

Computereinsatz in der Mathematik

Eberhard Luik

SS 2019

1 Einführung in Latex

1.1 Einleitung

1.1.1 Was ist Latex

Latex (auch mit der Schrift \LaTeX dargestellt) ist ein äußerst flexibles, umfangreiches Programmpaket zur Erstellung von Dokumenten in Buchdruckqualität und bestens geeignet für mathematisch-naturwissenschaftliche Texte. Es ist für alle wichtigen Betriebssysteme (Windows, Mac, Linux) frei verfügbar. Für mathematische Texte ist Latex inzwischen internationaler Standard.

1.1.2 Die Entstehung von Latex

Mit der Verbreitung der Computer entstand der Wunsch, damit auch Texte mit komplizierten (mathematischen) Formeln zu verarbeiten. Zu diesem Zweck entwickelte Donald E. Knuth Ende der siebziger Jahre das Textsatzsystem **TeX** (Tau Epsilon Chi). Die Anwendung dieses Systems ist aber kompliziert und erfordert Kenntnisse im Buchdruck, um dem Text ein ansprechendes Erscheinungsbild zu verleihen.

Vor diesem Hintergrund entwickelte Laslie Lamport Anfang 1980 das Paket **Latex**, das dem Anwender diese drucktechnischen Details abnimmt. Inzwischen gibt es zu Latex viele Zusatzpakete (sogenannte *usepackages*), z.B. für mathematische Symbole, viele Schriftarten oder um Graphiken bzw. digitale Bilder einzubinden.

1.1.3 Wie funktioniert Latex

Im Gegensatz zu anderen bekannten Textverarbeitungssystemen wie z.B. Word arbeitet Latex nicht nach dem **wysiwyg**-Prinzip (**what you see is what you get**). Das bedeutet, dass das gewünschte Erscheinungsbild des Textes bei der Eingabe nicht sofort am Bildschirm angezeigt wird.

Vielmehr müssen Text und Formatierungskommandos mit einem Editor zunächst in eine Datei eingegeben werden. Danach erzeugt das Latex-Formatierungsprogramm eine weitere Datei mit dem gewünschten Layout, welche dann am Bildschirm betrachtet oder auf dem Drucker ausgegeben werden kann. Entsprechend ist bei der Korrektur oder Änderung des Erscheinungsbildes vorzugehen.

Diesbezüglich arbeitet Latex wie ein Compiler bei Programmiersprachen.

1.1.4 Arbeitsschritte in Latex

Aus der oben beschriebenen Funktionsweise ergeben sich die nötigen Arbeitsschritte (unter Linux):

1. Eingabe in eine Datei

Text und Formatierungskommandos sind mit Hilfe eines Editors in eine Datei mit dem Namen `<name>.tex` einzugeben. Die Datei muss den Zusatz `.tex` haben; `<name>` ist frei wählbar.

Unter Linux kommen vor allem die Editoren **emacs** und **kile** zum Einsatz.

Der Aufruf dieser Editoren lautet:

```
kile <name>.tex    (z.B. kile seite1.tex)
emacs <name>.tex  (z.B. emacs seite1.tex).
```

Prinzipiell kann für die Eingabe jedoch jeder Editor verwendet werden, der den Text im ASCII-Format abspeichert.

2. Formatierung

Der Aufruf

```
latex <name>    (z.B. latex seite1)
```

(ohne den Zusatz `.tex`) erzeugt eine neue Datei mit dem Namen

```
<name>.dvi    (z.B. seite1.dvi),
```

welche das Erscheinungsbild unseres Textes enthält (in Maschinencode).

3. Betrachten am Bildschirm

Diese neu erzeugte Datei können wir mit dem Kommando

```
kdvi <name>    (z.B. kdvi seite1)
```

auf dem Bildschirm anschauen. Erst jetzt wird das Erscheinungsbild unseres Textes ersichtlich. Das `kdvi`-Kommando braucht die Datei `<name>.dvi`. Falls diese nicht existiert, so erfolgt eine Fehlermeldung.

In den neueren Linux-Versionen gibt es auch das Kommando

```
okular <name>.dvi    (z.B. okular seite1.dvi)
```

Sind noch Änderungen erwünscht, so müssen die Schritte 1 und 2 (in dieser Reihenfolge) wiederholt werden.

4. Ausgabe auf dem Drucker

Mit dem Befehl

```
dvipdf <name>    (z.B. dvipdf seite1)
```

wird aus der Datei `<name>.dvi` eine pdf-Datei mit dem Namen `<name>.pdf` erzeugt. Dabei steht pdf für *portable document format*. Dies ist das Standard-Format für den Austausch von Texten zwischen verschiedenen Usern bzw. verschiedenen Plattformen (Betriebssystemen).

pdf-Dateien kann man auf dem Drucker ausgeben, mit dem *Acrobat-Reader* bzw. mit dem Kommando `okular` am Bildschirm betrachten oder per Email verschicken. Damit kann auch ein anderer Benutzer unseren (mit Latex erzeugten) Text lesen, ohne das Programmpaket Latex auf seinem Rechner installieren zu müssen.

Anmerkungen:

1. Neuere Editoren (z.B. Kile oder Emacs) erlauben es, die in 2. - 4. beschriebenen Kommandos aus dem Editor heraus zu starten.
2. Zur Formatierung gibt es auch den Befehl

```
pdflatex <name>    (z.B. pdflatex seite1),
```

welcher aus der Datei `<name>.tex` sofort die pdf-Datei `<name>.pdf` erzeugt.

1.2 Grundlagen von Latex

1.2.1 Grundstruktur eines Latex-Dokuments

Ein Latex-Dokument besteht aus einem Vorspann (Präambel) und dem eigentlichen Textteil einschließlich Kommandos für das Layout.

In der Präambel werden die Einstellungen für das Dokument vorgenommen, wie zum Beispiel die Dokumentklasse (Buch, Artikel, Brief,...), Textbreite und Texthöhe, Lage der Ränder, Art der Seitennummerierung. Außerdem werden zusätzliche Programmpakete (`\usepackages`) geladen, etwa für die deutsche Silbentrennung oder für weitere mathematische Sonderzeichen.

