

Hinweise und Tipps zu Latex

Mira Schedensack

4. Dezember 2024

1 Einleitung

Dieses Dokument soll einige Hinweise zur Benutzung von Latex geben. Es ist keine ausführliche Anleitung zu Latex, sondern soll dazu führen, dass der Leser relativ schnell und ohne all zu große eigenständige Recherche in der Lage ist, in Latex einen Bericht zur Praktikumsaufgabe zu verfassen. Zudem soll das Latex-template `latex_template.tex` helfen, einen schnellen Einstieg in das Verfassen des Berichts zu finden. In jenem Dokument ist bereits eine grobe Struktur inklusive Titelseite erstellt.

2 Allgemeine Hinweise

Die folgende Liste enthält allgemeine Hinweise zu Latex.

- Kapitel bzw. **Abschnitte** werden mit `\chapter{}` (nur in Buch-Stilen), `\section{}`, `\subsection{}`, etc. erzeugt. Ein Stern (also z.B. `\section*{}`) führt dazu, dass keine Nummer vergeben wird. Für die Gliederung von kurzen Abschnitten kann gegebenenfalls auch `\paragraph` benutzt werden.
- **Absätze** werden durch eine Leerzeile erzeugt. Der Befehl `\\` erzeugt einen Zeilenumbruch. Der Unterschied ist, dass nach einem Zeilenumbruch keine Einrückung erfolgt, die aber bei Absätzen erwünscht ist.
- Der Befehl `\noindent` führt dazu, dass (nach einem Absatz) der Text nicht eingerückt wird.
- **Zitate** werden als

`\cite[Satz~3.8]{FreundHoppe2007}`

geschrieben. Dies erzeugt [FH07, Satz 3.8]. Die Tilde führt dazu, dass zwischen Satz und 3.8 kein Zeilenumbruch stattfindet.

- (Deutsche) **Anführungszeichen** werden mit `\glqq` und `\grqq` erzeugt. Um nach dem rechten Anführungszeichen ein Leerzeichen zu erzeugen, kann der Backslash `\` benutzt werden.
- **Umlaute** können mit `\` und Anführungszeichen vor dem jeweiligen Vokal erzeugt werden, also z.B. ä, ö, ü, Ä. Das Esszett kann mit `\ss` erzeugt werden.
- Mit dem Befehl `\newpage` wird eine **neue Seite** angefangen. Dieser Befehl sollte nur dort benutzt werden, wo es wirklich (inhaltlich) notwendig ist, z.B. vor der Selbstständigkeitserklärung.
- Die **Silbentrennung** wird automatisch ausgeführt, wenn das Paket `babel` mit der Option `ngerman` geladen ist. Möchte man per Hand mitteilen, wo ein Wort getrennt werden kann (z.B. werden Wörter mit Umlauten, die mit `\` und Anführungszeichen eingegeben sind, nicht richtig getrennt), dann kann man dies mit `\-` eingeben, z.B. `An\-\fuh\-\rungs\-\zei\-\chen`. Latex entscheidet dann, ob und wo das Wort am besten getrennt wird.
- Nach Abkürzungen mit Punkt sollte `\` stehen, wenn danach ein Leerzeichen kommt, also zum Beispiel z.B.`\`. Dies führt dazu, dass der Abstand nach dem Punkt nicht größer ist als ein üblicher Abstand zwischen zwei Wörtern.
- Wenn kein Zeilenumbruch erwünscht ist (aber ein Abstand), kann die Tilde benutzt werden. Dieser Befehl sollte z.B. verwendet werden, wenn auf eine Abbildung oder einen Abschnitt verwiesen wird, also z.B. auf Abschnitt 2. In diesem Fall sollte das Wort „Abschnitt“ und „2“ nicht durch einen Zeilenumbruch voneinander getrennt werden.
- Die Umgebung **center** zentriert den Text. Die ist zum Beispiel für Bilder oder Tabellen nützlich.
- Aufzählungen können mit der `itemize`-Umgebung oder mit der `enumerate`-Umgebung erzeugt werden. Vor den Stichpunkten steht bei der `itemize`-Umgebung ein `bullet` (wenn sie verschachtelt ist, ein `Minus`), bei der `enumerate`-Umgebung werden die einzelnen Punkte nummeriert. Mit dem optionalen Argument hinter einem `item`, kann dieses Zeichen verändert werden. Also erzeugt

```

\begin{itemize}
  \item erster Stichpunkt
  \item[(2)] zweiter Stichpunkt
  \item[$\bullet$] dritter Stichpunkt
\end{itemize}

```

zum Beispiel

- erster Stichpunkt
- (2) zweiter Stichpunkt
 - dritter Stichpunkt
- Ein Inhaltsverzeichnis kann mit `\tableofcontents` erzeugt werden, ist aber nur bei längeren Dokumenten (z.B. einer Abschlussarbeit) sinnvoll.

