LaTeX-Schnellkurs

Henrik Gebauer, 2010, Version 1.0.2

1 Allgemeines

1.1 Grundsätzliches

TeX-Code zu schreiben hat viel Ähnlichkeit mit Programmieren. Den Code kann man mit jedem beliebigen Texteditor schreiben, allerdings gibt es einige Programme, die einem beim Erstellen helfen, indem sie einige Schlüsselwörter farbig darstellen und andere Vereinfachungen anbieten. Für Linux gibt es beispielsweise `Kile` oder `emacs`. Der Code wird kompiliert, d.h. er wird ausgewertet und es wird eine Datei im PDF- oder DIV-Format erstellt. Das Kompilieren wird bei vielen Editoren vereinfacht, sodass nur noch ein Button geklickt werden muss. Ansonsten benutzt man die Kommandozeile, hier z.B. für den Compiler `pdflatex`:

```
$ pdflatex hallo-welt.tex
```

Dies erstellt die Datei `hallo-welt.pdf`, die das fertige Dokument enthält. Bei Syntax- oder anderen Fehlern wird das Kompilieren abgebrochen oder eine Warnung angezeigt.

Grundgerüst einer LaTeX-Datei:

```
\documentclass[a4paper,10pt]{scrartcl}  % legt den Dokumenttyp fest

\usepackage[utf8x]{inputenc}    % lädt das Paket inputenc mit dem Parameter "utf8x"
%% weitere Präambel

\begin{document}
%% hier steht der eigentliche Inhalt
\end{document}
```

Die erste Zeile stellt A4-Papier und die Schriftgröße 10pt ein und lädt als Vorlage die Dokumentklasse `scrartcl`. Es gibt noch andere, aber `scrartcl` ist meistens eine gute Wahl. Wie man sieht beginnen Kommentare mit dem Prozentzeichen und enden mit dem Zeilenende. Die zweite Zeile lädt das Paket `inputenc` und übergibt die Information, dass der Code UTF-8-kodiert ist (unter Linux bei vielen Editoren voreingestellt). Statt `utf8x` ist auch `latin1` üblich. In der „Präambel“ können noch weitere Pakete geladen werden, einige Einstellungen getroffen oder eigene Befehle definiert werden. Ganz nützlich ist erstmal folgende Liste

```
\usepackage{ngerman}     % deutsche Rechtschreibung (wichtig für automatische Silbentrennung)
\usepackage{array}       % erweiterte Tabellen
\usepackage{graphicx}    % Grafiken
\usepackage{ziffer}      % deutsche Konvention: Komma als Dezimaltrenner
\usepackage{amsmath,amssymb,amsthm}  % einige Mathe-Befehle
\usepackage{hyperref}    % Web-Adressen, z.B, im Literaturverzeichnis,
                         % können mit \url{Adresse} gesetzt werden

\setlength{\parindent}{0em}   % neue Absätze nicht einrücken
\setlength{\parskip}{0.8ex}   % sondern etwas Platz dazwischen lassen
```

1.2 Befehle

Es gibt zwei Arten von Befehlen

1. Die meisten haben die Form Backslash (\) plus Name, z.B. `\alpha` für α. Es wird erkannt, dass der Name vorbei ist, sobald ein Nicht-Buchstabe folgt.
2. Manche Befehle sind ein einzelnes Sonderzeichen, z.B. der Unterstrich `_` für einen Index an einer Variablen. `a_1` wird zu a_1.

Einige Befehle benötigen ein (manchmal auch mehrere) Argumente. Diese werden in geschwungene Klammern eingefasst, z.B. `\sqrt{16}` für $\sqrt{16}$. Wenn man nur ein einzelnes Zeichen (z.B. die 1 bei `a_1`) als Argument übergeben will, kann man die Klammern auch weglassen. Einen längeren Index würde man also mit `a_{i,j}` (für $a_{i,j}$) setzen. Bei Befehlen mit mehreren Argumenten hat man dann mehrere geschweifte Klammern hintereinander, z.B. `\frac{a}{b}` für $\frac{a}{b}$.