Eine an unsere Bedürfnisse angepasste Präambel sieht wie folgt aus:

```

\documentclass[12pt]{article}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[ngerman]{babel}
\usepackage{lmodern}
\usepackage{latexsym}
\usepackage{amsfonts}
\usepackage{amssymb}
\usepackage{amsmath}
\pagestyle{empty}
\topmargin -1cm
\textheight 25cm
\textwidth 16.0 cm
\oddsidemargin 0.2cm
\begin{document}

\end{document}

```

Eventuell müssen Sie in der dritten Zeile den Eintrag `utf8` in der eckigen Klammer durch `latin1` ersetzen. Dies hängt davon ab, welche Kodierung Ihr Editor beim Abspeichern verwendet. Werden im Erscheinungsbild des Textes die deutschen Umlaute nicht richtig dargestellt, so ist in der Regel in der Präambel die falsche Kodierung gewählt worden.

Mit dem Kommando `\pagestyle` wird die Art der Seitennummerierung festgelegt. Hier sind folgende Parameter möglich:

<code>empty</code>	Keine Seitennummerierung.
<code>plain</code>	Die Seitennummerierung erfolgt als zentrierte Fußzeile.
<code>headings</code>	Die Seitenzahl wird als Kopfzeile zusammen mit einer automatisch gewählten Überschrift-Information ausgegeben. In der Regel ist dies die augenblickliche Abschnittsüberschrift.
<code>myheadings</code>	Die Seitenzahl wird als Kopfzeile zusammen mit einer selbst zu definierenden Überschrift ausgegeben.

Der eigentliche Text (einschließlich Kommandos für das Erscheinungsbild) steht zwischen den Befehlen

```
\begin{document}
```

```
\end{document}
```

1.2.2 Steuerzeichen

Einige Zeichen werden vom Latex-Formatierer als Steuerzeichen interpretiert, so z. B.

- `\` leitet einen Latex-Befehl ein.
- `%` dient als Kommentarzeichen: der Text rechts von `%` bis zum Zeilenende wird von Latex ignoriert.
- `$` wird für die Kennzeichnung von mathematischen Formeln verwendet.
- `&` ist das Trennungszeichen für die einzelnen Spalteneinträge in Tabellen.
- `{` bzw. `}` dienen zur Markierung von Blöcken.

Sollen diese Zeichen im Text erscheinen, so sind sie mit `\backslash`, `\%`, `\$`, `\&`, `\{` bzw. `\}` einzugeben.

1.2.3 Text- und Mathematik-Modus

Latex unterscheidet zwischen einem Text-Modus (Voreinstellung) und einem Mathematik-Modus (für mathematische Symbole und Formeln). Im professionellen Buchdruck haben die Buchstaben a bzw. A und die mathematischen Größen a bzw. A ein unterschiedliches Erscheinungsbild.

Das Umschalten in den Mathematik-Modus erfolgt

1. im laufenden Text durch `$... $`. Die Zeichen zwischen den Dollarzeichen werden im Layout als mathematische Symbole dargestellt.
2. für eine gesonderte Formelzeile durch `\[... \]` bzw. `$$... $$`.

Latex-Befehle für mathematische Symbole funktionieren nur im Mathematik-Modus. Das folgende kleine Beispiel (ohne Präambel) demonstriert das Umschalten in den Mathematik-Modus. Die Texteingabe

```
Im professionellen Buchdruck haben die Buchstaben a bzw. A
und die mathematischen Größen $a$ bzw. $A$ ein unterschiedliches
Erscheinungsbild.
```

```
\vspace{1ex} \newline
```

```
Umfangreiche Formeln, wie zum Beispiel
```

```
\[
```

```
K := \{ (a,b) \in \mathbb{R}^2 \mid : \mid : a^2 + b^2 = 1 \}
```

```
\]
```

```
werden in eine eigene Formelzeile gesetzt.
```

liefert das Layout

Im professionellen Buchdruck haben die Buchstaben a bzw. A und die mathematischen Größen a bzw. A ein unterschiedliches Erscheinungsbild.

Umfangreiche Formeln, wie zum Beispiel

$$K := \{(a, b) \in \mathbb{R}^2 : a^2 + b^2 = 1\}$$

werden in eine eigene Formelzeile gesetzt.

1.2.4 Zeilen- und Seitenumbruch

Zeilen- und Seitenumbruch sowie die Seitennummerierung erfolgen in Latex automatisch. Durch das Kommando

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

in der Präambel wird beim Zeilenumbruch die deutsche Silbentrennung verwendet. Dabei treten jedoch vereinzelt Fehler auf. So wird beispielsweise das Wort Strichoperationen falsch getrennt: Stri-choperationen. Für eine korrekte Silbentrennung muss man in diesem Fall dem Latex-Formatierer die Trennung bei der Texteingabe vorschreiben: Strich\ -operationen.

Das Kommando `\newline` bzw. `\\` bewirkt eine neue Zeile im Layout. Eine neue Zeile bei der Texteingabe hat keinen Einfluss auf den Zeilenumbruch im Layout.

Dagegen bewirkt eine Leerzeile im Eingabetext einen neuen Absatz mit einem vergrößerten Zeilenabstand, wobei das erste Wort eingerückt wird. Mit dem Befehl `\noindent` verhindert man das Einrücken.

Das Kommando `\newpage` erzwingt eine neue Seite.

1.2.5 Abstände im Text

Latex wählt die Abstände zwischen den Worten, zwischen den Zeilen und in den Formeln automatisch. Es gibt jedoch einige Befehle, um hierauf Einfluss zu nehmen.

Horizontale Abstände

Am flexibelsten ist das Kommando `\hspace{<abstand>}`. Dabei besteht `<abstand>` aus einer Dezimalzahl und einer Maßeinheit. Als Maßeinheit sind u. A. erlaubt:

absolute Maßeinheiten	relative Maßeinheiten
cm Zentimeter	em Breite von m
mm Millimeter	ex Höhe von x

Relative Maße haben den Vorteil, dass bei Änderung der Schriftgröße die Abstände ebenfalls angepasst werden.

