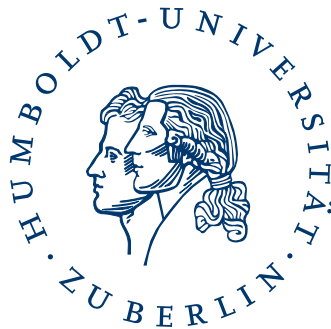


Ausarbeitung zum Vortrag  
MATHEMATISCHER FORMELSATZ IN  $\text{\LaTeX}$



von

RAFFAEL DZIKOWSKI

Erstellt im Rahmen des Proseminares  
*Wissenschaftliches Publizieren mit  $\text{\LaTeX}$*   
im SoSe 2006

## Inhaltsverzeichnis

1	Das $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ -Projekt	1
2	Grundlagen des mathematischen Formelsatzes in $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$	1
3	Mathematische Formeln setzen	4
4	Anwendungen aus Beispielgebieten der Mathematik	9
5	Zeilen- und Seitenumbrüche innerhalb von <i>display</i> -Umgebungen	16
6	Referenzieren von Formeln	18
7	Eine fertige Umgebung für den Satz von Beweisen	19

## Vorbemerkungen

Wir haben im Laufe des Semesters viele Vorteile des  $\text{\LaTeX}$ -Systems gegenüber den marktüblichen WYSIWYG-Textprozessoren wie z. B. *Microsoft Word*<sup>TM</sup> kennengelernt.

Eine wesentliche Stärke von  $\text{\LaTeX}$  ist jedoch bisher noch nicht angesprochen worden, nämlich der außerordentlich gut ausgereifte mathematische Formelsatz. Es ist keineswegs verwunderlich, dass  $\text{\LaTeX}$  auch die komplexesten mathematischen Konstrukte in einer für den Betrachter sehr ansprechenden Form setzen kann, denn dies war schließlich die primäre Motivation für DONALD E. KNUTH, als er das  $\text{\TeX}$ -System entwarf. In  $\text{\LaTeX}$  wurden über die ohnehin schon großen Möglichkeiten von  $\text{\TeX}$  noch weitere Verbesserungen am Formelsatz durchgeführt. Ein Zusatzpaket hat sich besonders durchgesetzt und wird heutzutage für alle mit  $\text{\LaTeX}$  gesetzten mathematischen Publikationen verwendet, deshalb soll es hier nicht vorenthalten werden.

## 1 Das $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{\LaTeX}$ -Projekt

Die AMERICAN MATHEMATICAL SOCIETY (AMS) hat für ihre Zwecke eine Reihe von Zusatzpaketen für  $\text{\LaTeX}$  entwickelt, mit deren Einbindung die Standard- $\text{\LaTeX}$ -Befehle erweitert und teilweise sogar verbessert werden. Deswegen ist es ratsam, bei jedem Dokument, das mathematische Elemente enthält, in der Präambel den Befehl

```
\usepackage{amsmath}
```

stehen zu haben, der das Hauptpaket einbindet. Darüber hinaus gibt es noch viele andere  $\mathcal{A}\mathcal{M}\mathcal{S}$ -Zusatzpakete die bei Bedarf erwähnt werden. Leser die am Satz *rein* mathematischer Dokumente mit  $\text{\LaTeX}$  interessiert sind, seien darauf verwiesen, dass die AMS über die Zusatzpakete hinaus noch eigene Dokumentenklassen für mathematische Abhandlungen anbietet. Die Beschreibung dieser Klassen ist aber nicht Gegenstand dieser Ausarbeitung, da sie den Rahmen sprengen würde (Für eine ausführliche Dokumentation siehe [www.ams.org](http://www.ams.org)). Außerdem gehen durch die Benutzung dieser Klassen einige wichtige Vorteile des KOMA SCRIPT-Paketes verloren, was durch die bloße Einbindung des  $\text{amsmath}$ -Paketes nicht passiert.

## 2 Grundlagen des mathematischen Formelsatzes in $\text{\LaTeX}$

Alles, was in die Rubrik Mathematik fällt, muss in  $\text{\LaTeX}$  im sogenannten *math mode* (im Folgenden Mathe-Modus genannt) gesetzt werden. Um  $\text{\LaTeX}$  zu veranlassen, in diesen Modus überzugehen, muss eine mathematische Umgebung geöffnet werden. Hierbei ist zu beachten, dass innerhalb dieser Umgebungen andere Regeln gelten als für den normalen Textsatz. Jegliche Zwischenraumrechnungen im Mathe-Modus werden von  $\text{\LaTeX}$  übernommen, was dazu führt,

dass Leerzeichen innerhalb mathematischer Umgebungen keinen Einfluss auf das Druckergebnis haben. Sie können zwar im Quellcode der Übersichtlichkeit halber auftreten, werden aber bei der Kompilation des Dokumentes bis auf drei Ausnahmen vollständig ignoriert. Es gilt:

1. Leerzeichen nach Befehlen ohne Parameter sind wesentlich. Sie signalisieren  $\LaTeX$  das Befehlsende.
2. Leerzeichen, die im Parameter eines Befehles auftauchen, der temporär in den Textbearbeitungszustand zurückwechselt. (Ein Beispiel hierfür wäre der vom Paket `amsmath` bereitgestellte Befehl `\text{<arg>}`, den wir später kennenlernen werden)
3. Ein Leerzeichen, das direkt auf den `\` folgt (maskiertes Leerzeichen) wird nicht ignoriert und erscheint als solches in der Ausgabe

In den meisten Fällen, ist das von  $\LaTeX$  erzeugte Druckergebnis optimal, denn Zwischenräume werden automatisch dort gesetzt, wo sie nach der Logik des beschriebenen mathematischen Konstruktes hingehören. In Ausnahmefällen kann jedoch eine Anpassung der Zwischenräume durch den Benutzer nötig sein. Hierfür sind im Wesentlichen fünf Befehle vorgesehen:

- `\`, erzeugt einen kleinen Abstand (Dieser Befehl ist als einziger von den vorgestellten auch im Textmodus erlaubt)
- `\;` erzeugt einen mittleren Abstand
- `\quad` und `\qquad` erzeugen größere Abstände
- `\!` führt dazu, dass ein vorhandener Abstand *verkleinert wird*

Ein weiterer wichtiger Unterschied zum Textbearbeitungszustand ist, dass *Absätze innerhalb mathematischer Umgebungen verboten* sind. Mathematische Formeln *müssen* innerhalb eines Absatzes stehen. Dies bedeutet insbesondere, dass im Quellcode innerhalb einer mathematischen Umgebung keine Leerzeilen auftreten können (diese werden von  $\LaTeX$  als Absatzende interpretiert).

