LaTeX-BEAMER Course

Katharina Hoff

June 2, 2007

Contents

1 Introduction to LaTeX

LaTeX is a document preparation system for high quality typesetting. It provides an easy access to the powerful TeX language introduced by Knuth (1979). LaTeX can be used to create almost any type of document, ranging from scientific arcticles to books and conference posters. This script gives an introduction on how to create scientific presentations with LaTeX using the BEAMER-class.

1.1 Literature Recommendations for LaTeX

This script gives only a very short summary on the most basic functions of LaTeX. If you are considering to use LaTeX in your daily life, I recommend you to get a book. One very complete and helpful example is **The LaTeX Companion**, second edition, by Frank Mittelbach and Michel Goossens, Addison Wesley, 2004, ISBN 0-201-36299-6. When I last checked, the price was 60 US$. Much cheaper and much more compact, though in German, is **LaTeX kurz & gut** by Matthias Kalle Dahlheimer, O'Reilly, 1998. This book gives a very concise reference for the most important commands.

1.2 Obtaining LaTeX for your own Computer

In order to use LaTeX on any computer, three parts need to be installed:

1. LaTeX itself,
2. a text editor and
3. a document viewer.

1.2.1 LaTeX

LaTeX ('Lamport's TeX'(Lamport, 1999)) is a macro-package that enables the user to 'translate' text with markup-commands to a typesetted document. The system requirements of LaTeX are comparably low and free distributions are available for almost any of today's computer systems. These are links to the distributions for three of the most commonly used platforms:

- Windows: http://miktex.org
- MacOS: http://www.tug.org/mactex
- Linux: teTeX is already contained in most Linux distributions.

Having installed LaTeX on your system, you can now 'compile' a *.tex file to e.g. *.dvi in the command line. Nowadays, many graphically embedded LaTeX environments are available as LaTeX-Editors.

1.2.2 Text Editor

Any text editor may be used for editing a *.tex-file which may subsequently be compiled via the command line. However, graphically embedded LaTeX-environments are much more convenient for most users. They usually contain a text-window where the source file is edited, several pop-up menus which offer – among other features – LaTeX command insertion, direct access to the LaTeX-documentation and buttons for source file compilation to several formats. Some useful LaTeX-editors are:

- Windows: TeXnicCenter http://www.toolscenter.org

- MacOS: TeXShop http://www.uoregon.edu/ koch/texshop
- KDE Linux: Kile http://kile.sourceforge.net

1.2.3 Document Viewer

Natively, LaTeX compiles to dvi (device independent) file format. The *.dvi-file contains a 'device-independent' description of the text layout. dvi can be accessed by so-called drivers, e.g. for document preview, but usually, the *.dvi-file will be further converted to e.g. pdf or ps format. For document preview, a dvi-viewer will be needed (usually, either your system already contains a document viewer that is dvi-compatible, or your integrated LaTeX environment will bring one along). Just in case you realize that you are missing a dvi-viewer, Wikipedia links to several good dvi-viewers:
http://en.wikipedia.org/wiki/DVI_(file_format).

As mentioned above, dvi-files are usually converted to more popular file formats post compilation. It is a good idea to have Acrobat Reader installed (http://www.adobe.com/de/products/acrobat) for viewing full functionality .pdf-files. GhostView (http://www.cs.wisc.edu/ ghost) provides access to *.ps-files (and *.pdf-files).

A special version of LaTeX, pdflatex, compiles *.tex directly to *.pdf-files. The usage of pdflatex is particularly interesting if you want to include *.jpg or *.png files in your document (LaTeX natively only supports *.eps).

