RUG LaTeX Course

Siep Kroonenberg

Contents

Preface

This is the book accompanying a LaTeX course for first-year econometrics students at the Rijksuniversiteit Groningen. The aim of the course is to guide you through your first steps with LaTeX, with pointers to additional LaTeX resources to further your LaTeX skills.

Some topics are delegated to appendices because they did not fit in with the course. Consult them as needed. There is an accompanying downloadable zip file with sample code.

The second chapter deals with practicalities. It gives some hints how you can set up your own TeX environment, and also introduces the TeX installation of the Rijksuniversiteit Groningen.

The TeXstudio editor used in the course is free, cross-platform and has palettes of mathematical symbols. This last feature is useful for LaTeX beginners who have to typeset a lot of mathematics, or at least for their teacher. But most of this book is editor-agnostic, and the launcher, see below, makes it easy to switch editors.

The university installation and the TeX Live launcher

A few words about our local setup: the university's TeX Live installation resides on the network. It includes a launcher with menu entries for the various components of the installation. Upon first use the launcher does some necessary initialization, such as adjusting the search-path and setting up file associations. You can rerun this initializer at any time if something seems messed up.

The current incarnation of the launcher is written in a Windows-specific scripting language AutoIt. I am slowly rewriting it in C. In the new version, it will be possible to define the menu and other details in a configuration file; making it easy to adapt the launcher to other setups. I hope to publish it eventually in some form.

The launcher replaces an earlier initialization script which created a menu and file associations, and added TeX Live to the user searchpath. It consisted of a batch file / Perl script combination which made use of the built-in Perl of TeX Live. A version is available at `www.tug.org/texlive/w32client.html`.

1 Introduction

This introduction to LaTeX is no more than that, an introduction. Hopefully, by the end of it you will see the point of LaTeX, and are able to explore it further on your own.

This chapter gives a bit of background, which may help you understand the philosophy of LaTeX.

1.1 History

LaTeX is pretty old, and its roots are even older.

In 1977 Donald Knuth, a mathematician, started creating his typesetting system TeX as a tool for better typesetting, expecially of mathematics. The first version of TeX appeared in 1978. The second version, which was a complete rewrite, appeared in 1982.

The TeX system is based on a markup language. Contrary to html, another popular markup language, TeX is designed for precision typesetting of complex texts, and is also a macro language.

In the early 1980s Leslie Lamport provided a set of macros providing features such as automatic chapter- and section numbering, footnotes and automatic cross-referencing.

LaTeX allowed authors to concentrate on the meaning and structure of documents, rather on appearance. This is called *structural markup*.

The LaTeX macros cover most uses of TeX: besides articles, reports and books, slides and letters are also provided for. Most TeX users started using the LaTeX macros for their documents, adding their own stuff as needed, rather than starting their own macro package from scratch.

1.2 The TeX/LaTeX ecosystem

TeX and LaTeX caught on, and a large ecosystem grew up around it. Users around the world contributed macros, fonts, support for non-Western languages, manuals and utilities. The vast majority is free.

The *CTAN* archives are the main repository for TeX-related material, but it is likely that your TeX installation already contains everything you need, or provides an easy way to add it.

The sites of TeX user groups such as `http://www.tug.org` and `http://www.ntg.nl/` host various mailing lists and have links to other TeX- and LaTeX-related sites. My own site `http://tex.aanhet.net/rugtex/` is about the RuG TeX installation and also contains some useful links.

1.3 Working with LaTeX

LaTeX is not a wysywyg wordprocessor. Preparing a document with LaTeX is a three-step cycle:

1. Enter your text, with markup, in a text editor such as Notepad or TeXstudio or Emacs.

2. 'Compile' your document, *i.e.* convert it to pdf; more on that later.

3. Preview it in a pdf viewer such as Adobe Reader or SumatraPDF or the pdf viewer built into the editor, if there is one.

Below you see a marked-up source fragment at the left and two typeset results at the right:

1.1 Some mathematics

The econometric model is confronted with observed data and the parameters are estimated by econometric techniques, as shown in equation 1 below:

```
\subsection{Some mathematics}\label{math}

The econometric model is confronted with observed
data and the parameters are estimated by
econometric techniques, as shown in equation
\ref{eq:smm} below:
\begin{equation}
Y_t = G_1 (Y, X, \hat{\Theta}, \hat{U})
\label{eq:smm}
\end{equation}
```

$$Y_t = G_1(Y, X, \hat{\Theta}, \hat{U}) \qquad (1)$$

1.1 Some mathematics. The econometric model is confronted with observed data and the parameters are estimated by econometric techniques, as shown in equation 1 below:

$$Y_t = G_1(Y, X, \hat{\Theta}, \hat{U}) \qquad (1)$$

The difference in appearance is entirely due to changes in the document header. LaTeX allows you to forget about layout and typography until you are ready for it.

LaTeX excels at automatic numbering and cross-referencing; as well as at other long-document features such as bibliographies and indexing. Best of all, these keep working efficiently and reliably as the size of the project increases.

Hopefully, you will discover that LaTeX can greatly streamline your publishing process.

2   Getting started

2.1 Getting (La)TEX

2.1.1 Editor

An obvious component of a (La)TEX installation is the editor in which you enter your documents, and which compiles them into a pdf file. For this course we use the TeXstudio editor, which is free, and is available for both Windows and Linux. It offers extensive assistance for entering LaTEX code and math.

Since LaTEX files are plain text files, you can view them in any editor, but you should edit them in an editor which adds no binary stuff. You can use a general programmers editor or even Notepad, but a dedicated LaTEX editor can give you a lot more help.

2.1.2 TEX system

The editor is actually the least important part; it delegates all the hard work to the TEX system in the background.

A TEX system is a complex ensemble of command-line programs, macros, fonts and data files. The two major alternatives are TEX Live and MiKTEX, both free. The RuG installation uses TEX Live, which is available for Windows, Mac OS and Unix/Linux, and is included in most Linux distributions. The other one, MiKTEX, is Windows-only. Both contain all the bits and pieces that you are likely to need, although TEX Live is a bit more comprehensive. They both include the TeXworks editor.

2.1.3 More advice

My web page `http://tex.aanhet.net/rugtex/home_inst.html` gives more detail about collecting your own TEX system and download links. Another useful beginners resource is `http://www.tug.org/begin.html`.

Two important points:

- Make sure TEX Live or MiKTEX is installed before you run the editor for the first time. This enables the editor to configure itself automatically for your TEX system.
- A full LaTEX install takes a long time; do not attempt this, *e.g.*, while in class.

2.2 The RuG TEX Live launcher

The Rijksuniversiteit Groningen has a ready-to-run LaTEX installation, complete with TEX Live and several editors. It also includes various additional utilities, some of which will be mentioned later in this book.
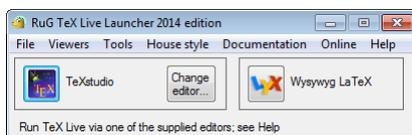

Figure 2.1. The TEX Live launcher

For a standard university UWP computer, you should have a menu item *Start / Programs / Text Processing / TeX Live RuG nnnn*. This invokes the TeX Live Launcher, see Figure 2.1. The TeX Live launcher is also available in a remote session. From this launcher you can start up your favorite LaTeX editor, consult documentation and do some configuration and maintenance. Take a moment to browse the launcher menus.

The 'RuG TeX Live website' item in the *Online* menu points to the web pages for our TeX installation, one of which explains how to install LaTeX at home.

The button at the left, labeled '(La)TeX editor', invokes your selected default editor. The launcher offers you a couple of additional choices besides TeXstudio; plus the option to select an editor of your own.

2.3 Next: let the system display file extensions

If you work on a file `somefile.tex`, then TeX is going to generate various auxiliary files such as `somefile.aux` and `somefile.log`. For Windows, the following steps ensure that file extensions are visible, so you can tell all these files apart:

- Windows XP/7
  - Click on 'My Computer'
  - Click on Tools / Folder Options. Windows 7: if you do not see Tools, hit the left Alt key first.
- Windows 8 (⊞ indicates the Windows- or logo key)
  - Type ⊞ -D for desktop mode
  - Type ⊞ -E for File Explorer
  - Click on the Options logo
- Select the View tab
- Uncheck 'Hide extensions for known filetypes'
- Click 'Apply to All Folders'
- Click OK

2.4 TeXstudio

Figure 2.3 shows the TeXstudio edit screen. The editing area is surrounded by various toolbars, a structure view on the left and optionally a tabbed information area at the bottom.

This latter item is rather useful. If you do not see it, you can make it visible by clicking on the second-left item ▣ at the lower left corner of the editor window.

While you are at it, you can also right-click on an empty area of the toolbar or menu bar to get rid of some of the toolbar clutter; everything is already available via the menus.

Also look through the TeXstudio menus, in particular:

- The Tools menu and its Commands submenu for running LaTeX and various utilities; see section 2.5
- The LaTeX menu for inserting various LaTeX macros
- The Math menu for inserting LaTeX macros for math

You can also type LaTeX macros by hand once you know them.
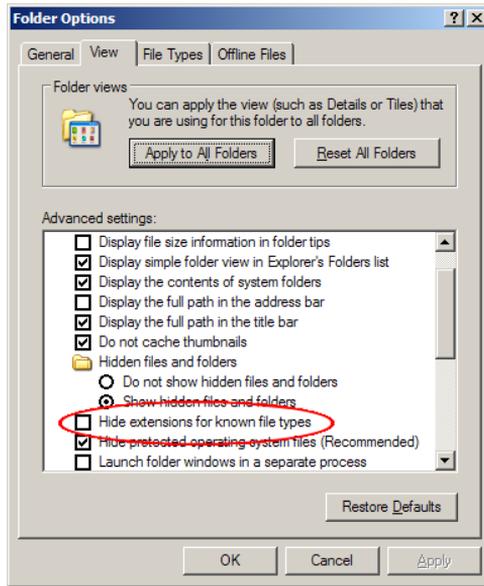
Figure 2.2. Letting Windows display file extensions

2.5 First document

Now we are going to do the entire cycle: text entry, compilation and previewing from the LaTeX editor TeXstudio. Create a new document by clicking on *File / New* and type the following code:
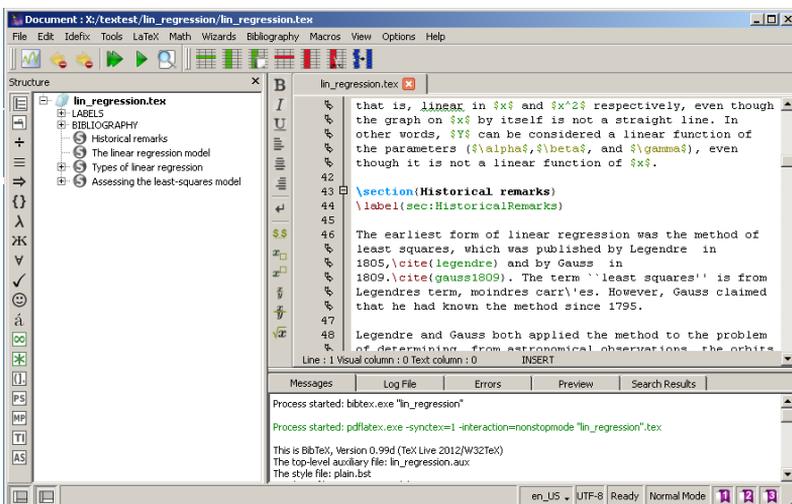
Figure 2.3. TeXstudio, a LaTeX editor

```
\documentclass{article}
\begin{document}
Hello, world!
\end{document}
```

This is a complete LaTeX document. Setup is done in the *preamble*, *i.e.* the \documentclass line and anything else before \begin{document}. In this case, we just specified that we wanted an article rather than, *e.g.*, a book or a letter. Actual content goes between \begin{document} and \end{document}.

2.5.1 Compiling

Save the document as, *e.g.*, X:\latexdocs\hello.tex. Then click the Build button (▶). If all is well a pdf-preview window should pop up; see figure 2.4.

