

Marie Curie Schule
Am Sportplatz 1
30952 Ronnenberg

Facharbeit
im Seminarfach

Kleines L^AT_EX Tutorial Teil II

Verfasser: Asja Lippert und Wiebke Matteikat

Fachlehrer: Karsten Hohaus

Abgabetermin: 22. Dezember 2006

Vorwort

Dieses Tutorial befasst sich mit dem Programm \LaTeX und ist in Anlehnung an „ \LaTeX Das Praxisbuch“¹ geschrieben. Zudem wird das Grundwissen aus dem „Kleinen \LaTeX Tutorial Teil I“² vorausgesetzt. In diesem Tutorial wird speziell auf die Teilgebiete „Tabellen“, „Abbildungen“, „das Beschriften von Tabellen und Abbildungen“ sowie „Das Einbinden von Formeln“ eingegangen. Um allerdings einen möglichst leichten Einstieg in die Themengebiete zu gewährleisten, sind die nötigsten Informationen und Anwendungen beim Anfertigen von Tabellen noch einmal in „Grundlagen einfacher Tabellen“ zusammengefasst.

Tabellen mit fester Gesamtbreite werden in diesem Tutorial nicht behandelt, da dies durch die Komplexität den Rahmen einer Facharbeit sprengen würde. Außerdem fehlen im Umgang mit Tabellen über mehrere Seiten die Parameterangaben, da diese in der Quelle nicht ausreichend erklärt wurden und so unverständlich blieben. Genauso wurde das Themengebiet „Schräge / Senkrechte Beschriftung in Tabellen“ in diesem Tutorial nicht behandelt, da es auch nach langer intensiver Beschäftigung noch unverständlich und fehlerhaft blieb. Im Themengebiet „Bilder aus Dateien einbinden“ werden nur die wichtigsten Parameter angegeben. So ist das Festlegen der Dateierweiterung und des Suchpfades für Abbildungen nicht erwähnt und auch Hintergrundbilder werden in diesem Tutorial nicht behandelt. Das Drehen von Bildern ist nur mit dem Parameter „angle“ und nicht mit der „rotate-box“ aufgeführt.

Beim „Einbinden von Formeln“ wird das `\usepackage{amsmath}` nicht vorausgesetzt. Nicht alle Bearbeitungsmöglichkeiten von Wurzeln, Brüchen und der Gesamtformel werden angesprochen. Zudem werden komplexere Formeln dem Formeleditor überlassen, der selbsterklärend ist.

Wie schon im „Kleinen \LaTeX Tutorial I“ befinden sich auch in diesem Tutorial Übungsaufgaben. Allerdings sind Übungsaufgaben nur zum Themengebiet „Formeln“ gestellt. Da es zum Erstellen komplizierter aufgebauter Tabellen sehr viele unterschiedliche Befehle vor allem zum Verwenden der Linien gibt, ist es leider nicht möglich zu diesem Themengebiet Übungsaufgaben zu stellen. Dennoch können die Tabellen aus 1. nachempfunden werden und daraus selbstständig eigene Tabellen erstellt werden. Zum Themengebiet „Abbildungen“ sind keine Übungsaufgaben vorhanden, da die Abbildungen und das Script im gleichen Ordner gespeichert sein müssen um das Anzeigen der Abbildungen zu ermöglichen. Um das Beschriften von Tabellen und Abbildungen zu üben, können die selbst erstellten Tabellen anhand der gegebenen Befehle beschriftet werden.

¹Verfasser: Elke und Michael Niedermair; Auflage 2005

²Verfasser: Jan Betting und Niklas Peschke; 2006

Inhaltsverzeichnis

1 Tabellen	4
1.1 Grundlagen einfacher Tabellen	4
1.2 Schwierigere Tabellen	6
1.2.1 Layout aufwerten	6
1.2.2 Zellen zusammenfassen	14
1.2.3 Tabellen über mehrere Seiten	15
2 Abbildungen	16
2.1 Bilder aus einer Datei einbinden	16
2.2 Gleitende Abbildungen	18
2.3 Bilder im Text platzieren	19
3 Beschriften von Tabellen und Abbildungen	25
4 Grundlagen für das Schreiben von Formeln	26
4.1 Grundlegende Mathematik-Schreibweisen	26
4.2 Die Elemente von Formeln	27
4.3 Gestaltung von Formeln	34
5 Übungsaufgaben	35
6 Nachwort	37
7 Literaturverzeichnis	38
8 Anhang	39
8.1 Lösungen der Übungsaufgaben	39

1 Tabellen

1.1 Grundlagen einfacher Tabellen

Um mit \LaTeX Tabellen zu erstellen wird folgender Befehl verwendet:

```
\begin{tabular}{Spaltenform}
...Tabelleninhalte...
\end{tabular}
```

Die wichtigsten Dinge zum Anfertigen von einfachen Tabellen werden hier noch einmal erklärt:

- **Spaltenform**

Gleich zu Beginn einer Tabelle wird durch Eingabe von Buchstaben zeitgleich festgelegt, wie viele Spalten es gibt und wie dessen Inhalte ausgerichtet werden sollen. Soll eine Tabelle beispielsweise drei Spalten haben, so könnte dies der Befehl sein:

```
\begin{tabular}{l c r}
...Tabelleninhalt...
\end{tabular}\
```

Die Buchstaben haben dabei folgende Bedeutung:

- l: der Spalteninhalt wird linksbündig gesetzt.
- c: der Spalteninhalt wird zentriert gesetzt.
- r: der Spalteninhalt wird rechtsbündig gesetzt.

- **Tabulatorstopp**

Der Tabulatorstopp wird in \LaTeX verwendet, um zwischen den Spalten einer Tabelle einen bestimmten Abstand anzulegen. Der einzugebene Befehl dazu lautet:

```
\hspace{Abstand}
```

Folgendes Beispiel erklärt weitere nötige Befehle:

Note	Mädchen	Jungen	Gesamt
1	1	1	2
2	1	3	4
3	3	5	8
4	5	6	11
5	1	1	2
Gesamt	11	16	27

```
\begin{tabular}{|l||c|c||r|}
\hline    Note & Mädchen & Jungen
& Gesamt \\
\hline    1 & 1 & 1 & 2 \\
\hline    2 & 1 & 3 & 4 \\
\hline    3 & 3 & 5 & 8 \\
\hline    4 & 5 & 6 & 11 \\
\hline    5 & 1 & 1 & 2 \\
\hline    Gesamt & 11 & 16 & 27 \\
\hline \end{tabular}
```

Abbildung 1: Erklärungsbeispiel

- |

Der senkrechte Strich kann vor, zwischen und hinter den Buchstaben l, c und r in der “Spaltenform“ platziert werden. Er hat dabei die Aufgabe die Spalten durch senkrechte Linien voneinander zu trennen. Außerdem besteht die Möglichkeit durch mehrfache Eingabe des Befehls auch beliebig viele senkrechte Linien zur Spalten-trennung zu erreichen.

- &

Wird zum Trennen der Spalten verwendet.

- \\

Fügt einen Zeilenumbruch ein.