Weitere Möglichkeiten für (horizontale) Abstände sind

`\,` verkürzte Leerstelle
`\quad` vergrößerter Abstand
`\qquad` stark vergrößerter Abstand

Beispiele

Eingabe	Layout	Eingabe	Layout
<code>abcd efgh</code>	abcd efgh	<code>abcd\,efgh</code>	abcd efgh
<code>abcd\quad efgh</code>	abcd efgh	<code>abcd\hspace{1cm}efgh</code>	abcd efgh
<code>abcd\qquad efgh</code>	abcd efgh	<code>abcd\hspace{1em}efgh</code>	abcd efgh

Vertikale Abstände

Für vertikale Abstände zwischen den einzelnen Zeilen gibt es analog den Befehl

`\vspace{<abstand>}`.

Dieses Kommando bewirkt, dass beim nächsten Zeilenumbruch ein entsprechender Abstand eingefügt wird.

Für Abstände zwischen den einzelnen Absätzen gibt es die Befehle `\smallskip` (kleiner Abstand), `\medskip` (mittlerer Abstand) und `\bigskip` (großer Abstand).

Anmerkung: Fällt im Erscheinungsbild das `hspace`-Kommando auf das Zeilenende bzw. auf den Zeilenanfang, so wird es ignoriert. Analoges gilt für das `vspace`-Kommando am Seitenende bzw. Seitenanfang. Ist hier trotzdem ein bestimmter Abstand gewünscht, so wird

`\hspace*{<abstand>}` bzw. `\vspace*{<abstand>}`

verwendet.

1.2.6 Schriftarten und Schriftgrößen

Für die Wahl des Schriftbildes gibt es standardmäßig folgende Möglichkeiten:

Name	Kommando	Layout
roman	<code>\rm</code>	ergibt dieses Schriftbild
bold face	<code>\bf</code>	ergibt dieses Schriftbild
slanted	<code>\sl</code>	<i>ergibt dieses Schriftbild</i>
italic	<code>\it</code>	<i>ergibt dieses Schriftbild</i>
typewriter	<code>\tt</code>	ergibt dieses Schriftbild
sans serif	<code>\sf</code>	ergibt dieses Schriftbild
small caps	<code>\sc</code>	ERGIBT DIESES SCHRIFTBILD

Beispiel: Die Eingabe

Voreingestellt ist die Schriftart `{\bf roman}`. Sie wird so lange verwendet, bis ein Befehl für eine andere Schriftart erfolgt. Wird ein neue Schriftart in einem mit geschweiften Klammern markierten Block gewählt, so gilt diese Änderung nur in diesem Block.

`\newline`

`\sc` Ab jetzt wird auf die Schriftart `{\tt small caps}` umgeschaltet.

erzeugt das Schriftbild

Voreingestellt ist die Schriftart **roman**. Sie wird so lange verwendet, bis ein Befehl für eine andere Schriftart erfolgt. Wird ein neue Schriftart in einem mit geschweiften Klammern markierten Block gewählt, so gilt diese Änderung nur in diesem Block.

AB JETZT WIRD AUF DIE SCHRIFTART **small caps** UMGESCHALTET.

In Latex gibt es viele weitere Schriftsätze, deren Aufruf jedoch komplizierter ist. Hier verweisen wir auf die Literatur.

Latex kennt folgende Schriftgrößen:

<code>\Huge</code>	Beispiel
<code>\huge</code>	Beispiel
<code>\LARGE</code>	Beispiel
<code>\Large</code>	Beispiel
<code>\large</code>	Beispiel
<code>\normalsize</code>	Beispiel
<code>\small</code>	Beispiel
<code>\footnotesize</code>	Beispiel
<code>\scriptsize</code>	Beispiel
<code>\tiny</code>	Beispiel

Voreinstellung ist `normalsize`, deren Größe im Layout in der ersten Zeile der Präambel

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

in der eckigen Klammer festgelegt wird. Hier sind `10pt`, `11pt` oder `12pt` möglich. Dabei ist `pt` (points) die Maßeinheit: $72.27 \text{ pt} = 1 \text{ inch} = 2.54 \text{ cm}$.

1.2.7 Latex-Umgebungen

Eine *Umgebung* hat die Struktur

```
\begin{<umgebung>}
  <text>
\end{<umgebung>}
```

und bewirkt, dass der innerhalb dieser Umgebung stehende Text anders behandelt wird, z.B. als Tabelle oder als mathematische Gleichung. Einige Umgebungen haben optionale und/oder zwingende Parameter:

```
\begin{<umgebung>}[<optpar>]{<zwingpar>}
```

Beispiel: Für Texte, die zentriert gesetzt werden sollen (etwa Gedichte), gibt es die `center` - Umgebung.

```
\begin{center}
Dieser Text wird \\ zentriert gesetzt;
\end{center}
```

Dieser Text wird
zentriert gesetzt;

1.3 Wichtige Elemente im Text-Modus

1.3.1 Listen

Latex stellt einige Umgebungen für Listen bereit, und zwar in Abhängigkeit von der gewünschten Darstellung der einzelnen Listenelemente.

Markierung durch ein Symbol

Hierzu gibt es die `itemize` - Umgebung:

```
\begin{itemize}
  \item <text>
  \item <text>
  :
\end{itemize}
```

Diese Umgebung darf auch geschachtelt werden. Es werden dann unterschiedliche Markierungssymbole verwendet, wie das folgende Beispiel zeigt:

<pre>\begin{itemize} \item Punkt 1 \item Punkt 2 \begin{itemize} \item Unterpunkt 1 \item Unterpunkt 2 \begin{itemize} \item Unterpunkt 2.1 \item Unterpunkt 2.2 \end{itemize} \end{itemize} \end{itemize} \item Punkt 3 \end{itemize}</pre>	<ul style="list-style-type: none"> • Punkt 1 • Punkt 2 <ul style="list-style-type: none"> – Unterpunkt 1 – Unterpunkt 2 <ul style="list-style-type: none"> * Unterpunkt 2.1 * Unterpunkt 2.2 • Punkt 3
--	---

Anmerkung: Zur besseren Lesbarkeit empfehlen wir, den Text innerhalb einer Latex-Umgebung etwas einzurücken (dies ist insbesondere bei der Fehlersuche von Vorteil). Für den Latex-Formatierer ist ein Einrücken nicht erforderlich; er kommt auch damit zurecht:

```
\begin{itemize}\item Punkt 1\item Punkt 2\begin{itemize}\item Unterpunkt 1
\item Unterpunkt 2\begin{itemize}\item Unterpunkt 2.1\item Unterpunkt 2.2
\end{itemize}\end{itemize}\item Punkt 3\end{itemize}
```

