8 & 0.000674 & 1.79 & 0.04089 \\
10 & 421.0 & 0.001035 & 3.59 & 0.04089 \\
20 & 640.2 & 0.001565 & 7.18 & 0.04089 \\
\end{tabular}
```

Compare the result with table 7. Note that the commands provided by the ctable package (see par. 1.4) cannot be used for the horizontal lines: these commands leave vertical space above and below every cell and so the background color would be interrupted; see for example table 22.

4.2 How to color rows

The `\rowcolor` command can be used to color rows: it has the same syntax of `\columncolor` and must be placed at the beginning of the row. If both `\columncolor` and `\rowcolor` are used in a table, the latter prevails. For example table 24 is obtained with the same code of table 2 just adding the command

```
\rowcolor[gray]{.8}
```

Table 23: Table with a column with colored background (ctable package).

D (in)	P_u (lbs)	u_u (in)	β	G_f (psi·in)
5	269.8	0.000674	1.79	0.04089
10	421.0	0.001035	3.59	0.04089
20	640.2	0.001565	7.18	0.04089

Table 24: Table with a row with colored background (\rowcolor command).

D (in)	P_u (lbs)	u_u (in)	β	G_f (psi·in)
5	269.8	0.000674	1.79	0.04089
10	421.0	0.001035	3.59	0.04089
20	640.2	0.001565	7.18	0.04089

Sometimes the background color is not used to highlight certain rows but to separate them (as horizontal lines do). In this case the xcolor package is very handy because it provides commands to automatically color all the odd and even lines of the table. The table option of the package must be loaded in the preamble

```
\usepackage[table]{xcolor}
```

and then the following command must be used right before the tabular environment

```
\rowcolors{row}{odd-row color}{even-row color}
```

where

row is the number of the first row to be colored,

odd-row color is the color to use with odd rows (if this parameter is blank it means no color),

even-row color is the color to use with even rows (if this parameter is blank it means no color).

Table 25: Table with colored odd rows obtained with the `xcolor` package.

D (in)	P_u (lbs)	u_u (in)	β	G_f (psi · in)
5	269.8	0.000674	1.79	0.04089
10	421.0	0.001035	3.59	0.04089
20	640.2	0.001565	7.18	0.04089

Table 26: Table with a colored cell obtained with the `\cellcolor` command.

Force	Force is a vector quantity defined as the rate of change of the momentum of the body that would be induced by that force acting alone.
Moment of a force	Moment of a force with respect to an origin is defined as the cross product of the position vector (with respect to the same origin) and the force.

For example table 25 is exactly the same as table 7 but, instead of horizontal lines (`\hrule`), it uses gray background for odd rows. The code that is used to create this table is

```
\rowcolors{2}{gray!35}{}
\begin{tabular}{ccccc}
...
\end{tabular}
```

4.3 How to color single cells

The command `\cellcolor` can be used to add a background color to single cells: it works as `\columncolor` and `\rowcolor`, and prevails on both these. `\cellcolor` can be placed everywhere in the cells to which it has to be applied.

For example table 26 is obtained just adding the following command to the code of table 21,

```
\cellcolor[gray]{0.8}
```

5 Automatic generation of tables

Often the tables that appear in \LaTeX documents contain data generated by other programs, for example statistical or mathematical software. \LaTeX tables can be easily and automatically generated by other programs since their syntax is very simple. Typically the cells of a row are just separated by `&` and each row is ended by `\` (see par. 1.2.1).

5.1 Stata

An interesting application of this concept has been presented for the statistical program Stata.⁷ Gini [9] explains how to use Stata to automatically generate \LaTeX tables with colored rows (see par. 4) and tables in which some cells are highlighted with boldface (for example those that contains the maximum or the minimum of a series of data).

5.2 Spreadsheets

Many freeware softwares, as Excel2LaTeX⁸ and Spreadsheet2LaTeX⁹, that automatically generates \LaTeX code ready to be compiled starting from a formatted¹⁰ table in a spreadsheet. In general it is quite easy to program macros to make \LaTeX code starting from a generic spreadsheet.

6 Specialities

6.1 Line styles

As said in par. 4, the `colortbl` package can be used to modify the color of horizontal and vertical lines; anyway this should be avoided in scientific documents and so the details are left to the reader.

7. Stata is a registered trade mark of StataCorp LP.

8. Excel2LaTeX is available at <ftp://ftp.tex.ac.uk/tex-archive/support/excel2latex/xl2latex.zip>.

9. Spreadsheet2LaTeX is available at <http://pegasus.rutgers.edu/~elflord/unix/software/spreadsheet2latex/>.

10. For example with colors, horizontal and vertical lines.

The `arydshln` package allows to use dashed lines. The commands `\hdashline` and `\cdashline` corresponds respectively to `\hline` and `\cline`.

6.2 Notes inside a table

The standard L^AT_EX command for notes (`\footnote`) does not work inside tables because the `tabular` environment does not allow the command to place text at the bottom of the page. Many solutions and packages can be used to overcome this problem; the best ones are reported below.

In general, two cases can arise:

- a) the note must follow the numeration of the other notes in the text;
- b) the numeration of the notes inside tables must be different (for example with letters).

The package `footnote` provides a solution for case (a). This package defines the `savenotes` environment that allows to insert notes even inside a `tabular` environment and place them at the bottom of the page. If we want to allow the presence of notes in all the tables of the document, we just need to add to the preamble the command

```
\makesavenoteenv{tabular}
```

For example table 27 is obtained with the following code

```
\begin{savenotes}
\begin{table}[..] \caption{...} \label{...} \centering
\begin{tabular}{...} \toprule
...
Moment of a force & Moment of a force with respect to an origin
is defined as the cross product of the position vector\footnote{With
respect to the same origin.} and the force.\bottomrule
\end{tabular}
\end{table}
\end{savenotes}
```

The `ctable` package, on the other hand, offers a solution to case (b). The `ctable` environment, provided by the homonymic package, offers the command `\tmark` which places the symbol of the note (in this case the default numbering style is with lowercase letters) and the command `\tnote{...}` that contains the text of

11. With respect to the same origin.

Table 27: Table with a note that follows the numbering of the other notes in the text and is placed at the bottom of the page (footnote package).

Force	Force is a vector quantity defined as the rate of change of the momentum of the body that would be induced by that force acting alone.
Moment of a force	Moment of a force with respect to an origin is defined as the cross product of the position vector ¹¹ and the force.

Table 28: Table with its own numbering for the notes and with notes placed right after the table (ctable package).

Force	Force is a vector quantity defined as the rate of change of the momentum of the body that would be induced by that force acting alone.
Moment of a force	Moment of a force with respect to an origin is defined as the cross product of the position vector ^a and the force.

^a With respect to the same origin.

the note. The notes are placed right below the table and not at the bottom of the page. For example table 28 is obtained with the following code

```
\begin{ctable}%
[caption = ...,label = ...]{...}%
{\tmark{With respect to the same origin.}}%
{...\\midrule
Moment of a force & Moment of a force with respect to
an origin is defined as the cross product of the position
vector\tmark and the force.\\bottomrule}
\end{ctable}
```

Table 29: Table obtained with the slashbox package.

Function \ Argument	0	$\pi/2$
sin	0	1
cos	1	0

6.3 Slashed boxes

Sometimes the first cell on the top left of the table contains two arguments: the first one describes the content of the first column and the second one that of the first row. This solution, not recommended for scientific documents, can be obtained with the command `\backslashslashbox` provided by the slashbox package. For example the code

```
\begin{tabular}{|l|>{\$}c<{\$}|>{\$}c<{\$}|}\hline
\backslashslashbox{Function}{Argument} &0&\pi/2\\\hline
$\sin$ & 0 & 1\\\hline
$\cos$ & 1 & 0\\\hline
\end{tabular}
```

produces table 29.

The slashbox package draws a figure with two labels on the sides of an oblique line and then places this figure inside the cell of the table. The package uses L^AT_EX picture mode that has many restrictions and does not produce figures of high quality. The pict2e package can be used to improve the quality.

Acknowledgments

The author would like to thank Valeria Angeli for her assistance during the translation of this article from the italian version [19]. The author also would like to thank editorial staff of Ars T_EXnica, on which the original paper was published in Italian, that authorized the publication of this article on The PracT_EX Journal.

References

- [1] C. Beccari. *L^AT_EX, Guida a un sistema di editoria elettronica*. Hoepli, 1991.

- [2] D. P. Carlisle. *Packages in the 'graphics' bundle*, 1999. <ftp://tug.ctan.org/pub/tex-archive/macros/latex/required/graphics/grfguide.pdf>.
- [3] L. Caucci and M. Spadaccini. *Gestione di Figure e Tabelle con L^AT_EX*, 2005. http://www.guit.sssup.it/downloads/fig_tut.pdf.
- [4] P. W. Daly. *Graphics and Colour with L^AT_EX*, 1998. <http://tex.loria.fr/graph-pack/grf/grf.pdf>.
- [5] A. Diller. *L^AT_EX Line by Line: Tips and Techniques for Document Processing*. John Wiley & Sons, 1999.
- [6] R. Fairbairns. *The UK T_EX FAQ*, 2006. <http://www.ctan.org/tex-archive/help/uk-tex-faq/letterfaq.tex>.
- [7] S. Fear. *Publication quality tables in L^AT_EX*, 2005. <http://www.ctan.org/tex-archive/macros/latex/contrib/booktabs/booktabs.pdf>.
- [8] P. Flynn. *A beginner's introduction to typesetting with L^AT_EX*, 2005. <http://www.tug.org/tex-archive/info/beginlatex/>.
- [9] R. Gini. *Generazione automatica di tabelle con L^AT_EX e Stata*, 2004. http://www.guit.sssup.it/downloads/dispensa_gini.pdf.
- [10] M. Goossens, F. Mittelbach, and A. Samarin. *The L^AT_EX Companion*. Addison-Wesley, 1995.
- [11] G. Grätzer. *First Steps in L^AT_EX*. Springer Verlag, 1999.
- [12] G. Grätzer. *Math into L^AT_EX*. Birkhauser, 2000.
- [13] E. Gregorio. *L^AT_EX, Breve guida ai pacchetti di uso più comune*, December 2005. <http://profs.sci.univr.it/~gregorio/breveguida.pdf>.
- [14] J. Hahn. *L^AT_EX for Everyone: A Reference Guide and Tutorial for Typesetting Documents Using a Computer*. Prentice Hall, 1993.
- [15] D. J. Higham and D. F. Griffiths. *Learning L^AT_EX*. Society for Industrial and Applied Mathematics, 1997.
- [16] D. Knuth. *The T_EXbook*. Addison-Wesley, 1992.

- [17] H. Kopka and P. W. Daly. *A Guide to L^AT_EX – Document Preparation for Beginners and Advanced Users*. Addison-Wesley, 1995.
- [18] L. Lamport. *L^AT_EX: A Document Preparation System, User’s Guide and Reference Manual*. Addison-Wesley, 1994.
- [19] L. F. Mori. Tabelle su L^AT_EX_{2 ϵ} —pacchetti e metodi da utilizzare. *Ars T_EXnica*, (2):31–47, October 2006.
- [20] L. F. Mori. Scrivere la tesi di laurea con L^AT_EX_{2 ϵ} . In *G_UITmeeting 2005*, Pisa, Italy, October 2005. G_UIT (Italian T_EX Users Group).
- [21] T. Oetiker, H. Partl, I. Hyna, and E. Schlegl. *Una (mica tanto) breve introduzione a L^AT_EX_{2 ϵ}* , 2000. <http://www.ctan.org/tex-archive/info/lshort/italian/itlshort.pdf>.
- [22] M. Trettin. *Elenco dei “peccati” degli utenti di L^AT_EX_{2 ϵ}* , 2005. <http://www.tug.org/tex-archive/info/l2tabu/italian/l2tabuit.pdf>.
- [23] H. Voß. *Rotating Text, Tabulars and Images*, 2003. <http://perce.de/LaTeX/rotating.pdf>.

[Journal](#)
[home page](#)
[General](#)
[information](#)
[Submit an](#)
[item](#)
[Download](#)
[style files](#)
[Copyright](#)
[Contact us](#)

THE PracTeX Journal



Babel, how to enjoy writing in different languages

Enrico Gregorio

Abstract

This paper explains how to use the Babel package by Johannes Braams to write multi-lingual documents in LaTeX.

Enrico Gregorio is professor of Algebra at the University of Verona (Italy). He became involved with TeX in the late eighties, not only for writing scientific papers, but also for producing books in various fields. Since 2000 he has taught a LaTeX course at his university, attended by many students each year. He is an active member of the Italian TeX users' group, GuIT. He can be reached at gregorio@sci.univr.it

- [PDF version of paper](#)
- [Comment on this paper](#)
- [Send submission idea to editor](#)

Page generated March 14, 2007 ; [TUG home page](#); [search](#); [contact webmaster](#).

Babel, how to enjoy writing in different languages

Enrico Gregorio

Email Enrico.Gregorio@univr.it
Address Dipartimento di Informatica
Università di Verona
Italy

1 Introduction

Anybody who needs to write in languages different from English should know about babel. At the beginning of L^AT_EX tags like ‘Chapter’ or ‘Table of Contents’ were hardwired in the format and changing them required hunting for those strings in the file `lplain.tex` and doing things like

```
\makeatletter  
\renewcommand{\@chapapp}{Capitolo}  
\makeatother
```

just to change the tag for Italian. In those days, customizing L^AT_EX for a non-English language was a nightmare, not to mention the fact that you could use only one language at a time, so canning, say, `itlplain.fmt` from a modified source of L^AT_EX which loaded Italian hyphenation patterns.

Then it came Johannes Braams with his babel which acted on the innards of L^AT_EX by changing the tags to suit many European languages. It was still necessary to build different formats for different languages, this was finally overcome by the introduction of T_EX3 with its multilingual capabilities.

And then it came Thomas Esser with teT_EX and other distributions for different operating systems which made easier to define a unique format with many hyphenation patterns available.

Later again L^AT_EX2_ε allowed for simplified syntax in the loading of packages, with the possibility of passing them *options*. Now Spaniards can say

```
\documentclass[a4paper]{book}  
\usepackage[spanish]{babel}
```

and their first chapter will be named, accordingly, *Capítulo*.

In the meantime babel has been incorporated in every official distribution of L^AT_EX, so that everyone can benefit from it.

We won't go into the details, focusing instead on the use of babel, but a word about how this package accomplishes its tasks can be useful. All fixed tags are put into commands; for example, the standard L^AT_EX class book defines

```
\newcommand{\chaptername}{Chapter}
```

and uses `\chaptername` in the definition of `\chapter`. Thus it is easy for babel to change: it is sufficient for it to issue the command

```
\renewcommand{\chaptername}{Capítulo}
```

when it loads the settings for Spanish. Well, actually babel doesn't do in precisely this way, but this is unimportant from a user's point of view. For users the good news is that babel changes the tags *automatically*.

2 Loading the package

The babel is loaded in the usual way

```
\usepackage[languages]{babel}
```

where *languages* denotes a comma separated list of language names as understood by babel; you find a list in Table 1. Just to be explicit, these are examples of correct usage:

```
\usepackage[italian]{babel}
```

```
\usepackage[frenchb,english]{babel}
```

```
\usepackage[greek,uppersorbian,ukrainian]{babel}
```

The rule is to declare all languages that will be needed in the document, keeping *last* the main one, that we desire to typeset the tags with. Some of the names are just synonyms of each other (portuges and portuguese, for example, or acadian and canadien; of course, canadian is not synonymous with canadien). Some of the names are there for historical reasons and mean just the same thing (this is the case of french, francais and frenchb).

One notable exception is english, which could mean either British English or American English, depending on the choices of the manager of the T_EX system

Table 1 List of babel languages

acadian	estonian	ngerman
afrikaans	finnish	norsk
albanian	francais	nynorsk
american	french	polish
australian	frenchb	polutonikogreek
austrian	galician	portuges
bahasa	german	portuguese
bahasai	germanb	romanian
basam	greek	russian
basque	hebrew	samin
brazil	hungarian	scottish
brazilian	icelandic	serbian
breton	indon	slovak
british	indonesian	slovene
bulgarian	interlingua	spanish
canadian	irish	swedish
canadien	italian	turkish
catalan	latin	ukrainian
croatian	lowersorbian	uppersorbian
czech	magyar	welsh
danish	malay	UKenglish
dutch	meyalu	USenglish
english	naustrian	
esperanto	newzealand	

you are using; usually it means the US variety. If you are unsure about that choice, you can specify UKenglish or USenglish as option to the package.

Another peculiarity worth noting is the presence of the pairs *german*, *ngerman* and *austrian*, *naustrian*. The *n*-versions refer to the *New Orthography* (*Neue Rechtschreibung*) of German. In Table 2 we find how ‘Chapter’ is written in some languages.

Table 2 The word ‘Chapter’ in some languages

Brazilian	Czech	Greek	Samin	Ukrainian	Welsh
Capítulo	Kapitola	Κεφάλαιο	Kapihttal	Розділ	Pennod

Some readers not accustomed with European countries will be astonished by the number of languages. They will be amazed on learning that at least as many are missing than are supported. There are not Lithuanian or Luxembourgish among the others; tens of local languages are not worth supporting, as their native speakers are so few. Luxembourg is one of the most important cities of the European Union, seat of the European Court of Justice; Lithuania has one of the best basketball national teams in the world. One of the four official languages of Switzerland does not appear in the list, Romansh.

Anyway, let’s return to the supported languages. If your document uses only one of them, say Portuguese, then the invocation should be

```
\usepackage[portuges]{babel}
```

You could also give *portuges* as option to `\documentclass`, so that you pass it also to all packages that understand it, for example *varioref*. No problem here, just start writing and all your tags will be in Portuguese: the *thebibliography* environment in an article will start a new section named *Referências*.

Now here it comes the difficult part: writing a document which requires at least two languages. Say a literature commentary where some citations in the original language are needed. Let’s say I’m Italian and I want to write a paper on the Czech writer Karel Čapek, the inventor of the word *robot* in the book “RUR, Rossumovi univerzální roboti” published in 1920.

My problem is that I don’t want to type my paper with a Unicode keyboard because input would be painful. I would have to choose between a Latin1 or a Latin2 keyboard and none of them is suited for both languages: while Latin2 has all the diacritics needed for Czech, it lacks all grave accents on vowels that I must use in Italian. Then I make a decision: Italian is the main language, so I choose a Latin1 keyboard and use *babel* facilities for typing Czech.

In Table 3 you find some text in the two languages, while in Table 4 is shown how to input the Czech text.

Table 3 An example of text in two languages. First the beginning of “RUR”, in Czech, then a short passage on the life and works of Karel Čapek, in Italian, with an explanation of the above passage.

Domin: (sedí u velikého amerického psacího stolu v otáčecím křesle. Na stole žárovka, telefon, těžítka, pořadač dopisů, atd., na stěně vlevo velké mapy s lodními a železničními liniemi, veliký kalendář, hodiny, jež ukazují něco málo před polednem; na stěně vpravo tištěné plakáty: “Nejlacinější práce: Rossumovi Roboti” “Tropičtí Roboti, nový vynález. Kus 150 d.” “Každý si kup svého Robota!” “Chcete zlevnit svoje výrobky? Objednejte si Rossumovy Roboty.” Dále jiné mapy, dopravní lodní řád, tabulka s telegrafickými záznamy kursů atd. V kontrastu k této výzdobě stěn je na zemi nádherný turecký koberec, vpravo kulatý stůl, pohovka, kožená klubovní křesla a knihovna, v níž místo knih stojí láhve s vínem a kořalkami. Vlevo pokladna. Vedle Dominova stolu psací stroj, na němž píše dívka Sulla.)

Karel Čapek nacque a Malé Svatonovice nel 1890. Laureatosi in filosofia a Praga, nel 1917–1920 fu redattore di «Fogli nazionali» e del «Giornale del popolo». Cominciò a scrivere racconti insieme al fratello Josef. Karel Čapek morì a Praga nel 1938. Di Čapek va ricordata la vasta produzione drammatica: «RUR» (Rossum’s Universal Robots, 1920) importante nella fantascienza perché per la prima volta è usato il termine di «robot» riferito alla macchina antropomorfa entrato poi nell’uso universale.

Il brano riportato sopra è l’inizio del dramma «RUR», scritto sotto forma di testo teatrale. Il testo è in parentesi perché il personaggio Domin sta parlando al telefono.

As you can note by looking at Table 4, the passage in Czech is put into the environment otherlanguage*. In typing it I have used some abbreviations for inputting characters with diacritics. Czech uses four kinds of diacritics: acute accent, caron, ring and apostrophe. The only letters which can get two of them above them are ‘e’ and ‘u’. The caron over ‘D’ and ‘T’ becomes an apostrophe for the lowercase letter. The acute accent over a vowel denotes the long sound.

Table 4 An example of input in a language different from the main one of the text.

```
\begin{otherlanguage*}{czech}
\textbf{Domin}: (sedí u velikého amerického psacího stolu
v~otá"cecím k"resle. Na stole "zárovka, telefon, t"e"zítka,
po"rada" c dopis"u, atd., na st"en"e vlevo veliké mapy s~lodními
a "zelezni"cními liniemi, velik"y kalendá"r, hodiny, je"z
ukazují n"eco málo p"red polednem; na st"en"e vpravo ti"st"ené
plakáty: "<Nejlacin"ej"ší práce: Rossumovi Roboti"> "<Tropi"ctí
Roboti, nov"y vynález. Kus 150 d."> "<Ka"zd"y si kup svého
Robota!"> "<Chcete zlevnit svoje v"yrobky? Objednejte si
Rossumovy Roboty."> Dále jiné mapy, dopravní lodní "řád,
tabulka s~telegrafick"ymi záznamy kurs"u atd. V~kontrastu
k~tétu v"yzdob"e st"en je na zemi nádhern"y tureck"y koberec,
vpravo kulat"y st"ul, pohovka, ko"žená klubovní k"resla a
knihovna, v~ní"z místo knih stojí láhve s~vínem a ko"ralkami.
Vlevo pokladna. Vedle Dominova stolu psací stroj, na n"em"z
pí"se dívka Sulla.)
\end{otherlanguage*}
```

Here is a table; in the left columns of each block I have written the ‘official’ way of typing the character and in parentheses the ‘unofficial’ one I used (see later).

$\backslash' \{A\}$ (Á) → Á	$\backslash v \{C\}$ ("C) → Č	$\backslash q \{D\}$ ("D) → Ď	$\backslash' \{E\}$ (É) → É
$\backslash v \{E\}$ ("E) → Ě	$\backslash' \{I\}$ (Í) → Í	$\backslash q \{L\}$ ("L) → Ĺ	$\backslash v \{N\}$ ("N) → Ň
$\backslash' \{O\}$ (Ó) → Ó	$\backslash v \{R\}$ ("R) → Ř	$\backslash v \{S\}$ ("S) → Š	$\backslash q \{T\}$ ("T) → Ť
$\backslash' \{U\}$ (Ú) → Ú	$\backslash w \{U\}$ ("U) → Ů	$\backslash' \{Y\}$ ("Y) → Ý	$\backslash v \{Z\}$ ("Z) → Ž
$\backslash' \{a\}$ (á) → á	$\backslash v \{c\}$ ("c) → č	$\backslash q \{d\}$ ("d) → ď	$\backslash' \{e\}$ (é) → é
$\backslash v \{e\}$ ("e) → ě	$\backslash' \{i\}$ (í) → í	$\backslash q \{l\}$ ("l) → ĺ	$\backslash v \{n\}$ ("n) → ň
$\backslash' \{o\}$ (ó) → ó	$\backslash v \{r\}$ ("r) → ř	$\backslash v \{s\}$ ("s) → š	$\backslash v \{t\}$ ("t) → ť
$\backslash' \{u\}$ (ú) → ú	$\backslash w \{u\}$ ("u) → ů	$\backslash' \{y\}$ ("y) → ý	$\backslash v \{z\}$ ("z) → ž

The convention of prefixing a character with a double quote was already used before babel by a package called `german` on which babel is based. Germans had a very tough time typing their documents in L^AT_EX, and shorthands like "a instead of $\backslash' \{a\}$ were very useful.

Now, let’s come back to the business.

We want the document to be in Italian, but with Czech interspersed in it. Therefore the loading of babel will be through

```
\usepackage[czech,italian]{babel}
```

Remember! The last called language is the main language of the document, so in this case the table of contents would be named *Indice*, as is customary in Italian.

It is not advisable to pass the language option to `\documentclass`, in this case, because of possible bad interaction with the packages. Therefore, if you want to load also `varioref`, for example, you would write

```
\documentclass[a4paper]{report}
...
\usepackage[czech,italian]{babel}
\usepackage[italian]{varioref}
...
```

Sometimes there are strange interactions between packages; a rule of thumb in case incomprehensible error messages are issued by $\text{T}_{\text{E}}\text{X}$ is:

- call every package taking a language option *after* `babel`; this must be done always;
- move *before* `babel` the other packages until the error message is not issued any more;
- call `hyperref` last, unless some package requires explicitly to be called after it.

3 Hyphenation

With $\text{T}_{\text{E}}\text{X}3$ we can hyphenate correctly as many languages as we want, provided the hyphenation rules are made known to it in advance. Well, there is a limit imposed by memory, which can be increased in case of need, and a limit of 256 different set of rules. While this number is less than the number of languages in the world, it should be sufficient for most of us.

The choice of hyphenation rules known to $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ has to be done in advance, when the $\text{T}_{\text{E}}\text{X}$ system is configured. However, all distributions make it possible to rebuild the formats in order to allow for new hyphenation rules. The mechanism specific to a distribution is explained in the distribution's documentation. For $\text{MacT}_{\text{E}}\text{X}$, for example, there is a "Configure" button and one of the things to do is the choice of hyphenation rules. The same applies to $\text{MiK}_{\text{T}}\text{E}_{\text{X}}$ or $\text{T}_{\text{E}}\text{XLive}$.

Some `babel` languages share those rule: `portuges` and `brazil`, for example, or `german` and `austrian`. They are named differently by `babel` because they have different tags in some cases.

Portugueses call *Índice* the Index, while *Brazileiros* call it *Índice Remissivo*. The month of January is *Januar* in *Deutschland*, while it is *Jänner* in *Österreich*.

A very different case is English: hyphenation rules are *not* the same for British English and American English, being them two very different languages.¹ On this account, the rules correspond to distinct choices during the setup.

If hyphenation for a language is not enabled, the rules for English are applied. This can be inconvenient, but there is no provision, up to now, to set things in a different way. In case you are not in charge of the system and want to use, say, Indonesian, a poor person's remedy until the system manager accepts to include the rules for it in the format is to write the following incantation before loading `babel`:

```
\makeatletter
\chardef\l@bahasa=255
\makeatother
```

How to find the exact string to put between @ and = for other languages? In the log file. If you request a language with unknown hyphenation rules, \LaTeX warns you on the screen in this way:

```
No hyphenation patterns were loaded for
the language 'Bahasa Indonesia'
```

You find that message also in the log file and immediately after it you will see the right string to use:

```
\l@bahasa = a dialect from \language0
```

After the incantation, Indonesian will not be hyphenated at all and you will need to cope with bad paragraphs by inserting "soft hyphens", i.e., the command `\-`. This is surely better than getting wrong hyphens. Hopefully, the system manager will be nice enough to enable the language for you: give your thanks, cut off the incantation, erase the soft hyphens and enjoy automatic hyphenation for Indonesian.²

Don't ask what the incantation does; after all, magicians never explain their tricks.

1. G.B. Shaw said that the USA and the UK are two countries divided by a common language. Oscar Wilde said with his characteristic wit "we have really everything in common with America nowadays, except, of course, the language".

2. If *you* are the system manager, then ask some \TeX guru around the discussion forums or, as a last resort, read the manual of your \TeX distribution.

4 Using babel

We have already seen that the last option to the package is considered the main language of the document. For example, this changes automatically the format for dates, when you say `\today`:

US English	March 14, 2007
Czech	14. března 2007
Greek	14 Μαρτίου 2007
Italian	14 marzo 2007
Samin	njukčamánu 14. b. 2007
Ukrainian	14 березня 2007 р.
Welsh	14 Mawrth 2007

Of course you need to input correctly another date, following the standard of the language. Or you can make L^AT_EX work:

```
\newcommand{\printdate}[3]{\%  
  \day=#1\relax\month=#2\relax\year=#3\relax\today}}
```

US English	December 25, 2004
Czech	25. prosince 2004
Greek	25 Δεκεμβρίου 2004
Italian	25 dicembre 2004
Samin	juovlamánu 25. b. 2004
Ukrainian	25 грудня 2004 р.
Welsh	25 Rhagfyr 2004

All these dates are printed giving the same command `\printdate{25}{12}{2004}` in a context where the appropriate language is enforced. If you prefer to give dates in ISO format, there is another incantation:

```
\def\getisodate#1/#2/#3/{%  
  \day=#3\relax\month=#2\relax\year=#1\relax\today}  
\def\printisodate#1{\def\theisodate{#1}%  
  \expandafter\getisodate\theisodate/}}
```

After this we can use something like `\printisodate{1907/5/26}` to get, according to the language, the following results:

US English	May 26, 1907
Czech	26. května 1907
Greek	26 Μαΐου 1907
Italian	26 maggio 1907
Samin	miessemánu 26. b. 1907
Ukrainian	26 травня 1907 р.
Welsh	26 Mai 1907

The package `isodate` has some features to do this thing, if you fear using the magic above. Who was born in that day?

Now we come to the main part of the section: how to change language in the middle of the document.

4.1 A long document with the first part in one language and the second one in another

Suppose you have to write a booklet consisting of four chapters; the first two are in Czech, while the front matter (Introduction), the last two chapters the back matter (an appendix, say) are in Italian. You want that the word ‘Chapter’ comes out in Czech for the first two chapters and in Italian in the last two. In Table 5 there is how to set the document.

If you typeset the sample (changing the dots into some meaningful text), you will notice that the first chapter has the title prefixed by “Kapitola 1”, while the title of the third one will be prefixed by “Capitolo 3”.

To be a bit more formal, the command

```
\selectlanguage{language}
```

changes the complete setup of the document to the chosen *language*, which of course must have been declared at the beginning. All tags after the command `\selectlanguage{czech}` will come out as defined for the Czech language. If the language has also particular typesetting rules, like French, these will be enforced.

In Table 6 you find an example, the same text in French and Italian, with the relative code.

As you can see, in the French text the question mark is preceded by an interword space, as it is customary in French typography, but it is not necessary (although possible) to leave a space in front of it in the input. The same happens for the colon, the semicolon, the exclamation mark and the (French) quotes, but `babel` takes care of these details, not the user. Well, something is left to the user: the ellipsis is typed `...` in the French text, not with `\ldots`.

Table 5 An example of a two language document

```
\documentclass[a4paper]{book}
\usepackage[czech,italian]{babel}
...
\begin{document}
...
\frontmatter
\chapter{Introduzione}
...
\mainmatter
\selectlanguage{czech}
\chapter{Karel \v{C}apek}
...
\chapter{\v{C}eská Republika}
...
\selectlanguage{italian}
\chapter{La vita di Karel \v{C}apek}
...
\chapter{Le opere di Karel \v{C}apek}
...
\appendix
\chapter{Appendice}
...
\end{document}
```

Some languages, notably Hebrew and Arabic, write right to left. Arabic is not supported by babel, but Hebrew is. For switching to right to left typesetting or conversely, the environment `otherlanguage` must be used. Otherwise its effect is the same as `\selectlanguage`. I'm not going to discuss Hebrew typesetting here, I'll only give a code example.

```
\usepackage[hebrew,spanish]{babel}
...
\begin{document}
<text in Spanish>
\begin{otherlanguage}{hebrew}
<text in Hebrew>
\end{otherlanguage}
```

Table 6 The same text in two languages

Qu'est-ce qui a pu faire disjoncter Zizou, dimanche 9 juillet, en finale du Mondial? Qu'est-ce qu'a bien pu lui dire l'Italien Marco Materazzi pour recevoir un coup de tête en plein torse? Depuis la sortie sur carton rouge du capitaine de l'équipe de France, les médias du monde entier se posent la question, tentent de décrypter sur les lèvres de l'Italien la teneur des – supposées – insultes... et les hypothèses se multiplient.