Auch Zeilenumbrüche sind in den Standard- $\LaTeX$ -Umgebungen nicht - wie wir es aus dem Textbearbeitungszustand gewohnt sind - durch die Eingabe von `\\` möglich. Es werden jedoch im Weiteren Verlauf dieser Ausarbeitung noch Umgebungen erwähnt, die den Zeilenumbruch erlauben.

## 2.1 Mathematische Umgebungen

Grundsätzlich gibt es zwei Arten von mathematischen Umgebungen:

- Umgebungen für Formeln, die *innerhalb* einer Textzeile auftreten (*inline*)
- Umgebungen für *abgesetzte* Formeln (*displayed*)

Für den Inline-Modus wird die `math` Umgebung verwendet. Diese kann in der gewohnten Weise

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

benutzt werden. Da aber damit zu rechnen ist, dass mathematische Formeln in  $\text{\LaTeX}$ -Dokumenten gehäuft auftreten, gibt es für diese Umgebung noch andere, abkürzende Schreibweisen, die die gleiche Wirkung haben. Zum Einen die Schreibweise

```
\(
<Formel>
\)
```

und zum Anderen auch die noch kürzere (und deswegen am häufigsten gebrauchte) Schreibweise

```
 $<Formel>$ 
```

Für den Satz von abgesetzten Formeln gibt es die Umgebung `displaymath`. Abgesetzte Formeln erscheinen horizontal zentriert in der Ausgabe. Auch für diese Umgebung existiert neben der üblichen Notation

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

eine kürzere Variante

```
\[
<Formel>
\]
```

#### HINWEIS 1:

*In einigen  $\text{\LaTeX}$ -Quelltexten taucht die Schreibweise  $\$<Formel>\$$  für abgesetzte Formeln auf. Das doppelte Dollarzeichen ist jedoch kein  $\text{\LaTeX}$ - sondern ein `plain-TeX`-Kommando und kann bei einer Verwendung mit  $\text{\LaTeX}$  dazu führen, dass die Abstände vor und nach der abgesetzten Formel nicht korrekt gesetzt werden. Daher ist es empfehlenswert immer die  $\text{\LaTeX}$ -Variante `\[<Formel>\]` zu verwenden.*

Eine weitere Standardumgebung für den Satz von mathematischen Formeln ist die Umgebung `equation`. Deren Funktion ist im Prinzip analog zu der von `displaymath`. Der einzige Unterschied ist, dass Formeln, die in der `equation`-Umgebung gesetzt werden, von  $\text{\LaTeX}$  mit einer fortlaufenden Formelnummer versehen, was man an den später angeführten Beispielen noch sehen wird.

Grundsätzlich lässt sich jede Formel in jeder der beschriebenen Umgebungen setzen. Der Inline-Modus sollte aber nur für unkomplizierte Formeln die nicht zu viel Platz beanspruchen verwendet werden. Komplexere Gebilde wie zum Beispiel das Integralzeichen werden im Inline-Modus zwar auf die entsprechende Größe gestaucht, sehen aber dadurch sehr unschön aus und stören den

Lesefluss, da sich die Zeilenabstände ändern. Für solche Formeln sollte man eine der *display*-Umgebungen wählen.

## 2.2 Besonderheiten des Mathe-Modus

In  $\text{\LaTeX}$  gibt es eine Reihe von Zeichen, die nur im Mathe-Modus erlaubt sind und dort eine spezielle Funktion haben. Außer dem schon erwähnten  $\$$ -Zeichen gehören insbesondere die Zeichen  $\sim$  und  $\_$  dazu. Möchte man diese Zeichen ausgeben, so muss ihnen ein  $\backslash$  vorangestellt werden, durch den sie ihre spezielle Funktion verlieren und als normale Zeichen behandelt werden (bei dem  $\sim$ -Zeichen bewirkt der  $\backslash$ , dass es *über* das folgende Zeichen gesetzt wird). Der  $\backslash$  selbst, den man besonders bei mathematischen Mengenkonstruktionen öfter benötigen wird, kann über den Befehl `\backslashbackslash` ausgegeben werden.

## 2.3 Zusammensetzung mathematischer Formeln

Formeln bestehen aus Konstanten, Variablen und Operatoren, die diese miteinander verknüpfen. Darüber hinaus gibt es eine Vielzahl mathematischer Symbole. Nur wenige der benötigten Zeichen findet man direkt auf der Computertastatur.

Zu beachten ist, dass alle eingegebenen Buchstaben von  $\text{\LaTeX}$  im Mathe-Modus als Variablen interpretiert werden und entsprechend der mathematischen Konventionen kursiv mit einem definierten Absatzand gesetzt werden. Zahlen können direkt über die Tastatur eingegeben werden, genau wie die folgenden mathematischen Symbole:  $+$ ,  $-$ ,  $=$ ,  $/$ ,  $:$ ,  $!$ ,  $'$ ,  $,$ ,  $|$ ,  $($ ,  $)$ ,  $[$  und  $]$ . Alle anderen Zeichen, die in Formeln auftreten können, werden über die entsprechenden  $\text{\LaTeX}$ -Befehle erzeugt.