- \hline

Der Befehl \hline wird zum Einfügen von horizontalen Linien zwischen den Zeilen verwendet. Es besteht auch hier die Möglichkeit durch mehrfache Eingabe des Befehls beliebig viele horizontale Linien zur Zeilentrennung zu erhalten.

1.2 Schwierigere Tabellen

1.2.1 Layout aufwerten

Um mit \LaTeX optisch anspruchsvollere Tabellen zu erstellen, sollte am Anfang jedes tabellenenthaltenden Dokuments ein weiteres Ergänzungspaket eingebunden werden.

```
\usepackage{booktabs}
```

Unterschiedliche Linien

Zu einer optisch anspruchsvolleren Tabelle trägt auch die Anwendung von unterschiedlichen Linien bei. \LaTeX bietet dabei folgende Möglichkeiten:

- **Linie für den Tabellenkopf**

Eine horizontale Linie für den Tabellenkopf mit einem größeren Zeilenabstand unterhalb der Linie, entsteht durch den folgenden Befehl:

```
\toprule
```

Die Dicke der Linie kann mit dem Befehl `\heavyrulewidth` verändert werden, was bereits am Anfang des Dokuments geschehen sollte, da so automatisch alle Tabellen, die den Befehl `\toprule` enthalten einheitlich dick gezeichnet werden.

Der folgende Befehl veranschaulicht die Eingabe im Script:

```
\setlength{\heavyrulewidth}{1pt}
```

Ausschlaggebend für die Dicke der Linie ist allerdings nur der Inhalt der letzten Klammer (hier 1pt).

- **Linie mit größerem Zeilenabstand**

Der folgende Befehl wirkt ähnlich wie der Befehl `hline`. Der Unterschied besteht darin, dass ein größerer Zeilenabstand definiert wird. Außerdem kann mit dem Befehl `lightrulewidth` die Liniendicke verändert werden.

```
\midrule
```

- **Linie für den Tabellenfuß**

Der folgende Befehl wird nur für den Tabellenfuß verwendet. Auch hier wird ein größerer Zeilenabstand definiert.

```
\bottomrule
```

- **Verkürzte Linie**

Eine verkürzte Linie, die also nicht von der linken bis zur rechten Grenze der Tabelle verläuft, kann mit dem folgenden Befehl erstellt werden. Ein zusätzlicher Leerraum wird mit `\extracolsep` eingefügt.

```
\cmidrule(trim){n-m}
```

Für den Parameter „trim“ wird die Platzierung der verkürzten Linie in Form der Buchstaben `l`, `l{breite}`, `r` oder `r{breite}` eingetragen.

- **Doppellinie**

Um zwei aufeinanderfolgende verkürzte Linien zu erstellen, ist es nicht möglich einfach doppelt den Befehl `\cmidrule{trim}{n-m}` einzufügen, da im Skript ein weiterer Befehl zwischen den Doppelbefehl gestellt werden muss. Dieser lautet:

```
\morecmidrules
```

Im Skript würde dies zum Beispiel folgendermaßen aussehen:

```
\cmidrule(r){1-2} \morecmidrules \cmidrule(r){1-2}
```

- **Zusätzlicher Zeilenabstand**

Ein zusätzlicher Zeilenabstand kann auch eingefügt werden, ohne dass der Befehl `\cmidrule(trim){n-m}` verwendet wird. Der dafür verwendete Befehl muss nach `\\` folgen.

```
\addlinespace[länge]
```

Folgendes Beispiel zeigt, welche deutlich positiven Veränderungen die Anwendung unterschiedlicher Linien mit sich bringt.

Vorher

Eiskarte		
Eis	Zubereitung	Preis (Euro)
1 Kugel	In der Waffel	0,70
2 Kugeln	In der Waffel	1,20
3 Kugeln	In der Waffel	1,80

```
\begin{tabular}{l l r} \hline
\multicolumn{2}{c}{Eiskarte} \\
\cline{1-2} Eis & Zubereitung &
Preis (Euro) \\ \hline
1 Kugel & In der Waffel & 0,70 \\
2 Kugeln & In der Waffel & 1,20 \\
3 Kugeln & In der Waffel & 1,80 \\
\hline
\end{tabular}
```

Abbildung 2: Tabelle ohne Gebrauch unterschiedlicher Linien

Nachher

Eiskarte		
Eis	Zubereitung	Preis (Euro)
1 Kugel	In der Waffel	0,70
2 Kugeln	In der Waffel	1,20
3 Kugeln	In der Waffel	1,80

```
\begin{tabular}{l l r} \toprule
\multicolumn{2}{c}{Eiskarte} \\
\cmidrule(r){1-2} Eis &
Zubereitung & Preis (Euro) \\
\midrule
1 Kugel & In der Waffel & 0,70 \\
2 Kugeln & In der Waffel & 1,20 \\
3 Kugeln & In der Waffel & 1,80 \\
\bottomrule
\end{tabular}
```

Abbildung 3: Tabelle mit unterschiedlichen Linien

Ausrichtung der Spalte am Dezimaltrenner

Die Ausrichtung der Spalte am Dezimaltrenner bedeutet, dass im Deutschen alle Kommas, im Amerikanischen alle Punkte, die die ganze Zahl von ihren Dezimalstellen trennen, in der Spalte genau untereinander stehen.

In \LaTeX wird dazu ein weiteres Ergänzungspaket benötigt. Dieses baut auf dem Paket `array` auf.

```
\usepackage{dcolumn}
```

Durch das Ergänzungspaket ist ein neuer Spaltentyp verfügbar, der folgendermaßen aussieht:

```
\D{eingabe-trenner}{ausgabe-trenner}{kommastellen}
```

Für den „eingabetrenner“ wird das Zeichen angegeben, das im Script als Dezimaltrenner verwendet wird.

Für den „ausgabebetrenner“ wird ein beliebiges Zeichen eingegeben, welches bei der Ausgabe des Dokuments als Dezimaltrenner erscheinen soll.

Bei „kommastellen“ wird eine beliebige Zahl eingegeben, die angibt wieviele Nachkommastellen die Zahlen bekommen sollen und so zeitgleich die Spaltenbreite mitbestimmt.

Das nachfolgende Beispiel soll das Ausrichten der Dezimalspalte noch einmal verdeutlichen:

Eiskarte		
Eis	Zubereitung	Preis (Euro)
1 Kugel	In der Waffel	0,70
100 Kugeln	In der Waffel	50,230
500 Kugeln	In der Waffel	100,8940

```
\begin{tabular}{1 l D{,}{,}{-1}}
\toprule
\multicolumn{2}{c}{Eiskarte} \\
\cmidrule(r){1-2}
Eis & Zubereitung &
\multicolumn{1}{c}{Preis (Euro)}
\\
\cmidrule(r){1-1}
\cmidrule(lr){2-2}
\cmidrule(l){3-3}
1 Kugel & In der Waffel & 0,70 \\
100 Kugeln & In der Waffel & 50,230 \\
500 Kugeln & In der Waffel & 100,8940 \\
\bottomrule
\end{tabular}
```

Abbildung 4: Erläuterung des Spaltentyps D

❗: Die Spalte der Tabelle, in der die Dezimaltrenner übereinanderstehen sollen, in der also der Spaltentyp D verwendet wird, muss zusätzlich mit `\multicolumn` (siehe Zeile 6 Script) einem anderen Spaltentyp zugeordnet werden, damit nicht die Überschrift am Komma platziert wird.