```
\selectlanguage{french}
Qu'est-ce qui a pu faire disjoncter
Zizou, dimanche 9~juillet, en finale
du Mondial? Qu'est-ce qu'a bien pu
lui dire l'Italien Marco Materazzi
pour recevoir un coup de tête en
plein torse? Depuis la sortie sur
carton rouge du capitaine de
l'équipe de France, les médias du
monde entier se posent la question,
tentent de décrypter sur les lèvres
de l'Italien la teneur des
--~supposées~-- insultes... et les
hypothèses se multiplient.
```

Che cosa ha fatto scatenare Zizou, domenica 9 luglio, alla finale dei Mondiali? Che cosa ha potuto dirgli l'italiano Marco Materazzi per ricevere una testata in pieno petto? Dopo l'uscita per il cartellino rosso del capitano della squadra francese, i mezzi di comunicazione del mondo intero si pongono queste domande, tentano di interpretare dalle labbra dell'italiano il tenore dei presunti insulti... e le ipotesi si moltiplicano.

```
\selectlanguage{italian}
Che cosa ha fatto scatenare Zizou,
domenica 9~luglio, alla finale dei
Mondiali? Che cosa ha potuto dirgli
l'italiano Marco Materazzi per
ricevere una testata in pieno
petto? Dopo l'uscita per il
cartellino rosso del capitano della
squadra francese, i mezzi di
comunicazione del mondo intero si
pongono queste domande, tentano di
interpretare dalle labbra
dell'italiano il tenore dei presunti
insulti\ldots\ e le ipotesi si
moltiplicano.
```

```
<text in Spanish>
\end{document}
```

4.2 Some text only in a different language

Most of the time, however, it is only necessary to type some passages in a language different from the main one. Changing the tags would be useless in this case. For this purpose there are a command and an environment which are perfectly equivalent to one another. They are

```
\begin{otherlanguage*}{<language>}
<text>
\end{otherlanguage*}
```

and

```
\foreignlanguage{<language>}{<text>}
```

The environment form is suitable for longer passages consisting of one or more paragraphs; the command form is best employed for short phrases, in line citations, for example.

The text inside the environment or the argument of the command form can be typed with the shorthands provided by babel.

Let's see an example. We know that in German we can input letters with the *Umlaut* by prefixing them with a double quote; a feature very handy for people using a Latin2 keyboard which misses those symbols.

These are the first words of the *Hymn to Joy* by Schiller, set to music by Beethoven: *Freude, Schöner Götterfunken, Tochter aus Elysium*. They were typed as

```
\foreignlanguage{german}{Freude, Sch"oner
G"otterfunken, Tochter aus Elysium}
```

Of course it is not possible to list here all the shorthands for each language. The user guide to babel lists them all.

One feature particularly interesting is the command `\iflanguage`. Suppose that some language prefers boldface type for emphasis rather than italics. You are writing a document in two languages and don't want to use different commands for the same logical structure. So, assume that in Italian we want boldface for emphasis and in English, the main language, we want italics: we define a command in this way

```
\newcommand{\Emph}{\iflanguage{italian}{\textbf}{\emph}}
```

After this we can write something like what is in Table 7. No, in Italian we don't use italics for emphasis, should we?

Table 7 Example of `\iflanguage`

It's easy to put <i>emphasis</i> on words.	È facile mettere parole in evidenza .
--------------------------------------------	----------------------------------------------

```
\selectlanguage{english}
It's easy to put
\Emph{emphasis} on words.
```

```
\selectlanguage{italian}
È facile mettere parole
in \Emph{evidenza}.
```

More programming is possible by taking into account that the current language name is available through the command `\language`. Some more specialized commands are available, but they are not for the common user.

5 Adding or changing tags

Sometimes the tags provided by default just don't fit a user's needs. Suppose that instead of "Figura" as the tag for the figures you want "Illustrazione", for an article with Italian as its main language. The correct way to cope with this is *not* to redefine the command `\figurename` that, you have guessed, contains the tag.

The problem is that, after a language shift with `\selectlanguage` and a return to Italian, the figures will be labeled again with "Figura". This happens because `babel` keeps a list of those tags and does a bunch of `\renewcommand` stuff, but the original tag remains.

It's not difficult to arrange this; the tags for Italian are restored with the command `\captionsitalian` and so all we need is to add our definition to it:

```
\addto\captionsitalian{\renewcommand{\figurename}{Illustrazione}}
```

in the preamble will set everything as we want.

It is in order to give a list of all predefined tags. You find it in Table 8. Their meaning can be changed in the same way as in the example before, on a language basis.

You can also add tags of your own. Say that you need a tag for algorithms. This should be "Algorithm" in English and "Algoritmo" in Italian. The method is analogous to the one for changing, but first we have to make the symbolic name known to `LATEX`; therefore we write in the preamble:

Table 8 The list of predefined tags, with their value in English

<code>\prefacename</code> (Preface)	<code>\tablename</code> (Table)
<code>\refname</code> (References)	<code>\partname</code> (Part)
<code>\abstractname</code> (Abstract)	<code>\enclname</code> (encl)
<code>\bibname</code> (Bibliography)	<code>\ccname</code> (cc)
<code>\chaptername</code> (Chapter)	<code>\headtoname</code> (To)
<code>\appendixname</code> (Appendix)	<code>\pagename</code> (Page)
<code>\contentsname</code> (Contents)	<code>\seename</code> (see)
<code>\listfigurename</code> (List of Figures)	<code>\alsiname</code> (see also)
<code>\listtablename</code> (List of Tables)	<code>\proofname</code> (Proof)
<code>\indexname</code> (Index)	<code>\glossaryname</code> (Glossary)
<code>\figurename</code> (Figure)	

```
\newcommand{\algorithmname}{}  
\addto\captionenglish{\renewcommand{\algorithmname}{Algorithm}}  
\addto\captionitalian{\renewcommand{\algorithmname}{Algoritmo}}
```

Of course you will add the tag definition to every language for which you need to use that new tag.

[Journal](#)
[home page](#)
[General](#)
[information](#)
[Submit an](#)
[item](#)
[Download](#)
[style files](#)
[Copyright](#)
[Contact us](#)

THE PracTeX Journal



Hacking DVI files

Jin-Hwan Cho

Abstract

This paper is devoted to the first step in developing a new DVI editing utility, called DViasm. Editing DVI files consists of three parts: disassembling, editing, and assembling. DViasm disassembles a DVI file into a human-readable text format which is more flexible than DTL, and assembles the output back into a DVI file. DViasm is quite useful for people who have a DVI file without a TeX source, and need to modify the document. It enables attaching a preprint number, a watermark, or an emblem to a document without touching the TeX source. DViasm is quite useful even to a TeX expert who wants to modify a few words in a large document. We discuss in the paper how DViasm plays a role as supplementary to TeX. The current version supports only the standard DVI file format as DVItyp and DTL. The next versions will support 16-bit TeX extensions including Omega, pTeX, and XeTeX.

Jin-Hwan Cho received his PhD in Mathematics in 1999. In TeX society he has been known as a co-author of DVIPDFMx. He gave two talks about DVIPDFMx at the TUG conference in 2003 and 2005. He can be reached at chofchof@ktug.or.kr

- [PDF version of paper](#)
- [Comment on this paper](#)
- [Send submission idea to editor](#)

Page generated March 14, 2007 ; [TUG home page](#); [search](#); [contact webmaster](#).

Hacking DVI files: Birth of DViasm

Jin-Hwan Cho

Email chofchof@ktug.or.kr

Address Department of Mathematics, The University of Suwon, Republic of Korea

Abstract This paper is devoted to the first step of developing a new DVI editing utility, called DViasm. Editing DVI files consists of three parts: disassembling, editing, and assembling. DViasm disassembles a DVI file into a human-readable text format which is more flexible than DTL, and assembles the output back to a DVI file.

DViasm is quite useful for people who have a DVI file without $\text{T}_{\text{E}}\text{X}$ source, but needs to modify the document. It enables us to put a preprint number, a watermark, or an emblem on the document without touching the $\text{T}_{\text{E}}\text{X}$ source. DViasm is quite attractive to even a $\text{T}_{\text{E}}\text{X}$ expert who wants to modify a few words in his document more than hundred pages long.

We discuss in the paper how DViasm plays a role as supplementary to $\text{T}_{\text{E}}\text{X}$. The current version supports only the standard DVI file format as DVItypex and DTL. The next versions will support 16-bit $\text{T}_{\text{E}}\text{X}$ extensions including Omega, $\text{p}\text{T}_{\text{E}}\text{X}$, and $\text{X}_{\text{3}}\text{T}_{\text{E}}\text{X}$.

1 Introduction

Have you ever heard of DVI, not the Digital Visual Interface¹ but the DeVice-Independent file format? At least ten years ago every $\text{T}_{\text{E}}\text{X}$ user knew what DVI is and used DVI utilities to view and print out the $\text{T}_{\text{E}}\text{X}$ results. However, recent $\text{T}_{\text{E}}\text{X}$ users drew attention to DVI less and less because $\text{pdf}\text{T}_{\text{E}}\text{X}$ outputs directly to the PDF ² file format. It is true without doubt that PDF is more powerful than DVI in

1. A video interface standard designed to maximize the visual quality of digital display devices such as flat panel LCD computer displays and digital projectors [Wikipedia, <http://en.wikipedia.org/wiki/DVI>].

2. PDF (Portable Document Format) is an open file format created and controlled by Adobe Systems, for representing two-dimensional documents in a device independent and resolution independent fixed-layout document format [Wikipedia, <http://en.wikipedia.org/wiki/PDF>].

almost all aspects. Then, do we have to obsolete DVI as PostScript is gradually replaced by PDF?

The DVI file format was designed by David R. Fuchs in 1979, in contrast to the release of PDF version 1.0 in 1993. It is intended to be both compact and easily interpreted by a machine [4, §14]. The most powerful aspect of DVI compared to PDF is nothing but *simplicity*. Imagine the speed of three previewers of DVI, PostScript, and PDF, and compare also the file size of the three different file formats. Furthermore, simplicity enables us to control DVI files in various ways. One of them is to edit DVI files directly, that is the main object of this paper.

There are many applications of editing DVI files. The most critical situation is when we have a DVI file without \TeX source, but we want to modify or to add something to the document. A technical editor may want to put a preprint number on each paper without touching the \TeX source. He may also want to put a watermark or an emblem on every paper.

Editing a DVI file is much quicker for a \TeX novice than learning \TeX , when all he wants is to give some decorations in his document, but has some trouble in writing \TeX codes. It is even quite attractive to a \TeX expert who wants to modify a few words in his document more than hundred pages long.

Since a DVI file consists of binary data, it must be converted to a human readable text format to inspect and edit its contents. One of the DVI utilities for these purposes is DVItypex [4] written by Donald E. Knuth since 1982. It has two chief purposes, one is to validate DVI files and the other is to give an example for developers of DVI utilities [4, §1]. DVItypex is a nice utility to inspect the contents of a DVI file because of its human readable text output. However, it lacks the procedure converting the output back to a DVI file.

A real DVI editing utility is the DTL (Device-independent Text Language) package [5] developed by Geoffrey Tobin. It includes two utilities `dv2dt` and `dt2dv` for converting from DVI to DTL and vice versa. It is notable that there is a one-to-one correspondence between DTL and DVI, and that DTL does not require TFM font metric files in contrast to DVItypex. However, DTL is not flexible for ordinary \TeX users. For example, users must choose a correct command from 'r1' to 'r4' according to the amount of the right move. Moreover, the latest version of DTL was released in 1995, and so it did not support extended DVI formats

generated by Omega³ or Japanese pTeX.⁴

The roadmap to develop a new DVI editing utility, called DViasm, consists of three steps. This paper is devoted to the first step in which DViasm is introduced with several examples. The current version of DViasm supports only the standard DVI file format as DVItypex and DTL, but is more flexible than DTL.

In the second step we will focus on 16-bit characters, for instance, Chinese, Japanese, Korean, and Unicode, to support Omega, pTeX, and the subfont scheme⁵ which enables us to use 16-bit characters in TeX and pdfTeX. DViasm will communicate with the kpathsea library in the final step so that it will read font metric information from TFM, OFM, JFM, TrueType, and OpenType font files. It means that DViasm will also support XeTeX⁶ which reads font metric information directly from the font file itself.

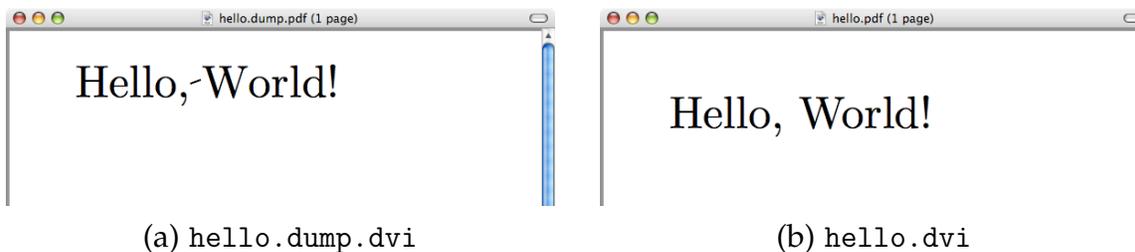
2 Prerequisite

2.1 Download and installation

The current version of DViasm is written in the Python programming language.⁷ Why Python not C? The main reason is that Python does not require compiling and linking to get an executable file. Thus, DViasm consists of a single Python program `dviasm.py` with human-readable text format and it can run on any platform in which Python is installed. If speed-up is required later, some parts of DViasm will be translated into the C programming language.

The development of DViasm is controlled by Subversion, a popular version control system, and all revisions of DViasm can be downloaded at [2]. From now on we assume that `dviasm.py` is in the working directory. The basic usage of DViasm will be out if the option `--help` is attached as follows.

-
3. An extension of TeX by John Plaice and Yannis Haralambous, <http://omega.enstb.org>.
 4. ASCII Nihongo TeX by ASCII Corporation, <http://www.ascii.co.jp/pb/ptex/index.html>.
 5. The subfont scheme is a way of splitting the set of 16-bit characters into 256 characters or less, the number of characters that TFM can accommodate [3].
 6. A typesetting system based on a merger of TeX with Unicode and Mac OS X font technologies, by Jonathan Kew, <http://scripts.sil.org/xetex>.
 7. Python is a dynamic object-oriented programming language that runs on almost all operating systems. Just type 'python' and hit the return key in the terminal to check whether Python is already installed or not. If not installed, visit the official website <http://www.python.org>.



(a) `hello.dump.dvi`

(b) `hello.dvi`

Figure 1: DVI result generated by DViasm (a) and by \TeX (b)

```
python dviasm.py --help
```

2.2 Creating a DVI file without \TeX

We first try to save the following three lines as `hello.dump`. Note that the number in the beginning of each line is just the line number for reference.

