

Ein allererstes L^AT_EX-Dokument

Steffen Mecke* Alter Ego

23. April 2007

Zusammenfassung

Dieses Dokument soll als Anleitung für die allerersten Schritte zur Erstellung einer Ausarbeitung oder eines Artikels mit L^AT_EX dienen. Der Quelltext (siehe Anhang A) enthält weitere wichtige Hinweise.

1 Einleitung

T_EX (sprich: „tech“) ist ein Textsatzsystem, das von Prof. Donald E. Knuth entwickelt wurde. L^AT_EX (sprich „latech“) ist ein in den Jahren 1982-1986 von Leslie Lamport geschriebenes Makropaket für T_EX. Beide sind dazu gedacht (wissenschaftliche) Texte professionell zu veröffentlichen. Diese Datei (`einfuehrung.tex`) soll dazu dienen, die wichtigsten Konzepte vorzustellen, die zur Erstellung eines eigenen L^AT_EX-Dokumentes nötig sind. Abschnitt 2 stellt die technischen Voraussetzungen und die Vorgehensweise vor. In Abschnitt 3 werden ein paar wichtige Befehle und Umgebungen kurz vorgestellt. Weiterführende Literatur wird im letzten Abschnitt besprochen.

2 Von der Idee zum Dokument

Folgende Dinge braucht man für das Arbeiten mit L^AT_EX:

- Einen Texteditor. Unter Linux z.B. `emacs`, `vi`, `kate` oder `kile`, unter Windows z.B. WinEdit ([14]), TeXnic ([13], eigentlich eine komplette Entwicklungsumgebung für L^AT_EX), Emacs oder, wenn es sein muss, Notepad. LyX ([15]) gibt es sogar für Linux, Mac und Windows. Officesysteme wie Word oder Staroffice sind weniger geeignet, da sie nicht dazu gedacht sind, reine Textdateien zu erzeugen und zu bearbeiten.
- Eine L^AT_EX-Installation. Diese ist bei praktisch allen Linux-Distributionen bereits dabei (z.B. `tetex`). Für Windows gibt es unter anderen MiKTeX ([11]) oder fpTeX ([12]).
- Ein Programm zur Betrachtung von dvi- und/oder Postscript- und/oder pdf-Dateien. Für Linux sind das etwa `xdvi`, `gv` und `acroread` oder `xpdf`. Für Windows gibt es GS View ([17]) und den Acrobat Reader ([16]). Siehe auch [18].

Einige der empfohlenen Editoren haben extra L^AT_EX-Modi, mit „syntax highlighting“ und speziellen Befehlen für das Erstellen, Übersetzen und Betrachten von L^AT_EX-Dateien, die die Arbeit sehr erleichtern.

*Universität Karlsruhe, `Steffen.Mecke(at)kit.edu`

Um Bilder zu erzeugen, die in einem Dokument verwendet werden sollen, braucht man noch ein geeignetes Zeichenprogramm. Es sollte Bilder im Postscript-Format (`.eps`) und/oder im pdf-Format (`.pdf`) schreiben können. Geeignet sind z.B. `xfig` (für Linux) oder `ipe` ([19], für Linux oder Windows).

Es gibt noch eine Reihe weiterer Werkzeuge, die die Arbeit mit \LaTeX erleichtern, wie zum Beispiel `BibTeX`. Aber für den Anfang sollten diese erst mal reichen.

Folgende Schritte sind notwendig:

1. Die Eingabedatei mit dem Editor schreiben (z.B. `einfuehrung.tex`).
2. Die Eingabedatei kompilieren. Unter Linux geht das mit dem Befehl

```
# latex einfuehrung
und
# dvips einfuehrung.dvi
```

zur Erzeugung einer Postscript-Datei, bzw. mit

```
# pdflatex einfuehrung.tex
```

zur Erzeugung einer pdf-Datei. Manchmal muss man dies zwei mal hintereinander machen, da Querverweise (`\ref{}`, `\cite{}`) oft erst beim zweiten Mal richtig gesetzt werden.
3. Falls ein Fehler in der Eingabedatei war, hält `latex` an dieser Stelle an und wartet auf eine Eingabe. Man kann dann entweder versuchen, durch Eingabe von `<Enter>` einfach weiterzumachen oder durch Eingabe von `x` abbrechen.
4. Das Ergebnis anschauen. Unter Linux – je nachdem, was für eine Art von Datei erzeugt wurde – mit einem der drei folgenden Kommandos:

```
# xdvi einfuehrung.dvi
# gv einfuehrung.ps
# acroread einfuehrung.pdf
```
5. Eventuelle (Tipp-)Fehler in der Eingabe korrigieren, erneut kompilieren und so weiter ...