Für manche Befehle gibt es zusätzliche optionale Argumente, d.h. sie können angegeben werden, müssen aber nicht. Diese werden in eckige Klammern eingefasst und stehen vor den anderen Argumenten, z.B. bei `\sqrt[3]{8}` für $\sqrt[3]{8}$.

1.3 Umgebungen

Es gibt Umgebungen, innerhalb denen die Befehle etwas anders verarbeitet werden. Sie beginnen mit dem Befehl `\begin{name}` und enden mit `\end{name}`. Z.B. die Umgebung `center` für zentrierten Text:

```
\begin{center}
  Hallo Welt!
\end{center}
```

wird zu

Hallo Welt!

Man kann auch Umgebungen ineinander schachteln. Um z.B. ein Bild zu zentrieren, kann man eine Bild-Umgebung in eine Zentrier-Umgebung packen. Wichtig ist, dass die Umgebung die zuletzt geöffnet wird auch zuerst wieder geschlossen wird.

Es gibt auch noch einige spezielle Umgebungen ohne die Befehle `\begin` und `\end`:

- die Umgebungen von `\(` bis `\)` oder von `$` bis `$` oder von `\[` bis `\]` für Formeln (siehe unten)
- die Umgebung von `{` bis `}`, die nichts tut, außer dass sie Dinge zusammen fasst (z.B. wie oben gesehen bei Argumenten). Diese Umgebung braucht man auch für Befehle, die bis zum Ende einer Umgebung gelten. Z.B. gilt `\itshape` für kursiven Text bis zum Ende der Umgebung. `{\itshape hallo du}` wird zu *hallo du*. Lässt man die Klammern weg, wird der Rest des Dokuments kursiv.

1.4 Wechsel zwischen Mathe- und Text-Modus

Es wird zwischen dem Text-Modus und dem Mathe-Modus unterschieden. Sie haben jeweils ihre eigenen Befehle, die auch nur in dem jeweiligen Modus benutzt werden dürfen, sonst gibt es eine Fehlermeldung.

Normalerweise ist man im Text-Modus. Es gibt einige Umgebungen, die in den Mathe-Modus schalten (siehe dort).

2 Text-Modus

Normalen Text kann man einfach eintippen und er wird dann angezeigt. Mehrere Leerzeichen werden zu einem zusammen gefasst und Zeilenumbrüche werden wie Leerzeichen behandelt. Zwischen Absätzen wird eine Zeile frei gelassen.

2.1 einige Umgebungen

Name	Bedeutung	Beispiel
`center`	Text (oder Abbildungen usw.) zentrieren	hallo
`itemize`	Liste, jeder Eintrag beginnt mit `\item`	• erster Eintrag • zweiter Eintrag
`enumerate`	Aufzählung, jeder Eintrag beginnt mit `\item`	1. erster Eintrag 2. zweiter Eintrag
`tabular`	Tabelle (siehe unten)	
`table`	Tabelle (siehe unten)	
`figure`	Abbildung (siehe unten)	

2.2 einige Befehle

Name	Bedeutung	Beispiel
`\\`	Zeilenumbruch ohne einen neuen Absatz zu beginnen	
`\%`	ein Prozentzeichen	%
`\&`		&
`\-`	Silbentrenner (falls LaTeX es selbst nicht hinbekommt)	
`~`	zusätzliches Leerzeichen	
`{\bfseries Text}`	fett	**Text**
`{\itshape Text}`	kursiv	*Text*
`\section{Überschrift}`	Abschnnitt	
`\subsection{Überschrift}`	Unterabschnnitt	
`\subsubsection{Überschrift}`	Unterunterabschnnitt	
`\newpage`	neue Seite	
`\tableofcontents`	Inhaltsverzeichnis	
`\vspace{2em}`	vertikaler Freiraum der Höhe 2em	

3 Mathe-Modus

Buchstaben werden normalerweise als Variablen behandelt und werden schräg gedruckt. Leerzeichen und Zeilenumbrüche werden ignoriert.

Symbole, die es auf der Tastatur gibt, z.B. = oder + können normalerweise benutzt werden.

