

Der L^AT_EX-Begleiter

Zweite überarbeitete und erweiterte
Auflage

Frank Mittelbach

L^AT_EX3-Project, Mainz, Deutschland

Michel Goossens

CERN, Genf, Schweiz

mit Johannes Braams, David Carlisle,
und Chris Rowley

und Beiträgen von
Christine Detig und Joachim Schrod



ein Imprint von Pearson Education

München • Boston • San Francisco • Harlow, England
Don Mills, Ontario • Sydney • Mexico City
Madrid • Amsterdam

Höhere Mathematik

8.1 Eine Einführung in $\mathcal{A}\mathcal{M}\mathcal{S}$ - \LaTeX	478
8.2 Umgebungen für abgesetzte Formeln	481
8.3 Matrixähnliche Umgebungen	498
8.4 Komplexere Gebilde und Beschriftungen	502
8.5 Symbole mit variablen Formen	508
8.6 Text in Formeln	512
8.7 Feinabstimmung des Formellayouts	514
8.8 Fonts in Formeln.	522
8.9 Symbole in Formeln.	540

Die Standardversion von \LaTeX eignet sich hervorragend, um einfache mathematische Formeln in Dokumente zu integrieren. Wenn man jedoch zahlreiche komplexe abgesetzte Formeln oder sehr anspruchsvolle mathematische Konstrukte setzen will, reicht \LaTeX allein nicht aus. Man kann zwar neue Befehle oder Umgebungen definieren, die das Eintippen der Formeln erleichtern, das ist jedoch nicht die beste Lösung. Die American Mathematical Society (AMS) stellt ein umfangreiches Paket namens `amsmath` zur Verfügung, das die Arbeit an mathematischen Dokumenten zeitsparender und einheitlicher gestaltet.¹ Es bildet das Kernstück einer Sammlung von Paketen, die unter dem Namen $\mathcal{A}\mathcal{M}\mathcal{S}$ - \LaTeX [8] geführt werden und ist Hauptthema dieses Kapitels. Ein nützliches Buch von George Grätzer [62] beschäftigt sich ebenfalls im Detail mit diesen Paketen.

Dieses Kapitel gibt anhand von Beispielen Einblick in einen Großteil der zahlreichen Funktionen dieser sowie einiger nahe verwandter Pakete. Außerdem verweist es auf einzelne andere relevante Pakete und führt in wichtige Hintergründe für das Setzen von Formeln in \TeX ein. Damit deckt es auch einige der Standard- \LaTeX -Funktionen für das Setzen und Gestalten von Formeln ab, und enthält einige allgemeine Tipps für den Formelsatz, obgleich dies nicht das Hauptziel des Kapitels ist.

¹Dieses Paket basiert auf den Erweiterungen zu `plainTeX`, die als $\mathcal{A}\mathcal{M}\mathcal{S}$ - \TeX bekannt sind.

Es ist auch definitiv kein umfassendes Handbuch für den fachgerechten Formelsatz mit \LaTeX . Viele der Beispiele sollen spezielle Sachverhalte anschaulicher machen und sind deshalb weder ein Maßstab für ansprechende Gestaltung noch für gute Mathematik und auch nicht unbedingt für vorbildliche \LaTeX -Programmierung.

Das Buch *Math into Type* [157] von Ellen Swanson ist eine Anleitung, wie man Formeln entsprechend der Ende des 20. Jahrhunderts gültigen US-Praxis setzt. Wie man diese Anleitung mit \TeX oder ebenso mit Standard- \LaTeX umsetzen kann, darüber geben die Kapitel 16–18 des Buches *The \TeX book* [87] von Donald Knuth detailliert Aufschluss.

Für den Großteil des in diesem Kapitel beschriebenen Materials muss man mindestens das `amsmath`-Paket in der Präambel seines Dokumentes laden. Wenn weitere Pakete erforderlich sind, enthalten die jeweiligen Beispiele einen entsprechenden Hinweis. Eine detaillierte Installations- und Gebrauchsanleitung ist in den einzelnen Paketen enthalten.

8.1 Eine Einführung in $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\LaTeX}$

Das $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\LaTeX}$ -Projekt wurde 1987 ins Leben gerufen und drei Jahre später wurde $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\LaTeX}$ in der Version 1.0 veröffentlicht. Es handelte sich um die originale Umsetzung der mathematischen Funktionalitäten von $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\TeX}$ von Michael Spivak in \LaTeX . Sie wurde von Frank Mittelbach und Rainer Schöpf durchgeführt, die als Berater für die American Mathematical Society arbeiteten und dabei von Michael Downes unterstützt wurden, der bei AMS im technischen Bereich tätig war. 1994 wurden zusammen mit David Jones weitere Arbeiten vorgenommen. Diese Arbeiten wurden von Michael Downes koordiniert, und die Pakete wurden seitdem unter seiner Anleitung und der Schirmherrschaft der AMS weiter unterstützt und wesentlich verbessert.¹

*Ein Dankeschön an
einen großartigen
Menschen!*

Michael Downes hätte dieses Kapitel verfasst, wenn er nicht im Frühjahr 2003 gestorben wäre. Ein Großteil des Textes basiert auf der Dokumentation, die er für $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\LaTeX}$ geschrieben hat. Darum ist dies ein aus tiefstem Herzen ausgesprochenes Zeichen der Anerkennung der verbliebenen Autoren dieses Buches, um das Leben und das Werk eines großartigen Freundes und Kollegen zu ehren, mit dem wir gemeinsam viele spannende Programmierabenteuer in den unerforschten Tiefen von \TeX bestanden haben.

*Verfügbare
Paketoptionen*

Das `amsmath`-Paket kennt nur wenige Optionen. Die meisten betreffen lediglich die genaue Positionierung der „Grenzen“ für verschiedene Arten mathematischer Operatoren (Abschnitt 8.4.4) oder die Platzierung der Formel-tags (Abschnitt 8.2.4).

Die folgenden drei Optionen werden oft im `\documentclass`-Befehl als Klassenoptionen verwendet. Sie werden jedoch auch richtig interpretiert, wenn man das `amsmath`-Paket mit dem `\usepackage`-Befehl lädt.

`reqno` Gleichungsnummern (Tags) rechts positionieren (Voreinstellung).

`leqno` Gleichungsnummern (Tags) links positionieren.²

¹Einige Informationen in diesem Kapitel basieren auf der Dokumentation, die mit $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\LaTeX}$ verteilt wird (mit freundlicher Genehmigung der American Mathematical Society).

²Die $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\LaTeX}$ -Dokumentenklassen verwenden die Voreinstellung `leqno`.

`fleqn` Gleichungen mit einem festen Einzug vom linken Rand positionieren anstatt diese zu zentrieren.

Die $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ -Distribution enthält auch Komponenten, die mithilfe des `\usepackage`-Befehls einzeln geladen werden können. Dies sind zum einen kleinere Pakete, die einige der Funktionen des `amsmath`-Paketes separat verfügbar machen:

`amsopn` Stellt `\DeclareMathOperator` zum Definieren neuer Operatornamen bereit, wie etwa `\Ker` und `\esssup`.

`amstext` Stellt den Befehl `\text` zur Verfügung, um Textfragmente im richtigen Schriftgrad zu formatieren.

Die folgenden Pakete erweitern `amsmath` um zusätzliche Funktionen und müssen explizit geladen werden. Sie werden hier lediglich der Vollständigkeit halber aufgeführt. *Erweiterungspakete*

`amscd` Definiert einige Befehle, mit denen sich kommutative Diagramme einfacher erzeugen lassen. Dazu wird die `CD`-Umgebung bereitgestellt (siehe Abschnitt 8.3.4 auf Seite 500). Diagonale Pfeile werden nicht unterstützt.

`amsthm` Bietet eine Methode zum Deklarieren theoremähnlicher Gebilde sowie eine `proof`-Umgebung. Es wurde in Abschnitt 3.3.3 auf Seite 146 besprochen.

`amsxtra` Stellt verschiedene Kleinigkeiten bereit um die Rückwärtskompatibilität zu gewährleisten, wie zum Beispiel `\fracwithdelims`, `\accentedsymbol` sowie Befehle, die Akzente als Hochstellungen positionieren.

`upref` Lässt `\ref` alle Querverweisnummern in einer aufrechten Serifenschrift setzen, unabhängig vom Kontext.

Die wichtigste Dokumentation zu diesen Paketen ist der *User's Guide for the amsmath Package (Version 2.0)* [8].

Die aktuelle $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ -Sammlung umfasst drei Dokumentenklassen: `amsart`, `amsproc` und `amsbook`, entsprechend den $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ -Klassen `article`, `proc` beziehungsweise `book`. Sie wurden zur Formatierung von Manuskripten entwickelt, die bei der AMS [6] eingereicht werden sollen. Es spricht jedoch nichts gegen eine Verwendung für andere Zwecke. Die Klassendateien laden automatisch das `amsmath`-Paket, so dass ein Dokument einfach mit `\documentclass{amsart}` eingeleitet werden kann. Sie werden in diesem Buch nicht eigens beschrieben, da ihre Schnittstelle den $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ -Standardklassen sehr ähnelt. Näheren Aufschluss über ihre Verwendung gibt [6]. *Die $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ -Dokumentenklassen*

Einige Informationen in diesem Kapitel beziehen sich auf eine weitere Paketsammlung der American Mathematical Society und zwar auf die *AMSfonts*-Distribution. Ihre Pakete, die im Folgenden aufgeführt sind, richten verschiedene Schriften und Befehle für den Formelsatz ein. *Die AMSfonts-Sammlung*

amsfonts Definiert die Befehle `\mathfrak` und `\mathbb` und richtet folgende Fonts ein: `msam` (zusätzliche mathematische Zeichen A), `msbm` (zusätzliche mathematische Zeichen B und Blackboard Bold), `eufm` (Euler Fraktur) sowie zusätzliche Schriftgrade von `cmmib` (fettkursiver Mathematikfont und griechische Kleinbuchstaben in Fettdruck) und `cmbasy` (fette mathematische Zeichen und fette Schreibschrift).

amssymb Definiert die Namen aller mathematischen Symbole der AMSfonts-Sammlung. Diese Befehle werden in Abschnitt 8.9 besprochen. Das Paket lädt automatisch das `amsfonts`-Paket.

eufrak Richtet, wie in Abschnitt 7.7.10 besprochen, die Fonts für die Buchstaben der Euler Fraktur ein (`\mathfrak`). Dieses Alphabet wird auch durch das `amsfonts`-Paket bereitgestellt.

eucal Lässt `\mathcal` anstelle der üblichen Computer Modern Schreibschrift die Euler Schreibschrift verwenden (nähere Einzelheiten in Abschnitt 7.7.10).

Alle diese Pakete kennen die Option `psamsfonts`, die \LaTeX anweist, die Y&Y/Blue Sky Research-Version dieser Schriften aus der AMSfonts-Sammlung zu verwenden. Das ist nur sinnvoll, wenn die Fonts auch installiert sind; man erhält sie über CTAN und sie sind in vielen neueren \LaTeX -Distributionen bereits per Voreinstellung enthalten. Die vorgenannten Pakete sind hauptsächlich im *User's Guide to AMSFonts Version 2.2d* [9] dokumentiert.

Einige wichtige Warnhinweise

Vorsicht
bei zerbrechlichen
Befehlen

Viele der in diesem Kapitel beschriebenen Befehle sind zerbrechlich und müssen daher in bewegten Argumenten mit dem Befehl `\protect` geschützt werden (siehe Anhang B.1 auf Seite 927). Wenn also seltsame Fehlermeldungen auftreten, ist die Ursache wahrscheinlich ein fehlendes `\protect`.

Keine Kurzformen
für Umgebungen

Es ist absolut nicht ratsam, Kurzformen für \LaTeX -Umgebungen zu verwenden. Bei den in diesem Kapitel beschriebenen Formelumgebungen von `amsmath` wirken sich solche Kurzformen immer verheerend aus – Finger weg! Aus sehr ähnlichen Gründen lassen sich auch keine Verbatim-Umgebungen oder -Befehle in diesen Umgebungen verwenden. Hier einige Beispiele für verhängnisvolle Deklarationen:

```
\newenvironment{mlt}{\begin{multline}}{\end{multline}}
\newcommand\bga{\begin{gather}} \newcommand\ega{\end{gather}}
```

Beide erzeugen Fehlermeldungen der Art „`\begin{...} ended by ...`“. Man kann aber folgendermaßen Synonyme und Varianten für diese Umgebungen definieren:

```
\newenvironment{mlt}{\multline}{\endmultline}
\newenvironment{longgather}{\allowdisplaybreaks\gather}{\endgather}
```

Dabei müssen sie jedoch die Befehlsform einer existierenden Umgebung als letzten Befehl im „begin-Code“ und den entsprechenden Befehl `\end...` am Anfang des „end-Codes“ aufweisen. Nähere Einzelheiten hierzu sind in Abschnitt A.1.3 erläutert.

<code>equation</code>	<code>equation*</code>	Eine Zeile, eine Formel
<code>multline</code>	<code>multline*</code>	Eine nicht ausgerichtete mehrzeilige Formel, eine Formelnummer
<code>gather</code>	<code>gather*</code>	Mehrere nicht ausgerichtete Formeln
<code>align</code>	<code>align*</code>	Mehrere Formeln mit unterschiedlicher Ausrichtung
<code>flalign</code>	<code>flalign*</code>	Mehrere Formeln: horizontal gedehnte Form von <code>align</code>
<code>split</code>		Eine einfache Ausrichtung innerhalb einer mehrzeiligen Formel
<code>gathered</code>		Eine „Miniseite“ mit nicht ausgerichteten Formeln
<code>aligned</code>		Eine „Miniseite“ mit mehrfacher Ausrichtung

Tabelle 8.1: Formel-Umgebungen des `amsmath`-Paketes

8.2 Umgebungen für abgesetzte Formeln

Das `amsmath`-Paket definiert verschiedene Umgebungen für abgesetzte Formeln. Sie umfassen ein- und mehrzeilige Formeln mit einem oder mehreren Ausrichtungspunkten und verschiedenen Nummerierungsvarianten für Formelteile innerhalb der Umgebungen.