Also have a look at the message area below the editing area (figure 2.5). If there are problems or you are not getting the output you expect then you can inspect the Errors tab.
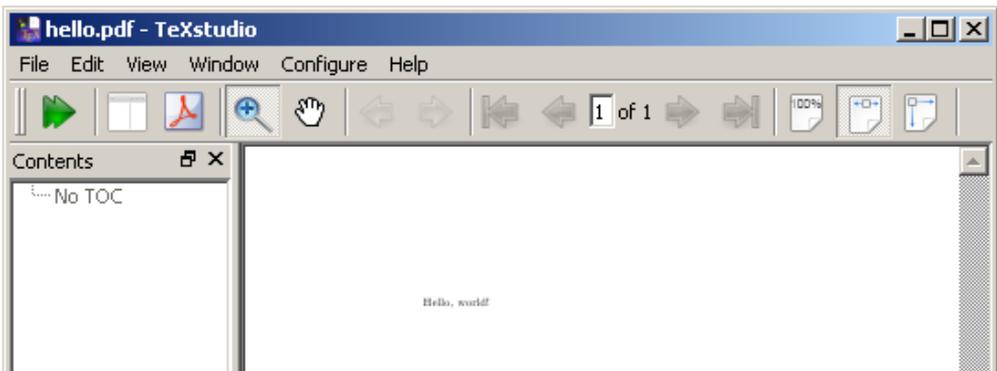
Figure 2.4. The built-in previewer of TeXstudio
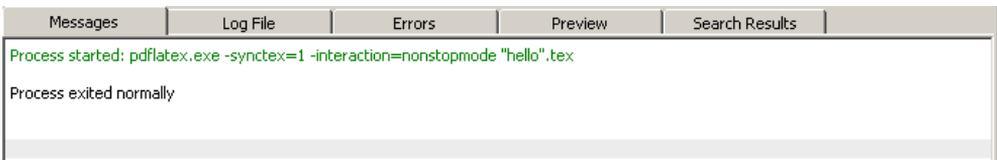
Figure 2.5. Messages tab after a successful compilation

You can read more about compiling in Section 3 of the online help: *Help / User Manual....*. It also explains what to do in case of errors.

2.6 Documentation

Built-in help. The Help menu of TeXstudio provides both help for TeXstudio itself and a brief LaTeX reference.

The launcher Documentation- and Online menus contain shortcuts to several other useful manuals and online resources.

The *LaTeX Introduction* menu entry points to a book-length introduction which covers all the basics. It is also a nice demonstration of the bookmarking and hyperlinking that you get virtually for free with LaTeX, and which makes the pdf very convenient to consult on-screen. The next menu item, *LaTeX Reference*, is the full version of the built-in LaTeX reference.

The [UK TeX] *FAQ* is another useful resource.

2.6.1 The documentation list

You can gain access to documentation about packages in one of the following ways:

RUG Launcher   In the Documentation menu, a menu item *All TeX Live documentation by package* invokes an html file `doc.html` which contains links to virtually all package manuals.

Standard TeX Live   The menu item *Start / Programs / TeX Live* nnnn */ TeX Live Documentation* invokes an html page which includes a link *Available package documentation*, which is this same file.

MiKTeX   lacks such a file, but you can still browse to the *CTAN Catalogue*, `http://mirror.ctan.org/help/Catalogue/brief.html` and consult package documentation there.

We shall refer to this list, however accessed, as 'the documentation list.'

But you can also use the `texdoc` (TeX Live) or `mthelp` (MiKTeX) command-line utility.

2.6.2 Tip: view pdfs with narrow margins

For better use of your screen pixels, you may wish that your pdf viewer zoomed in on the printed part of the page. Several pdf viewers can do this automatically. In Adobe Reader, select *View / Zoom / Fit Visible*. In SumatraPDF, part of our TeX installation, select *Zoom / Fit Content*.

2.7 Practice files

This course comes with a zipfile `practice.zip` with some example `.tex` files, a subdirectory `figures` of graphic files for later in the course and a second subdirectory `bibtex` relating to bibliography management[1]. Is is available from *CTAN* in the `info/latexcourse-rug` subdirectory.

Right-click the zip file after downloading, select 'Extract All...' and pick a suitable directory for unpacking, *not* some obscure directory for temporary files.

Several chapters conclude with suggestions for practicing, which usually refer to files from this zipfile.

---
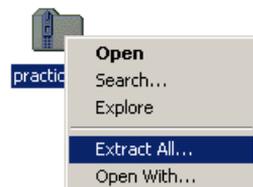
1. This last topic is no longer part of the course, but when you start using LaTeX for real, it is highly advisable to learn the LaTeX bibliography system.

3 Basics

Keep *The Not So Short Introduction* handy; as mentioned previously, it is in the launcher menu under *Documentation*, or search for 'lshort' in the documentation list, see section 2.6.1.

Start a new LaTeX document as described in section 2.5, with content

```
\documentclass{article}
\begin{document}
Hello, world!
\end{document}
```

Hello, world!

You may already have guessed that macros start with \ and that a parameter can be enclosed in braces { }. A construct \begin{something}…\end{something} is called an *environment*.

Now try out some of the syntax below on your new LaTeX document.

3.1 Paragraphs

You need to separate paragraphs with empty lines in the input file; a single linebreak is equivalent to a space.

```
A linebreak in the source
creates a space in the pdf output.
```

A linebreak in the source creates a space in the pdf output.

```
An empty line in the source ends a paragraph.
```

An empty line in the source ends a paragraph.

3.2 Comments

The percent character, %, is the comment character; LaTeX ignores it and everything following it on the same line, including the linebreak itself.

```
one
%ignore
tw%
o
```

one two

3.3 Control sequences and –characters

LaTeX commands often take the form of a backslash followed by a series of letters, *e.g.*,

```
\LaTeX
```

LaTeX

A control sequence swallows succeeding spaces, so you sometimes have to follow it with {} or ~:

```
\LaTeX code
```

LaTeXcode

```
\LaTeX~code, \LaTeX{} code, \LaTeX.
```

LaTeX code, LaTeX code, LaTeX.

Rendering control characters literally:

	function	*render literally with*
%	comment character	\%
{ }	parameter; grouping	\{ \}
\	starts control sequence	\textbackslash
\\	newline (!)	

3.4 Grouping

A pair of braces can also localize the effect of a command:

```
x z {\footnotesize x z} x z
```
x z x z x z

3.5 Text formatting

The classfile and stylefiles will take care of many changes in text attributes, *e.g.*, in section heads and in bibliographies. Do not style these items manually. Appendix A contains some simple recipes for adjusting style globally.

Below, we describe the more common commands for styling text.

3.5.1 Bold and italic

These commands work on all subsequent text within the current block:

```
normal \itshape italic \bfseries bolditalic
\upshape bold \mdseries normal
```
text *italic* ***bolditalic*** **bold** normal

Argument form:

```
normal \textit{italic} \textbf{bold}
```
normal *italic* **bold**

These are the basic text formatting commands:

	'from now on'	*argument form*
italic	\itshape	\textit{...}
upright	\upshape	\textup{...}
bold	\bfseries	\textbf{...}
medium	\mdseries	\textmd{...}
monospaced	\ttfamily	\texttt{...}

Some people recommend replacing \textit with \emph, which is short for *emphasized*, as being more in line with structural markup.

TeXstudio has buttons for bold **B** and italic *I* on the inner vertical toolbar.

3.5.2 Text sizes

Predefined text sizes; note that some may come out the same:

\tiny
\scriptsize   \normalsize
\footnotesize   \large   \LARGE
\small   \Large   \huge

3.6 Special characters

Here a short list of typographic characters and how you can create them in LaTeX, even if you use only typewriter characters in your input:

	output	*code*
Single quotes	' '	`` ` ' ``
Double quotes	" "	```` `` '' ````
Non-breaking space		`~`
Hyphen	-	`-`
En-dash	–	`--`
Em-dash	—	`---`
Accented characters	é	`\'e`
	ï	`\"\i`

Using accented *input* characters requires loading the `inputenc` package in the preamble:[1]

```
\usepackage[utf8]{inputenc}
```

```
...
" ä ï © « " and `` \"a \"\i{} \textcopyright{} << ''
```
        " ä ï © « " and " ä ï © « "

This method does not cover all unicode characters, and if you type a lot of code then you may prefer control codes anyway.

For full unicode support, you should use the XeTeX or LuaTeX engines; see appendix sections A.6.1 and B.2.

3.6.1 Hyphens and dashes

Please be aware that not every horizontal dash is the same. A few examples of proper use:

En-dashes for ranges: `7--9` for '7–9', or to set off – part of – a sentence.

Em-dashes also for setting off—part of—a sentence, but now without surrounding spaces[2].

A plain *hyphen* '-' is appropriate for hyphenation and for compound words such as 'cross-referencing'.

3.7 Lists: itemize, enumerate and description

Itemize (unnumbered list):

```
\begin{itemize}
\item camel
\item rabbit
\end{itemize}
```

- camel
- rabbit

Enumerate (numbered list):

```
\begin{enumerate}
\item soup
\item main course
\item dessert
\end{enumerate}
```

1. soup
2. main course
3. dessert

Description lists:

```
\begin{description}
\item[One] This is a short term.
\item[Quetzalquatl] Mexican god, about whom we
  could tell a lot if only we had the time and
  inclination.
\end{description}
```

One   This is a short term.

Quetzalquatl   Mexican god, about whom we could tell a lot if only we had the time and inclination.

---

1. `latin1` is an alternative to `utf8`, but if we go beyond ASCII input at all, then let us do it right and adopt unicode.
2. Or, better with thin spaces `\,`

Lists can be nested:

```
\begin{enumerate}
\item soup
\item main course
  \begin{itemize}
  \item tortilla filled with meat and vegetables
  \item refried beans
  \end{itemize}
\item dessert
\end{enumerate}
```

1. soup
2. main course
   - tortilla filled with meat and vegetables
   - refried beans
3. dessert

Here, we see an example of *optional* parameters, which are enclosed in square brackets.

3.8 LaTeX classes

Each LaTeX document starts with a \documentclass line, which selects a class file. Class files define available features and a default look. Some important LaTeX document classes:

- article (no chapters)
- report
- book

The above classes are very similar in the features they support. You can add features or change the appearance by loading packages:[3]

```
\documentclass[10pt,a4paper]{article}
\usepackage[utf8]{inputenc}
\usepackage{amsmath}
\usepackage{amsfonts}
\usepackage{amssymb}
```

3.9 Sectioning commands

The standard classes listed above have a predefined sectioning hierarchy: parts, chapters (not for articles), sections, subsections, subsubsections, paragraphs and subparagraphs.

All these commands all have an optional and a required parameter, *e.g.*

```
\section[Short title]{A very long and impossibly involved title,
which will never fit in a page header}
\subsection{A short enough title}
```

Sectioning titles may turn up in page headers or in an automatically generated table of contents. If a title isn't short and simple, you definitely should use an optional parameter which won't cause trouble when it is reused in page headers or in a table of contents.

3.9.1 Bookmarks and clickable cross-references with hyperref

The hyperref package will create bookmarks from your sections, and also make all the cross-references in your pdf clickable. Add an option colorlinks if you do not like the boxes around links:

```
\usepackage[colorlinks]{hyperref}
```

---

3. This is a preamble generated by the TeXstudio Quick Start wizard.

3.10 Title

Publications customarily start with some sort of title page or -block. LaTeX creates such a title with the \maketitle command. You should already have defined an author and title with corresponding commands.

The \author- and \title commands can be placed either in the preamble or in the body of the LaTeX source. The \maketitle command belongs in the body.

Here is an example of an article with a \usepackage command, a title block, a table of contents and sections:

```
\documentclass{article}
\usepackage{newpxtext,newpxmath} % palatino font
\begin{document}
\title{Title of article}
\author{My name}
\maketitle
\thispagestyle{empty}

\tableofcontents

\section{A section}\label{sec:ASection}

See section \ref{sec:ASection} on page
\pageref{sec:ASection}.

\subsection{A subsection}

That's all, folks!

\end{document}
```

> # Title of article
>
> My name
>
> June 17, 2014
>
> ## Contents
>
> **1 A section**   **1**
>   1.1 A subsection 1
>
> # 1 A section
>
> See section 1 on page 1.
>
> ## 1.1 A subsection
>
> That's all, folks!