Zahlen automatisch runden

Mit \LaTeX ist es möglich, dass Zahlen einer Spalte auf eine bestimmte Anzahl Nachkommastellen gerundet werden. Dazu wird ein weiteres Ergänzungspaket benötigt, welches auf dem Paket `fltpoint` aufbaut.

```
\usepackage{rccol}
```

Durch das Einfügen dieses Ergänzungspakets entsteht ein neuer Spaltentyp `R`. Dieser sieht folgendermaßen aus:

```
R[eingabetrenner][ausgabetrenner]{vorkommastellen}{nachkommastellen}
```

Die optionalen Parameter:

- Runden der Zahlen aktivieren

```
rounding
```

- Runden der Zahlen deaktivieren

```
norounding
```

- Komma als Dezimaltrenner

```
comma (Alternativeingabe: german)
```

- Punkt als Dezimaltrenner

```
point (Alternativeingabe: english oder USenglish)
```

Zwei der optionalen Parameter können in die eckigen Klammern für den Spaltentyp `R` eingetragen werden. Wird dies nicht getan, so arbeitet `LATEX` mit den Voreinstellungen des Ergänzungspakets.

Der Begriff `eingabetrenner` gibt das Zeichen an, welches innerhalb der Tabelle als Dezimaltrenner verwendet wurde.

Der Begriff `ausgabetrener` gibt an, welches Zeichen bei der Ausgabe des Dokuments gedruckt werden soll.

In die erste geschweifte Klammer sollten die Vorkommastellen der größten gebrauchten Zahl eingegeben werden, da sonst das Anzeigen dieser Zahl nicht möglich ist. Alle anderen Zahlen werden dann angepasst.

Je nachdem für welchen Zweck die Werte der Tabelle gebraucht werden (Preistabelle in Euro, Messungen eines physischen Versuchs, ...), sollte entschieden werden wieviele Nachkommastellen für den Inhalt der Spalte am Geeignetsten sind und diese Zahl in die zweite geschweifte Klammer eingetragen werden.

Folgendes Beispiel zeigt die Anwendung des Spaltentyps `R`:

Artikel	Größe	Preis (Euro)
Jeans	S	19,99
Jeans	M	21,93
Nerz	L	195,70

```

\begin{tabular}{l c R{3}{2}}
\toprule
Artikel & Größe & 
\multicolumn{1}{c}{Preis (Euro)} \\
Jeans & S & 19,98765 \\
Jeans & M & 21,92785 \\
Nerz & L & 195,6995 \\
\bottomrule
\end{tabular}

```

Abbildung 5: Erläuterung des Spaltentyps `R`

Farbig gestaltete Tabellen

Da es bei Arbeiten mit größeren oder auch kleineren Tabellen häufig vorkommt, dass der Leser zwischen den Zeilen der Tabelle verrutscht, ist es möglich die Lesbarkeit durch farbliche Unterscheidung der Zellen zu verbessern.

Zwei weitere Ergänzungspakete sind nötig um eine farbliche Unterscheidung zu ermöglichen:

```
\usepackage{xcolor}
\usepackage{colortbl}
```

Die folgenden Befehle erlauben die farbige Gestaltung von Tabellen:

```
\columncolor{farbe hintergrund}{farbe text}
\rowcolor{farbe}
\rowcolors{2}{lightgray}{white}
```

In die eckige Klammer wird für `farbmodell` die Farbe eingesetzt.

Der Vorteil am Befehl `\rowcolors` liegt darin, dass bei einmaliger Eingabe vor der Tabelle, automatisch jede zweite Zeile grau hinterlegt wird. Allerdings weist dieser Befehl leider auch Nachteile auf: Bei der einmaligen Eingabe des Befehls wird automatisch jede zweite Zeile aller im Skript enthaltenen Tabellen grau hinterlegt, was ja nicht unbedingt immer erwünscht ist.

Ein größerer Nachteil besteht allerdings darin, dass, wenn im Skript Tabellen mit zusammengefassten Zeilen vorhanden sind, die Inhalte dieser durch farbig hinterlegte Zeilen oft einfach überschrieben werden und so später nicht mehr sichtbar sind. Dies ist allerdings ein von \LaTeX noch nicht überarbeiteter Fehler, da dies auch mit den anderen Befehlen zur Farbhinterlegung nicht vermeidbar ist.

Folgendes Beispiel zeigt, wie man mit Hilfe des Befehls `columncolor` die Hintergrundfarbe auf rot und den Text auf weiß (Z.3) einstellt. Mittels des Befehls `\rowcolor` (Z.1) kann der Zeilenhintergrund farbig gestaltet werden. Auch dies veranschaulicht das Beispiel.

Eiskarte		
1 Kugel	In der Waffel	0,70
2 Kugeln	In der Waffel	1,20
3 Kugeln	In der Waffel	1,80

```

\rowcolors{2}{lightgray}{white}
\begin{tabular}{l c r} \toprule
\multicolumn{3}{>{\columncolor{red}}
\color{white}}c}%
{\large Eiskarte} \\\ \hline
1 Kugel & In der Waffel & 0,70
\\\ \hline
2 Kugeln & In der Waffel & 1,20 \\\
3 Kugeln & In der waffel & 1,80 \\\
\bottomrule
\end{tabular}

```

Abbildung 6: Tabelle mit Farbe

1.2.2 Zellen zusammenfassen

Oft sollen Zellen einer Tabelle aus esthetischen oder auch funktionalen Gründen zusammengefasst werden. Dabei wird unterschieden zwischen Zusammenfassung von Zeilen und Spalten.

Spalten zusammenfassen

Um mehrere Spalten einer Tabelle zusammenzufassen, wird der Befehl `\multicolumn` verwendet. Dieser sieht genauer betrachtet so aus:

```
\multicolumn{num}{spaltenform}{text}
```

Mit diesem Befehl werden `num` Spalten in der Ausrichtung der `spaltenform` `l`, `c`, oder `r` zusammengefasst. `text` ist die Beschriftung der neu entstandenen Spalte.

Ein mögliches Eingabebeispiel wäre folgendes:

```
\multicolumn{1-3}{c}{Eiskarte}
```

In diesem Beispiel würden die Spalten 1-3 zusammengefasst werden. Die Beschriftung „Eiskarte“ der neuen Spalte wäre zentriert ausgerichtet.

Zeilen zusammenfassen

Ein weiteres Ergänzungspaket ist nötig damit mehrere Zeilen zusammengefasst werden können:

```
\usepackage{multirow}
```

Das Ergänzungspaket stellt dabei den Befehl `\multirow` zur Verfügung. Die vollständige Eingabe sieht aber folgendermaßen aus:

```
\multirow{nrows}{width}{eingabe}
```

`nrows` gibt die Anzahl der Zeilen an, die zusammengefasst werden sollen.

`width` gibt die Breite an, die der Text einnehmen soll.

