

Einführungskurs in L^AT_EX

Andreas Lochmann

29. Juni 2019

Inhaltsverzeichnis

1	Grundlagen	3
1.1	Philosophie	3
1.2	Syntax	3
1.3	Gliederung eines Dokuments	4
1.4	Umlaute und Anführungsstriche	5
1.5	<code>enumerate</code> , <code>itemize</code> und Quellcode	5
1.6	Übungen	6
2	Mathematischer Formelsatz	7
2.1	Math-Mode	7
2.2	Mehr Math-Mode	8
2.3	<code>array</code> und <code>tabular</code>	9
2.4	Theoreme	10
2.5	Verlinkungen	11
2.6	Übungen	11
3	Grafiken	12
3.1	<code>includegraphics</code>	12
3.2	<code>figure</code> und <code>table</code>	13
3.3	Tabellenkalkulationen	14
3.4	Inkscape	14
3.5	Tikz	14
3.6	Übungen	15
4	L^AT_EX anpassen	16
4.1	<code>DeclareMathOperator</code> und <code>newcommand</code>	16
4.2	<code>newtheorem</code> und <code>newenvironment</code>	17
4.3	Counter	18
4.4	Das Seitenlayout	20
4.5	Zeilenabstände und -einzüge	21

4.6	include, input, Style-Files	21
4.7	Übungen	22
5	Briefe	23
5.1	Umbrüche	23
5.2	Typografische Details	23
5.3	Übungen	24
6	Beamer-\TeX	24
6.1	Unterschied zwischen Präsentationen und Artikeln	24
6.2	Die Präambel	24
6.3	Die Titelfolie	26
6.4	Der Inhalt	26
6.5	Anzeigen	27
6.6	Übungen	27
A	Verwendung amerikanischer Tastaturen	29
B	Leerzeichen	29
C	Grafiken aus Maple einbinden	30
D	Alte Befehle zum Seitenlayout	31
E	Auflösen rekursiver Makrodefinitionen	32
F	Bib\TeX	32
G	Index	33

1 Grundlagen

1.1 Philosophie

- \LaTeX ist keine Textverarbeitung, sondern ein Textsatzsystem, d.h. es nimmt eine bestehende Datei (Quelldatei) und wandelt sie gemäß bestimmten Vorgaben um in ein druckfähiges Format (Ausgabeformat). Dieses kann sein: DVI, PS, PDF oder (selten) HTML.
- Die Vorgaben sind teils in der Quelldatei in Form von Befehlen vorgegeben, teils im Programm selbst implementiert. Beispielsweise trennt \LaTeX automatisch Wörter, die über das Zeilenende hinausgehen, gemäß internen Regeln. Diese können umgangen werden, indem innerhalb eines Wortes der Befehl $\backslash-$ eingefügt wird.
- Insbesondere ist \LaTeX also kein Texteditor. Unix-Philosophie: Lieber viele kleine Werkzeuge, die sich kombinieren lassen, als wenige große Werkzeuge. Daher wird zur Erstellung der Quelldatei ein weiteres Programm (der Editor) benötigt, den man nach eigenem Geschmack frei wählen kann.
- Inzwischen gibt es aber Editoren, die \LaTeX integriert haben, auf den Windows-Rechnern in den Computer-Pools sind zwei zur Auswahl installiert: TeXnic-Center (umfangreich) und TeXworks (minimalistisch). Diese werden wir nutzen.
- In den Quelldateien wird normalerweise nicht gesagt, wo sich Zeilenumbrüche befinden, wie lang eine Seite ist etc. Dies wird alles von \LaTeX entschieden, das immerhin speziell für diesen Zweck entwickelt wurde. Ebenso die Seitenzahlen, Abschnittsnummern, Querbezüge, Inhaltsverzeichnis, Nummerierung der Fußnoten. Auch welche Textteile z.B. fett gedruckt werden sollen bestimmt zumeist \LaTeX ; der Nutzer gibt lediglich an, welche Funktion der jeweilige Text erfüllt (z.B. Abschnittsüberschrift). Diese Einstellungen lassen sich aber bei Bedarf alle neu setzen, wenn auch evtl. mit viel Aufwand.

1.2 Syntax

- \LaTeX basiert auf \TeX , und erweitert diese um viele bequeme Funktionen. Nachteilig wirkt sich dabei aus, dass durch die Erweiterungen jetzt mehrere Syntaxen nebeneinander existieren.
- Befehle starten mit einem Backslash “ \backslash ”. Manche Befehle nehmen Optionen entgegen, die dann in geschweifte Klammern gesetzt werden. Optionale Argumente stehen in eckigen Klammern. Beispiele:

```
 $\rightarrow$ ,  $\frac{3}{5}$ ,  $\sqrt[3]{x}$ 
```

- Die Quelldatei wird wie ein Computerprogramm von oben nach unten und von links nach rechts nach links gelesen und Befehle in dieser Reihenfolge abgearbeitet. Alles, was kein Befehl (oder Option zu einem Befehl) ist, wird als regulärer Text interpretiert und erscheint entsprechend in der Ausgabedatei.
- Manche Befehle funktionieren wie Schalter, deren Wirkung auf den jeweiligen Block begrenzt ist. Ein Block ist dabei ein Teil des Textes, der von geschweiften Klammern umgeben ist. Beispiel:

```
{\em Dieser Text ist hervorzuheben.}
```

- Eine dritte Variante sind Umgebungen, die mit `\begin{...}` eingeleitet und mit `\end{...}` geschlossen werden. Beispiel:

```
\begin{center} Dieser Text ist zentriert. \end{center}
```

- Alte Befehle, die noch aus dem zugrundeliegenden T_EX-System stammen, nehmen ihre Optionen ohne geschweifte Klammern entgegen. Beispiel:

```
\topmargin4mm
```

- Kommentare werden mit `%` eingeleitet und gelten eine Zeile lang.

1.3 Gliederung eines Dokuments

Eine L^AT_EX-Quelldatei beginnt immer mit dem Befehl `\documentclass`, also beispielsweise

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

Dies definiert einige globale Optionen. Am wichtigsten ist dabei das letzte Argument, der Stil des Dokuments: `article`. (Alternativen sind z.B.: `book`, `letter`, `slides`, `report`, ...) Dazu wird eingestellt: Schriftgröße 12pt, einseitig zu formatieren (Seitenzahlen immer an derselben Stelle!). Weitere mögliche Optionen sind z.B. `a4paper`, `german`, `ngerman`.

Es folgt eine Reihe von `\usepackage`-Befehlen, die zusätzliche Module laden, um beispielsweise Grafiken einzubinden oder zusätzliche Symbole zur Verfügung zu stellen.

Das Dokument an sich beginnt mit

```
\begin{document}
```

Daran schließen sich die Definition des Titels, Autors etc. an, sowie ein Befehl, um den Titel zu erzeugen:

```
\title{Untergruppenverbände und ihre Anwendungen}
\author{Manuel Mustermann}
\maketitle
```

Die weitere Gliederung des Dokuments erfolgt mit den Befehlen `\abstract`, `\section`, `\subsection`, `\subsubsection` und der `appendix`-Umgebung, im Fall von Büchern zusätzlich `\part` und `\chapter`.

Es schließt sich ein Literaturverzeichnis an:

```
\begin{thebibliography}{9999}
\bibitem[Bi1]{Birkhoff}G. Birkhoff, {\it Lattice Theory}, 3rd ed. (1960)
\bibitem[dH]{dlHarpe}P. de la Harpe, {\it Geometric Group Theory}
\end{thebibliography}
```

(“9999” ist ein sogenanntes Tiefenmuster.) Das Dokument endet mit der Zeile:

```
\end{document}
```

Mittels `\cite` kann man dann auf die Einträge im Literaturverzeichnis zurückgreifen, im Beispiel also `\cite{Birkhoff}` oder `\cite{dlHarpe}`.

1.4 Umlaute und Anführungsstriche

Die Kodierung normaler Buchstaben und Satzzeichen ist durch ASCII genormt und daher im normalen Gebrauch kein Problem. Dagegen gibt es mehrere Möglichkeiten, Umlaute durch Bytefolgen zu kodieren. Aktuell sind zwei wichtig (betriebssystemunabhängig!):

1. Isolatin-1 (ISO 8859-1) und dessen Varianten Isolatin-15 und Windows-1252
2. Unicode (üblicherweise in der Version UTF-8)

Der jeweilige Zeichensatz wird ausgewählt durch das Einbinden eines passenden Pakets. Diese sind in obiger Reihenfolge:

1. `\usepackage[latin1]{inputenc}` oder `\usepackage{isolatin1}`
2. `\usepackage[utf8]{inputenc}` oder in besonderen Fällen
`\usepackage[utf8x]{inputenc}`, ggbf. `\usepackage{ucs}`

Unter bestimmten Umständen muss man sich auf ASCII verlassen. In diesen Fällen lassen sich die Umlaute und das “ß” wie folgt schreiben:

```
\"a, \"o, \"u, \"A, \"O, \"U, \ss{}
```

(Man beachte die Klammern in `\ss{}`.)