Nummerierung der Listenpunkte

Für nummerierte Aufzählungen gibt es die `enumerate` - Umgebung.

```
\begin{enumerate}
  \item <text>
  \item <text>
  :
\end{enumerate}
```

Beispiel

```
\begin{enumerate}
  \item Wintersemester
    \begin{enumerate}
      \item Analysis I
      \item Lineare Algebra I
    \end{enumerate}
  \item Sommersemester
    \begin{enumerate}
      \item Analysis II
      \item Lineare Algebra II
      \item Coma
    \end{enumerate}
\end{enumerate}
```

1. Wintersemester
 - (a) Analysis I
 - (b) Lineare Algebra I
2. Sommersemester
 - (a) Analysis II
 - (b) Lineare Algebra II
 - (c) Coma

Stichworte für die Listenpunkte

Eine dritte Möglichkeit ist die `description` - Umgebung. Hier werden die einzelnen Punkte durch ein fettgedrucktes Wort markiert.

```
\begin{description}
  \item [<wort1>] <text>
  \item [<wort2>] <text>
  :
\end{description}
```

Beispiel:

In dieser Vorlesung werden folgende Themen behandelt:

```
\begin{description}
  \item[Latex:] In den ersten drei Wochen geben wir eine
    kurze Einführung in das Textverarbeitungssystem Latex.
  \item[Matlab:] Der Schwerpunkt dieser Lehrveranstaltung
```

```

    liegt auf der Vermittlung der Grundlagen von Matlab.
    Diese Kenntnisse werden im nächsten Semester für die
    Vorlesung {\sc Numerik~I} benötigt.
\end{description}

```

In dieser Vorlesung werden folgende Themen behandelt:

Latex: In den ersten drei Wochen geben wir eine kurze Einführung in das Textverarbeitungssystem Latex.

Matlab: Der Schwerpunkt dieser Lehrveranstaltung liegt auf der Vermittlung der Grundlagen von Matlab. Diese Kenntnisse werden im nächsten Semester für die Vorlesung NUMERIK I benötigt.

1.3.2 Tabellen

Für Tabellen gibt es die `tabular` - Umgebung, welche folgenden formalen Aufbau hat:

```

\begin{tabular}[<pos>]{<spalten>}
  <text11> & <text12> & ... & <text1n> & \\
  <text21> & <text22> & ... & <text2n> & \\
  & & & & & \\
  & & & & & \\
\end{tabular}

```

Der **optionale** Parameter `<pos>` dient zur horizontalen Ausrichtung der Tabelle innerhalb der laufenden Textzeile:

<code>t</code> (top)	oberste Tabellenzeile ist auf dem Level der Textzeile
<code>b</code> (bottom)	unterste Tabellenzeile ist auf dem Level der Textzeile
ohne opt. Param.	Tabellenmitte ist auf dem Level der Textzeile

Der **obligatorische** Parameter `<spalten>` enthält die Anzahl und die Ausrichtung der Spalten. Für jede Spalte ist genau eines der folgenden drei Formatierungszeichen anzugeben:

<code>l</code> (left)	linksbündige Ausrichtung der betreffenden Spalte
<code>c</code> (center)	zentrierte Ausrichtung der betreffenden Spalte
<code>r</code> (right)	rechtsbündige Ausrichtung der betreffenden Spalte

Bei der Texteingabe dient `&` als Trennungzeichen für die einzelnen Spalten und `\\` als Zeilenende.

Beispiel

```

top \hspace{1em}
\begin{tabular}[t]{lcr}
  3 & 145 & -2 \\
 -12 & 1 & 35 \\
 10 & -5 & 165
\end{tabular}
\hspace{1em} bottom \hspace{1em}
\begin{tabular}[b]{lcr}
  3 & 145 & -2 \\
 -12 & 1 & 35 \\
 10 & -5 & 165
\end{tabular}
\hspace{1em} zentriert \hspace{1em}
\begin{tabular}{lcr}
  3 & 145 & -2 \\
 -12 & 1 & 35 \\
 10 & -5 & 165
\end{tabular}

```

					3	145	-2					
					-12	1	35			3	145	-2
top	3	145	-2	bottom	10	-5	165	zentriert	-12	1	35	
	-12	1	35						10	-5	165	
	10	-5	165									

Selbstverständlich kann eine Tabelle auch mit einem Rahmen versehen werden. Waagerechte Linien erhält man mit dem Befehl `\hline`, senkrechte durch Angabe von `|` im obligatorischen Parameter `<spalten>`.

Beispiel

```

\begin{center}
\begin{tabular}{|c|c|c|c|} \hline
  & Teilnehmer & bestanden & Prozent \\
\hline \hline
gesamt & 143 & 112 & 78.3 \% \\
\hline \hline
Bachelor & 86 & 69 & 80.2 \% \\
\hline \hline
Diplom/Lehramt & 57 & 43 & 75.4 \% \\
\hline \hline
Math. BA & 18 & 15 & 83.3 \% \\
\hline \hline
Math. Dipl. & 14 & 9 & 64.3 \% \\
\hline \hline
Lehramt & 41 & 33 & 80.5 \% \\
\hline \hline
MFÜ & 32 & 29 & 90.6 \% \\
\hline \hline
Phys. BA & 31 & 24 & 77.4 \% \\
\hline \hline
Sonstige & 7 & 2 & 28.6 \% \\
\hline \hline
\end{tabular}
\end{center}

```

	Teilnehmer	bestanden	Prozent
gesamt	143	112	78.3 %
Bachelor	86	69	80.2 %
Diplom/Lehramt	57	43	75.4 %
Math. BA	18	15	83.3 %
Math. Dipl.	14	9	64.3 %
Lehramt	41	33	80.5 %
MFÖ	32	29	90.6 %
Phys. BA	31	24	77.4 %
Sonstige	7	2	28.6 %

1.3.3 Gliederung

Ein größerer Text ist überlicherweise gegliedert, z.B. in Kapitel, Paragraphen, Abschnitte, Unterabschnitte. Die Überschriften werden meist nummeriert und in Extrazeilen mit größerer (Fett)Schrift gesetzt.

