

Bergische Universität Wuppertal

Fachschaft Physik



L^AT_EX: Einführung und Installation

In der folgenden Dokumentation soll es um die ersten Schritte mit dem Textsatzsystem L^AT_EX gehen. Sie richtet sich dabei an Anfänger, die noch nicht mit einem derartigen System gearbeitet haben und somit keine Erfahrung mitbringen. Genauer bezogen wird sich dabei auf die aktuellste Version L^AT_EX 2_ε, deren Einrichtung sowie auf die Erstellung von Dokumenten mit pdfL^AT_EX. Außerdem erfolgt eine Auflistung nützlicher Pakete, sowie eine Übersicht über deren Funktionen und Vorteilen. Anschließend werden einige häufig verwendete Beispiele exemplarisch dargelegt.

Inhaltsverzeichnis

1	Einleitung	2
1.1	Der \LaTeX -Grundsatz	2
1.2	Ein Minimalbeispiel	3
2	Installation und Einrichtung	4
2.1	Windows: MiKTeX	4
2.2	Mac OS X: MacTeX	4
2.3	Linux: TeXLive	4
2.4	Vergleich	5
3	Erste Schritte mit \LaTeX	5
3.1	Die Dokumentklasse	5
3.2	Wichtige Pakete	6
4	Beispiele	8
4.1	Einfache Strukturierung	8
4.2	Grafik mit einer Bildunterschrift einfügen	9
4.3	Tabellen erstellen	10
4.4	Mathematikmodus und Formeln	12
4.5	Einheiten	13
5	Referenzen	13

1 Einleitung

\LaTeX , gesprochen [la-tɛç], ist *das* Textsatzsystem, wenn es um wissenschaftliche Arbeiten und Publikationen in naturwissenschaftlichen Fachgruppen an Universitäten weltweit geht. Beim Verfassen der Thesis in der Physik führt quasi kein Weg daran vorbei. Aus diesem und diversen anderen Gründen empfiehlt es sich, sich möglichst früh im Studienverlauf mit dem Konzept eines typografischen „what you see is what you mean“-Systems auseinanderzusetzen. Aller Anfang ist schwer, und das ändert sich auch bezüglich \LaTeX nicht, aber an und für sich ist es nicht umständlicher, als Word zu benutzen. Es ist lediglich *anders*, und wenn man ein wenig übt, lassen sich schneller deutlich bessere Resultate erzielen als beispielsweise mit Word. Insbesondere der Formelsatz verschafft \LaTeX hier einen deutlichen Vorsprung.

Die Legitimierung der Verwendung von Latex ist im Grunde ganz einfach: Wenn man gewisse Grundregeln verfolgt, ist es leicht, stيلةchte und ansehnliche Dokumente zu erstellen, die allen wissenschaftlichen Standards genügen.

1.1 Der \LaTeX -Grundsatz

Die Idee hinter Latex ist einfach: Der Autor konzentriert sich auf den Inhalt und die Struktur des Dokuments, während das Textsatzsystem \TeX die Formatierung übernimmt. Dabei können sogenannte „Pakete“ verwendet werden, die es erlauben, kompliziertere Strukturen umzusetzen. Die allgemeine Vorgehensweise zum Erstellen eines Dokuments in Latex ändert sich dabei nie:

- **Manuskript (Editor):** Zunächst muss das Manuskript des Dokuments (ähnlich einem Quellcode) in einem Editor verfasst werden. Es wird in eine Textdatei `Dokument.tex` geschrieben. Ein Minimalbeispiel findet sich in Abschnitt 1.2. Dafür gibt es zunächst keine Beschränkung. Hierfür kann Notepad++, gedit, nano oder jeder andere Texteditor verwendet werden. Oft bieten sich spezielle Latex-Editoren wie „TexMaker“ oder „TexStudio“ an, die viele nützliche Peripheriefunktionen bieten.
- **Setzen (pdf \LaTeX):** Der Sourcecode des fertigen Manuskripts muss nun verarbeitet werden. Eine einfache Methode hierfür bietet pdf \LaTeX . Unter Ubuntu kann ein Latexdokument zum Beispiel mit dem Befehl

```
$ pdflatex Dokument.tex
```

kompiliert werden, wodurch neben den Dateien `Dokument.log`, `Dokument.toc` und `Dokument.aux` auch das Dokument `Dokument.pdf` erstellt wird. Die ersten drei Dateien enthalten dabei Informationen über die Struktur des Dokuments (wie zum Beispiel die Informationen zur Erstellung des Inhaltsverzeichnisses) sowie einen log-Eintrag über den Vorgang des Kompilierens. Dabei ist zu beachten, dass pdf \LaTeX das Manuskript linear „abarbeitet“. Das heißt, das Inhaltsverzeichnis wird unter Umständen angelegt, bevor pdf \LaTeX weiß, welche Kapitel es überhaupt gibt. Dann muss das Dokument mehrfach kompiliert werden. Unter allen Betriebssystemen kann zum Beispiel auch das Programm „TexStudio“ oder eine ähnliche Software verwendet werden. Dort gibt es einen Button, mit dem sofort automatisch kompiliert werden kann.