1.5 enumerate, itemize und Quellcode

Im Text lassen sich Listen mit Hilfe der Umgebungen `enumerate` (durchnummeriert) und `itemize` (ohne Nummerierung) erstellen:

```

\begin{enumerate}
\item
Erster Punkt.
\item
Außerdem.
\end{enumerate}

```

1. Erster Punkt.
2. Außerdem.

Um Quellcode auf bequeme Weise einzubinden, kann man das Paket `listings` einbinden. Der Quellcode wird dann in einer `lstlisting`-Umgebung eingebracht:

```

\usepackage{listings}
...
\lstset{language=Python,numbers=left,numberstyle=\footnotesize}
\begin{lstlisting}
import math

x = math.sqrt(2)
for j in range(50):
    x = x*x/math.sqrt(2)
    print(x)
\end{lstlisting}

```

```

1 import math
2
3 x = math.sqrt(2)
4 for j in range(50):
5     x = x*x/math.sqrt(2)
6     print(x)

```

Quellcode kann auch von der Quelldatei selber aus eingelesen werden, dazu kann man `\lstinputlisting{dateiname.c}` verwenden.

1.6 Übungen

1. Machen Sie sich mit TeXnicCenter, Texmaker und/oder TeXworks vertraut und erzeugen Sie ein erstes Dokument mit Titel, Autor, Abstract, mehreren Abschnitten, etwas Text und einer Fußnote (`\footnote`).
2. Prüfen Sie, welches Paket Sie einbinden müssen, um Umlaute richtig darzustellen, und was passiert, wenn Sie das falsche Paket verwenden.

3. Erstellen Sie eine Liste mit mindestens fünf Unterpunkten. Bauen Sie in den dritten Unterpunkt eine weitere Liste ein und untersuchen Sie das Verhältnis von „Oberliste“ zu „Unterliste“ in Hinblick auf Nummerierung und Zeileneinzug.
4. Was geschieht, wenn Sie in einer `enumerate-` bzw. `itemize-` Umgebung den Befehl `\item[g]` anstelle von `\item` verwenden?
5. Fügen Sie ein Literaturverzeichnis mit zwei Einträgen hinzu und referenzieren Sie auf diese Einträge in Ihrem Text (`\cite`).

2 Mathematischer Formelsatz

2.1 Math-Mode

Mathematische Formeln müssen angekündigt werden, damit \LaTeX sie richtig setzen kann. Formeln, die in einer normalen Textzeile stehen sollen, werden mit $\$$ -Zeichen umringt, z.B.:

Aus $x=2x+4$ folgt sofort $x=-\sqrt{16}$.

Sollen die Formeln in einer einzelnen Zeile separat stehen, wird $\$$ verwendet. Dies empfiehlt sich z.B. für besonders hohe oder lange Formeln.

Sollen mehrere Gleichungen untereinander stehen, so dass die Gleichheitszeichen direkt untereinander sind, verwendet man die `eqnarray`-Umgebung und setzt die Gleichheitszeichen zwischen `&`-Zeichen. Die Zeilen werden mittels `\` voneinander getrennt:

```
\begin{eqnarray}
x &=& 2x + 4 \\
&<& x + 4
\end{eqnarray}
```

$$x = 2x + 4 \tag{1}$$

$$< x + 4 \tag{2}$$

Sollen die Zeilen nicht durchnummeriert werden, verwendet man stattdessen `eqnarray*`. Es gibt weitere vergleichbare Umgebungen, z.B. die `align`-Umgebung, die jeweils etwas anders funktionieren und andere Abstände erzeugen. Von der AMS wird die `align`-Umgebung empfohlen. Für die Erzeugung von mehr als drei Spalten ist `align` besonders praktisch:

```
\begin{align}
x_1 &= 2a + 4 & y_1 &= 3a + 3 \\
x_2 &= 3a + 4 & y_2 &= 2a + 3
\end{align}
```

`\end{align}`

$$x_1 = 2a + 4 \qquad y_1 = 3a + 3 \qquad (3)$$

$$x_2 = 3a + 4 \qquad y_2 = 2a + 3 \qquad (4)$$

Probleme hat `align`, wenn man Gleichheitszeichen nutzen will, über die Text geschrieben wird:

```
\begin{eqnarray*}
x &=& 2x+3 \\
y &\stackrel{\text{analog}}{=}& 2y+3
\end{eqnarray*}
\begin{align*}
x &= 2x+3 \\
y &\stackrel{\text{analog}}{=} 2y+3
\end{align*}
```

$$x = 2x + 3$$

$$y \stackrel{\text{analog}}{=} 2y + 3$$

$$x = 2x + 3$$

$$y \stackrel{\text{analog}}{=} 2y + 3$$

Hintergrund: `eqnarray` erzeugt genau drei Spalten mit den Bündigkeiten rechts, zentriert, links. `align` dagegen erzeugt beliebig viele Spalten mit den Bündigkeiten rechts, links, rechts, links, ...

Die verschiedenen Symbole findet man im Netz z.B. in Form der Datei `symbols.pdf` (man suche im Internet nach `LaTeX symbols.pdf`). Auch die Seite

<http://detexify.kirelabs.org/classify.html>

kann praktisch sein, sie versucht per Hand gezeichnete Symbole zu identifizieren.

2.2 Mehr Math-Mode

- `\frac`, `\sqrt`, `\int`, `\sum`, `^`, `_`
- Klammern: `\{`, `\}`, `\left`, `\right`

$$\backslash big(, \backslash Big(, \backslash bigg(, \backslash Bigg(\Rightarrow (, (, (, ($$

- Griechische Buchstaben, `\mu`, `\nu`
- Pfeile: `\leftarrow`, `\rightarrow`, `\Leftarrow`, `\Rightarrow`, `\longleftarrow`, `\longrightarrow`, `\Leftrightarrow`, `\Longleftarrow`, `\Longrightarrow`, `\leftrightharpoonup`, `\Leftrightharpoonup`, `\longleftarrow`, `\longrightarrow`, `\uparrow`, `\downarrow`, `\hookrightarrow`, `\twoheadrightarrow`, `\mapsto`

$\leftarrow, \rightarrow, \Leftrightarrow, \Rightarrow, \longleftarrow, \longrightarrow, \uparrow, \downarrow, \hookrightarrow, \twoheadrightarrow, \mapsto$

- `\ldots`, `\cdots`
- Akzente, `\imath`, `\jmath`, `\overline`, `\underbrace`, `\stackrel`
- `\exists`, `\forall`, `\cup`, `\cap`, `\vee`, `\wedge`, `\subset`, `\supset`, `\leq`, `\geq`, `\neq`, `\times`, `\otimes`, `\oplus`
- `\mathcal`, `\mathscr`, `\mathfrak`, `\mathbb`
- Abstände mittels `\,`, `\;`, `\quad`, `\qquad`, `\!`. Auch interessant: `\hfill`.
- Pakete: `amsmath` (Befehle), `amssymb` (Symbole- und Schriften), `amsthm` (Umgebungen), `amsfonts`, `euscript` mit Option `[mathscr]`, `stmaryrd`.

2.3 array und tabular

Die `array`- und `tabular`-Umgebungen erlauben z.B. Tabellen, Matrizen und Fallunterscheidungen, falls die `eqnarray`-Umgebung nicht ausreicht. Dabei ist die `array`-Umgebung speziell für den Math-Mode gedacht, während die sehr ähnliche `tabular`-Umgebung im Text-Modus eingesetzt wird. Beispiele:

```
$$ \left(\begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array}\right) $$
```

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

```
$$ \begin{array}{rcl} & x & = & 2x + 4 \\ & \Rightarrow x & = & -4 \\ & \Rightarrow x^2 & = & 16 \end{array} $$
```

$$\begin{aligned} & x = 2x + 4 \\ \Rightarrow & x = -4 \\ \Rightarrow & x^2 = 16 \end{aligned}$$

```

\begin{tabular}{r|c|l} \hline
Kuchen & 3 & Äpfel \\
& 1 kg & Mehl \\ \hline
Cola & 1 & Flasche \\ \hline
\end{tabular}

```

Kuchen	3	Äpfel
	1 kg	Mehl
Cola	1	Flasche

In Arrays und Tabellen lässt sich die Bündigkeit einstellen, indem man `r` verwendet für rechtsbündig, `l` für linksbündig und `c` für zentriert.

Will man im Text selbst die Bündigkeit verändern, benutzt man die Umgebungen `center`, `flushleft`, `flushright`.