2 Getting Started with LaTeX

LaTeX is a markup language which uses certain commands to indicate text formatting for the compiler. The source-file of every LaTeX-document has the file-ending *.tex. The article class, which is meant to format scientific articles, will be used in order to introduce the main features of LaTeX. A typical *.tex file with the article class looks like this:

```
\documentclass[a4paper,10pt]{article}

%opening... % is the comment sign in LaTeX!
\title{My First \LaTeX{}-document}
\author{XXX}

\begin{document}

\maketitle % will create a title containing author, title and date

\begin{abstract}
Insert your abstract here!
\end{abstract}

\section{This could be the first main section-title}
Insert the main text for your first section here!

\end{document}
```

After giving this file a name with the ending *.tex, it may be compiled with LaTeX or pdflatex in the command line or via the compilation button of your LaTeX editor. For various reasons, you should compile your document twice, or better three times before opening it. This is particularly important to get all references correctly counted.
Important things to keep in mind when editing *.tex-files at all times:

Command line:
`latex my.tex`
or
`pdflatex my.tex`

1. If you `\begin` something (e.g. the document, or a figure, or a table), then you also have to `\end` it before the document ends.
2. Pay attention which type of brackets you are using!
3. % is the sign for a 'comment' which means that everything following this sign will be ignored by LaTeX.

2.1 Document Structure

LaTeX offers structuring commands which facilitate giving your document a logical structure. The ARTICLE CLASS supports in hierarchic order:

```
\part{Headline of your part}
\section{Headline of your section}
\subsection{}
\subsubsection{}
\paragraph{}
\subparagraph{}
```

A structure command does not only format your headline with the appropriate font-size and weight, but also gives a heading number that reflects the hierarchic order, stores the headline for the table of contents (and if enabled for header or footer of the current page).

The table of contents will be produced by the `\tableofcontents` command (see figure 1 on page 5).

2.2 Text Formatting

```
Normal Text
\textit{Italic Text}
\textbf{Bold Text}
\textsc{Yet another format}
```

Normal Text
Italic Text
Bold Text
YET ANOTHER FORMAT

2.3 Lists

The `\itemize` environment will present every `\item` of a list after a globally specified symbol (mostly a black dot by default) or after a symbol or word given in edgy brackets:

```
\documentclass[a4paper,10pt]{article}

\usepackage{graphicx} % <- preamble: include graphicx-package for figures!

\title{My First \LaTeX{}-document}
\author{XXX}

\begin{document}

\maketitle

\begin{abstract}
Insert your abstract here!
\end{abstract}

\tableofcontents % <- a table of contents containing the headers of
parts, sections, subsections and subsubsections will be displayed at this
position in the document.

\section{This could be the first main section-title}
Insert the main text for your first section here!

\begin{figure}
 \centering % <- not always necessary
 \includegraphics[width=10cm, height=5cm]{myfigure.png} % <- giving
                                         % height & width is optional
 \caption{Your first beautiful figure.} % <- figure caption
 \label{myfig:1} % <- figure label for referring to the figure in text
\end{figure}

You may refer to your figure with \ref{myfig:1} on page
\pageref{myfig:1}.

\end{document}
```

Figure 1: Example *.tex-file with figure insertion and table of contents generation.

```
\begin{itemize}
 \item First item
 \item Second item
 \item[*] Third item
\end{itemize}
```

- First item
- Second item

* Third item

The `\enumerate` environment replaces the symbol by the number of the current `\item`:

```
\begin{enumerate}
 \item First item
 \item Second item
 \item Third item
\end{enumerate}
```

1. First item
2. Second item
3. Third item

2.4 Inserting Figures

One way of including figures in a LaTeX document is using the graphicx-package. 'Using a package' means that you have to 'include' the specified package in the 'preamble' of your document, meaning after \documentclass but before \begin{document}. Different image formats are supported by different drivers, e.g. you may use *.png and *.jpg with pdflatex, but *.eps will only work with LaTeX, not with pdflatex. As for we will mostly use pdflatex for typesetting presentations, consider converting your *.eps images with another external software like Photoshop or GIMP (http://www.gimp.org).