In diesem Abschnitt hat der Begriff „Formel“ eine ganz bestimmte Bedeutung: Er bezeichnet einen *logisch* separaten Abschnitt einer Formelgruppe, der häufig aus Gründen der Bezugnahme nummeriert und außerdem mit einem Label (normalerweise der Nummer in Klammern) versehen ist, das häufig als *Tag* bezeichnet wird.

Tabelle 8.1 zeigt eine vollständige Liste aller Umgebungen, die man zum Setzen abgesetzter Formeln (oder Formelgruppen) benötigt. Die Mehrzahl dieser Umgebungen wird zusammen mit entsprechenden Anwendungsbeispielen in diesem Abschnitt abgehandelt. Soweit das sinnvoll ist, existieren sie auch als Sternformen ohne Nummerierungen oder Tags.

Die Beispiele zu den Ausrichtungsumgebungen verwenden noch weitere Befehle des `amsmath`-Paketes. An dieser Stelle ist ein detailliertes Verständnis ihrer Funktionsweise nicht notwendig, die späteren Abschnitte halten jedoch für interessierte Leser weitere Informationen bereit. Die Darstellungsbreite (`display width`) ist das Maß, das den rechten und linken Rand (oder die Weite) der abgesetzten Formeln definiert. In den Beispielen wird diese Breite durch dünne blue vertikale Linien am linken und rechten Rand der Darstellung angezeigt.

Soweit nicht anders angegeben, sind die Formeln aller Beispiele in diesem Kapitel zentriert und ihre Formelnummern oder Tags stehen rechts (entsprechend der Voreinstellungen des `amsmath`-Paketes). Wenn das `amsmath`-Paket oder die Dokumentenklasse mit der Option `leqno` geladen wird, befinden sich die Tags auf der linken Seite der Formeln.

Bsp.
8-2-1

$$(1) \quad (a + b)^2 = a^2 + 2ab + b^2$$

$$\sin^2 \eta + \cos^2 \eta = 1$$

```
\usepackage[leqno]{amsmath}
\begin{equation} (a+b)^2=a^2+2ab+b^2 \end{equation}
\[ \sin^2\eta+\cos^2\eta = 1 \]
```

Die Option `fleqn` setzt die Formeln mit einem festen Einzug vom linken Rand anstatt in der Kolumne zentriert. Dazu muss normalerweise die Breite

des Einzugs in der Präambel festgelegt werden. Sie hat den Wert der elastischen Länge `\mathindent`, die auf die Breite des Einzugs der ersten Listenebene voreingestellt ist – was wahrscheinlich nicht der gewünschten Größe entspricht! Man beachte die Unterschiede zwischen dem vorigen und dem nächsten Beispiel. In diesem speziellen Fall ist die Option `reqno` eigentlich überflüssig (da sie voreingestellt ist). Sie zwingt die Formelnummer jedoch immer auf die rechte Seite, egal wie die Einstellungen in der Dokumentenklasse lauten.

$$(a + b)^2 = a^2 + 2ab + b^2$$

(1)

$$\sin^2 \eta + \cos^2 \eta = 1$$

```
\usepackage[fleqn,reqno]{amsmath}
\setlength{\mathindent}{1pc}
\begin{equation} (a+b)^2=a^2+2ab+b^2 \end{equation}
\[ \sin^2\eta+\cos^2\eta = 1 \]
```

Bsp.
8-2-2

Wie die weiteren Beispiele noch zeigen, werden `&` und `\`, genau wie in Standard- \LaTeX , als Spalten- und Zeilenbegrenzungen in abgesetzten, ausgerichteten Formeln verwendet. Wie sie genau benutzt werden, ändert sich jedoch je nach `amsmath`-Umgebung (siehe nächster Abschnitt).

8.2.1 Ein Vergleich mit Standard- \LaTeX

Einige der mehrzeiligen Formelumgebungen lassen das Ausrichten von Formelteilen zu. Im Gegensatz zu den \LaTeX -Umgebungen `eqnarray` und `eqnarray*` verwendet das `amsmath`-Paket eine etwas andere und direktere Methode zum Markieren der Ausrichtungspunkte. Die \LaTeX -Umgebung `eqnarray*` ähnelt der Umgebung `array` mit `{rcl}` als Präambel und erfordert daher zwei `&`-Zeichen zum Kennzeichnen der beiden Ausrichtungspunkte. In den entsprechenden `amsmath`-Strukturen gibt es nur einen Ausrichtungspunkt (ähnlich wie bei einer `{rl}`-Präambel), so dass nur ein `&`-Zeichen links des auszurichtenden Symbols (normalerweise ein Relationssymbol) verwendet werden sollte.

Die `amsmath`-Strukturen schreiben feste Abstände an den Ausrichtungspunkten vor, während die `eqnarray`-Umgebung je nach den Einstellungen für die Umgebung `array` zusätzlichen Leerraum erzeugt. Die Unterschiede werden im nächsten Beispiel deutlich, in dem dieselbe Gleichung nacheinander mithilfe der Umgebungen `equation`, `align` und `eqnarray` gesetzt wird. Dabei sind die Abstände in der `eqnarray`-Umgebung, gemessen an den üblichen Standards für das Setzen von Formeln, zu groß.

$$x^2 + y^2 = z^2$$

(1)

$$x^2 + y^2 = z^2$$

(2)

$$x^3 + y^3 < z^3$$

(3)

$$x^2 + y^2 = z^2$$

(4)

$$x^3 + y^3 < z^3$$

(5)

```
\usepackage{amsmath}
\begin{equation}
x^2 + y^2 = z^2
\end{equation}
\begin{align}
x^2 + y^2 &= z^2 \ \backslash
x^3 + y^3 &< z^3
\end{align}
\begin{eqnarray}
x^2 + y^2 &= z^2 \ \backslash
x^3 + y^3 &< z^3
\end{eqnarray}
```

Bsp.
8-2-3

Genau wie in Standard- \LaTeX werden die Zeilenenden in einer `amsmath`-Formel durch einen `\`-Befehl (oder das Ende der Umgebung) festgelegt. Da das Umbrechen von Zeilen in Formeln normalerweise voraussetzt, dass man die Struktur der Formel genau versteht, wird es allgemein als eine Aufgabe angesehen, die über die Fähigkeiten heutiger Software hinausgeht. Eines der letzten größeren Projekte von Michael Downes befasste sich jedoch mit eben diesem Problem. Es führte zu dem Paket `breqn` (nähere Einzelheiten in [43]).

Im Gegensatz zu `eqnarray` erlauben die `amsmath`-Umgebungen entsprechend ihrer Voreinstellung keine Seitenumbrüche zwischen ihren Zeilen (vgl. Abschnitt 8.2.10).

Ein weiterer Unterschied betrifft die Verwendung der Befehle `*` oder `\[länge]` innerhalb von Umgebungen für abgesetzte Formeln. Bei `amsmath` dürfen zwischen `\` und `[` oder `*` keine Leerzeichen stehen, sonst wird das optionale Argument oder der Stern nicht erkannt. Der Grund liegt darin, dass eckige Klammern und Sterne häufig in Formeln vorkommen, und diese Einschränkung soll verhindern, dass eine echte eckige Klammer der Formel fälschlicherweise als Anfang eines optionalen Argumentes interpretiert wird.

*Leerzeichen nach \\
werden nicht ignoriert*

Schließlich gibt es noch eine weniger offensichtliche Änderung, die Anwendern sehr wahrscheinlich keine Probleme bereiten wird: in Standard- \LaTeX ist der Parameter `\mathindent` keine elastische Länge, in `amsmath` wird er jedoch dazu. Die Ursachen und Konsequenzen dieser Änderung werden in der Datei `amsmath.dtx` besprochen, der dokumentierten Quelldatei des `amsmath`-Paketes.

8.2.2 Eine Formel in einer Zeile

Die `equation`-Umgebung erzeugt eine einzelne Formel mit automatisch generierter Nummer (oder generiertem Tag), welche je nach verwendeter Option ganz außen links oder rechts positioniert wird (siehe Abschnitt 8.2.11); `equation*` ergibt dasselbe Ergebnis, jedoch ohne Tag.¹

Die Existenz eines Tags hat dabei keinerlei Auswirkung auf die Positionierung der Inhalte. Wenn in der einen Zeile nicht genügend Platz ist, wird das Tag nach oben oder unten verschoben: in die Zeile davor, wenn die Formelnummern links stehen, und in die nächste Zeile, wenn sie rechts stehen.

Bsp.
8-2-4

$(1) \quad n^2 + m^2 = k^2 \quad p > 2$	<pre> \usepackage[leqno]{amsmath} \begin{equation*} n^2 + m^2 = k^2 \end{equation*} \begin{equation} n^p + m^p \neq k^p \quad \quad p > 2 \end{equation} </pre>
---	--

8.2.3 Eine Formel in mehreren Zeilen: ohne Ausrichtung

Die `multline`-Umgebung ist eine Abwandlung der `equation`-Umgebung für Formeln, die nicht in eine einzelne Zeile passen. In dieser Umgebung muss man die Zeilenumbrüche mit `\` festlegen, da sie nicht automatisch gefunden werden.

¹Standard- \LaTeX verfügt ebenfalls über die Umgebung `equation`, nicht aber über die Sternform `equation*`, da letztere der Standardumgebung `displaymath` entspricht.

Die erste Zeile in dieser Umgebung wird mit einem Einzug vom linken Rand ausgerichtet, und die letzte Zeile mit dem gleichen Einzug vom rechten Rand.¹ Die Größe des Einzugs entspricht dabei dem Wert des Längenparameters `\multlinegap`, der mit den \LaTeX -Befehlen `\setlength` und `\addtolength` verändert werden kann.

In einer `multline`-Umgebung mit mehr als zwei Zeilen wird jede Zeile, außer der ersten und der letzten, einzeln innerhalb der Breite der Darstellung zentriert (es sei denn, man verwendet die Option `fleqn`). Mit dem Befehl `\shoveleft` bzw. `\shoveright` in einer Zeile kann man diese jedoch auch nach links oder rechts ausrichten.

Eine `multline`-Umgebung stellt eine einzige (logische) Formel dar und verfügt daher nur über ein einziges Tag, `multline*` über gar keines. Daher können einzelne Zeilen nicht durch die Befehle `\tag` oder `\notag` verändert werden. Wenn ein Tag vorhanden ist, wird es mit der Voreinstellung `reqno` ganz rechts in der letzten Zeile positioniert, mit der Option `leqno` ganz links in der ersten Zeile.

Erste Zeile einer `multline`

Zentrierte innere Zeile

Eine rechte Innere

Noch eine zentrierte Innere

Weitere zentrierte Innere

Eine linke Innere

Letzte Zeile einer `multline` (1)

```
\usepackage{amsmath}
\begin{multline}
\text{Erste Zeile einer multline} \\
\text{Zentrierte innere Zeile} \\
\shoveright{\text{Eine rechte Innere}} \\
\text{Noch eine zentrierte Innere} \\
\text{Weitere zentrierte Innere} \\
\shoveleft{\text{Eine linke Innere}} \\
\text{Letzte Zeile einer multline}
\end{multline}
```

Bsp.
8-2-5

Das nächste Beispiel zeigt, wie sich `\multlinegap` auswirkt. In der ersten Einstellung stehen die „*dy*“s genau untereinander und es sieht so aus, als fehle bei der ersten Zeile das Tag.

$$\sum_{t \in \mathbf{T}} \int_a^t \left\{ \int_a^t f(t-x)^2 g(y)^2 dx \right\} dy$$

$$= \sum_{t \notin \mathbf{T}} \int_t^a \left\{ g(y)^2 \int_t^a f(x)^2 dx \right\} dy \quad (2)$$

```
\usepackage{amsmath}
\begin{multline} \tag{2}
\sum_{t \in \mathbf{T}} \int_a^t
\biggl\lbracket \int_a^t f(t-x)^2 \,
g(y)^2 \, dx \biggr\rbrace \, dy \\
= \sum_{t \notin \mathbf{T}} \int_t^a
\biggl\lbracket g(y)^2 \int_t^a
f(x)^2 \, dx \biggr\rbrace \, dy
\end{multline}
```

Bsp.
8-2-6

Wenn der Wert des Parameters auf null gesetzt wird, ändert sich wegen des Tags an der zweiten Zeile nichts, während die erste Zeile nach links ver-

¹Man sollte die `multline`-Umgebung niemals für einzeilige Formeln verwenden, da sich die Auswirkungen nicht vorhersehen lassen.

schoben wird. Dadurch wird deutlicher, dass es sich nur um eine einzige Formel handelt.

Bsp.
8-2-7

$$\sum_{t \in \mathbf{T}} \int_a^t \left\{ \int_a^t f(t-x)^2 g(y)^2 dx \right\} dy$$

$$= \sum_{t \notin \mathbf{T}} \int_t^a \left\{ g(y)^2 \int_t^a f(x)^2 dx \right\} dy \quad (2)$$

```
\usepackage{amsmath}
\setlength{\multlinegap}{0pt}
\begin{multline} \tag{2}
\sum_{t \in \mathbf{T}} \int_a^t
\biggl\lbrace \int_a^t f(t-x)^2 \,
g(y)^2 \, dx \biggr\rbrace \, dy \\
= \sum_{t \notin \mathbf{T}} \int_t^a
\biggl\lbrace g(y)^2 \int_t^a
f(x)^2 \, dx \biggr\rbrace \, dy
\end{multline}
```

8.2.4 Eine Formel in mehreren Zeilen: mit Ausrichtung

Wenn eine einfache Ausrichtung für eine einzelne mehrzeilige Formel benötigt wird, ist die `split`-Umgebung fast immer die beste Wahl. Sie markiert den Ausrichtungspunkt in jeder Zeile mit einem einzelnen `&`-Zeichen.