### HINWEIS 2:

*Konstanten sollten per Konvention immer aufrecht in Schriftart Roman gesetzt werden. Dies kann mit dem Befehl `\mathrm{<Konstante>}` erreichen (ansonsten erscheint die Konstante als Variable).*

# 3 Mathematische Formeln setzen

Nach dieser Einführung geht es in diesem Abschnitt nun um das tatsächliche Setzen von Formeln. Es kann hier natürlich nur ein kleiner Ausschnitt aus den Möglichkeiten, die  $\text{\LaTeX}$  bietet, besprochen werden aber hat man das Muster, nach dem der Formelsatz funktioniert erst einmal verstanden, kann man sich relativ schnell in bisher unbekannte Gebiete einarbeiten.

## 3.1 Exponenten und Indizes

Für Exponenten und Indizes gibt es die oben schon angesprochenen Befehlszeichen  $\sim$  und  $\_$ . Ein Beispiel verdeutlicht an dieser Stelle am besten deren Benutzung:

**L<sup>A</sup>T<sub>E</sub>X CODE**

```

1 \[
2 x^2 + 2x + a_k
3 \]

```

**AUSGABE**

$$x^2 + 2x + a_k$$

Wie man an diesem Beispiel sehen kann, bewirkt das Befehlszeichen `^`, dass das *unmittelbar folgende* Zeichen hochgestellt wird, wo hingegen der `_` eine Tiefstellung des nächsten Zeichens bewirkt. Zu beachten ist, dass Exponenten und Indizes, die aus mehr als einem Zeichen bestehen, in *geschweifte* Klammern gesetzt werden müssen, da sonst nur das erste Zeichen zum Exponenten oder zum Index wird und der Rest in normaler Schriftgröße dahinter erscheint. Ein Beispiel:

**L<sup>A</sup>T<sub>E</sub>X CODE**

```

1 a_{ij} - b_{ij} = x^{27}

```

**AUSGABE**

$$a_{ij} - b_{ij} = x^{27}$$

Das Schöne an L<sup>A</sup>T<sub>E</sub>X ist, dass die Schriftgröße für Exponenten und Indizes automatisch angepasst wird so dass der Benutzer dem System nur mitteilen muss, dass an der entsprechenden Stelle ein Exponent oder ein Index zu folgen hat. Alles andere erledigt L<sup>A</sup>T<sub>E</sub>X. Außerdem ist es möglich Mehrfachindizierung oder Mehrfachexponenzierung zu setzen, wenn die Zeichen entsprechend von Gruppenklammern umschlossen werden. Um dies zu verdeutlichen folgen auch hier wieder Beispiele:

**L<sup>A</sup>T<sub>E</sub>X CODE**

```

1 \[
2 a^j_k \quad \text{(Gleichzeitige Hoch- und Tiefstellung)}
3 \]

```

**AUSGABE**

$$a_k^j \quad (\text{Gleichzeitige Hoch- und Tiefstellung})$$

Der Exponent und der Index beziehen sich also hier, entgegen der Erwartung auf das *a*. Die Reihenfolge, in der der Index und der Exponent auftauchen ist in diesem Fall unwesentlich. Der Code `a_k^j` würde die gleiche Ausgabe liefern. Mit entsprechender Klammerung können wir aber auch folgende Ausgabe erreichen:

**L<sup>A</sup>T<sub>E</sub>X CODE**

```

1 \[
2 a^{\{j\_k\}} \quad \text{(Ein indizierter Exponent)}
3 \]

```

AUSGABE

$$a^{j_k} \quad (\text{Ein indizierter Exponent})$$

Man sieht, dass auch in diesem Fall die passende Schriftgröße automatisch gewählt wird.

Die beiden Befehle `^` und `_` sind sehr vielseitig einsetzbar und können in verschiedenem Kontext Verschiedenes bewirken. So stehen sie beispielsweise in Verbindung mit dem Summenzeichen für den unteren und den oberen Summationsindex und in Verbindung mit einem Integralzeichen entsprechend für die untere und obere Integrationsgrenze.

Ein abgesetztes Summenzeichen wird auf folgende Weise erzeugt

**L<sup>A</sup>T<sub>E</sub>X CODE**

```

1 \[
2 \sum_{\{k=0\}}^{\{10\}} x^{\{k\}}
3 \]

```

AUSGABE

$$\sum_{k=0}^{10} x^k$$

**HINWEIS 3:**

*Dies ist keine direkte Eigenschaft der Befehlszeichen `^` und `_`, sondern folgt aus der Definition von `\sum` und `\int` als **mathematische Operatoren**.*

## 3.2 Brüche

Brüche werden in **L<sup>A</sup>T<sub>E</sub>X** mit dem Befehl `\frac{<Zähler>}{<Nenner>}` erzeugt. Hierbei können im Zähler und Nenner beliebige mathematische Konstrukte auftauchen. Die Länge des Bruchstriches wird in Abhängigkeit von den eingegebenen Konstrukten berechnet.

**L<sup>A</sup>T<sub>E</sub>X CODE**

```

1 \[
2 \frac{\{3\}}{\{4\}}
3 \]

```

AUSGABE

$$\frac{3}{4}$$

aber auch mit komplexeren Brüchen hat **L<sup>A</sup>T<sub>E</sub>X** keine Probleme:



$\LaTeX$  CODE

```

1 \[
2 \frac{\sum_{k=1}^n x^k + k!}{a^2 - b^2}
3 \]

```

## AUSGABE

$$\frac{\sum_{k=1}^n x^k + k!}{a^2 - b^2}$$

Die Schriftgröße in der Zähler und Nenner erscheinen, wird ebenfalls automatisch berechnet. Besonders bei Doppel- und mehrfachbrüchen kann es dabei passieren, dass Teile des Bruches schwer lesbar sind (dies kann auch in einer Umgebung für abgesetzte Formeln passieren, obwohl man intuitiv erwarten würde, dass sich  $\LaTeX$  den nötigen Platz einfach nimmt). Zur Verdeutlichung ein zugegebenermaßen leicht übertriebenes Beispiel:

 $\LaTeX$  CODE

```

1 \[
2 \frac{
3   \frac{4x+5y}{2} - \frac{300z + 45b - 5}{8z^2-2}
4   }
5   {\frac{9y+30z-7}{\frac{5}{7}}}
6   }
7 \]

```

## AUSGABE

$$\frac{\frac{4x+5y}{2} - \frac{300z+45b-5}{8z^2-2}}{\frac{9y+30z-7}{\frac{5}{7}}}$$

In solchen Situationen schafft der `\dfrac`-Befehl des Paketes `amsmath` Abhilfe. Zum Vergleich:

 $\LaTeX$  CODE

```

1 \[
2 \frac{
3   \dfrac{4x+5y}{2} - \dfrac{300z + 45b - 5}{8z^2-2}
4   }
5   {\dfrac{9y+30z-7}{\dfrac{5}{7}}}
6   }
7 \]

```

## AUSGABE

$$\frac{4x + 5y}{2} - \frac{300z + 45b - 5}{8z^2 - 2}$$

$$\frac{9y + 30z - 7}{\frac{5}{7}}$$

### 3.3 Klammern in mathematischen Formeln

In mathematischen Formeln werden oft Klammern benötigt um Prioritäten zu verändern. So kann man zum Beispiel folgendes schreiben

`LATEX` CODE

```
1 \[
2 (a + b)\cdot c = a \cdot c + b \cdot c
3 \]
```

AUSGABE

$$(a + b) \cdot c = a \cdot c + b \cdot c$$

Wenn jedoch komplexere mathematische Konstrukte geklammert werden sollen, kann es bei dieser Art von Klammerung schnell zu ungewollten Ergebnissen kommen wie das folgende Beispiel verdeutlicht:

`LATEX` CODE

```
1 \[
2 1 + (\frac{1}{1-x^2})^3
3 \]
```

AUSGABE

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

Deswegen sollten den entsprechenden Klammersymbolen stets die Befehle `\left` und `\right` vorangestellt werden. Diese veranlassen `LATEX` dazu, eine für die Formel angemessene Klammergröße zu wählen. Die Ausgabe des vorigen Beispiels sieht mit der entsprechenden Modifikation schöner aus. Zum Vergleich:

`LATEX` CODE

```
1 \[
2 1 + \left(\frac{1}{1-x^2}\right)^3
3 \]
```

AUSGABE

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

da `LATEX` nur eine sehr begrenzte Anzahl von Klammergrößen zur Verfügung hat, kann es vorkommen, dass die automatisch gewählten Klammern zum Beispiel zu groß ausfallen. Für solche Fälle wird dem Benutzer die Möglichkeit gegeben, die gewünschte Klammergröße mit den Befehlsvarianten

1. `\bigl`, `\bigr`
2. `\Bigl`, `\Bigr`

3. `\Biggl`, `\Biggr`

selbst anzupassen.

### 3.4 Wurzeln

Wurzeln werden mit dem Befehl `\sqrt[<Grad>]{<Argument>}` gesetzt, wobei der Grad im Falle der Quadratwurzel weggelassen werden kann und im Argument ein beliebiges mathematisches Konstrukt stehen kann. Ein Beispiel:

**L<sup>A</sup>T<sub>E</sub>X CODE**

```

1 \[
2 \sqrt{4} = 2 \quad \sqrt[3]{27} = 3
3 \]
```

**AUSGABE**

$$\sqrt{4} = 2 \quad \sqrt[3]{27} = 3$$

### 3.5 Integrale

Das Integralzeichen kann über den Befehl `\int` gesetzt werden. Bei Bedarf können auch Integrationsgrenzen angegeben werden. Dies geschieht in der selben Weise wie auch schon beim Summenzeichen. Hier ein Beispiel:

**L<sup>A</sup>T<sub>E</sub>X CODE**

```

1 \[
2 \int_{0}^{\infty} f(x) dx
3 \]
```

**AUSGABE**

$$\int_0^{\infty} f(x) dx$$

Über die Befehle `\iint`, `\iiint`, `\idotsint` können Mehrfachintegrale erzeugt werden.

## 4 Anwendungen aus Beispielgebieten der Mathematik

### 4.1 Anwendung: Lineare Algebra

Es sollen hier kurz einige nützliche Umgebungen für den Satz von Vektoren und Matrizen beschrieben werden.

### 4.1.1 Matrizen

In Standard- $\text{\LaTeX}$  werden Matrizen mithilfe der `array`-Umgebung definiert. Hier soll aber der leichtere Weg über die intuitiveren  $\mathcal{A}\mathcal{M}\mathcal{S}$ -Umgebungen für alle gebräuchlichen Arten von Matrizen gegangen werden. Die Art, in der die Matrixelemente in diesen Umgebungen notiert werden, ist vergleichbar mit der Art und Weise auf die Tabellen gesetzt werden. In  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{\LaTeX}$  gibt es sechs verschiedene Umgebungen für Matrizen, die hier alle am Beispiel aufgeführt werden:

$\text{\LaTeX}$  CODE

```

1 \[
2 \begin{matrix}
3 a & b & c \\
4 d & e & f \\
5 g & h & i
6 \end{matrix}
7 \]
```

AUSGABE

$$\begin{matrix} a & b & c \\ d & e & f \\ g & h & i \end{matrix}$$

$\text{\LaTeX}$  CODE

```

1 \[
2 \begin{pmatrix}
3 a & b & c \\
4 d & e & f \\
5 g & h & i
6 \end{pmatrix}
7 \]
```

AUSGABE

$$\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix}$$

$\text{\LaTeX}$  CODE

```

1 \[
2 \begin{bmatrix}
3 a & b & c \\
4 d & e & f
\end{bmatrix}
\]
```

```

5 g & h & i \\
6 \end{bmatrix}
7 \]