Notice the use of cross-referencing commands \label, \ref and \pageref.

Warning. Cross-references usually require more than one (pdf)LaTeX run before they are correctly resolved. This is also true for automatically generated text such as tables of contents. After each LaTeX run, be sure to check the errors tab below the editing area for errors and warnings.

3.11 Footnotes and 'thanks'

In the LaTeX source footnotes are placed in the running text. The \footnote command generates both the mark in the running text and the footnote itself at the bottom. As with sectioning, footnotes are numbered automatically:

```
Here comes a footnote.\footnote{%
This is the footnote.}
And some more text.
```

> Here comes a footnote.[1] And some more text.
>
> ---
> [1]This is the footnote.

A special case is a footnote attached to the title or author of an article. Note that the footnote should be *inside* the title- or author parameter.

```
\title{Sample title\thanks{%
 Supported by a grant}}
\author{A.U. Thor\thanks{%
 And another grant}}
\maketitle

First line of regular text.\footnote{%
 With a regular footnote.} And some more text.
```

> # Sample title*
>
> ## A.U. Thor†
>
> June 17, 2014
>
> First line of regular text.¹ And some more text.
>
> ———————
>
> *Supported by a grant
> †And another grant
> ¹With a regular footnote.

3.12 Practice

Start out with a new document as described in section 2.5. Use this document to try out the code samples from this chapter.

If you feel ready to try bigger things, you can try to typeset some real text. If you have nothing suitable of your own, you can turn to Wikipedia articles such as `http://en.wikipedia.org/wiki/Factors_of_production`. You can copy-and-paste pieces of text from the web page to your own LaTeX document.

Try to recreate the structure, not the appearance, *e.g.*, use sectioning commands instead of manually making headings bold, and let LaTeX create the table of contents. Also pay attention to proper quotes and typographic characters.

Consult `basics_sample.tex` from the practice zip (see section 2.7) as an example of a complete, structured LaTeX document.

4   Math

4.1   Amsmath

Although you can do a lot of math typesetting with LaTeX alone, we shall assume that amsmath and related packages are loaded, *e.g.* with a command

```
\usepackage{amsmath,amsfonts,amssymb}
```

in the preamble, *i.e.* between \documentclass{. . . } and \begin{document}.

For documentation, click in the launcher *Documentation / AmsMath User Guide*, or search the documentation list, see section 2.6.1.

4.2   Math mode: Inline and display math

Math in running text is bracketed between $ characters:[1]

```
Simple bits of math in running text,
enclosed in \$ characters: $x$ or
$\alpha$ or $\sum_i n_i$
```

Simple bits of math in running text, enclosed in $ characters: x or α or $\sum_i n_i$

Notice that *ordinary letters are italicized* in math mode.

More elaborate formulas are better typeset as *display math*, on a line by itself[2]. Notice the more spacious typesetting of indices in display math mode.

```
\[ x = \sum_{i=0}^\infty y_i \]
```

$$x = \sum_{i=0}^{\infty} y_i$$

Display math with automatic equation numbering:

```
\begin{equation}
x = \sum_{i=0}^\infty y_i \label{firstequation}
\end{equation}
See equation \ref{firstequation}
on page \pageref{firstequation}.
```

$$x = \sum_{i=0}^{\infty} y_i \tag{4.1}$$

See equation 4.1 on page 17.

This is yet another example of automatically generated numbers which can be used for cross-referencing.

4.3   Mathematical notation

Many symbols listed below can be entered via the TeXstudio interface; either via the Math menu or via the panel at the left. But you can also type the code directly.

---

1. Alternative codings: \(. . . \) or \begin{math} . . . \end{math}.
2. Alternative codings: \begin{displaymath} . . . \end{displaymath} and, only with the amsmath package: \begin{equation*} . . . \end{equation*}.

4.3.1 Greek letters

```
lowercase: $\alpha, \beta, \epsilon,
\varepsilon, \gamma, \phi, \psi,
\xi, \pi, \sigma, \omega$ \\
uppercase: $\Gamma, \Phi, \Psi, \Xi,
 \Pi,\Sigma, \Omega$
```

lowercase: $\alpha, \beta, \epsilon, \varepsilon, \gamma, \phi, \psi, \xi, \pi, \sigma, \omega$
uppercase: $\Gamma, \Phi, \Psi, \Xi, \Pi, \Sigma, \Omega$

4.3.2 Mathematical accents

```
$x', \hat{a}, \acute{e}, \bar{\imath},
\vec{o}, \dot{u}, \ddot{v},
\vec{\dot{Y}}$
```

$x', \hat{a}, \acute{e}, \bar{\imath}, \vec{o}, \dot{u}, \ddot{v}, \vec{\dot{Y}}$

Note \imath for a dotless i, and the last example which stacks two accents on top of each other.

4.3.3 Various symbols

Arithmetic and relational operators

```
$\alpha = \theta - \gamma \times \zeta$\\
$x < y$ and $a > b$ \\
$u \leq v$ and $i \geq j$ \\
$\sigma \pm \tau$ and $\beta \sim \rho$
```

$\alpha = \theta - \gamma \times \zeta$
$x < y$ and $a > b$
$u \leq v$ and $i \geq j$
$\sigma \pm \tau$ and $\beta \sim \rho$

Arrows

```
$\leftarrow, \Rightarrow,
\uparrow, \Downarrow,
\leftrightarrow,
\longleftrightarrow$
```

$\leftarrow, \Rightarrow, \uparrow, \Downarrow, \leftrightarrow, \longleftrightarrow$

4.3.4 Finding symbols

Many symbols are already available via the TeXstudio interface. But for a very comprehensive list, consult the document 'Comprehensive Symbol list', which is part of the TeX Live documentation. Search for 'comprehensive' in the documentation list (see 2.6.1).

4.3.5 Functions

```
Do not write $log 100 = 2$ but \\
$\log 100 = 2$ \\
$\ln 100 = 4.605$ \\
$\sin(45) = 0.707$
```

Do not write $log 100 = 2$ but
$\log 100 = 2$
$\ln 100 = 4.605$
$\sin(45) = 0.707$

4.4 Various constructs

For the samples below, we use display math, since many of them take up too much height to fit within a standard line of text. Note the use of braces { and } to collect several letters and symbols into one argument.

Subscripts and superscripts

```
\[ x_i, x_{i+1}, a^2, b^{x+y} \]
```

$$x_i, x_{i+1}, a^2, b^{x+y}$$

Roots, without and with optional parameter

```
\[ \sqrt{x+y}, \sqrt[n]{2} \]
```

$$\sqrt{x+y}, \sqrt[n]{2}$$
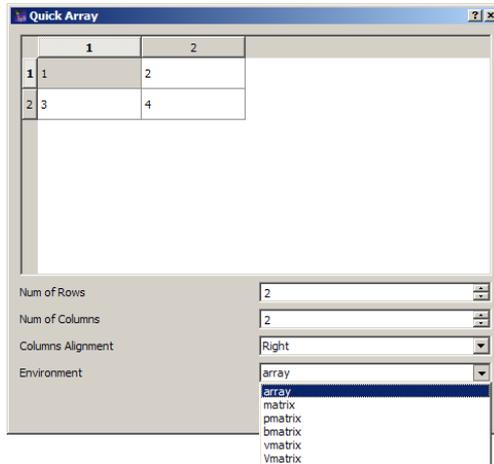
Figure 4.1. Quick Array wizard

Two styles of fractions and regular text within display math

```
\[ x/y \text{ and } \frac{\alpha}{\beta + \gamma} \]
```

$$x/y \text{ and } \frac{\alpha}{\beta + \gamma}$$

Sums, products and integrals

```
\[ \sum_i x_i = \prod_{i=2}^7 i+1 =
\int_{z=0}^\infty z^2 \]
```

$$\sum_i x_i = \prod_{i=2}^7 i + 1 = \int_{z=0}^\infty z^2$$

Ellipsis (dots), on the baseline and higher up

```
\[ x_0 \ldots x_{100},
 x_0 + \cdots + x_{10} \]
```

$$x_0 \ldots x_{100}, x_0 + \cdots + x_{10}$$

4.5 Arrays/matrices

LaTeX arrays:

```
\[ \begin{array}{lcr}
 0.15   &   3a  &  0 \\
 0.0003 &  501d & 10 \\
 0.011  &   2c  &  1
\end{array} \]
```

$$\begin{array}{lcr}
0.15 & 3a & 0 \\
0.0003 & 501d & 10 \\
0.011 & 2c & 1
\end{array}$$

In the second parameter above, lcr, each of the three letters 'lcr' specify the alignment of one column: left, centered and right.

TeXstudio has a 'Quick Array' wizard to create a first approximation, see Figure 4.1. The wizard assumes that the text cursor is between math mode delimiters such as \[...\].

Matrices, amsmath-style:

```
\[ \begin{matrix}
x & y & z \\
.0 & .01 & .001
\end{matrix} \]
```

$$\begin{matrix}
x & y & z \\
.0 & .01 & .001
\end{matrix}$$

Notice the absence of column specifications; all columns are centered.

You get built-in round brackets '( )' with pmatrix and square brackets '[ ]' with bmatrix. See the amsmath documentation for more variations.

```
\[ \begin{pmatrix}
x & y & z \\
.0 & .01 & .001
\end{pmatrix} \begin{bmatrix}
x & y & z \\
.0 & .01 & .001
\end{bmatrix} \]
```

$$\begin{pmatrix} x & y & z \\ .0 & .01 & .001 \end{pmatrix} \begin{bmatrix} x & y & z \\ .0 & .01 & .001 \end{bmatrix}$$

Matrix with various ellipses:

```
\[ \begin{bmatrix}
   a_{11} & \ldots & a_{1m} \\
   \vdots & \ddots & \vdots \\
   a_{n1} & \ldots & a_{nm}
 \end{bmatrix} \]
```

$$\begin{bmatrix} a_{11} & \ldots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \ldots & a_{nm} \end{bmatrix}$$

or

```
\[ \begin{bmatrix}
   a_{11} & \ldots & a_{1m} \\
   \hdotsfor{3} \\
   a_{n1} & \ldots & a_{nm}
 \end{bmatrix} \]
```

$$\begin{bmatrix} a_{11} & \ldots & a_{1m} \\ \hdotsfor{3} \\ a_{n1} & \ldots & a_{nm} \end{bmatrix}$$

Bracketing with large delimiters:

```
\[ \left( \begin{array}{rr}
      10 & 100 \\
       a &   b
      \end{array}\right) \]
```

$$\left(\begin{array}{rr} 10 & 100 \\ a & b \end{array} \right)$$

This also works with braces '{ }' and square brackets '[ ]'. If you need only one of the two braces, use '.' for the other one:

```
\[ \left\{ \begin{array}{c} a \\
 b \end{array} \right. \]
```

$$\left\{ \begin{array}{c} a \\ b \end{array} \right.$$

4.6 Multiline equations

There are various constructs for multiline equations. Basic LaTeX has the eqnarray and eqnarray* environments, the first with, the second without automatic numbering.

But we shall just give an example of the amstex align and align* environments:

```
\begin{align}
f(x) &= (a + b)^2 \nonumber \\
     &= a^2 + 2ab + b^2\label{AnEquation} \\
     &\ne (a+b)(a-b)\label{AnOther}
\end{align}
See equation \ref{AnEquation} and \ref{AnOther}.
```

$$
\begin{aligned}
f(x) &= (a + b)^2 \\
&= a^2 + 2ab + b^2 \quad (4.2) \\
&\ne (a + b)(a - b) \quad (4.3)
\end{aligned}
$$

See equations 4.2 and 4.3.

The & character defines the alignment. You see that every line get its own number, unless it is suppressed with a \nonumber command.