Bsp.
8-2-8

$$(a-b)^3 = (a-b)(a-b)^2$$

$$= (a-b)(a^2 - 2ab + b^2) \quad (1)$$

$$= a^3 - 3a^2b + 3ab^2 - b^3$$

```
\usepackage{amsmath}
\begin{equation}
\begin{split}
(a-b)^3 &= (a-b)(a-b)^2 \\
&= (a-b)(a^2 - 2ab + b^2) \\
&= a^3 - 3a^2b + 3ab^2 - b^3
\end{split}
\end{equation}
```

Da ihr Inhalt immer nur aus einer einzigen (logischen) Formel besteht, erzeugt `split` selbst keine Nummerierung, so dass es auch keine Sternvariante gibt. Soweit erforderlich stellt die äußere Umgebung für abgesetzte Formeln die benötigten Tags zur Verfügung.

Abgesehen von Befehlen wie `\label` oder `\notag`, die kein sichtbares Material erzeugen, sollte eine `split`-Struktur normalerweise die gesamte aufzuteilende Formel umfassen. Diese kann entweder aus einer ganzen `equation`- oder `equation*`-Umgebung bestehen oder aus einer ganzen Zeile einer `gather`- oder `gather*`-Umgebung (siehe Abschnitt 8.2.5).

Durch die (voreingestellte) Option `centertags` wird das Tag (und alles weitere Material der Formel außerhalb der `split`-Umgebung) an der Gesamthöhe des `split`-Materials vertikal zentriert. Durch die Option `tbtags` wird das Tag, sofern es rechts steht, an der letzten Zeile, und wenn es links steht, an der ersten Zeile der `split`-Umgebung ausgerichtet.

Bsp.
8-2-9

$$(a+b)^3 = (a+b)(a+b)^2$$

$$= (a+b)(a^2 + 2ab + b^2)$$

$$= a^3 + 3a^2b + 3ab^2 + b^3 \quad (1)$$

```
\usepackage[tbtags]{amsmath}
\begin{equation}
\begin{split}
(a+b)^3 &= (a+b)(a+b)^2 \\
&= (a+b)(a^2 + 2ab + b^2) \\
&= a^3 + 3a^2b + 3ab^2 + b^3
\end{split}
\end{equation}
```

Im nächsten Beispiel wird die horizontale Position mit dem Befehl `\phantom` justiert. Er wird zunächst in der Präambel verwendet, um ein „unsichtbares Relationssymbol“ in der Breite seines Argumentes (hier: $=$) zu definieren. Innerhalb des Beispiels dient er der Ausrichtung bestimmter Zeilen, indem er sie mit einer „Phantom- oder unsichtbaren Unterformel“ einleitet (siehe Abschnitt 8.7.2 auf Seite 516). Das leere Klammersymbol `\mathord{\{}` entspricht `\mathord{\{}` und dient als unsichtbarer „Buchstabe“ der Breite null, der benötigt wird, um innerhalb von $+h$ den richtigen Abstand herzustellen (ohne `\{}` ergibt sich folgende Darstellung: $+h$).

```

\usepackage{amsmath}
\newcommand{\relphantom}[1]{\mathrel{\phantom{#1}}}
\newcommand{\ve{\varepsilon}} \newcommand{\tve{t_{\varepsilon}}}
\newcommand{\vf{\varphi}} \newcommand{\yvf{y_{\varphi}}}
\newcommand{\bfE{\mathbf{E}}}
\begin{equation}
\begin{split}
f_{h,\ve}(x,y)
&= \ve \bfE_{x,y} \int_0^{\tve} L_{x,y\varphi}(u) \varphi(x) \, du \quad \backslash\backslash
&= h \int L_{x,z} \varphi(x) \rho_x(dz) \quad \backslash\backslash
&\relphantom{=} \{ + h \biggl[
&\quad \frac{1}{\tve}
&\quad \biggl( \bfE_{y} \int_0^{\tve} L_{x,y^x(s)} \varphi(x) \, ds
&\quad - \tve \int L_{x,z} \varphi(x) \rho_x(dz)
&\quad \biggr) + \quad \backslash\backslash
&\relphantom{=} \phantom{\{ + h \biggl[ }
&\quad \frac{1}{\tve}
&\quad \biggl( \bfE_{y} \int_0^{\tve} L_{x,y^x(s)} \varphi(x) \, ds
&\quad - \bfE_{x,y} \int_0^{\tve} L_{x,y\varphi}(s) \varphi(x) \, ds
&\quad \biggr) \biggr]
\end{split}
\end{equation}

```

Man beachte, dass das Tag mit der Formelnummer in die Zeile unterhalb des dargestellten Materials verschoben wurde. Auch wenn das keine sehr kluge Entscheidung zu sein scheint, ist es doch das beste Ergebnis, das die eingebauten automatischen Routinen des Systems hier erzeugen können.

$$\begin{aligned}
 f_{h,\varepsilon}(x,y) &= \varepsilon \mathbf{E}_{x,y} \int_0^{t_\varepsilon} L_{x,y\varphi}(\varepsilon u) \varphi(x) \, du \\
 &= h \int L_{x,z} \varphi(x) \rho_x(dz) \\
 &\quad + h \left[\frac{1}{t_\varepsilon} \left(\mathbf{E}_y \int_0^{t_\varepsilon} L_{x,y^x(s)} \varphi(x) \, ds - t_\varepsilon \int L_{x,z} \varphi(x) \rho_x(dz) \right) + \right. \\
 &\quad \left. \frac{1}{t_\varepsilon} \left(\mathbf{E}_y \int_0^{t_\varepsilon} L_{x,y^x(s)} \varphi(x) \, ds - \mathbf{E}_{x,y} \int_0^{t_\varepsilon} L_{x,y\varphi}(\varepsilon s) \varphi(x) \, ds \right) \right] \quad (1)
 \end{aligned}$$

Bsp.
8-2-10

8.2.5 Formelgruppen ohne Ausrichtung

Die Umgebung `gather` kann zwei oder mehr Formeln in einer einzigen Darstellung zusammenfassen, ohne sie zueinander auszurichten. Dabei wird jede Formel separat innerhalb der Darstellungsbreite zentriert und hat, soweit erforderlich, ein eigenes Tag. Jede Zeile einer `gather`-Umgebung stellt eine einzelne (logische) Formel dar.

Bsp.
8-2-11

$(a + b)^2 = a^2 + 2ab + b^2$	(1)	<code>\usepackage{amsmath}</code>	
$(a + b) \cdot (a - b) = a^2 - b^2$	(2)	<code>\begin{gather}</code>	
		<code>(a + b)^2 = a^2 + 2ab + b^2</code>	<code>\\</code>
		<code>(a + b) \cdot (a - b) = a^2 - b^2</code>	
		<code>\end{gather}</code>	

Mit dem Befehl `\notag` innerhalb einer logischen Zeile lässt sich die Formelnummer für diese Zeile unterdrücken. Die Umgebung `gather*` unterdrückt alle Formelnummern.

Bsp.
8-2-12

$D(a, r) \equiv \{z \in \mathbf{C} : z - a < r\}$		<code>\usepackage{amsmath}</code>	
$\text{seg}(a, r) \equiv \{z \in \mathbf{C} : \Im z < \Im a, z - a < r\}$	(1)	<code>\begin{gather}</code>	
$C(E, \theta, r) \equiv \bigcup_{e \in E} c(e, \theta, r)$	(2)	<code>D(a,r) \equiv \{ z \in \mathbf{C}</code>	
		<code>\colon z - a < r \}</code>	<code>\notag \\</code>
		<code>\operatorname{seg}(a, r) \equiv</code>	
		<code>\{ z \in \mathbf{C} \colon</code>	
		<code>\Im z < \Im a, \ z - a < r \}</code>	<code>\\</code>
		<code>\mathbf{C}(E, \theta, r) \equiv</code>	
		<code>\bigcup_{e \in E} c(e, \theta, r)</code>	
		<code>\end{gather}</code>	

8.2.6 Formelgruppen mit einfacher Ausrichtung

Die Umgebung `align` sollte für zwei oder mehr Formeln in einer einzigen abgesetzten Darstellung mit vertikaler Ausrichtung verwendet werden. Ihre einfachste Form markiert den Ausrichtungspunkt in jeder Zeile mit einem einzelnen `&`-Zeichen, das normalerweise direkt vor dem Relationssymbol steht.

Bsp.
8-2-13

$(a + b)^3 = (a + b)(a + b)^2$	(1)	<code>\usepackage{amsmath}</code>	
$= (a + b)(a^2 + 2ab + b^2)$	(2)	<code>\begin{align}</code>	
$= a^3 + 3a^2b + 3ab^2 + b^3$	(3)	<code>(a + b)^3 &= (a + b) (a + b)^2</code>	<code>\\</code>
		<code>&= (a + b)(a^2 + 2ab + b^2)</code>	<code>\\</code>
		<code>&= a^3 + 3a^2b + 3ab^2 + b^3</code>	
		<code>\end{align}</code>	
$x^2 + y^2 = 1$	(4)	<code>\begin{align}</code>	
$x = \sqrt{1 - y^2}$	(5)	<code>x^2 + y^2 &= 1</code>	<code>\\</code>
		<code>x &= \sqrt{1-y^2}</code>	
		<code>\end{align}</code>	

8.2.7 Mehrfache Ausrichtung: `align` und `flalign`

Eine `align`-Umgebung kann mehr als einen Ausrichtungspunkt enthalten. Sie enthält so viele Spaltenpaare wie nötig und ähnelt einer `array`-Umgebung mit einer Präambel der Form `{r1r1...}`. Wenn sie aus n solcher `rl`-Spaltenpaare

auch nur innerhalb von Formelumgebungen genutzt und sie erzeugen selbst keine Tags.

$$\left. \begin{array}{l} x^2 + y^2 = 1 \\ x = \sqrt{1 - y^2} \\ \text{sowie } y = \sqrt{1 - x^2} \end{array} \right\} \begin{array}{l} (a + b)^2 = a^2 + 2ab + b^2 \\ (a + b) \cdot (a - b) = a^2 - b^2 \end{array} \quad (1)$$

```
\usepackage{amsmath}
\begin{equation}
\begin{aligned}
x^2 + y^2 &= 1 && \\\
x &= \sqrt{1-y^2} && \\\
\text{sowie } y &= \sqrt{1-x^2} && \\
\end{aligned}
\end{equation}
\begin{gathered}
(a + b)^2 = a^2 + 2ab + b^2 \quad \\\
(a + b) \cdot (a - b) = a^2 - b^2
\end{gathered}
```

Bsp.
8-2-18

Dieselben Formeln lassen sich, wenn auch nicht sehr schön, ebenso mit unterschiedlicher vertikaler Ausrichtung in den Umgebungen setzen.

$$\left. \begin{array}{l} x^2 + y^2 = 1 \\ x = \sqrt{1 - y^2} \\ \text{sowie } y = \sqrt{1 - x^2} \end{array} \right\} \begin{array}{l} (a + b)^2 = a^2 + 2ab + b^2 \\ (a + b) \cdot (a - b) = a^2 - b^2 \end{array} \quad (1)$$

```
\usepackage{amsmath}
\begin{equation}
\begin{aligned}[b]
x^2 + y^2 &= 1 && \\\
x &= \sqrt{1-y^2} && \\\
\text{sowie } y &= \sqrt{1-x^2} && \\
\end{aligned}
\end{equation}
\begin{gathered}[t]
(a + b)^2 = a^2 + 2ab + b^2 \quad \\\
(a + b) \cdot (a - b) = a^2 - b^2
\end{gathered}
```

Bsp.
8-2-19

Sie werden auf vielfältige Weise verwendet, z.B. um berühmte Formeln kreativ und sinnvoll zu gruppieren. Nebenbei gehören diese Formelumgebungen, die Miniseiten erzeugen, zu den wenigen Umgebungen des amsmath-Paketes, die so robust sind, dass man sie ohne Probleme innerhalb anderer Definitionen verwenden kann, wie im folgenden Beispiel:

```
\usepackage{amsmath,bm}
\newenvironment{rcase}{\left.\begin{aligned}}
{\end{aligned}\right\}}
\begin{equation*}
\begin{rcase} \bm{B}' &= -c\nabla\times\bm{E} && \\\
&\bm{E}' &= c\nabla\times\bm{B} - 4\pi\bm{J}\,, && \\
\end{rcase} \quad \text{\text{Maxwell'sche Gleichungen}}
\end{equation*}
```

$$\left. \begin{array}{l} \mathbf{B}' = -c\nabla \times \mathbf{E} \\ \mathbf{E}' = c\nabla \times \mathbf{B} - 4\pi\mathbf{J} \end{array} \right\} \text{Maxwell'sche Gleichungen}$$

Bsp.
8-2-20

Mithilfe des Befehls `\minalignsep` kann man auch den Abstand zwischen den Spaltenpaaren einer `aligned`-Umgebung einstellen, wie im nachfolgenden Beispiel zu sehen.

`\abovedisplayshortskip`, `\belowdisplayshortskip` Der (normalerweise kleinere) vertikale Abstand, der oberhalb und unterhalb einer „kurzen abgesetzten Formel“ zugegeben wird (`Opt plus 3pt` bzw. `6pt plus 3pt minus 3pt`). Unter einer *kurzen abgesetzten Formel* versteht man eine Formel, die rechts vom Ende der vorhergehenden Textzeile beginnt.

Bei genauerem Hinsehen kann man die Auswirkungen dieser Längenparameter im folgenden Beispiel erkennen. Die zweite Formel weist geringere Abstände auf, da der Text oberhalb nicht mit ihr überlappt.

Bisher gilt das Folgende:

$$X = a \quad a = c$$

und daher gilt

$$X = c \quad (1)$$

Und so ist die Formel nur von wenig Platz umgeben!

```
\usepackage{amsmath}
```

Bisher gilt das Folgende:

```
\[ X = a \quad a = c \]
```


und daher gilt

```
\begin{equation} X = c \end{equation}
```

Und so ist die Formel nur von wenig Platz umgeben!

Bsp.
8-2-23

Da die vier Parameter `\abovedisplay..` und `\belowdisplay..` vom aktuellen Schriftgrad abhängig sind, können sie nicht mithilfe des Befehls `\setlength` in der Dokumentenpräambel geändert werden. Stattdessen müssen ihre Einstellungen in den Definitionen von `\normalsize`, `\small` usw. angepasst werden, was normalerweise in einer Dokumentenklasse geschieht.