The \includegraphics command is used for inserting an external figure into your document. (Keep that in mind for for inserting figures LaTeX presentations!). In order to make you figure a float object with a caption and a label, you need to put the image into a so-called figure environment. It begins with a \begin{figure} and ends with an \end{figure}. A float object is an object for which LaTeX tries to find the optimal position within the document, thereby 'floating' it to an appropriate position.

A figure caption is entered below \includefigure and after the \caption command (see example). A label which may be used to refer to this picture from the text-body is inserted after the command \label. You may also put the \label command inside the \caption brackets, but never put label **before** caption. An example for figure insertion is shown in figure 1 on page 5.

2.5 Formatting Tables

Formatting tables in LaTeX requires some attention and can be painful. I will further on describe the basic principles of typesetting a simple table in LaTeX from scratch, which will in most cases be sufficient for tables in a presentation. However, there are several ways to circumvent the procedure of manually typesetting tables in LaTeX. One possibility is to edit your table in gnumeric (http://www.gnome.org/projects/gnumeric/downloads.shtml). Gnumeric is a free spreadsheet software, which is able to read and write most table formats, e.g. various Microsoft Excel formats. The option File -> Export offers export to LaTeX-format. The resulting file may be compiled as a stand alone document or you can copy/paste the table part (which is well commented) into any of your other *.tex documents. In that case, make sure to include all required packages in the preamble of your document (also copy/paste).

2.5.1 Using the tabular Environment

The easiest way to edit a table from scratch is using the tabular environment. After the tabular command, the type and number of columns are specified:

c indicates a column with centred text

r indicates a column with text aligned to the right side of the column

l indicates a column with text aligned to the left side of the column

p indicates a column with text aligned to the left side of the column, the column width may be specified in curly brackets

| draws a vertical line

```
\begin{tabular}{l|c|r|p{2cm}}
Col1 & Col2 & Col3 & Col 4\\
\hline
XXX & XXX & XXX & XXX\\
23 & 34 & 23 & 34 \\
\end{tabular}
```

Col1	Col2	Col3	Col 4
XXX	XXX	XXX	XXX
23	34	23	34

A horizontal line is inserted by the command `hline`, the content of each column in each line is separated from its neighbours with an &, a new line is introduced by double-backslashes.

2.5.2 Making your Table a `float` Object

Similar to the `figure` environment, it is possible to put `tabulars` into a `table` environment. Inside the `table` environment, the `caption` and `label` command may be used analogue to the `figure` environment.

2.6 Mathematical Equations

One of the main advantages of LaTeX is equation typesetting. But as for most of you are not mathematicians, this section is kept very short. Generally, equations need to be typesetted in an equation environment. The most simple one (which you actually might use a lot because it comes handy for other special characters within your text as well) are $...$. A bit more advanced is the `eqnarray` environment:

```
\begin{eqnarray}
 (a + b)^2 &=& a^2 + 2ab + b^2\\
 (a - b)^2 &=& a^2 - 2ab + b^2\\
           &=& (a + b)^2 - 4ab
\end{eqnarray}
```

$$(a+b)^2 = a^2 + 2ab + b^2 \tag{1}$$

$$(a-b)^2 = a^2 - 2ab + b^2 \tag{2}$$

$$= (a+b)^2 - 4ab \tag{3}$$

A simple chemical formula is typesetted like this:

```
This is some stuff of which
I am not even sure whether it
exists: H$_{3}$PO$_{4}^{-}$.
Anyway, it shows you how a
formula may reside within your
text.
```

This is some stuff of which I am not even sure whether it exists: $H_3PO_4^-$. Anyway, it shows you how a formula may reside within your text.

2.7 Bibliography

LaTeX contains a complete bibliography-system which is called BibTeX. BibTeX has many advantages, e.g. the automated update of your bibliography as soon as you use a new citation from your BibTeX database, automated setting of the correct citation numbers, changing the bibliography formatting with one command etc.