The starred version omits the numbering:

```
\begin{align*}
f(x) &= (a + b)^2 \\
     &= a^2 + 2ab + b^2
\end{align*}
```

$$f(x) = (a + b)^2$$
$$= a^2 + 2ab + b^2$$

4.7 Fonts in math

4.7.1 Upright and italic

First, note that alphabetic characters will be italicized in math mode. Use \mathrm to get an upright version:

```
$E, \mathrm{E}, p, \mathrm{p}$
```

$E, \mathrm{E}, p, \mathrm{p}$

4.7.2 Bold

With bold, the situation is, unfortunately, a bit complicated. For regular 'latin' alphabetic characters, use \mathbf, which makes the character at the same time bold and upright:

```
$M, \mathbf{M}, v, \mathbf{v}$
```

$M, \mathbf{M}, v, \mathbf{v}$

For Greek characters and other symbols, try \boldsymbol instead of \mathbf:

```
$\Psi, \boldsymbol{\Psi},
\infty, \boldsymbol{\infty}$
```

$\Psi, \boldsymbol{\Psi}, \infty, \boldsymbol{\infty}$

If neither \mathbf nor \boldsymbol does the trick, load the bm package:

```
\usepackage{bm}
```

and try again.

4.7.3 Fancy math fonts

```
Blackboard: $\mathbb{B}$\\
Calligraphic: $\cal{A}$\\
Fraktur: $\mathfrak{A}$
```

Blackboard: $\mathbb{B}$
Calligraphic: $\mathcal{A}$
Fraktur: $\mathfrak{A}$

4.8 Macros

It can become pretty cumbersome to write something like \boldsymbol{\alpha} for α over and over again. You can define an abbreviation with the following code:

```
\newcommand{\balph}{\boldsymbol{\alpha}}
```

and then you just need to type \balph.

A macro can also have parameters. Below, [1] indicates the number of parameters and #1 indicates the first parameter.

```
\newcommand{\bvc}[1]{\vec{\mathbf{#1}}}
```

or, if you also want to use it in text without bothering with $ signs:

```
\newcommand{\bvc}[1]{\ensuremath{\vec{\mathbf{#1}}}}
```

With this definition you can type \bvc{x} rather than \vec{\mathbf{x}} or $\vec{\mathbf{x}}$ for $\vec{\mathbf{x}}$.

4.9 Practice

When trying out the code samples from this chapter, do not forget to load the AMS packages:

```
\documentclass{article}
\usepackage{amsmath,amsfonts,amssymb}
...
\begin{document}
...
\end{document}
```

Remember not to use inline math for displayed equations, see section 4.2.

The practice zip, see section 2.7, contains an example LaTeX file `math_sample.tex`.

When looking for real mathematical texts to convert to LaTeX, you may turn to Wikipedia pages such as `http://en.wikipedia.org/wiki/Linear_regression` or `http://en.wikipedia.org/wiki/L2_norm`, or use something of your own.

5 Tabulars

5.1 Basics

Outside math mode, the `tabular` environment provides tables, which can be considered the text counterpart of multicolumn arrays. As with math arrays, columns are separated with '&' and rows with '\\'.

TeXstudio has a tabular wizard, but it is not much help when things get hairy.

A very basic table:

```
\begin{tabular}{lcr}
small & whatever & 1 \\
big   & huh?     & 10000
\end{tabular}
```

small	whatever	1
big	huh?	10000

There is a preamble {lcr} which defines the alignment of the columns: left, center and right.

A table with some empty cells:

```
\begin{tabular}{lcr}
small & whatever \\
big   &         & 10000
\end{tabular}
```

small	whatever	
big		10000

You do not need to insert an ampersand & for empty cells at the end.

You can add vertical rules in the preamble and horizontal rules with an `\hline` command:

```
\begin{tabular}[t]{|l|r|r|}
\hline
& \textit{Butter} & \textit{Cheese} \\
\hline
2000 &  9.1 & 5.7 \\
\hline
2001 & 11.7 & 6.3 \\
\hline
2002 & 12.2 & 6.5 \\
\hline
\end{tabular}
```

	Butter	*Cheese*
2000	9.1	5.7
2001	11.7	6.3
2002	12.2	6.5

If you use horizontal rules at all, you should include the commands

```
\usepackage{array}
\setlength\extrarowheight{1pt}
```

in the preamble to get a little bit of space between rules and the cells below. You can also issue an `\extrarowheight` command in the middle of your document (from now on, `\extrarowheight` is set to 1pt). Fewer rules are usually better, see table 5.1.

Table 5.1. Fewer rules are usually better

	Butter	Cheese
2000	9.1	5.7
2001	11.7	6.3
2002	12.2	6.5

5.2 Partial rules

With a \cline command you can insert a horizontal rule that spans a range of columns:

```
\begin{tabular}{|lrr|}
\hline
& \textit{Butter} & \textit{Cheese} \\
\cline{2-3}
2000 &  9.1 & 5.7 \\
2001 & 11.7 & 6.3 \\
2002 & 12.2 & 6.5 \\
\hline
\end{tabular}
```

	Butter	Cheese
2000	9.1	5.7
2001	11.7	6.3
2002	12.2	6.5

5.3 Multicolumn

the \multicolumn macro lets you join columns, or change the alignment of a column. Its parameters are:

1. number of columns to merge

2. preamble

3. content

```
\begin{tabular}{|lrr|}
\hline
& \multicolumn{2}{c|}{Products} \\
\cline{2-3}
& \multicolumn{1}{c}{\textit{B.}}
& \multicolumn{1}{c|}{\textit{C.}} \\
\cline{2-3}
...
```

	Products	
	B.	C.
2000	910.1	5.7
2001	1111.7	6.3
2002	1112.2	66.5

5.4 Decimal alignment

Often, you can simply right-align, since typically all data in a column are specified with the same number of decimal digits. This is the case with the Butter / Cheese examples above.

If this isn't the case, you can put the following code in your preamble:

```
\usepackage{dcolumn}
\newcolumntype{d}[1]{D{.}{.}{#1}}
```

This lets you use column types d{n.m} with *n* digits before the decimal point and *m* after:

```
\begin{tabular}{|l|d{4.2}|d{4.1}|}
\hline
2000 &  910.1 & 5.7 \\
2001 & 1111.77 & 6 \\
2002 & 1112.2 & 6666.5 \\
\hline
\end{tabular}
```

2000	910.1	5.7
2001	1111.77	6
2002	1112.2	6666.5

5.5 Text columns

For multiline texts, there is the p{. . . } column specification:

```
\begin{tabular}{|lp{1.65in}|}
  \hline
  array  & An improved implementation of \LaTeX's
           tabular and array environment\\
  dcolumn & Provides decimal and other alignment
           for tabular- and array environments\\
\hline
\end{tabular}
```

array	An improved implementation of LaTeX's tabular and array environment
dcolumn	Provides decimal and other alignment for tabular- and array environments

Usually, text cells are far too narrow for good justification. Here, ragged right would be much better. This can be done with the array package, which provides syntax for adding LaTeX code before (and after) each column entry:

```
\usepackage{array}
\newcolumntype{P}[1]{%
  >{\raggedright\hspace{0pt}\arraybackslash}p{#1}}
\begin{tabular}{|l|P{1.65cm}|}
\hline
What is \TeX? & \TeX{} is a programming
 language for typesetting.\\
\hline
\end{tabular}
```

What is TeX?	TeX is a programming language for typesetting.

See the documentation of the array- and dcolumn packages for additional details on typesetting tabulars.

5.6 Floating tables

In LaTeX-speak, a table or figure 'floats' when its placement on the page does not necessarily match its placement in the LaTeX source. It may be moved to, *e.g.*, the top or bottom of a page, or get a page by itself, as in table 5.1. We shall discuss floating tables and figures in section 6.4 of the next chapter.

5.7 Shortcuts

Of course, nobody wants to re-key reams of numbers. There are several solutions:

- There is an excel2latex plugin for Excel, available from CTAN, that can create a LaTeX source with a tabular environment from a spreadsheet range. It took me some googling to get this package properly installed in MS Office 2010.
- Gnumeric is a spreadsheet program that can read OpenOffice/LibreOffice spreadsheets and export to LaTeX, although without a preamble. It was originally created for Linux, but Windows binaries are also available.

- There is a LaTeX package `odsfile` that can read OpenOffice/LibreOffice spreadsheets directly, *e.g.*:

```
\usepackage{odsfile}
...
\begin{tabular}{...}
\includespread[file=filename.ods,range=a3:f8]
\end{tabular}
```

  This package requires the lualatex engine, i.e. you need to compile your LaTeX source with lualatex instead of pdflatex. `odsfile` is part of our TeX Live installation. Search the documentation list (see 2.6.1) for 'odsfile'.

- If your data are in a simple text format, or at least in a reasonably simple binary format, it may be a nice programming exercise to convert them into LaTeX. Spreadsheets can export to `.csv`; which is such a format. Gnuplot is another such format. Search the documentation list for 'csv' or 'gnuplot' for existing solutions.

5.8 Practice

Do not forget to load the `array`- and `dcolumn` packages in the preamble[1]:

```
\usepackage{array,dcolumn}
```

No doubt, you have lots of tables and spreadsheets of your own to convert to LaTeX. Otherwise, you can find various table examples in Chapter 8 of *Unix Text Processing*, an old Unix text which has been republished in O'Reilly's Open Book Project: `http://oreilly.com/openbook/utp/`.

The practice zip, see section 2.7, contains:

- an example file `tabulars_sample.tex`
- Various files `some_data...` which together illustrate getting spreadsheet data into LaTeX.

---

1. Actually, dcolumn already loads array so there is no real need to load array explicitly.

6 Graphics for LaTeX

Broadly speaking, there are two ways to get pictures into your LaTeX output:

1. Create graphics externally, and load them with LaTeX commands

2. Add picture code directly to the LaTeX source.

The TikZ package offers a convenient general-purpose set of macros for programming diagrams, and there are several other options. However, in this course we shall only look at external graphics.

6.1 External graphics

Before we go any further, you should have some rudimentary understanding of graphics file formats. The most important distinction is between bitmaps and vectors.

Bitmaps are built up from pixels, *i.e.*, tiny blocks of solid color. The smaller the blocks, the sharper the picture and the bigger the file. If you scale them up too far, the blocks become
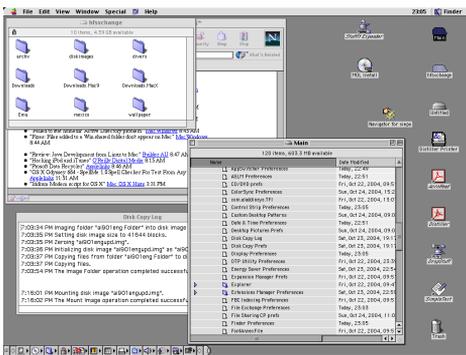

Figure 6.1. Bitmapped– or raster graphics: above a photograph, below a screenshot, both with an enlarged detail at the right
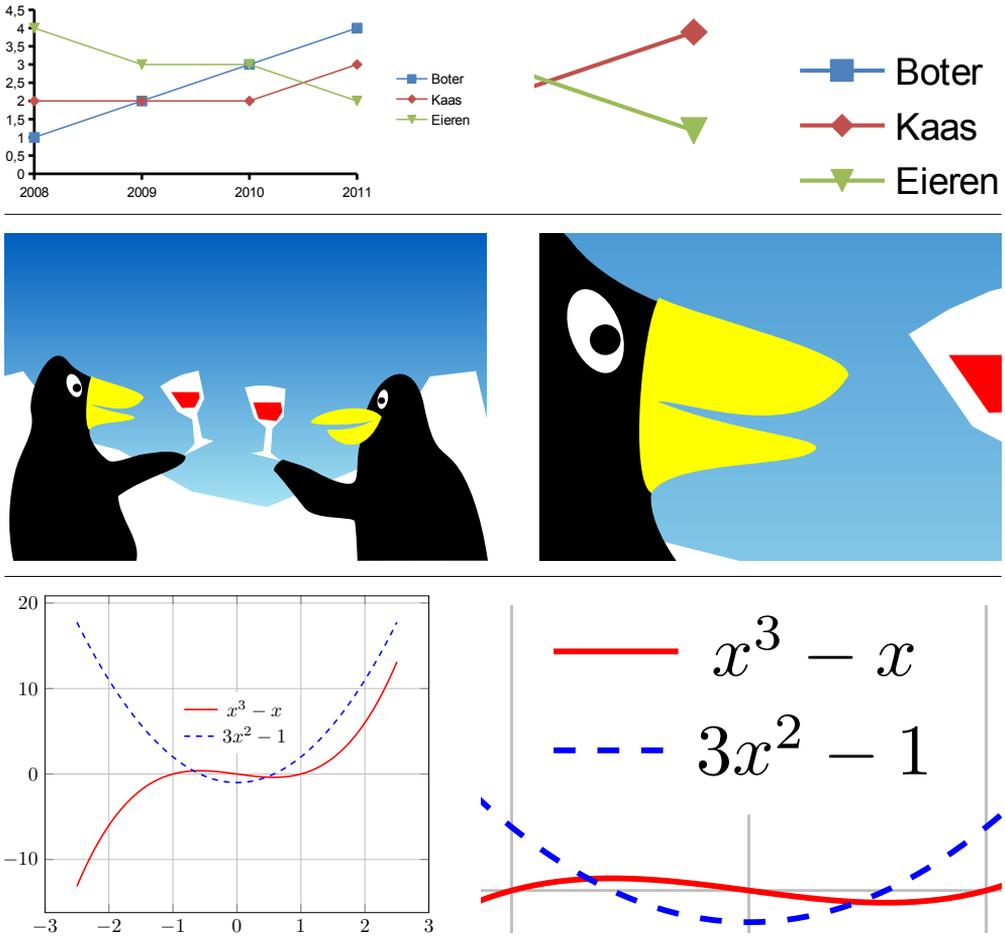
Figure 6.2. Vector art: a LibreOffice data plot, a drawing created with Skencil and Inkscape and a function plot generated with pgfplots

apparent, see figure 6.1.