In Latex hängt die Anzahl der Gliederungsebenen von der in der Präambel in

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

festgelegten Dokument-Klasse ab. In der von uns gewählten `article` - Klasse verwendet man im Normalfall die drei Ebenen

```
\section{<überschrift1>}
\subsection{<überschrift2>}
\subsubsection{<überschrift3>}
```

Darunter gibt es noch zwei weitere Gliederungsklassen, auf die wir hier nicht eingehen.

Für die `book` - bzw. `report` - Klasse gibt es oberhalb der `section` - Ebene noch eine Ebene

```
\chapter{<kapitel>}
```

Beim Aufruf einer dieser Ebenen vergibt Latex eine Gliederungsnummer. Fügt man später einen zusätzlichen Gliederungsbefehl ein, so wird diese Nummerierung automatisch angepasst. Mit dem Befehl

```
\setcounter{<ebene>}{<nr>}
```

können wir die Nummerierung beeinflussen.

Beispiel

```
\section{Differentialrechnung}
\subsection{Die Ableitung einer reellen Funktion}
\subsubsection{Die 1. Ableitung}
Hier steht ein Text.
\subsubsection{Die 2. Ableitung}
Hier folgt weiterer Text.
\subsubsection{Ableitungsregeln}
Hierher gehört zum Beispiel die Produktregel.
\subsection{Die Ableitung eines Vektorfeldes}
```

```
\setcounter{section}{5}
\section{Anwendungen der Differentialrechnung}
```

1 Differentialrechnung

1.1 Die Ableitung einer reellen Funktion

1.1.1 Die 1. Ableitung

Hier steht ein Text.

1.1.2 Die 2. Ableitung

Hier folgt weiterer Text.

1.1.3 Ableitungsregeln

Hierher gehört zum Beispiel die Produktregel.

1.2 Die Ableitung eines Vektorfeldes

6 Anwendungen der Differentialrechnung

1.3.4 Inhaltsverzeichnis

Latex speichert die Gliederungsüberschriften zusammen mit ihrer Nummerierung und Seitenzahl in der Datei

```
<name>.toc      (toc = table of contents).
```

Mit dem Kommando

```
\tableofcontents
```

wird daraus ein Inhaltsverzeichnis erzeugt, und zwar an der Stelle im Text, an der dieser Befehl steht (in der Regel nach dem Titelblatt). Dabei wird die Datei `<name>.toc` verwendet, die beim vorigen Latex-Aufruf erzeugt wurde. Deshalb muss das Latex-Kommando zweimal ausgeführt werden, um das korrekte Inhaltsverzeichnis zu erhalten.

1.3.5 Der verbatim-Modus

Die Eingabe des Textes (mehrere Leerzeichen, Zeilenumbruch) hat keinen Einfluss auf das Layout. Gelegentlich möchte man jedoch den Text einschließlich Steuerbefehle so ausgegeben, wie er eingetippt wurde. Dazu dient die `verbatim` - Umgebung. Der zwischen

```
\begin{verbatim}
      :
\end{verbatim}
```

eingeschlossene Text (einschließlich Steuerbefehle) wird in der Schriftart `\tt` so dargestellt, wie er eingegeben wurde. Die darin enthaltenen Latex-Kommandos werden ignoriert.

Beispiel: Die Eingabe von

```
\begin{verbatim}
  \section{Differentialrechnung}
  \subsection{\Large Die Ableitung einer reellen Funktion}
  \subsubsection{Die 1. Ableitung}
  Hier steht ein Text
  \subsubsection{Die 2. Ableitung}
  Hier folgt weiterer Text
  \subsubsection{Ableitungsregeln}
  Hierher gehört zum Beispiel die Produktregel.
\end{verbatim}
```

liefert folgendes Erscheinungsbild:

```
\section{Differentialrechnung}
\subsection{\Large Die Ableitung einer reellen Funktion}
\subsubsection{Die 1. Ableitung}
Hier steht ein Text
\subsubsection{Die 2. Ableitung}
Hier folgt weiterer Text
\subsubsection{Ableitungsregeln}
Hierher gehört zum Beispiel die Produktregel.
```

1.3.6 Textboxen

Eine Box (oder auch ein Kasten) besteht aus einem Text, der von Latex wie ein einzelnes Zeichen behandelt wird. Dafür gibt es eine Reihe von Möglichkeiten; wir behandeln hier nur eine kleine Auswahl.

Zu den vertikalen Boxen gehören die `parbox` und die `minipage` - Umgebung:

```
\parbox[<pos>]{<breite>}{<text>}
```

bzw.

```
\begin{minipage}[<pos>]{<breite>}
  <text>
\end{minipage}
```

Der optionale Parameter `<pos>` dient der horizontalen Ausrichtung und hat dieselbe Bedeutung wie bei der `tabular` - Umgebung (vgl. Abschnitt 1.3.2).

Der obligatorische Parameter `<breite>` gibt die Breite der Box an, die wie eine kleine Seite behandelt wird: am Zeilenende (= Breite der Box) wird automatisch umgebrochen. Die Höhe der Box richtet sich nach der Größe des Textes.

Schließlich wird durch `\fbox{<text>}` der ausgewählte Text eingerahmt.

Beispiel

```
In der \fbox{laufenden Zeile} kommt hier eine
  \fbox{\fbox{ \begin{minipage}[t]{5cm}
    eingerahmte Minipage, welche aus zwei Zeilen besteht.
  \end{minipage} }}
Anschließend geht es in der normalen Textzeile weiter.
```

In der laufenden Zeile kommt hier eine eingerahmte Minipage, welche aus zwei Zeilen besteht. Anschließend geht es in der normalen Textzeile weiter.

1.3.7 Querverweise

In Lehrbüchern oder wissenschaftlichen Arbeiten sind Querverweise auf andere Textstellen üblich. In Latex wird die Stelle, auf die Bezug genommen werden soll, markiert durch das Kommando

```
\label{<marke>}
```

welches meistens nach einer Überschrift oder in einer mathematischen Formel steht. Damit enthält <marke> die aktuelle Nummerierung, auf die an anderer Textstelle durch

```
\ref{<marke>}
```

verwiesen wird. Der Name <marke> ist frei wählbar, muss aber im gesamten Latex-Dokument eindeutig sein. Durch das Kommando

```
\pageref{<marke>}
```

verweist man auf die entsprechende Seite.