1.2 Ein Minimalbeispiel

Im Folgenden soll ein Minimalbeispiel eines L^AT_EX-Codes erstellt werden, um eine Idee zu vermitteln, wie ein Dokument aufgebaut ist. Jedes Dokument besteht hierbei aus zwei übergeordneten Bereichen:

- **Präambel:** In der sogenannten „Präambel“ werden alle Einstellungen und Definitionen für das Dokument getätigt. Sie fungiert somit im weitesten Sinne als Deklarationsbereich. Hier werden alle Pakete eingebunden, sowie genauere globale Dokumenteinstellungen getätigt.
- **Dokumentbereich:** Das eigentliche Dokument ist der Bereich, in dem der Inhalt des Dokuments zu finden ist. Er beginnt nach der Präambel mit dem Befehl `\begin{document}` und endet mit `\end{document}`. Zwischen diesen Befehlen kann das Dokument erstellt, aber kein Paket mehr hinzugefügt werden. Dies geschieht ausschließlich in der Präambel.

Ein sehr einfaches Beispiel eines Dokuments sähe wie folgt aus:

```

----- Dokument.tex -----
1 \documentclass[12pt, a4paper]{article} % Definition der Dokumentklasse
2 \usepackage[ngerman]{babel}
3 \usepackage[T1]{fontenc}           % Hier werden Pakete eingebunden,
4 \usepackage[utf8]{inputenc}       % in der Form:
5 \usepackage{lmodern}              % \usepackage[options]{package}
6 %Ende der Präambel
7
8 \begin{document}                  % Beginn des Dokuments
9 Hello to the LaTeX-World.         % Text
10 \end{document}                   % Ende des Dokuments
```

In Abschnitt 3 werden die hier verwendeten Pakete und ihr Zweck genauer erläutert., in Abschnitt 4 befinden sich einige Code-Beispiele.

2 Installation und Einrichtung

Bevor Dokumente erstellt werden können, ist es nötig, sich eine L^AT_EX-Distribution zu installieren. Offensichtlich ist es sinnvoll, die neueste Version (L^AT_EX 2_ε) zu verwenden, was bei allen bekannten Distributionen der Fall ist. Im Folgenden werden die bekanntesten Distributionen vorgestellt. Dabei wird sich auf die Verwendung von L^AT_EX über die freie Software TeXStudio¹ bezogen, die auf allen bekannten Betriebssystemen verfügbar ist.

2.1 Windows: MiKTeX

MiKTeX ist eine Latex-Distribution für Microsoft Windows, die besonders einfach zu handhaben ist. Es genügt, sich den Basic-Installer von der Homepage² des Entwicklers herunterzuladen und zu installieren. Die verfügbaren Pakete können nun über den mitgelieferten „Package Manager“ betrachtet und verwaltet werden. Eine angenehme Eigenschaft von MiKTeX ist, dass dieser aber prinzipiell nicht gebraucht wird. Wenn man nämlich z. B. den Editor TexStudio installiert und versucht, ein Dokument zu kompilieren, welches ein nicht installiertes Paket verwenden will, erscheint direkt die Nachfrage, ob das entsprechende Paket installiert werden soll. Die darüber gestartete Installation geschieht automatisch im Hintergrund, ohne dass etwas Zusätzliches getan werden muss. Somit kann jedes in MikTex verfügbare Paket quasi sofort benutzt werden, was im Hinblick auf die Paketverwaltung zunächst keine Wünsche offen lässt. Es ist auch möglich, sich direkt eine vollständige Installation herunterzuladen. Es ist dann - bis auf Updates - nicht mehr nötig, Pakete zu installieren. Es werden allerdings über 2 GB an Speicherplatz benötigt.

MiKTeX bietet somit eine Latex-Distribution, die einfach zu handhaben ist und wenig Arbeitsaufwand bedeutet, sodass man sich auf das Verfassen des Dokuments konzentrieren kann. MiKTeX ist allerdings nur für Windows verfügbar.

2.2 Mac OS X: MacTeX

Für Mac gibt es eine angepasste Version von MikTex namens „MacTex“. (Diese Distribution habe ich nicht verwendet, da ich kein laufendes Betriebssystem OSX habe. Informationen dazu müsste jemand hinzufügen, der sich damit auskennt.)