Vector graphics are built up from mathematical shapes: lines, arcs, bézier curves, text objects, see figure 6.2. They scale well. Avoid converting vector graphics to bitmap.

Pdflatex and the other TeX engines can only work with certain types of graphic files:

pdf can contain both bitmapped and vector elements.

eps is closely related to pdf and can also contain both bitmapped and vector elements. It will be converted behind the scenes to pdf, at least if the TeX installation allows it[1].

png is a bitmapped format. It is first choice for screenshots.

---

1. If you need more control over the eps to pdf conversion, or need conversion the other way, or need to crop margins, have a look at epspdftk, available as the *PostScript- and pdf conversions* utility in the Utilities submenu of the RuG TeX Live launcher, and at its command-line back end epspdf.

Figure 6.3. Raster and vector combined

jpg   or jpeg is a bitmapped format with lossy compression[2]. It is first choice for photo-
graphic images.

6.2   Producers of graphic files

Mathematical software (R, MATLAB, Octave, Gnuplot) can generate eps and sometimes pdf.

Professional illustration software can usually export to eps and pdf. Inkscape is a capable
free alternative to commercial products such as Adobe Illustrator and CorelDRAW.

OpenOffice/LibreOffice and MS Office can export documents and selections of documents
to pdf.

Figure 6.2 shows two vector graphic files created by external programs and one created
by a LaTeX macro package.

I am not going to list programs for bitmapped graphics. There are many good ones, often
free or inexpensive.

Download *Figures in LaTeX* for a more in-depth although not quite up-to-date discussion.

6.3   Including an external graphics file

Graphics inclusion is not built into the LaTeX core. The graphicx package provides this
facility. You need to load it in the preamble with

```
\usepackage{graphicx}
```

You can place a figure in your document with code such as

```
\includegraphics{A_picture}
```

Normally, you don't need to specify the extension. Pdflatex will look for `A_picture.jpg`,
`A_picture.png` and `A_picture.pdf`.

With the above code, the graphic file should be in the same directory as your .tex file.
With a command

```
\includegraphics{figures/A_picture}
```

---

2. To reduce file size, bitmapped images are usually compressed. For png this is done in a lossless way, *i.e.*, the
decompressed image and the original image contain identical information. Jpeg is compressed in a lossy way, *i.e.*,
information gets lost. However, jpeg compression works very well for photographic images, which can be reduced
to 10% of their original file size without visible loss of quality.

pdflatex will look in the `figures` subdirectory.

Make sure to use a relative path, forward slashes and no spaces or funny characters in file- or directory names: 'figures/Apicture' is fine, 'c:\Documents and Settings\your name \A picture' is not. The TeXstudio Insert Graphics wizard tries to produce the right syntax.

If the picture is too large or too small, you can scale it to the desired size with a `width` or `height` parameter:

`\includegraphics[height=.3in]{figures/mouse}`

'width=\linewidth' may also come in handy.

You can also rotate a picture with an `angle` parameter. Figure 6.4 has been inserted with

`\includegraphics[width=.7in,angle=180]{figures/mouse}`

6.4 Floating figures and tables

If you place large objects such as figures or tables at their natural position in the text stream, you tend to get awkward page breaks. Therefore, they are usually placed inside a 'float', which means in LaTeX-speak an environment which may be moved elsewhere: to, *e.g.*, the top or bottom of a page, or to a page by itself.

LaTeX defines two float environments: the `table`- and the `figure` environment. It is possible to define more. Figure- and table floats are numbered separately.

Within both environments, a `\caption` command is defined. In the examples below there is a `\label` command after the `\caption` command for cross-referencing.

Table 5.1 on page 24 has been placed with the following code:

```
\begin{table}[b]
  \caption{Fewer rules are usually better}
  \label{tab:rules}
  \centering
  \begin{tabular}[t]{lrr}
  ...
  \end{tabular}
\end{table}
```

and Figure 6.4 on page 30 with:

```
\begin{figure}[b]
  \centering
  \includegraphics[width=.7in,angle=180]{figures/mouse}
  \caption{An upside-down figure}\label{fig:float}
\end{figure}
```
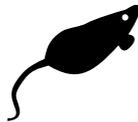
Figure 6.4. An upside-down figure

Codes [t] [b] or [tb] are optional *placement specifiers.* They indicate preferred placement of the float on the page: bottom (b), top (t), here (h) or (float) page (p). Default: [tbp].

Note also the \centering command for centering the content of the environment. This command has no effect on the caption.

If you have many floating figures and tables, it helps placement if you have some or all of the following commands in the preamble[3]:

```
\setcounter{topnumber}{2}
\setcounter{bottomnumber}{2}
\setcounter{totalnumber}{3}
\setcounter{dbltopnumber}{2}
\renewcommand{\topfraction}{.9}
\renewcommand{\textfraction}{.1}
\renewcommand{\bottomfraction}{.75}
\renewcommand{\floatpagefraction}{.9}
\renewcommand{\dblfloatpagefraction}{.9}
\renewcommand{\dbltopfraction}{.9}
```

With these commands, LaTeX is more willing to put several floats on a single page and to devote a larger portion of the page to floats without resorting to a dedicated float page.

Wrapping text around a figure requires an additional package. There are several to choose from, but the *CTAN Catalogue* recommends wrapfig and floatflt.

6.5 Practice documents for graphics and floats

The file float_sample.tex demonstrates both graphics inclusion and floats (several figures and one table).

The figures subdirectory contains graphics files used in float_sample.tex. All the files in this directory, with the exception of diamond.eps, can be loaded directly by pdflatex, and the latter file will be converted on-the-fly to pdf.

---

3. You can copy-and-paste this code from the practice file float_sample.tex.
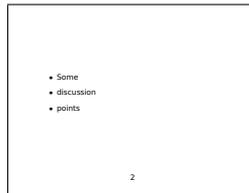
7 Presentations

Currently, the most popular presentations package is Beamer, and that is the package that we are going to discuss.

7.1 Alternatives

However, there are alternatives. For instance, if you have minimalistic tastes then you could simply set up suitable page dimensions with the geometry package:

```
\usepackage[%
 paperwidth=108mm,
 paperheight=81mm,
 width=88mm,
 height=62mm,
 top=9mm,
 footskip=20pt]{geometry}
```

For my own presentation classfiles I start out along these lines.

Other presentation classfiles besides Beamer are `seminar`, `prosper` and `powerdot`.

7.2 Getting started with Beamer

Beamer comes with elaborate but unwieldy documentation; search the documentation list (see 2.6.1) for 'beameruserguide.pdf'.

For a faster start, I added `beamer_sample.tex` to the practice files. You can also dig up the 'solutions' files from the official documentation under the *<TEX Live root>*\texmf-dist\doc\ latex\beamer\solutions folder.

7.3 Slides are frames

Beamer presentations consist of series of frames:

```
\documentclass{beamer}
...
\begin{frame}{Frame title}
  some content
\end{frame}

\begin{frame}
  \frametitle{Another title}
  more content
\end{frame}
```

The frame title can be specified as the second argument to `\frame`, via a `\frametitle` command, or omitted altogether.

There are various ways to reveal a frame in a stages. In Beamer terminology, these successive stages are *overlays*. A simple way to create them is with the \pause command:

```
\begin{frame}
  \frametitle{Points}
  \begin{itemize}
  \item Some
    \pause
  \item discussion
    \pause
  \item points
  \end{itemize}
\end{frame}
```
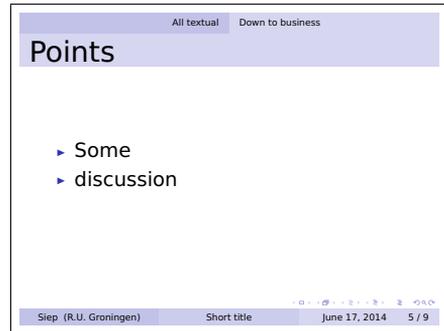

However, there are far more complicated options for overlays. Chapter 9 of the Beamer manual gives more details.

7.4 Themes

Beamer uses themes to control different aspects of the presentation: layout, colors, fonts and headers and footers. The manual shows examples of different themes such as the default theme (no \usetheme command), Antibes, Bergen, Madrid and PaloAlto.

Default theme
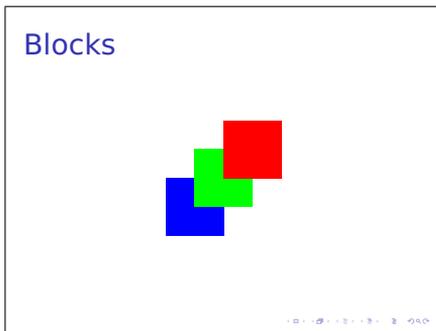
PaloAlto
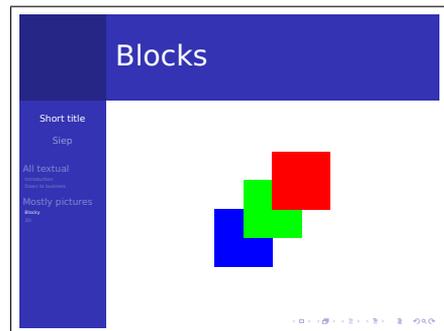
Instead of such a comprehensive theme, you can also load component themes. The example from section 7.3 uses:

```
\useoutertheme{infolines} % info at top and bottom
\usecolortheme{seahorse}  % color theme
```

Read Part III of the manual for details.

7.5 Modes

Beamer makes it possible to combine an article and a presentation into a single source. There is a \mode<thismode>{...} command to tell Beamer that the contents between braces only applies to *thismode*, where *thismode* can be presentation or article.

7.6 What about sections?

You can use sectioning commands between frames. They may or may not be used in presentation mode, depending on your theme: some themes will display them in the page header or in a sidebar; see the illustrations in section 7.3 and 7.4. They will also be listed by a \tableofcontents command, which you can put into a frame.

7.7 Figures and tables

In a presentation, there is not much point in 'floating' an object. Beamer provides non-floating figure- and table environments for people who want the associated captioning, numbering and cross-referencing.

7.8 Practice

Play around with beamer_sample.tex from the zipfile and with the solution templates from the Beamer documentation. Things to try:

• Display bulleted lists progressively by inserting \pause commands.
• Include graphics, either with or without a figure environment.
• Try out various themes.
• See how sectioning commands show up in the output under different themes.

A Changing the appearance

This chapter is not part of the course, but people who are particular about the looks of their documents can find here some tips to modify the appearance of a document. These tips use only preamble commands, staying within the spirit of LaTeX.