```

AUSGABE

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

L<sup>A</sup>T<sub>E</sub>X CODE

```

1 \[
2 \begin{Bmatrix}
3 a & b & c \\
4 d & e & f \\
5 g & h & i \\
6 \end{Bmatrix}
7 \]

```

AUSGABE

$$\begin{Bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{Bmatrix}$$

L<sup>A</sup>T<sub>E</sub>X CODE

```

1 \[
2 \begin{vmatrix}
3 a & b & c \\
4 d & e & f \\
5 g & h & i \\
6 \end{vmatrix}
7 \]

```

AUSGABE

$$\begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix}$$

L<sup>A</sup>T<sub>E</sub>X CODE

```

1 \[
2 \begin{Vmatrix}
3 a & b & c \\
4 d & e & f \\

```

```

5 g & h & i \\
6 \end{Vmatrix}
7 \]

```

AUSGABE

$$\left\| \begin{array}{ccc} a & b & c \\ d & e & f \\ g & h & i \end{array} \right\|$$

Bei Matrizen wie auch manch anderen mathematischen Schreibweisen machen sich Auslassungspunkte oft sehr nützlich. Von diesen gibt es mehrere Varianten in  $\LaTeX$ . Es folgt eine Auflistung der gebräuchlichsten Befehle:

Befehl	Ausgabe	Wann erlaubt
<code>\ldots</code>	...	Textbearbeitungszustand, Mathe Modus
<code>\cdots</code>	...	Mathe Modus
<code>\vdots</code>	⋮	Mathe Modus
<code>\ddots</code>	⋱	Mathe Modus

#### 4.1.2 Vektoren

Da Vektoren Spezialfälle von Matrizen sind, können sie leicht mithilfe der Matrizenumgebungen gesetzt werden. Es gibt noch andere Arten, auf die man einen Vektor darstellen kann aber die hier aufgeführte ist eine der möglichen:

$\LaTeX$  CODE

```

1 \[
2 \vec{v} :=
3 \begin{pmatrix}
4 a \\
5 b \\
6 c
7 \end{pmatrix}
8 \]

```

AUSGABE

$$\vec{v} := \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

#### 4.1.3 Lineare Gleichungssysteme

Nun folgt als letztes Beispiel aus der linearen Algebra noch eine Möglichkeit, wie man lineare Gleichungssysteme mit  $\LaTeX$  setzen kann. Bei diesen genügt es nicht dass die Formeln zentriert untereinander erscheinen, denn die einzelnen Gleichungen müssen aneinander ausgerichtet werden. Dies kann mithilfe

der align-Umgebung geschehen. Diese ist im Prinzip mit der displaymath-Umgebung vergleichbar da die Formeln in dieser Umgebung ebenfalls abgesetzt erscheinen. Sie bietet aber zusätzlich noch die Möglichkeit die Formeln mithilfe von Spaltentrennzeichen aneinander auszurichten und Zeilenumbrüche zu setzen. Hier ein Beispiel für die Verwendung dieser Umgebung:

#### L<sup>A</sup>T<sub>E</sub>X CODE

```

1 \begin{align}
2 2x_1 + 4x_2 + 7x_3 &= 5 \\
3 5x_1 + 9x_2 + 7x_3 &= b \\
4 8x_1 - \frac{1}{2}x_2 + \frac{3}{4}x_3 &= 10 \\
5 \end{align}

```

#### AUSGABE

$$2x_1 + 4x_2 + 7x_3 = 5 \quad (1)$$

$$5x_1 + 9x_2 + 7x_3 = b \quad (2)$$

$$8x_1 - \frac{1}{2}x_2 + \frac{3}{4}x_3 = 10 \quad (3)$$

Wie man an diesem Beispiel sehen kann, versieht L<sup>A</sup>T<sub>E</sub>X jede Zeile der align-Umgebung mit einer eigenen Nummer. Möchte man das unterdrücken aber trotzdem die Vorteile dieser Umgebung nutzen, so sollte man die align\*-Umgebung nutzen, die für solche Anforderungen konzipiert wurde.

#### HINWEIS 4:

Die Nummerierung innerhalb der align-Umgebung ist nicht etwa auf das einzelne Gleichungssystem sondern auf das gesamte Dokument bezogen. So bekommt jede im Dokument auftauchende (nummerierte) Gleichung ihre eigene fortlaufende und somit eindeutige Nummer. Ganz gleich ob sie innerhalb der equation-Umgebung oder der align-Umgebung gesetzt wurde. Dasselbe gilt natürlich auch für die hier nicht vorgestellten anderen nummerierten Umgebungen für abgesetzte Formeln. Die Art und Weise, in der nummeriert wird, ist durch die Dokumentenklasse bestimmt. So erfolgt zum Beispiel bei den Dokumentenklassen report und book die Nummerierung zweistufig (Kapitel und Nummer der Formel seit Anfang des Kapitels). In der Dokumentenklasse article hingegen, erfolgt eine Nummerierung global auf das ganze Dokument bezogen.

## 4.2 Anwendung: Analysis

Nun noch einige wenige Beispiele von mathematischen Konstrukten, wie sie oft in der Analysis angetroffen werden. An dieser Stelle sei angemerkt, dass L<sup>A</sup>T<sub>E</sub>X an Variablensymbolen noch viel mehr zu bieten hat, als nur das lateinische Alphabet. So können zum Beispiel alle griechischen Buchstaben auf intuitive Weise erhalten werden, indem man ihrem Namen einen \ voranstellt. Hierbei ist

darauf zu achten, dass L<sup>A</sup>T<sub>E</sub>X zwischen Groß- und Kleinschreibung unterscheidet. So ergibt zum Beispiel der Befehl `\gamma` das kleine  $\gamma$  wo hingegen der Befehl `\Gamma` das große  $\Gamma$ .

### 4.2.1 Funktionen

Am Beispiel der Betragsfunktion folgen nun einige sehr nützliche Befehle und Umgebungen.