2.4 Theoreme

Mit der Einbindung des `amsthm`-Pakets steht der Befehl `\newtheorem` für die Präambel zur Verfügung. Mit ihm lassen sich bequem sogenannte *Theorem-Umgebungen* definieren:

```

\newtheorem{Lemma}{Lemma}
\newtheorem{Beispiel}[Lemma]{Beispiel}
\newtheorem{Korollar}[Lemma]{Korollar}
\newtheorem{Satz}{Theorem}
\newtheorem{Hauptsatz}[Satz]{Hauptsatz}
\newtheorem{Exa}{Example}

```

Das optionale Argument gibt dabei an, wie gezählt werden soll: In diesem Fall werden Beispiele und Korollare gemeinsam mit Lemmata gezählt, während Sätze und Hauptsätze einer separaten Zählung unterliegen.

Will man bei der Zählung beispielsweise die Kapitel- und Abschnittszahl hinzufügen, so dass nach jedem Abschnitt der Zähler zurückgesetzt wird, verwendet man

```

\newtheorem{Lemma}{Lemma}[section]

```

In obigem Beispiel überträgt sich diese Zählweise auf Beispiele und Korollare, aber nicht auf Sätze, Hauptsätze und Beispiele.

Text in Theorem-Umgebungen ist üblicherweise kursiv gesetzt. Um nun innerhalb des Math-Modus einen normalen Text zu schreiben, verwendet man `\textnormal` und `\text`, die sich geringfügig unterscheiden (`\text` ist AMS-TEX):

```

\begin{Lemma}
Hier gilt:
 $2=0$  \; \textnormal{(da Charakteristik 2)} \; \;

```

```
\text{(nach Voraussetzung)} $$
\end{Lemma}
```

Lemma 1. *Hier gilt:*

$$2 = 0 \text{ (da Charakteristik 2) (nach Voraussetzung)}$$

2.5 Verlinkungen

Man kann mit `\label` einzelnen Elementen des Dokuments einen Namen zuordnen. Dieser wird nicht im Dokument angezeigt, aber mittels `\ref` lässt sich auf diesen Punkt zurückverweisen. L^AT_EX fügt dann die jeweilige Nummer ein, Beispielsweise des Satzes:

```
\begin{Theorem} \label{Satz_von_Schmitz}
```

Hier steht der Satz.

```
\end{Theorem}
```

Nach Satz `\ref{Satz_von_Schmitz}` auf Seite `\pageref{Satz_von_Schmitz}` gilt ...

Je nach Einstellung kann man damit (und mit `\href` für Internetadressen) funktionsfähige Verlinkungen in einem pdf-Output erzeugen; ggbf. muss erst noch das Paket `hyperref` eingebunden werden.

Ein Inhaltsverzeichnis wird mit dem Befehl `\tableofcontents`, Tabellen- und Bildverzeichnisse mit `\listoftables` bzw. `\listoffigures` automatisch erstellt.

Hinweis: Für sämtliche Verlinkungen, Indexe und Verzeichnisse muss das Dokument *zweimal* kompiliert werden, in besonders komplizierten Fällen sogar noch öfter!

2.6 Übungen

1. Setzen Sie folgende Formeln mit bzw. ohne Nummerierung um:

$$C_a := \begin{cases} \begin{pmatrix} 1 & a \\ -1 & 1 \end{pmatrix} & \text{falls } a \neq 0 \\ 0 & \text{sonst} \end{cases}$$

$$\sin \gamma := \sum_{n=0}^{\infty} \frac{(-1)^n \cdot \gamma^{2n+1}}{(2n+1)!} \quad (5)$$

$$\int_0^1 \frac{1}{1+x^2} \cdot \underbrace{(x^4 \cdot (1-x)^4)}_{x^4 - 4x^5 + 6x^6 - 4x^7 + x^8} dx = \frac{22}{7} - \pi \quad (6)$$

2. Schreiben Sie eine Wahrheitstabelle für den Ausdruck

$$(a \vee b) \wedge (\neg a \vee \neg b).$$

3. Variablen werden grundsätzlich kursiv geschrieben, Funktionen dagegen aufrecht. Machen Sie sich den Unterschied zwischen `\sin` und `sin` sowie zwischen `\int` und `\int\limits` klar.
4. Setzen Sie das Integral (6) in eine eigene `theorem`-Umgebung, schreiben Sie einen „Beweis“ darunter, in dem Sie auf die Formelnummer aus der Definition für $\sin \gamma$ verweisen. Das Wort „Beweis:“ sollte fett gedruckt sein. Schließen Sie den Beweis ab mit einem rechtsbündigen „□“ (`\hfill`).
5. Fügen Sie eine weitere nummerierte Formel *vor* dem Satz aus Teil (5) ein, und beobachten Sie, wie sich nach zweifachem Kompilieren die Referenz auf C_a im Beweis verändert.
6. Benutzen Sie die `array`-Umgebung und Pfeile, um folgendes Diagramm zu realisieren:

$$\begin{array}{ccccc} & V \otimes V & \xrightarrow{\mu} & V & \\ \text{id} \otimes e & \uparrow & \# & \uparrow & \text{id} \\ & V & \xrightarrow{\cong} & V & \end{array}$$

3 Grafiken

3.1 includegraphics

Grafiken lassen sich einbinden, dazu benötigt man aber ein zusätzliches Paket, z.B. `graphicx`. Dokumentation:

<http://www.ctan.org/tex-archive/macros/latex/required/graphics/grfguide.ps>

Ein Bild wird an der entsprechenden Stelle im Fließtext eingebunden mit

```
\includegraphics[width=7cm]{dateiname_mit.endung}
```

(wobei das `width`-Argument optional ist).

Diejenigen Formate, die man einbinden kann, hängen leider vom Zielformat ab: Benutzt man den Befehl `latex`, so erzeugt man `dvi`-Dateien. In `dvi`-Dateien kann man Encapsulated Postscript (`eps`) einbinden, aber keine anderen Formate (außer durch Verwendung von `dvipdfm`). Aus `dvi` lassen sich dann `ps` und `pdf` durch Konvertierungsprogramme gewinnen.

Verwendet man dagegen `pdflatex`, um direkt eine `pdf`-Datei zu erstellen, kann man zwar `jpeg`, `png` und `pdf`-Dateien einbinden, nicht aber `eps`!

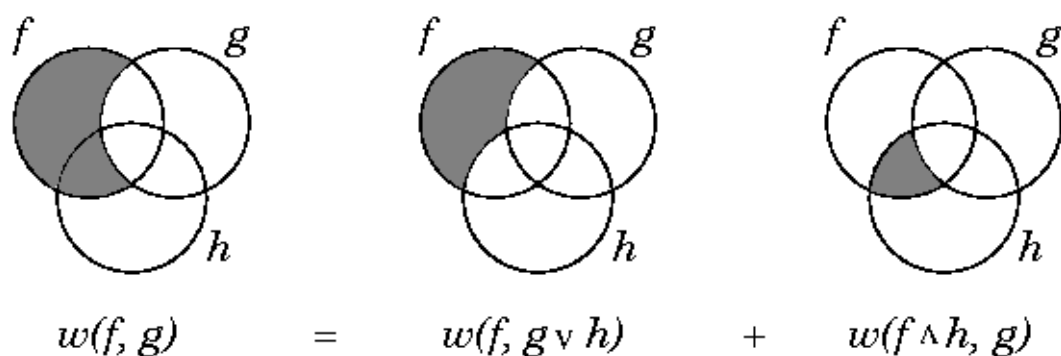


Abbildung 1: A simple rule of difference valuations.

3.2 figure und table

Bilder und größere Tabellen werden üblicherweise nicht im Fließtext untergebracht, sondern davon getrennt, ganz oben oder ganz unten auf einer Seite angezeigt, mit einem Untertitel versehen und nummeriert, so dass man vom Fließtext aus auf das Bild oder die Tabelle referenzieren kann. Dies alles geschieht durch die `figure`- und `table`-Umgebungen, siehe Abbildungen 1 und 2.

```
\begin{figure}[t]
\centering
\includegraphics{difference_valuation.png}
\caption{A simple rule of difference valuations.}
\label{FIG_difference_valuation}
\end{figure}

\begin{figure}[b]
\centering
\includegraphics[width=12cm]{funktion_aufgabe4.png}
\caption{A function.}
\label{FIG_function}
```

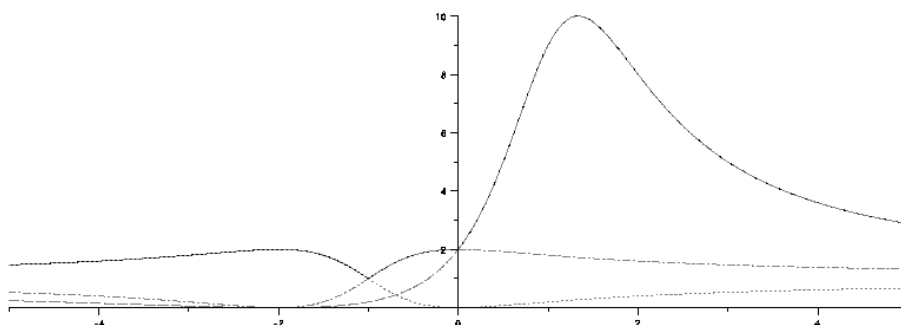


Abbildung 2: A function.