`eingabe` ist der verwendete Text, der als Inhalt der zusammengefassten Zeilen erscheint. Dieser kann durch `\\` in beliebig viele Zeilen geteilt werden.

Das folgende Beispiel verdeutlicht dies:

	Zeile 1
	Zeile 2
	Zeile 3
	Zeile 4

```
\begin{tabular}{|c|c|} \hline
\multirow{4}{2cm}{Inhalt}
& Zeile 1\\
& Zeile 2\\
& Zeile 3\\
& Zeile 4\\
\hline \end{tabular}
```

Abbildung 7: Beispiel für Zeilenzusammenfassung

1.2.3 Tabellen über mehrere Seiten

Seltener werden Tabellen benötigt, die über mehrere Seiten gehen. Auch dies ist mit \LaTeX möglich. Der Vorteil Tabellen über mehrere Seiten gerade mit \LaTeX zu erstellen, liegt darin, dass das Programm selbstständig den Tabellenkopf und -fuß auf den anderen Seiten hinzufügt ohne dass diese erneut eingegeben werden müssen.

Das folgende Ergänzungspaket muss ergänzt werden, damit Tabellen über mehrere Seiten überhaupt möglich sind:

```
\usepackage{longtable}
```

Durch das Ergänzungspaket kann folgender Befehl verwendet werden:

```
\begin{longtable}{spaltenform}
...Tabelleninhalt...
\end{longtable}
```

Die Spaltenformen (Ausrichtung der Spalteninhalte) und auch der Zeilenumbruch entsprechen den Befehlen der normalen Tabelle.

Durch Eingabe verschiedener Parameter wird festgelegt, welche Zeilen oder Spalten auf jeder Tabellenseite wieder erscheinen sollen.

2 Abbildungen

Mit \LaTeX lassen sich auch Bilder einbinden. Das Format des Bildes ist dabei egal, da ein Bild für \LaTeX nur eine Box mit einer Höhe und einer Breite darstellt. Das Ausgabeformat erzeugt ein entsprechender Treiber (z.B. dvips, pdftex, usw.). Somit besteht die einzige Einschränkung darin, dass der Treiber das Bildformat unterstützen muss. Unterstützt der Treiber ein Format nicht, so kann dieses Format mit einem Hilfsprogramm in ein passendes Format umgewandelt werden.

Um Bilder einbinden zu können, wird das Ergänzungspaket „graphicx“ benötigt.

```
\usepackage[<parameter>]{graphicx}
```

Über Parameter kann das Aussehen des Bildes beeinflusst werden.

Folgende Parameter stehen zur Auswahl:

- `treiber`
Dieser Parameter legt den Treiber für das Ausgabeformat fest (Bsp.: dvips, pdftex, ...).

Es empfiehlt sich jedoch, keinen Treiber anzugeben. In diesem Falle versucht das Programm selbst heraus zufinden, welches System verwendet wird und stellt sich entsprechend ein. Somit bleibt die Austauschbarkeit des Dokuments erhalten.

- `draft`
Durch diesen Parameter wird die Verarbeitungsgeschwindigkeit in der Entwurfsphase erhöht, da nur ein der Bildgröße entsprechender Rahmen gezeichnet wird, in dem der Dateiname der Grafik angegeben ist. Das Bild selbst wird nicht mit eingebunden.
- `final`
Dieser Parameter ist das Gegenstück zu `draft`. Hier wird die Grafik mit eingebunden und angezeigt.

2.1 Bilder aus einer Datei einbinden

Um Bilder aus Dateien einzubinden, wird der Befehl „`\includegraphics`“ verwendet. Wichtig ist, dass die Datei und das Skript im gleichen Ordner gespeichert sind.

```
\includegraphics[<parameter>]{<datei>}
```

Die Dateiendung sollte hier nicht mit angegeben werden, da das Programm entsprechend der Treiberdefinition automatisch die Dateiendung anhängt und sich so die entsprechende Dateiendung selbst sucht. Folgende Werte stehen als Parameter zur Verfügung:

- `width=dim`

Mit diesem Parameter wird die Breite des Bildes festgelegt. Man muss allerdings beachten, dass die Originalbreite entsprechend gestaucht bzw. gestreckt wird. Die Höhe wird automatisch der `BoundingBox` angepasst, wenn diese nicht explizit angegeben wird.

- `height=dim`

Dieser Parameter legt die Höhe fest. Hierbei gilt wie bei dem vorherigen Parameter, dass die Originalhöhe entsprechend gestaucht bzw. gestreckt wird. Auch hier wird die Breite entsprechend der `BoundingBox` angepasst, wenn sie nicht ausdrücklich angegeben wird.

- `totalheight=dim`

Dieser Parameter wird benötigt, wenn das Bild gedreht wurde. Er legt ebenfalls die Gesamthöhe fest.

- `draft`

Dieser Parameter wirkt wie im vorherigen Abschnitt angegeben.

- `final`

Auch dieser Parameter wirkt wie im vorherigen Abschnitt angegeben.

- `keepaspectratio`

Ohne diesen Parameter kann das Bild verzerrt dargestellt werden. Er sorgt bei der Angabe von Höhe und Breite dafür, dass das Seitenverhältnis beibehalten wird und die Werte entsprechend angepasst werden.

```
\includegraphics [width=8cm,height=6cm,keepaspectratio]{Bild}
```

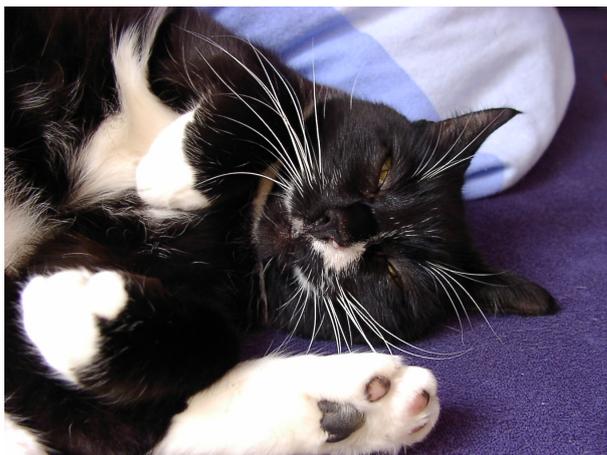


Abbildung 8: Ergebnis: Bilder aus einer Datei einbinden

2.2 Gleitende Abbildungen

Oft tritt das Problem auf, dass eine Abbildung zu groß ist und nicht mehr auf die entsprechende Seite passt, sondern auf die nächste Seite verschoben wird. Dieses stört häufig den Gesamteindruck. Mit \LaTeX kann so etwas vermieden werden, indem die Abbildung (oder Tabelle) in eine gleitende Umgebung eingebunden wird. So kann mit verschiedenen Positionsangaben bestimmt werden, an welche Stelle die Abbildung alternativ gesetzt wird. Der allgemeine Aufruf lautet:

```
\begin{figure}[<position>]
\includegraphics[<parameter>]{Bild}
\end{figure}
```

Folgende Positionen können verwendet werden.

- h

Das h steht für hier.