Beispiel

```
\subsubsection{Tabellen}
\label{latex_tabelle}
:
```

An einer anderen Textstelle wird dann auf diesen Punkt (diese Seite) verwiesen:

```
Wir verweisen auf die in Abschnitt \ref{latex_tabelle}
beschriebene {\tt tabular}-Umgebung. \newline
Die {\tt tabular}-Umgebung wird auf Seite \pageref{latex_tabelle}
beschrieben.
```

```
Wir verweisen auf die in Abschnitt 1.3.2 beschriebene tabular-Umgebung.
Die tabular-Umgebung wird auf Seite 10 beschrieben.
```

1.3.8 Literaturverzeichnis

Zu wissenschaftlichen Arbeiten gehört die Angabe der verwendeten Literatur (in der Regel am Ende des Textes). Ebenso muss an den betroffenen Textstellen auf die entsprechende Literatur verwiesen werden.

Das Literaturverzeichnis wird erzeugt durch

```
\begin{thebibliography}{<muster>}
  \bibitem{<name1>} <text>
  \bibitem{<name2>} <text>
  \bibitem{<name3>} <text>
  :
\end{thebibliography}
```

Der Eintrag für `<muster>` bestimmt die Breite für das Einrücken von `<text>`. Handelt es sich um eine zweistellige Anzahl von Literaturangaben, so könnte man

```
\begin{thebibliography}{99}
```

verwenden.

Das Zitieren der entsprechenden Literatur im Text erfolgt durch

```
\cite{<name1>}
```

Beispiel

Wir verweisen auf die in `\cite{roesch}` gemachten Ausführungen.

```
\begin{thebibliography}{99}
\bibitem{brunner} {\bf Brunner; Brück:} Mathematik für Chemiker,
    2. Auflage. Springer 2008.
\bibitem{zachmann} {\bf Zachmann; Jüngel:} Mathematik für Chemiker,
    6. Auflage. Wiley-VCH 2007.
\bibitem{reinsch} {\bf Reinsch:} Mathematik für Chemiker,
    Teubner 2004.
\bibitem{roesch} {\bf Rösch:} Mathematik für Chemiker, Springer 1993.
\bibitem{scherfner} {\bf Scherfner; Senkbeil:}
    Lineare Algebra für das erste Semester.
    Pearson Studium 2006.
\end{thebibliography}
```

Wir verweisen auf die in [4] gemachten Ausführungen.

Literatur

- [1] **Brunner; Brück:** Mathematik für Chemiker, 2. Auflage. Springer 2008.
- [2] **Zachmann; Jüngel:** Mathematik für Chemiker, 6. Auflage. Wiley-VCH 2007.
- [3] **Reinsch:** Mathematik für Chemiker, Teubner 2004.
- [4] **Rösch:** Mathematik für Chemiker, Springer 1993.
- [5] **Scherfner; Senkbeil:** Lineare Algebra für das erste Semester. Pearson Studium 2006.

1.4 Wichtige Elemente im Mathematik-Modus

1.4.1 Die Schriftart `mathbb`

Zur Darstellung der Zahlbereiche \mathbb{N} , \mathbb{Z} , \mathbb{R} , ... gibt es in Latex die Schriftart `mathbb`, welche nur im Mathematikmodus möglich ist. Dieser Zeichensatz ist nur für Großbuchstaben definiert.

Der Aufruf erfolgt durch das Kommando

```
\mathbb{<buchstabe>}
```

und liefert das folgende Alphabet:

A, B, C, D, E, F, G, H, I, J, K, A, L, M,
N, O, P, Q, R, S, T, U, V, W, X, Y, Z,

1.4.2 Griechische Buchstaben

α	<code>\alpha</code>	A	A	ν	<code>\nu</code>	N	N
β	<code>\beta</code>	B	B	ξ	<code>\xi</code>	Ξ	<code>\Xi</code>
γ	<code>\gamma</code>	Γ	<code>\Gamma</code>	o	o	O	O
δ	<code>\delta</code>	Δ	<code>\Delta</code>	π	<code>\pi</code>	Π	<code>\Pi</code>
ϵ	<code>\epsilon</code>	E	E	ρ	<code>\rho</code>	P	P
ε	<code>\varepsilon</code>	E	E	ϱ	<code>\varrho</code>	P	P
ζ	<code>\zeta</code>	Z	Z	σ	<code>\sigma</code>	Σ	<code>\Sigma</code>
η	<code>\eta</code>	H	H	τ	<code>\tau</code>	T	T
θ	<code>\theta</code>	Θ	<code>\Theta</code>	υ	<code>\upsilon</code>	Υ	<code>\Upsilon</code>
ι	<code>\iota</code>	I	I	ϕ	<code>\phi</code>	Φ	<code>\Phi</code>
κ	<code>\kappa</code>	K	K	χ	<code>\chi</code>	X	X
λ	<code>\lambda</code>	Λ	<code>\Lambda</code>	ψ	<code>\psi</code>	Ψ	<code>\Psi</code>
μ	<code>\mu</code>	M	M	ω	<code>\omega</code>	Ω	<code>\Omega</code>

1.4.3 Mathematische Symbole

\subset	<code>\subset</code>	\supset	<code>\supset</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>
\cup	<code>\cup</code>	\cap	<code>\cap</code>
\in	<code>\in</code>	\ni	<code>\ni</code>
\notin	<code>\notin</code>	\neq	<code>\neq</code>
\leq	<code>\leq</code>	\geq	<code>\geq</code>
\ll	<code>\ll</code>	\gg	<code>\gg</code>
\approx	<code>\approx</code>	\neq	<code>\neq</code>
\sim	<code>\sim</code>	\simeq	<code>\simeq</code>
$ $	<code>\mid</code>	\parallel	<code>\parallel</code>
\pm	<code>\pm</code>	\mp	<code>\mp</code>
$*$	<code>\ast</code>	\circ	<code>\circ</code>
\times	<code>\times</code>	\perp	<code>\perp</code>