#### L<sup>A</sup>T<sub>E</sub>X CODE

```

1 \[
2 \begin{split}
3 \vert \cdot \vert : \mathbb{R} \rightarrow \mathbb{R}^+ \cup \{0\}, \quad x \in \mathbb{R} \mapsto |x| \in \mathbb{R}^+ \cup \{0\}
4 \to \mathbb{R}^+ \cup \{0\} \quad \cup \{0\},
5 & \quad x \in \mathbb{R} \quad \mapsto
6 \vert x \vert \in \mathbb{R}^+ \cup \{0\}
7 \cup \{0\} \\
8 & \quad x \in \mathbb{R} :=
9 \begin{cases}
10 \phantom{-} x & \text{falls } x \geq 0 \\
11 -x & \text{sonst}
12 \end{cases}
13 \end{split}
14 \]

```

#### AUSGABE

$$|\cdot| : \mathbb{R} \rightarrow \mathbb{R}^+ \cup \{0\}, \quad x \in \mathbb{R} \mapsto |x| \in \mathbb{R}^+ \cup \{0\}$$

$$|x| := \begin{cases} x & \text{falls } x \geq 0 \\ -x & \text{sonst} \end{cases}$$

#### HINWEIS 5:

- Um den Befehl `\mathbb{R}` nutzen zu können, muss in der Präambel vorher das Paket `bm` mit `\usepackage` geladen werden.
- Der Befehl `\phantom{<beliebige Zeichenfolge>}` erzeugt eine leere Box von der Größe des im Argument angegebenen Inhaltes. Er kann für Positionierungskorrekturen in mathematischen Umgebungen verwendet werden.

Hier noch ein Beispiel für die Verwendung von Quantoren:

#### L<sup>A</sup>T<sub>E</sub>X CODE

```

1 Seien  $M$  und  $N$  Mengen. Eine Funktion

```



```

2 $f: M \to N, \quad x \mapsto f(x)$
3 ist \emph{surjektiv}, wenn gilt:
4 \[
5 \forall x, x' \in M: f(x) = f(x') \Rightarrow x = x'
6 \]

```

**AUSGABE**

Seien  $M$  und  $N$  Mengen. Eine Funktion  $f: M \rightarrow N$ ,  $x \mapsto f(x)$  ist *surjektiv*, wenn gilt:

$$\forall x, x' \in M: f(x) = f(x') \Rightarrow x = x'$$

### 4.2.2 Vordefinierte Funktionsnamen und Text innerhalb von mathematischen Formeln

Am Anfang dieser Ausarbeitung wurde bereits erwähnt, dass im Mathe-Modus jeder Buchstabe als Variable interpretiert und auch dementsprechend gesetzt wird. Nun gibt es in der Mathematik aber viele Schreibweisen, und Buchstabenfolgen, die eine genau definierte Bedeutung haben und sich deswegen von Variablen optisch unterscheiden müssen.

Viele dieser Bezeichnungen wurden in  $\text{\LaTeX}$  als Befehle integriert. Deren Aufruf bewirkt, dass die entsprechenden Bezeichnungen in aufrechter Schrift gesetzt werden und somit von den anderen Formelelementen unterscheidbar werden. Ein Beispiel:

 **$\text{\LaTeX}$  CODE**

```

1 \[
2 \lim_{x \to \infty} \frac{1}{\exp(x)} = 0
3 \]

```

**AUSGABE**

$$\lim_{x \rightarrow \infty} \frac{1}{\exp(x)} = 0$$

Mit Hilfe des schon mehrfach erwähnten Befehls `\text{<text>}` kann an man jeder Stelle in einer mathematischen Umgebung kurzzeitig in den Textbearbeitungszustand gewechselt werden. Der als Argument angegebene Text erscheint dann in der normalen, im Dokument verwendeten Schriftart.

Manchmal möchte man innerhalb einer mathematischen Umgebung für abgesetzte Formeln - z. B. inmitten einer Gleichungsfolge - eine möglicherweise längere Textpassage zur Erläuterung einfügen. Der Befehl `\text{<text>}` ist für eine solche Art von Texteschüben eher ungeeignet.

Stattdessen sollte in solchen Situationen der Befehl `\intertext{<Textpassage>}` verwendet werden. Die Textpassage erscheint dann linksbündig als eigene Zeilengruppe. Der Vorteil des Befehls ist, dass bei ausgerichteten Umgebungen die Ausrichtung der Formel erhalten bleibt. Dies wäre mit einem Schließen der Umgebung vor der Textpassage und einem erneuten Öffnen nachdem diese beendet ist, nicht gegeben. Das folgende Beispiel demonstriert die Wirkung des Befehls

`\intertext{<Textpassage>}` anhand einer `align`-Umgebung.

**L<sup>A</sup>T<sub>E</sub>X CODE**

```

1 \begin{align*}
2 (x + iy)(x - iy) &= x^2 - ixy + ixy - i^2 y^2 \\
3 &= y^2 + y^2 \quad \text{\quad} \text{\texttt{\textbackslash}text{da}} \quad \text{\texttt{\textbackslash}}; \quad i^2 = -1 \\
4 \intertext{Dagegen ist}
5 (x + iy)^2 &= x^2 + 2ixy + i^2 y^2 = x^2 + 2ixy - y^2 \\
6 (x - iy)^2 &= x^2 - 2ixy + i^2 y^2 = x^2 - 2ixy - y^2 \\
7 \end{align*}

```

**AUSGABE**

$$\begin{aligned} (x + iy)(x - iy) &= x^2 - ixy + ixy - i^2 y^2 \\ &= y^2 + y^2 \quad \text{da } i^2 = -1 \end{aligned}$$

Dagegen ist

$$\begin{aligned} (x + iy)^2 &= x^2 + 2ixy + i^2 y^2 = x^2 + 2ixy - y^2 \\ (x - iy)^2 &= x^2 - 2ixy + i^2 y^2 = x^2 - 2ixy - y^2 \end{aligned}$$

Die `*`-Variante der `align`-Umgebung bewirkt, dass die Formelnummerierung unterdrückt wird.