2.3 Linux: TeXLive

TeXLive ist eine der umfangreichsten Latex-Distributionen, die zudem cross-platform-kompatibel ist. Er soll hier exemplarisch für Linux vorgestellt werden. Unter Ubuntu kann TexLive mit der Paketverwaltung (z.B. dem Advanced Packaging Tool „apt“) über die Shell mittels

```
$ sudo apt-get install texlive texlive-lang-german texlive-latex-extra
```

installiert werden. Hierbei wird die deutsche Sprachunterstützung sowie eine Vielzahl oft verwendeter Pakete installiert. Es ist auch möglich, eine komplette Installation über das Paket `texlive-full` zu erhalten. Damit werden ausnahmslos alle Sprachdateien und Pakete installiert, die zur Verfügung stehen. Wie bei MikTex gilt auch hier: Damit muss in

¹Homepage: <http://texstudio.sourceforge.net/>

²Homepage: <http://miktex.org/>

Zukunft kein Paket mehr installiert werden, die Installation benötigt aber mehr als 2 GB an Speicherplatz auf der Festplatte.

TeXLive liefert einen eigenen Paketmanager (`tlmgr`), mit dem einzelne Pakete installiert werden können. Unter Umständen muss dieser nach der TeXLive-Installation noch mit dem Befehl `$ tlmgr init-usertree` initialisiert werden, wenn `tlmgr` im usermode verwendet werden soll. Mit den Befehlen

```
$ tlmgr install <package1> <package2>
```

und

```
$ tlmgr update -all
```

können Pakete installiert und alle installierten Pakete auf dem aktuellen Stand gehalten werden. In anderen Linux-Distributionen kann diese Vorgehensweise abweichen. So lassen sich in Fedora Pakete direkt mit dem Paketmanager `Yum` installieren. Programme wie `TeXStudio` können mit TeXLive wie bei den anderen Betriebssystemen auf die Latex-Installation zugreifen und dementsprechend automatisch kompilieren.

Unter Windows enthält TeXLive genau wie `MikTeX` einen Paketmanager mit einer GUI, sodass sich das Installieren und Aktualisieren von Paketen so einfach wie nur möglich gestaltet. Lediglich das automatische Installieren von fehlenden Paketen bei Bedarf wird von TeXLive (im Vergleich zu `MiKTeX`) nicht unterstützt (nach dem aktuellen Wissensstands des Autors).

2.4 Vergleich

Da \LaTeX plattformunabhängig ist und sich die Distributionen nur merklich in den Paketverwaltungen unterscheiden, ist es grundsätzlich egal, auf welchem Betriebssystem man \LaTeX verwendet. Es ist unwahrscheinlich, dass ein häufig genutztes Paket in einer Latex-distribution nicht verfügbar ist. Die Unterschiede in der Benutzerfreundlichkeit dürften marginal sein. Viel eher spielt die Wahl des verwendeten Editors eine Rolle. Es empfiehlt sich, einen dedizierten Editor zu verwenden. Die beiden genannten Programme `TeXMaker` und `TeXStudio` eignen sich beide hervorragend und bieten eine übersichtliche und handliche \LaTeX -Umgebung.

3 Erste Schritte mit \LaTeX

Im Folgenden wird auf die grundlegenden Einstellungen beim Erstellen eines \LaTeX -Dokuments eingegangen. Hierbei wird zunächst auf das zuvor gezeigte Minimalbeispiel auf Seite 3 Bezug genommen.

3.1 Die Dokumentklasse

Die erste und grundlegendste Eigenschaft, die in einem neuen Dokument gesetzt wird, ist die sogenannte „Klasse“ des Dokuments. Diese wird mit dem Befehl

```
\documentclass [Optionen] {Klasse}
```

instruiert. Damit werden die allgemeinen Dokumenteigenschaften festgelegt, zum Beispiel, ob es sich um einen kurzen Artikel, ein ganzes Buch, einen Brief oder auch eine Präsentation handelt. Dabei wird auch der Dokumentstil bestimmt, z. B. die Dokumentstruktur (Kapitel, Unterkapitel). Später ist es möglich, sich eigene Klassen zu schreiben, um ein Höchstmaß an Individualität zu erreichen. Für Arbeiten an der Universität sind die Standardklassen aber absolut ausreichend. Für die Protokolle in den Anfängerpraktika und dem Elektronikpraktikum bietet sich die Klasse „article“, für weitergehende Arbeiten unter Umständen schon die Klasse „report“ an.