3 Formeln

Um im Text in den Mathe-Modus zu wechseln, werden Dollarzeichen verwendet. Die Formel $a = b$ wird also mit `$a=b$` erzeugt.

Um abgesetzte Formeln zu schreiben, gibt es verschiedene Umgebungen, z.B. `equation` für einzeilige Formeln und `align` für ein- oder mehrzeilige. Die Umgebungen mit einem Stern werden nicht nummeriert, also z.B.

$$a = b \quad \text{dies ist die equation-Umgebung,} \tag{1}$$

oder

$$a = b, \tag{2}$$

$$\text{dies ist die align-Umgebung,} \tag{3}$$

oder nicht-nummeriert

$$a = b \quad \text{dies ist die equation-Umgebung mit Stern,}$$

oder

$$a = b,$$

dies ist die align-Umgebung mit Stern.

Achtung: In den Mathe-Umgebungen dürfen keine Leerzeilen stehen!

Möchte man eine mehrzeilige Formel mit nur einer Nummer erzeugen, kann dies mit der `aligned`-Umgebung innerhalb der `equation`-Umgebung geschehen,

$$\begin{aligned} a &= b, \\ c &= d. \end{aligned} \tag{4}$$

Möchte man auf Formeln verweisen, muss man ihnen mit `\label{}` einen Namen geben und kann dann mit `\eqref{}` darauf verweisen, also z.B. auf Formel (4). Benutzt man `\ref{}`, erhält man 4, also die Nummer der Formel ohne Klammern. Bei Formeln sollte man allerdings `\eqref{}` verwenden. Verweist man hingegen auf Abschnitte, Tabellen oder Bilder, benutzt man `\ref{}`, also z.B. bei Abschnitt 3.

Formeln sollten nur dann eine Nummer bekommen, wenn diese auch benutzt werden!

Gleichheits- oder Ungleichheitszeichen können mit `&` untereinander gesetzt werden. Beispiele:

$$a = b,$$

$$a + b = c,$$

oder auch

$$a = b, \quad c = 2d$$

$$a + b = c, \quad abcd = 3d + 4a.$$

Die Multiplikation (auch von Matrizen mit Vektoren) wird durch ein Hintereinander-Schreiben erzeugt, also z.B. $Ax = b$. Das Skalarprodukt kann durch `\cdot` erzeugt werden, $x \cdot y$. Geht eine Formel über mehr als eine Zeile und fängt die neue Zeile mit einer Multiplikation an, dann wird diese mit `\times`, also $\times a$ erzeugt.

Die Division wird entweder mit `/` geschrieben (im Text ist dies die bevorzugte Variante) oder als Bruch mit `\frac{Zähler}{Nenner}`,

$$\frac{a}{b}.$$

Indizes und auch Zeichen, die unterhalb von etwas gesetzt werden, werden mit `_` geschrieben, Potenzen und Zeichen, die oberhalb von etwas gesetzt werden, werden mit `^` geschrieben, also z.B.

$$a_k, \quad b^2, \quad \sum_{k=1}^n, \quad \lim_{x \rightarrow 0}$$

Abstände können (wie auch im Text) mit `\,`, `\;`, `\;`, `\enspace`, `\quad`, `\qquad` generiert werden, z.B.

$$x_k = 1/k \quad \text{für alle } k = 1, \dots$$

Leerzeichen im Quell-Text werden im Mathe-Modus ignoriert. Die Größe von Klammern kann mit `\big`, `\Big`, `\bigg`, `\Bigg` manuell angepasst werden, dies ergibt

$$(\,, (\,, (\,, (\,, (\quad \{\,, \{\,, \{\,, \{\,, \{\,$$

(die erste Klammer ist eine normale Klammer). Oder sie werden automatisch mit `\left` und `\right` angepasst, dies ergibt

$$(a^b), \left(\int a \right).$$

Wenn auf einer Seite keine Klammer stehen soll, kann statt `\right` auch `\.` stehen. Dies ergibt

$$\left\{ \int a . \right.$$

Für eine Fallunterscheidung wird allerdings die Umgebung `cases` benutzt, also z.B.

$$a \bmod 2 = \begin{cases} 0 & \text{wenn } a \text{ gerade,} \\ 1 & \text{wenn } a \text{ ungerade.} \end{cases}$$