`\end{figure}`

Das optionale Argument gibt an, wo das Bild am besten positioniert werden sollte: **t** für top, **b** für bottom, **p** für eine eigene Seite, **h** für die Stelle, an der das Bild definiert wird. Wie und auf welcher Seite das Bild dann aber tatsächlich positioniert wird, entscheidet L^AT_EX anhand von Richtwerten zur Ästhetik und zum Lesefluss.

Die `table`-Umgebung funktioniert analog zur `figure`-Umgebung, mit dem Unterschied, dass unter dem Bild nicht „Abbildung“ steht, sondern „Tabelle“, und die Nummerierung von der Abbildungs-Nummerierung unabhängig ist.

Mit `\listoffigures` und `\listoftables` werden Bild- und Tabellenverzeichnisse angelegt, wie das folgende:

Abbildungsverzeichnis

1	A simple rule of difference valuations.	13
2	A function.	13

3.3 Tabellenkalkulationen

Diagramme lassen sich sehr einfach aus LibreOffice Calc exportieren, dazu muss man nur auf das (defokussierte!) Diagramm rechts-klicken und „Als Grafik exportieren...“ wählen. Ein Export als `png` wie auch als `eps` ist möglich.

Microsoft Excel fehlt aktuell die Exportmöglichkeit für Bilder. Stattdessen kann man entweder ein Screenshot aufnehmen oder das Diagramm kopieren, in Microsoft Word als Bild einfügen (nicht als Diagramm einfügen!) und anschließend mittels Rechts-Klick als Bild exportieren.

3.4 Inkscape

Inkscape ist ein Programm, um sogenannte Vektorgrafiken zu erzeugen (z.B. `svg`-Format). Vektorgrafiken sind frei skalierbar und z.B. geeignet für die meisten Diagramme. Man verwendet Inkscape wie normale Zeichenprogramme. Unter `Speichern unter...` kann man anschließend die Grafik als `eps` oder `pdf` abspeichern oder als `png` exportieren und per `\includegraphics` einbinden.

3.5 Tikz

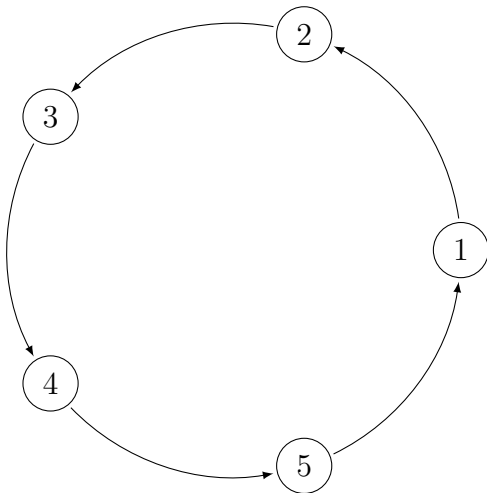
`tikz` ist ein kraftvolles Paket zur Erzeugung vieler verschiedener Diagramme. Wir wollen es uns anhand eines einfachen Diagramms anschauen. Der folgende Code

stammt von Jerome Tremblay:

```
% A simple cycle
% Author : Jerome Tremblay
\documentclass{article}
\usepackage{tikz}
\begin{document}
\begin{tikzpicture}

\def \n {5}
\def \radius {3cm}
\def \margin {8} % margin in angles, depends on the radius

\foreach \s in {1,...,\n}
{
  \node[draw, circle] at ({360/\n * (\s - 1)}:\radius) {$\s$};
  \draw[->, >=latex] ({360/\n * (\s - 1)+\margin}:\radius)
    arc ({360/\n * (\s - 1)+\margin}:{360/\n * (\s)-\margin}:\radius);
}
\end{tikzpicture}
\end{document}
```



3.6 Übungen

1. Betten Sie die folgenden Dateien in Ihr Dokument ein:

<http://upload.wikimedia.org/wikipedia/commons/2/29/Algebra-Unit.png>

<http://www.mathematik.uni-marburg.de/~lochmann/linalg1/linalg1.jpg>

http://www.mathematik.uni-marburg.de/~lochmann/difference_valuation.eps

2. Erstellen Sie mit Inkscape oder einem anderen Programm ein einfaches Diagramm, um zu verdeutlichen, was eine Schnittmenge ist, und betten Sie es in Ihr Dokument ein.
3. Erzeugen Sie mit Hilfe eines Tabellenkalkulationsprogramms ein Diagramm (sinnvoll sind hierfür Säulen-, Balken- und Kreisdiagramme) zu folgender Tabelle und binden Sie es in Ihr Dokument ein:

2. Quartal 2015	30.714 €
3. Quartal 2015	12.210 €
4. Quartal 2015	42.989 €
1. Quartal 2016	35.651 €
2. Quartal 2016	8.276 €

4. Erstellen Sie mit Hilfe eines Tabellenkalkulationsprogramms den Graphen der Sinus-Funktion über deren erste Periode $[0, 2\pi]$ und binden Sie ihn in Ihr Dokument ein.

Die folgenden Übungen setzen voraus, dass Sie Maple installiert haben:

5. Erstellen Sie mit Maple einen 2D-Plot des Realteils der Riemannschen ζ -Funktion (in Maple: `Zeta(z)`) für $z = \frac{1}{2} + ix$. Wählen Sie dabei das Intervall für x so, dass die Irregularität von ζ gut sichtbar wird. (Imaginäre Einheit: `I`, Realteil: `Re(z)`)

Fügen Sie als zweite Funktion den Imaginärteil hinzu, färben Sie den Graphen des Realteils schwarz und den des Imaginärteils grau.

Exportieren Sie die Grafik und binden Sie sie in Ihr Dokument ein.

6. Erstellen Sie mit Maple einen 3D-Plot der Funktion $\sin(xy)$ für den Bereich $x, y \in [-3, 3]$ (mit angepassten Optionen!), und binden Sie ihn in Ihr Dokument ein.

4 L^AT_EX anpassen

4.1 DeclareMathOperator und newcommand

Auch wenn schon viele mathematische Funktionen als L^AT_EX-Befehle vordefiniert sind, passiert es immer wieder, dass man eine neue definieren muss. Dies geschieht mittels `\DeclareMathOperator`, z.B.

```
\DeclareMathOperator{\rk}{rank}
```

Dieser Befehl benötigt die AMS-Pakete, siehe auch

<ftp://ftp.ams.org/ams/doc/amsmath/short-math-guide.pdf>.

Etwas allgemeiner sind dagegen `\newcommand` und `\renewcommand`, ein paar Beispiele:

```
\newcommand{\beq}{\begin{eqnarray*}}
\newcommand{\eeq}{\end{eqnarray*}}
\newcommand{\Z}{\mathbb{Z}}
\newcommand{\fg}[1]{\Z/{#1}\Z}
\renewcommand{\setminus}{\smallsetminus}
\renewcommand{\d}{\textnormal{d}}

\newcommand{\Loesung}[2]{ \ \ [2mm]
{\bf Behauptung:} {#1} \ \ [2mm]
{\bf Beweis:} {#2} \hfill $\square$ \bigskip}
```

Das optionale Argument bei `\[re]newcommand` gibt die Zahl der Argumente an.

`\[re]newcommand` wird rekursiv aufgelöst, was zu Fehlern führt, wenn man z.B. mittels `\renewcommand` ein existierendes Makro umdefinieren will, und dabei auf der alten Definition aufbauen möchte. Wie man dann vorgehen kann, wird in Anhang E dargestellt.

4.2 newtheorem und newenvironment

Ein Dokument sollte so gegliedert sein, dass man gut erkennt, wo Sätze stehen, wo die zugehörigen Beweise sind etc. Dazu bietet L^AT_EX den Befehl `\newtheorem` an; dieses erzeugt eine neue Umgebung, zusammen mit einem neuen Zähler, der spezifisch zu dieser Umgebung gehört. Man kann mehrere Umgebungen mit demselben Zähler versehen, indem man ein optionales Argument verwendet. Ein Beispiel:

```
\newtheorem{Thm}{Theorem}
\newtheorem{Lem}[Thm]{Lemma}
\newtheorem{Exa}{Example}
...
\begin{Lem}
There is a natural number $n \neq 2$.
\end{Lem}
\begin{Thm}
There is at least one odd natural number.
\end{Thm}
\begin{Exa}
$n=113$ is an odd natural number.
\end{Exa}
Lemma 1. There is at least one odd natural number.
Theorem 2. There is a natural number  $n \neq 2$ .
```

Beweis. Due to Lemma 1 there is at least one odd $n \in \mathbb{N}$. 2 is not odd, hence $n \neq 2$. □

Example 1. $113 \in \mathbb{N}$ is odd and therefore not 2.