```
1 [page 1 0 0 0 0 0 0 0 0 0]
2 fnt: cmr10 at 50pt
3 set: 'Hello, World!'
```

Then run the following command in the terminal

```
python dviasm.py hello.dump -o hello.dump.dvi
```

to get a new DVI file, `hello.dump.dvi`. Its contents are shown in Figure 1(a). Notice that all DVI results in this paper are converted to PDF with DVIPDFMx⁸ version 20061211. The DVI result can also be converted to PostScript with DVIPS,⁹ and viewed in the screen with DVI previewers, `xdvi`,¹⁰ `dviout`,¹¹ or `yap`.¹²

Each page begins with the opening square bracket followed by the string 'page' (without colon), ten numbers, and the closing square bracket. Among the numbers the first one usually stands for the page number. In the second line the DVI

8. A DVI to PDF converting utility by Shunsaku Hirata and Jin-Hwan Cho, <http://project.ktug.or.kr/dvipdfmx/>. It is an extension of DVIPDFM written by Mark A. Wicks, <http://gaspra.kettering.edu/dvipdfm/>.

9. A DVI to PostScript converter by Tom Rokicki, <http://www.radicaleye.com/dvips.html>.

10. A DVI previewer in X Window system by Paul Vojta, <http://math.berkeley.edu/~vojta/xdvi.html>.

11. The most popular DVI previewer in Japan that supports p \TeX , <http://akagi.ms.u-tokyo.ac.jp/dviout-ftp.html>.

12. The DVI previewer in the MiK \TeX system by Christian Schenk, <http://www.miktex.org>.

command ‘fnt:’ selects the Computer Modern font, cmr10 scaled at 50 pt. In the last line the text ‘Hello, World!’ is typeset by the command ‘set:’.

2.3 Disassembling a DVI file

We now try to disassemble a DVI file. At first, make a T_EX file `hello.tex` with the following line,

```
\nopagenumbers \font\fnt=cmr10 at 50pt \noindent\fnt Hello, World! \bye
```

and run T_EX (not L^AT_EX) to get `hello.dvi`. The result is shown in Figure 1(b).

One may find easily two different points between (a) and (b) in Figure 1. The first one is the location of the text,¹³ and the other one is the ‘cross for ł and Ł’¹⁴ in (a) instead of the blank space in (b). Looking the figures closely, one more different point can be found. There is no kerning between the two characters ‘W’ and ‘o’ in (a). The kerning information is stored in TFM font metric files so that DViasm needs to communicate with the kpathsea library to fetch the information. Then, DViasm no longer works if whole T_EX system is not installed. This is the reason why DTL and the current version of DViasm do not require TFM font metric files.

To see the differences exactly, let us disassemble `hello.dvi` with DViasm by

```
python dviasm.py hello.dvi
```

to get the output¹⁵ in Code 1. One can see four new commands, ‘push:’, ‘pop:’, ‘right:’, and ‘down:’. An amount of move follows ‘right:’ and ‘down:’ as an argument. The meaning of the two commands looks clear.

However, there are two things to keep in mind. The coordinate system of DVI is different from the Cartesian coordinate system¹⁶ used in PostScript and PDF. In DVI the x -coordinate increases from left to right as the Cartesian coordinate

13. The upper left corner of the paper has the coordinate $(-1\text{ in}, -1\text{ in})$, since the default x - and y -offsets are both one inch as usual. So the reference point of ‘H’ is the origin $(0,0)$ in Figure 1(a). However, it is common to place the upper left corner of ‘H’ at the origin as Figure 1(b).

14. The ASCII code of the blank space `_` is 32 and the 32th glyph in cmr10 is the cross for ł and Ł.

15. DViasm always outputs to the standard output (`stdout`) if the `-o` option is not specified.

16. The Cartesian coordinate system is used to determine each point uniquely in a plane through a pair of numbers (x,y) , usually called the x -coordinate and the y -coordinate of the point [Wikipedia, http://en.wikipedia.org/wiki/Cartesian_coordinate_system].

```

1  [preamble]
2  id: 2
3  numerator: 25400000
4  denominator: 473628672
5  magnification: 1000
6  comment: ' TeX output 2007.01.24:1740'
7
8  [postamble]
9  maxv: 667.202545pt
10 maxh: 469.754990pt
11 maxs: 2
12 pages: 1
13
14 [font definitions]
15 fntdef: cmr10 (10.0pt) at 50.0pt
16
17 [page 1 0 0 0 0 0 0 0 0]
18 push:
19   down: -14.0pt
20 pop:
21   down: 643.202545pt
22 push:
23   down: -608.480316pt
24   push:
25     fnt: cmr10 (10.0pt) at 50.0pt
26     set: 'Hello,'
27     right: 16.666687pt
28     set: 'W'
29     right: -4.166702pt
30     set: 'orld!'
31   pop:
32 pop:
33   down: 24.0pt

```

Code 1: Disassembled output of `hello.dvi` by DViasm

system, but the y -coordinate increases from top to bottom, the opposite of the Cartesian coordinate system. The next one is that all positions in DVI are specified not absolutely but relatively. It is nonsense in DVI to give a command like “go to the coordinate (100 pt, 100 pt).” Only ‘right:’ and ‘down:’ are allowed in DVI.

How do we move to a specific position in DVI? Instead, we can use the two commands ‘push:’ and ‘pop:’. The command ‘push:’ stores the current position in the stack, and ‘pop:’ restores the position saved in the stack to the current position.

3 DVI commands

We now assume that the lines in Code 1 from the 17th line to the end are saved as `hello.dump`. The first example is to put some mark at the origin (0,0) to know the exact location in the paper. It is achieved by inserting two lines after the first line as Code 2.

DVI has only two drawing commands, ‘putrule:’ and ‘setrule:’. Both commands draw a box filled with black. The first and the second arguments indicate the size of the height and the width of the box, respectively. Do not confuse the

```

1 [page 1 0 0 0 0 0 0 0 0 0]
2 putrule: 1cm 0.5pt
3 putrule: 0.5pt 1cm
4 push:
5   down: -14.0pt
6 pop:
7 ... (skip) ...

```



Code 2: Put some mark at the origin (0,0).

order of height and width. The command 'setrule:' is the same as 'putrule:' except for moving to the right by the amount of the width after drawing the box.

The next example is to put a box filled with red *under* the text. Since DVI has no color command, we used in Code 3 the special command 'xxx:' that will be explained in the next section.

```

8 ... (skip) ...
9 down: -608.480316pt
10 xxx: 'color push rgb 1 0 0'
11 putrule: 10pt 4in
12 xxx: 'color pop'
13 push:
14 ... (skip) ...

```



Code 3: Put a box filled with red under the text.

Exercise. Put the red box *over* the string to hide the overlapped part of the text.

We list below the commands used in DVlasm. There are two types of arguments, string and length. The string type consists of text string surrounded by either apostrophes (') or double quotation marks ("). It has the same format as the Python string type.¹⁷ The length type is either an integer or a floating point number followed by unit (e.g., sp, pt, bp, mm, cm, in.)¹⁸ If no unit is specified, the number is in unit of sp by default. The argument of 'fnt:' is exceptional. The name of the font is given without apostrophes.

17. We can input any 8-bit character with hexadecimal value *hh* by '\xhh'. Thus, '\\ must be used to type the escape character '(backslash)'.
18. 1 in = 2.54 cm = 25.4 mm = 72 bp = 72.27 pt, and 1 pt = 2¹⁶ sp = 65,536 sp

command	argument		description
set:	string		draw [string] and move to the right by the total width of the string
put:	string		draw [string] without moving to the right
setrule:	length1	length2	draw a box with width [length2] and height [length1] and then move to the right by [length2]
putrule:	length1	length2	draw a box with width [length2] and height [length1] without moving to the right
push:			save the current position to the stack
pop:			restore the position in the stack to the current position
right:	length		move to the right by [length] move to the left if [length] is negative
down:	length		move down by [length] move up if [length] is negative
fnt:	name at length		select the font [name] scaled at [length] [name] does not allow spaces
xxx:	string		DVI special command to be processed by DVI utilities; see the next section

There are more move commands as follows. We refer to [4, §15] for details.

command	argument		description
w:	length		the same as right:, but [length] is stored in the 'w' variable
x:	length		the same as right:, but [length] is stored in the 'x' variable
y:	length		the same as down:, but [length] is stored in the 'y' variable
z:	length		the same as down:, but [length] is stored in the 'z' variable
w0:			move to the right by the length in the 'w' variable
x0:			move to the right by the length in the 'x' variable
y0:			move down by the length in the 'y' variable
z0:			move down by the length in the 'z' variable

4 DVI specials

We have seen all DVI commands in the previous section. There is no command for color, graphics, and transformation in DVI. But we already know that they are possible in \TeX . How do they work?

The answer is the DVI special command 'xxx:'. It is the only way for \TeX to communicate with DVI utilities. However, each DVI utility supports its own DVI specials. For example, neither DVIPDFM nor DVIPDFMx support a PostScript literal special containing PostScript codes. On the other hand, almost none of the PDF specials work with DVIPS.

In this section we introduce common DVI specials and show some examples using DViasm. All materials in this section are based on the talk of the author at TUG 2005 conference [1].

4.1 Page specials

There are two kinds of page specials. The first example shows how to resize the page of the previous example in Code 3.

`papersize=[width],[height]` changes the size of whole pages. But it has no effect on the paper size that can be changed by the command line option or by the configuration file (supported by DVIPS*,¹⁹ DVIPDFM, and DVIPDFMx).

`pdf:pagesize width [length] height [length]` changes the size of the page containing this special (supported by DVIPDFM*(?) and DVIPDFMx).

```
1 [page 1 0 0 0 0 0 0 0 0 0]
2 xxx: 'papersize=6in,3in'
3 putrule: 1cm 0.5pt
4 putrule: 0.5pt 1cm
5 push:
6   down: -14.0pt
7 pop:
8 ... (skip) ...
```

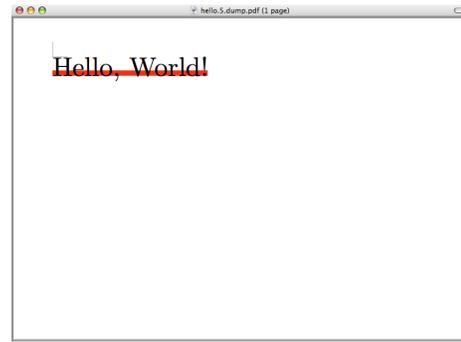


The second example is to make the paper landscape from portrait.

19. * denotes originality and (?) means that the behavior looks mysterious or buggy.

`landscape` swaps the width and the height of the paper size (supported by DVIPS*, DVIPDFM, and DVIPDFMx).

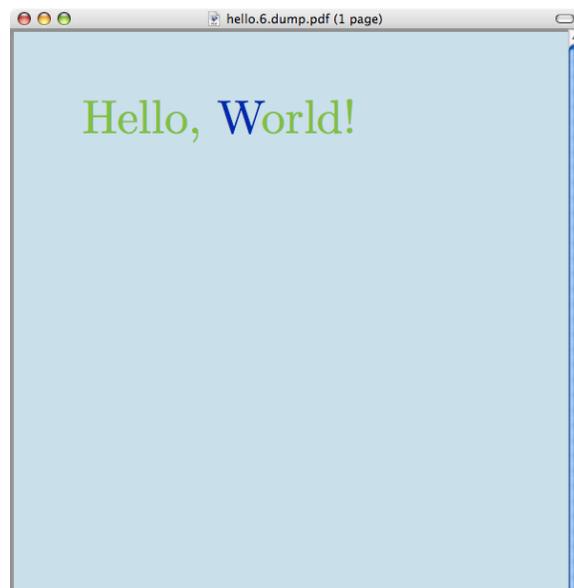
```
1 [page 1 0 0 0 0 0 0 0 0 0]
2 xxx: 'landscape'
3 putrule: 1cm 0.5pt
4 putrule: 0.5pt 1cm
5 push:
6   down: -14.0pt
7 pop:
8 ... (skip) ...
```



4.2 Color specials

All common color specials are originated by DVIPS. Various types of colors can be specified in color specials: `[PScolor]` must be one of `'cmyk [c] [m] [y] [k]'`, `'rgb [r] [g] [b]'`, `'hsb [h] [s] [b]'`, `'gray [g]'`, and predefined color names, where the value of each color component is a number between 0.0 and 1.0. We refer to [6, pp. 12–13] and [1, p. 11] for PDF color specials which are easier to understand than PS color specials.

```
1 [page 1 0 0 0 0 0 0 0 0 0]
2 xxx: 'background cmyk .183 .054 0 0'
3 down: 643.202545pt
4 push:
5   down: -608.480316pt
6   xxx: 'color push LimeGreen'
7   push:
8     fnt: cmr10 (10.0pt) at 50.0pt
9     set: 'Hello,'
10    right: 16.666687pt
11    xxx: 'color push rgb 0 0 .625'
12    set: 'W'
13    xxx: 'color pop'
14    right: -4.166702pt
15    set: 'orld!'
16  pop:
17  xxx: 'color pop'
18 pop:
```



`background [PScolor]` sets a fill color for the background (supported by DVIPS*, DVIPDFM, and DVIPDFMx).

`color push [PScolor]` saves the current color on the color stack and sets the current color to the given one (supported by DVIPS*, DVIPDFM, and DVIPDFMx).

`color pop` pops a color from the color stack and sets the current color to be that color (supported by DVIPS*, DVIPDFM, and DVIPDFMx).

`color [PScolor]` clears the color stack, and saves and sets the given color (supported by DVIPS*, DVIPDFM(?), DVIPDFMx).

4.3 Image specials

PostScript provides one image special ‘psfile’ for including EPS graphics file. Every EPS file has a bounding box information. For example, the bounding box of the EPS file²⁰ in the following example is

```
%%BoundingBox: 17 171 567 739
```

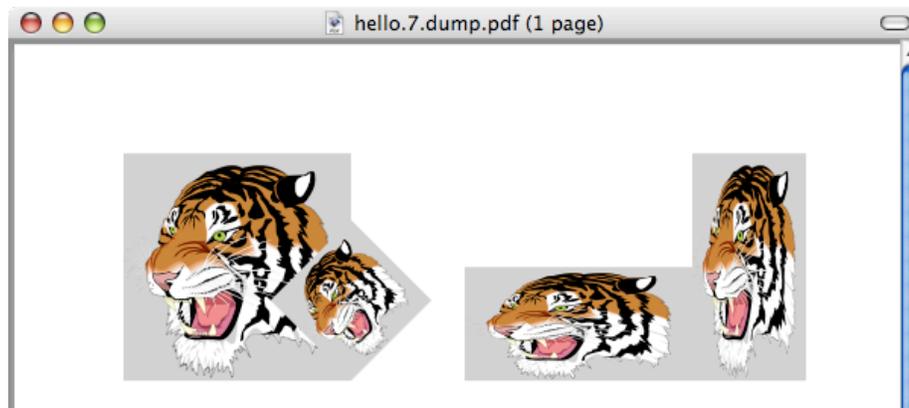
Four options `llx`, `lly`, `urx`, and `ury` are used to specify the clipping area of the EPS file, and two options `rwi` and `rhi` (0.1 bp unit) are used to resize the clipped area.

```
psfile=[name] hsize=[num] vsize=[num]
hoffset=[num] voffset=[num]
hscale=[num] vscale=[num] angle=[num]
llx=[num] lly=[num] urx=[num]
ury=[num] rwi=[num] rhi=[num] [clip]
```

(supported by DVIPS*, DVIPDFM, and DVIPDFMx).

```
1 [page 1 0 0 0 0 0 0 0 0]
2 down: 150bp
3 xxx: 'psfile=tiger.eps rhi=1500 llx=17 lly=171 urx=617 ury=771 clip'
4 right: 150bp
5 xxx: 'psfile=tiger.eps rhi=750 llx=17 lly=171 urx=617 ury=771 angle=45 clip'
6 right: 75bp
7 xxx: 'psfile=tiger.eps rwi=1500 rhi=750 llx=17 lly=171 urx=617 ury=771 clip'
8 right: 150bp
9 xxx: 'psfile=tiger.eps rwi=750 rhi=1500 llx=17 lly=171 urx=617 ury=771 clip'
```

20. It is tiger.eps that can be found in the examples directory of Ghostscript, the most popular interpreter for the PostScript language and for PDF under GPL license. Visit <http://www.ghostscript.com/awki> for more information.



Neither DVIPDFM nor DVIPDFMx has internal PostScript interpreting routine so that they cannot process EPS files without Ghostscript or other PostScript distill utilities. Instead, both DVI utilities support JPEG and PDF image files that are not processed in DVIPS. The PDF image special for JPEG and PDF images has reader-friendly syntax. We refer to [6, p. 13] and [1, pp. 12–14] for examples.

```
pdf:image width [length] height [length]
depth [length] rotate [num]
scale [num] xscale [num] yscale [num]
bbox [ulx] [uly] [lrx] [lry]
matrix [a] [b] [c] [d] [x] [y] ([name])
```

(supported by DVIPDFM*(?)
and DVIPDFMx).

4.4 Transformation specials

It is possible in L^AT_EX to rotate and scale text and figure. But DVIPS has no transformation special for this purpose. Instead, it enables us to insert literal PostScript code.

" [PScode] inserts literal PostScript code surrounded by a pair of `gsave` and `grestore` to have no effect on the rest of the document (supported by DVIPS* only).

ps: [PScode] inserts literal PostScript code without a pair of `gsave` and `grestore` (supported by DVIPS* only).