In order to use BibTeX in an efficient way (meaning as a database-system), you need to create a second file in the directory of your *.tex file. The filename needs to end on *.bib. In this file, you enter all the entries that you want to cite (and usually you do that once, keep the *.bib file and use it for your next document, again). The most important format for you is probably the one for citing an article:

```
@Article{key, %e.g. author:year
author = {},
title = {},
journal = {},
year = {},
OPTkey = {},%OPT entries are optional!
OPTvolume = {},
OPTnumber = {},
OPTpages = {},
OPTmonth = {},
OPTnote = {},
OPTannote = {}
}
```

The *.bib database has no other content than a list of many references. If you are planning on re-using a *.bib database, it makes a lot of sense to give intuitive keys, for example author:year. The key is what you use for citing that paper in your *.tex file.

Several good BibTeX citation managers are freely available. They offer a graphical interface to your database. It makes sense to use such a citation manager when your database gets huge or when you are sharing a reference database in your department. Until then, you might be fine with using intuitive keys ;-)

Apart from creating a BibTeX database, you need to include several commands in your LaTeX document. For author/year citation according to German DIN, you need to include the package natbib in the preamble of your document (you might also have to install natbib, first, it is available at http://www.ctan.org/tex-archive/macros/latex/contrib/natbib. An intuitive reference for natbib-usage is also given on http://merkel.zoneo.net/Latex/natbib.php).

Furthermore, the `bibliographystyle` needs to be specified in the preamble. `apalike` works fine with natbib.

At the position in your text where the complete bibliography shall be displayed, you need to include the command \bibliography{bib-file}.

References listed in the bib-database can be cited within the LaTeX document by using the \cite{key} command. *Usually.* In case of using the natbib package, however, you need to know at least two different citing commands: \citep{key}, which produces an author-year citation in brackets (normal case) and \citet{key}, which lists the author outside the brackets, e.g. if you want to say Mueller (1999) said that life is fun (`citet`). But this has been shown to be a wrong assumption (Frustrated PhD-student, 2007) (`citep`).

```
\documentclass[a4paper,10pt]{article}
\usepackage{natbib}
\bibliographystyle{apalike}

\begin{document}
Someone said something great \citep{Author:year}.
\citet{Author:year} said that....

\bibliography{name-of-your-bib-without-ending}
\end{document}
```

2.8 Special Characters

LaTeX commands themselves contain a lot of special characters which are meaningful for formatting text. So-called punctuation marks are *escaped* with \:

```
\&     \$     \%     \#     \_    \{     \}
```

Other characters, e.g. greek symbols, need to be put in between $...$ signs (see http://www.agu.org/meetings/mtabsLTX.htm for a concise summary of special characters and greek symbols). Two other important signs are:

```
Underscore: $\textunderscore$
Backslash: $\backslash$
```

Underscore: _
Backslash: \

While single quotes can just be typed as they are (’ results in ’), double quotes need a command:

```
'Simple quote'
''Cheated double quote''
```

’Simple quote’
”Cheated double quote”

The *German double quotes* only work if you include the package `ngerman` in the preamble of your document: `\glqq{}German double quotes\grqq{}`. This package also changes the headers of figures and tables to Tabelle and Abbildung.

Speaking about German... just in case you might need to use German or other *Umlaute*:

```
\"A \"a \"O \"o \"U \"u \ss{}
\AA{} \aa{} \O{} \o{} \AE{} \ae{}
\'E \'e \`E \`e \^A \^a \^O \^o
\^U \^u \^I \^i \^E \^e \c{c} \~n
```

Ä ä Ö ö Ü ü ß
Å å Ø ø Æ æ
É é È è Â â Ô ô
Û û Î î Ê ê ç ñ

2.9 Useful LaTeX Links

http://www.latex-project.org
http://texcatalogue.sarovar.org
http://www.tug.org/texshowcase
MacOS: http://latex.yauh.de/links_en.html

3 The beamer Class

3.1 Obtaining the beamer Class

The LaTeX BEAMER class is freely available for download at https://sourceforge.net/projects/latex-beamer. In addition, you will need the `pgf` and `xcolor` packages, which are available at the same website. After extracting the archives, you may either put the package folders into you current working directory, meaning into the folder where you keep the presentation file you are working on. This will enable you for using the BEAMER class when you are working in this specific directory. Alternatively, you may place package folders into your global tex-tree (on Linux mostly `/usr/share/texmf/tex/latex` and run `texhash` in your commandline.