1.4.4 Pfeile

\rightarrow	<code>\rightarrow</code>	\leftarrow	<code>\leftarrow</code>
\uparrow	<code>\uparrow</code>	\downarrow	<code>\downarrow</code>
\Rightarrow	<code>\Rightarrow</code>	\Leftarrow	<code>\Leftarrow</code>
\Uparrow	<code>\Uparrow</code>	\Downarrow	<code>\Downarrow</code>
\longrightarrow	<code>\longrightarrow</code>	\longleftarrow	<code>\longleftarrow</code>
\Longrightarrow	<code>\Longrightarrow</code>	\Longleftarrow	<code>\Longleftarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>	\updownarrow	<code>\updownarrow</code>
\Leftrightarrow	<code>\Leftrightarrow</code>	\Updownarrow	<code>\Updownarrow</code>
\longleftrightarrow	<code>\longleftrightarrow</code>	\Leftrightarrow	<code>\Leftrightarrow</code>
\iff	<code>\iff</code>	\rightarrow	<code>\to</code>
\mapsto	<code>\mapsto</code>		

1.4.5 Mathematische Akzente

Die mathematischen Akzente erhält man wie folgt:

<code>\hat{a}</code>	\hat{a}
<code>\bar{a}</code>	\bar{a}
<code>\tilde{a}</code>	\tilde{a}
<code>\vec{a}</code>	\vec{a}
<code>\dot{a}</code>	\dot{a}
<code>\ddot{a}</code>	\ddot{a}

Für einige dieser Akzente gibt es eine Breitversion:

<code>\widehat{AB}</code>	\widehat{AB}
<code>\widetilde{x-y}</code>	$\widetilde{x-y}$

1.4.6 Standardfunktionen

Für die Standardfunktionen werden die in der Mathematik üblichen Bezeichnungen mit einem vorangestellten Backslash verwendet. Beachten Sie den Unterschied im Schriftbild:

richtig: $\exp(ix) = \cos(x) + i \sin(x)$ `\exp(ix) = \cos(x) + i \sin(x)`

falsch: $exp(ix) = cos(x) + isin(x)$ `$exp(ix) = cos(x) + i sin(x)$`

Wurzelfunktionen erhält man durch das `\sqrt` - Kommando, zum Beispiel

$$\text{\sqrt{2x+1}} \quad \sqrt{2x + 1}$$

$$\text{\sqrt[5]{2x+1}} \quad \sqrt[5]{2x + 1}$$

1.4.7 Hochzahlen und Indizes

Für obere Indizes wird das `^` - Zeichen, für untere Indizes wird das `_` - Zeichen verwendet. Sollen mehrere Zeichen hoch- bzw. tiefgestellt werden, so sind sie in geschweifte Klammern einzuschließen.

Beispiele

$$\text{\$(a+b)^{13}\$} \quad (a + b)^{13}$$

$$\text{\$A_i^k\$} \quad A_i^k$$

$$\text{\$2^{2^{i+1}}\$} \quad 2^{2^{i+1}}$$

Beachten Sie den Unterschied zwischen `\$a^{13}\$` a^{13} und `\$a^13\$` a^{13} .

1.4.8 Summen, Produkte, Integrale

Für Summen, Produkte und Integrale gibt es zwei Möglichkeiten, abhängig vom gewünschten Erscheinungsbild:

$$\text{\$\sum_{i=1}^{10} a_i\$} \quad \sum_{i=1}^{10} a_i$$

$$\text{\$\sum\limits_{i=1}^{10} a_i\$} \quad \sum_{i=1}^{10} a_i$$

$$\text{\$\prod_{i=1}^{10} a_i\$} \quad \prod_{i=1}^{10} a_i$$

$$\text{\$\prod\limits_{i=1}^{10} a_i\$} \quad \prod_{i=1}^{10} a_i$$

$$\text{\$\int_a^b f(x) \, dx \$} \quad \int_a^b f(x) dx$$

$$\text{\$\int\limits_a^b f(x) \, dx\$} \quad \int_a^b f(x) dx$$

In Formelzeilen kann auf den `limits` - Befehl verzichtet werden.

Mit dem `shortstack`-Kommando können mehrere Zeilen übereinander gesetzt werden:

$$\text{\$\sum\limits_{\scriptsize \shortstack{\$i=1\$ \\ \$i \neq k\$}}^m (t_i - t_k)\$}$$

erzeugt die Ausgabe $\sum_{\substack{i=1 \\ i \neq k}}^m (t_i - t_k)$.

1.4.9 Grenzwerte

Auch für die Darstellung von Grenzwerten gibt es diese zwei Varianten. In diesem Zusammenhang wird häufig auch das ∞ -Zeichen gebraucht, welches in Latex durch das Kommando `\infty` erzeugt wird.

$$\text{\$}\lim_{x \to \infty} e^{-x} = 0\text{\$} \qquad \lim_{x \rightarrow \infty} e^{-x} = 0$$

$$\text{\$}\lim\limits_{x \to \infty} e^{-x} = 0\text{\$} \qquad \lim_{x \rightarrow \infty} e^{-x} = 0$$

1.4.10 Brüche

Zur Darstellung von Brüchen verwendet man das Kommando

$$\text{\$}\frac{\langle \text{zähler} \rangle}{\langle \text{nenner} \rangle}\text{\$}$$

Selbstverständlich sind in Brüchen wieder Brüche möglich.

Beispiele

$$\text{\$}\frac{a+b}{c-d}\text{\$}$$

$$\text{\$}\frac{2 + \frac{1}{1 + \exp(2-x)}}{\frac{x^3 + 2x + 1}{x-1}}\text{\$}$$

$$2 + \frac{1}{1 + \exp(2-x)} \\ \frac{x^3 + 2x + 1}{x-1}$$

1.4.11 Dynamische Klammern

Latex kann Klammern an die Größe von Formeln anpassen. Dies wird erreicht durch

$$\text{\$}\left\langle \text{klammer} \right\rangle \dots \text{\$}\right\langle \text{klammer} \right\rangle\text{\$}$$

Wichtig: zu jedem `\left` gehört ein `\right`. Dabei darf es sich um unterschiedliche Klammern handeln.