Die oben verwendete `proof`-Umgebung wird durch das Paket `amsthm` definiert. Man kann aber auch selber Umgebungen ganz nach Belieben definieren:

```
\newenvironment{Beweis}
  {\bf Beweis }
  {\hfill$\square$}

\newenvironment{History}
  {\textbf{Historical Remark}$\;$}
  {}

\newenvironment{Question}
  {\textbf{Question}$\;$}\begin{itshape}}
  {\end{itshape}}
```

Das erste Argument gibt an, was bei `\begin{...}` ausgeführt werden soll, das zweite, was bei `\end{...}` ausgeführt werden soll.

4.3 Counter

Kapitelnummern, Seitenzahlen usw. werden von L^AT_EX als Counter (Zähler) bezeichnet und sind über eigene Befehle zugänglich, sollte die automatische Berechnung einmal nicht das gewünschte Ergebnis liefern. Beispielsweise kann man mit

```
\setcounter{page}{110}
```

die Seitenzahl festsetzen, oder mit

```
\addtocounter{page}{10}
```

um einen bestimmten Betrag erhöhen.

Will man, dass beispielsweise der `equation`-Counter mit jedem Abschnitt zurückgesetzt wird, setzt man

```
\numberwithin{equation}{section}
```

Vordefinierte Counter sind:

- Für die Gliederung des Dokuments: `part`, `chapter`, `section`, `subsection`, `subsubsection`, `paragraph`, `subparagraph`
- `page`
- `figure` und `table`

- `footnote` und `mpfootnote` (für Fußnoten innerhalb einer `minipage`)
- `equation`
- Für die vier ersten Verschachtelungstiefen der `enumerate`-Umgebung: `enumi`, `enumii`, `enumiii`, `enumiv`

Will man den Inhalt eines Counters im Dokument ausgeben, verwendet man grundsätzlich `\the<COUNTER>`, beispielsweise gibt `\thechapter` die aktuelle Kapitelnummer an. Durch die Trennung des eigentlichen Zählers von einem eigenen Befehl zur Ausgabe des Zählers kann man auf einfache Weise die Formatierung des Zählers im Text mittels `\renewcommand` verändern. Beispielsweise führt

```
\renewcommand{\thesection}{\alph{section}}
```

zur Verwendung von Kleinbuchstaben statt Zahlen, wann immer eine Section-Nummer ausgegeben werden soll; dies betrifft die `\section`- und `\subsection`-Befehle ebenso wie Referenzen und das Inhaltsverzeichnis.

<code>\alph</code>	Kleinbuchstaben
<code>\Alph</code>	Großbuchstaben
<code>\arabic</code>	Dezimalsystem mit arabischen Ziffern
<code>\roman</code>	Römisches Zahlssystem mit Kleinbuchstaben
<code>\Roman</code>	Römisches Zahlssystem mit Großbuchstaben
<code>\fnsymbol</code>	Symbole, die für Fußnoten benutzt werden
<code>\value</code>	Umwandlung in eine Zahl für interne Rechnungen

Eine Besonderheit stellen die Nummerierungen in der `enumerate`-Umgebung dar, da hier `\labelenum...` aufgerufen wird, welches seinerseits `\theenum...` aufruft. `\labelenum...` dient dazu, zusätzliche Formatierungszeichen (wie Klammern, angehängte Punkte oder Fettdruck) einzufügen, die bei einer Referenzierung nicht auftreten sollen.

Will man beispielsweise, dass eine `enumerate`-Umgebung, die innerhalb einer anderen `enumerate`-Umgebung steht, die Nummerierung der ersten Umgebung mit ausgibt, und alle Nummerierungen in arabischen Zahlen stattfinden, sowie fettgedruckt und mit schließender Klammer ausgegeben werden, kann man dies mittels

```
\renewcommand{\theenumi}{\arabic{enumi}}
\renewcommand{\theenumii}{\theenumi.\arabic{enumii}}
\renewcommand{\theenumiii}{\theenumii.\arabic{enumiii}}
\renewcommand{\theenumiv}{\theenumiii.\arabic{enumiv}}
\renewcommand{\labelenumi}{\textbf{\theenumi}}
\renewcommand{\labelenumii}{\textbf{\theenumii}}
\renewcommand{\labelenumiii}{\textbf{\theenumiii}}
\renewcommand{\labelenumiv}{\textbf{\theenumiv}}
```

bewerkstelligen:

- 1) Erster Punkt
- 2) Hier findet eine Diversifizierung statt:
 - 2.1) Teilaspekt A
 - 2.2) Teilaspekt B
 - 2.3) Teilaspekt C gliedert sich wiederum auf:
 - 2.3.1) C1
 - 2.3.2) C2
- 3) Dritter Punkt

Weitere Tricks zum Umgang mit der `enumerate`-Umgebung stellt das Paket `enumitem` zur Verfügung.

4.4 Das Seitenlayout

Um die Ränder anzupassen, lässt sich am einfachsten das `geometry`-Paket verwenden. Beispiel:

```
\usepackage[left=3cm,right=3cm,top=2cm,bottom=3cm,includeheadfoot]{geometry}
```

(Ränder jeweils 3 bzw. 2 cm, dabei sollen Kopf- und Fußzeilen als Teil des Textes gerechnet werden – lässt man `includeheadfoot` weg, werden sie als Teil des Randes gerechnet.) Im laufenden Dokument kann man das Seitenlayout mit Hilfe der Befehle `\geometry`, `\newgeometry` und `\restorygeometry` anpassen, falls nötig.

Bei zweiseitigen Dokumenten (Argument `twoside` in `\documentclass`) beziehen sich „links“ und „rechts“ stets auf die Ränder der ungeraden Seiten – für die geraden Seiten werden die Werte vertauscht, um ein symmetrisches Gesamterscheinungsbild zu gewährleisten.

Um Kopf- und Fußzeilen zu verändern, kann man `\pagestyle` und `\pagenumbering` verwenden, beispielsweise kann man durch `\pagestyle{empty}` die Generierung von Seitenzahlen, Kopf- und Fußzeilen vollständig unterdrücken. Um das Design einer einzelnen Seite zu ändern, kann man `\thispagestyle` einsetzen. Will man Kopf- und Fußzeilen individueller gestalten, sollte man das Paket `fancyhdr` verwenden, siehe auch

<ftp://ctan.tug.org/tex-archive/macros/latex/contrib/fancyhdr/fancyhdr.pdf>

Dieses Dokument verwendet beispielsweise

```
\usepackage{fancyhdr}
\pagestyle{fancy}
```

Die Befehle `\fancyhead` und `\fancyfoot` können genutzt werden, um den Inhalt von Kopf- und Fußzeilen zu definieren.

4.5 Zeilenabstände und -einzüge

Den Zeilenabstand verändert man am besten mit Hilfe des Pakets `setspace`:

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

Für Änderungen im laufenden Dokument bietet `setspace` die Umgebungen `singlespace`, `onehalfspace` und `doublespace` an.

Feinere Einstellungen zum Zeilenabstand bietet der Befehl `\linespread`, der als Argument einen Streckungsfaktor (z.B. 1.2) erwartet.

Der Einzug zu Beginn eines Absatzes kann man beispielsweise durch

```
\setlength{\parindent}{1em}
```

anpassen (`1em` ist eine relative Längenangabe). Will man den Einzug auf Null setzen und dafür den Abstand zwischen zwei Absätzen erhöhen, sollte man einfach das Paket `parskip` einbinden:

```
\usepackage{parskip}
```

4.6 include, input, Style-Files

Bei größeren Projekten ist es sinnvoll, den Quelltext auf mehrere Dateien aufzuteilen. Man verwendet dann `\input` oder `\include`, wobei als Argument der Dateiname ohne `.tex`-Endung angegeben wird. `\include` erzeugt zusätzlich eine neue Seite.

Eine sinnvolle Aufteilung besteht darin, die Befehldefinitionen bzw. die ganze Präambel der Datei in eine eigene Datei auszugliedern.