3 Elemente eines \LaTeX -Dokuments

Ein \LaTeX -Dokument besteht aus einfachem Text und speziellen \LaTeX -Kommandos. Das Grundgerüst sieht so aus:

```
\documentclass{article}
% Praeambel
\usepackage{Paket1, Paket2}
% ...
\begin{document}
  Text
\end{document}
```

Normaler Text wird einfach getippt, wie er ist. \TeX kümmert sich automatisch um die richtigen Abstände zwischen den Wörtern (egal ob dort ein oder mehrere Leerzeichen stehen), Zeilen- und Seitenumbrüche und vieles mehr. Vor Satzzeichen sollte man jedoch kein Leerzeichen lassen, das sieht sonst nicht gut aus. Absätze kennzeichnet man durch mindestens eine Leerzeile.

\TeX -Kommandos werden durch den Backslash (`\`) eingeleitet. Sie beziehen sich in der Regel auf das nächste Zeichen oder ihr Argument, das gewöhnlich in geschweifte Klammern eingeschlossen wird, etwa `\texttt{emacs}` (erzeugt `emacs`). Dann gibt es noch *Umgebungen*, etwa die `itemize`-Umgebung. Ihr Argument ist alles, was zwischen `\begin{itemize}` und `\end{itemize}` steht.

Viele Kommandos sind nicht automatisch vorhanden, sondern werden durch besondere *Pakete* bereitgestellt, die erst mit dem `\usepackage{}`-Befehl geladen werden müssen.

Will man hinter einem Befehl unbedingt ein Leerzeichen einfügen, kann man dies etwa wie folgt erreichen: `\LaTeX\ ist sch"on`

Manchmal will man an einen Zeilenumbruch an einer bestimmten Stelle verhindern (etwa Kapitel 1). Das kann man das mit Hilfe der Tilde tun: `Kapitel~1`. Umlaute (äöüß usw.) erzeugt man entweder durch die Kommandos `"a"o"u"s` oder – falls wie hier das `babel`-Paket mit der Option `german` verwendet wird – durch `"a"o"u"s`. Eine **fette Schriftart** wählt man mit dem Befehl `\textbf{fetter Text}` aus. Damit sollte man jedoch sehr sparsam umgehen. *Wichtige Begriffe* sollte man lieber mit `\emph{Wichtig}` kursiv setzen.

Mit dem Befehl `\includegraphics[Optionen]{Dateiname_ohne_Endung}` bindet man ein Bild ein. Um Postscript Dateien zu erzeugen sollten die Bilder im (Encapsulated-)Postscript-Format (`.eps`) vorliegen, für PDF-Dateien sollten auch die Bilder in dieses Format (`.pdf`) haben. Ein Bild sollte immer in eine `figure`-Umgebung gesetzt werden, damit man ihm eine Bildunterschrift und einen Bezeichner (`\label{fig:beispiel2}`) geben und sich später darauf beziehen kann (hier mit `\ref{fig:beispiel2}`). Die optionalen Parameter `[htbp]` bewirken, dass die Abbildung automatisch an eine Stelle im Text „fließt“, wo sie ästhetisch schön wirkt. Das funktioniert auch meistens gut! Das Beispiel in [Abbildung 1](#) wurde mit dem Zeichenprogramm `ipe` erstellt – beachten Sie die Beschriftung!

```
\begin{figure}[htbp]
  \centering
  \includegraphics{fig/beispiel}
  \caption{Eine Beispielabbildung.}
  \label{fig:beispiel1}
\end{figure}
```

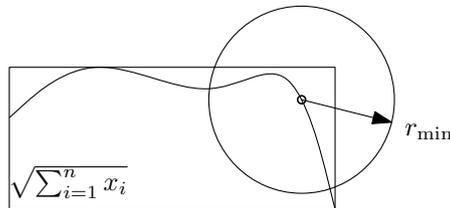


Abbildung 1: Eine Beispielabbildung.

Eine Formel wie in [Abbildung 1](#) wird mit der Zeichenfolge `$$\sqrt{\sum_{i=1}^n x_i}$$` produziert: $\sqrt{\sum_{i=1}^n x_i}$. Man unterscheidet zwischen Formeln im Fließtext, die zwischen `$`-Zeichen eingeschlossen werden und Formeln, die alleine in einer Zeile stehen und mit `\[... \]` gekennzeichnet werden, z.B.

$$g(x) := \sqrt{\sum_{i=1}^n x_i} \quad \forall x \in \mathbb{R}_{\geq 0}$$

Außerdem gibt es noch nummerierte Formeln (wie [\(1\)](#)), auf die man sich später im Text beziehen kann mit `\ref{eq:forces}`. Sie werden mit einer `equation`-Umgebung erzeugt, etwa

$$f_{\text{rep}}(p_u, p_v) := \frac{c_\rho}{\|p_v - p_u\|^2} \cdot \overrightarrow{p_u p_v} \tag{1}$$

Sätze und Definitionen kann man ebenfalls verwalten:

Definition 3.1 Ein Graph heißt planar, wenn er sich kreuzungsfrei in die Ebene einbetten lässt.