3.2 Literature recommendation for beamer class

The file `beamer/doc/beameruserguide.pdf` in your BEAMER package is an excellent documentation.

3.3 Using the beamer class

3.3.1 The beamer Preamble

Several small changes are necessary to switch your article into a presentation file. An example document preamble with comments is given in figure 2 on page 11.

The important style-arguments for changing the looks of a presentation are `\usetheme` and `\usecolortheme`. A `theme` defines the general looks of your slides, for example whether you have 'progress bar' on the left, on the right, at the top, at the bottom or not at all. `themes` come along with standard colors. In some cases, you can change the colors by changing to another `colortheme`. Table 1 gives an overview about `themes` and `colorthemes` that are contained in the current version of BEAMER.

themes	**colorthemes**
AnnArbor	albatross
Antibes	beaver
Bergen	beetle
Berkeley	crane
Berlin	default
Boadilla	dolphin
CambridgeUS	dove
Copenhagen	fly
Darmstadt	lily
Dresden	orchid
Frankfurt	rose
Goettingen	seagull
Hannover	seahorse
Ilmenau	whale
JuanLesPins	wolverine
Luebeck	
Madrid	
Malmoe	
Marburg	
Montpelier	
PaloAlto	
Pittsburgh	
Rochester	
Singapore	
Szeged	
Warsaw	

Table 1: `themes` and `colorthemes` contained in BEAMER version 3.06. (`themes` and `colorthemes` are not related to each other by the arrangements in this table.)

3.3.2 Structuring a Presentation

The BEAMER class supports the same LaTeX structuring commands as the `article` class (see page 4). A first step in planning your presentation (after fixing the preamble) should be typesetting a structure (example see figure

```
\documentclass{beamer}

\mode<presentation>
{
  \usetheme{Warsaw} % compulsory argument
  \usecolortheme{Hannover} % optional argument
}

\usepackage[latin1]{inputenc} % use these packages if you are working on
\usepackage[german]{babel}    % Windows and need German ’Umlauts’

\title[Short version] % (Short version optional)
{Presentation title}

\author[Author, Anders] % (optional, only in case of several authors)
{F.~Author\inst{1} \and S.~Anders\inst{2}}

\institute{
  \inst{1}%
  Institute for Microbiology\\
  University of Somewhere
  \and
  \inst{2}%
  Institute for Philosophy\\
  University of Somewhere Else}

\date[KPT 2003] % (optional, conference name shortcut)
{Conference for Something, 2010}

% In case you want to use a logo (e.g. in png format):
\pgfdeclareimage[height=0.5cm]{university-logo}{university-logo-filename}
\logo{\pgfuseimage{university-logo}}

% If you want to display your table of contents at every subsection:
\AtBeginSubsection[]
{
  \begin{frame}<beamer>{Table of Contents}
    \tableofcontents[currentsection,currentsubsection]
  \end{frame}
}

\begin{document}

% All your slides will be inserted HERE!

\end{document}
```

Figure 2: Example for a LaTeX-BEAMER presentation preamble.

3 on page 12). This structure will be displayed on slides where you use the `\tableofcontents` command and depended on the used theme, they will be displayed somewhere on all slides in a ’progress bar’. The headlines given in

sectioning commands will *not* serve as headers for your slides! Take care to keep the Headers short - because space in the progress bars is generally limited.

```
\section{Introduction}
  \subsection{Some Metabolic Process}
  \subsection{Your Favourite Protein}
\section{Experiments}
  \subsection{Methods}
  \subsection{Experimental Setup}
  \subsection{Results}
\section{Discussion}
```

Figure 3: Example for a presentation structure.