3.1 einige Umgebungen

Umgebungen, die den Mathe-Modus aktivieren:

Name	Bedeutung
`equation`	einzeilige Formel (mit Nummer der Gleichung)
`equation*`	einzeilige Formel (ohne Nummer)
`\[...\]`	Kurzschreibweise für `\begin{equation*}...\end{equation*}`
`align`	mehrere Formeln untereinander (mit Nummer), wobei die Zeichen `&` untereinander ausgerichtet werden (aber nicht angezeigt werden), um z.B. =-Zeichen untereinander zu positionieren. Die Zeilen werden mit `\\` getrennt.
`align*`	wie `align` ohne Nummer. z.B. für eine mehrzeilige Gleichung
`\(...\)`	Formel im Fließtext
`$...$`	alternative Schreibweise für `\(...\)`

Umgebungen innerhalb des Mathe-Modus:

Name	Bedeutung	Beispiel
`cases`	Fallunterscheidung, Ausrichtung wie bei Tabellen	$\delta_{ab} = \begin{cases} 1 & a = b \\ 0 & a \neq b \end{cases}$
`array`	Matrix (wie Tabelle, siehe unten)	$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$

3.2 einige Befehle

Name	Bedeutung	*Beispiel*
`\delta, \omega, \Delta, \Omega, ...`	griechische Buchstaben	$\delta, \omega, \Delta, \Omega$
`\sqrt[n]{r}`	Wurzel	$\sqrt[n]{r}$
`_1`	Index	a_1
`^1`	Exponent	a^1
`_{unten}^{oben}`	Grenzen	$\sum_a^b, \int_a^b$
`\sin, \cos, \exp, \min, \lim...`	einige Funktionen usw.	$\sin, \cos, \exp, \min, \lim ...$
`\lim_{a \to 0}`	Limes (o.ä.) mit Grenze	$\lim_{a \to 0}$
`\infty`	unendlich	∞
`\sum, \int, \prod`	Summe, Integral, Produkt	$\sum, \int, \prod$
`\Rightarrow, \rightarrow, \to \Leftrightarrow`	Pfeile	$\Rightarrow, \rightarrow, \to, \Leftrightarrow$
`\propto`	proportional	$\propto$
`\not =`	etwas durchstreichen (hier =)	$\neq$
`<, >, \leq, \geq, \approx`		$<, >, \leq, \geq, \approx$
`\vec{a}`	Vektor	$\vec{a}$
`\dot{a}, \ddot{a}`	Zeitableitung	$\dot{a}, \ddot{a}$
`\partial`	Ableitung	∂
`\frac{a}{b}`	Bruch	$\frac{a}{b}$
`\hbar`		$\hbar$
`\pm`		$\pm$
`A^{\dagger}`		$A^\dagger$
`\dots`		$\dots$
`\nabla`		∇
`\underbrace{Formel 1}_{Formel 2}`	unterklammen	$\underbrace{Formel1}_{Formel2}$
`\stackrel{abc}{=}`	Text auf =-Zeichen (o.ä.)	$\stackrel{abc}{=}$
`\exists, \forall`		$\exists, \forall$

`\left( \right),`
 	
`\left[ \right], ...`
	Klammern, die sich
	
	in der Größe anpassen
		$\left(\frac{1}{n}\right)^2$

`\mathrmText`	aufrechter Text (z.B. Kommentare, Einheiten)	$5\ \mathrm{km\ s^{-1}}$
`\cdot`	Mal-Punkt	$a \cdot b$
`~`	Leerzeichen	$a\ b$

4 Abbildungen

Bilder werden so eingefügt:

```
\begin{figure}[htb]
  \centering                                          % Bild zentrieren
  \includegraphics[width=0.9\textwidth]{Dateiname}  % das eigentliche Bild (90% der Textbreite)
  \caption{Beschriftung}
  \label{name}                                        % Bezeichner für Verweise
\end{figure}
```

Der Dateiname wird ohne Endung angegeben. Am einfachsten ist es, wenn sich das Bild im gleichen Ordner befindet, wie das LaTeX-Dokument.

Die dritte Zeile fügt das Bild ein und kann auch für sich alleine stehen. Die anderen Zeilen sind aber trotzdem meist sinnvoll.

