Tipps zu LaTeX

Das LaTeX 2ε-Sündenregister
oder
Veraltete Befehle, Pakete und andere Fehler

Mark Trettin*

Version 1.8 vom 19. Dezember 2004

Zusammenfassung

Angeregt durch eine Diskussion in der deutschsprachigen TeX-Newsgroup[1] über das wiederholte Auftauchen von veralteten und „schlechten“ Paketen und Befehlen, habe ich mich entschlossen, diese kleine Übersicht zu schreiben.

Ich versuche in diesem Artikel die gängigsten Fehler zu zeigen und Alternativen anzubieten. Diese Übersicht soll weder Einführungen wie l2kurz [11] noch die De-TeX-FAQ [7] ersetzen, sondern lediglich einen kleinen Überblick bieten.

Für Vorschläge, Verbesserungen und Kommentare bin ich dankbar. Ach ja, bevor Anfragen kommen: Ja, ich habe Times/Helvetica[2]/Courier benutzt, allerdings nur um die Datei möglichst klein zu halten. ;-)

Copyright © 2003, 2004 by Mark Trettin.

This material may be distributed only subject to the terms and conditions set forth in the *Open Publication License*, v1.0 or later (the latest version is presently available at `http://www.opencontent.org/openpub/`).

Ich bedanke mich bei Ralf Angeli, Christoph Bier, Christian Faulhammer, Jürgen Fenn[3], Ulrike Fischer, Yvon Henel[4], Yvonne Hoffmüller, David Kastrup, Markus Kohm, Thomas Lotze, Frank Mittelbach, Heiko Oberdiek, Walter Schmidt, Stefan Stoll, Knut Wenzig, Emanuele Zannarini[5] und Reinhard Zierke für Tipps, Anmerkungen und Korrekturen. Falls ich jemanden vergessen haben sollte, bitte ich um eine Mail.

*E-Mail: `Mark.Trettin@gmx.de`

[1] `de.comp.text.tex`

[2] Arial in der Darstellung des Acrobat Reader

[3] Englische Übersetzung: `CTAN:info/l2tabu/english/l2tabuen.pdf`

[4] Französische Übersetzung: `CTAN:info/l2tabu/french/l2tabufr.pdf`

[5] Italienische Übersetzung: `CTAN:info/l2tabu/italian/l2tabuit.pdf`

Inhaltsverzeichnis

1 „Todsünden“ 3
- 1.1 *a4.sty*, *a4wide.sty* . . . 3
- 1.2 Layoutänderungen . . . 3
- 1.3 Änderungen von Paketen und Klassen . . . 3
- 1.4 Änderung des Zeilenabstandes mittels `\baselinestretch` . . . 3
- 1.5 Absatzeinzug und -abstand (`\parindent`, `\parskip`) . . . 4
- 1.6 Abgesetzte Formeln mit `$$...$$` . . . 5
- 1.7 `\def` vs. `\newcommand` . . . 5
- 1.8 Verwendung von `\sloppy` . . . 5

2 Veraltete Befehle, Klassen und Pakete 7
- 2.1 Befehle . . . 7
  - 2.1.1 Änderung des Schriftstils . . . 7
  - 2.1.2 Mathematische Brüche (`\over` vs. `\frac`) . . . 8
  - 2.1.3 Zentrierung mit `\centerline` . . . 8
- 2.2 Klassen und Pakete . . . 8
  - 2.2.1 *scrlettr.cls* vs. *scrlttr2.cls* . . . 8
  - 2.2.2 *epsf.sty*, *psfig.sty*, *epsfig.sty* vs. *graphics.sty*, *graphicx.sty* . . . 9
  - 2.2.3 *doublespace.sty* vs. *setspace.sty* . . . 9
  - 2.2.4 *fancyheadings.sty*, *scrpage.sty* vs. *fancyhdr.sty*, *scrpage2.sty* . . . 9
  - 2.2.5 Die *caption.sty*-Familie . . . 9
  - 2.2.6 *isolatin.sty*, *umlaut.sty* vs. *inputenc.sty* . . . 9
  - 2.2.7 *t1enc.sty* vs. *fontenc.sty* . . . 11
  - 2.2.8 *natdin.bst* vs. *dinat.bst* . . . 11
- 2.3 Schriften . . . 11
  - 2.3.1 *times.sty* . . . 11
  - 2.3.2 *mathptm.sty* . . . 12
  - 2.3.3 *pslatex.sty* . . . 12
  - 2.3.4 *palatino.sty* . . . 12
  - 2.3.5 *mathpple.sty* . . . 12
  - 2.3.6 Aufrechte griechische Buchstaben . . . 13
  - 2.3.7 *euler.sty* vs. *eulervm.sty* . . . 13

3 Verschiedenes 14
- 3.1 Gleitumgebungen – „figure“, „table“ . . . 14
- 3.2 Der Anhang . . . 14
- 3.3 Mathematiksatz . . . 14
- 3.4 Die Verwendung von `\graphicspath` . . . 15
- 3.5 Die `\*name`-Makros . . . 16

A Beispiel zu `\sloppy` 17

Literatur

[1] AMERICAN MATHEMATICAL SOCIETY: *User's Guide for the* `amsmath` *Package*. Dezember 1999, Version 2.0.
URL: `CTAN:macros/latex/required/amslatex/`.