```

1 [page 1 0 0 0 0 0 0 0 0]
2 xxx: 'papersize 2in,2in'
3 xxx: '" Goldenrod newpath 0 0 moveto 50 0 lineto 0 0 50 0 90 arc closepath fill'
4 xxx: '" Dandelion newpath 0 0 moveto 0 50 lineto 0 0 50 90 180 arc closepath fill'
5 xxx: '" Apricot newpath 0 0 moveto -50 0 lineto 0 0 50 180 270 arc closepath fill'
6 xxx: '" Peach newpath 0 0 moveto 0 -50 lineto 0 0 50 270 0 arc closepath fill'
7 xxx: 'color gray 1'
8 fnt: ptmr8r at 50pt
9 xxx: 'ps:gsave'
10 put: 'A'
11 xxx: 'ps:currentpoint currentpoint translate 90 rotate neg exch neg exch translate'
12 put: 'A'
13 xxx: 'ps:currentpoint currentpoint translate 90 rotate neg exch neg exch translate'
14 put: 'A'
15 xxx: 'ps:currentpoint currentpoint translate 90 rotate neg exch neg exch translate'
16 put: 'A'
17 xxx: 'ps:grestore'

```

On the other hand, DVIPDFM and DVIPDFM x have a PDF transformation special for rotation and scaling, etc. Note that literal PDF codes are used in the following example.

`pdf:btrans [the same option as pdf:image]` applies the specified transformation to all subsequent text (supported by DVIPDFM* and DVIPDFM x).

`pdf:etrans` concludes the action of the immediately preceding `pdf:btrans` special (supported by DVIPDFM* and DVIPDFM x).

`pdf:content [PDFcode]` inserts literal PDF code surrounded by a pair of `q` and `Q` to have no effect on the rest of the document (supported by DVIPDFM* and DVIPDFM x).

`pdf:literal [PDFcode]` inserts literal PDF code without a pair of `q` and `Q` (supported by DVIPDFM x * only).



```

1 [page 1 0 0 0 0 0 0 0 0]
2 xxx: 'papersize 2in,2in'
3 xxx: 'color Goldenrod'
4 xxx: 'pdf:content 0 0 m 50 0 l 50 25 25 50 0 50 c f'
5 xxx: 'color Dandelion'

```

```

6 xxx: 'pdf:content 0 0 m 0 50 1 -25 50 -50 25 -50 0 c f'
7 xxx: 'color Apricot'
8 xxx: 'pdf:content 0 0 m -50 0 1 -50 -25 -25 -50 0 -50 c f'
9 xxx: 'color Peach'
10 xxx: 'pdf:content 0 0 m 0 -50 1 25 -50 50 -25 50 0 c f'
11 xxx: 'color gray 1'
12 fnt: ptmr8r at 50pt
13 put: 'A'
14 xxx: 'pdf:btrans rotate 90 scale .5'
15 put: 'A'
16 xxx: 'pdf:btrans rotate 90 scale 2'
17 put: 'A'
18 xxx: 'pdf:btrans rotate 90 scale 2'
19 put: 'A'
20 xxx: 'pdf:etrans'
21 xxx: 'pdf:etrans'
22 xxx: 'pdf:etrans'

```

Up to now we discussed common DVI specials originated by DVIPS in usual. However, there are many PDF specials not mentioned in this section. DVIPDFM originates almost all PDF specials, and its manual [6] is a good source. Moreover, the author discussed in his talk at TUG 2005 [1] how differently the three DVI utilities, DVIPS, DVIPDFM, and DVIPDFMx behave on the same special command.

5 Conclusion

Imagine that one has a DVI file without \TeX source, but he or she wants to modify or to add something to the document. For example, a technical editor may want to put a preprint number on each paper, which was not fixed at the time of writing. He may also want to put a watermark or an emblem on every paper.

We also imagine a \TeX novice who wants to give some decorations in his document, but has some trouble in writing \TeX codes. Is it the best advice for him to learn \TeX ? It might be if he has enough time. If not, DViasm is an alternative. In fact, he may learn DVI commands more quickly than \TeX commands. Even DViasm may be quite attractive to a \TeX expert who wants to modify a few words in his document more than hundred pages long.

DViasm is written for these purposes as supplementary to \TeX and its extensions. It must keep in mind that DViasm is not an alternative program for \TeX .

Neither line breaking nor page breaking is, and will be supported. As mentioned in the beginning of the paper, DViasm is in the first stage. We will discuss in the next paper how to support 16-bit characters in DViasm. Any comment will be helpful to make a better program.

References

- [1] Jin-Hwan Cho, *Practical Use of Special Commands in DVIPDFMx*, TUG 2005 International Typesetting Conference at Wuhan China, <http://project.ktug.or.kr/dvipdfmx/doc/tug2005.pdf>
- [2] Jin-Hwan Cho, *The DViasm Python Script*, <http://svn.ktug.or.kr/viewvc/dviasm/?root=ChoF>
- [3] Jin-Hwan Cho and Haruhiko Okumura, *Typesetting CJK languages with Omega, TeX, XML, and Digital Typography*, Lecture Notes in Computer Science **3130** (2004), 139–148.
- [4] Donald E. Knuth, *The DVItypewriter processor* (Version 3.6, December 1995), [CTAN:systems/knuth/texware/dvitype.web](http://ctan.org/systems/knuth/texware/dvitype.web).
- [5] Geoffrey Tobin, *The DTL Package* (Version 0.6.1, March 1995), [CTAN:dviware/dtl/](http://ctan.org/dviware/dtl/).
- [6] Mark A. Wicks, *Dvipdfm User's Manual* (Version 0.12.4, September 1999), <http://gaspra.kettering.edu/dvipdfm/dvipdfm-0.12.4.pdf>.

[Journal](#)
[home page](#)
[General](#)
[information](#)
[Submit an](#)
[item](#)
[Download](#)
[style files](#)
[Copyright](#)
[Contact us](#)

THE PracTeX Journal



Writing and checking complete proofs in TeX

Bob Neveln and Bob Alps

Abstract

TeX files are text files which are readable by other programs. Mathematical proofs written using TeX can be checked by a Python program provided they are expressed in a sufficiently strict proof language. Such a language can be constructed using only a few extensions beyond the syntax of Morse's book, one being the incorporation of explicit theorem number references into the syntax. Such a program has been applied to and successfully checked the theorems in a significant initial segment of a book length mathematical manuscript.

Bob Neveln got his B.S. in mathematics from Caltech in 1967, (where he learned mathematical analysis from Don Knuth), and received his PhD from Northwestern University in 1976 for a dissertation on the formal language of A.P. Morse. Since 1987 he has held a joint appointment in mathematics and computer science at Widener University. He began using TeX (LaTeX) when he wrote *Linux Assembly Language Programming* which was published in 2000. He is a peace activist and met his wife, Buthaina Hawas, in Baghdad while there in January of 2003 with Academics for Peace. They now have a one year old boy, Ibraheem. Contact him at `neveln@cs.widener.edu`

Bob Alps received a Ph.D. in mathematics from Northwestern University in 1979. His field of study was definitions in Morse languages. Bob has worked as a pension consulting actuary for the Wyatt Company, Sedgwick Noble Lowndes, The Segal Company, and currently for Towers Perrin in Chicago. Bob currently serves as Examination General Officer for the Society of Actuaries examinations in probability and financial mathematics. Bob can be reached at `Bob.Alps@towersperrin.com` or by phone at 312.201.5819.

- [PDF version of paper](#)
- [Comment on this paper](#)
- [Send submission idea to editor](#)

Writing and Checking Complete Proofs in T_EX

Bob Neveln

Bob Alps

Abstract T_EX files are text files which are readable by other programs. Mathematical proofs written using T_EX can be checked by a Python program provided they are expressed in a sufficiently strict proof language. Such a language can be constructed using only a few extensions beyond the syntax of Morse's book [5], one being the incorporation of explicit theorem number references into the syntax. Such a program has been applied to and successfully checked the theorems in a significant initial segment of a book length mathematical manuscript.

1 Introduction

The present work is an unplanned side-effect of a book project by the authors [7]. As work on the book progressed proofs were written more and more carefully. Programs in Python were developed to check the mathematical syntax, then to renumber theorems following insertions or deletions and finally to check the proofs written.

These developments were possible because the book is written in T_EX using a formal mathematical language. Although most mathematical text is intended to be formalizable, usually in terms of first order predicate logic, it is almost never formal as written. Checking proofs written in a conventional style would consequently require a formalization step requiring clarification of the author's intentions on many details. Checking proofs written in a formal language obviates these difficulties. In the work presented here we use a syntax derived from that of A. P. Morse. In his book, *A Theory of Sets* [5], he presented a formal syntax which was used to express all the definitions and theorems in the book. ([6] contains a mathematical exposition of this syntax.) A key feature of his treatment of

mathematical language was the inclusion of definitions themselves into the formal syntax, see [1]. The first theorem of the book was given a complete proof, but no attempt was made to continue the presentation of complete proofs. Indeed with the small set of inference rules given this would not have been feasible.

The formal syntax of Morse's book enabled the creation of a program capable of parsing its language and checking some of its theorems using an expanded inference rule set, as early as 1966 at Sandia Laboratories, [3]. Soon after that most of the mathematics in the book was checked by W.W. Bledsoe working at MIT.

This paper describes additions to Morse's syntax implemented in T_EX and Python programs which together enable writing and checking complete proofs. The resulting environment is a work in progress.

2 Tools and Files

Unix utilities are based on the idea that it is good to have many tools each of which does a single task well. The environment described here to enable writing and checking complete proofs consists of many different T_EX files and Python programs. As noted by Richter in [8], it is easy to write Python scripts which conveniently operate on T_EX files. Those described here include:

- a program which checks the syntax of the mathematics in the T_EX file,
- a program which renumbers the propositions,
- a program which adds horizontal space to mathematical expressions not ideally rendered by T_EX,
- a program which checks a proof whose number is given as a command line argument.

The logic on which proofs depend is supplied in a variety of ways. Some logic is built into the parser which parses $(x < y < z)$ for example as $(x < y \wedge y < z)$. Some logic is built into the checking program which uses the commutative and associative properties of "and" as well as the transitivity of numerous relations including logical implication and set inclusion. Most of the logic resides in a file of rules of inference which is consulted in a blind linear search each time a step of

the proof to be checked is attempted. Another file consists of propositions which are generally recognized as obvious such as

$$(x \in A \wedge A \subset B \rightarrow x \in B)$$

Further logic consisting of material which is at least as elementary, but ordinarily “below the radar” of everyday mathematics is listed in a special appendix added to the work being checked. It uses the logic developed in [2].

The tools in the set create an environment in which a work cycle related to an ongoing paper or book, consisting of steps like that involved in writing a computer program:

1. Add the statement of a theorem and its proof to a \TeX file.
2. Run \TeX to get a viewable DVI file and detect \TeX errors.
3. Run the parser to find mathematical syntax errors.
4. Run the check program to find logical errors and gaps in the proof.

At the very end it is also useful to run a program which uses the parser to add horizontal spacing at points where \TeX would otherwise crowd symbols.

Because all the logical steps which can be checked at this time are quite small, the process is both arduous and tedious.

3 Proof Syntax

The basis for the proof syntax is the mathematical syntax of Tony Morse’s book, [5]. Changes to Morse’s mathematical syntax including additional abbreviation schemes and restrictions on the format of bound variable forms are introduced but do not alter the mathematical language markedly. To get a notation capable of expressing complete proofs just a few additional elements suffice.

3.1 Mathematical Syntax

Morse’s mathematical syntax can be done in \TeX if two main obstacles are overcome.

1. Morse's language contains some special symbols. These can be created with Metafont.
2. It also contains a great many operators with non-italicized Latin letters. In plain T_EX there are 32 of these defined using macros such as

```
\def\sin{\mathop{\rm sin}}
```

We have found the need for hundreds of these. They have been stored in a separate include file.

3.2 Reference Numbering

An important element in the proof syntax described here is the inclusion of theorem numbers themselves into the syntax.

An example from the manuscript [7] follows:

```
\tabc 1.17 $(b\in\bfun \Iff \Patch_0 b\in\U)$
\lineb Proof:
\notea 1$(b \in \bfun$
\linec $\c\Patch_0 b\in \SI \rng b\setdif\dmn b$\By 1.16
\linec $\c\Patch_0 b \in \U)$ $\By 01.14
\lineb
\notea 2$(\Patch_0 b\in\U$
\linec$\c\ex\Patch_0 b$ \By 01.8
\linec$\c b\in\bfun)$ $\By 1.13
\lineb \Bye .1, .2
\lineb
```

In this example a theorem numbered 1.17 is stated and proved. The statement involves the plain T_EX macro ‘\in’ as well as other macros such as ‘\c’ for ‘\rightarrow’ and ‘\Iff’ for ‘\leftrightarrow’. The ‘\tabb’, ‘\notea’ and ‘\By’ macros perform space formatting, but also serve as reference handles for the checking program. For example Theorem 1.16, which is referred to at the end of the second line of the proof, must be identified by a ‘\tabc’ macro. The ‘\lineb’ and ‘\linec’ macros have only a space formatting role. The ‘\Bye’ macro prints QED and indicates that the theorem itself is to be checked.

References such as the closing ‘.1’ and ‘.2’ refer to the notes tagged by the ‘\notea’ macros. The zero-plus references 01.14 and 01.8 point to the file of “obvious” theorems.

Propositions which are referenced must have a traditional number-dot-number identification which is used to invoke them in proofs. This numbering convention is similar to that produced by L^AT_EX but less flexible. It is used instead of L^AT_EX because its use requires slightly less labor and the labor involved in specifying references is a large component of the work of specifying a complete proof.

A Python program is needed for renumbering the theorems when theorems are inserted, deleted, or moved.

3.3 Significant Punctuation

Reference notations may include punctuation. The punctuation marks must be identical to the corresponding mark in the rule of inference. If rules are marked in such a way that rules of a similar nature get similar punctuation, then a meaning is associated with the punctuation mark. The semi-colon for example is used in references that have a major premise followed by minor premises. For example if in note 5 below we prove a result q by using a theorem $(p \rightarrow q)$ which is numbered 1.23 and we have previously obtained p in note 3 then we might have the following note to establish q :

Note 5 ($-a \in \mathbb{Z}$) ‡ 1.23; .3

In order for this note to be checked there must be a theorem 1.23 such as

Thm 1.23 ($x \in \mathbb{Z} \rightarrow -x \in \mathbb{Z}$)

a previous note 3

Note 3 ($a \in \mathbb{Z}$)

and a rule of inference which has the form

From: $(p \rightarrow q); p$

Infer: q

The semi-colon in the reference limits the number of rules which match a given inference. The intended meaning of the semi-colon is that it sets the “major premise” apart from the “minor premises.” At present approximately 190 of the stored inference rules use the semi-colon to separate major and minor premises. Another example of such a rule is the following rule:

From: $(p \rightarrow q \leftrightarrow r); q$
 Infer: $(p \rightarrow r)$

Further developments towards a syntax of reference expressions will no doubt be found useful.

3.4 Given-Hence Blocks

Notes which are not proven but which are merely “given” may be introduced using $\ddagger G$, in place of a proof reference. These remain in force until a “hence” referring to them is encountered. The “hence” attaches the given notes to the “henced note” as explicit hypotheses. The “henced” note is tagged using $\ddagger H$ as a proof reference. For example we might have:

Note 2 $(x \in A)$	$\ddagger G$
..	
Note 7 $(x \in B)$	$\ddagger .2, \dots$
Note 8 $(x \in A \rightarrow x \in B)$	$\ddagger .7 H .2$

The variables introduced in each Given note are local to that block. Reference may be made to notes 2-7 only from within that block, only so long as note 2 is in force in other words.

3.5 Local Definitions

Sometimes it is useful to introduce locally defined variables. To do this we may “set” a variable to a described object. A note of this form is justified by $\ddagger S$ and it retains validity as long as the variables on which it depends do. For example given a non-empty set A it is useful to have a name for a member of A .

Note 2 ($A \neq \emptyset$)	‡ G
Note 3 ($a \equiv \text{an } x(x \in A)$)	‡ S
Note 4 ($a \in A$)	‡ .2,.3

This feature of the proof syntax depends on using a logic which allows descriptions, see [2].

3.6 Reasoning Chains

A note may consist of lines, all but the first of which are introduced by some transitive relation. In this case each consecutive pair of lines defines a step to be given its own proof. The note is then telescoped when used as a reference. For example:

Note 7 ($A \subset B$	‡ ...
$\subset C$)	‡ ...

Here the inclusions ($A \subset B$) and ($B \subset C$) are checked separately, but if note 7 is referred to later, just the inclusion ($A \subset C$) will be invoked by this reference.

4 The Unifier

Each step to be checked is matched against rules of inference in a blind linear search. Each rule whose sequence of arguments and punctuators matches with numerical references and punctuators in the reference note is submitted to a unifier. If a unification is found the step is checked.

The unifier is based on standard first order unification. It does however go beyond standard first order unification in two ways. Although much less general than [4], it allows the terms of a conjunction to be re-ordered in order to accomplish a match. It also attempts to match the second order variables which occur in Morse's language.

It is written to succeed or fail quickly. It may fail to find a unifier even when one exists. For example if a conjunction with n conjuncts is matched against a conjunction ' $(p \wedge q)$ ', where ' p ' and ' q ' are unmatched variables, this unification will not be attempted because of the $(2^n - 2)$ different possible matchings. A rule

of inference must avoid presenting such unifications to the checker or it will be ignored. The unifier does not aim at any ambitious sort of completeness.

5 Results and Prospects

The manuscript being checked contains over 1200 theorems, with proofs in various stages of completion. Roughly the first 120 of these have been checked. This number is increasing as the work progresses.

As the work proceeds, bugs are encountered in the checking program, as well as cases which should check but do not. The program is then revised, rules of inference and “obvious” theorems are stored to the reference files. There are now almost 600 rules of inference and over 400 “obvious” theorems in the zero-plus references file. The checking program now contains about 4000 lines of code. The manuscript also has appendices containing over 150 elementary results which are referenced in the proofs.

The program takes as input the number of a single theorem to be checked. Although Python is an interpreted language, the program executes in a few seconds on a machine of recent vintage.

The proof syntax at its present stage of development is and should be “low-level.” Once avenues of checkable proof begin flowing it will be time for the appearance of higher levels of expression which will attenuate to some extent the labor of picking through all the details of a proof.

Because very elementary facts must be individually listed in a file, mathematics involving significant amounts of computation would not be suitable for this treatment. It is intended for abstract mathematics more or less without restriction. Abstract algebra, or any mathematics which uses sets “endowed” with structure, raises problems of formalization. These problems were encountered in [7] and one approach to coping with them is discussed in Appendix D of this work.

6 Observations

We close with a few observations.

1. Including the details necessary to get a proof to check requires roughly an order of magnitude more time than writing a conventional proof.

2. Proofs stated in checkable detail become longer by a factor of two or three.
3. Reading checkable proofs requires slightly more effort on the part of a knowledgeable reader than reading a conventional proof.
4. Checkable proofs can be read by mathematicians who are not specialists.

7 Conclusion

Despite its preliminary and incomplete nature the checking program as it stands now shows that it is practicable to write and check complete proofs, given a willingness to adopt a formal language and to submit to the discipline of itemizing all necessary references.

Readers interested in trying out these programs may download them from:

<http://cs.widener.edu/proofcheck>

References

- [1] R.A. Alps. *A Translation Algorithm for Morse Systems*, PhD dissertation, Northwestern University, 1979.
- [2] R.A. Alps and R.C. Neveln, A Predicate Logic Based on Indefinite Description and Two Notions of Identity. *Notre Dame Journal of Formal Logic* 22(3) 1981.
- [3] W.W. Bledsoe and E.J. Gilbert. *Automatic Theorem Proof-Checking in Set Theory*, Sandia Laboratories Research Report SC-RR-67-525, July 1967.
- [4] J. Gallier and W. Snyder. Complete sets of transformations for general E-unification. *Theoretical Computer Science*, 67:203–260, 1989.
- [5] A.P. Morse. *A Theory of Sets* Second Edition. Academic Press 1984.
- [6] R.C. Neveln. *Basic Theory of Morse Languages*, PhD dissertation, Northwestern University, 1975.
- [7] Bob Neveln and Bob Alps. *Foundations of the Topology of Manifolds*, (book in preparation).

[8] William Richter. T_EX and Scripting Languages. *TUGboat*, 25(1), 2004.

[Journal](#)
[home page](#)
[General](#)
[information](#)
[Submit an](#)
[item](#)
[Download](#)
[style files](#)
[Copyright](#)
[Contact us](#)

THE PracTeX Journal



Travels in TeX Land: The post-typesetting phase of producing a book

David Walden

Abstract

In this column in each issue I muse on my wanderings around the TeX world. In my columns in issues [2006-2](#) (www.tug.org/pracjournal/2006-2/walden) and [2006-3](#) (www.tug.org/pracjournal/2006-3/walden) I described some methods I use to make it easier to draft large documentation projects (issue 2006-2) and my experience of taking the final steps of turning a book manuscript into a published book (issue 2006-3).

In this column I give a final report on the book project I last discussed in TPJ issue 2006-3, discuss "self-publishing" at some length, and mention some of my other recent activities in TeX Land.

Dave Walden is retired after a career as an engineer, engineering manager, and general manager involved with research and development of computer and other high tech systems. More history is at www.walden-family.com/dave. He may be contacted at dave@walden-family.com

- [PDF version of paper](#)
- [Comment on this paper](#)
- [Send submission idea to editor](#)

Page generated March 14, 2007 ; [TUG home page](#); [search](#); [contact webmaster](#).

Travels in T_EX Land: The post-typesetting phase of producing a book

David Walden

Abstract In this column in each issue I muse on my wanderings around the T_EX world. In my columns in issues 2006-2 (www.tug.org/pracjournal/2006-2/walden) and 2006-3 (www.tug.org/pracjournal/2006-3/walden) I described some methods I use to make it easier to draft large documentation projects (i.e., books) and my experience of taking the final steps of turning a book manuscript into a published book.

In this column I give a final report on the book project I last discussed in TPJ issue 2006-3, discuss “self-publishing” at some length, and mention some of my other recent activities in T_EX Land.

1 Introduction

Mostly *The PracT_EX Journal* discusses how to use the various T_EX-based systems. However, presumably we often use T_EX in order to actually get something published and distributed. This column primarily deals with the post-T_EX (but equally important) phase of the publishing cycle.

2 Picking up where I left off

I ended my column in issue 2006-3 saying the following about North American and European distribution for my book, *Breakthrough Management*, that had recently been published in India.

With the book published in India, my next problem will be printing and distribution in North American and Europe. Various issues come to mind:

1. *Who should the publisher-in-name be (I will be the actual publisher)? Should I ask the Indian publisher to keep its name on the book for this hemisphere, should I seek another industry consortium, or should I make up my own publisher name? There are also related ISBN number issues.*
2. *Will I have to resize the text block (and reformat the pages), or will I want to improve the overall typesetting in light of what I see in the Indian printing? In particular, will the Minion font I chose for my main text reproduce well using the printing technologies of print-on-demand printers?*
3. *I believe that by printing via a print-on-demand company (e.g., Lightning source), I will have distribution (via Ingram) to U.S. on-line book stores (such as Amazon.com and bn.com). Will connecting to Lightning Source's UK group also provide distribution via on-line book stores in Europe?*
4. *I have the option of selling books direct from my website in the United States (www.walden-family.com/breakthrough). What parallel arrangement might I make for Europe?*

Here is what happened.

1. I did continue to use the Indian publisher's name and ISBN number, for now. This seemed easier and took a shorter time than dealing with the issues relating to finding a new publisher.
2. The book was printed in the United States with no change in text block size.
3. Primarily because of fears that photographic images would not reproduce adequately well using Lightning Source for print-on-demand (fears that came from the participating in the Yahoo-based Self-Publishing discussion group), I had the book printed by a more or less local printer. Lightning Source is purportedly upgrading its printing technology now, and I may well issue a version of the book via Lightning Source within a few months, which should also then make the book available for sale by Amazon.

4. I have been selling the book via my website using PayPal to collect credit card payments and shipping copies to buyers in the United States and overseas. This has required lots of study of the United States Postal Service rate structure which seems to be made up of many discontinuous fragments. However, purportedly (again from the Yahoo discussion group) many potential purchasers of books fear dealing with independent sellers — people mostly want to buy from Amazon directly. This is another reason for eventually moving to printing by Lightning Source (or finding another form of retailer who people are not afraid to buy from at a distance).

3 Ease of printing in the digital age

In general, I was pleasantly surprised with how easy it was to have a book printed that was prepared in \LaTeX . I sent the PDF output of \LaTeX to the printer in India via the Internet, and the book was printed without any additional interaction by me except to check a printer's proof (which was fine).

In the United States, I requested by email quotes from eight printers using the general format suggested in Pete Masterson's book, *Book Design and Production: A Guide for Authors and Publishers* that is aimed at small publishers and self-publishers. I received several plausible quotes by return email. One of the geographically closest printers (Ames On Demand of Somerville, Massachusetts, about 60 miles from my home on Cape Cod) also had nearly the best price. I phoned him and asked him two questions: (1) I told him I already had a ready-to-print PDF and wondered if I would have to make any adjustments to my page layout (e.g., text block size) for him to print the book on his presses; (2) Could he send me an example of a photographic image that had been printed on his presses (Xerox 6180 for the black and white text and iGen3 for the color cover) so I could review the reproduction quality.

The printer suggested that I send him my PDF file and he would send me back a proof. I was thinking he would print *a page* with a photographic image and send it to me. I was happily surprised when the over-night delivery truck arrived two days later with a finished, bound proof of the whole book including

my cover art.¹ Obviously this is an advantage of digital printing. Since finished copies are being printed a page at a time rather than on large offset sheets with multiple different pages on each sheet, it is easy for the printer to run the whole book through his digital printer (just like he would do successively for hundreds of copies).

The proof sent by the printer looked great. I asked him to slightly shift the title on the book spine (I sent him an adjusted copy of the cover art), to slightly shift the text block on the page to increase the inside margins and decrease the outside margins (he was able to do this without me touching anything in L^AT_EX), to bind the book going forward using a matte rather than glossy coating, and to make a one line change to the back-of-the-title page (I provided a single new PDF page which the printer used to replace the previous page in my whole-book PDF file). He did these things, a whole-finished-book proof arrived at my home a few days later, and I gave him the go ahead to print 250 copies of the book (his quoted price was the same per book for volumes of 250, 500 and 1,000 books, so it was an easy decision to print the minimum of the quoted number of copies). A few more days later, I picked up the printed copies at the printer's loading dock with my little pickup truck (I did the pickup myself to save shipping costs).

The printer says he will keep my PDF files as he has adjusted them and can print more copies at any time.

4 Self-publishing more generally

Communications and transportation technology is leading to disintermediation (elimination of middle men) in many fields. Publishing is no exception. Self-publishing is becoming an increasingly feasible option. The rest of this section sketches some of the things I believe I have accurately learned about self-publishing as an alternative to working with a traditional publisher (but see the note about self-publishing resources at the end of this section).

1. I created my cover art using Adobe Illustrator. Yuri Robbers' paper in this issue describes creating cover art using PSTricks (from the T_EX collection of tools) as well as more general issues of cover design.

4.1 Using a traditional publisher

If you are an author and you want massive PR and bookstore distribution for your sales, then you need a traditional publisher. Except, they won't give you massive PR unless they think your book is going to be wildly popular or you are already a wildly popular author.² You should also seek a traditional publisher if you want someone else to foot the development bill (editing, illustration, permission, design and layout, indexing, printing, and perhaps an advance for you). Except, it is hard to get a traditional publisher interested in you (you probably need an agent or personal contact). (Of course, the amount of effort to self publish is considerable; if your book has a decent chance of making a profit, putting that same amount of effort into finding a publisher rather than into self-publishing may well result in you finding a publisher.)

If you succeed in getting a traditional publisher, the publisher is going to want you to sign a contract that gives it the worldwide rights in all media, and your work may well be tied up so you have to ask the publisher's permission to reuse a chapter elsewhere. For books on specialized topics (e.g., scientific, medical, technical, or professional), the publisher may ask you to promise to buy some minimum number of copies (e.g., 1,000) to cover their costs. If the book is no longer selling many copies (i.e., is essentially out of print), you will have to ask the publisher to revert the rights to you, and there is a possibility that the publisher will refuse (with the possibility of print-on-demand, I worry publishers may choose this option more often and to keep the book technically in print).

Working with a traditional publisher, you may get 10 percent of the gross (the publisher's discounted sales price, e.g., 55 percent off list price) after returns (in the publishing industry in the United States it is apparently traditional that book stores can send back for full credit all the books they ordered even if they are damaged). Thus, a \$30 book will sell wholesale for under \$13.50 and you will get perhaps 10 percent of that.³

2. Dan O. Snow, coauthor with Dan Poynter of *U-PUBLISH.COM 4.0: A 'Living Book' to Help You Compete with the Giants of Publishing* has a slightly different point of view. In an email of February 24, Dan said, "[T]here are nearly *ten times* more outlets for books than bookstores...and they are easier to target, usually pay more, pay faster, and return fewer (if any) unsold books.

3. The 10 percent figure I use in this paragraph is by way of example. Marion Gropen, who

4.2 One alternative to a traditional publisher

One alternative to a traditional publisher is vanity or subsidized publishing where they make their money having you pay for their services, you get a few books, and that's the end of it. They may have a website where your book is sold, but you won't get a very big share of the price. Also, they are the publisher of record, which ties you up in various ways. This is the worst option: you give up control to someone else and can't get much back for it, although it may be the right option for someone who just wants to print a few copies of a book to give to family members and friends.

4.3 Self-publishing—a better alternative

The other alternative to a traditional publisher is self-publishing. In self-publishing you do the development work yourself or hire someone to do the various parts for you. You get the ISBN numbers (approx. \$300 for a block of 10 in the United States—see <http://www.isbn.org>⁴), and you control how you get the book printed and by whom, the distribution, and what limited rights you give to other people as it is beneficial to you.

My guess is that the development cost of the typical self-published book done with high quality is something in the vicinity of \$4,000 to \$6,000 (hiring an editor, an illustrator, a design and layout person, permissions, and an indexer), although I suppose it could easily reach \$10,000. Of course, you can decide to

is an expert on the publishing business, in a January 18 email said:

“Royalty rates vary substantially with the type of book. Small houses may pay different rates than the norm; but for larger houses, the norms are practically carved in stone. For example, a trade non-fiction hardback author gets 10 percent of list price (also known as the suggested retail price) for the first 5,000 copies sold, net of returns. The next 5,000 copies yield 12.5 percent of list. And thereafter, the author gets 15 percent of list. Advances are usually calculated to cover something like the expected earnings for the first 6 months of the title's life, although this varies widely.

“Mass market fiction tends to run from 5 percent of list to 8 percent. Trade paperbacks are usually between 7 percent and 10 percent. The break points at which the rates step up vary with formats and market segments.

“Scientific, medical, technical, professional, and academic publishers generally pay upon net sales (gross sales after discounts and returns). Again, rates and breakpoints will vary depending upon format and market segment.”

4. You will have to Google for the ISBN agency for your country.

do some of that yourself if you have the skill or interest. (I did the design and layout for my current book myself because I wanted to learn how to do it and wanted to experience “going all the way” with L^AT_EX)

4.4 Printing options

There are several printing options: *(a)* traditional lithography which only becomes economical at 1,000 to 3,000 copies, but it is probably necessary for books that have lots of color or non-line-art that you want to print well (line art works fine without lithographic printing); *(b)* short run lithography which is not economical on a per book basis but may be the way to go if you need the quality even though you may think you cannot sell more than a few hundred books; *(c)* digital printing (e.g., sort of like your home laser printer but fast) which can be done very well or not quite as well depending on the care that is taken (I have seen very fine black and white photos done with digital printing). Print-on-demand (POD), where you can buy one to hundreds of copies at a time, typically uses digital printing.

There are POD printers, e.g., lightningsource.com and lulu.com,⁵ which will take your PDF file of the text of your book and the PDF of your cover, charge you a modest setup fee, store your book electronically on their computers, and print one or many copies for you at a relatively fix price per copy, e.g., \$6, whenever you order them. If I had a book without photographic or fine art images (e.g., a management book or a novel), this would be a good option.

Books printed by Lightning Source Inc. (LSI) also reach wholesaler catalogs (particularly that of Ingram which is a sister or parent company of LSI) such that anyone can order the book from the wholesaler.⁶ You set the list price and the discounted price, and LSI fills orders as they come in, e.g., from a retailer such as Amazon, and basically sends you the difference between your discounted sales price and their printing cost. If you list the book at \$30 and discount it to a wholesale price of 50 percent, they will send you the difference

5. lulu.com is not exactly self-publishing, but this is an inexpensive easy way to get some books in print. For instance, this is the approach TUG president Karl Berry has used to make his *Eplain* and *Fontname* documents available in bound hardcopy format.

6. However, my understanding is that self-published books printed by LSI typically do not get in a hard copy Ingram catalog that book stores look at as part of deciding what to order.

between \$15 and the price they charge you to print a copy for each one that is sold, which will probably net you something like \$10. Note that Amazon, etc., will now list your book but the book stores still will not carry it except to order it when a customer prepays because they insist on being able to return books they order for their book shelves. If only few people order your book from Amazon, Amazon will show a multi-day shipping period which may discourage buyers, but if your book starts to sell well and you offer a wholesale discount Amazon thinks is appropriate (e.g., 55 percent), then Amazon may then begin to inventory it and list a shortened delivery time.