Vorsicht 
bei Leerzeilen um
abgesetzte Formeln

Viele Autoren sind so umsichtig, ihr Quelldokument durch Leerzeilen zwischen bestimmten Teilstücken übersichtlicher zu gestalten. In den meisten Fällen, wie z.B. vor und nach Überschriften, stören diese Leerzeilen auch nicht. Das ist jedoch nicht immer so. Besonders um und innerhalb von Umgebungen zur Darstellung abgesetzter Formeln muss man sehr vorsichtig sein: Eine Leerzeile vor einer solchen Umgebung wird anders formatiert als erwartet, denn sie wird in einen leeren Absatz umgewandelt, der nur die unsichtbare Box für den Einzug des Absatzes enthält. Daher ist die nächste Darstellung von Leeräumen der Größe `..displayshortskip` umgeben. Insgesamt ergibt sich also ein ziemlich großer (möglicherweise zu großer) Abstand oberhalb der Formeln (eine ganze Leerzeile plus `\abovedisplayshortskip`) und ein ziemlich kleiner Abstand unterhalb, wie das Beispiel zeigt.

Leerzeile oberhalb der Darstellung:

$$a \neq b$$

In beiden Fällen zu viel Platz oberhalb!

$$a \neq b$$

... und nicht viel Platz unterhalb!

(1)

```
\usepackage{amsmath}
```

Leerzeile oberhalb der Darstellung:

```
\[ a \neq b \]
```

In beiden Fällen zu viel Platz oberhalb!

```
\begin{equation} a \neq b \end{equation}
```

```
\ldots\ und nicht viel Platz unterhalb!
```

Bsp.
8-2-24

Wenn das `amsmath`-Paket geladen ist, zeigen alle Umgebungen für abgesetzte Formeln dieses Verhalten. In Standard- \LaTeX erscheinen die Abstände

im Zusammenhang mit `\[` jedoch seltsamerweise mehr oder weniger richtig.

Leerzeile oberhalb der Darstellung:

$$a \neq b$$

Weißraum nun korrekt, aber nicht hier:

$$a \neq b$$

In diesem Fall weniger Platz danach!

Leerzeile oberhalb der Darstellung:

```
\[ a \neq b \]
```

Weißraum nun korrekt, aber nicht hier:

```
\begin{equation} a \neq b \end{equation}
```

In diesem Fall weniger Platz danach!

Bsp.
8-2-25

(1)

Insgesamt lässt sich also sagen: Niemals Leerzeilen ober- oder unterhalb abgesetzter Formeln verwenden!

Automatische Seitenumbrüche vor und nach jeder Umgebung für abgesetzte Formeln werden durch zwei interne Penalty-Parameter gesteuert: `\predisplaypenalty` für Umbrüche vor einer Umgebung ist auf den Wert 10000 voreingestellt, das heißt Umbrüche sind dort nicht zulässig; `\postdisplaypenalty` für Umbrüche nach einer Umgebung ist auf 0 voreingestellt, d.h. Umbrüche sind erlaubt. Die Werte entsprechen den Vorgaben von Standard- \LaTeX und werden durch `amsmath` nicht geändert.

Seitenumbrüche um abgesetzte Formeln...

Im Gegensatz zu Standard- \LaTeX sind die `amsmath`-Umgebungen für abgesetzte Formeln jedoch so voreingestellt, dass sie keine Seitenumbrüche zwischen den Zeilen einer Umgebung erlauben. Das liegt daran, dass korrekte Seitenumbrüche an diesen Stellen sehr stark vom Aufbau der einzelnen Formeln abhängen, so dass sie häufig der individuellen Aufmerksamkeit des Autors bedürfen.

... und zwischen den Formeln

In `amsmath` lassen sich manuelle Seitenumbrüche am besten mit dem Befehl `\displaybreak` vornehmen. Er sollte jedoch nur dort eingesetzt werden, wo er unbedingt benötigt wird, um innerhalb einer Formeldarstellung eine Seite zu umbrechen. Der Befehl muss vor dem `\\`-Befehl stehen, an dem der Seitenumbruch erfolgen darf. Dabei gilt er nur für diese Zeile und darf auch nur in Umgebungen eingesetzt werden, die eine vollständige, abgesetzte Formel erzeugen. Ähnlich wie der Standard- \LaTeX -Befehl `\pagebreak` (siehe Abschnitt 6.2.2 in [106]) besitzt `\displaybreak` ein optionales Argument, das eine ganze Zahl im Wert von null bis vier sein kann, wobei der Wert die Dringlichkeit des Seitenumbruchs anzeigt: `\displaybreak[0]` bedeutet „hier darf umbrochen werden“ ohne den Seitenumbruch zu forcieren; `\displaybreak` ohne optionales Argument entspricht `\displaybreak[4]` und erzwingt einen Seitenumbruch. Mit dem gleichen Befehl lässt sich jedoch weder die Wahrscheinlichkeit für einen Umbruch senken, noch ein Umbruch verhindern. Man beachte, dass es sinnlos ist einen Umbruch in einer „Miniseiten-Darstellung“ vorzunehmen, da diese Umgebungen niemals auf zwei Seiten verteilt werden.

Genau wie das Einfügen von Zeilen- und Seitenumbrüchen in normalen Texten, geben derlei Korrekturen dem Dokument den letzten Schliff. Sie sollten daher auch erst durchgeführt werden, wenn es so gut wie fertig gestellt ist. Andernfalls kann es passieren, dass man immer und immer wieder nachbessern muss, sobald sich der Inhalt des Dokumentes ändert.

Der Befehl `\allowdisplaybreaks`, der die \LaTeX -üblichen Geltungsbereichsregeln befolgt, wirkt wie ein `\displaybreak`-Befehl vor jedem Zeilenende aller Formelumgebungen. Er besitzt das gleiche optionale Argument wie `\displaybreak`. Seitenumbrüche lassen sich innerhalb des Geltungsbereiches eines `\allowdisplaybreaks`-Befehls mit `*` unterdrücken.

Ein `\displaybreak`-Befehl setzt sich sowohl über die Voreinstellung als auch über einen geltenden `\allowdisplaybreaks`-Befehl hinweg.

8.2.11 Formelnummern und Tags

Die Tags für Formeln werden in \LaTeX normalerweise automatisch generiert und enthalten eine gedruckte Entsprechung des \LaTeX -Zählers `equation`. Das geschieht in drei Stufen: Zunächst wird der Wert des `equation`-Zählers gesetzt (normalerweise durch Erhöhen), dann wird das Tag formatiert und dann wird es an der richtigen Position gedruckt.

In der Praxis sind die beiden ersten Vorgänge fast immer miteinander verknüpft. Der Wert des `equation`-Zählers wird also nur erhöht, wenn das Tag mit seiner Entsprechung automatisch gedruckt wird. Wenn eine Umgebung für abgesetzte Formeln Varianten mit und ohne Stern enthält, erzeugen diejenigen ohne Stern automatisch ein Tag für jede logische Formel (und erhöhen den `equation`-Zähler), die Sternformen jedoch nicht.

Bei diesen Varianten lässt sich das Tag (und damit die Erhöhung des Zählerwertes) für eine bestimmte logische Formel mithilfe des Befehls `\notag` (oder `\nonumber`¹) vor dem `\\`-Befehl unterdrücken. Mithilfe des Befehls `\tag` vor `\\` kann man das voreingestellte, automatische Tag durch ein selbst gestaltetes ersetzen (oder ein neues einfügen). Das Argument dieses Befehls kann jeden beliebigen normalen Text enthalten, der dann (in normalen Klammern) als Tag für diese Formel gesetzt wird.

Man beachte: Auch durch den Befehl `\tag` wird der Wert des Zählers nicht erhöht. Das voreingestellte Tag hat also nur das gleiche Erscheinungsbild wie `\tag{\theequation}`, es handelt sich jedoch um unterschiedliche Formen. Die Sternvariante `\tag*` setzt den Text ihres Argumentes ohne Klammern (und ohne irgendwelche sonstigen Zeichen, die möglicherweise durch eine bestimmte Dokumentenklasse vorgegeben sind).

	<code>\usepackage{amsmath}</code>			
$x^2 + y^2 = z^2$	(1)	<code>\begin{align}</code>		
$x^3 + y^3 = z^3$		<code>x^2+y^2 &= z^2 \label{eq:A}</code>	<code>\\</code>	
$x^4 + y^4 = r^4$	(*)	<code>x^3+y^3 &= z^3 \notag</code>	<code>\\</code>	
$x^5 + y^5 = r^5$	*	<code>x^4+y^4 &= r^4 \tag{**}</code>	<code>\\</code>	
$x^6 + y^6 = r^6$	(1')	<code>x^5+y^5 &= r^5 \tag*{**}</code>	<code>\\</code>	
$A_1 = N_0(\lambda; \Omega') - \phi(\lambda; \Omega')$	(2)	<code>x^6+y^6 &= r^6 \tag{\ref{eq:A}\$'}</code>	<code>\\</code>	
$A_2 = \phi(\lambda; \Omega') \phi(\lambda; \Omega)$	AUCH (2)	<code>A_1 &= N_0 (\lambda ; \Omega')</code>	<code>- \phi (\lambda ; \Omega')</code>	
		<code>A_2 &= \phi (\lambda ; \Omega')</code>	<code>\, \phi (\lambda ; \Omega)</code>	
		<code>A_3 &= \mathcal{N} (\lambda ; \omega)</code>	<code>\tag*{AUCH (\theequation)} \\</code>	
		<code>\end{align}</code>		

Bsp.
8-2-26

¹Die Befehle `\notag` und `\nonumber` sind austauschbar.

Das Besondere an diesem Beispiel ist, dass mithilfe der Befehle `\label` und `\ref` in Zeilen 1 und 5 versucht wird, eine Art „relativer Nummerierung“ der Formeln herzustellen.

Um Querverweise auf Formeln zu erleichtern, setzt der Befehl `\eqref` (siehe Beispiel 8-2-30 auf Seite 497) die Formelnummer automatisch in Klammern und fügt, falls erforderlich, eine Kursivkorrektur hinzu. In Abschnitt 2.4 auf Seite 69 werden außerdem allgemeinere Lösungen für den Umgang mit Querverweisen behandelt.

Querverweise auf Formeln

8.2.12 Feinpositionierung von Tags

Die optimale Positionierung von Formeltags kann in mehrzeiligen Darstellungen zu einer recht komplexen Aufgabe werden. Die Umgebungen für abgesetzte Formeln versuchen unter allen Umständen zu vermeiden, dass die Formel die Formelnummer überdrückt. Dazu wird das Tag, soweit erforderlich, sogar nach oben oder unten in eine eigene Zeile verschoben. Die Schwierigkeit eine abgesetzte Formel korrekt zu gestalten kann hin und wieder dazu führen, dass die Position eines Tags nachgebessert werden muss. Das nächste Beispiel zeigt, was passieren kann, und wie man solche Schwierigkeiten behebt. Die automatisch gewählte Position lässt eindeutig zu wünschen übrig.

```
\usepackage{amsmath}
\begin{equation}
\begin{split}
\lvert I_2 \rvert & \rvert \int_0^T \psi(t) \left\{ u(a, t) - \int_{\gamma(t)}^a \frac{d\theta}{k}(\theta, t) \int_a^\theta c(\xi) u_t(\xi, t) d\xi \right\} dt \rvert \\
& \leq C_6 \left\| \int_\Omega \widetilde{S}_{a,-}^{-1,0} W_2(\Omega, \Gamma_l) \right\| \left\| |u| \overset{\circ}{\rightarrow} W_2^{\bar{A}}(\Omega; \Gamma_r, T) \right\|
\end{split}
\end{equation}
```

Bsp.
8-2-27

$$\begin{aligned}
 |I_2| &= \left| \int_0^T \psi(t) \left\{ u(a, t) - \int_{\gamma(t)}^a \frac{d\theta}{k}(\theta, t) \int_a^\theta c(\xi) u_t(\xi, t) d\xi \right\} dt \right| \\
 &\leq C_6 \left\| \int_\Omega \widetilde{S}_{a,-}^{-1,0} W_2(\Omega, \Gamma_l) \right\| \left\| |u| \overset{\circ}{\rightarrow} W_2^{\bar{A}}(\Omega; \Gamma_r, T) \right\|
 \end{aligned} \tag{1}$$

Einen recht einfachen Weg, das Erscheinungsbild einer solchen Formel zu verbessern, bietet die Umgebung `align` mit dem Befehl `\notag` in der ersten Formelzeile:

```
\begin{align}
\lvert I_2 \rvert &= \left\lvert \int_0^T \psi(t) \left\{ u(a,t) - \int_{\gamma(t)}^a \frac{d\theta}{k}(\theta,t) \int_a^\theta c(\xi) u_t(\xi,t) d\xi \right\} dt \right\rvert && \notag \\
&\dots && \\
&\leq C_6 \left\| f \int_\Omega \tilde{S}_{a,-}^{-1,0} W_2(\Omega, \Gamma_l) \right\| \left\| |u| \overset{\circ}{\rightarrow} W_2^{\tilde{A}}(\Omega; \Gamma_r, T) \right\| && \\
&\dots && \\
\end{align}
```

Das Ergebnis kann sich sehen lassen, aber es missbraucht das logische Markup - indem es davon ausgeht, dass die Formelnummern rechts stehen!

$$\left| I_2 \right| = \left| \int_0^T \psi(t) \left\{ u(a,t) - \int_{\gamma(t)}^a \frac{d\theta}{k}(\theta,t) \int_a^\theta c(\xi) u_t(\xi,t) d\xi \right\} dt \right|$$

$$\leq C_6 \left\| f \int_\Omega \tilde{S}_{a,-}^{-1,0} W_2(\Omega, \Gamma_l) \right\| \left\| |u| \overset{\circ}{\rightarrow} W_2^{\tilde{A}}(\Omega; \Gamma_r, T) \right\| \quad (1)$$

Bsp.
8-2-28

Es gibt außerdem den Befehl `\raisetag`, der die vertikale Position der aktuellen Formelnummer weiter justiert, allerdings *nur* wenn sie automatisch von ihrer „normalen Position“ verschoben wurde. Um ein solches Tag z.B. um 6pt anzuheben¹, könnte man `\raisetag{6pt}` schreiben. Man kann versuchen die vorige Formel mit `\raisetag` anzupassen. Es ist jedoch gar nicht so einfach, den richtigen Wert zu finden: Hier scheint ein Wert von `1.2\baselineskip` zu passen!