Satz 3.1 (Satz von Euler) In einem zusammenhängenden planaren Graph $G = (V, E)$ mit $|V| = n$, $|E| = m$ und f Anzahl der Facetten gilt für jede seiner planaren Einbettungen

$$n - m + f = 2.$$

Beweis. Nicht ganz trivial. □

Lemma 3.2 Sei G ein planarer, einfacher Graph mit $n \geq 3$ Knoten. $d_{max}(G)$ bezeichne den Maximalgrad in G und n_i die Anzahl der Knoten in G mit Grad i , $0 \leq i \leq d_{max}(G)$. Dann gilt

$$6 \cdot n_0 + 5 \cdot n_1 + 4 \cdot n_2 + 3 \cdot n_3 + 2 \cdot n_4 + n_5 \geq n_7 + 2 \cdot n_8 + 3 \cdot n_9 + \dots + (d_{max}(G) - 6) \cdot n_{d_{max}} + 12.$$

Algorithmus 1 zeigt, wie man Algorithmen mit deutschen Schlüsselwörtern setzen kann. Man kann sogar Referenzen auf einzelne Zeilen setzen, z.B. Zeile 6.

Algorithmus 1 : FACTORIAL(n)

Vorbedingung : $n \in \mathbb{N}$

```
1 wenn  $n < 0$  dann
2   └ zurück Fehler // Dumm gelaufen!
3  $r \leftarrow 1$ 
4  $k \leftarrow 1$ 
5 solange  $k < n$  wiederhole
6   └  $k \leftarrow k + 1$ 
7   └  $r \leftarrow r \cdot k$ 
8 zurück  $r$ 
```

4 Weiterführende Quellen

Da dieser Text bewusst sehr knapp gehalten ist, kann er viele Fragen sicher nicht klären. Darum sollte man sich nach den ersten „Gehversuchen“ eine ausführlichere Einführung anschauen. Das „Merkblatt“ [4], soll dabei helfen, einige typische Anfängerfehler zu vermeiden. Eine gute, online erhältliche Kurzeinführung ist [5]. Die wichtigsten WWW-Seiten zum Thema \LaTeX sind [1], [2] und [3]. Dort findet man unter anderem auch [6]. Eine weitere brauchbare Kurzreferenz ist [7]. Zu den Standardwerken in Buchform gehören [8], [9] und [10].

Literatur

- [1] DANTE, Deutschsprachige Anwendervereinigung TeX e.V. dante.de 4
- [2] CTAN - Comprehensive TeX Archive Network. ctan.org 4
- [3] TeX Users Group. tug.org 4
- [4] A. Wolff, Tipps fürs wissenschaftliche Schreiben mit \LaTeX . 2004. [PDF](#) 4
- [5] W. Schmidt, J. Knappen, H. Partl und I. Hyna, *$\text{\LaTeX} 2_{\epsilon}$ -Kurzbeschreibung*, 2003. [PDF](#) 4
- [6] Mark Trettin, *Das $\text{\LaTeX} 2_{\epsilon}$ -Sündenregister*, 2004. [PDF](#) 4

- [7] Hypertext Help with LaTeX. www.giss.nasa.gov/latex 4
- [8] Helmut Kopka, *LaTeX Einführung*. Addison-Wesley 4
- [9] L. Lamport, *Das LaTeX-Handbuch*. Addison-Wesley, 1994. Auf Deutsch oder Englisch erhältlich. 4
- [10] M. Goossens, F. Mittelbach und A. Samarin, *Der LaTeX-Begleiter*. Addison Wesley, 1996. Auf Deutsch oder Englisch erhältlich. 4
- [11] MiKTeX. miktex.org 2
- [12] fpTeX. fptex.org 2
- [13] TeXnicCenter. texniccenter.org 2
- [14] WinEdit. winedit.com 2
- [15] LyX. lyx.org 2
- [16] Acrobat Reader. adobe.com 2
- [17] Ghostscript. www.cs.wisc.edu/~ghost 2
- [18] AMS – Document Viewers. www.ams.org/publications/viewers.html 2
- [19] Ipe. ipe.compgeom.org 2

A Quelltext (einfuehrung.tex)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Name: einfuehrung.tex
% Autor: Steffen Mecke, 2004 (Steffen.Mecke(at)kit.edu)
% Version: 1.2, 23. Apr. 2007
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Anleitung zum Anschauen (unter Linux): %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Mit den folgenden Befehlen kann die hier vorliegende
% LaTeX-Quelldatei kompiliert und das Ergebnis betrachtet werden:
%
% > latex einfuehrung.tex
% > latex einfuehrung.tex
% > xdvi einfuehrung.dvi
%
% oder
%
% > latex einfuehrung.tex
% > latex einfuehrung.tex
% > dvips -o einfuehrung.ps einfuehrung.dvi
% > gv einfuehrung.ps
%
% oder
%
% > pdflatex einfuehrung.tex
% > pdflatex einfuehrung.tex
% > acroread einfuehrung.pdf
%

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Erl"auterungen zum Quelltext: %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Kommentare werden durch % eingeleitet;
% sie werden beim "Ubersetzen nicht ber"ucksichtigt.