- t

Passt der vorausgehende Text voll auf die Seite, wird das Objekt am Anfang der Seite vor dem Text platziert. Sollte der Text länger als die Seite sein, wird die Abbildung zu Beginn der nächsten Seite platziert. Dabei wird der laufende Text bis zum Seitenwechsel vorgezogen, also vor das Objekt gesetzt.

- b

Dieser Parameter funktioniert genauso wie der Parameter t, nur dass das Objekt am Ende der Seite platziert wird.

- p
Für das Objekt wird eine eigene Seite eingerichtet. Auf ihr können auch mehrere Objekte platziert werden.
- !
Mit diesem Parameter werden die nachfolgenden Grundeinstellungen aufgehoben.

Alle Parameter können auch miteinander kombiniert werden.

2.3 Bilder im Text platzieren

Bilder zentrieren

Um Bilder zu zentrieren, wird folgender Befehl benötigt:

```
\begin{figure}[<position>]  
\centering  
\includegraphics{Bild}  
\end{figure}
```



Abbildung 9: Beispiel: Bilder zentrieren

Bilder rechts und links verschieben

Mit dem Befehl „flushleft“ bzw. „flushright“ lassen sich Bilder rechts- bzw. linksseitig verschieben. Der allgemeine Aufruf lautet:

```
\begin{flushleft}
\includegraphics[<parameter>]{<datei>}
\end{flushleft}
```



Abbildung 10: Ergebnis: Bilder nach rechts verschieben

Bilder nebeneinander setzen

Um Bilder nebeneinander zu setzen, können minipage-Umgebungen benutzt werden. Der Befehl könnte beispielsweise so aussehen:

```
\begin{figure}
\begin{minipage}[b]{.5\linewidth}
\includegraphics[keepaspectratio,width=\linewidth]{Bild links}
\caption{Bildunterschrift links}
\end{minipage}
\hspace{.1\linewidth}
\begin{minipage}[b]{.5\linewidth}
\includegraphics[width=\linewidth]{Bild rechts}
\caption{Bildunterschrift rechts}
\end{minipage}
\end{figure}
```

Durch die Zeilen 2 und 7 wird die Breite jeder Hälfte auf 50 Prozent der Zeilenbreite gesetzt. Dabei ist zu beachten, dass die Bildgröße auf jeden Fall kleiner gewählt wird als die Breite der minipage-Umgebung. Sonst kommt die Meldung „overflow hbox“. Durch den Parameter werden beide Bilder unten ausgerichtet. Mit diesem Befehl können auch Tabellen und andere Objekte neben dem Bild platziert werden.



Abbildung 11: Beispiel: Bilder nebeneinander setzen

Drehen von Bildern

Über den Parameter „angle“ kann bestimmt werden, um welchen Winkel das Bild gedreht werden soll.

```
\begin{figure}[h]  
\includegraphics [angle=30, scale=.15]{Bild}  
\end{figure}
```



Abbildung 12: Ergebnis: Drehen von Bildern

Skalieren von Bildern

Bilder können mit dem Befehl „`resizebox`“ auf eine bestimmte Breite bzw. Höhe skaliert werden.

```
\resizebox{<breite>}{<höhe>}{<Bild>}
```

Gibt man für die Breite oder die Höhe „!“ ein, berechnet \LaTeX mit diesen Wert entsprechend dem Größenverhältnis.

```
\begin{figure}[h]
\resizebox{!}{6cm}{\includegraphics{eisbaer}}
\end{figure}
```



Abbildung 13: Ergebnis: Skalieren von Bildern

Bilder kann man auch über den Parameter „`scale`“ skalieren. Der allgemeine Aufruf lautet:

```
\includegraphics [scale=<zahl>]{Bild}
```

Text neben dem Bild

Über den Befehl „`\parpic`“ kann ein Text links bzw. rechts neben das Bild geschrieben werden. Soll der Text rechts neben dem Bild stehen, wird für die Position ein „`r`“ eingegeben; soll der Text links neben dem Bild stehen, wird für die Position ein „`l`“ angegeben. Dazu wird allerdings das Zusatzpaket „`picins`“ benötigt.

```

\begin{figure}
\parpic[<position>]{\includegraphics[<parameter>]{<Bild>}}
<Text>
\end{figure}

```

Problematisch ist es, wenn der Text neben dem Bild weniger Zeilen als die Höhe des Bildes hat. Dann fügt \LaTeX den folgenden Text direkt an den Text neben dem Bild an. Dabei schreibt er den anschließenden Text in das Bild. In diesem Falle ist es besser, das Bild zu zentrieren und den Text über oder unter das Bild zusetzen. Weiterhin besteht die Möglichkeit mit dem Befehl „ $\text{\vspace{cm}}$ “ den Abstand unter dem Text, der neben das Bild gesetzt werden soll, anzugeben.

```

\begin{figure}[h]
\parpic(scale=.15)[r]{\includegraphics[scale=.15]{<Bild>}} Rechts auf dem
Bild ist eine schwarze Katze zu sehen. Diese schwarze Katze heißt Lilli.
Sie wurde am 11.05.2000 geboren. Ihre Mutter ist meine Katze Nelli, die
unter dem Punkt „Bilder aus Dateien einbinden\“ zu sehen ist.
\end{figure}

```

Rechts auf dem Bild ist eine schwarze Katze zu sehen. Diese schwarze Katze heißt Lilli. Sie wurde am 11.05.2000 geboren. Ihre Mutter ist meine Katze Nelli, die unter dem Punkt „Bilder aus Dateien einbinden“ zu sehen ist.



Abbildung 14: Ergebnis: Text neben dem Bild

Bilder mit Rahmen

Um ein Bild mit einem Rahmen zu versehen, wird das Ergänzungspaket „caption2“ benötigt. Folgende Möglichkeiten stehen zur Verfügung:

- `frameenv`

Dieser Befehl erzeugt einen einfachen Rahmen.

- `dashenv`

Der Rahmen um das Bild wird gestrichelt.

- `ovalenv`

Die Ecken des Rahmens werden abgerundet.

- `shadowenv`

Der Rahmen wird mit einem Schatten versehen.

```
\begin{figure}[h]
\centering
\begin{shadowenv}[10cm]
\includegraphics[keepaspectratio,scale=.2]{<Bild>}
\end{shadowenv}
\end{figure}
```



Abbildung 15: Ergebnis: Der Rahmen wird mit einem Schatten versehen

3 Beschriften von Tabellen und Abbildungen

Damit die Tabellen und Abbildungen eines Dokuments beschriftet werden können, muss ein weiteres Ergänzungspaket eingefügt werden.

```
\usepackage{caption2}
```

Um diese Tabelle oder Abbildung nun zu beschriften, muss hinter den Befehl oder der Abbildung der folgende Befehl eingegeben werden:

```
\caption
```

Das folgende Skript soll dies veranschaulichen:

```
\begin{tabular}{l c r}  
...Tabelleninhalt...  
\end{tabular}  
\caption{Name des Objekts}
```

4 Grundlagen für das Schreiben von Formeln

4.1 Grundlegende Mathematik-Schreibweisen

\LaTeX erkennt Formeln nicht automatisch. Damit \LaTeX Formeln erkennt, müssen sie in eine Mathematik-Umgebung eingebunden werden. Hierzu stehen zwei Möglichkeiten zur Auswahl. Die eine besteht darin, dass die Formel innerhalb einer Textzeile eingebunden wird. Die andere Möglichkeit ist, die Formel in einer Extrazeile vom Text abzusetzen.