3.2 Wichtige Pakete

Es sollen nun einige Pakete vorgestellt werden, die in den meisten Dokumenten zu finden sind, weil durch sie grundlegende Eigenschaften festgelegt werden.

3.2.1 Zeichenkodierung und Sprache

L^AT_EX und pdfL^AT_EX unterstützen standardmäßig nur den Ascii-Zeichensatz. Um Sonderzeichen und Umlaute direkt über die Tastatur für die Verwendung im Dokument eingeben zu können, werden Zeichensatz-Encoder benötigt, die mit den Paketen

```
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
```

eingebunden werden. Diese Pakete sind deshalb quasi eine Notwendigkeit. Hiermit können Zeichen des Unicodes (z.B. ä, ö, ü) intuitiv verwendet werden. Zusätzlich sollte das Paket

```
\usepackage{lmodern}
```

verwendet werden, um die Type-1 Schriftart in pdf-Dateien schöner darzustellen (was durch die hohe Geschwindigkeit heutiger Rechner im Gegensatz zu früher möglich ist).

L^AT_EX wurde an und für sich für die Sprache Englisch entwickelt. Da das Textsatzsystem T_EX die Formatierung des Textes übernimmt, muss pdfL^AT_EX mitgeteilt werden, wie die verwendete Sprache behandelt werden soll. Dabei geht es hauptsächlich um die korrekte Worttrennung und länderspezifische Formatkonventionen wie die Darstellung eines Datums. Hierzu reicht es, das Paket

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

einzubinden. Die Option `ngerman` unterscheidet sich im Vergleich zu `german` im Hinblick auf die neue deutsche Rechtschreibung.

Griechische Buchstaben werden wegen ihrer häufigen Verwendung als Variablen von L^AT_EX automatisch kursiv geschrieben, was sich für ihre Einbindung in Fließtext mit dem Paket

```
\usepackage{lmodern}
```

beheben lässt. Nun können griechische Buchstaben mit einem vorangesetzten 'up' aufrecht geschrieben werden, zum Beispiel bei einem Alpha mit `\upalpha` anstatt `\alpha`.

3.2.2 Rand- und Formateinstellungen

Wenn die Textweite bzw. die Randbreite für jeden Rand individuell eingestellt werden soll, kann das Paket

```
\usepackage[top=0cm, bottom=0cm, left=0cm, right=0cm]{geometry}
```

verwendet werden. Dadurch lassen sich die Randabstände einzeln in Zentimetern oder anderen Einheiten angeben (cm = Zentimeter, mm = Millimeter, pt = Punkt, in = Inches, em = Breite eines großen 'M' in der aktuellen Schriftart).

Wenn ein einzelner Abschnitt oder eine einzelne Grafik im Querformat dargestellt werden soll, kann man dies mit dem Paket

```
\usepackage{lscape}
```

erreichen. Einzelne Passagen werden dabei in die Befehle `\beginlandscape` und `\endlandscape` eingefasst.

3.2.3 Pakete für den mathematischen Textsatz

Um die Unterstützung für den mathematischen Formelsatz zu optimieren, empfiehlt es sich, erweiternde Pakete über

```
\usepackage{amsmath}  
\usepackage{amsfont}  
\usepackage{amssymb}
```

einzubinden. Hierdurch wird der mathematische Zeichensatz ergänzt, neue Umgebungen wie `align` werden hinzugefügt (vgl. Abschnitt 4.4) und die Darstellung von Formeln wird optimiert.

Für die mathematische Umgebung als auch im normalen Fließtext bietet sich gerade für physikalische Protokolle das Paket

```
\usepackage[separate-uncertainty = true,output-decimal-marker={,}]{siunitx}
```

für die korrekte Darstellung von Einheiten an, wobei in den eckigen Klammern optional der Punkt bei englisch dargestellten Dezimalzahlen in ein Komma umgestellt wird. Damit werden Einheiten aufrecht anstatt wie mathematische Variablen kursiv angezeigt und ein halbes Leerzeichen zwischen dem Wert und seiner Einheit eingefügt. Auch die Angabe von Fehler bzw. Unsicherheiten lässt sich hiermit ganz einfach durchführen.

3.2.4 Grafiken einfügen

Wenn Skizzen, Grafiken und Plots eingefügt werden sollen, benötigt man das Paket

```
\usepackage{graphicx}
```

für die Einbindung der Dateitypen `.jpg`, `.png` und `.pdf`, wobei man aber den Dateitypen `.eps` nicht verwenden kann. Für letzteres kann das Paket `epstopdf` eingebunden werden,

welches intern .eps-Dateien in .pdf-Dateien konvertiert. Diese Methode lässt sich mit pdfL^AT_EX hervorragend verwenden.