% Jedes Dokument beginnt mit einem \documentclass-Befehl.
% Hier wird als Klasse ‘‘article’’ mit der Option ‘‘a4paper’’ gew"ahlt.
\documentclass[a4paper]{article}

\usepackage{a4wide} % vermindert die Seiten"ander

% In deutschen Texten werden Abs"atze "ublicherweise anders dargestellt als in
% englischen, n"amlich, statt durch eine Einr"uckung (indent) der ersten Zeile
% durch zus"atzlichen Platz (skip) zwischen zwei Abs"atzen. So wird diese Art
```

```

% der Absatzgestaltung eingestellt:
\setlength\parindent{Opt}
\setlength\parskip{\medskipamount}

% fuer Fortgeschrittene: statt article kann man (fuer deutsche Texte) auch
% scrartcl benutzen. Dann kann a4wide und die beiden \setlength-Befehle
% wegfallen!
% Eine andere M"oglichkeit ist report bzw. screpr

% Bevor es richtig los geht, folgen jetzt einige spezielle Befehle. Beim
% ersten Lesen kann man diese getrost "uberspringen und direkt zu
% \begin{document} gehen.

% entfernt man die Kommentare an den beiden folgenden Zeilen, kann man im
% Quelltext Umlaute direkt verwenden:
% \usepackage[latin1]{inputenc}
% \usepackage[T1]{fontenc}

% Mit dem \usepackage-Befehl werden zus"atzliche Pakete eingebunden,
% die bestimmte Befehle bereitstellen:
\usepackage{amsmath, amssymb} % unentbehrliche mathematische Symbole

% F"ur S"atze und Beweise
\usepackage[thmmarks,amsmath]{ntheorem}
% Definiert eine theorem-Umgebung mit "Uberschrift ‘Satz’
\newtheorem{theorem}{Satz}[section]
% Definiert eine theorem-Umgebung mit "Uberschrift ‘Lemma’. Diese
% benutzt den selben Z"ahler wie theorem.
\newtheorem{lemma}[theorem]{Lemma}
% definiert eine theorem-Umgebung mit "Uberschrift ‘Definition’. Diese
% benutzt ihren eigenen Z"ahler.
\newtheorem{definition}{Definition}[section]
% and finally: proofs
\theoremstyle{nonumberplain}
\theoremsymbol{\ensuremath{\Box}}
%\theoremsymbol{\rule{1ex}{1ex}}
\theorembodyfont{\upshape}
\theoremseparator{.}
\newtheorem{Beweis}{Beweis}

% Auch Paketen k"onnen Optionen mitgegeben werden (hier ‘german’).
\usepackage[german]{babel} % n"utzlich f"ur deutsche (oder sonstige
% nicht-englische) Texte. Erm"oglicht zum
% Beispiel , deutsche Umlaute durch "a, "o, "u
% statt \a \o \u zu setzen oder deutsche
% Anf"uhrungsstriche durch ".

\usepackage{moreverb} % f"ur das \verbatiminput-Kommando; wird meistens
% nicht gebraucht

```

```

\usepackage{url} % zum Setzen von URLs

% Die folgenden Befehle helfen dabei, wahlweise Postscript oder PDF-Dateien zu
% erzeugen. Manche Pakete verlangen n"amlich verschiedenen Optionen je nachdem,
% welche Art von Datei erzeugt werden soll.
\newif\ifpdf
\ifx\pdfoutput\undefined
\pdffalse
\else
\pdftrue
\fi
\ifpdf
\pdfoutput=1
\usepackage[pdftex]{graphicx} % zum Einbinden von Bildern
\else
\usepackage{graphicx} % zum Einbinden von Bildern
\fi

\ifpdf
% n"utzlich um Querverweise in Dokumenten zu erm"oglichen
\usepackage[pdftex, pdfpagelabels, colorlinks, citecolor=blue, backref]{hyperref}
\else
\usepackage[hypertex, naturalnames]{hyperref}
\renewcommand{\href}[2]{\url{#1}}
% alternativ k"onnte man stattdessen auch
% \usepackage[hypertex, pdfpagelabels, colorlinks, citecolor=blue, backref,
%           pagebackref]{hyperref}
% verwenden.
\fi

% Ein Paket zum Satz von Pseudocode; unterst"utzt mehrere Sprachen
% siehe algorithm2e.pdf f"ur weitere Dokumentation
\usepackage[algo2e, german, ruled, vlined]{algorithm2e}
\SetKwInOut{REQUIRE}{Vorbedingung}
\SetKwFor{solange}{solange}{wiederhole}{Ende}
\dontprintsemicolon