Um die Formel in den Text einzubinden, wird folgender Befehl verwendet:

```
\begin{math}
Formel
\end{math}
```

Anstelle des oben genannten Befehls kann man die Formel auch in Dollarzeichen setzen.

```
$Formel$
```

Um die Formel in eine eigene Zeile zu schreiben, stehen zwei Möglichkeiten zur Auswahl. Bei dem Befehl „equation“ werden die Formeln automatisch horizontal zentriert und fortlaufend nummeriert. Die Formelnummer wird rechtsbündig angezeigt.

```
\begin{displaymath}
Formel
\end{displaymath}
```

```
\begin{equation}
Formel
\end{equation}
```

Der Befehl „displaymath“ kann auch durch eckige Klammern ersetzt werden.

```
\[Formel\]
```

Hier einige Beispiele zu den beschriebenen Befehlen.

Dies ist ein Beispiel für eine Formel, die in einen laufenden Text eingebunden wird. Die Beispielformel lautet $y=2x-5$. Das ist doch gar nicht so schwer.

Dies ist ein Beispiel für eine Formel, die in einen laufenden Text eingebunden wird. Die Beispielformel lautet $y = 2x - 5$. Das ist doch gar nicht so schwer.

Hier sehen Sie ein Beispiel, in dem die Formel in eine eigene Zeile geschrieben wird.

```
\begin{equation}
f(x)=6x+3
\end{equation}
```

Da der Befehl „equation“ verwendet wurde, wird die Formel auch nummeriert.

Hier sehen Sie ein Beispiel, in dem die Formel in eine eigene Zeile geschrieben wird.

$$f(x) = 6x + 3 \tag{1}$$

Da der Befehl „equation“ verwendet wurde, wird die Formel auch nummeriert.

i Bei manchen Formelzeichen muss man aufpassen, wenn man sie in den Text einbindet. Einige Formelzeichen wie z.B. das Summenzeichen oder das Integralzeichen sind höher als der Resttext. \LaTeX staucht diese entsprechend zusammen, was nicht unbedingt erwünscht ist. Formeln mit solchen Zeichen sollten nach Möglichkeit abgesetzt werden.

Beim Umgang mit Formeln sind folgende Punkte zu beachten:

- Mathematische Formeln müssen innerhalb eines Absatzes stehen; Leerzeichen können nicht benutzt werden
- Alle Abstände werden automatisch nach der Form der Formel bestimmt oder müssen mit den Befehlen `\`, bzw. `\quad` angegeben werden.

4.2 Die Elemente von Formeln

Formeln bestehen aus Konstanten (also Zahlen) und Variablen, die durch Operatoren verbunden werden. \LaTeX verwendet dabei für Konstanten und für Variablen verschiedene Schriftarten in einer mathematischen Formel. Für Konstanten verwendet es die Schriftart Roman, Variablen werden kursiv gesetzt.

Folgende Symbole können direkt über die Tastatur eingegeben werden³:

+ - = / : ! ' | [] ()

³aus: \LaTeX Das Praxisbuch von Elke und Michael Niedermair, 2005 erschienen

Klammersymbole

Klammern werden oft verwendet, um Formeln zu strukturieren. Tauchen viele Klammern in einer Formel auf, kann man die zugehörigen Klammern deutlich machen, indem man verschiedene Klammerformen verwendet. Dies erhöht die Lesbarkeit und man behält leichter den Überblick. Folgende Klammersymbole sind bei \LaTeX vorhanden⁴:

Symbol	Eingabe	Symbol	Eingabe
(())
⟨	<code>\langle</code>	⟩	<code>\rangle</code>
[[]]
	<code> \\$</code>		<code>\\$ </code>
⌈	<code>\lceil</code>	⌋	<code>\rceil</code>
⌊	<code>\lfloor</code>	⌋	<code>\rfloor</code>
{	<code>\{</code>	}	<code>\}</code>
\	<code>\backslash</code>		

i Damit geschweifte Klammern angezeigt werden, muss ihnen ein Backslash vorangestellt werden.

Um die richtige Größe für die Klammern festzulegen, kann der Befehl „left“ vor die öffnende und der Befehl „right“ vor die schließende Klammer gesetzt werden. Dadurch passt \LaTeX die Klammergröße automatisch an. Zu beachten ist, dass die Befehle immer zusammen auftreten. Die öffnende und schließende Klammer muss aber nicht identisch sein. So kann man auch eine geschweifte Klammer als öffnende und eine runde Klammer als schließende Klammer verwenden.

```
$ \left\{112-56\right\}$
```

$$\{112 - 56\}$$

Bei einigen Formeln oder beispielsweise bei stückweise definierten Funktionen braucht man nur eine Klammer. Indem man einen Punkt vor das „left“ bzw. „right“ setzt, wird diese Klammer unsichtbar.

```
\[ H(x) = \left\{
\begin{array}{r@{\quad:\quad}l}
0 & x \leq 0 \\
1 & x > 0
\end{array}
\right. \]
```

$$H(x) = \begin{cases} 0 & : x \leq 0 \\ 1 & : x > 0 \end{cases}$$

⁴aus: \LaTeX Das Praxisbuch von Elke und Michael Niedermair, 2005 erschienen

☐ Durch den Befehl `\begin{array}` wurde in dem obigen Beispiel eine array-Umgebung verwendet. Hierzu wird das Usecpackage „array“ benötigt. Sie funktioniert ähnlich wie eine Tabelle. In diesem Beispiel wird durch die array-Umgebung erreicht, dass die Spalten mit einem Doppelpunkt getrennt werden. Mit `\quad` wird der Abstand, mit dem die Spalten getrennt werden, angegeben. In diesem Fall haben wir zwei Spalten, die durch `&` gekennzeichnet werden.

Die Klammergrößenanpassung mit diesen Befehlen passt nicht immer gut, da bestimmte Größen vorgegeben sind und deshalb nicht optimal angepasst werden können. Ist die Formel nur ein wenig größer als die Klammer, wird trotzdem die nächstgrößere Klammer verwendet, die natürlich zu groß ist. Die Klammergröße lässt sich durch den Befehl „bigl“ bzw. „bigr“ direkt beeinflussen. Durch unterschiedliche Groß- und Kleinschreibung der Buchstaben können verschieden große Klammern erzeugt werden. Folgende Befehle werden dazu benötigt⁵:

Befehl	Anzeige
<code>(,)</code>	$(o)\left(\frac{n}{m}\right)$
<code>\left(, \right)</code>	$(o)\left(\frac{n}{m}\right)$
<code>\bigl(, \bigr)</code>	$(o)\left(\frac{n}{m}\right)$
<code>\Bigl(, \Bigr)</code>	$(o)\left(\frac{n}{m}\right)$
<code>\biggl(, \biggr)</code>	$(o)\left(\frac{n}{m}\right)$
<code>\Biggl(, \Biggr)</code>	$(o)\left(\frac{n}{m}\right)$

Mit diesen Befehlen kann man die Klammergrößen innerhalb einer Formel variieren, was die Lesbarkeit fördert.

```
\begin{math}
\Bigl(2*(82-59)-21\Bigr)\Bigl(3*(67+44)-(10+53)\Bigr)
\end{math}
```

$$(2 * (82 - 59) - 21) (3 * (67 + 44) - (10 + 53))$$

Abbildung 16: Ergebnis: Klammersymbole

Durch die Bestimmung der Klammergrößen mit „bigl“ bzw. „bigr“ kann vermieden werden, dass Klammerkonstruktionen im laufenden Text den Zeilenabstand vergrößern.