Für spezifische Aufgaben wurden bereits Befehlsets geschrieben. Diese sind dann häufig in Form sogenannter Style-Files (`.sty`) verfügbar und werden mit `\usepackage` eingebunden. Beispielsweise werden Aufgabenblätter häufig mit einem gemeinsamen Style-File erstellt, und viele Verlage bieten Style-Files zum Download an, um Fachartikel an das spezifische Layout einer Zeitschrift anzupassen. Hier ist z.B. das Style-File der DMV:

```
ftp://ftp.emis.de/pub/EMIS/journals/DMJDMV/tex/doc-latex.sty
```

4.7 Übungen

1. Schreiben Sie ein Makro für 3×3 -Diagonalmatrizen.
2. Realisieren Sie folgende Ausgabe:

Lemma 2. *Ist $f : V \rightarrow W$ linear, so gilt*

$$\dim V = \dim \ker f + \dim \text{bild } f .$$

Beispiel 3. *Es sei $f : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ definiert durch die Matrix*

$$\mathcal{M}_B^B(f) := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} .$$

Dann ist offenbar $\dim \ker f = 1$, $\dim \text{bild } f = 2$ und $\dim V = 1+2 = 3$.

Korollar 4. *Es sei $f : V \rightarrow W$ injektiv, dann ist $\dim \text{bild } f = \dim V$.*

Korollar 5. *Ist $f : V \rightarrow V$ linear, so ist f injektiv genau dann, wenn f surjektiv ist.*

3. Ändern Sie `\subset` so um, dass standardmäßig `\subseteq` ausgegeben wird.
4. Definieren Sie einen neuen `\nsqrt`-Befehl so, dass `\nsqrt{x}` den Ausdruck $\sqrt[x]{x}$ zurückgibt, aber auch ein optionaler Parameter angenommen wird, so dass beispielsweise `\nsqrt[3]{x}` zur Ausgabe $\sqrt[3]{x}$ führt (suchen Sie dafür im Internet nach `\newcommand`).
5. Definieren Sie Umgebungen für Lemmas, Korollare und Beispiele. Googlen Sie den Befehl `\swapnumbers` und nutzen Sie ihn.
6. Definieren Sie eine Umgebung für Hauptsätze, bei denen der Satz oben und unten von einer durchgehenden Linie umgeben ist (`\rightarrow \underline`, `\hspace`, `\vspace`, `\linewidth`, `\hrulefill`, oder alternativ das Paket `framed`):

Main Theorem 3. *Two is more than One.*

7. Betten Sie Ihre Bilder und Tabellen mit `figure` bzw. `table` ein. Suchen Sie dann ein genügend großes Bild, das eine ganze Seite in Anspruch nehmen würde, betten Sie es auf einer eigenen Seite ein und sorgen Sie dafür, dass diese Seite keine Kopf- und Fußzeilen hat (also insbesondere auch keine Seitenzahl).
8. Spalten Sie Ihre Datei in drei Teildateien auf (Befehle, Vorwort, Hauptteil), wobei die Seiten des Vorworts in kleinen römischen Zahlen nummeriert sein sollen, die des Hauptteils in normalen arabischen Zahlen.

5 Briefe

Wir analysieren verschiedene Beispiele, basierend auf den Paketen `letter`, `dinbrief`, `scrlttr2`, `g-brief` und `g-brief2`.

5.1 Umbrüche

Einen Zeilenumbruch erzwingt man mit `\` (optionales Argument: Höhe des Zeilenumbruchs, also beispielsweise `\[5mm]`), einen Seitenumbruch mit `\newpage`. Will man einen Seitenumbruch nur empfehlen, benutzt man `\goodbreak`. \LaTeX ist in der Lage, fast alle Wörter korrekt zu trennen, solange `[n]german` geladen ist. In seltenen Fällen muss man die richtige Silbentrennung mit `\-` vorgeben (alternativ kann man die Schreibung global mittels `\hyphenation` vorgeben, leider funktioniert `\hyphenation` nur dann mit Umlauten, wenn eine sogenannte T1-Kodierung mittels `\usepackage[T1]{fontenc}` verwendet wird). Ein Beispiel für ein falsch getrenntes Wort ist das Wort „betriebssystemu-nabhängig“ aus Abschnitt 1.4. Andere Beispiele sind „Blauf-lossenhering“ und „Unterricht-sentwurf“.

Wird `[n]german` nicht geladen, werden weitere Wörter falsch getrennt, da in der englischen Sprache nach Morphemen und nicht nach Silben getrennt wird. Beispiele hierfür sind „Gle-ichzeitig“ und „Brather-ing“.

5.2 Typografische Details

Eine Besonderheit stellen verschieden langen Striche (`-`, `-`, `-`, `—`) da:

- ist im Textmodus ein Viertelgeviertstrich, z.B. als Binde- und Trennstrich.
- ist im Mathemodus ein Minuszeichen (etwas dicker als ein Halbgeviertstrich).
- ist ein Halbgeviertstrich, z.B. als Bis-Strich und Gedankenstrich im Deutschen.
- ist ein Geviertstrich, der im Englischen als Gedankenstrich verwendet wird.

Will man ein Minuszeichen im Text verwenden, sollte man es stets im Mathemodus setzen, beispielsweise `\$-1\$` (`-1`) statt `-1` (`-1`). Beispiel:

In contrast to [Mü] (pages 12–14)—who set x to be the -1 -fold of y —we want x to be identically y .

Man beachte dabei, dass im Deutschen Leerzeichen vor und nach dem Gedankenstrich geschrieben werden, im Englischen dagegen nicht.

5.3 Übungen

1. Schreiben Sie Briefe mit Hilfe der verschiedenen Umgebungen.
2. Stellen Sie fest, ob die Theorem-Umgebungen und der Math-Mode funktionieren, und verwenden Sie sie.
3. Überlegen Sie sich, wie Sie mit L^AT_EX Serienbriefe erstellen könnten.

6 Beamer-TEX

6.1 Unterschied zwischen Präsentationen und Artikeln

- Kein individuelles zurückblättern (\Rightarrow man benötigt Kontrolle über den Seitenumbruch, und muss diesen auch von Hand definieren),
- wenig Zeit pro Folie, d.h. Zuhörer kommen evtl. nicht mit,
- wenig Platz (keine langen Formeln, keine Beweise),
- Möglichkeiten zum Einsatz von Farbe,
- Möglichkeiten zum Einsatz von Grafiken,
- man neigt zum bloßen Vorlesen der Folien.

6.2 Die Präambel

Ein Beamer-TEX-Dokument wird durch `\documentclass[11pt]{beamer}` eingeleitet. Anschließend wählt man mit `\usetheme` ein Thema aus. Themen tragen Städtenamen (Klassiker: Antibes, Berlin, Copenhagen, Goettingen, Madrid, Malmoe, Warsaw). Mit den folgenden Befehlen kann man weitere Details einstellen:

```
\usecolortheme{
  albatross | beaver | beetle | crane | default | dolphin |
  dove | fly | lily | orchid | rose | seagull | seahorse |
  sidebar tab | structure | whale | wolverine
}
```

```
\usefonttheme{
  default | professional fonts | serif | structure bold |
  structure italic serif | structure small caps serif
}
```

```
\useinnertheme{
```



```

    circles | default | inmargin | rectangles | rounded
}

\useoutertheme{
    default | infolines | miniframes | shadow | sidebar |
    smoothbars | smoothtree | split | tree
}

```

Dabei sind weitere Optionen möglich, z.B. `\usefonttheme[onlymath]{serif}` (sehr empfehlenswert).

Bei der Auswahl von Thema und Farbthema sollte man bedenken, dass die äußere Form der Präsentation nur Hilfsmittel für den Vortrag ist, und die Aufmerksamkeit des Zuhörers auf den Stoff lenken soll. Daher werden üblicherweise die diskreten Farben (Schwarz, Weiß, Grau- und Blautöne) verwendet, und weitere Farben (Grün, Rot) nur selten, und dann zur Akzentuierung eingesetzt (z.B. Hauptsätze). Die Farben sollten natürlich zueinander passen. Eine Zusammenstellung verschiedener Kombinationen von Thema und Farbthema findet man hier:

<http://www.hartwork.org/beamer-theme-matrix/>

Will man selbst eine Farbe vorgeben, so kann man

```
\usecolortheme[...]{structure}
```

einsetzen. Dabei ist der optionale Parameter entweder `named=`, gefolgt von einem Farbnamen (z.B. `Apricot`), oder eine Farbwertdefinition, z.B. `RGB={a, b, c}`, wobei `a`, `b`, `c` den Rot-, Grün- bzw. Blauwert angibt (auf einer Skala von 0 bis 255).