Variablen werden in Latex in Formeln (automatisch) kursiv gesetzt. Operatoren werden dagegen aufrecht gesetzt, vergleiche zum Beispiel

$$\exp(x) \text{ (durch } \backslash\exp(x) \text{ erzeugt),}$$

$$\exp(x) \text{ (durch } \exp(x) \text{ erzeugt).}$$

Streng genommen bedeutet die zweite Version, dass die Variablen e , x und p miteinander multipliziert werden. Will man einen Operator benutzen, der nicht in einem der üblichen Pakete enthalten ist (z.B. `amsmath`), so kann man sich einen Operator durch `\operatorname{}` definieren, also z.B.

$$\operatorname{mid}(T) \text{ (durch } \backslash\operatorname{mid}(T) \text{ erzeugt).}$$

Wir dieser Ausdruck mit einem Index benutzt, wird der Index rechts unten gesetzt. Soll der Index direkt unterhalb stehen, kann man `\operatorname*{}` benutzen,

$$\operatorname{mid}_k, \quad \operatorname*{mid}_k$$

Aufrechter Text (also nicht kursiv) kann auch mit `\textrm{}` erzeugt werden. Für Text, der innerhalb einer Formel-Umgebung stehen soll, wird der Befehl `\text{}` benutzt.

Formeln gehören zum Text. Wenn ein Satz mit einer abgesetzten Formel aufhört, muss ein Punkt hinter der Formel stehen, also z.B. bei

$$a = b.$$

4 Definitionen, Sätze, usw.

Für die Verwendung von Definitionen, Sätzen oder ähnlichem wird am besten das Paket `amsthm` geladen. Die jeweiligen Umgebungen werden nun zu Beginn (also noch vor `\begin{document}`) definiert. Möchte man zum Beispiel Umgebungen für Sätze, Lemmata, Definitionen und Bemerkungen benutzen, kann man das wie folgt machen:

```

\newtheorem{theorem}{Satz}
\newtheorem{lemma}[theorem]{Lemma}

\theoremstyle{definition}
\newtheorem{definition}[theorem]{Definition}
\newtheorem{bemerkung}[theorem]{Bemerkung}

```

Das erste Argument im Befehl `\newtheorem` definiert den Namen, mit dem man die jeweilige Umgebung aufruft. Das zweite Argument ist das, was im Dokument angezeigt wird. Der Aufruf

```

\begin{theorem}
  Dies ist ein Satz.
\end{theorem}

```

führt dann zu

Satz 1. *Dies ist ein Satz.*

Der Beweis wird dann in die Umgebung

```

\begin{proof}
  ...
\end{proof}

```

gesetzt. Dies erzeugt

Beweis. ... □

Das optionale Argument `theorem` bei

```

\newtheorem{lemma}[theorem]{Lemma}

```

führt dazu, dass die Lemmata in der gleichen Zählung nummeriert werden, wie die Sätze.

```

\begin{lemma}
  Dies ist ein Lemma.
\end{lemma}

```

führt dann also zu

Lemma 2. *Dies ist ein Lemma.*

da es schon einen Satz gab. Während bei einem Satz der Text in kursiv erscheint, kann durch

```

\theoremstyle{definition}

```

der Stil so verändert werden, dass der Text aufrecht gesetzt wird. Der Text in

Abbildung 1: Dies ist die erste figure-Umgebung, aber ohne Bild (in Wirklichkeit sollte die figure-Umgebung natürlich immer (mindestens) ein Bild enthalten).

```
\begin{definition}
  Dies ist eine Definition.
\end{definition}
```

führt dann also zu

Definition 3. Dies ist eine Definition.

Dieser Stil wird üblicherweise für Definitionen verwendet und kann z.B. auch für Bemerkungen verwendet werden.

5 Darstellung von code

Um in einem Text Namen von Funktionen oder Variablen aus einem code zu verdeutlichen, ist der Befehl `\texttt{}` hilfreich. Zum Beispiel erzeugt der Befehl `\texttt{solve\_LGS}` die Ausgabe `solve_LGS`.

Längere Abschnitte von code können zum Beispiel mit dem Paket listings abgebildet werden.

6 Abbildungen

Abbildungen sollten in einer figure-Umgebung stehen. Eine figure-Umgebung wird dann von latex dort hingesetzt, wo es sinnvoll ist. Wenn man sich im Text auf die Abbildung bezieht, wird diese dann referenziert. Dies sieht dann wie folgt aus

```
\begin{figure}
  \caption{\label{f:namederAbbildung} Hier kommt die Bildunterschrift. }
\end{figure}
```

Mit `\ref{f:namederAbbildung}` kann dann auf die Abbildung verwiesen werden. Abbildung 1 zeigt eine solche figure-Umgebung, in der noch kein Bild hinzugefügt wurde.

In Abbildung 2 wurde ein Bild mit dem Befehl `\includegraphics{dateiname}` eingefügt. Das Bild wurde übrigens auch mit latex (mit pgfplot) erzeugt, siehe dazu die Datei `convhist_expl.tex`.