3.2.5 Zitate und Quellenangaben

Insbesondere für Grafiken werden häufig Quellenangaben benötigt, die mit dem Paket

```
\usepackage{cite}
```

leichter erfolgen können. Dazu fügt man die jeweilige Quelle am Ende des Dokuments in einer Literaturverzeichnis zwischen `\begin{thebibliography}{x}` und `\end{thebibliography}` ein. Dabei ist x die maximale Anzahl der Einträge und kann z. B. standardmäßig auf 99 gesetzt werden. Ein Eintrag im Literaturverzeichnis folgt dem Schema

```
\bibitem{Bezeichnung} Beschreibung  
\url{Internetlink}, Zugriff am 01.01.2015.
```

Dieser Eintrag kann an passender Stelle zitiert werden mit `\cite{Bezeichnung}`.

4 Beispiele

Abschließend sollen einige der vorgestellten Pakete praktisch angewandt werden. Es wird dargelegt, wie Strukturen in Grafiken und Tabellen erstellt werden, die häufig benötigt werden.

4.1 Einfache Strukturierung

Einfache Strukturierungsmethoden wie Absätze, Leerzeilen, ein Seitenumbruch oder ein Zeilenumbruch werden in L^AT_EX selbstverständlich etwas anders gehandhabt als in Word - dafür weiß man auch meistens, was weshalb passiert.

- Wenn man den Befehl `\` im Code nach einem Satz platziert, wird ein manueller Zeilenumbruch durchgeführt. Wird im Code zusätzlich danach eine Leerzeile gelassen, so geschieht dies auch im Dokument. Der nachfolgende Absatz wird dann automatisch eingerückt. (Dieses automatische Einrücken lässt sich mit dem Befehl `\noindent` am Anfang des Absatzes unterdrücken.)
- Mit dem Befehl `\newpage` wird ein Seitenumbruch erzwungen.
- Wird der Befehl `\tableofcontents` am Anfang³ des Dokuments platziert, wird automatisch ein Inhaltsverzeichnis erstellt. Mit den Befehlen `\listoffigures` bzw. `\listoftables` kann ein automatisches Abbildungs- bzw. Tabellenverzeichnis erstellt werden.
- Die Dokumentklasse `article` enthält genügend Strukturbefehle für die physikalischen Praktika. Mit dem Befehl `\section{Titel}` kann ein Kapitel begonnen werden. Die Unterstrukturierung geschieht mit den Befehlen `\subsection{Titel}` und `\subsubsection{Titel}`. Welche Unterstruktur noch im Inhaltsverzeichnis zu sehen sein soll, kann pauschal manuell eingestellt werden. Dies und vieles mehr (zu viel an dieser Stelle) kann mit dem Befehl `\setcounter{Zähler}{Wert}` eingestellt werden. **Hinweis:** Wenn ein Abschnitt nicht nummeriert werden soll, kann der Befehl `\section*` verwendet werden, wodurch dieser Abschnitt zusätzlich aus dem Inhaltsverzeichnis verschwindet.

³Nach dem Befehl `\begin{document}` und dem Deckblatt

4.2 Grafik mit einer Bildunterschrift einfügen

Um Grafiken einzubinden, verwendet man das zuvor eingeführte Paket `graphicx`. Grafiken (und auch Tabellen) werden üblicherweise in sogenannten „floating objects“ erstellt. Eine solche Gleitumgebung wird von pdfL^AT_EX automatisch formatiert und an einer passenden Stelle im Text platziert. Im Fall einer Grafik wird üblicherweise die Umgebung `figure` verwendet. Im Folgenden ist ein Beispielcode dargestellt:

```
Dokument3.tex
1 (...)
2 \begin{figure}[ht]
3   \centering
4   \includegraphics[scale = 0.5]{Bild.pdf}
5   \caption{Das ist das Uni-Logo.}
6   \label{fig:bild}
7 \end{figure}
8 Wie in Abbildung \ref{fig:bild} zu erkennen ist, (...)
```

Der Block zwischen `\begin{figure}[ht]` und `\end{figure}` stellt die Gleitumgebung dar. Der Parameter `[ht]` steht für „here, top“ und bedeutet, dass die Figur im Dokument genau an der gleichen Stelle wie im Code platziert wird oder falls das nicht optimal sein sollte am Anfang der Seite („top“).⁴ Mit `\centering` wird die Umgebung zentriert, sodass die Grafik und ihre Unterschrift mittig dargestellt werden. Mit `\label` lässt sich ein Referenzpunkt erstellen. Zum Zeitpunkt des Erstellens des Textes kann der Autor noch nicht wissen, welche Nummer die Abbildung im fertigen Dokument bekommen wird. Auf diese Weise kann man sich mittels `\ref{fig:Skizze}` auf die noch unbekannte Nummer der Abbildung beziehen. Im fertigen Dokument wird die Referenz dann durch die entsprechende Zahl ersetzt. Solche Referenzen kann man ebenfalls bei Formeln, ganzen Kapiteln oder an jeder anderen Stelle im Dokument verwenden. Insgesamt ergäbe der obige Codeabschnitt die folgende Ausgabe:

⁴Mit dem Paket `float` steht der Parameter `[H]` zur Verfügung, der das Einbringen an exakt der gleichen Stelle wie im Code erzwingt.