Das nächste Beispiel zeigt eine sinnvollere Verwendung: Mithilfe von `\raisetag` mit einem negativen Argument wird das linksseitige Tag nach unten in die abgesetzte Formel verschoben.

```
(I) \usepackage[leqno]{amsmath}
\begin{gather}
\text{Vorzeichenfunktion: } \mathcal{S}(x) = \begin{cases} -1 & x < 0 \\ 0 & x = 0 \\ 1 & x > 0 \end{cases} && \begin{aligned}
&\text{\text{Vorzeichenfunktion: } \ } \\
&\mathcal{S}(x) = \begin{cases} -1 & \& x < 0 \\ 0 & \& x = 0 \\ 1 & \& x > 0 \end{cases} \\
&\end{cases}
\end{aligned} \\
\end{gather}
```

Bsp.
8-2-29

Hier wurde eine `gather`-Umgebung mit einer einzelnen Zeile verwendet, da der `\raisetag`-Befehl (nur) in der `equation`-Umgebung unglücklicherweise keine Auswirkungen zeigt (er ist in Low-Level- \TeX programmiert).

Solche Korrekturen gehören zum „letzten Schliff“, genau wie das Einfügen von Zeilen- und Seitenumbrüchen in normalen Texten. Sie sollten daher

¹Die Beschreibung in der Datei `amsmath.dtx` lässt annehmen, dass ein positiver Wert das Tag immer seiner „normalen Position“ annähert, also nach unten für linksseitige Tags. Das ist in der derzeitigen Version jedoch nicht der Fall.

auch erst durchgeführt werden, wenn das Dokument so gut wie fertig ist. Andernfalls kann es passieren, dass man immer und immer wieder nachbessern muss, sobald sich der Inhalt des Dokumentes ändert.

8.2.13 Untersequenzen in der Nummerierung

Das `amsmath`-Paket verfügt über die Umgebung `subequations`, welche die „Unternummerierung von Formeln“ mit Tags der Form (2a), (2b), (2c) usw. unterstützt. Alle mit Tags versehenen Formeln dieser Umgebung werden mithilfe der folgenden zwei normalen \LaTeX -Zähler nummeriert: `parentequation` und `equation`.

Das nächste Beispiel zeigt, dass sich das Tag bis zu einem gewissen Grad umdefinieren lässt. Dabei ist zu beachten, dass die geänderte Definition für `\theequation` innerhalb der `subequations`-Umgebung stehen muss! (Veränderungen an Zählern werden in Anhang A.1.4 besprochen.)

$f = g \quad (1a)$ $f' = g' \quad (1b)$ $\mathcal{L}f = \mathcal{L}g \quad (1c)$	<pre> \usepackage{amsmath} \begin{subequations} \label{eq:1} \begin{align} f &= g & \label{eq:1A} \\ f' &= g' & \label{eq:1B} \\ \mathcal{L}f &= \mathcal{L}g & \label{eq:1C} \end{align} \end{subequations} \begin{subequations} \label{eq:2} \renewcommand\theequation{\theparentequation\roman{equation}} \begin{align} f &= g & \label{eq:2A} \\ f' &= g' & \label{eq:2B} \\ \mathcal{L}f &= \mathcal{L}g + K & \label{eq:2C} \end{align} \end{subequations} </pre>	<p>Man beachte die Beziehung zwischen (1) und (2): nur 1c und 2iii sind verschieden.</p> <p>Man beachte die Beziehung zwischen <code>\eqref{eq:1}</code> und <code>\eqref{eq:2}</code>: nur <code>\ref{eq:1C}</code> und <code>\ref{eq:2C}</code> sind verschieden.</p>
--	---	---

Bsp.
8-2-30

Die Umgebung `subequations` muss die abgesetzten Formeln, auf die sie sich auswirkt, *umschließen*. Sie selbst sollte außerdem nicht in sich selbst verschachtelt sein. Jede `subequations`-Umgebung erhöht den „Hauptzähler“ um eins. Ein `\label`-Befehl innerhalb der Umgebung, aber außerhalb der einzelnen (logischen) Formeln, erzeugt einen `\ref`-Verweis auf die Hauptnummer (z.B. auf 2 und nicht auf 2i).

8.2.14 Zurücksetzen des Formelzählers

Es ist üblich, Formeln innerhalb von Abschnitten oder Kapiteln mit Tags wie (1.1), (1.2), (1.3) ..., (2.1), (2.2) ... zu nummerieren. Diese Art der Nummerierung lässt sich in `amsmath` leicht mithilfe der `\numberwithin`-Deklaration erzielen.¹ So ergibt beispielsweise `\numberwithin{equation}{section}` aus

¹Wie der Name andeutet, kann `\numberwithin` auf jedes Zählerpaar angewendet werden, wobei das Ergebnis aufgrund möglicher Komplikationen nicht immer zufrieden stellend ist. Man vergleiche die Informationen zum Befehl `\@addtoreset` in Anhang A.1.4.

mehreren Elementen zusammengesetzte Formelzeilen, welche die Abschnittsnummer enthalten, wobei gleichzeitig der Formelzähler am Anfang jedes neuen Abschnitts zurückgesetzt wird.

8.3 Matrixähnliche Umgebungen

Das `amsmath`-Paket bietet eine Reihe von matrixähnlichen Umgebungen, die alle der Umgebung `array` in Syntax und Layout gleichen. Komplexe mathematische Layouts auf diese Weise zu betrachten ist eine gute Übung, da sich recht viele zweidimensionale mathematische Strukturen und tabellenähnliche Anordnungen so beschreiben lassen.

Deaktivierte alte Befehle

Drei dieser Umgebungen ersetzen alte Befehle, die tief in Standard- \LaTeX verborgen liegen: die Umgebungen `cases` (im nächsten Abschnitt erläutert) sowie `matrix` und `pmatrix` (im darauf folgenden Abschnitt erläutert). Da die alten Befehlsformen eine völlig andere Schreibweise benutzen, gehören sie nicht wirklich zu \LaTeX und können auch nicht gemeinsam mit den hier beschriebenen Umgebungen verwendet werden. Das `amsmath`-Paket erzeugt sogar eine ausführliche Fehlermeldung, wenn einer der alten Befehle benutzt wird (siehe Seite 946). Wenn man andererseits den Fehler begeht, die Umgebungen des `amsmath`-Paketes zu verwenden, ohne dieses zuvor geladen zu haben, erhält man wahrscheinlich die folgende Fehlermeldung: „Misplaced alignment tab character &“ (falsch platziertes Ausrichtungszeichen &).

8.3.1 Die `cases`-Umgebung

Konstruktionen wie die folgende, wo eine einzige Gleichung einige Varianten hat, kommen in der Mathematik häufig vor. Für solche Konstruktionen stellt `amsmath` die `cases`-Umgebung bereit. Sie erzeugt ein beschriftetes Array mit zwei linksbündigen Spalten.

$$P_{r-j} = \begin{cases} 0 & \text{für } r - j \text{ ungerade,} \\ r! (-1)^{(r-j)/2} & \text{für } r - j \text{ gerade.} \end{cases} \quad (1)$$

```
\usepackage{amsmath}
\begin{equation} P_{r - j} =
\begin{cases}
0 & \text{für } \$r - j\$ ungerade,} \\
r! \, \, (-1)^{\{(r - j)\}/2} & \text{für } \$r - j\$ gerade.}
\end{cases}
\end{equation}
```

Bsp.
8-3-1

Man beachte den `\text`-Befehl und den in den Text „eingebetteten Formelmodus“. Mithilfe der Umgebung `aligned` lassen sich weitere `cases`-ähnliche Umgebungen definieren, wie etwa in Beispiel 8-2-20 auf Seite 490.

8.3.2 Die Matrixumgebungen

Die Matrixumgebungen ähneln den `array`-Umgebungen von \LaTeX . Im Unterschied zu diesen besitzen sie jedoch kein Argument, in dem das Format der Spalten bestimmt wird. Stattdessen wird ein Standardformat mit bis zu 10 zentrierten Spalten vorgegeben. Auch die Abstände unterscheiden sich leicht

von den Voreinstellungen der array-Umgebung. Das folgende Beispiel veranschaulicht die Matrixumgebungen `matrix`, `pmatrix`, `bmatrix`, `Bmatrix`, `vmatrix` und `Vmatrix`.¹

Bsp.
8-3-2

$$\begin{array}{cc} 0 & 1 \\ 1 & 0 \end{array} \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \\ \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \left\{ \begin{array}{cc} 1 & 0 \\ 0 & -1 \end{array} \right\} \\ \begin{vmatrix} a & b \\ c & d \end{vmatrix} \left\| \begin{array}{cc} i & 0 \\ 0 & -i \end{array} \right\|$$

```
\usepackage{amsmath}
\begin{gather*}
\begin{matrix}
\begin{matrix} 0 & 1 \\ 1 & 0 \end{matrix} \\
\begin{matrix} 0 & -i \\ 1 & 0 \end{matrix} \\
\begin{matrix} 1 & 0 \\ 0 & -1 \end{matrix} \\
\begin{matrix} a & b \\ c & d \end{matrix} \\
\begin{matrix} i & 0 \\ 0 & -i \end{matrix}
\end{matrix}
\end{gather*}
```

Die maximale Anzahl der Spalten in einer Matrixumgebung wird durch den Zähler `MaxMatrixCols` festgelegt, der mithilfe der Standardzählerbefehle von L^AT_EX verändert werden kann. Wie in der Standard-array-Umgebung wird der Spaltenzwischenraum durch den Wert von `\arraycolsep` bestimmt, jedoch ohne an den Seiten des Arrays Freiraum hinzuzufügen. Mit mehr Spalten hat L^AT_EX auch etwas mehr Arbeit und benötigt geringfügig größere Ressourcen. Für die heutzutage üblichen T_EX-Installationen sind diese Einschränkungen jedoch kaum von Bedeutung, so dass man den Wert ohne merkliche Einbußen in der Verarbeitungsgeschwindigkeit auf 20 oder sogar noch höher einstellen kann.

Bsp.
8-3-3

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

```
\usepackage{amsmath}
\setcounter{MaxMatrixCols}{20}
\left[ \begin{matrix}
a&b&c&d&e&f&g&h&i &\cdots \\
&a&b&c&d&e&f&g&h &\cdots \\
&&a&b&c&d&e&f&g &\cdots \\
&&&a&b&c&d&e&f &\cdots \\
&&&&\ddots &\ddots &\dots &\dots &\dots &\dots
\end{matrix} \right]
```

Dieses Beispiel zeigt auch, wie man den Befehl `\hdotsfor` verwendet, um eine Reihe von Punkten in einer Matrix zu erzeugen, die sich über eine vorgegebene Anzahl von Spalten (hier 4) erstreckt. Der Punktabstand lässt sich über das optionale Argument (hier 2) variieren, das als Multiplikator für den voreingestellten Zwischenraum dient; dieser beträgt 3 mathematische Einheiten (math units, siehe Anhang A.1.5). Der schmale Leerraum und die Klammergruppe `\, \{ }` am Ende jeder Zeile verschönern lediglich das Layout. Gemeinsam erzeugen sie zwei schmale Leerräume einer Breite von ungefähr 6mu bzw. 1/3em. (Abstände in Formeln werden in Abschnitt 8.7.6 auf Seite 521 ausführlicher besprochen.)

Für die Darstellung einer kleinen Matrix innerhalb von Texten ist die `smallmatrix`-Umgebung am besten geeignet. Man beachte, dass sie den Zei-

¹Man beachte die zuvor erfolgten Warnungen über mögliche Probleme im Zusammenhang mit `matrix` und `pmatrix`.

lenabstand nicht vergrößert, obwohl die Zeile vor der kleinen Matrix Wörter mit Unterlängen enthält.

Um den Einfluss einer Matrix auf die sie umgebenden Zeilen in einem Absatz zu zeigen, steht sie hier: $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ und danach folgt genug Text, dass zumindest eine ganze Zeile unterhalb der Matrix verläuft.

```
\usepackage{amsmath}
```

Um den Einfluss einer Matrix auf die sie umgebenden Zeilen in einem Absatz zu zeigen, steht sie hier:

```
$ \left( \begin{smallmatrix}
1 & 0 \\ 0 & -1 \end{smallmatrix} \right) $
```

und danach folgt genug Text, dass zumindest eine ganze Zeile unterhalb der Matrix verläuft.

Bsp.
8-3-4

8.3.3 Mehrzeilige Tief- und Hochstellungen

Der Befehl `\substack` wird meistens dazu benutzt, um innerhalb einer Tief- oder Hochstellung mehrere Zeilen zu erzeugen, wobei `\\` als Zeilenbegrenzung dient.

Die `subarray`-Umgebung bietet eine etwas allgemeinere Anordnungsmöglichkeit, in der man die Zeilen auch links- oder rechtsbündig anordnen kann, anstatt zentriert. Sowohl bei der Umgebung wie auch beim `\substack`-Befehl ist jedoch zu beachten, dass sie bei Hoch- oder Tiefstellung in geschweiften Klammern stehen müssen.

$$\sum_{\substack{0 \leq i \leq m \\ 0 < j < n}} P(i, j)$$

(1)

```
\usepackage{amsmath}
```

```
\begin{gather}
```

```
\sum_{\substack{0 \leq i \leq m \\ 0 < j < n}} P(i, j) \\
\sum_{\begin{subarray}{l} i \in \Lambda \\ 0 \leq i \leq m \\ 0 < j < n \end{subarray}} P(i, j)
```

(2)

```
\end{subarray}} P(i, j)
```

```
\end{gather}
```

Bsp.
8-3-5

8.3.4 Kommutative Diagramme

Einige Befehle zum Erzeugen einfacher kommutativer Diagramme, die auf Arrays basieren, sind in einem separaten Paket namens `amscd` verfügbar. Es bietet einige nützliche Kurzformen für beschriftete Pfeile und andere Konnektoren. Es ist jedoch nicht sehr vielseitig – die Konnektoren stehen z.B. nur in horizontaler und vertikaler Form zur Verfügung.