3.3.3 Making Slides

Slides reside in the `frame` environment. Similar to the `figure` or `table` environment, every slide is enclosed by \\`begin`{`frame`} and \\`end`{`frame`}. You have to place your slide in the appropriate section/subsection of your presentation. The command \\`frametitle`{ } specifies the actual header of your slide - which might be identical to the section or subsection title. It is also possible to give subtitles (\\`frametitle`{ }). Example:

```

\section{Introduction}
  \begin{frame}
   \frametitle{Introduction or Another Title}

  \end{frame}
```

Within the `frame` environment, you may use most existing LaTeX features, e.g. figure insertion, tabulars, item lists etc.

3.3.4 Blocks & Columns

The `blocks` and `columns` environments are useful commands for arranging things beautifully on your slide, e.g. a table in a `block` with a header next to a list. `Blocks` and `columns` can be combined: You may make single `columns` a `block` or you may define several `columns` within a `block`. Usage:

```

\begin{columns}
 \begin{column}{5cm} % width = 5cm
  % insert content of first column
 \end{column}
 \begin{column}{5cm} % width = 5cm
  % insert content of second column
 \end{column}
\end{columns}
```

```
\begin{block}{Blocktitle}
 %some content, e.g. a figure or a list
\end{block}
```

3.3.5 Step-by-step Uncovering and Overlays

Sometimes it is useful to *uncover* things *step-by-step.* It is easiest and most straight-forward to uncover a list by simply adding an *uncovering-tag* to the `\item` command: `\item<1>` will be visible with the first click on your slide, `\item<2>` will uncover on the second click and so on. Be aware of the following: An item will only be visible during the *click-number* specified in the brackets. If you would like to have the first item still visible while the second item is already uncovered, you would need to specify the uncovering mode for the first item like this: `\item<1,2>`. Alternatively to the numbering system, you may insert a `\pause`-command between the different items (this will not result in 'disappearing' items).

The `onlyenv` environment is useful for uncovering entire blocks, graphics, tables and other 'big' things. Usage:

```
\begin{onlyenv}<1>
 % Something will be visible after the first click
\end{onlyenv}
\begin{onlyenv}<2>
 % Something else will appear after the second click
\end{onlyenv}
```

Sometimes, it is tricky to create correct overlays, esp. if you try to first show a table with certain dimensions in a block and then replace the table by a figure with slightly different dimensions. In that case, the block might change its length, which will be visible to the audience. A lazy 'work-around' is the usage of `\vspace{0.1cm}`, which introduces a vertical space of a defined length.

3.3.6 Creating Handouts

Handouts from your BEAMER presentation file may be created in various ways. Fast and easy is using the `handout` option:

```
\documentclass[handout]{beamer}
```

You may also insert the following two lines into your preamble, in order to print several slides on one page of your handout:

```
\usepackage{pgfpages}
\pgfpagesuselayout{2 on 1}[a4paper, border shrink=5mm]
```

A different approach is using the `article` class and including the package `beamerarticle`:

```
\documentclass[a4paper]{article}
\usepackage${beamerarticle}
```

The addition of two `modes` allows to mark certain parts of your document for inclusion only in the article or presentation output (used in ¡¿ brackets).

```
\mode<article>{\usepackage{fullpage}}
\mode<presentation>{\usetheme{Goettingen}}
```

References

Knuth, D. E. (1979). *Tex and Metafont - New Directions in Typesetting.* Number ISBN 0-932376-02-9. Digital Press, 12 Crosby Drive, Bedford, MA 01730, USA.

Lamport, L. (1999). *A Document Preparation System. Users's Guide and Reference Manual.* Number ISBN 0-201-52983-1. Addison-Wesley Longman, Amsterdam.