**HINWEIS 6:**

*Eine Besonderheit des Befehls `\intertext{<Textpassage>}` ist, dass er nur unmittelbar nach einem Zeilenumbruch (`\`) verwendet werden darf und somit auch nur in Umgebungen, die diesen Zeilenumbruch zulassen.*

## 5 Zeilen- und Seitenumbrüche innerhalb von *display*-Umgebungen

### 5.1 Der Zeilenumbruch in abgesetzten Mathematik-Umgebungen

Wir wissen bereits aus dem Grundlagen-Teil dieses Pamphlets, dass in den Standard Umgebungen für abgesetzte Formeln, die  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  zur Verfügung stellt (`displaymath` und `equation`) ein Zeilenumbruch nicht erlaubt ist.

Wir haben andererseits auch schon Umgebungen kennengelernt, in denen sowohl ein Umbruch als auch eine Anordnung der Formeln nach den Wünschen des Benutzers möglich ist (beispielsweise die `align` Umgebung).

Nun soll eine noch allgemeinere mathematische Umgebung für abgesetzte Formeln diskutiert werden, in der ebenfalls Zeilebumbrüche wie auch eine individuelle Anordnung der Formeln möglich wird.

Diese Umgebung wird vom `amsmath` Paket zur Verfügung gestellt und heißt `split`. Das Besondere an dieser Umgebung ist, dass sie zwar mit

```

\begin{split}
\vdots
\end{split}

```

die gewohnte Syntax hat, jedoch **nie als eigenständige Umgebung geöffnet werden darf**. Die `split`-Umgebung ist eine *eingebettete* Umgebung und darf *nur innerhalb anderer Umgebungen für abgesetzte Formeln* benutzt werden. Dies scheint zunächst eine unbequeme Einschränkung zu sein, jedoch muss man bedenken, dass die `split`-Umgebung in *jeder anderen Umgebung für abgesetzte Formeln* benutzt werden darf (also auch in solchen wie `displaymath` oder `equation`). Diese Eigenschaft ist in der Praxis äußerst nützlich.

Verwendet wird die `split`-Umgebung analog zur `align`-Umgebung.

#### L<sup>A</sup>T<sub>E</sub>X CODE

```

1 \[
2 \begin{split}
3 2x +4y &= 9\\
4 5x - 3y &= 7\\
5 \end{split}
6 \]

```

#### AUSGABE

$$\begin{aligned}
 2x + 4y &= 9 \\
 5x - 3y &= 7
 \end{aligned}$$

#### HINWEIS 7:

Bei dieser Umgebung ist darauf zu achten, dass nach einer Positionierungsmarke am Ende der Zeile ein Zeilenumbruch erfolgen muss es sei denn es handelt sich dabei um die letzte Zeile. Mit anderen Worten: Zwischen zwei aufeinander folgenden Positionierungsmarken muss ein Zeilenumbruch erfolgen.

## 5.2 Der Seitenumbruch innerhalb von abgesetzten Mathematik-Umgebungen

Oft ist es für aufwendigere Rechnungen notwendig, eine längere Gleichungsfolge aufzuschreiben. Dies wird man in L<sup>A</sup>T<sub>E</sub>X üblicherweise innerhalb einer der Umgebungen für abgesetzte Formeln tun. Wenn man sich dabei für eine der  $\mathcal{A}\mathcal{M}\mathcal{S}$ -Umgebungen entscheidet, sollte man bedenken, dass diese Umgebungen selbstständig *keinen Seitenumbruch* vornehmen, auch wenn ein solcher in der konkreten Situation notwendig wäre.

Die Konsequenz daraus ist, dass in einem solchen Fall die *gesamte* Gleichungsfolge auf die nächste Seite gesetzt wird. Dies führt zu unschönen Druckergebnissen, da die Seite vor der `display`-Umgebung dann meist halb leer bleibt.

Ein solches Verhalten war von den Autoren der  $\mathcal{A}\mathcal{M}\mathcal{S}$ -Zusatzpakete durchaus gewollt, denn die Struktur mathematischer Formeln ist oft komplex, so dass ein

Fall, in dem ein Seitenumbruch nötig wird, die Aufmerksamkeit eines menschlichen Betrachters verdient.

Manuell lässt sich der Seitenumbruch natürlich mit entsprechenden Befehlen durchführen. Der meist genutzte Befehl hierfür ist

```
\displaybreak[<Präferenzwert>]
```

Dieser Befehl sollte *immer vor* einem `\` auftauchen und gilt auch *nur für einen Seitenumbruch an genau dieser Stelle*. Der optionale Präferenzparameter ist eine ganze Zahl zwischen 0 und 4, wobei `\displaybreak[0]` bedeutet, dass *es zulässig ist hier einen Seitenumbruch durchzuführen*. Die Befehlsvariante `\displaybreak` ohne Parameter hat dieselbe Wirkung wie `\displaybreak[4]` und *erzwingt einen Seitenumbruch an dieser Stelle*.

Wie bereits erwähnt, bezieht sich der Befehl `\displaybreak` immer nur auf den unmittelbar folgenden `\`. Er müsste also vor jedem Zeilenumbruchsbefehl, an dessen Stelle auch ein Seitenumbruch denkbar wäre, wiederholt werden.

Falls bei *jedem* `\` in der jeweiligen display-Umgebung ein Seitenumbruch denkbar wäre, kann dies global durch den Befehl

```
\allowdisplaybreaks[<Präferenzwert>]
```

am Anfang der Umgebung ausgedrückt werden. Die Angabe dieses Befehls ist gleichwertig mit der Angabe von `\displaybreak[<Präferenzwert>]` vor *jedem* Zeilenumbruch, wobei der bei `\allowdisplaybreaks` angegebene Präferenzwert bei jedem `\displaybreak` übernommen wird.