Zwar ließe sich die `picture`-Umgebung für komplexere kommutative Diagramme verwenden, aber für die meisten ernsthaften Arbeiten auf diesem Gebiet benötigt man doch eines der umfangreicheren Pakete. Dazu gehören das `XY-pic`-System von Kristoffer Rose (siehe [58, Kap. 5]) und seine Erweiterung [11] von Michael Barr, das `diagram`-System [22, 23] von Francis Borceux und das `kuvio`-Paket [156] von Anders Svensson.

In der `CD`-Umgebung erhält man durch die Zeichenfolgen `@>>>`, `@<<<`, `@VVV` und `@AAA` nach rechts, links, unten bzw. oben weisende Pfeile.¹

¹Auf Tastaturen ohne die Zeichen `<` und `>` können alternativ die Notationen `@)))` und `@(((` genutzt werden.

Die nachfolgenden Beispiele zeigen zudem die Verwendung des Befehls `\DeclareMathOperator` (siehe Abschnitt 8.6.2).

Bsp.
8-3-6

$$\begin{array}{ccccc}
 \text{cov}(L) & \longrightarrow & \text{non}(K) & \longrightarrow & \text{cf}(K) \\
 \downarrow & & \uparrow & & \uparrow \\
 \text{add}(L) & \longrightarrow & \text{add}(K) & \longrightarrow & \text{cov}(K)
 \end{array}$$

```

\usepackage{amsmath,amscd}
\DeclareMathOperator\add{add}
\DeclareMathOperator\cf{cf}
\DeclareMathOperator\cov{cov}
\DeclareMathOperator\non{non}
\[\ \begin{CD}
\cov(L) @>>> \non(K) @>>> \cf(K) \\
@VVV @AAA @AAA \\
\add(L) @>>> \add(K) @>>> \cov(K)
\end{CD} \]

```

Die Pfeile werden nach folgendem Verfahren beschriftet: Bei horizontalen Pfeilen wird das Material zwischen dem ersten und zweiten `>`- oder `<`-Symbol als Hochstellung, und jenes zwischen dem zweiten und dritten als Tiefstellung gesetzt. Entsprechend wird das Material zwischen dem ersten und zweiten bzw. zweiten und dritten `A` oder `V` vertikaler Pfeile als links- bzw. rechtsseitige Beschriftung gesetzt; dieses Format wird im nächsten Beispiel benutzt, um den Operator `End P` rechts neben dem Pfeil zu positionieren.

Die Zeichenfolgen `@=` und `@|` erzeugen horizontale bzw. vertikale Doppellinien. Anstelle eines sichtbaren Pfeils lässt sich auch, soweit erforderlich, (mit der Notation `@.`) auch ein „Null-Pfeil“ setzen um ein Array auszufüllen.

Bsp.
8-3-7

$$\begin{array}{ccc}
 S^{W_\Lambda} \otimes T & \xrightarrow{j} & T \\
 \downarrow & & \downarrow_{\text{End } P} \\
 (S \otimes T)/I & \xlongequal{\quad} & (Z \otimes T)/J
 \end{array}$$

```

\usepackage{amsmath,amscd}
\DeclareMathOperator{\End}{End}
\[\ \begin{CD}
S^{\mathcal{W}_\Lambda} \otimes T @>j>> T \\
@VVV @VV{\text{End } P}V \\
(S \otimes T)/I @= (Z \otimes T)/J
\end{CD} \]

```

In Standard- \LaTeX lässt sich ein ähnliches Layout erzeugen, das jedoch nicht annähernd so gut aussieht:

Bsp.
8-3-8

$$\begin{array}{ccc}
 S^{W_\Lambda} \otimes T & \xrightarrow{j} & T \\
 \downarrow & & \downarrow_{\text{End } P} \\
 (S \otimes T)/I & = & (Z \otimes T)/J
 \end{array}$$

```

\[\begin{array}{ccc}
S^{\mathcal{W}_\Lambda} \otimes T & \xrightarrow{j} & T \\
\Big\downarrow & & \Big\downarrow_{\text{End } P} \\
(S \otimes T)/I & = & (Z \otimes T)/J
\end{array}\]

```

Das Beispiel zeigt deutlich, wie viel bessere Ergebnisse das `amscd`-Paket erzeugt: Die Notation ist sehr viel einfacher und das Paket setzt z.B. längere horizontale Pfeile und verteilt die Abstände zwischen den Elementen des Diagramms passender. Mithilfe der spezialisierteren Pakete kann man sogar noch ansprechendere Ergebnisse erzielen.

8.3.5 delarray – Begrenzungszeichen für Arrays

In diesem Abschnitt wird eine nützliche allgemeine Erweiterung für das array-Paket (vgl. Abschnitt 5.2 auf Seite 252) beschrieben, mit deren Hilfe man öffnende und schließende erweiterbare Begrenzungsymbole (vgl. Abschnitt 8.5.3) um eine mathematische array-Umgebung setzen kann. Das Paket delarray wurde von David Carlisle geschrieben und sein Nutzen wird anhand des nächsten, etwas seltsam aussehenden Beispiels veranschaulicht. (Man beachte, dass dieses Paket unabhängig vom amsmath-Paket eingesetzt werden kann, wobei es, falls erforderlich, automatisch das array-Paket lädt.)

$$(X \ Y) \left[\begin{array}{cc} A & B \\ C & D \end{array} \right] \left(\begin{array}{l} L \\ M \end{array} \right)$$

```

\usepackage{delarray}
\[
\begin{array}[t] ( {cc} ) X & Y \end{array}
\begin{array}[t] [ {cc} ] A & B \ \backslash C & D \ \end{array}
\begin{array}[b] \lgroup{cc}\rgroup L \ \backslash M \end{array}
\]

```

Bsp.
8-3-9

Die Begrenzungszeichen werden links und rechts der „Präambeldekларation“ (hier {cc}) platziert. Es muss sich um Begrenzungszeichen handeln, die in Tabelle 8.3 auf Seite 511 aufgeführt sind.

Das vorige Beispiel zeigt auch die nützlichste Funktion dieses Paketes: Es verwendet die optionalen Argumente [t] und [b], die nicht in den Matrixumgebungen des amsmath-Paketes verfügbar sind. Sie zeigen, dass die delarray-Syntax anders wirkt als das Umschließen der array-Umgebung mit \left und \right, wodurch sowohl die Begrenzungszeichen als auch das Array selbst angehoben werden.

8.4 Komplexere Gebilde und Beschriftungen

Dieser Abschnitt erläutert einige Befehle, die eine Vielzahl mittelgroßer mathematischer Gebilde erzeugen, einschließlich beschrifteter Symbole und bruchähnlicher Objekte.

8.4.1 Beschriftete Pfeile

Die Befehle \xleftarrow und \xrightarrow erzeugen horizontale Beziehungspfeile ähnlich denjenigen für kommutative Diagramme in Abschnitt 8.3.4; sie können ober- und/oder unterhalb beschriftet werden, wobei sich die Länge der Pfeile automatisch nach dem Text richtet. Diese Pfeile sind normalerweise nur in einer Größe verfügbar. Dadurch eignen sie sich wahrscheinlich nicht für beispielsweise Brüche, Tief- oder Hochstellungen.

Die Beschriftungen unter- und oberhalb der Pfeile werden über ein optionales und ein obligatorisches Argument des Befehls festgelegt.

$$0 \xleftarrow[\zeta]{F \times \Delta(n-1)} E^{\partial_0 b}$$

```

\usepackage{amsmath}
\[
0 \xleftarrow[\zeta]{ F \ \times \ \Delta (n - 1) }
\xrightarrow{\partial_0 \alpha(b)} E^{\partial_0 b}
\]

```

Bsp.
8-4-1

8.4.2 Kettenbrüche

Der Befehl `\cfrac` erzeugt Reihen von Brüchen, die als „Kettenbrüche“ bekannt sind. Per Voreinstellung sind alle Zähler zentriert; mithilfe der optionalen Argumente `[l]` oder `[r]` lassen sie sich nach links oder rechts ausrichten.

Bsp.
8-4-2

$$\sqrt{2} + \frac{1}{\sqrt{3} + \frac{1}{\sqrt{4} + \frac{1}{\sqrt{5} + \frac{1}{\sqrt{6} + \dots}}}}$$

```
\usepackage{amsmath}
\begin{equation*}
\cfrac {1}{\sqrt{2}} +
\cfrac {1}{\sqrt{3}} +
\cfrac {1}{\sqrt{4}} +
\cfrac[r] {1}{\sqrt{5}} +
\cfrac[l] {1}{\sqrt{6}} + \dotsb \}}\}
\end{equation*}
```

8.4.3 Eingerahmte Formeln

Der Befehl `\boxed` setzt einen Rahmen um sein Argument; er funktioniert wie der Befehl `\fbox`, nur dass sein Inhalt im Formelmodus erscheint (vgl. die in Abschnitt 10.1 beschriebenen Befehle).

Bsp.
8-4-3

$$\boxed{W_t - F \subseteq V(P_i) \subseteq W_t} \quad (1)$$

```
\usepackage{amsmath}
\begin{equation}
\boxed { W_t - F \subseteq V(P_i) \subseteq W_t }
\end{equation}
```

8.4.4 Grenzpositionen

Tief- und Hochstellungen an Integralen, Summen oder anderen Operatoren können entweder direkt über und unter den mathematischen Operatoren positioniert werden, oder an den normalen Stellen für Tief-/Hochstellungen rechts der Operatoren. Man sagt, sie „zeigen Grenzen“, wenn das hoch- bzw. tiefgestellte Material (d.h. „die Grenzen“) an hervorgehobener Stelle über und unter und nicht neben dem jeweiligen Symbol oder Operatornamen platziert wird. Typischerweise werden im laufenden Text „keine Grenzen gezeigt“ (um zu vermeiden, dass die Zeilen auseinander gezogen werden); in abgesetzten Formeln hängt ihre Position vom jeweiligen Operator ab. Das nächste Beispiel veranschaulicht die in \LaTeX voreingestellten Positionen.

Bsp.
8-4-4

$$\sum_{i=1}^n \int_0^\infty \lim_{n \rightarrow 0}$$

Text: $\sum_{i=1}^n, \int_0^\infty, \lim_{n \rightarrow 0}$.

```
\[
\sum_{i=1}^n \quad \int_0^\infty \quad \lim_{n \to 0}
\]
```

Text: $\sum_{i=1}^n$, \int_0^∞ , $\lim_{n \to 0}$.

Die Platzierung von Tief- und Hochstellungen an Integralen, Summen und anderen Operatoren ist häufig eine Frage der jeweiligen Layoutrichtlinien einer Zeitschrift. Um dieser Tatsache gerecht zu werden, bietet das `amsmath`-Paket eine umfangreiche Liste von Optionen für ihre Positionierung. In der folgenden Zusammenfassung bezeichnet *Voreinstellung* das Ver-

halten des `amsmath`-Paketes in Kombination mit einer `LATEX`-Standardklasse, ohne eine der genannten Optionen. In Verbindung mit einer der `AMS-LATEX`-Dokumentenklassen sind unter Umständen andere Voreinstellungen aktiv.

`intlimits, nointlimits` Nur für abgesetzte Formeln; platziert Hoch- und Tiefstellungen an integralartigen Symbolen ober- und unterhalb bzw. seitlich (Voreinstellung).

`sumlimits, nosumlimits` Nur für abgesetzte Formeln; platziert Hoch- und Tiefstellungen an summenartigen Symbolen (auch „große Operatoren“ genannt) ober- und unterhalb (Voreinstellung) bzw. seitlich. Diese Option betrifft auch weitere große Operatoren – \prod , \coprod , \otimes , \oplus usw. – jedoch nicht die verschiedenen Integrale.

`namelimits, nonamelimits` Wie `sumlimits` oder `nosumlimits`, jedoch für bestimmte „Operatornamen“ wie z.B. `det`, `inf`, `lim`, und `max`, `min`, deren Tiefstellungen normalerweise direkt unterhalb erscheinen, zumindest in abgesetzten Formeln.

Die Positionierung an einzelnen Symbolen oder Namen lässt sich auch direkt steuern, und zwar mithilfe eines der folgenden `TEX`-Basisbefehle nach dem Symbol oder Operatornamen: `\limits`, `\nolimits`, oder `\displaylimits`. Der letzte Befehl bestimmt, dass der Operator nur dann „Grenzen zeigt“, wenn ein Formelstil für abgesetzte Formeln verwendet wird. Er ist die Voreinstellung für alle Symbole der Klasse `Operator1` und für `\mathop`-Konstruktionen. Wenn ein Operator außerhalb einer abgesetzten Formel „Grenzen zeigen“ soll, muss dies einzeln mit dem `\limits`-Befehl deklariert werden. Vergleicht man das nächste Beispiel mit Beispiel 8-4-4, so fällt auf, dass einige Befehle keine Wirkung zeigen, da sie lediglich die Voreinstellung bestätigen.

$\sum_{i=1}^n \int_0^{\infty} \lim_{n \rightarrow 0}$	<code>\[</code> <code>\sum\nolimits_{i=1}^n \quad \quad \int\limits_0^{\infty}</code> <code>\quad \quad \quad \lim\displaylimits_{n \to 0}</code> <code>\]</code>	<code>Text: $\sum\nolimits_{i=1}^n$, \int_0^{∞}, $\lim_{n \rightarrow 0}$.</code> <code>$\sum\nolimits_{i=1}^n$, \int_0^{∞}, $\lim_{n \rightarrow 0}$.</code>
---	--	--

Bsp.
8-4-5

8.4.5 Mehrfachintegrale

Die Befehle `\iint`, `\iiint` und `\iiint` erzeugen Mehrfachintegrale mit ausgleichenen Zwischenräumen, sowohl im normalen Text als auch in abgesetzten Formeln. Der Befehl `\idotsint` erzeugt zwei Integralzeichen, die durch Punkte getrennt sind. Das folgende Beispiel zeigt außerdem, wie der Befehl `\limits` die Voreinstellung für die Integralkonstruktionen umgeht und die Grenze V unterhalb des Symbols platziert.