A.1 Empty lines instead of paragraph indentation

Use the `parskip` package. Add the following line to the preamble:

```
\usepackage{parskip}
```

The left sample below is typeset without, the right one with this package:

It was equally impossible to do the plainest right and to undo the plainest wrong without the express authority of the Circumlocution Office.

It was equally impossible to do the plainest right and to undo the plainest wrong without the express authority of the Circumlocution Office.

If another Gunpowder Plot had been discovered half an hour before the lighting of the match, nobody would have been justified in saving the parliament until there had been half a score of boards, half a bushel of minutes and a family-vault full of ungrammatical correspondence, on the part of the Circumlocution Office.

If another Gunpowder Plot had been discovered half an hour before the lighting of the match, nobody would have been justified in saving the parliament until there had been half a score of boards, half a bushel of minutes and a family-vault full of ungrammatical correspondence, on the part of the Circumlocution Office.

This also takes care of vertical spacing of `itemize`- and `enumerate` environments. This is still just a quick hack; for a professional result all measurements should be harmonized.

A.2 Double-spacing

This looks awful, but is often demanded for draft printouts. A line

```
\usepackage[doublespacing]{setspace}
```

or, less radically

```
\usepackage[onehalfspacing]{setspace}
```

in the preamble will do the trick.

A.3 Display math alignment

A documentclass option `fleqn`:

```
\documentclass[fleqn]{article}
```

ensures that displayed equations are not centered but left-aligned, with a fixed indentation from the left. The left sample below has the default centered alignment of equations. The right one has the option applied and has left-aligned equations:

$$\Delta \ln \left(\frac{Q}{L} \right)_{0,T} = c_0 + \gamma \left(\frac{I^G}{Q} \right)_{0,T} + \delta \qquad (1)$$

$$\Delta \ln \left(\frac{Q}{L} \right)_{0,T} = c_0 + \gamma \left(\frac{I^G}{Q} \right)_{0,T} + \delta \qquad (1)$$

The price and demand elasticities can now be *calculated by*:

$$\varepsilon_{Cj} = \frac{\partial \ln C^*}{\partial \ln j} = \frac{\partial C^*}{\partial j} \frac{j}{C^*} \qquad (2)$$

The price and demand elasticities can now be *calculated by*:

$$\varepsilon_{Cj} = \frac{\partial \ln C^*}{\partial \ln j} = \frac{\partial C^*}{\partial j} \frac{j}{C^*} \qquad (2)$$

A.4 Page dimensions

```
\usepackage[textwidth=10cm,textheight=17cm]{geometry}
```

There are a lot of options, also for page headers and -footers, and for an asymmetric layout. Again, search the documentation list (see 2.6.1).

A.5 Font size

For a slightly larger font, use the 11pt- or 12pt document class option:

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

The default is 10pt. This only works for these predefined sizes. The ext- classes from the extsizes package provide a few more sizes. Otherwise, you are in for a fair amount of coding.

A.6 Fonts

Several packages change the font for the entire document. However, good math fonts are in short supply, so choices are limited if you want matching math typesetting. Check out *LaTeX font Catalogue* or search for 'font' in the *CTAN Catalogue*. The typeset samples below have the required preamble commands on the left. Palatino:

```
\usepackage[TS1,T1]{fontenc}
\usepackage{amsmath}
\usepackage{newpxtext,newpxmath}
```

$$\Delta \ln\left(\frac{Q}{L}\right)_{0,T} = c_0 + \gamma \left(\frac{I^G}{Q}\right)_{0,T} + \delta \tag{1}$$

The price and demand elasticities can now be *calculated by*:

$$\varepsilon_{Cj} = \frac{\partial \ln C^*}{\partial \ln j} = \frac{\partial C^*}{\partial j}\frac{j}{C^*} \tag{2}$$

The Bitstream Charter-based font setup of a previous edition of these notes:

```
\usepackage{amsmath}
\usepackage[charter]{mathdesign}
```

$$\Delta \ln\left(\frac{Q}{L}\right)_{0,T} = c_0 + \gamma \left(\frac{I^G}{Q}\right)_{0,T} + \delta \tag{1}$$

The price and demand elasticities can now be *calculated by*:

$$\varepsilon_{Cj} = \frac{\partial \ln C^*}{\partial \ln j} = \frac{\partial C^*}{\partial j}\frac{j}{C^*} \tag{2}$$

A.6.1 System fonts with XeLaTeX and LuaLaTeX

The XeTeX- and LuaTeX TeX engines support system fonts, including non-latin scripts and modern Unicode-based OpenType fonts. This requires the fontspec package.

Both expect Unicode input, although the usual TeX notations, (*e.g.*, \"e for ë and ` ' for quotes) are also valid. Do *not* use the inputenc package.

XeTeX was originally developed to gain access to Mac OS system fonts. Later, it was ported to Linux and Windows.

LuaTeX has wider ambitions[1], but what matters here is that it has adopted XeTeX's support for system fonts and OpenType fonts.

In many cases you can switch between lualatex and xelatex without changing your LaTeX sources.

Both are available in TeXstudio via the *Tools / Commands* menu, but you can also set one of these as the default via *Options / Configure TeXstudio... / Build / Default Compiler*.

---

1. In section 5.7, we already encountered its ability to read OpenOffice spreadsheets.

Warning. The `fontspec` package needs information about the fonts that it is going to load. XeLaTeX and LuaLaTeX each have their own font database. If they compile a document which uses `fontspec` and they fail to find a font database that they like, then a new one needs to be built.[2] This can take a *long* time.

So please think twice before using `fontspec` on a university workstation while in class!

Cambria. The Cambria font family is especially useful since it contains a full set of mathematical symbols. It is present on most Windows systems and is bundled with, *e.g.*, the free PowerPoint viewer:

```
\usepackage{amsmath}
\usepackage{fontspec}
\usepackage{unicode-math}
\setmainfont{Cambria}
\setmathfont{Cambria Math}
```

$$\Delta \ln\left(\frac{Q}{L}\right)_{0,T} = c_0 + \gamma \left(\frac{I^G}{Q}\right)_{0,T} + \delta \tag{1}$$

The price and demand elasticities can now be *calculated by*:

$$\varepsilon_{Cj} = \frac{\partial \ln C^*}{\partial \ln j} = \frac{\partial C^*}{\partial j} \frac{j}{C^*} \tag{2}$$

Some standard Windows fonts. For less serious applications, you may want to typeset a single passage in a decorative font. If the `fontspec` package is loaded, you can change midway to another font with the `\fontspec` *command*:

```
\usepackage{fontspec}
...
Temporarily switch to {%
\fontspec[Mapping=tex-text]{Comic Sans MS}
Comic Sans MS} and to {%
\fontspec[Mapping=tex-text]{Tahoma}
Tahoma} and back.
```

Temporarily switch to **Comic Sans MS** and to **Tahoma** and back.

A.6.2   OpenType fonts in TeX Live and MiKTeX

TeX Live and MiKTeX contain several OpenType fonts, which can be used with XeLaTeX and LuaLaTeX. If these are not also available as system fonts then selecting them can be a bit tricky, especially with XeLaTeX. The safe solution is to specify them with the full filename, including extension but excluding the directory:

```
\usepackage{amsmath}
\usepackage{fontspec}
\usepackage{unicode-math}
% a fail-safe way of specifying TL fonts
\setmainfont[
  Ligatures=TeX,
  BoldFont=texgyretermes-bold.otf,
  ItalicFont=texgyretermes-italic.otf,
  BoldItalicFont=texgyretermes-bolditalic.otf]%
{texgyretermes-regular.otf}
\setmathfont[
 math-style=ISO,bold-style=ISO,vargreek-shape=TeX]%
{texgyretermes-math.otf}
```

$$\Delta \ln\left(\frac{Q}{L}\right)_{0,T} = c_0 + \gamma \left(\frac{I^G}{Q}\right)_{0,T} + \delta \tag{1}$$

The price and demand elasticities can now be *calculated by*:

$$\varepsilon_{Cj} = \frac{\partial \ln C^*}{\partial \ln j} = \frac{\partial C^*}{\partial j} \frac{j}{C^*} \tag{2}$$

This way, LaTeX can find the font files in the same way as all its other files.

---

2. LuaLaTeX does this automatically.

B Language support

This chapter briefly discusses of LaTeX's support for non-english and multilingual typesetting. This is not part of the course and there are no practice files.

B.1 Babel

TeX and LaTeX supports many languages, also within the same document. For LaTeX, language support is provided by the Babel package. Its principal tasks are proper hyphenation and translation of text strings such as 'Table of Contents' and 'Chapter'.

For, *e.g.*, Dutch hyphenation and Dutch text strings, use the following code in the preamble:

```
\usepackage[dutch]{babel}
```

It is also possible to use several languages in one document:

```
\documentclass{article}
\usepackage[UKenglish,dutch]{babel}
\begin{document}

\tableofcontents

\begin{abstract}
  Samenvatting van het artikel.
\end{abstract}

\section{Een sectie}

Afbreken: overgangsregelingen papierversnipperaar
overl"ydensadvertentie.

\selectlanguage{UKenglish}

\section{An English section}

Enchanted surreptitious  interpretation
disingenuous
\end{document}
```

Inhoudsopgave

1	**Een sectie**	**1**
2	**An English section**	**1**

Samenvatting

Samenvatting van het artikel.

1 Een sectie

Afbreken: overgangsregelingen papierversnipperaar overlijdensadvertentie.

2 An English section

Enchanted surreptitious interpretation disingenuous

See *e.g.* section 2.5 in *The Not So Short Introduction* for more particulars.

B.2 Non–western scripts with XeTeX and LuaTeX

For far-eastern and Arab languages and scripts, we turn to the newer TeX engines XeTeX and LuaTeX.

The Polyglossia package replaces Babel for XeLaTeX and LuaLaTeX. The documentation of this package includes several typeset examples of non-Western scripts, see the documentation list.

C   Bibliographic references

For bibliographic references, LaTeX uses the BibTeX database system. This chapter, which is not part of the course, takes a look at:

- incorporating bibliographic references during the compilation process, see section C.2.1.
- the evolution of citation- and bibliography handling, see section C.3.
- the BibTeX database format, and software to build and maintain such a database, see sections C.6 and C.10.

  BibTeX practice files are in a separate subdirectory `bibtex` of the practice zipfile.

C.1   The why and how of BibTeX

Different journals have different requirements for bibliographic references:

- What to include, in what order
- Capitalization
- Punctuation
- Abbreviations
- Styling of text

With a bibliography manager you can maintain a database of bibliographic references and have it generate the references for any paper in the required format.

Below a very short LaTeX source which references entries from an existing BibTeX database (you can find the full BibTeX entries on page 44):

```
\documentclass{article}
\bibliographystyle{plain}
\begin{document}

See \cite{lcompanion} and \cite{biboostrum}.

\bibliography{bibdemo} % this line specifies bibdemo.bib as database
\end{document}
```

As you can guess, `lcompanion` and `biboostrum` are keys of database entries:

```
@Article{biboostrum,
  author = "Piet van Oostrum",
  title = "Een tutorial over het gebruik van {Bib{\TeX}}",
  year = 2004,
  ...
}
@Book{lcompanion,
  Author = "Frank Mittelbach and Michel Goossens",
  Title = "The {\LaTeX{}} Companion",
```

```
    year = 2004,
    ...
}
```

We get the following output:

See [1] and [2].

References

[1] Frank Mittelbach and Michel Goossens. *The LaTeX Companion*. Addison-Wesley, 2nd edition, 2004.

[2] Piet van Oostrum. Een tutorial over het gebruik van BibTeX. *MAPS*, 30:66–86, 2004.

You can put the `\bibliographystyle` command anywhere before the `\bibliography` command.[1]

Bibliography styles are provided by `.bst` files: `plain.bst` in the example above, `alpha.bst` in the example from section C.3.1; both hidden deep inside the TeX Live directory structure.