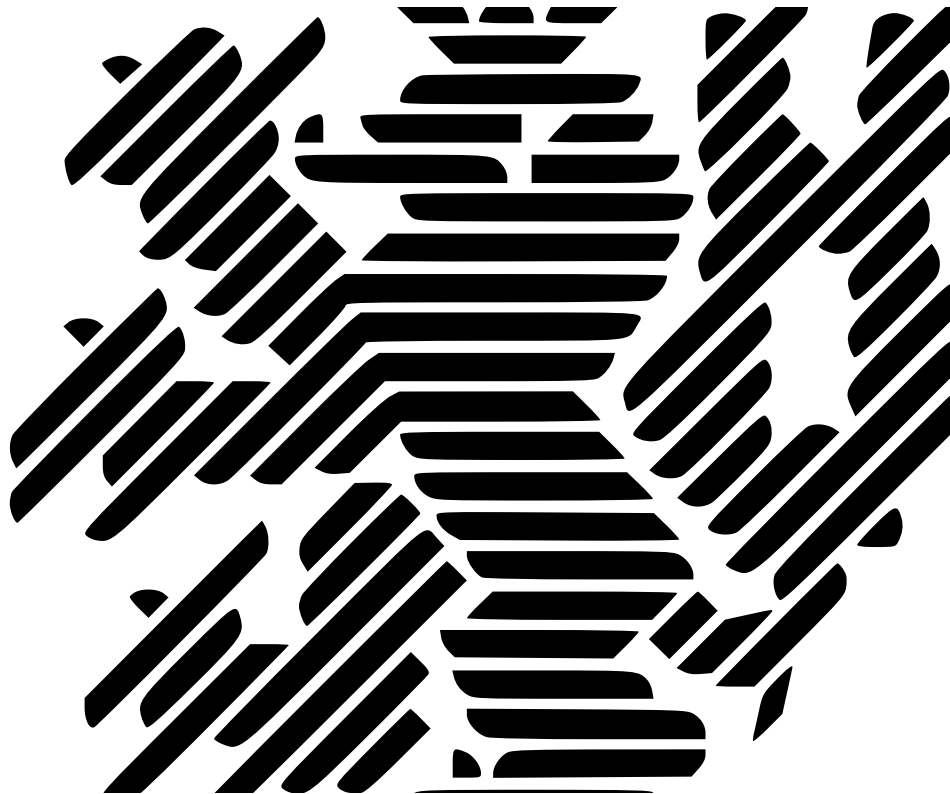


Abbildung 1: Das ist das Uni-Logo.

Wie in Abbildung 1 zu erkennen ist, (...)

Nach diesem Prinzip können beliebige Grafiken eingefügt werden. Der Stil der Bildunterschrift kann in der Präambel in den Paketoptionen des Pakets `caption` angepasst werden. Es ist zum Beispiel möglich, den Bezeichner „Abbildung 1“ fett zu drucken oder in „Figur 1“ zu ändern, wenn dies gewünscht ist.

Für fortgeschrittene Nutzer gibt es zum Beispiel die Möglichkeit, mehrere Abbildungen mit einzelnen „Unterüberschriften“ nebeneinander zu platzieren - um zum Beispiel einen Vorher-Nachher-Effekt zu verdeutlichen oder den Einfluss eines bestimmten Parameters auf Messergebnisse zu visualisieren. Dies kann mit dem Paket `subcaption` erledigt werden.