Benutzt man

```
\documentclass[xcolor=dvipsnames]{beamer}
```

so kann man die folgenden 68 vordefinierten Farben nutzen (evtl. plattformabhängig):

```

GreenYellow, Yellow, Goldenrod, Dandelion, Apricot, Peach,
Melon, YellowOrange, Orange, BurntOrange, Bittersweet,
RedOrange, Mahogany, Maroon, BrickRed, Red, OrangeRed,
RubineRed, WildStrawberry, Salmon, CarnationPink, Magenta,
VioletRed, Rhodamine, Mulberry, RedViolet, Fuchsia,
Lavender, Thistle, Orchid, DarkOrchid, Purple, Plum,
Violet, RoyalPurple, BlueViolet, Periwinkle, CadetBlue,
CornflowerBlue, MidnightBlue, NavyBlue, RoyalBlue, Blue,
Cerulean, Cyan, ProcessBlue, SkyBlue, Turquoise, TealBlue,
Aquamarine, BlueGreen, Emerald, JungleGreen, SeaGreen,
Green, ForestGreen, PineGreen, LimeGreen, YellowGreen,
SpringGreen, OliveGreen, RawSienna, Sepia, Brown, Tan,
Gray, Black, White

```

Weitere Einstellungen erreicht man über `\setbeamertemplate`, beispielsweise fügt

```
\setbeamertemplate{footline}[frame number]
```

Seitenzahlen in der Fußnote ein, während

```
\setbeamertemplate{headline}
{%
}
```

```
\setbeamertemplate{navigation symbols}
{%
}
```

die Kopfzeile und die Navigationssymbole löscht.

6.3 Die Titelfolie

Die Titelfolie wird wie folgt definiert:

```
\title[Kurzer Titel]{Langer Titel,\ \ der sich auch über
                    mehrere \ \ Zeilen erstrecken kann}
\subtitle{Untertitel, falls vorhanden}
\author[M.\ Mustermann]{Martha Mustermann}
\institute[Göttingen]{Georg-August-Universität Göttingen}
\date{Datum oder Anlass}
\logo{...}
\titlegraphic{\includegraphics [width=2cm,height=2cm]{datei}}

\begin{document}

[... ggf. \setlength und andere Einstellungen ...]

\begin{frame}
\titlepage
\end{frame}
```

6.4 Der Inhalt

Ein einzelnes Bild heißt „Folie“ („slide“, „overlay“). Ein „Frame“ kann aus mehreren Folien bestehen, z.B. wenn nach und nach Informationen aufgedeckt werden. Zur Definition eines Frames wird die `frame`-Umgebung verwendet, als Argumente folgen Titel und Untertitel des Frames, beispielsweise:

```

\begin{frame}{Nicht-euklidische Geometrie}{Sphärische Geometrie}
[...]
\end{frame}

```

Titel und Untertitel werden je nach eingestelltem Thema in der Kopf-, Fuß- oder Seitenleiste angezeigt. Die Frames können wiederum mit `\part`, `\section` und `\subsection` in einen größeren Zusammenhang gestellt werden.

Ein Frame wiederum wird räumlich gegliedert in Blöcke, die auch graphisch voneinander abgesetzt sind. Davon sind drei Sorten vordefiniert: `block`, `exampleblock` und `alertblock` (alles drei Umgebungen), die sich auch farblich unterscheiden können. Diese drei Blöcke nehmen als Argument einen Titel entgegen (z.B. „Theorem“ oder „Remark“).

Folgende Möglichkeiten existieren, um ein Frame zeitlich in mehrere Slides zu gliedern:

- Der `\pause`-Befehl wartet effektiv an dieser Stelle, bis eine Taste gedrückt wird.
- Der `\item`-Befehl erhält eine Erweiterung: `\item<...>` zeigt seinen Inhalt nur auf den durch `<...>` spezifizierten Slides, das kann z.B. sein: `<4->` oder `<-3,5,7,9->`.
- `\uncover<...>{...}` und `\only<...>{...}` zeigen ihre jeweiligen Argumente nur auf den entsprechenden Slides; die Syntax ist dieselbe wie bei `\item`.
- Auch die Umgebungen nehmen ein `<...>`-Argument an, z.B.:

```

\begin{alertbox}<2-4>{Titel}
[...]
\end{alertbox}

```

- Und auch `\alert` akzeptiert ein `<...>`-Argument, so dass neu erscheinende Elemente markiert werden können.

Man kann die verdeckten Elemente stattdessen in einem Grauton anzeigen lassen, wenn man in der Präambel `\setbeamercovered{transparent}` setzt.

6.5 Anzeigen

Im Acrobat Reader: STRG+L, in evince: F5.

6.6 Übungen

1. Erzeugen Sie das Skelett einer Präsentation, je einen `block`, `exampleblock` und `alertblock`, und experimentieren Sie stichprobenartig mit verschiedenen Stilen

und Farbthemen. Diskutieren Sie in Zweier- oder Dreiergruppen die Lesbarkeit und die Nutzbarkeit Ihrer Präsentationen.

2. Machen Sie sich den Unterschied zwischen `\only` und `\uncover` klar.
3. Stellen Sie fest, dass `\alert` auch im Math-Mode funktioniert und verwenden Sie dies zusammen mit `\uncover`, um folgende Rechnung für $a, b \in \mathbb{F}_2^n$ schrittweise zu erklären:

$$\begin{aligned}
 (a + b) \wedge (a - b) &= a \wedge (a - b) + b \wedge (a - b) \\
 &= \underbrace{a \wedge a}_0 + \underbrace{a \wedge (-b)}_{=-(a \wedge b)} + b \wedge a + \underbrace{b \wedge (-b)}_{=-(b \wedge b)=0} \\
 &= -(a \wedge b) + b \wedge a \\
 &= b \wedge a + b \wedge a \\
 &= 2b \wedge a = 0 \quad (\text{da Charakteristik } 2)
 \end{aligned}$$

4. Schreiben Sie eine Präsentation, in der Sie die Funktion

$$f: \mathbb{R} \rightarrow \mathbb{R}, \quad x \mapsto (x^2 - 5x + 6)(x - 1)$$

diskutieren.

5. Schreiben Sie eine Präsentation, um zu beweisen, dass die Funktionen $\exp(-x^2)$ und $(1 + x^2)^{-1}$ nicht identisch sind, z.B. indem Sie die Funktion $\exp(-x^2) \cdot (1 + x^2)$ ableiten.
6. Sie wollen in einem Frame ein kurzes Lemma darstellen und anschließend nacheinander verschiedene Beispiele und Korollare unter dem Lemma anzeigen, auf jedem Slide eines (d.h. das jeweils vorangehende Beispiel wird wieder ausgeblendet). Überlegen Sie sich, wie Sie dies realisieren können (`\overlayarea`).

Das Beispiel aus Übung 4.7.2 ist hierfür geeignet.

7. Benutzen Sie `\tabular` und `\alert`, um aus der Liste der Zahlen 1 bis 20 auf vier Slides nacheinander diejenigen zu markieren, die durch 2, 3, 5 und 7 teilbar sind.

Weitere Themen

A Verwendung amerikanischer Tastaturen

Einige Kolleginnen und Kollegen verwenden gerne amerikanische Tastaturen, da der Backslash auf diesen Tastaturen leichter zu erreichen ist.

Wenn man auf einer solchen Tastatur einen deutschsprachigen Text schreiben möchte, bietet das Paket `ngerman` folgende abkürzende Schreibweise:

`"a, "o, "u, "A, "O, "U, "s`

Hierbei sollte beachtet werden, dass das Anführungszeichen durch `ngerman` zu einem Steuersymbol umdeklariert wird und nicht weiter als reguläres Symbol zur Verfügung steht.

Für englische Anführungszeichen verwendet man:

`'Text'` \Rightarrow `"Text"`

Deutsche Anführungszeichen erhält man über

`\glqq Text\grqq` \Rightarrow `„Text“`

`\glq Text\grq` \Rightarrow `,Text‘`

`"Text"` \Rightarrow `„Text“`

dagegen liefert `"Text"` eine Fehlermeldung. Zuweilen werden in deutschsprachigen Texten auch französische Anführungszeichen benutzt. Diese erhält man durch:

`\flqq Text\frqq` oder `"<Text">` \Rightarrow `«Text»`

B Leerzeichen

Nach dem Satzzeichen steht ein Leerzeichen. Davor nicht. Diese Regel kann `LATEX` nicht realisieren, sondern muss vom Autor selbst in seinem Text korrekt umgesetzt werden. Manchmal werden geschützte Leerzeichen benötigt, diese werden mit `~` eingegeben. Beispielsweise sollte nach Möglichkeit `„Dr. Faust“` nicht umgebrochen werden, da sonst der Lesefluss gestört ist. Also verwendet man `Dr.~Faust`, Beispiel:

Dr. Faust Dr. Faust Dr. Faust Dr. Faust Dr. Faust Dr. Faust Dr. Faust
 A Dr. Faust Dr. Faust Dr. Faust Dr. Faust Dr. Faust Dr. Faust Dr. Faust
 AB Dr. Faust Dr. Faust Dr. Faust Dr. Faust Dr. Faust Dr. Faust Dr. Faust
 ABC Dr. Faust Dr. Faust Dr. Faust Dr. Faust Dr. Faust Dr. Faust

Dr. Faust

vergleiche

Dr. Faust Dr. Faust Dr. Faust Dr. Faust Dr. Faust Dr. Faust Dr. Faust
A Dr. Faust Dr. Faust Dr. Faust Dr. Faust Dr. Faust Dr. Faust Dr. Faust
AB Dr. Faust Dr. Faust Dr. Faust Dr. Faust Dr. Faust Dr. Faust Dr. Faust Dr.
Faust
ABC Dr. Faust Dr. Faust Dr. Faust Dr. Faust Dr. Faust Dr. Faust Dr. Faust Dr.
Faust

Daneben gibt es noch `\`, als schmales geschütztes Leerzeichen, das in Abkürzungen steht („z. B.“), vor Einheiten („3 m“) und als Tausendertrennzeichen („123 456,56“).