`name` wird durch einen internen Namen ersetzt (darf nur Buchstaben und _ enthalten). Diesen Namen kann für Verweise (siehe unten) benutzen. Wenn man nicht auf das Bild verweisen will, kann man die Zeile auch weglassen.

Man kann auch mehrere Bilder nebeneinander setzen, einfach noch ein weiterer `\includegraphics`-Befehl. Dann muss die Breite entsprechend klein sein, damit sie nebeneinander passen.

`[htb]` steht dafür, wo das Bild platziert werden soll. h für here, t für top, b für bottom. Also in diesem Fall am liebsten direkt an der Stelle, wo es steht, wenn das nicht passt oben auf die Seite, wenn das auch nicht passt, unten auf die Seite. Man sollte also nicht schreiben „Auf dem folgenden Bild sieht man...“, sondern „Auf Abb. 1 sieht man...“, weil man nie weiß, wo genau das Bild platziert wird. Man kann auch das Zeichen ! hinter eine Position setzen, dann wird die Wirkung verstärkt. Ein Bild mit `[h!tb]` wird also in aller Regel an der Stelle eingefügt, an der es im Code vorkommt.

5 Tabellen und Matrizen

Für Tabellen gibt es die `tabular`-Umgebung. Beispiel:

```
\begin{tabular}{l|rcp{4cm}}
  Hänsel & und  & Gretel & verliefen                                    \\ \hline
  sich   & im   & Wald.  & Es war so finster und auch so bitterkalt. \\
\end{tabular}
```

wird

Hänsel	und	Gretel	verliefen
sich	im	Wald.	Es war so finster und auch so bitterkalt.

Spalten werden mit `&` getrennt, Zeilen mit `\\`. Eine horizontale Linie wird mit `\hline` eingefügt. `{l|rcp{4cm}}` steht dafür, dass die Tabelle vier Spalten haben soll, bei der die erste links- (`l`), die zweite rechtsbündig (`r`), die dritte zentriert (`c`) sein soll und die vierte ist Blocksatz mit 4cm-Breite und automatischem Zeilenumbruch. Ansonsten gibt es keine automatischen Zeilenumbrüche und zu volle Zeilen erzeugen eine Warnung (und ein hässliches Ergebnis). Zwischen der ersten und zweiten Spalte soll eine Linie (`|`) gezogen werden.

Matrizen werden genauso gesetzt wie Tabellen, nur dass es nicht `tabular` heißt, sondern `array` und dass man im Mathe-Modus sein muss. Die Matrix ist erstmal auch nur eine Tabelle. Wenn man sie in Klammern setzen möchte, kann man `\left(` und `\right)` benutzen.

Beschriftete und benannte (für Verweise) Tabellen werden in eine `table`-Umgebung verpackt, die fast so aussieht wie `figure` für Abbildungen.

```
\begin{table}[htb]
  \centering
  \begin{tabular}{Spalten}
   ...
  \end{tabular}
  \caption{Beschriftung}
  \label{name}
\end{table}
```

6 Verweise (z.B. auf Gleichungen oder Abbildungen oder ins Literaturverzeichnis)

Abbildungen, Tabellen und nummerierte Gleichungen können mit `\label{name}` benannt werden. Der Befehl muss noch innerhalb der entsprechenden Umgebung stehen. An anderer Stelle kann man dann mit `\ref{name}` einen Verweis darauf einfügen, d.h. die Nummer der Abb., Tabelle oder Gl. wird dort eingefügt. Mit dem Befehl `\pageref{name}` wird die Seitennummer eingefügt, auf der die Gl. usw. steht.

Zitate ins Literaturverzeichnis macht man ähnlich. Die Einträge im Literaturverzeichnis haben alle einen Namen mit `\bibitem{name}` Name bekommen. Mit `\cite{name}` fügt man ein Zitat ein, d.h. eckige Klammern mit einer Zahl, z.B. [1]. Mit `\cite[zusatz]{name}` kann man noch zusätzliche Informationen angeben, z.B. die Seitenzahl, [1, zusatz].