Abbildung 3 zeigt zwei Bilder, die im Dokument mit Hilfe des Paktes tikz erstellt wurden.

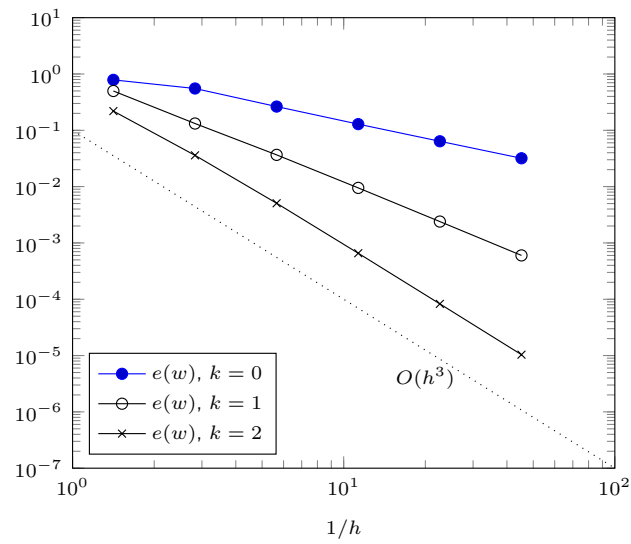


Abbildung 2: Hier wurde ein Bild als pdf-Datei erzeugt und eingefügt.

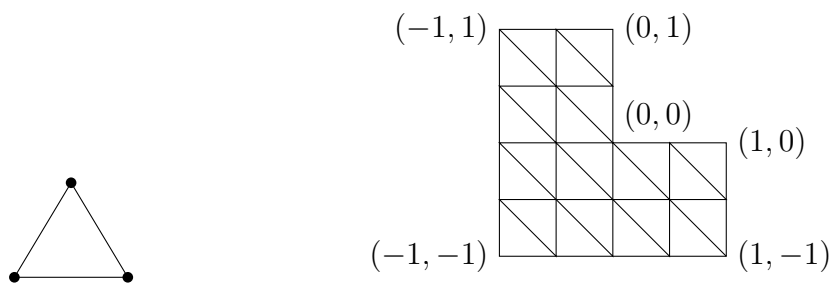


Abbildung 3: Hier sind zwei Bilder zu sehen, die mit dem Paket tikz erstellt wurden.

1	links	mittig	rechts
2	222	1	43
3	21	1000	5032

Tabelle 1: Dies ist eine Beispiel-Tabelle mit zwei linksbündigen, einer mittig ausgerichteten und einer rechtsbündigen Spalte.

7 Tabellen

Eine Tabelle sollte in einer `table`-Umgebung stehen. Diese setzt die Tabelle dorthin im Text, wo sie gut hinpasst. Dies ist analog zur `figure`-Umgebung. Die `table`-Umgebung enthält die Tabelle und eine Ueberschrift, sie sieht also wie folgt aus:

```
\begin{table}
  \begin {tabular}{...}
    ...
  \end {tabular}
  \caption{\label{tab:namederTabelle} Hier kommt die Tabellenunterschrift.}
\end{table}
```

Für die `tabular`-Umgebung muss am Anfang spezifiziert werden, wie viele Spalten die Tabelle hat und ob sie links- oder rechtsbündig oder mittig ausgerichtet werden. Außerdem können auch senkrechte Linien zwischen den Spalten angegeben werden. Die Tabelle

```
\begin {tabular}{l|lcr}
  ...
\end {tabular}
```

Hat beispielsweise erst eine linksbündige Spalte, dann folgt eine vertikale Linie und danach eine linksbündige, dann eine mittig ausgerichtete und dann eine rechtsbündige Spalte. Die Spalteneinträge werden dann mit einem `&` voneinander getrennt, die Zeile wird (wie sonst auch) mit `\\` umgebrochen. Ein weiterer nützlicher Befehl ist der Befehl `\hline`, der eine waagerechte Linie erzeugt. Die Tabelle 1 wurde also durch

```
\begin {tabular}{l|lcr}
  1 & links & mittig & rechts\\
  \hline
  2 & 222 & 1 & 43\\
  3 & 21 & 1000 & 5032
\end {tabular}
```

erzeugt.

1	links	mittig	rechts
2	222	1	43
3	21	1000	5032

Tabelle 2: Dies ist dieselbe Tabelle wie Tabelle 1, aber mit einer zusätzlichen center-Umgebung.

Literatur

- [FH07] Roland W. Freund and Ronald H. W. Hoppe. *Stoer/Bulirsch: Numerische Mathematik 1*. Springer-Lehrb. Berlin: Springer, 10th revised ed. edition, 2007.