[2] DAVID P. CARLISLE: *Packages in the 'graphics' bundle*. Januar 1999.
URL: `CTAN:macros/latex/required/graphics/`.

[3] MARKUS KOHM, FRANK NEUKAM und AXEL KIELHORN: *Das KOMA-Script-Paket*. September 2004, Version 2.9t.
URL: `CTAN:macros/latex/supported/koma-script/`.

[4] THE LaTeX3 PROJECT TEAM: *LaTeX 2ε for class and package writers*. März 1999.
URL: `CTAN:macros/latex/doc/clsguide.pdf`

[5] THE LaTeX3 PROJECT TEAM: *LaTeX 2ε font selection*. September 2000.
URL: `CTAN:macros/latex/doc/fntguide.pdf`

[6] THE LaTeX3 PROJECT TEAM: *LaTeX 2ε for authors*. Juli 2001.
URL: `CTAN:macros/latex/doc/usrguide.pdf`

[7] BERND RAICHLE, ROLF NIEPRASCHK und THOMAS HAFNER: *Fragen und Antworten (FAQ) über das Textsatzsystem TeX und DANTE, Deutschsprachige Anwendervereinigung TeX e.V.* WWW, September 2003, Version 72.
URL: `http://www.dante.de/faq/de-tex-faq/`.

[8] WALTER SCHMIDT: *The Euler Virtual Math Fonts for use with LaTeX*. Januar 2004, Version 3.0a.
URL: `CTAN:fonts/eulervm/`

[9] WALTER SCHMIDT: *The upgreek package for LaTeX 2ε*. Mai 2001, Version 1.0.
URL: `CTAN:macros/latex/contrib/supported/was/`.

[10] WALTER SCHMIDT: *Using common PostScript fonts with LaTeX*. April 2002, PSNFSS Version 9.0.
URL: `CTAN:macros/latex/required/psnfss/psnfss2e.pdf`

[11] WALTER SCHMIDT, JÖRG KNAPPEN, HUBERT PARTL und IRENE HYNA: *LaTeX 2ε-Kurzbeschreibung*. April 1999, Version 2.1.
URL: `CTAN:info/lshort/german/`.

[12] AXEL SOMMERFELD: *Setzen von Abbildungs- und Tabellenbeschriftungen mit dem caption-Paket*. Juli 2004, Version 3.0c.
URL: `CTAN:macros/latex/contrib/caption/`.

1 „Todsünden“

In diesem Abschnitt habe ich die wohl schlimmsten Fehler zusammengetragen, die in schöner Regelmäßigkeit in `de.comp.text.tex` auftauchen und den dortigen Regulars entweder die Zornesröte ins Gesicht oder die Tränen in die Augen treiben. `;-)`

1.1 *a4.sty*, *a4wide.sty*

Diese „beiden“ Pakete sollten nicht mehr verwendet und ersatzlos aus dem LaTeX-Quelltext gestrichen und durch die Klassenoption a4paper ersetzt werden. Abgesehen davon, dass das Layout der meisten dieser Pakete typografisch mehr als fragwürdig ist, existieren mehrere verschiedene, zu einander inkompatible Versionen. Man kann sich also nicht einmal sicher sein, dass auf einem anderen Rechner das Dokument gleich (schlecht?) aussieht.

1.2 Layoutänderungen

Die von den Standardklassen (*article.cls*, *report.cls*, *book.cls*) verwendeten Satzspiegel wirken häufig zu groß. Abhilfe bieten die entsprechenden Klassen (*scrartcl.cls*, *scrreprt.cls*, *scrbook.cls*) aus dem KOMA-Script-Paket, oder das ebenfalls dort enthaltene *typearea.sty*. Die dazugehörige Dokumentation scrguide [3] enthält viele weiterführende Informationen.

Wenn man wirklich einen anderen Satzspiegel als den von z. B. *typearea.sty* erzeugten benötigt, dann sollte man bitte die Pakete *geometry.sty* oder *vmargin.sty* verwenden und nicht versuchen, „zu Fuß“ `\oddsidemargin` & Co. zu ändern.

Unter gar keinen Umständen sollte man an `\hoffset` bzw. `\voffset` herumfummeln, außer man kennt sich sehr gut mit den TeX-Interna aus.