% einige n"utzliche Abk"urzungen
\newcommand{\N}{\ensuremath{\mathbb{N}}}
\newcommand{\R}{\ensuremath{\mathbb{R}}}
\newcommand{\Z}{\ensuremath{\mathbb{Z}}}
\newcommand{\C}{\ensuremath{\mathbb{C}}}
\newcommand{\Q}{\ensuremath{\mathbb{Q}}}
\newcommand{\cP}{\ensuremath{\mathcal{P}}\xspace}
\newcommand{\cNP}{\ensuremath{\mathcal{NP}}\xspace}

% Hier endet die so genannte Pr"aaambel und es beginnt das eigentliche
% Dokument.
\begin{document}
% Titel und Autoren des Dokumentes

```

```

\title{Ein allererstes \LaTeX-Dokument} \author{Steffen
  Mecke\thanks{Universit"at Karlsruhe, \texttt{Steffen.Mecke(at)kit.edu}}
  \and Alter Ego}
\date{\today}

\maketitle %erzeugt den Titel

% Die abstract-Umgebung dient dazu, eine Zusammenfassung darzustellen.
\begin{abstract}
  Dieses Dokument soll als Anleitung f"ur die allerersten Schritte zur
  Erstellung einer Ausarbeitung oder eines Artikels mit \LaTeX\ dienen. Der
  Quelltext (siehe Anhang~\ref{sec:source}) enth"alt weitere wichtige
  Hinweise.
\end{abstract}

% Abschnitte werden durch \section, \subsection usw. gekennzeichnet. LaTeX
% k"ummert sich dann automatisch um die Nummerierung und vieles mehr.
\section{Einleitung}
\label{sec:einl} % auf diese Marke kann man sich sp"ater im Text beziehen

\TeX\ (sprich: "‘tech"’) ist ein Textsatzsystem, das von Prof. Donald E.
Knuth entwickelt wurde. \LaTeX{} (sprich "‘latech"’) ist ein in den Jahren
1982-1986 von Leslie Lamport geschriebenes Makropaket f"ur \TeX. Beide
sind dazu gedacht (wissenschaftliche) Texte professionell zu
ver"offentlichen. Diese Datei (\texttt{einfuehrung.tex}) soll dazu
dienen, die wichtigsten Konzepte vorzustellen, die zur Erstellung eines
eigenen \LaTeX-Dokumentes n"otig sind. Abschnitt~\ref{sec:tech} stellt die
technischen Voraussetzungen und die Vorgehensweise vor. In
Abschnitt~\ref{sec:algo} werden ein paar wichtige Befehle und Umgebungen
kurz vorgestellt. Weiterf"uhrende Literatur wird im letzten Abschnitt
besprochen.

\section{Von der Idee zum Dokument}
\label{sec:tech}

Folgende Dinge braucht man f"ur das Arbeiten mit \LaTeX:

% Die itemize-Umgebung stellt Aufz"ahlungen dar
\begin{itemize}
\item Einen Texteditor. Unter Linux z.B. \texttt{emacs}, \texttt{vi},
  \texttt{kate} oder \texttt{kile}, unter Windows z.B. WinEdit
  (\cite{winedit}), % \cite ist fr Literaturhinweise gedacht. An dieser
  % Stelle wird ein Verweis auf einen der
  % \bibitem-Eintraege am Ende dieser Datei
  % eingefgt. Diese koennen auch vom bibtex-Programm
  % "‘automatisch"’ aus einer .bib-Datei erzeugt werden.
  TeXnic (\cite{texnic}), eigentlich eine komplette Entwicklungsumgebung
  f"ur \LaTeX), Emacs oder, wenn es sein muss, Notepad. LyX (\cite{lyx})
  gibt es sogar f"ur Linux, Mac und Windows. Officesysteme wie Word oder
  Staroffice sind weniger geeignet, da sie nicht dazu gedacht sind, reine
  Textdateien zu erzeugen und zu bearbeiten.
\item Eine \LaTeX-Installation. Diese ist bei praktisch allen
  Linux-Distributionen bereits dabei (z.B.~\texttt{tetex}). F"ur Windows gibt

```

es unter anderen MiKTeX (`\cite{miktex}`) oder fpTeX (`\cite{fptex}`).

- \item Ein Programm zur Betrachtung von dvi- und/oder Postscript- und/oder pdf-Dateien. F"ur Linux sind das etwa `\texttt{xdvi}`, `\texttt{gv}` und `\texttt{acroread}` oder `\texttt{xpdf}`. F"ur Windows gibt es GS View (`\cite{ghostscript}`) und den Acrobat Reader (`\cite{acrobat}`). Siehe auch `\cite{ams_viewers}`.

\end{itemize}

Einige der empfohlenen Editoren haben extra \LaTeX-Modi, mit "syntax highlighting" und speziellen Befehlen f"ur das Erstellen, "Ubersetzen und Betrachten von \LaTeX-Dateien, die die Arbeit sehr erleichtern.

Um Bilder zu erzeugen, die in einem Dokument verwendet werden sollen, braucht man noch ein geeignetes Zeichenprogramm. Es sollte Bilder im Postscript-Format (`\texttt{.eps}`) und/oder im pdf-Format (`\texttt{.pdf}`) schreiben k"onnen. Geeignet sind z.B. `\texttt{xfig}` (f"ur Linux) oder `\texttt{ipe}` (`\cite{ipe}`, f"ur Linux oder Windows).

Es gibt noch eine Reihe weiterer Werkzeuge, die die Arbeit mit \LaTeX{} erleichtern, wie zum Beispiel Bib\TeX. Aber f"ur den Anfang sollten diese erst mal reichen.

Folgende Schritte sind notwendig:

```
% Die enumerate-Umgebung ist f"ur nummerierte Aufz"ahlungen da
\begin{enumerate}
\item Die Eingabedatei mit dem Editor schreiben
      (z.B. ~\texttt{einfuehrung.tex}).
\item Die Eingabedatei kompilieren. Unter Linux geht das mit dem Befehl\
      \verb|# latex einfuehrung|\
      und \
      \verb|# dvips einfuehrung.dvi|\
      zur Erzeugung einer Postscript-Datei, bzw. mit\
      \verb|# pdflatex einfuehrung.tex|\
      zur Erzeugung einer pdf-Datei. Manchmal muss man dies zwei mal
      hintereinander machen, da Querverweise (\verb|\ref{}|,\verb|\cite{}|) oft
      erst beim zweiten Mal richtig gesetzt werden.
\item Falls ein Fehler in der Eingabedatei war, h"alt \verb|latex| an
      dieser Stelle an und wartet auf eine Eingabe. Man kann dann entweder
      versuchen, durch Eingabe von \verb|<Enter>| einfach weiterzumachen oder
      durch Eingabe von \verb|x| abbrechen.
\item Das Ergebnis anschauen. Unter Linux -- je nachdem, was f"ur eine Art von
      Datei erzeugt wurde -- mit einem der drei folgenden Kommandos:\
      \verb|# xdvi einfuehrung.dvi| \
      \verb|# gv einfuehrung.ps| \
      \verb|# acroread einfuehrung.pdf|
\item Eventuelle (Tipp-)Fehler in der Eingabe korrigieren, erneut
      kompilieren und so weiter \dots
\end{enumerate}

\section{Elemente eines \LaTeX-Dokuments}
\label{sec:algo}
```

Ein `\LaTeX`-Dokument besteht aus einfachem Text und speziellen `\LaTeX`-Kommandos. Das Grundgerüst sieht so aus:

```
\begin{verbatim}
\documentclass{article}
% Praeambel
\usepackage{Paket1, Paket2}
% ...
\begin{document}
  Text
\end{document}
\end{verbatim}
```

Normaler Text wird einfach getippt, wie er ist. `\TeX` kümmert sich automatisch um die richtigen Abstände zwischen den Wörtern
% ein Zeilenumbruch wirkt wie ein Leerzeichen
(egal ob dort ein oder
% mehrere Leerzeichen wirken wie ein Leerzeichen
mehrere Leerzeichen stehen), Zeilen- und Seitenumbrüche und
viele mehr. Vor Satzzeichen sollte man jedoch kein Leerzeichen lassen, das
sieht sonst nicht gut aus. Absätze kennzeichnet man durch mindestens eine
Leerzeile.

`\TeX`-Kommandos werden durch den Backslash (`\backslash`) eingeleitet. Sie beziehen sich in der Regel auf das nächste Zeichen oder ihr Argument, das gewöhnlich in geschweifte Klammern eingeschlossen wird, etwa `\verb+\texttt{emacs}+` (erzeugt `\texttt{emacs}`). Dann gibt es noch `\emph{Umgebungen}`, etwa die `\texttt{itemize}`-Umgebung. Ihr Argument ist alles, was zwischen `\verb|\begin{itemize}|` und `\verb|\end{itemize}|` steht. Viele Kommandos sind nicht automatisch vorhanden, sondern werden durch besondere `\emph{Pakete}` bereitgestellt, die erst mit dem `\verb+\usepackage{+}`-Befehl geladen werden müssen.

Will man hinter einem Befehl unbedingt ein Leerzeichen einfügen, kann man dies etwa wie folgt erreichen:

```
\verb+\LaTeX\ ist schon+\
```

Manchmal will man an einen Zeilenumbruch an einer bestimmten Stelle verhindern (etwa `\mbox{Kapitel} 1`). Das kann man das mit Hilfe der Tilde tun: `\verb|Kapitel~1|`. Umlaute ("a"o"u" usw.) erzeugt man entweder durch die Kommandos `\verb|"a"o"u"s|` oder -- falls wie hier das `\texttt{babel}`-Paket mit der Option `\texttt{german}` verwendet wird -- durch `\verb|"a"o"u"s|`. Eine `\textbf{fette Schriftart}` wählt man mit dem Befehl `\verb|\textbf{fetter Text}|` aus. Damit sollte man jedoch sehr sparsam umgehen. `\emph{Wichtige Begriffe}` sollte man lieber mit `\verb|\emph{Wichtig}|` kursiv setzen.

Mit dem Befehl `\verb|\includegraphics[Optionen]{Dateiname_ohne_Endung}|` bindet man ein Bild ein. Um Postscript Dateien zu erzeugen sollten die Bilder im (Encapsulated-)Postscript-Format (`\texttt{.eps}`) vorliegen, für PDF-Dateien sollten auch die Bilder in dieses Format (`\texttt{.pdf}`) haben. Ein Bild sollte immer in eine `\texttt{figure}`-Umgebung gesetzt werden, damit man ihm eine Bildunterschrift und einen Bezeichner (`\verb|\label{fig:beispiel2}|`) geben und sich später darauf beziehen kann (hier mit `\verb|\ref{fig:beispiel2}|`). Die optionalen Parameter

`\verb|http|` bewirken, dass die Abbildung automatisch an eine Stelle im Text "flie"st", wo sie "asthetisch sch"on wirkt. Das funktioniert auch meistens gut! Das Beispiel in Abbildung~\ref{fig:beispiel} wurde mit dem Zeichenprogramm `\texttt{tipe}` erstellt~-- beachten Sie die Beschriftung!

```
\begin{verbatim}
\begin{figure}[htbp]
  \centering
  \includegraphics{fig/beispiel}
  \caption{Eine Beispielabbildung.}
  \label{fig:beispiel1}
\end{figure}
\end{verbatim}
```

```
\begin{figure}[htbp]
  \centering
  \includegraphics{fig/beispiel}
  \caption{Eine Beispielabbildung.}
  \label{fig:beispiel}
\end{figure}
```

Eine Formel wie in Abbildung~\ref{fig:beispiel} wird mit der Zeichenfolge `\verb|$\sqrt{\sum_{i=1}^n x_i}$|` produziert: $\sqrt{\sum_{i=1}^n x_i}$. Man unterscheidet zwischen Formeln im Flie"stext, die zwischen `\$`-Zeichen eingeschlossen werden und Formeln, die alleine in einer Zeile stehen und mit `\verb|[\[\dots \verb|]|` gekennzeichnet werden, z.B.
% Anmerkung: `$$` sollte nicht statt `\[` verwendet werden.

```
\[
g(x) := \sqrt{\sum_{i=1}^n x_i} \quad \forall x \in \mathbb{R}_{\geq 0}
\]
```

Au"serdem gibt es noch nummerierte Formeln (wie~(\ref{eq:forces})), auf die man sich sp"ater im Text beziehen kann mit `\verb|\ref{eq:forces}|`. Sie werden mit einer `\texttt{equation}`-Umgebung erzeugt, etwa

```
\begin{equation}
  \label{eq:forces}
  f_{\mathrm{rep}}(p_u, p_v) := \frac{c_{\rho}}{\|p_v - p_u\|^2} \cdot
  \overrightarrow{p_u p_v}
\end{equation}
```

S"atze und Definitionen kann man ebenfalls verwalten:

```
\begin{definition}\label{definition:planar}
  Ein Graph hei"st \emph{planar}, wenn er sich kreuzungsfrei in die Ebene
  einbetten l"asst.
\end{definition}
```

```
\begin{theorem}[Satz von Euler] \label{satz:euler}
  In einem zusammenhngenden planaren Graph  $G=(V,E)$  mit  $|V|=n$ ,  $|E|=m$ 
  und  $f$  Anzahl der Facetten gilt fr jede seiner planeren Einbettungen
  \[
  \%
  n-m+f=2.
  \]
```

```
\end{theorem}
\begin{Beweis}
  Nicht ganz trivial.
```

```

\end{Beweis}
\begin{lemma}\label{satz:euler2}
  Sei  $G$  ein planarer, einfacher Graph mit  $n \geq 3$  Knoten.  $d_{\max}(G)$ 
  bezeichne den Maximalgrad in  $G$  und  $n_i$  die Anzahl der Knoten in  $G$ 
  mit Grad  $i$ ,  $0 \leq i \leq d_{\max}(G)$ . Dann gilt
  \[
  6 \cdot n_0 + 5 \cdot n_1 + 4 \cdot n_2 + 3 \cdot n_3 + 2 \cdot n_4 + n_5 \geq \dots
  n_7 + 2 \cdot n_8 + 3 \cdot n_9 + \dots + (d_{\max}(G) - 6) \cdot n_{d_{\max}} + 12.
  \]
\end{lemma}

```

Algorithmus~\ref{alg:faculty} zeigt, wie man Algorithmen mit deutschen Schl"usselw"ortern setzen kann. Man kann sogar Referenzen auf einzelne Zeilen setzen, z.B. Zeile~\ref{alg:someline}.