Beispiel

$$\text{\$}\left(\sum \limits_{i=1}^{10} a_i \right)^2\text{\$}$$

$$\left(\sum_{i=1}^{10} a_i \right)^2$$

Ohne dynamische Klammerung ergibt sich

$$\text{\$}\left(\sum \limits_{i=1}^{10} a_i \right)^2\text{\$}$$

$$\left(\sum_{i=1}^{10} a_i \right)^2$$

1.4.12 Vektoren und Matrizen

Für Vektoren und Matrizen gibt es die `array` - Umgebung, welche wie die `tabular` - Umgebung aufgebaut ist, jedoch nur im Mathematik-Modus funktioniert:

```

\begin{array}[<pos>]{<spalten>}
  <element11> & <element12> & \dots & <element1n> & \\
  <element21> & <element22> & \dots & <element2n> & \\
  & & \vdots & & & \\
\end{array}

```

Die Parameter `<pos>` und `<spalten>` entsprechen denen der `tabular`-Umgebung, vgl. Abschnitt 1.3.2.

Im Zusammenhang mit Matizen treten folgende Symbole auf:

```

\cdots \dots \vdots \ddots \dots \ldots

```

Beispiel

```

\[
  B =
  \begin{array}{cccc}
    1 & 2 & 3 & 4 \\
    5 & 6 & 7 & 8
  \end{array}
  , \hspace{1em}
  A = \left(
  \begin{array}{ccc}
    a_{11} & \cdots & a_{1n} \\
    \vdots & & \vdots \\
    a_{n1} & \cdots & a_{nn}
  \end{array}
  \right)
\]
```

$$B = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{pmatrix}, \quad A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}$$

Einfacher (aber nicht mehr so flexibel) ist die `pmatrix`-Umgebung, welche alle Spalten zentriert ausrichtet und die runden Klammern automatisch setzt.

Beispiel

```

\[
  \begin{pmatrix}
    a_{11} & a_{12} & a_{13} \\
    a_{21} & a_{22} & a_{23} \\
    a_{31} & a_{32} & a_{33}
  \end{pmatrix}
\]
```

1.4.13 Formeln mit automatischer Nummerierung

Dafür gibt es die `equation`-Umgebung. Diese wird im Text-Modus aufgerufen und wählt automatisch den Mathematik-Modus.

```
\begin{equation}
...
\label{<name>}
\end{equation}
```

Dabei ist `<name>` frei wählbar.

Beispiel

Gegeben sei die Funktion

```
\begin{equation}
F(x,y) = \frac{2xy}{\sqrt{1 + x^2 + (y-1)^2}};
\mbox{ für } (x,y) \in \mathbb{R}^2.
\label{formel1}
\end{equation}
```

Für die Funktion in (`\ref{formel1}`) bestimmen wir die partielle Ableitung

```
\[
\frac{\partial F(x,y)}{\partial x}
\]
```

Gegeben sei die Funktion

$$F(x, y) = \frac{2xy}{\sqrt{1 + x^2 + (y - 1)^2}} \quad \text{für } (x, y) \in \mathbb{R}^2. \quad (1)$$

Für die Funktion in (1) bestimmen wir die partielle Ableitung

$$\frac{\partial F(x, y)}{\partial x}.$$

In diesem Beispiel tritt das Symbol `\partial` `\partial` auf, welches in der Mathematik für partielle Ableitungen verwendet wird. Mit `\mbox{<text>}` wird eine Textbox im Mathematik-Modus erzeugt.

Die gewöhnliche Ableitungen einer reellen Funktion erhalten wir durch

$$\begin{aligned} f^{\prime}(x) &\rightsquigarrow f'(x) \quad , \\ f^{\prime\prime}(x) &\rightsquigarrow f''(x) \quad , \\ f^{(n)}(x) &\rightsquigarrow f^{(n)}(x) \quad . \end{aligned}$$

1.4.14 Mehrzeilige Formeln

Hier gibt es zwei Möglichkeiten. Ist eine automatische Nummerierung der einzelnen Formelzeilen gewünscht, so wird die `eqnarray`-Umgebung verwendet, ansonsten wird die `eqnarray*`-Umgebung. Diese Formeln bestehen aus genau 3 Spalten, wobei die erste

Spalte rechtsbündig, die zweite zentriert und die dritte linksbündig dargestellt wird.

```
\begin{eqnarray}
  <ausdruck1> & & <ausdruck2> & & <ausdruck3> \label{ <name1> } \\
  <ausdruck4> & & <ausdruck5> & & <ausdruck6> \label{ <name2> } \\
  & & & & \vdots \\
\end{eqnarray}
```

Beispiel 1 (mit Nummerierung)

Gegeben sei das Gleichungssystem

```
\begin{eqnarray}
  x + y & = & 10 \label{f1} \\
  x^2 + y^2 + z^2 & = & 30 \label{f2} \\
  2xy + xyz + 4z & = & 90 \label{f3}
\end{eqnarray}
```

Wir lösen in (\ref{f1}) nach x auf
und setzen das Ergebnis in (\ref{f2})
und (\ref{f3}) ein.

Gegeben sei das Gleichungssystem

$$\begin{aligned} x + y &= 10 & (2) \\ x^2 + y^2 + z^2 &= 30 & (3) \\ 2xy + xyz + 4z &= 90 & (4) \end{aligned}$$

Wir lösen in (2) nach x auf und setzen das Ergebnis in (3) und (4) ein.

Sollen einzelne Zeilen nicht mit einer Nummer versehen werden, so verwendet man in der entsprechenden Formelzeile das Kommando `\nonumber`.

Beispiel 2 (ohne Nummerierung)

Gegeben sei das Gleichungssystem

```
\begin{eqnarray*}
  x + y & = & 10 \\
  x^2 + y^2 + z^2 & = & 30 \\
  2xy + xyz + 4z & = & 90
\end{eqnarray*}
```

Gegeben sei das Gleichungssystem

$$\begin{aligned} x + y &= 10 \\ x^2 + y^2 + z^2 &= 30 \\ 2xy + xyz + 4z &= 90 \end{aligned}$$

Anmerkung: Wenn Sie das `amsmath`-Paket in der Präambel eingebunden haben, so können Sie für mehrzeilige Formeln auch die `align`-Umgebung verwenden. Diese liefert in einigen Fällen ein besseres Layout.

Beispiel (ohne Nummerierung)

Gegeben sei das Gleichungssystem

```
\begin{align*}
x + y &= 10 \\
x^2 + y^2 + z^2 &= 30 \\
2xy + xyz + 4z &= 90
\end{align*}
```

Gegeben sei das Gleichungssystem

$$\begin{aligned}x + y &= 10 \\x^2 + y^2 + z^2 &= 30 \\2xy + xyz + 4z &= 90\end{aligned}$$