¹In Abschnitt 8.9.1 auf Seite 540 werden die verschiedenen Klassen mathematischer Symbole besprochen.

Bsp.
8-4-6

$$\int_V \mu(v, w) \, du \, dv$$

$$\iiint_V \mu(u, v, w) \, du \, dv \, dw$$

$$\iiiiiint_V \mu(t, u, v, w) \, dt \, du \, dv \, dw$$

$$\int_V \cdots \int \mu(z_1, \dots, z_k) \, \mathbf{dz}$$

```
\usepackage{amsmath}
\begin{gather*}
\iint \limits_V \mu(v, w)
\ ,du \ ,dv \\
\iiint \limits_V \mu(u, v, w)
\ ,du \ ,dv \ ,dw \\
\iiiiiint \limits_V \mu(t, u, v, w)
\ ,dt \ ,du \ ,dv \ ,dw \\
\idotsint \limits_V \mu(z_1, \dots, z_k) \ ,\mathbf{dz}
\end{gather*}
```

8.4.6 Modulo-Operationen

Die Befehle `\mod`, `\bmod`, `\pmod` und `\pod` dienen im `amsmath`-Paket dazu, spezielle Abstandskonventionen der „mod“-Notation für Äquivalenzklassen ganzer Zahlen zu realisieren. Zwei dieser Befehle, `\mod` und `\pod`, sind Varianten von `\pmod`, die von einigen Autoren bevorzugt werden. Der Befehl `\mod` setzt keine Klammern, während `\pod` die Klammern erzeugt, dafür aber den Text „mod“ unterdrückt. Durch `amsmath` werden die Abstände von `\pmod` in eingebetteten (nicht abgesetzten) Formeln verkleinert.

Bsp.
8-4-7

$$u \equiv v + 1 \pmod{n^2}$$

$$u \equiv v + 1 \bmod n^2$$

$$u = v + 1 \pmod{n^2}$$

$$u = v + 1 \pod{n^2}$$

```
\usepackage{amsmath}
\begin{align*}
u &\equiv v + 1 \pmod{n^2} \\
u &\equiv v + 1 \bmod n^2 \\
u &= v + 1 \pmod{n^2} \\
u &= v + 1 \pod{n^2}
\end{align*}
Eingebettetes Layout: $ u = v + 1 \pmod{n^2} $
\begin{gather*}
(m \bmod n) = k^2 \ , \ ; \ \quad
x \equiv y \pmod b \ , \ ; \ \quad \\
x \equiv y \pmod c \ , \ ; \ \quad
x \equiv y \pod d \ , \ .
\end{gather*}
```

8.4.7 Brüche und ähnliche Gebilde

Zusätzlich zu dem gängigen Befehl `\frac` stellt das `amsmath`-Paket die Befehle `\dffrac` und `\tfrac` als praktische Abkürzungen für `{\displaystyle\frac ...}` und `{\textstyle\frac ...}` zur Verfügung. (Die unterschiedlichen mathematischen Formelstile werden ausführlicher in Abschnitt 8.7.1 auf Seite 515 besprochen.)

Bsp.
8-4-8

$$\frac{1}{k} \log_2 c(f) \quad \frac{1}{k} \log_2 c(f) \quad (1)$$

$$\sqrt{\frac{1}{k} \log_2 c(f)} \quad \sqrt{\frac{1}{k} \log_2 c(f)}$$

```
\usepackage{amsmath}
\begin{equation}
\frac{1}{k} \log_2 c(f) \quad \frac{1}{k} \log_2 c(f)
\quad \tfrac{1}{k} \log_2 c(f)
\end{equation}
Text: $ \sqrt{\frac{1}{k} \log_2 c(f)} \quad \sqrt{\frac{1}{k} \log_2 c(f)} $, $.
```

Für Binomialkoeffizienten wie $\binom{n}{k}$ kann man die entsprechenden Befehle `\binom`, `\dbinom` und `\tbinom` verwenden.

$$\binom{k}{2}2^{k-1} + \binom{k-1}{2}2^{k-2} \quad (1)$$

Text: $\binom{k}{2}2^{k-1} + \binom{k-1}{2}2^{k-2}$.

```

\usepackage{amsmath}
\begin{equation}
\binom{k}{2} 2^{k-1}
+ \tbinom{k-1}{2} 2^{k-2}
\end{equation}
Text: $ \binom{k}{2} 2^{k-1}
+ \dbinom{k-1}{2} 2^{k-2} $.

```

Bsp.
8-4-9

$\frac{\textit{lbegrenz}}{\textit{rbegrenz}}^{\textit{stärke}}_{\textit{stil}}\left\{\frac{\textit{zähler}}{\textit{nenner}}\right\}$

Alle diese `\binom`- und `\frac`-Befehle sind Sonderformen des allgemeineren Befehls für Brüche, `\genfrac`, der über sechs Argumente verfügt: Bei den ersten beiden Argumenten *lbegrenz* und *rbegrenz* handelt es sich um den linken bzw. rechten Begrenzer. Sie müssen entweder beide leer oder beide gefüllt sein. Will man nur ein Begrenzungszeichen setzen, muss man auf der „leeren“ Seite einen Punkt „.“ einfügen. Das dritte Argument *stärke* überschreibt die voreingestellte Linienstärke des Bruchstrichs. Der Befehl `\binom` z.B. verwendet hier eine Stärke von 0pt für eine unsichtbare Linie. Bleibt dieses Argument leer, so hat die Linienstärke den Wert, der in den Schrifteinstellungen für den Mathematiksatz vorgegeben ist. Die Beispiele in diesem Kapitel verwenden die in Tabelle 8.2 auf der nächsten Seite aufgeführten Voreinstellungen für die verschiedenen Stile (vgl. auch Abschnitt 8.7.1).

Mithilfe des vierten Argumentes *stil* kann man den automatisch gewählten Formelstil, der auch die Schriftgrößen bestimmt, außer Kraft setzen. Er kann als Wert eine ganze Zahl von 0-3 annehmen, die für `\displaystyle`, `\textstyle`, `\scriptstyle` bzw. `\scriptscriptstyle` steht. Wenn dieses Argument leer bleibt, wird der Stil nach den normalen Regeln für Brüche ausgewählt (wie in Tabelle 8.5 auf Seite 515 beschrieben). Die letzten beiden Argumente sind einfach der Zähler (*zähler*) und der Nenner (*nenner*). Zur Veranschaulichung hier mögliche Definitionen für `\frac`, `\tfrac` und `\binom`:

```

\newcommand\frac [2]{\genfrac {}{}{}{#1}{#2}}
\newcommand\tfrac [2]{\genfrac {}{}{}{#1}{#2}}
\newcommand\binom [2]{\genfrac {}{}{0pt}{}{#1}{#2}}

```

Wenn man natürlich eine besonders komplexe Notation (wie z.B. eine mit `\genfrac` realisierte) in seinem Dokument wiederholt verwenden möchte, tut man sich selbst (und dem Lektor) einen Gefallen, wenn man mithilfe des Befehls `\newcommand`, wie im vorigen Beispiel, einen sprechenden Namen als Abkürzung für diese Notation definiert.

Die ursprünglichen \TeX -Befehle für Brüche, Binomialkoeffizienten und ähnliche Gebilde `\over`, `\overwithdelims`, `\atop`, `\atopwithdelims`, `\above` und `\abovewithdelims` (die Standard- \LaTeX von \TeX geerbt hat), erzeugen in Kombination mit dem `amsmath`-Paket Fehlermeldungen.

Formelstil	Voreingestellte Stärke (ungefähr)
text/display	0.40pt
script	0.34pt
scriptscript	0.24pt

Tabelle 8.2: Voreingestellte Linienstärke der verschiedenen Formelstile

8.4.8 Punkte als Akzente

Die Befehle `\dot` und `\ddot` für mathematische Akzente werden ergänzt durch `\dotted` und `\ddddot`, die dreifache bzw. vierfache Punkte als Akzente erzeugen.

Bsp.
8-4-10

```

\usepackage{amsmath}
 $\dot{S} \quad \ddot{P} \quad \dotted{Q} \quad \ddddot{R}$ 

```

Wenn man eigene mathematische Akzente benutzen möchte, sollte man vielleicht das `accents`-Paket von Javier Bezos verwenden. Es hält, neben weiteren Funktionen, Methoden zum Definieren von Mehrfachakzenten (siehe `\accentset` im Beispiel) und von unterhalb stehenden Akzenten (`\underaccent`, `\undertilde`) bereit. Es lässt sich mit dem `amsmath`-Paket kombinieren. Weitere Details sind in [20] zu finden.

Bsp.
8-4-11

```

\usepackage{accents}
\[ \accentset{\ast}{\hat{X}} \quad \hat{\accentset{\ast}{h}} \quad \mathcal{M} \quad \underaccent{\diamond}{C} \quad \underaccent{\sim}{M} \quad \underaccent{\sim}{ABC}
\]

```

8.4.9 amsxtra – Akzente als hochgestellte Zeichen

Neben einigen weniger wichtigen Funktionalitäten stellt dieses Paketes eine Sammlung einfacher Befehle bereit, mit denen Akzente als hochgestellte Zeichen an einer Unterformel platziert werden können:

Bsp.
8-4-12

```

\usepackage{amsxtra}
 $(xyz)^{\cdots} \quad (xyz)^{\ddot{\cdot}} \quad (xyz)^{\cdot}$ 
 $(xyz)^{\spdddot} \quad (xyz)^{\spddot} \quad (xyz)^{\spdot}$ 
 $(xyz)^{\spbreve} \quad (xyz)^{\spcheck} \quad (xyz)^{\sphat} \quad (xyz)^{\sptilde}$ 

```

8.4.10 Zusätzliche Beschriftungen

Standard- \LaTeX verfügt über den Befehl `\stackrel`, mit dem ein hochgestelltes Zeichen über einem Relationssymbol platziert werden kann. Das `amsmath`-Paket stellt zusätzlich die Befehle `\overset` und `\underset` bereit. Mit ihrer Hilfe lässt sich Material nicht nur über oder unter Relationssymbolen, sondern auch über oder unter beliebigen regulären Symbolen (Klasse `Ordinary`) oder binären Operatoren platzieren. Sie werden genauso wie die Grenzen über oder unter einem Summenzeichen gesetzt.

Der Befehl `\sideset` ergänzt die anderen durch seine besondere Funktion: Er fügt Operatorsymbolen wie \sum oder \prod zusätzlich zu den „normalen“ Grenzen (die ober- und unterhalb stehen) noch weitere Beschriftungen hinzu. Sie erscheinen an den Positionen für Tief- und Hochstellungen, sowohl links als auch rechts seitlich der Operatoren.

$$\overset{*}{X} > \underset{*}{X} \iff \sum'_{a,b \in \mathbf{R}^*} \overset{a}{\underset{b}{X}} = X$$

```
\usepackage{amsmath}
\[ \overset{*}{X} > \underset{*}{X}
\iff \sideset{*}{'}\sum_{a,b \in \mathbf{R}^*}
\overset{a}{\underset{b}{X}} = X \]
```

Bsp.
8-4-13

Dieses recht komplexe Beispiel zeigt, wie man ein Produktzeichen rundum beschriften kann.

$$\prod_{\substack{i=1 \\ k>1}}^n \prod_{j=2}^m \mathcal{T}_{i,j}^k$$

```
\usepackage{amsmath}
\[ \sideset_{i=1}^n_{j=2} \prod_{k>1}
\mathcal{T}_{i,j}^k \]
```

Bsp.
8-4-14

8.5 Symbole mit variablen Formen

Bei den meisten \LaTeX -Befehlen für den Formelsatz erwartet man, dass sie ein fest vorgegebenes Symbol produzieren. Tatsächlich gibt es aber eine ganze Reihe von Fällen, in denen die genaue Form gar nicht festgelegt ist (selbst wenn Zeichensatz und Schriftgrad feststehen). Bestimmte \TeX -Funktionen für den Mathematiksatz können sogar Gebilde erzeugen, die im Prinzip je nach Bedarf beliebig groß werden können.

Eine solche kontextabhängige Anpassungsfähigkeit ist für den Mathematiksatz sehr wichtig und dieser Abschnitt behandelt einige ihrer Aspekte. Soweit nicht anders erwähnt, sind alle in diesem Abschnitt besprochenen Befehle in Standard- \LaTeX verfügbar.

Ein bekanntes, aber nicht sehr aufregendes Beispiel für diese Anpassungsfähigkeit sind die Zeichen für mathematische Operatoren, wie z.B. `\sum` und `\prod`, die normalerweise aber nur in zwei Größen vorliegen: einer kleineren für Fließtexte und einer größeren für abgesetzte Formeln. Diese Symbole sind in Tabelle 8.25 auf Seite 552 aufgeführt.

8.5.1 Auslassungspunkte ...

Standard- \LaTeX verfügt über eine Reihe verschiedener Auslassungspunkte für den Mathematiksatz: `\ldots`, `\cdots` usw. Wenn das `amsmath`-Paket geladen ist, sollten die Auslassungspunkte im Formelmodus jedoch fast immer einfach mit `\dots` gesetzt werden.¹

Die vertikale Position der Auslassungspunkte (auf der Grundlinie oder mittig) sowie der umgebende Leerraum werden automatisch gewählt, passend zu dem Symbol, das auf den Befehl `\dots` folgt. Ist das nächste Symbol z.B. ein Pluszeichen, so werden die Punkte dazu zentriert, ist es ein Komma, erscheinen sie auf der Grundlinie. Sie bestehen immer aus drei Punkten, jedoch

¹Die Befehle `\dots` und `\ldots` können auch im Textmodus genutzt werden, wo beide immer einfache Auslassungspunkte für Text erzeugen.

mit unterschiedlichem Zwischenraum. Diese Voreinstellungen des `amsmath`-Paketes können in einer Klassendatei geändert werden, sofern andere Konventionen gelten.