Als weitere Leerzeichen verschiedener Größe stehen zur Verfügung: `\,`, `\;`, `\quad` und `\qquad`, sowie der Rückschritt `\!`.

Ein `\quad` ist die Breite eines „M“ im aktuellen Font, in dieser Einheit gilt:

Befehl	Breite	Beispiel
<code>\!</code>	$-3/18 \text{ quad}$	
<code>\,</code>	$3/18 \text{ quad}$	
<code>\;</code>	$5/18 \text{ quad}$	
<code>\</code>	$6/18 \text{ quad}$	
<code>\quad</code>	1 quad	
<code>\qquad</code>	2 quad	

Der Befehl `\:` steht normalerweise für $4/18 \text{ quad}$, ist aber nicht mit dem `tikz`-Paket verträglich.

Das normale Leerzeichen und „~“ sind flexibel.

C Grafiken aus Maple einbinden

Maple kann Funktionen plotten und diese Plots als `eps`, `png` oder `jpeg` exportieren. Dazu ruft man in Maple ein neues Arbeitsblatt (Worksheet) auf und benutzt den Befehl

```
plot(Funktion(x), x=min..max, y=min..max, numpoints=Stützpunkte)
```

(Die hinteren Argumente sind dabei optional, aber sehr praktisch.) Für mehrere Funktionen benutzt man

```
plot([Funktion1(x), Funktion2(x), ... ], ...)
```

Parametrische Plots sind möglich mit

`plot([x-Wert(t), y-Wert(t), t=min .. max], ...)`

3D-Plots erstellt man mittels

`plot3d(Funktion(x,y), x=min..max, y=min..max, numpoints=Stützpunkte)`

Man beachte, dass es für 3D-Plots kein z -Argument gibt, um die z -Achse anzupassen; diesem Problem kann man mit den Funktionen `min` und `max` begegnen.

Man kann anschließend das Diagramm modifizieren, indem man auf die freie Fläche bzw. auf den jeweiligen Graphen im Diagramm rechts-klickt. Dokumente, die zum Druck gedacht sind, sollten Grafiken in Grautönen enthalten, da je nach Einstellung farbige Anteile garnicht oder in unerwartetem Grauton gedruckt werden. Es empfiehlt sich also, die Graphen bereits im Vorfeld anzupassen.

Mittels Rechts-Klick und „Export“ kann man dann das fertige Diagramm als Grafik exportieren.

D Alte Befehle zum Seitenlayout

Die folgenden Befehle zählen als veraltet und können mit modernen Paketen zu paradoxen Ergebnissen führen. Stattdessen sollte man z.B. das `geometry`-Paket verwenden.

Das normale Seitenlayout von \LaTeX ist an einen optimalen Lesefluss angepasst, d.h. breite Ränder, bestimmte relative Schriftgrößen, ein spezieller Zeichensatz (Computern Modern), Zeilen mit einer Höchstzahl an Zeichen, um beim lesen nicht in der Zeile zu verrutschen.

Dennoch lässt sich das Layout durch eine Vielzahl an Parametern verändern. Dieser Text hat z.B. folgendes festgelegt:

```
\usepackage[widemargins]{a4}      % Weite Ränder bei A4-Format
\parindent0em                    % Absätze werden nicht eingerückt
\parskip1.5ex plus0.5ex minus0.5ex % Abstand zwischen zwei Absätzen
\topmargin4mm                   % Abstand Seitenrand <-> Kopfzeile
\headheight5mm                  % Höhe der Kopfzeile
\headsep6mm                     % Abstand Kopfzeile <-> Texttrumpf
\topskip5mm                      % Zusätzl. Freiraum am ob. Seitenrand
\textwidth150mm                 % Breite Texttrumpf
\evensidemargin15mm             % Linker Rand bei geraden Seiten
\oddsidemargin15mm             % Linker Rand bei ungeraden Seiten
\textheight230mm                % Höhe des Texttrumpfs
\hoffset -10mm                  % Zusätzl. Freiraum am li. Seitenrand
\footskip10mm                   % Abstand Fußzeile <-> Seitenrand
```

Abstände können (wie im `\parskip`-Beispiel) auch mit Fehlermargen definiert werden, um L^AT_EX größere Freiheit im Ausrichten der Absätze zu gewähren.

E Auflösen rekursiver Makrodefinitionen

`\newcommand` und `\renewcommand` werden rekursiv aufgelöst, was zu Problemen führen kann, wenn man ein Makro umdefinieren will, und dabei auf dem Makro selbst aufbauen möchte. Hier muss man mit dem alten T_EX-Befehl `\let` arbeiten, der eine andere Syntax besitzt und im Gegensatz zu `\[re]newcommand` nicht rekursiv aufgelöst wird. Ein Beispiel:

```
\newcommand{\origforall}{\} \let\origforall=\forall
\renewcommand{\forall}{\,\origforall\,}
```

F BibTeX

Das Paket BibTeX ist ein Ersatz für die `thebibliography`-Umgebung. Dazu erstellt man eine externe Datei (Endung `.bib`), die die nötigen Quellenangaben in einer speziellen Syntax enthält. Diese externe Datei kann dann von mehreren Dokumenten verwendet werden, da BibTeX automatisch erkennt, welche Quellenangaben aus der externen Datei tatsächlich mit `\cite` referenziert werden – nur diese Quellen werden dann auch in das jeweilige Dokument übernommen.

Die BibTeX-Syntax ist etwas gewöhnungsbedürftig, und es wird ein weiteres, externes Programm namens `bibtex` benötigt, um die Referenzen in einem Dokument zu erkennen und die nötigen Literaturangaben bereitzustellen. Daher ist der Einsatz von BibTeX nur dann zu empfehlen, wenn man plant, viele Artikel mit einander ähnlichen Quellen zu schreiben.

Hier ist Beispielauszug aus einer BibTeX-Datei:

```
@article{AG99,
  shorthand="AG99",
  title={Braided {H}opf algebras over non-abelian groups},
  author={Andruskiewitsch, Nicol{\`a}s and Gra{\~n}a, Mat{\i}as},
  journal={Boletin de la Academia Nacional de Ciencias en C{\`o}rdoba},
  volume={63},
  pages={45--78},
  year={1999},
}

@misc{Zoo,
```



```

shorthand="GranaZoo",
title={Nichols algebras of nonabelian group type, zoo of examples},
author={Gra{\~n}a, Mat{\i}as},
note={Available at http://mate.dm.uba.ar/\$\sim\$matiasg/zoo.html},
year={2000}
}

```

```

@book{SakQM,
shorthand={Sak11},
title={Modern Quantum Mechanics},
author={J.J. Sakurai and J. Napolitano},
publisher={Addison-Wesley},
year={2011},
notes={Second Edition}
}

```

Die drei Quellen würden nun mit `\cite{AG99}`, `\cite{Zoo}` und `\cite{SakQM}` referenziert. Über verschiedene Optionen lässt sich dann im Dokument einstellen, wie die Quellenangaben im Dokument formatiert werden sollen; anstelle der `thebibliography`-Umgebung würde man beispielsweise am Ende des Dokuments die zwei Zeilen

```

\bibliographystyle{abbrv}
\bibliography{DateiNameOhneEndungBib}

```

einfügen. Die Zeile `\bibliographystyle{abbrv}` legt fest, dass die Referenzen im Haupttext numerisch angezeigt werden (z.B. „[12]“), während beispielsweise `\bibliographystyle{abbrv}` die `shorthand`-Abkürzungen anzeigt (also z.B. „[SakQM]“). Weitere Stile sind verfügbar; üblicherweise wird der Stil allerdings vom jeweiligen Journal vorgegeben, in dem man veröffentlichen möchte, und ist oft bereits in ihren `.sty`-Dateien voreingestellt.

G Index

Zur Erstellung eines Index kann man das Paket `makeidx` verwenden (Befehle `\index{...}` und `\makeindex`, außerdem ist der externe Aufruf eines Indexprozessors nötig).