⁵aus: L^AT_EXDas Praxisbuch von Elke und Michael Niedermair, 2005 erschienen

Senkrechte Striche

In der Mathematik werden oft senkrechte Striche zu verschiedenen Zwecken benötigt. Für einfache senkrechte Striche wird der Befehl „`lvert`“ bzw. „`rvert`“ verwendet. Um doppelte senkrechte Striche zu erzeugen, wird der Befehl „`lVert`“ bzw. „`rVert`“ benötigt.

Hoch- und Tiefstellen von Zeichen

In der Mathematik werden häufig Exponenten und Indizes benötigt. Für einen Exponenten wird das Zeichen eingeegeben und für einen Index das Zeichen `_`. Die Schriftgröße wird dabei automatisch verringert. Beide Zeichen können auch in beliebiger Reihenfolge kombiniert werden.

Sollen Exponent und Index nicht direkt übereinander, sondern versetzt stehen, kann man geschweifte Klammern nach dem Index einfügen. Dadurch wird der Exponent hinter den Index verschoben.

Bestehen Index oder Exponent aus mehr als einer Zahl, müssen deren Bestandteile in geschweifte Klammern gesetzt werden.

```
\begin{math}
x_n{}_{i-1}
\end{math}
```

$$x_n^{i-1}$$

Wurzeln

Der Befehl für eine Wurzel lautet „`\sqrt`“.

```
\sqrt[<n>]{<arg>}
```

Bei diesem Befehl steht „`n`“ für den Grad der Wurzel und „`arg`“ gibt den Teil unter der Wurzel an. Höhe und Länge der Wurzel passt \LaTeX automatisch an.

```
\begin{math}
\sqrt[5]{40-4}
\end{math}
```

$$\sqrt[5]{40-4}$$

Brüche

Brüche werden mit dem Befehl „`\frac`“ erstellt.

```
\frac{<Zähler>}{<Nenner>}
```

L^AT_EX passt die Länge des Bruchstriches automatisch dem längsten Argument an. Werden Brüche im Text verwendet, werden diese automatisch vertikal zum Text zentriert. Bei verschachtelten Brüchen wird die Lesbarkeit dadurch beeinträchtigt, dass die Schriftgröße automatisch immer kleiner wird. Dieses kann durch den Befehl „`\displaystyle \frac`“ vermieden werden.

```
\begin{math}
\frac{x^4 + x^2 + x}{\sqrt{x^2}}
\end{math}
```

$$\frac{x^4 + x^2 + x}{\sqrt{x^2}}$$

```
\begin{math}
\displaystyle
\frac{x^4 + x^2 + x}{\sqrt{x^2}}
\end{math}
```

$$\frac{x^4 + x^2 + x}{\sqrt{x^2}}$$

Wurzeln und Brüche lassen sich auch beliebig kombinieren.

$$\sqrt[3]{\frac{x^2 - x - 1}{\sqrt{x - 1}}}$$

Abbildung 17: Beispiel: Kombinieren von Wurzeln und Brüchen

Griechische Buchstaben

Häufig braucht man für mathematische Formeln neben den Konstanten und Variablen auch griechische Buchstaben. Sie werden erzeugt, indem man ihrem Namen einen Textbackslash voransetzt. Schreibt man den Namen groß, wird der Großbuchstabe angezeigt; schreibt man den Namen klein wird der Kleinbuchstabe angezeigt. Werden griechische Buchstaben im laufenden Text benötigt, müssen sie zwischen Dollarzeichen gesetzt werden. Die folgende Tabelle zeigt die Befehle für die einzelnen Buchstaben⁶.

⁶aus: L^AT_EX Das Praxisbuch von Elke und Michael Niedermair, 2005 erschienen

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

Abbildung 18: Griechische Buchstaben

In der Regel werden die griechischen Buchstaben zur Beschriftung von Winkeln verwendet. Um das Gradzeichen zu erzeugen, wird der Befehl „`\textdegree`“ hinter die Gradzahl geschrieben.

```
$ \beta $ =90\textdegree
```

 $\beta =90^\circ$

Mit den oben angegebenen Befehlen ist es nicht möglich, die griechischen Großbuchstaben kursiv zuschreiben wie die anderen Variablen. Dazu muss vor jedem Großbuchstaben ein „\var“ geschrieben werden.

```
\varOmega
```

Funktionsnamen

Funktionsnamen sollten möglichst durch L^AT_EX-Befehle eingegeben werden, damit L^AT_EX sie auch als Funktionsnamen erkennt. Sonst kann es passieren, dass sie für Variablen gehalten werden und ebenfalls kursiv dargestellt werden. Diese Funktionsnamen sind L^AT_EX bekannt⁷:

Funktion	Eingabe
arccos	<code>\arccos</code>
cos	<code>\cos</code>
cot	<code>\cot</code>
lim	<code>\lim</code>
log	<code>\log</code>
arcsin	<code>\arcsin</code>
ln	<code>\ln</code>
sin	<code>\sin</code>

Abbildung 19: Funktionsnamen

Einige Funktionen benötigen Grenzangaben wie z.B. „\lim“. Die Grenzwertangaben können wie Indizes angegeben werden.

```
\lim_{x\to\infty}\frac{2}{x}=0
```

$$\lim_{x \rightarrow \infty} \frac{2}{x} = 0$$

i Der Befehl „\to\infty“ gibt an, dass x gegen unendlich laufen soll.

L^AT_EX unterscheidet bei der Darstellungsweise, ob die Funktion im Text oder als abgesetzte Formel dargestellt werden soll. Soll die Funktion im Text eingefügt werden, wird die Grenzangabe in die gleiche Zeile geschrieben (siehe vorheriges Beispiel).

⁷ aus: L^AT_EX Das Praxisbuch von Elke und Michael Niedermair, 2005 erschienen

```
\[\lim_{x\to\infty}\frac{2}{x}=0\]
```

$$\lim_{x \rightarrow \infty} \frac{2}{x} = 0$$

Abbildung 20: Beispiel: Grenzwertfunktion in einer Extrazeile

4.3 Gestaltung von Formeln

Gerahmte Formeln

Möchte man Formeln besonders hervorheben, kann man diese einrahmen. Dazu müssen die Umgebungen `displaymath` oder, wenn die Formeln nummeriert werden sollen, `equation` verwendet werden. Anschließend wird in dieser Umgebung eine „`\fbox`“, welche die Formel enthält, definiert.

```
\begin{displaymath}
\fbox {$ \displaystyle
<Formeltext>$}
\end{displaymath}
```

```
\begin{equation}
\fbox {$ \displaystyle
<Formeltext>$}
\end{equation}
```

```
\{equation}
\fbox {$\displaystyle W=\frac {1}{2}*m*v^2$}
\end{equation}
```

$$\boxed{W = \frac{1}{2}m * v^2} \quad (2)$$

Abbildung 21: Ergebnis: Gestaltung von Formeln

5 Übungsaufgaben

Die folgenden Übungsaufgaben vertiefen die im Kleinen L^AT_EX Tutorial II erlernten Möglichkeiten zum Umgang mit Tabellen, Abbildungen und leichteren Formeln. Daher sollten diese nun ohne Probleme zu lösen sein.