```

\linesnumbered
\begin{algorithm2e}[htbp]
  \REQUIRE{ $n \in \mathbb{N}$ }
  \Wenn{ $n < 0$ }{\Zurueck Fehler \tcp{ Dumm gelaufen!} }
   $r \leftarrow 1$ ;
   $k \leftarrow 1$ ;
  \solange{ $k < n$ }
  {
     $k \leftarrow k + 1$ ; \nllabel{alg:someline}
     $r \leftarrow r \cdot k$ ;
  }
  \Zurueck{ $r$ }
\caption{\textsc{Factorial}( $n$ )}
\label{alg:faculty}
\end{algorithm2e}

```

\section{Weiterf"uhrende Quellen}

Da dieser Text bewusst sehr knapp gehalten ist, kann er viele Fragen sicher nicht kl"aren. Darum sollte man sich nach den ersten "Gehversuchen" eine ausf"uhrlichere Einf"uhrung anschauen. Das "Merkblatt" \cite{wolff}, soll dabei helfen, einige typische Anf"angerfehler zu vermeiden. Eine gute, online erh"altliche Kurzeinf"uhrung ist \cite{schmidt_latex_kurz}. Die wichtigsten WWW-Seiten zum Thema \LaTeX\ sind \cite{dante}, \cite{CTAN} und \cite{tug}. Dort findet man unter anderem auch \cite{trettin_latex_tod}. Eine weitere brauchbare Kurzreferenz ist \cite{nasa}. Zu den Standardwerken in Buchform geh"oren \cite{kopka_latex}, \cite{lampport_latex} und \cite{gossens_begleiter}.

```

% Die einfachste Art ein Literaturverzeichnis anzulegen:
\begin{thebibliography}{XY} %% f"ur die Nummerierung wird so viel Platz
  %% reserviert, wie das zweite Argument -- hier
  %% XY -- ben"otigt.
\bibitem{dante}{DANTE, Deutschsprachige Anwendervereinigung TeX e.V.
  \href{http://www.dante.de/}{dante.de}}

```

```

\bibitem{CTAN}{CTAN - Comprehensive TeX Archive Network.
  \href{http://www.ctan.org/}{ctan.org}}

\bibitem{tug}{TeX Users Group. \href{http://www.tug.org}{tug.org}}

\bibitem{wolff}{A. Wolff, Tipps f"urs wissenschaftliche Schreiben mit
  \LaTeX. 2004.
  \href{http://i11www.ilkd.uni-karlsruhe.de/~awolff/lehre/scientific_writing/writing_tips.pdf}%
  {PDF}}

\bibitem{schmidt_latex_kurz} W. Schmidt, J. Knappen, H. Partl und I. Hyna,
  \emph{\LaTeXe-Kurzbeschreibung}, 2003.\
  \href{http://www.ctan.org/tex-archive/info/german/LaTeX2e-Kurzbeschreibung/l2kurz2.pdf}{PDF}

\bibitem{trettin_latex_tod} Mark Trettin,
  \emph{Das \LaTeXe-S"undenregister}, 2004.\
  \href{l2tabu.pdf}{PDF}

\bibitem{nasa}{Hypertext Help with LaTeX.
  \href{http://www.giss.nasa.gov/latex/}{www.giss.nasa.gov/latex}}

\bibitem{kopka_latex}{Helmut Kopka, \emph{\LaTeX\ Einf"uhrung}. Addison-Wesley}

\bibitem{lamport_latex}{L. Lamport, \emph{Das \LaTeX-Handbuch}.
  Addison-Wesley, 1994. Auf Deutsch oder Englisch erh"altlich.}

\bibitem{gossens_begleiter}{M. Goossens, F. Mittelbach und A. Samarin,
  \emph{Der \LaTeX-Begleiter}. Addison Wesley, 1996. Auf Deutsch oder
  Englisch erh"altlich.}

\bibitem{miktex}{MiKTeX. \href{http://www.miktex.org/}{miktex.org}}

\bibitem{fptex}{fpTeX. \href{http://www.fptex.org/}{fptex.org}}

\bibitem{texnic}{TeXnicCenter.
  \href{http://www.texniccenter.org/}{texniccenter.org}}

\bibitem{winedit}{WinEdit. \href{http://www.winedit.com/}{winedit.com}}

\bibitem{lyx}{LyX. \href{http://www.lyx.org}{lyx.org}}

\bibitem{acrobat}{Acrobat Reader.
  \href{http://www.adobe.com/products/acrobat/}{adobe.com}}

\bibitem{ghostscript}{Ghostscript.
  \href{http://www.cs.wisc.edu/~ghost/}{www.cs.wisc.edu/~ghost}}

\bibitem{ams_viewers}{AMS -- Document
  Viewers. %
  \href{http://www.ams.org/publications/viewers.html}%
  {www.ams.org/publications/viewers.html}}

\bibitem{ipe}{Ipe. \href{http://ipe.compgeom.org/}{ipe.compgeom.org}}

```

```
\end{thebibliography}

\newpage
\appendix

\section{Quelltext (\texttt{einfuehrung.tex})}
\label{sec:source}

\verbatiminput{einfuehrung.tex}

\end{document} % Alles was unterhalb dieser Zeile steht, wird von TeX
                % einfach ignoriert
```