The other alternative is just a regular printer (big or small) who gives you the best terms when you ask a few for quotes.

4.5 Distribution

I use the word “distribution” in this paragraph in an informal sense, not in the publishing jargon sense. (In publishing, a distributor is a business that typically has an exclusive contract with the publisher (e.g., you), to find places that will sell your book.)

In the case where you use a regular printer, you can sell the book yourself (e.g., as I do via my website), you can consign it to someone else (e.g., your local historical society to sell), or you can join Amazon’s Advantage where you ship the books to them for inventory and they list and sell the books and take a commission. In another Amazon program (Marketplace), you inventory the book, they list it, they sell it and send you the order, you fulfill the order, and Amazon sends your share of what the customer paid (e.g., minus their sales commission). (You can also hire order fulfillment houses who will charge you \$3 or \$4 per book, i.e., you take in the money, you send them the order plus \$3 or \$4 plus postage for them to mail the book.)

You also can do combinations of the above. For instance, have Lightning Source print the book which will be wholesaled by Ingram (the largest commercial wholesaler in the United States) or Baker and Taylor (the largest library wholesaler) and simultaneously have the book printed by an another (perhaps higher quality) printer or buy copies from Lightning Source and sell them yourself one way or another.

Notice that Lightning Source has a branch in the UK which is connected

with the big European wholesalers such as The Bertram Group (www.bertrams.com) and Gardner's Book Service (www.gbsbooks.com). Amazon also has branches in other countries and their Amazon Marketplace (where they list the book but you fulfill the order) also permits you to sell and ship overseas.

Finally, since you control the book when you self-publish, you have flexibility to deal with multiple entities to "publish" the book rather than all rights being tied up with a single traditional publisher, although using multiple "publishing" paths may well not be the best overall marketing strategy. Nonetheless, you could let your local historical society be the "publisher" for your local history book, changing the title page to list them as publishing and having them get their own ISBN number which you will put in your electronic file, selling them copies wholesale, and letting them sell the book locally or to people who order it from them. Simultaneously, you could publish the book yourself (using your own ISBN number) for web-based sales to whomever orders the book from you.

4.6 Disintermediation and flexibility

The point of all this is that once you decide the traditional publisher is not for you and you prepare a ready-to-print file for your book, then, in return for your initial investment, you can control everything and make whatever deals you want, and modern printing and distribution technology offers many options you can use.

Of course, there are advantages in many situations to working with a traditional publisher. The advantages include the publisher's distribution channels, access to the publisher's editors, indexers, typesetters, art department and established distribution channels, the publisher's payment of the development costs of the book, and not going against the established model for how a book gets published. In my view, the major disadvantages of going with a traditional publisher are (a) the publisher ties up all the rights and the author loses control of his intellectual property, and (b) it is often hard to get a traditional publisher to take you on and let you produce the book you want to produce.

In any case, my experience has made it clear to me that it is now entirely feasible and relatively inexpensive, in cases where one is willing to forego the advantages of a traditional publisher, to self-publish. T_EX, L^AT_EX, ConT_EXt,

etc., are available for free for anyone who already knows one of them or is interested in learning to use one. PDFs (easily output by $\text{T}_{\text{E}}\text{X}$, et al.) appear to be a nearly universal way of transmitting ready-to-print manuscripts to a printer; even when the printer uses traditional offset printing with many pages per large sheet of paper, in my experience the printer takes care of whatever is required to turn a sequence of pages in a PDF file into many pages on a sheet. And capabilities like PayPal, Amazon, rapid international shipping, and website-based selling make it possible to sell a book world wide.

4.7 Marketing, promotion, and good business

Of course, practically speaking, you still have to promote the book in order to get people to want to buy it.

Marketing and promotion involve all of the things that happen with traditional publishers. If you hope to have the book reviewed, you must send out review copies, typically in advance of publication. Parallel short publications and presentations can help sales. Sending out some sort of notice to targeting mailing lists might help. Having good positive comments about your book on Amazon.com should help. And so on. Dan Poynter's (noted in the next section) has good content of promotion of self-published books.

Probably you are no more likely to make money self-publishing a book than you are to make money with a book published by a traditional publisher. Most books do not make a lot of money and most authors are not rich. However, since you may be putting up your own funds to develop and initially print a self-published book and may incur other liabilities, you do have to think about it in a businesslike way. You have to understand the difference between fixed and variable costs and the break even point. You probably don't want to invest in a big printing run before you have strong evidence you are going to sell a lot of copies. If you expect to sell a material number of books at a profit, you need to learn how normal businesses operate (e.g., approaches to limiting liability, registering as a business in your state, etc.) so you don't accidentally get into trouble.

If your book does become popular, you may make more money publishing it yourself, and many people apparently do make money self-publishing. Also, by self-publishing and retaining all the rights, you are free to later make a

deal with a traditional mainstream publisher, if one becomes interested after seeing the popularity of your book.

4.8 Resources for learning more about self-publishing

While I think I have presented some relatively accurate information about self-publishing, I am no expert. Here are a few useful books by experts on self-publishing:

- Dan Poynter’s book *The Self-Publishing Manual* has gone through many revisions; he also has a useful website: www.parapublishing.com/.
- Pete Masterson’s book (mentioned above) on *Book Design and Production, A Guide for Authors and Publishers* is more about the nuts and bolts of publishing (not the writing and promotion like Poynter’s book); he also has a useful website (<http://www.aeonix.com/>) including lots of useful lists, e.g., printers, book coaches, small publishers, etc.
- Morris Rosenthal has a good small book on Print-on-Demand Book Publishing; his website (<http://www.fonerbooks.com/contact.htm>) also reprints from his book the text on the economics of POD versus traditional publisher.
- Robert Bowie Johnson and Ron Pramschufer’s book *Publishing Basics* is a good, short, very basic introduction to things. Pramschufer also has a related business and useful website (where you may be able to get this short book for free): <http://www.selfpublishing.com/>.
- Aaron Shepard’s book *Aiming at Amazon* will be valuable reading for anyone hoping to sell his or her book via Amazon and similar companies.
- Another interesting website is <http://www.gropenassoc.com> — click on “Reference Desk”; this is run by one of the three people who moderates the Yahoo Self-Publishing discussion group. Among other things, she recommends several books on the business side of publishing that I only touched upon in subsection 4.7.

You can also hire people to help you with any of the aspects of self-publishing including hiring a “book coach” to guide you through the process.

I think one of the best things to do to understand self-publishing is to subscribe to the Yahoo Self-publishing discussion group and read the messages that go by for a few weeks. All of the people whose books I just listed and many other very knowledgeable people participate in this list. When it comes to discussing typesetting programs, the Yahoo Self-publishing discussions focus on InDesign, QuarkXPress, and other commercial layout and typesetting programs. Use of Microsoft Word for typesetting and layout is frequently denigrated in this discussion group, although Aaron Shepard has a book on how to use Word for these functions. Use of T_EX is not a popular topic of discussion within this group although its use is regularly encouraged by one of the group's three moderators (John Culleton — <http://tug.org/interviews/interview-files/john-culleton.html>); John himself does not use L^AT_EX, which he finds too confining, and instead uses ConT_EXt and T_EX itself. T_EX is sometimes mentioned by a few more of us in the discussion group who are (primarily) L^AT_EX users.

5 A few graphics issues

This issue of *The PracT_EX Journal* is primarily focused on graphics. Issues I had to think about in getting my book ready for printing were adjusting photographic images so they would print well and permissions to use photographic images.

I used Adobe Illustrator to create line drawings for previous management books and did so again for this book, particularly since I reused a few of the illustrations from previous books. These illustrations exist as EPS files which I have typically compiled into L^AT_EX using `\includegraphics`. For this book I used PDF_T_EX to compile my L^AT_EX files and thus had the option of using PDF files out of Illustrator. However, for reasons I don't understand, things always worked better (for instance, useful bounding boxes occurred automatically) when I saved an EPS file from Illustrator and then used `epstopdf` to convert the EPS file into a PDF file for inclusion by PDF_T_EX.

In addition to line drawings, my book also had quite a few photographic images either of real life people and scenes or of well known paintings. A number of the real life images were taken by my co-author, and therefore no

additional permission was needed from the photographer. In one case, we used a photographic image from circa 1933 by American photographer Doris Ullman. The rights to this photograph are owned by the University of Oregon, and we had to pay a use fee to the university to use this photograph in our book. We wanted to use a photographic image of a famous painting by Norman Rockwell (*The Runaway*, which appeared on the cover of the September 28, 1958, cover of the *Saturday Evening Post*). However the Rockwell family member who controlled the rights to this image apparently had a policy against granting reuse rights to self-published books, and we gave up trying to secure permission to use this image. Based on our failed attempt to use the Rockwell image and another previous experience, it appears to me that people dealing with reuse rights are fixated on dealing with traditional publishing companies. If a painting is not itself still under copyright protection (as the Rockwell painting apparently still is), the Bridgeman versus Corel decision http://en.wikipedia.org/wiki/Bridgeman_Art_Library_Ltd._v._Corel_Corporation seems like important reading, at least for books published under U.S. law.

Having decided to include black and white versions of color images in the book, the next question was how to adjust them so they reproduced well, particularly on the digital printers used by many print-on-demand (POD) printing companies. I received much advice from the Yahoo Self-publishing list, of which two bits were particularly relevant and useful. In an August 8, 2006, post to Yahoo Self-publishing, Ralph Alcorn advised:

I assume your original is color.

1. Sharpen a little bit—go to filters>unsharp mask and use 100% .5 pixels 0 threshold to start.
2. Go to adjustment>channel mixer and check monochrome. Fool around with the red green and blue sliders until you get something with strong contrasts that you like.
3. Shadow/Highlight is another adjustment that can make a dramatic difference if your original is very dark.

The printing process is going to make your print a little darker. At worst, POD should do as well as your B&W laser printer, so do something that looks good even on your own printer.

In an August 16, 2006, post to Yahoo Self-publishing, Walt Shiel said in response to my query:

Get a copy of *Print Publishing: A Hayden Shop Manual* by Donnie O’Quinn (ISBN 0-7897-2102-3) and read the section “Scanning and Adjusting Halftones” in Chapter 10. I think this book is [out of print] but used copies are readily available — we got ours from one of the Marketplace sellers on Amazon.

Then be sure to get a proof copy, so you can make additional adjustments, if necessary, before going final.

6 A few other stops on my wanderings through T_EX Land

This column in the previous issue was about text editors rather than about T_EX. And, since last issue, I have hardly used T_EX/L^AT_EX at all, despite being indirectly involved with T_EX on a daily basis:

- I have been dealing with all these post-L^AT_EX publishing issues I have just described.
- My TUG interviews (tug.org/interviews) are done with plain text email and HTML.
- I spent many hours writing Perl code to generate alphabetic author, title, and topic lists for *TUGboat* (<http://www.tug.org/tugboat/contents.html>).
- I paid an OCR specialist to scan and use her OCR program to turn a hard copy manuscript that was created on a typewriter in 1982 into a plain text file that I will turn into L^AT_EX as part of republishing a family oral history.

I am doing the first draft of this column on December 31, 2006. Rather than continuing to work on “peripheral” matters in the coming year, I hope to spend most of my time on actual T_EX use — republishing the above mentioned oral history and developing and publishing a couple of other books. Thus, T_EX

itself and the issues I run into in using it should be the subject this column once again.

In fact, as I was finalizing this column (February 2007), I stumbled upon the Antykwa Toruńska set of fonts. I recently joined the Dutch T_EX users group (NTG, <http://www.ntg.nl/english.html>) after discovering that a significant proportion of the articles in NTG's journal *MAPS* are in English. This led me to the Antykwa Toruńska set of fonts (J. Nowacki, "Antykwa Toruńska ver. 2.03," *MAPS*, voorjaar 2001, pp. 119–132⁷). I tried changing the font of this article to Antykwa Toruńska by saying `\usepackage{anttor}`, but T_EX told me it couldn't find an `anttor.sty` file. So I searched for Antykwa Torunska on CTAN and found nothing. Next I searched the TUG website for Antykwa Torunska where I found <http://www.tug.org/tex-archive/fonts/antt/tex/latex/antt/anttor.sty>. (Karl Berry says that `anttor` will be included in the next distribution of T_EX Live.) I saved this style file in the directory of this column and compiled the column again and it came out as you now see it (on my Windows XP system). (My next step will be to put this style file in my `localtexmf` directory and to give the MiK_TE_X command to update the file name database.) I can imagine using the Antykwa Toruńska fonts in the future for some serious purpose in some special document rather than just playing with it as I have done here.

Acknowledgements

The guest editor of this issue, Yuri Robbers, made many helpful suggestions. John Culleton reviewed the content on self-publishing for major errors. Marion Gropen gave me many especially useful suggestions for subtle improvement of the sections on self-publishing. Karl Berry reviewed section 6 for content and spotted many typos throughout the paper. The anonymous reviewers provided helpful corrections, as did editor Lance Carnes.

7. I'm not sure if this paper was previously published by the Polish T_EX users group, GUST.

Biographical note

David Walden is retired after a career as an engineer, engineering manager, and general manager involved with research and development of computer and other high tech systems. More history is at www.walden-family.com/dave.

[Journal](#)
[home page](#)
[General](#)
[information](#)
[Submit an](#)
[item](#)
[Download](#)
[style files](#)
[Copyright](#)
[Contact us](#)

THE PracTeX Journal



The "hacking for learning" paradigm in LaTeX — Some thoughts by a long-time LaTeX user

S. Parthasarathy

Abstract

This article argues a case for making hacking an accepted way of learning. It uses the example of LaTeX to show why hacking is not so bad as it is made out to be. It also gives some warnings on the down-side of hacking-for-learning.

Parthasarathy teaches discrete mathematics to undergraduate Computer Science students, at Hyderabad, India. He has branched into full-time teaching after a 25-year stint in the software industry. His association with LaTeX started in 1993, at the United Nations University, International Institute of Software Technology, Macau, where he worked with Prof. Dines Bjorner. His website <http://algolog.tripod.com/nupartha.htm> will give more specific details about him.

- [PDF version of paper](#)
- [Comment on this paper](#)
- [Send submission idea to editor](#)

Page generated March 14, 2007 ; [TUG home page](#); [search](#); [contact webmaster](#).

The “hacking for learning” paradigm in L^AT_EX — Some thoughts by a long-time L^AT_EX user

S. Parthasarathy*
drpartha@gmail.com

Abstract This article argues a case for making hacking an accepted way of learning. It uses the example of L^AT_EX to show why hacking is not so bad, as it is made out to be. It also gives some warnings on the down-side of hacking-for-learning.

1 Preamble

For several years now, traditional teaching, or learning, has been based on some well-known and accepted practices of instruction. Such methods have been continuously studied, and improved, using theories of cognitive sciences, psychology, pedagogy etc. Recent technologies have introduced a new way of learning, which proves to be far more effective than all traditional methods. This method of learning – known as **hacking** – has become a practical way to learn, although many traditional schools still do not recognise its effectiveness.

1.1 A question of definition

We will not launch yet another polemical debate on the meaning of hacking. The term has several usages, and meanings with several shades. In fact, a quick consultation of Wikipedia shows that there is a plethora of definitions of the term **hacking**, and a whole lot of controversies regarding its usage. The term, **hacking**, is often misunderstood, and used unfairly, in a derogatory sense. The

*Algologic Research and Solutions, 78 Sancharpuri Colony, Bowenpally, Secunderabad 500 011, India WWW-URL – <http://algolog.tripod.com/nupartha.htm>

“*New Hacker’s Dictionary*, third edition” by Eric S. Raymond and published by MIT Press in 1996 is a highly relevant work, since it defines “hack” and related words based on their initial use: 1) making a quick (perhaps dirty) change to a system, and 2) doing something incredibly tight and elegant. In simple terms, **hacking** is the act of *ripping open an existing product, and modifying it, to examine its working*. In this article, we will use the term **hacking** in this sense. *Thus, this is not a universal definition of the term*. The question of any motives, or intentions, is deliberately kept out, and is not relevant in this definition. For this same reason, the author feels that the term *ethical hacking* is an unnecessary redundancy.

Hackers have unfortunately earned the reputation of being crooks, or outlaws, who snoop into other people’s privacy, and exploit them. This is often true, but not always. The French have a nice word for this class of people – *bricoleurs*. We could also call them – *explorers*. In fact, FLOSS celebrities like *Linus Torvalds* or *Richard M Stallman* take pride in being labeled as hackers. Our definition given above, also follows this trend.

Hacking has become a culture, and a way of life, for people who have to depend on self-tutoring. Hacking is a very effective means of learning several subjects e.g. scripting languages like Python, and Ruby, HTML, Linux, shell scripting, and \LaTeX in a way, hacking is also used by people who wish to learn playing music. They pick up a known tune, score, or song, and try to play it by themselves. With some amount of trial and error all these people succeed reasonably fast.

The aim of hacking in \LaTeX is to get a feel for the commands used in the original text. The \LaTeX hacker is not interested in the contents/text. Since the original text will not be pilfered, the hacker need not feel unduly guilty.

Those who wish to protect their \LaTeX documents from hackers, always have the possibility of first creating the \LaTeX text, and then converting it into a closed format, like PDF. Thus, hacking poses no major danger to \LaTeX users. When a closed format, like PDF is used, even the commands are masked. Only the rendered version will be visible.

1.2 \LaTeX and hacking

\LaTeX is eminently suitable for the "hacking for learning" paradigm.

1. \LaTeX is a very rich language, with a very complex command repertoire. The

- learning curve for L^AT_EX is very flat. We could make it steeper, by hacking.
2. GUI tools and IDEs like Kile or Lyx can offer only very limited support and prompting.
 3. Hacking helps the learner to figure out how exactly to use a command. It is easy to use hacking, to verify what exactly a command does, or get answers for "what-if" type of questions. All that is needed, is to re-compile the source and view the results. This experimentation is often much faster, and more effective than the task of actually referring to lengthy books and manuals.
 4. Professional typesetting is a very profound subject. Many aspects of this subject are not easy to visualise. In addition to the commands, L^AT_EX texts depend on supporting resources which are "imported" by the user. This includes, documents classes, style sheets etc. The best, and the only way to appreciate these, would be to use them on some hacked text and try out various commands.
 5. As the learner gains confidence in L^AT_EX commands, the need to hack will gradually go down. Hacking is not an addiction in this case.

One must recall that a person is not always obliged to depend on others for his hacking needs. The learner can hack his own documents also (auto-hacking). Once a person has created a good document which suits his tastes, he can preserve it as a template document. He can build on this document, and similar template documents (containing other commands) and create richer documents. This will save considerable time for the user. The user can gradually build up his L^AT_EX skills using this approach.

2 The downside of "hacking for learning" approach

The author insists that hacking is only a supplement, and *is NOT a replacement for traditional learning methods*. Hacking must be used to reinforce traditional learning modes, and to accelerate the pace of learning. However, the following points must be kept in mind:

1. The learner may get a false sense of having mastered L^AT_EX due to the short-term success he sees.

2. The reason for using a certain command, is influenced by the context in which that command is used. This will be known to the learner only if he supplements his efforts with some reading and profound analysis.
3. The unscrupulous hacker may go beyond just innocent hacking.
4. Hackers have a habit of jumping to a conclusion, without actually thinking about the consequences and side effects. Excessive dependence on the "just do it" attitude will make hacking-for-learning a counter-productive exercise. The hacker may actually be losing time, instead of gaining time, because of this.
5. In the L^AT_EX world, like everywhere else, there are good programmers, and there are bad programmers. If the source document used for hacking is badly chosen, the hacker will have extra trouble trying to figure out, why things are done in a certain way. Or, he may end up learning the wrong way to use a command.
6. Hackers should possess enough discretion to recognise texts which do not need the help of hacking (like the current paper). In many cases, the prompting offered by front-ends like Lyx or Kile is enough. In such cases, **not using hacking** would be a faster way to create L^AT_EX documents.

3 Summing-up

The message this article tries to convey, is that hacking should not be denounced systematically. A certain proportion of the learner's time should be allotted for hacking existing L^AT_EX documents, so that the learning process becomes faster. The learner's confidence builds up when he sees success.

An overdose of anything can be lethal. Hacking, when practised in moderation, *along with other forms of learning*, can be an excellent way of jumpstarting the learning process for L^AT_EX . The author, who is a confirmed teacher, and a compulsive hacker, should know.

A sequel to this article will discuss certain principles of hacking, for L^AT_EX learners.

4 Postscript

This article was prepared under Suse Linux 10, using KDE-3.4.2 b, and the Kile 1.8 front-end. Hacking was not necessary, in view of the simplicity of the text.

* * * * *

[Journal home page](#)
[General information](#)
[Submit an item](#)
[Download style files](#)
[Copyright](#)
[Contact us](#)

THE PracTeX Journal



Ask Nelly: How do I create European style spacing within numbers?

The Editors

Abstract

Ask Nelly is a question and answer column. Nelly is the quiet person who sits at the back corner desk, who knows a lot, and when asked any question is always ready with a patient answer. If Nelly doesn't know the answer, Nelly will know an expert who has the answer. Feel free to [Ask Nelly](#) about any aspect of LaTeX, TeX, Context, etc.

- [Comment on this paper](#)
- [Send submission idea to editor](#)

TPJ

Q: Dear Nelly: When I use US style separators for thousands or decimals in LaTeX math mode, everything looks fine. When I use European separators, however, the spacings are all wrong. How can I fix this?

A: This is indeed a problem, as can easily be seen by typing:

```
\begin{eqnarray}
  3,141,592.654
  3.141.592,654
\end{eqnarray}
```

The first rendition, with US style spearators, gives correct spacing, but the second one, using European style, has too little space between the thousands, and too much at the decimal comma.

This difference is caused by the definition of the "." and "," in math mode. The symbol "." is defined as type "mathord" and the symbol "," as "mathpunct". This needs to be changed around for European style numbers. In case one has only a few such numbers in the document, use one of these two ad-hoc solutions:

```
\begin{eqnarray}
  3\mathpunct{.}141\mathpunct{.}592\mathord{,}654
  3\mathpunct{.}141\mathpunct{.}592,654
\end{eqnarray}
```

A document-wide change can be achieved by using these definitions in the document preamble:

```
\DeclareMathSymbol{.}{\mathpunct}{letters}{"3A}
\DeclareMathSymbol{,}{\mathord}{letters}{"3B}
```

Which is the reverse of their usual, US-style, definitions.

The above question was answered by **Yuri Robbers**, a member of the editorial board of this journal. He can be reached at yuri.robbers@gmail.com

TpJ

The answer given by Nelly (Yuri Robbers) to the problem of setting US or European style digit separators is correct as long as you don't care the correct typesetting of such entities. When I say "correct", I mean "according to the ISO regulations". I would be a blind man if I pretended everybody is following the ISO regulations, but I would assume that the requested separators may be tolerable in an informal handwritten letter, even in an invoice, but certainly not in a typeset book or a report (especially if typeset with our favourite software!).

Besides establishing the type of decimal separator (the decimal point for English texts and the decimal comma for text in other languages), the ISO regulation expressly forbid any other "inked" mark within the digit triplets; the only allowed separator is a thin space, therefore the example would be

```
\begin{eqnarray*}
  3\,141\,592.654 \\
  3\,141\,592,654
\end{eqnarray*}
```

The problem then is how to make the right spacing around the comma or around the point when they are used as decimal separators.

No problems for the point because the default settings in (La)TeX are correct.

In languages where the comma plays the role of a punctuation mark *and* of a decimal separator, it is necessary to exploit a specific package *icomma*, to be found at CTAN://macros/latex/contrib/was that defines the "intelligent comma". This same folder on CTAN contains several other little packages by Walter Schmidt,

among which fixmath, which adjusts several settings for typesetting mathematics to comply with ISO standards.

Alternatively it is necessary to juggle with primitive definitions by declaring the comma an active mathchar with math code "8000 and then assigningg this active char a definition that tests if the next token is a space or something else; if it's a space then the punctuation mark is used, otherwise the ordinary math symbol is used. Not difficult to write down, but certainly this is one of those definitions that I put in all my personal macro files.

```
\makeatletter
\DeclareMathSymbol{\virgola}{\mathpunct}{letters}{"3B}
\DeclareMathSymbol{\decimalcomma}{\mathord}{letters}{"3B}
\AtBeginDocument{\mathcode\,="8000}
{\catcode ,=\active \gdef,{\futurelet\let@token\m@thcomma}}
\def\m@thcomma{\ifx\let@token@sptoken\virgola\else\decimalcomma\fi}
\makeatother
```

If you add these definitions then you'll notice the difference when you write fractional numbers as well as when you type

$$\left(f(x, y) \right)$$

versus

$$\left(f(x, y) \right)$$

This more extensive answer has been provided by **Claudio Beccari**, a long time LaTeX user. In 1991 he wrote a book in Italian with the title "LaTeX — Guida a un sistema di editoria elettronica". Since then he is considered one of the gurus of the Italian TeX Users, even if this fame is totally undeserved. He has contributed several papers to TUGboat; he produced the hyphenation patterns for Italian and Latin, still in use today; he designed the default Greek fonts for use with Babel and supplied the Greek hyphenation patterns (luckily enough the Greek Users produced better patterns and these replaced Claudio's). Presently the Italian language definition file of the Babel package, although under the full control of Johannes Braams, is regularly updated and enriched by Claudio. He has participated in various TeX conferences and is a member of TUG GuIT, the official Group of the Italian TeX Users. He can be reached at claudio.beccari@polito.it

TPJ

[Journal](#)
[home page](#)
[General](#)
[information](#)
[Submit an](#)
[item](#)
[Download](#)
[style files](#)
[Copyright](#)
[Contact us](#)

THE PracTeX Journal



Distractions — Some chess problems created in LaTeX

The Editors

- [Comment on this paper](#)
- [Send submission idea to editor](#)

Some LaTeX chess problems

Chess problems are by nature graphical, which makes them a great subject for this issue's "Distractions". We present, therefore, a [paper by Dirk Baechle](#) on chess problems.

In this paper Dirk explains how his program *ChessTask* works and can be used to typeset chess problems in LaTeX using the *skak* package by Torben Hoffmann (available from your local CTAN mirror). He also shows several chess problems, and explains in detail how to solve them.

For those who want more, some other chess problems are presented below. These problems have been typeset by one of the editors using the far more ancient *chess* font and style by Piet Tutelaers, also available from CTAN. This gives you the opportunity to compare the results of these two approaches to typesetting chess problems.

[Additional chess problem 1](#)

[Additional chess problem 2](#)

And here are their solutions.

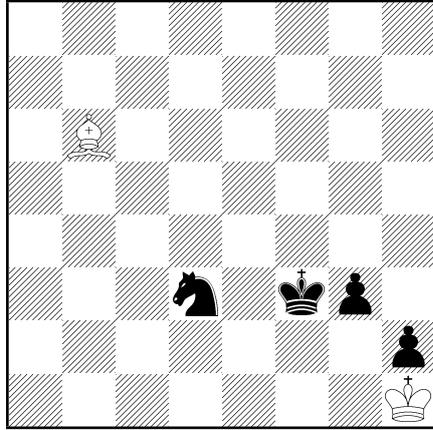
[Solution to additional chess problem 1](#)

[Solution to additional chess problem 2](#)

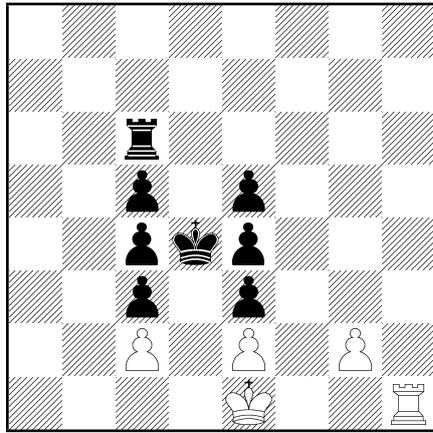
Enjoy!

Page generated March 14, 2007 ; [TUG home page](#); [search](#); [contact webmaster](#).

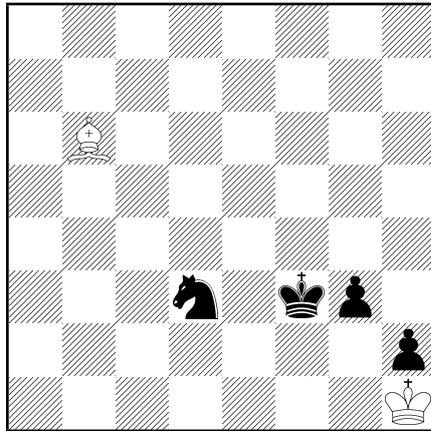
Black plays and mates in four moves (Alapin–Rubinstein, Vienna 1908).



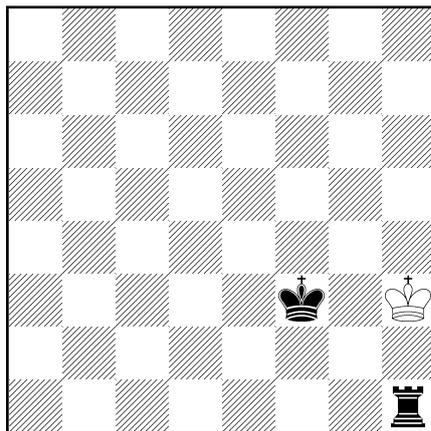
Mate in two (T.R. Dawson, 1934).



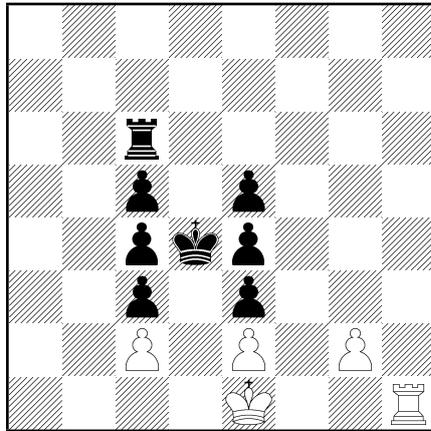
Black plays and mates in four moves (Alapin–Rubinstein, Vienna 1908).



- | | | |
|-----|--------|---------|
| 75. | ... | ♞d3–f2+ |
| 76. | ♔b6×f2 | g3×f2 |
| 77. | ♚h1×h2 | f2–f1 ♖ |
| 78. | ♚h2–h3 | ♖f1–h1 |



Mate in two (T.R. Dawson, 1934).



Note that the assignment does not state which player mates, and that is exactly the point here. If it is white's turn, then white plays 1. 0—0 and, irrespective of what black does, mates with 2. ♖d4-d1.

Should it, however, be black's turn, then black plays 1. ... ♜c6-a6! threatening 2. ... ♜a6-a1 mate. White is unable to do anything against this because castling (which helped white win just yet!) is illegal now since white's previous move *must* have been with either with the king or the rook.