Die Lösungen der Übungsaufgaben befinden sich im Anhang.

Umgang mit Formeln

Formeln einbinden

Geben Sie folgende Formel an:

$$y = 2x - 5$$

Setzen die die nachfolgende Formel in eine eigene Zeile zwischen den beiden Textteilen.

Jetzt probiere ich eine Formel in eine eigene Zeile zuschreiben und diese nummerieren zu lassen.

$$y = 8x + 2 \tag{3}$$

So schwer ist das wirklich nicht.

Verschiedene Klammergrößen

Benutzen Sie verschieden große Klammern. Verwenden Sie den Befehl „\Bigl“ bzw. „\Bigr“.

$$\left(7 * (17 - 7)\right) \left((66 + 44) * (37 + 53)\right)$$

Hoch- und Tiefstellen von Zeichen

Geben Sie folgenden Term an:

$$K_c^{2*5}$$

Wurzeln

Schreiben Sie folgende Wurzel:

$$\sqrt[3]{6 * 6}$$

Brüche

Stellen Sie folgenden Bruch nach:

$$\frac{1}{2}$$

Wurzeln und Brüche kombinieren

Kombinieren Sie Brüche und Wurzeln.

$$\frac{1}{\sqrt{x^2 - 2}}$$

Winkelangabe

Geben Sie folgenden Winkel an:

$$\gamma = 25^\circ$$

Grenzwertfunktion

Lassen Sie bei folgender Grenzwertfunktion x gegen unendlich laufen:

$$\lim_{n \rightarrow \infty} \frac{-1}{n} = 0$$

Gerahmte Formel

Rahmen Sie die nachfolgende Formel ein und lassen Sie diese nummerieren:

$$\boxed{W = m * g * h} \quad (4)$$

6 Nachwort

Die Facharbeit wurde von Asja Lippert und Wiebke Matteikat gemeinsam geschrieben. Asja Lippert hat die Seiten (4-15 und 25) und Wiebke Matteikat die Seiten (16-24 und 29-34) verfasst.

Vor- und Nachwort wurden gemeinsam erarbeitet. Die Übungsaufgaben zum Thema „Tabellen“ wurden von Asja Lippert erstellt. Wiebke Matteikat hat die Übungsaufgaben zum Thema „Einbinden von Formeln“ zusammengestellt.

Im Rahmen unserer Facharbeit verfassten wir dieses Tutorial eigenständig mit \LaTeX . Anfangs war es schwer sich in das Programm einzuarbeiten, da die Hauptquelle⁸ dieses Tutorials teilweise fehlerhaft beziehungsweise unvollständig gewesen ist. So konnten auch nach intensivster Auseinandersetzung einige Probleme nicht gelöst werden. Auch im weiteren Verlauf traten immer wieder Schwierigkeiten auf, deren Ursachen meist nicht leicht zu finden waren.

Eigene Vorstellungen mit \LaTeX umzusetzen bereitete erneut Schwierigkeiten, da hierzu oft komplexe Befehle nötig waren.

Insgesamt stellen wir fest, dass \LaTeX ein Programm ist mit dem man höherwertige Dokumente schreiben kann, das aber für Ungeübte und / oder Windowsnutzer wenig anwenderfreundlich ist. Wenn man allerdings die Befehlsstruktur beherrscht, so lassen sich auch umfangreiche Dokumente relativ leicht erstellen.

⁸ \LaTeX Das Praxisbuch von Elke und Michael Niedermair; erschienen 2005

7 Literaturverzeichnis

1. Dudenredaktion (Hrsg.): DUDEN, Rechtschreibung der deutschen Sprache, 21. Aufl.; Mannheim, 1996
2. Betting, Jan; & Peschke, Niklas: Kleines \LaTeX Tutorial I; Hannover, 2006
3. Niedermair, Elke und Michael: \LaTeX Das Praxisbuch 2. vollständig aktualisierte und überarbeitete Neuauflage; Poing: Franzis, 2005

8 Anhang

8.1 Lösungen der Übungsaufgaben

Damit die vorher durchgeführten Übungsaufgaben zum Umgang mit Tabellen und Formeln auch verglichen werden können, sind hier die Lösungsskripte.

Formeln einbinden

```
$y=2x-5$
```

Verschiedene Klammergrößen

Jetzt probiere ich eine Formel in eine eigene Zeile zuschreiben und diese nummerieren zu lassen.

```
\begin{equation}
```

```
y=8x+2
```

```
\end{equation}
```

So schwer ist das wirklich nicht.

Verschiedene Klammergrößen

```
\begin{math}
```

```
\Bigl(7*(17-7)\Bigr)\Bigl((66+44)*(37+53)\Bigr)
```

```
\end{math}
```

Hoch- und Tiefstellen von Zeichen

```
\begin{math}
```

```
$ K_c 2 * 5 $
```

```
\end{math}
```

Wurzeln

```
\begin{math}
```

```
\sqrt[3]{6*6}
```

```
\end{math}
```

Brüche

```
\begin{math}  
\frac{1}{2}  
\end{math}
```

Wurzeln und Brüche kombinieren

```
\begin{math}  
\displaystyle \frac{1}{\sqrt{\$ x^2 - 2 \$}}  
\end{math}
```

Winkelangabe

```
\gamma = 25 \text{degree}
```

Grenzwertfunktion

```
\lim_{n \to \infty} \frac{-1}{n} = 0
```

Gerahmte Formel

```
\begin{equation}  
\boxed{\$ W = m * g * h \$}  
\end{equation}
```

Versicherung der selbständigen Erarbeitung

Hiermit versichern wir, dass wir die Arbeit selbstständig angefertigt, keine anderen als die angegebenen Hilfsmittel benutzt und die Stellen der Facharbeit, die im Wortlaut oder im wesentlichen Inhalt aus anderen Werken entnommen wurden, mit genauer Quellenangabe kenntlich gemacht haben.

Verwendete Informationen aus dem Internet sind unserem Fachlehrer vollständig im Ausdruck zur Verfügung gestellt worden.

Hannover, den 5. März 2007

Asja Lippert

Wiebke Matteikat

Wir stimmen der Veröffentlichung unserer Arbeit durch Ausstellung in der Schülerbibliothek der Marie Curie Schule Ronnenberg zu.

Die Erklärung wird nur wirksam, wenn die Facharbeit mit mindestens 08 Punkten bewertet wird.

Asja Lippert

Wiebke Matteikat