Once you have prepared your LaTeX source and your BibTeX database, take the following steps:

1. Run `pdflatex` to generate the information that BibTeX needs

2. Run `bibtex` to generate the list of references

3. Run `pdflatex` to include the list of references

4. Run `pdflatex` again to resolve bibliographic references (just like other cross-references)

and check the typeset output after steps 1, 3 and 4. TeXstudio has a `bibtex` command in the Tools menu. You can also use keyboard shortcuts: F6, F11, F6, F6. Note that after further editing a single `pdflatex` run is enough as long as the list of references stays the same.

The Build button ▶ should do all required steps for you with a single mouse click.

C.2 nocite: entries without citations

You can use `\nocite` instead of `\cite` if you don't want an automatically generated citation in the running text but do want an entry in the list of references:

```
\nocite{lcompanion}
```

Tip. You can quickly create a formatted printout of your BibTeX database with the `\nocite{*}` command (see `testbib.tex` from the practice files):

```
\documentclass{article}
\bibliographystyle{plain}
\begin{document}
\nocite{*}
\bibliography{your_bibtex_file}
\end{document}
```

---

1. LaTeX used to be more restrictive in this respect. Therefore, many old-time LaTeX users still place the `\bibliographystyle` command right before the `\bibliography` command.

C.2.1 Practice

1. Run the `bibtex_sample` example: load `bibtex_sample.tex` in TeXstudio and carry out the four steps listed on page 40.

2. Collect a few fragments with bibliographic references from `http://en.wikipedia.org/wiki/Factors_of_production` into a LaTeX document, replacing footnotes with proper cite commands and a proper list of references. You can use the `factors.bib` bibliography database from the practice files.

C.3 Bibliography styles: three generations

C.3.1 The original BibTeX

The original BibTeX from 1988 uses a simple citation style: the entries in the list of references get an automatically generated label – either numerical or alphanumerical – for cross-referencing. We already saw numerical references. Here is an example with alphanumerical labels:

```
\documentclass{article}
\bibliographystyle{alpha}
\begin{document}

See \cite{lcompanion} and \cite{biboostrum}.

\bibliography{bibdemo}
\end{document}
```

See [MG04] and [vO04].

References

[MG04] Frank Mittelbach and Michel Goossens. *The LaTeX Companion*. Addison-Wesley, 2nd edition, 2004.

[vO04] Piet van Oostrum. Een tutorial over het gebruik van BibTeX. *MAPS*, 30:66–86, 2004.

C.3.2 The next step: author – Year citation styles

The classical bibliography styles didn't provide for author – year citation styles, such as in the examples below:

See Castaldo et al. from 2006 and Knuth (1990).

References

Castaldo, R. J., M. A. McKay, and V. Tosic (2006). Exposing gnu octave signal processing functions as extensible markup language (xml) web services. In *Proc. Canadian Conf. Electrical and Computer Engineering CCECE '06*, pp. 1442–1445.

Knuth, D. E. (1990). *The TeXbook*, Volume A of *Computers and Typesetting*. Reading, MA, USA: Addison-Wesley.

or, with a different bibliography style:

See Castaldo et al. from 2006 and Knuth (1990).

References

R. J. Castaldo, M. A. McKay, and V. Tosic, in *Proc. Canadian Conf. Electrical and Computer Engineering CCECE '06* (2006), pp. 1442–1445.

D. E. Knuth, *The TEXbook*, vol. A of *Computers and Typesetting* (Addison-Wesley, Reading, MA, USA, 1990).

Note the absence of labels in the list of references.

Author-year citation styles are provided by the natbib package and by various alternatives such as the apalike and harvard packages. natbib is the most popular one and can replace most of the others. This is the LaTeX preamble code:

```
\usepackage{natbib}
\bibliographystyle{chicago} % first example
%\bibliographystyle{apsrev} % second example
```

Cite commands in the running text:

```
See \citeauthor{castaldo2006} from \citeyear{castaldo2006} and
\cite{texbook}.
```

A sample of citation commands supported by natbib:

`\cite{tamethebeast}`	Markey, 2005
`\citet{tamethebeast}`	Markey (2005) 'in-text'
`\citep{tamethebeast}`	(Markey, 2005) 'parenthesized'
`\citet[noted] {lshort}`	Oetiker et al. (2011, noted)
`\citep*{lshort}`	(Oetiker et al., 2011)
`\citeauthor{tamethebeast}`	Markey
`\citeyear{lshort}`	2011

`\nocite` works as usual.

See the natbib manual for more variations and for customization options.

The practice files include a file bibtex_natbib_sample.tex to get you started.

C.4 Generating your own bibliography style with custom–bib/makebst

Journals sometimes have very specific requirements as to the formatting of bibliographic entries, without providing a bibliography style implementing this formatting. Even if a suitable style exists, it may be hard to find one, although the *UK TEX FAQ* has to offer some advice. One way out is creating your own .bst file.

If a style is almost correct, *and* if you can make some sense out of .bst files (a big if!), then maybe you can fix it yourself.

Another option is the makebst program. This is a two-step process. In the first step, you have to answer a very long list of questions. Your answers are written to an intermediate answers file, which you can edit afterwards if some answers turned out not to be quite right. In the second step, a .bst file is generated from this list of answers. Search the documentation list (see 2.6.1) for *custom-bib* or *makebst*.

C.5 The latest and greatest: biblatex

A radical reimplementation of bibliography support is biblatex. Bibliography styles aren't written in the unfamiliar .bst syntax but in LaTeX, and the role of BibTeX is reduced to collecting and sorting the bibliographic entries. LaTeX itself selects, arranges and formats the fields of the bibliographic entries. Advantages include:

- many variations in bibliography style can be realized simply with package options, without editing .bst files
- better support for non-Western languages
- more citation options, because LaTeX has access to all the bibliographic information
- easy per-chapter bibliographies

Fortunately, an old BibTeX database is still compatible with biblatex.

Getting started with biblatex

You can easily experiment with biblatex. If you include a package option natbib or natbib=true then you can keep using natbib cite commands in your LaTeX source.

You should also configure TeXstudio to use biber instead of BibTeX (*Options / Configure TeXstudio / Build / Default Bibliography*).

Below are biblatex preamble commands, assuming biber as backend; note that with biblatex the \bibliography command should be in the preamble:

```
\usepackage[style=numeric]{biblatex}
\bibliography{bibdemo}
```

And near the end:

```
\printbibliography
```

The complete source for the first example:

```
\documentclass{article}
\usepackage[style=numeric]{biblatex}
\bibliography{bibdemo}
\begin{document}
See \cite{lcompanion} and \cite{bacgri2003}.

\printbibliography
\end{document}
```

Output:

> See [2] and [1].
>
> ## References
>
> [1] Lance J. Bachmeier and James M. Griffin. "New evidence on asymmetric gasoline price responses". In: *The Review of Economics and Statistics* 85.3 (2003), pp. 772–776.
>
> [2] Frank Mittelbach and Michel Goossens. *The LaTeX Companion*. 2nd. Addison-Wesley, 2004.

An example with author-year citations and the new \citetitle command:

```
\documentclass{article}
\usepackage[bibstyle=authoryear,block=ragged]{biblatex}
\bibliography{bibdemo}

\begin{document}
See \citetitle{lcompanion} by \citeauthor{lcompanion} published in
\citeyear{lcompanion}.

\printbibliography
\end{document}
```

See *The LATEX Companion* by Mittelbach and Goossens published in 2004.

References

Mittelbach, Frank and Michel Goossens (2004). *The LATEX Companion*. 2nd.
Addison-Wesley.

The practice files include a file `bibtex_biblatex_sample.tex` to get you started with biblatex.

By now, there are quite a few biblatex styles (search for '`biblatex-`' in the CTAN Catalogue), and existing styles can be tweaked with options. Still, it is conceivable that none of the existing styles are usable. And there is no makebst (section C.4) for biblatex. It is also possible that the recipient has an antiquated TeX setup and is not willing or not able to handle biblatex. So it is too soon to assign the older solutions to the dustbin.

C.6 The BibTeX database format

This section describes the BibTeX database format. Like a LaTeX source, a BibTeX database is a plain text file. It has an extension `.bib`, and consists of a series of records such as the following:

```
@Article{biboostrum,
  author = "Piet van Oostrum",
  title = "Een tutorial over het gebruik van {Bib{\TeX}}",
  journal = "{MAPS}",
  volume = "30",
  pages = "66--86",
  year = 2004,
}
@Book{lcompanion,
  Author = "Frank Mittelbach and Michel Goossens",
  Title = "The {\LaTeX{}} Companion",
  Publisher = AW,
  year = 2004,
  Edition = "2nd",
}
```

Note the general structure: a BibTEX record consists of:

- The type of publication, *e.g.*, article or book
- A key, *e.g.*, `biboostrum` or `lcompanion`, which is used for citing
- A list of fields

The list of required and optional fields varies with the entry type. You can add additional fields, *e.g.*, as comments for yourself. Any field which is not required or optional will simply be ignored.

For most fields, the values should be enclosed in braces { and }, or in double quotes "". Values which are clearly numbers, such as years and volume numbers, may be entered 'bare'.

You should enclose LaTeX code in an additional set of braces to keep BibTEX from messing with it. You should do the same with all-caps words.

As to accented characters, the safe solution is always to use macros: `{\'e}` rather than é, although with some care accented letters may work ok, see section C.8.1.

C.7 BibTEX editors and bibliography managers

For creation and maintenance of your BibTEX database it is best to pick a program that uses BibTEX as its native format.

Our TEX Live installation includes the BibTEX editor *JabRef*, which is a Java program and therefore runs on all platforms where Java is installed. On Mac OS, *BibDesk* is a popular choice.

Editing manually with your LaTeX editor is another good option.

A general-purpose bibliography manager may work for you, but check its BibTEX export carefully. The university offers access to *RefWorks*. However, its BibTEX support leaves something to be desired.

Zotero is a popular Firefox extension for collecting and managing references. It can create bibliographic entries from *e.g.* Amazon pages. Below the Zotero BibTEX export from such an entry:

```
@book{voss_latex_2011,
  title = {Latex Quick Reference},
  isbn = {1906860211},
  publisher = {Uit Cambridge Ltd.},
  author = {Voss, Herbert},
  month = sep,
  year = {2011}
}
```

Further suggestions can be found at `http://tex.stackexchange.com/questions/33619/`.

There are other services which create BibTEX entries, such as `http://lead.to/amazon/`, or, if you have an ISBN number for a book, you can go to `http://ottobib.com/`. See also `http://tex.stackexchange.com/questions/143/` for more suggestions.
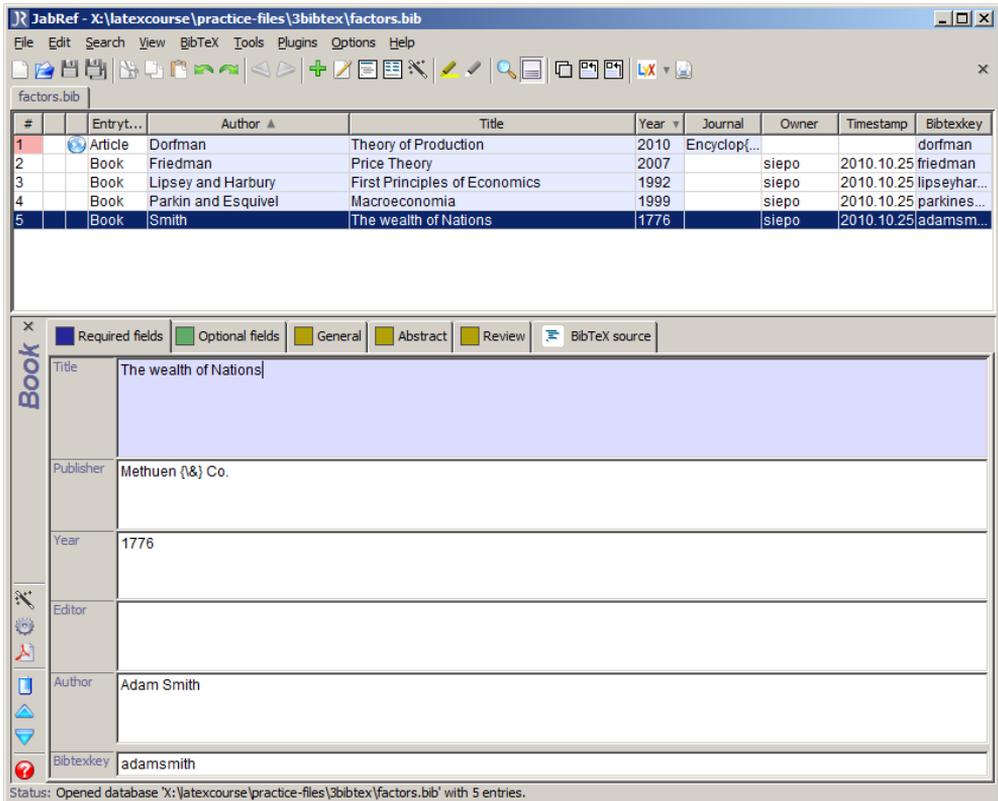
Figure C.1. JabRef BibTeX editor

C.8 Examples

Now let us have a more in-depth look at the BibTeX database format by looking at a series of examples.

```
@TECHREPORT{canond2003,
    author = "Marcel Canoy and Sander Onderstal",
    year = 2003,
    title = "Tight oligopolies: {I}n search of proportionate remedies",
    number = 29,
    institution = "{CPB} Netherlands Bureau for Economic Policy Analysis",
    address = "The Hague",
}
```

TECHREPORT is the type of the publication. Capitalization is not significant in BibTeX entry types and field names.