#### HINWEIS 8:

*Zu Beachten ist bei der Syntax dieser beiden Befehle, dass zwischen dem Befehl und dem [ kein Leerzeichen stehen darf.*

## 6 Referenzieren von Formeln

Nummerierte Formeln können innerhalb eines  $\LaTeX$ -Dokumentes referenziert werden, das sogar *ohne* deren Nummer zu kennen. Wird innerhalb einer *nummerierten* Mathematik-Umgebung der Befehl `\label{<marke>}` notiert, so kann an einer späteren Stelle im Dokument auf die entsprechende Formel Bezug genommen werden. Dies geschieht über den Befehl (`\ref{<marke>}`). Der Benutzer braucht auf diese Weise für die Referenzierung nur den Namen der von ihm selbst vergebenen Marke zu kennen. Intern wird diese Marke von  $\LaTeX$  dann in die tatsächliche Formelnummer umgesetzt. Diese erscheint auch in der Ausgabe. Die Klammern um den `\ref`-Befehl sind notwendig, um die Referenz mit dem Aussehen der Formelnummerierung abzugleichen. In nummerierten mathematischen Umgebungen setzt  $\LaTeX$  die Formelnummer standardmäßig in runde Klammern. Bei der Referenzierung wird jedoch nur die Nummer, nicht aber die umgebenden Klammern ausgegeben.

Die Klammern müssen nicht gesetzt werden, wenn anstelle des Befehls `\ref{<marke>}` der Befehl `\eqref{<marke>}` benutzt wird. Dieser wird vom

amsmath-Paket bereitgestellt. `eqref{<marke>}` setzt die Klammern automatisch und garantiert zudem, dass die Referenz *immer* aufrecht gesetzt wird. Die Wahl des Namens für die Marke ist dem Benutzer überlassen solange die Namensgebung im Dokument eindeutig bleibt.

#### HINWEIS 9:

Im Namen für die Marke können

- Buchstaben
- Ziffern und
- Satzzeichen

aufzutreten. Zu beachten ist außerdem, dass Groß- und Kleinschreibung unterschieden wird.

#### L<sup>A</sup>T<sub>E</sub>X CODE

```

1 \begin{equation}
2 \label{eq:binom}
3 \binom{n}{k} := \frac{n!}{k!(n-k)!}
4 \end{equation}
5 \ldots
6 Hier folgt eine Referenz auf Gleichung~\eqref{eq:binom}.

```

#### AUSGABE

$$\binom{n}{k} := \frac{n!}{k!(n-k)!} \quad (4)$$

... Hier folgt eine Referenz auf Gleichung (4).

In Beispielen wie diesem sollte das ~-Zeichen immer benutzt werden. Es unterbindet einen Zeilenumbruch zwischen dem Wort „Gleichung“ und der Gleichungsnummer.

#### HINWEIS 10:

Wenn ein Dokument, das Referenzen enthält übersetzt wird, sind für das endgültige Druckergebnis immer mindestens zwei Kompilationsläufe notwendig. Gleiches gilt wenn man ein Dokument, das Referenzen enthält verändert. Wenn aber ein erneuter Kompilationslauf notwendig wird, so wird dies stets von L<sup>A</sup>T<sub>E</sub>X in Form einer Warnung mitgeteilt.

## 7 Eine fertige Umgebung für den Satz von Beweisen

Eines der von der  $\mathcal{A}\mathcal{M}\mathcal{S}$  entwickelten Zusatzpakete stellt eine Umgebung zur Verfügung, mit der mathematische Beweise in einer Form gesetzt werden kön-

nen, wie man sie auch in Büchern antrifft.

Um diese benutzen zu können, muss in der Präambel das Zusatzpaket `amsthm` geladen werden. Dies geschieht wie gewohnt mit dem Befehl

```
\usepackage{amsthm}
```

Die Umgebung heißt `proof` und wird mit

```
\begin{proof}[<eigene Überschrift>]
<Beweistext>
\end{proof}
```

benutzt. `proof` ist *keine* mathematische Umgebung, das heißt, dass sich  $\LaTeX$  innerhalb dieser Umgebung im Textbearbeitungsmodus befindet. Man kann also Argumentationen im Fließtext führen und muss für mathematische Formeln eine der mathematischen Umgebungen öffnen.

Ein Vorteil der `proof`-Umgebung ist, dass sie auf die Einbindung von Sprachpaketen reagiert. Würde kein Sprachpaket in der Präambel geladen werden, so würde die `proof` Umgebung am Anfang des Beweises die Ausschrift „*Proof*“ liefern. Mit dem Laden des Paketes `babel` mit der Option `ngerman` (also `\usepackage[ngerman]{babel}`) in der Präambel passiert jedoch folgendes:

#### $\LaTeX$ CODE

```
1 \begin{proof}
2 Hier folgt der Beweis eines beliebigen
3 mathematischen Sachverhaltes
4 \end{proof}
```

#### AUSGABE

*Beweis.* Hier folgt der Beweis eines beliebigen mathematischen Sachverhaltes □

Der Parameter in eckigen Klammern ist optional. Wird er jedoch angegeben so wird die Ausschrift „*Beweis*“ durch seinen Inhalt ersetzt.

## Literatur

- [1] Niedermair, Elke; Niedermair, Michael:  $\LaTeX$  - Das Praxisbuch. Franzis Verlag 2005
- [2] Schlager, Petra; Thibud, Manfred: Wissenschaftlich mit  $\LaTeX$  arbeiten. Pearson Studium 2005
- [3] Goosens, Michel; Mittelbach, Frank; Samarin, Alexander: Der  $\LaTeX$  Begleiter. Pearson Studium 2002
- [4] Grätzer, George: MATH INTO  $\LaTeX$ . Birkhäuser 2000

- [5] Kopka, Helmut:  $\LaTeX$  - Band 1: Einführung. Pearson Studium 2002
- [6] Kopka, Helmut:  $\LaTeX$  - Band 2: Ergänzungen. Pearson Studium 2002