4.3 Tabellen erstellen

Tabellen werden wie Grafiken üblicherweise in einer Gleitumgebung (`table`) erstellt. TeX-Studio liefert für die Erstellung von Tabellen einen GUI-Assistenten, was einem lästige Arbeit ersparen kann. Alternativ gibt es Makros für Microsoft Excel bzw. LibreOffice Calc, mit denen man eine Tabelle aus dem jeweiligen Programm automatisch in \LaTeX -Code übersetzen kann. Dies ist oft sehr nützlich für das Tabellarisieren vieler Daten und hat den Vorteil, dass man Rechnungen in dem jeweiligen Kalkulationsprogramm durchführen und die Werte trotzdem schnell in das Dokument einfügen kann. Das folgende Beispiel soll das generelle Erstellen von Tabellen in \LaTeX verdeutlichen.

```

----- Dokument4.tex -----
1 | \begin{table}[h]
2 | \centering
```

```

3 \caption{Messdaten der Diodenkennlinie}
4 \label{tab:Daten}
5 \begin{tabular}{c|c|c}
6 R / Ohm & U / V & I / mA \\ \hline
7 80 & 0.594(5) & 0.764(1) \\
8 100 & 0.617(5) & 1.335(1) \\
9 150 & 0.659(5) & 3.61(1) \\
10 200 & 0.673(5) & 5.01(1) \\
11 250 & 0.706(5) & 11.0(1) \\
12 300 & 0.709(5) & 11.7(1) \\
13 400 & 0.715(5) & 13.4(1) \\
14 \end{tabular}
15 \end{table}
16 Wie in Tabelle \ref{tab:Daten} zu sehen ist, ...

```

- Mit `\begin{table}` bzw. `\end{table}` wird die Gleitumgebung für die Tabelle definiert. Diese sowie der Platzierungsoperator `[h]` verhalten sich analog zu der `figure`-Umgebung (siehe Abschnitt 4.2).
- Mit den Befehlen `\caption` und `\label` werden die Tabellenüberschrift und die Referenz auf die Tabellennummer erstellt, auf die mit `\ref` verwiesen werden kann (vgl. Code oben). Man beachte hier, dass der `Caption`-Befehl auch in der Manuskriptstruktur über der Tabelle angeordnet ist im Gegensatz zu Grafiken, wo dieser unter der eingefügten Grafik angesiedelt wird.
- Die Tabelle an sich wird mit einer `tabular`-Umgebung erstellt. In der Struktur `{c|c|c}` stehen die Buchstaben jeweils für eine Spalte, und die senkrechten Striche für die Trennungslinien zwischen den Spalten. „c“ bedeutet hier „centered“, also wird der Inhalt in jeder Zelle der Tabelle zentriert. Die Alternativen sind „l“ für left und „r“ für right. Der Inhalt der Tabelle wird zeilenweise angegeben, und die Inhalte zweier benachbarter Zellen werden mit einem `&` getrennt. Für eine horizontale Linie kann der Befehl `\hline` verwendet werden.

Das obige Beispiel ergibt schlussendlich die folgende Tabelle:

Tabelle 1: Messdaten der Diodenkennlinie

R / Ohm	U / V	I / mA
80	0.594(5)	0.764(1)
100	0.617(5)	1.335(1)
150	0.659(5)	3.61(1)
200	0.673(5)	5.01(1)
250	0.706(5)	11.0(1)
300	0.709(5)	11.7(1)
400	0.715(5)	13.4(1)

Wie in Tabelle 1 zu sehen ist, (...)

4.4 Mathematikmodus und Formeln

EINHEITEN EINFÜHREN Eine der wichtigsten und besten Funktionen von Latex ist der Formelsatz. Für gewöhnlich werden Formeln und mathematische Symbole auf zwei Arten in das Dokument eingebracht: als Umgebung oder inline. Um innerhalb einer Zeile in den Mathematik-Modus zu wechseln, benutzt man die Syntax $c^2 = a^2 + b^2$, was zu der Formel $c^2 = a^2 + b^2$ führt, die fließend in den Text eingebunden wird. Dies ist natürlich nur für kleine Formeln sinnvoll.

Um eine eigene Zeile für eine oder mehrere Formeln zu benutzen, gibt es Umgebungen wie `equation` oder `align`. Letztere kommt aus dem bereits erwähnten Paket `amsmath` und soll hier exemplarisch vorgestellt werden. Innerhalb einer Mathematikumgebung müssen die `$`-Zeichen nicht mehr verwendet werden. Latex befindet sich innerhalb der gesamten Umgebung im Mathematikmodus. Man betrachte hierfür das folgende Beispiel:

```
----- Dokument5.tex -----
1 \begin{align}
2 E &= m \cdot c^2 \\
3 \nonumber E^2 &= \left( pc \right)^2 + \left( mc^2 \right)^2 \\
4 \beta &= \left( \frac{v}{c} \right) & \quad \\
5 \gamma &= \left( \frac{1}{\sqrt{1 - \frac{v^2}{c^2}}} \right) \\
6 \end{align}
```

Dieser Code ergibt die folgende Ausgabe:

$$E = m \cdot c^2 \tag{1}$$
$$E^2 = (pc)^2 + (mc^2)^2$$
$$\beta = \left(\frac{v}{c} \right) \quad \gamma = \left(\frac{1}{\sqrt{1 - \frac{v^2}{c^2}}} \right) \tag{2}$$