The key canond2003 is used by the various cite commands.

{CPB} is enclosed in braces to protect it against BibTeX's automatic capitalization.

Also notice that the author field consists of two authors, each in *first last* format. The names are separated with 'and'.

C.8.1 Example: brace delimiters, alternate author syntax, accented letters

```
@Misc{clementsgalvao2001,
  author = {Clementz, Michael P. and Galv{\~a}o, Ana Beatriz},
  title = {A comparison of tests of non-linear cointegration with an application
      to the predictability of {US} interest rates using the term structure},
  year = 2001,
  howpublished = {Mimeo, Department of Economics, University of Warwick},
}
```

This example encloses values in braces rather than double quotes. This makes it possible to use values which include double quotes (nesting braces within braces is never a problem).

The author field uses the alternate syntax of *last, first*. The BibTeX manual (Patashnik (1988)) has more to say about the parsing of author's names.

Note that this entry has a different set of fields. The bibliography style determines which entry types are recognized and which fields are required or optional for each entry type.

Again, braces around US ensure that BibTeX leaves capitalization alone.

Accented characters
The above example includes a macro for an accented letter inside braces: {\~a}. You can use accented letters outright, but it requires care: make sure that the encoding, probably either latin1 or utf8, matches the LaTeX source, and that you include a preamble command

```
\usepackage[enc]{inputenc}
```

where *enc* should usually be utf8 or latin1, unless you use one of the modern TeX engines XeLaTeX or LuaLaTeX, which always expect utf8.

C.8.2 Example: a predefined abbreviation and a dummy field

```
@string{AW = "Addison-Wesley"}
 ...
@Book{lcompanion,
 Author = "Frank Mittelbach and Michel Goossens",
 Title = "The {\LaTeX{}} Companion",
 Publisher = AW,
 year = 2004,
 Edition = "2nd",
 ignorablefield = "too fat for my backpack",
}
```

You can define abbreviations with @string entries. You can also create a .bib file with @string entries, and load it before the actual database file.

This entry also uses a dummy field ignorablefield for private information.

C.8.3 Example: author names with a 'von' part; number ranges

```
@article{meycra2004,
  author = {Meyer, Jochen and von Cramon-Taubadel, Stephan},
  title = {Asymmetric Price Transmission: A Survey},
  year = 2004,
  journal = {Journal of Agricultural Economics},
  volume = 55,
  number = 3,
  pages = {581-611},
}
```

```
@inproceedings{ricejava,
  author = "Istiqomah Istiqomah and Manfred Zeller and
    Stephan von Cramon-Taubadel",
  title = "Volatility and Integration of Rice Markets in Java, Indonesia",
  booktitle = "Tropentag 2005",
  year = 2005,
}
```

These examples feature an author's name with a 'von' part, first in *von last, first* syntax, then in *first von last* syntax. Again, Patashnik (1988) explains handling of author's names.

Also note that BibTeX will expand the range 581-611 to 581--611, producing a proper n-dash in the typeset output.

You can read a full description of the .bib format in the original BibTeX documentation, Patashnik (1988). This documentation doesn't cover extensions from, *e.g.*, the Natbib- and biblatex packages and corresponding bibliography styles.

C.9 The url field

Some bibliography styles, including styles created with makebst/custom-bst, support an url field. It is even required for the biblatex online entry type. It is a good idea to load the url package, to make sure that special characters such as underscores and tildes are rendered correctly, and to enable line-breaking inside the url. LaTeX source:

```
\documentclass{article}
\usepackage{natbib}
\bibliographystyle{plainnat}
\usepackage{url}
...
\nocite{biboostrum}
\bibliography{bibdemo}
\end{document}
```

BibTeX entry:

```
@Article{biboostrum,
 author = "Piet van Oostrum",
 title = "Een tutorial over het gebruik van {Bib{\TeX}}",
 journal = "{MAPS}",
 volume = "30",
 pages = "66--86",
 year = 2004,
 url = "http://www.ntg.nl/maps/pdf/30_15.pdf",
 }
```

results in:

Piet van Oostrum. Een tutorial over het gebruik van BibTeX. *MAPS*, 30:66–86, 2004. URL http://www.ntg.nl/maps/pdf/30_15.pdf.

C.9.1 Urls in other fields
If the bibliography style doesn't use an url field, you can include an url in another field. 'Howpublished' is a good option, but only in combination with, *e.g.*, the Misc entry type,

since it is ignored by most other entry types. Alternatively, the 'Note' field is supported by almost all entry types.

```
@Manual{tamethebeast,
  author = "Nicolas Markey",
  title = "Tame {t}he {BeaST}, The B to X of {BibTeX}",
  year = 2005,
  note = "{\url{http://www.lsv.ens-cachan.fr/~markey/BibTeX/doc/ttb_en.pdf}}",
}
@Misc{some,
  author = "Au Thor",
  title = "Some title",
  howpublished = "\url{http://www.a.site.net/sometitle.html}",
}
```

LaTeX source:

```
\documentclass{article}
\bibliographystyle{plain}
\usepackage{url}
...
\nocite{*}
\bibliography{nourl}
\end{document}
```

[1] Nicolas Markey. *Tame the BeaST, The B to X of BibTeX*, 2005. `http://www.lsv.ens-cachan.fr/~markey/BibTeX/doc/ttb_en.pdf`.

[2] Au Thor. Some title. `http://www.a.site.net/sometitle.html`.

The natbib reimplementations plainnat, abbrvnat and unsrtnat of the corresponding classical bibliography styles do provide a url field and do not require the above workarounds.

More trickery for working around BibTeX's automatisms can be found in Markey (2005) and in the BibTeX chapter of the *UK TeX FAQ*.

C.10 Practice

Add entries to bibdemo.bib and check your work with testbib.tex and the \nocite{*} command, as described in section C.2.

A suggestion: Find a Wikipedia page with many references, *e.g.* `http://en.wikipedia.org/wiki/Economics`. Locate entries with enough bibliographic information and turn them into BibTeX entries. Or get more bibliographic details, or even a complete BibTeX entry via *e.g. Google Scholar* and other resources mentioned in `http://tex.stackexchange.com/questions/143/`.

C.11 Troubleshooting

C.11.1 Random things to try

Rerun LaTeX. Maybe you just need another LaTeX run to resolve the \cite commands.

Problems with old auxiliary files. Sometimes it helps to start with a clean slate by getting rid of old auxiliary files: click on 'Clean' on the Tools menu and rebuild.

This may be necessary if an error or incompatibility in one of the auxiliary files prevents

LaTeX from continuing. An incompatibility may arise if you change something in the bibliographic options.

Input encoding. Maybe there is a mismatch between the encoding of the BibTeX database export and the LaTeX source. Biblatex has a `bibencoding` option.

C.11.2   JabRef

Make sure you run only one copy of JabRef. In particular, do not load one BibTeX file into two JabRef sessions, which can easily happen.

Click *File / Save database* to make sure that BibTeX or biber gets up to date information.

C.11.3   Get more information

In TeXstudio, try to get more detailed information, *e.g.*, by clicking on the error tab of the tabbed pane under the edit area.

Log files can also be useful, but often contain masses of useless gibberish. The above-mentioned tabbed pane has a tab for the LaTeX log, but not for the BibTeX log, which may be more useful. You can load the BibTeX log into TeXstudio anyway: click on *File > Open*. Make sure that 'Files of type' is set to 'All files(*)' and then select `filename.blg`, assuming that the LaTeX file is called `filename.tex`. Another file to look at is `filename.aux`.

References

American Mathematical Society (1999–2002). *User's Guide for the* amsmath *Package*. Part of the amsmath package; included in most free TeX distributions.

Array package (2011). A new implementation of LaTeX's tabular and array environment, included in most TeX distributions.

BibDesk. GUI bibliography manager for the Mac. URL: http://bibdesk.sourceforge.net/.

CTAN. The Comprehensive TeX Archive Network. URL: http://mirror.ctan.org/.

CTAN Catalogue. URL: http://mirror.ctan.org/help/Catalogue/brief.html.

Dcolumn package (2001). Provides decimal and other alignment for tabular- and array environments, included in most TeX distributions.

Fairbairns, Robin, ed. *UK TeX FAQ*. URL: http://faq.tug.org/.

Getting Started with TeX, LaTeX, and Friends. URL: http://www.tug.org/begin.html.

Google Scholar. URL: http://scholar.google.com/.

JabRef. GUI bibliography manager written in Java. URL: http://jabref.sourceforge.net/.

Jørgensen, Palle, ed. *LaTeX font Catalogue*. URL: http://www.tug.dk/FontCatalogue/.

Knuth, Donald E. (1986). *The TeXbook*. Vol. A. Computers and Typesetting. Addison-Wesley.

Kopka, Helmut and Patrick W. Daly (2003). *Guide to LaTeX*. 4th ed. Addison-Wesley.

Kroonenberg, Siep (2012). *Figures in LaTeX*. URL: http://tex.aanhet.net/rugtex/figlatex.pdf.

Lamport, Leslie (1994). *LaTeX, A document preparation system*. 2nd ed. Addison-Wesley.

Lehman, Philipp (2012). *The biblatex Package*. Published as part of the biblatex package.

TeX Live. URL: http://www.tug.org/texlive/.

Markey, Nicolas (2005). *Tame the BeaST*. Exhaustive BibTeX tutorial, available from CTAN. URL: http://www.lsv.ens-cachan.fr/~markey/BibTeX/doc/ttb_en.pdf.

NTG, Dutch-language TeX Users Group. URL: http://www.ntg.nl/.

Oetiker, Tobias et al. (2011). *The Not So Short Introduction to LaTeX2_ε*. Included in most free TeX distributions. URL: http://mirror.ctan.org/info/lshort/.

Oostrum, Piet van (2004). "Een tutorial over het gebruik van BibTeX". In: *MAPS* 30, pp. 66–86. URL: http://www.ntg.nl/maps/pdf/30_15.pdf.

Pakin, Scott (2009). *The Comprehensive LaTeX Symbol List*. URL: http://mirror.ctan.org/info/symbols/comprehensive/.

Patashnik, Oren (1988). *BibTeXing*. Included in most TeX distributions.

RefWorks. Web-based reference manager. URL: http://www.refworks-cos.com/.

RuG TeX/LaTeX installation, homepage. URL: http://tex.aanhet.net/rugtex/.

Shell, Michael and David Hoadley (2007). *BibTeX tips and FAQ*. URL: http://mirror.ctan.org/biblio/bibtex/contrib/doc/btxFAQ.pdf.

Tantau, Till, Joseph Wright, and Vedran Miletić (2011). *The BEAMER class*. Published as part of the beamer package.

TeXstudio. URL: http://texstudio.sourceforge.net/.

TUG, TeX Users Group. URL: http://www.tug.org.

Zotero. Firefox extension for collecting and managing references. URL: http://www.zotero.org/.