Bsp.
8-5-1

Eine Reihe H_1, H_2, \dots, H_n , eine Summe $H_1 + H_2 + \dots + H_n$, ein orthogonales Produkt $H_1 \times H_2 \times \dots \times H_n$.

`\usepackage{amsmath}`

Eine Reihe H_1, H_2, \dots, H_n , eine Summe $H_1 + H_2 + \dots + H_n$, ein orthogonales Produkt $H_1 \times H_2 \times \dots \times H_n$.

Wenn die Punkte am Ende einer Formel stehen, gibt das nächste Objekt (also `\end`, `\`) oder `$`) keinen Aufschluss darüber, wie die Punkte positioniert werden sollen. In diesem Fall muss man deshalb die Position mit einem der folgenden Befehle selbst festlegen: `\dotsc` für „Punkte bei Kommas“, `\dotsb` für „Punkte bei binären Operatoren“, `\dotsm` für „Multiplikationspunkte“, `\dotsi` für „Punkte bei Integralen“ oder sogar `\dotso` für „keinen der genannten Fälle“. Diese Befehle sollten allerdings wirklich nur an diesen besonderen Stellen eingesetzt werden. In allen anderen Fällen ist der Befehl `\dots` vorzuziehen.

Im nächsten Beispiel werden die Auslassungspunkte im ersten Fall auf der Grundlinie und an den anderen Stellen vertikal zentriert gesetzt, mit jeweils passenden Abständen um die Punkte.

Bsp.
8-5-2

Eine Reihe H_1, H_2, \dots , eine Summe $H_1 + H_2 + \dots$, ein orthogonales Produkt $H_1 \times H_2 \times \dots$, und ein uneigentliches Integral:

`\usepackage{amsmath}`

Eine Reihe H_1, H_2, \dots , eine Summe $H_1 + H_2 + \dots$, ein orthogonales Produkt $H_1 \times H_2 \times \dots$, und ein uneigentliches Integral:
`\[\int_{H_1} \int_{H_2} \dots -\Gamma d\Theta \]`

$$\int_{H_1} \int_{H_2} \dots -\Gamma d\Theta$$

Die Symbole und Abstände, die in den verschiedenen Situationen durch den `\dots`-Befehl erzeugt werden, lassen sich durch Umdefinieren der Befehle `\dotsc`, `\dotsb`, `\dotsm` und `\dotsi` anpassen. Dies geschieht normalerweise in einer Klassendatei. Man könnte z.B. entscheiden, in manchen Fällen nur zwei Punkte zu verwenden.

8.5.2 Horizontale Erweiterungen

Im Prinzip lässt sich jedes Akzentzeichen für den Mathematiksatz so einrichten, dass es die passende Glyphen aus dem Angebot an verschiedenen Breiten erzeugt, sofern diese in den verfügbaren Fonts enthalten sind. Standard- \LaTeX kennt jedoch nur zwei derartige Befehle: `\widehat` und `\widetilde`.

In diesem Abschnitt werden einige Befehle beschrieben, die Konstruktionen erzeugen, welche diesen erweiterbaren Akzenten ähneln. Sie erzeugen alle zusammengesetzte Zeichen der mathematischen Klasse `Ordinary` (reguläre Zeichen, vgl. Abschnitt 8.9.1 auf Seite 540) und sind im nächsten Beispiel dargestellt.

$$\widehat{\psi_\delta(t)E_t h} = \widetilde{\psi_\delta(t)E_t h}$$

$$\overline{\psi_\delta(t)E_t h} = \underline{\psi_\delta(t)E_t h}$$

Ändern sich nicht bei Stilwechsel:

$$\overbrace{\psi_\delta(t)E_t h} = \underbrace{\psi_\delta(t)E_t h}$$

Benötigen `amsmath` für Stilwechsel:

$$\overrightarrow{\psi_\delta(t)E_t h} = \overleftarrow{\psi_\delta(t)E_t h}$$

Benötigen `amsmath`:

$$\overrightarrow{\psi_\delta(t)E_t h} = \overleftarrow{\psi_\delta(t)E_t h}$$

$$\overleftarrow{\psi_\delta(t)E_t h} = \overrightarrow{\psi_\delta(t)E_t h}$$

```
\usepackage{amsmath}
\begin{align*}
\widehat{\psi_\delta(t) E_t h}
&= \widetilde{\psi_\delta(t) E_t h} \quad \backslash\backslash
\overline{\psi_\delta(t) E_t h}
&= \underline{\psi_\delta(t) E_t h} \quad \backslash\backslash
\intertext{Ändern sich nicht bei Stilwechsel:}
\overbrace{\psi_\delta(t) E_t h}
&= \underbrace{\psi_\delta(t) E_t h} \quad \backslash\backslash
\intertext{Benötigen \textsf{amsmath} für
Stilwechsel:}
\overrightarrow{\psi_\delta(t) E_t h}
&= \overleftarrow{\psi_\delta(t) E_t h} \quad \backslash\backslash
\intertext{Benötigen \textsf{amsmath}:}
\underrightarrow{\psi_\delta(t) E_t h}
&= \underleftarrow{\psi_\delta(t) E_t h} \quad \backslash\backslash
\overleftrightharrow{\psi_\delta(t) E_t h}
&= \underleftrightharrow{\psi_\delta(t) E_t h}
\end{align*}
```

Bsp.
8-5-3

Die Einzelheiten bezüglich Verfügbarkeit und Eigenschaften dieser Befehle sind leider recht komplex. Sie sind jedoch im Beispiel zusammengefasst. „Stilwechsel“ bedeutet hier, dass die benutzten Zeichen sich mit dem verwendeten Formelstil ändern, so dass sie z.B. in Brüchen oder Tief- bzw. Hochstellungen korrekt erscheinen (siehe Abschnitt 8.7.1 auf Seite 515). Diejenigen, die den Stil nicht verändern, eignen sich nur für die oberste Ebene abgesetzter Formeln.

Ein weiteres horizontal erweiterbares Element in \LaTeX ist der Balken des Wurzelzeichens; diese Funktion wird am Ende des nächsten Unterabschnitts beschrieben.

8.5.3 Vertikale Erweiterungen

Es gibt viel mehr Möglichkeiten für vertikale Erweiterungen. Alle in Tabelle 8.26 auf Seite 553 abgebildeten Zeichen, sowie noch einige andere, sind potentiell erweiterbar. Die vollständige Liste ist in Tabelle 8.3 dargestellt. Diese Zeichen werden nur in bestimmten Zusammenhängen erweiterbar. Dazu müssen sie alle auf einer Konstruktion der folgenden Form basieren:¹

$$\langle \text{left } \langle \text{ext-open} \rangle \langle \text{Unterformel} \rangle \langle \text{right } \langle \text{ext-close} \rangle$$

Hinter $\langle \text{ext-open} \rangle$ und $\langle \text{ext-close} \rangle$ kann sich jedes Symbol aus Tabelle 8.3 verbergen (außer $\backslash \text{sqrt sign}$), sowie weitere Zeichen, wenn zusätzliche Pakete geladen sind. Sie müssen dazu als erweiterbare Symbole eingerichtet sein. Die Methoden hierzu sind in $\LaTeX 2_\epsilon$ *font selection* [109] beschrieben, das zu jeder \LaTeX -Distribution gehört. Es muss also ein Symbol verfügbar sein,

¹Wenn \LaTeX das Programm eTeX verwendet, lassen sich diese erweiterbaren Symbole auch mit $\backslash \text{middle}$ kombinieren.

$\left(\right)$	$\left(\right)$	$\left\{ \right\}$	<code>\{ \}</code>	$\left \right \left \right $	<code>\lVert \rVert</code>
$\langle \rangle$	<code>\langle \rangle</code>	$\left\{ \right\}$	<code>\lbrace \rbrace</code>	$\left \right $	<code>\lvert \rvert</code>
$\left(\right)$	<code>\lgroup \rgroup</code>	$\left[\right]$	<code>[]</code>	$ $	
$\int \int$	<code>\lmoustache \rmoustache</code>	$\left[\right]$	<code>\lbrack \rbrack</code>	$ $	<code>\vert</code>
\Downarrow	<code>\Downarrow</code>	$\left\lceil \right\rceil$	<code>\lceil \rceil</code>	$ $	<code>\arrowvert</code>
\Uparrow	<code>\Uparrow</code>	$\left\lfloor \right\rfloor$	<code>\lfloor \rfloor</code>	$ $	<code>\bracevert</code>
\Updownarrow	<code>\Updownarrow</code>	$\left[\right]$	<code>\llbracket \rrbracket</code> ^(STM)	$ $	<code>\Arrowvert</code>
\downarrow	<code>\downarrow</code>	$/$	<code>/</code>	$ $	<code>\l</code>
\uparrow	<code>\uparrow</code>	\backslash	<code>\backslash</code>	$ $	<code>\Vert</code>
\updownarrow	<code>\updownarrow</code>	.	.	$\sqrt{\quad}$	<code>\sqrtsign</code>

Blau gedruckte Zeichen benötigen entweder das `amsmath`-Paket oder, bei zusätzlichem ^(STM)-Zeichen, das `stmaryd`-Paket.

Ein Punkt (.) selbst ist kein erweiterbares Zeichen, aber er kann verwendet werden, um einen „unsichtbaren“ Begrenzer zu erzeugen.

Das `\sqrtsign`-Zeichen lässt sich nicht mit `\left`, `\right` oder `\middle` kombinieren.

Synonyme: `[\lbrack, [] \rbrack,]` `{ \lbrace, \{ } \rbrace, \}` `| \vert, |` `|| \Vert, \l`

Tabelle 8.3: Vertikal erweiterbare Symbole

das für die Abwesenheit der tatsächlichen Glyphen steht. Als Zeichen für dieses Symbol, das auch hin und wieder als *Null Delimiter* bezeichnet wird, wurde der Punkt (.) ausgewählt. Die Größe der tatsächlichen Glyphen, mit denen die erweiterbaren Symbole gesetzt werden, orientiert sich an der vertikalen Größe (Höhe und Tiefe) der gesetzten *Unterformel*, die zwischen ihnen liegt. Wie das genau vor sich geht und welche Parameter beteiligt sind, wird in Kapitel 17 und Anhang G (Regel 19) des Buches *The TeXbook* [87] erläutert. Man kann auch bestimmte Größen für diese Symbole festlegen (siehe Abschnitt 8.7.3 auf Seite 517).

Das Wurzelzeichen `\sqrtsign` ist noch verblüffender: Es wächst sowohl vertikal als auch horizontal um sich an die Größe seines Argumentes anzupassen. In \LaTeX wird es normalerweise über den Befehl `\sqrt` aufgerufen, der in

Abschnitt 8.7.4 auf Seite 517 näher erläutert wird.

$$\sqrt{1 + \sqrt{1 + \sqrt{1 + \sqrt{1 + \sqrt{1 + \sqrt{1 + x}}}}}}}$$

```
\[
\sqrtsign{1 + \sqrtsign{1 +
\sqrtsign{1 + \sqrtsign{1 +
\sqrtsign{1 + \sqrtsign{1 + x}}}}}}
\]
```

Bsp.
8-5-4

8.6 Text in Formeln

8.6.1 Der `\text`-Befehl

Fontwechselbefehle für Formeln, wie z.B. `\mathrm`, sind nicht dazu gedacht, Text in Formeln zu setzen. Selbst für einzelne Wörter verwendet man hierzu meistens am besten den Befehl `\text`, der dem L^AT_EX-Befehl `\mbox` gleicht, ihm jedoch insofern überlegen ist, dass er sicherstellt, dass der Text im richtigen Schriftgrad gesetzt wird. Als Schrift verwendet er den Font, der außerhalb der Formel aktuell im Gebrauch ist.

Falls $\Delta_{\text{max rauf}} = \Delta_{\text{min runter}}$
(für alle rauf und runter) dann
 $\Delta_{\text{Summe rauf}} = \Delta_{\text{Summe runter}} \quad (1)$

```
\usepackage{amsmath}
\begin{gather}
\text{Falls } \Delta_{\text{max rauf}}
= \Delta_{\text{min runter}} \quad \text{notag} \\
\text{(für alle rauf und runter) dann} \quad \text{notag} \\
\Delta_{\text{Summe rauf}}
= \Delta_{\text{Summe runter}} \\
\end{gather}
```

Bsp.
8-6-1

8.6.2 Operator- und Funktionsnamen

Die Namen allgemein bekannter mathematischer Funktionen (wie z.B. `log` und `sin`) und Operatoren (wie `max` und `lim`) werden traditionell als Wörter (oder Abkürzungen) in einer aufrechten Serifenschrift formatiert, um sie optisch von den kürzeren Variablennamen zu unterscheiden, die in „Formel-Kursivschrift“ gesetzt werden. Für die häufigsten Funktionsnamen gibt es vordefinierte Befehle, welche die korrekte typographische Umsetzung erzeugen; vgl. Tabelle 8.4 auf der nächsten Seite. Die meisten Funktionen sind bereits in Standard-L^AT_EX verfügbar; die in der Tabelle blau dargestellten benötigen das `amsmath`-Paket. Die mit (ℓ) markierten Funktionen können in abgesetzten Formeln „Grenzen zeigen“ (siehe Abschnitt 8.4.4).

$$\lim_{x \rightarrow 0} \frac{\sin^2(x)}{x^2} = 1$$

$$\lim_{n \rightarrow \infty} |a_{n+1}|/|a_n| = 0$$

$$\lim_{\rightarrow} (m_i^\lambda \cdot M)^* \leq \lim_{A/p \rightarrow \lambda(A)} A_p \leq 0$$

```
\usepackage[fleqn]{amsmath} \setlength\mathindent{0pt}
\newcommand\abs[1]{\lvert#1\rvert}
\begin{gather*}
\lim_{x \rightarrow 0} \frac{\sin^2(x)}{x^2} = 1 \\
\varliminf_{n \rightarrow \infty} |a_{n+1}| / |a_n| = 0 \\
\varinjlim (m_i^\lambda \cdot M)^* \leq \varprojlim_{A/p \rightarrow \lambda(A)} A_p \leq 0
\end{gather*}
```

Bsp.
8-6-2