An diesem Beispiel erkennt man zunächst im Code, dass Formeln in \LaTeX schnell unübersichtlich werden können. Daran gewöhnt man sich aber nach kurzer Zeit. Die Übersicht kann durch Absätze im Code erhöht werden, die keine Auswirkungen auf die Ausgabe der Formel im Dokument haben. Im obigen Code wird z. B. ein Absatz nach `\quad` gemacht. Wie in normalem \LaTeX -Code auch fügt man in der Ausgabe eine weitere Zeile für eine Formel mit dem Befehl `\\` ein. Die Formeln werden automatisch zentriert und nummeriert. Letzteres kann mit dem Befehl `\nonumber` verhindert werden. Alternativ kann man mit dem Code `align*` anstelle von `align` generell die Nummerierung innerhalb dieser Mathematik-Umgebung unterdrücken. Die `align`-Umgebung richtet die Formeln an der Stelle aus, wo sich das Zeichen `&` befindet. Sollen zwei Formeln so ausgerichtet werden, dass die Gleichheitszeichen übereinander stehen, sollte also wie im Beispiel in Zeile 2 und 3 vorgegangen werden.

In Formel (2) lässt sich der Unterschied zwischen der Verwendung von normalen Klammern `()` und mathematischen Klammern `\left(` und `\right)` betrachten. Nur bei den mathematischen Klammern (im Beispiel die rechte Formel) passt sich die Klammergröße automatisch an die Formelgröße an.

4.5 Einheiten

Für die Darstellung von Messwerten mit deren Unsicherheiten und den passenden Einheiten gibt es verschiedene Möglichkeiten, die hauptsächlich auf drei Varianten aufbauen. Man kann eine Einheit für sich angeben (mit `\si{Einheit}`), einen Wert inklusive Unsicherheit mit einer Einheit (mit `\SI{Wert(Unsicherheit)}{Einheit}`) oder eine dimensionslose Zahl inklusive Unsicherheit (mit `\num{Wert(Unsicherheit)}`). Im folgenden Beispiel werden einige Optionen gezeigt:

```

_____ Dokument6.tex _____
1 v = \SI{2,537(12)}{m\per s\squared} \\
2 $\eta$ = \num{0,28(1)} \\
3 Die Dichte $\rho$ hat die Einheit \si{kilogram \per \meter\cubed}.
```

Dieser Code ergibt die folgende Ausgabe:

```

v = (2,537 ± 0,012) m/s2
η = 0,28 ± 0,01
Die Dichte ρ hat die Einheit kg m-3.
```

Im obigen Beispiel wird erkenntlich, dass Einheiten mit ihrer Abkürzung oder nach einem Backslash ausgeschrieben verwendet werden können. Potenzen können mit `\squared` oder `\cubed` angegeben werden.

5 Referenzen

Da Latex weltweit von vielen Anwendern genutzt wird, gibt es reichlich Referenzen und Dokumentationen im Internet. Wenn man ein spezifisches Problem hat, hilft eine Suchmaschine häufig weiter. Zur allgemeinen Weiterbildung werden diesbezüglich besonders interessante Quellen vorgestellt:

- Das Paketarchiv „Comprehensive TeX Archive Network“, kurz CTAN, beinhaltet als Paketquelle zurzeit knapp 5000 Pakete - und zu den meisten existiert eine Dokumentation. Bei allgemeinem Interesse zu einem Paket ist diese Seite sicherlich eine der ersten Anlaufstellen. Homepage: <https://www.ctan.org/>
- Das \LaTeX -Wikibook ist eine gut strukturierte und auf die wichtigen Aspekte fokussierte \LaTeX -Dokumentation. Wenn man ein spezifisches Problem mit \LaTeX bei einer Suchmaschine eingibt, landet man oft hier. Es eignet sich hervorragend, um auch so einmal einen Blick darauf zu werfen. Das Wikibook findet sich hier: <http://en.wikibooks.org/wiki/LaTeX> (Es gibt auch eine deutsche Version, die aber bei weitem nicht so umfangreich ist.)
- Eine gute und vor allem kurz gehaltene Befehlsreferenz, auf die man oft auch über Suchmaschinen gelangt, ist <http://www.weinelt.de/latex/>
- Zu guter Letzt noch „A not so short introduction to $\LaTeX 2_{\epsilon}$ “. Dies ist ein sehr technisch orientiertes und detailliertes Referenzwerk, aus dem man sich dennoch einige Dinge aneignen kann. Homepage: <https://tobi.oetiker.ch/lshort/lshort.pdf>