1.3 Änderungen von Paketen und Klassen

Niemals Dokumentklassen (*article.cls*, *scrbook.cls* usw.) oder Stylefiles (*varioref.sty*, *color.sty*) direkt ändern! Entweder man bastelt sich Containerklassen bzw. -styles oder, man *kopiert* die Klassen/Styles, ändert die Kopie und speichert diese unter *anderem* Namen ab.

Ein Beispiel zur Erstellung von Containerklassen findet sich in der FAQ [7, Punkt 5.1.5].

Hinweis Generell sollte man solche zusätzlich installierten Klassen und Pakete entweder in den lokalen oder den `$HOME`-TeX-Baum speichern, damit bei einem Upgrade der TeX-Distribution diese Änderungen nicht überschrieben werden. Braucht man diese Änderungen nur in dem speziellen Projekt und möchte es weitergeben, dann könnte man die angepasste Klasse auch im aktuellen Projektverzeichnis speichern.

1.4 Änderung des Zeilenabstandes mittels `\baselinestretch`

Anpassungen von Parametern sollten generell auf der obersten dafür vorgesehenen Ebene der Benutzungsschnittstelle erfolgen. Eine Änderung des Durchschusses kann auf drei Ebenen durchgeführt werden:

1. Verwendung des Paketes *setspace.sty*
2. Verwendung der LaTeX-Anweisung `\linespread{<Faktor>}`
3. Umdefinition von `\baselinestretch`

Eine Umdefinition von Parametern wie `\baselinestretch` stellt die unterste Ebene dar und sollte deshalb Paketen vorbehalten bleiben. Eine bessere, da extra dafür vorgesehene Methode, ist die Verwendung von `\linespread{<Faktor>}`. Die beste Vorgehensweise stellt aber die Verwendung des Paketes *setspace.sty* dar, das sich bei der Änderung des Durchschusses um die Beibehaltung der Abstände in Fußnoten oder Listenumgebungen kümmert, deren Änderung meist nicht erwünscht ist.

Wenn man also einen größeren Durchschuss benötigt (z. B. eineinhalbzeilig oder zweizeilig), bietet sich folglich zuerst das Paket *setspace.sty* an. Möchte man hingegen nur kleine Anpassungen für andere Schriften als Computer Modern (z. B Palatino) machen, dann kann bzw. sollte man `\linespread{<Faktor>}` verwenden. Bei der Schrift Palatino würde sich zum Beispiel `\linespread{1.05}` anbieten.

1.5 Absatzeinzug und -abstand (`\parindent`, `\parskip`)

Den Absatzeinzug (`\parindent`) zu ändern, kann manchmal sinnvoll sein. Man sollte dabei aber beachten, dass

- man den Einzug mit einer schriftabhängigen Größe (em) und nicht mit einer absoluten Größe (mm) ändert. „Schriftabhängig“ bedeutet hier *nicht*, dass sich der Absatzeinzug bei Schriftgrößenänderungen automatisch anpasst, sondern dass der Wert der aktuell aktiven Schrift benutzt wird.
- man LaTeX-Syntax verwendet, da diese die wenigsten Probleme birgt. Zum Beispiel ist sie für externe Programme/Skripte leichter zu parsen[6], für den Benutzer besser zu warten und es gibt keine Inkompatibilitäten mit anderen Paketen (*calc.sty*).

  Ersetze: `\parindent=1em` durch `\setlength{\parindent}{1em}`

Wenn man hingegen *keinen* Absatzeinzug, dafür aber einen zusätzlichen Durchschuss als Absatzkennzeichnung wünscht, sollte man *nicht* einfach

```
\setlength{\parindent}{0pt}
\setlength{\parskip}{\baselineskip}
```

verwenden. Da sich `\parskip` auch auf Listen, Verzeichnisse und Überschriften auswirkt, ist diese Vorgehensweise nicht zu empfehlen.

Das Paket *parskip.sty* und die KOMA-Script-Klassen treiben einigen Aufwand, um diese Begleiterscheinungen zu vermeiden. Zur Verwendung der verschiedenen Optionen (parskip, halfparskip usw.) von KOMA-Script siehe den scrguide [3]. Wenn eine der KOMA-Script-Klassen verwendet wird, muss das Paket *parskip.sty nicht* noch zusätzlich geladen werden.

[6] syntaktisch analysieren, aufspalten

A Beispiel zu `\sloppy`

Hier ist der Beispiel-Code, welchen Markus Kohm veröffentlicht hat:

```
\documentclass{article}

\setlength{\textwidth}{20em}
\setlength{\parindent}{0pt}
\begin{document}
\typeout{First without \string\sloppy\space and underfull \string\hbox}

tatata tatata tatata tatata tatata tatata ta\-ta\-tata
tatata tatata tatata tatata tatata tatata tata\-tata
tatata tatata tatata tatata ta\-tatatatt\-ta
tatata tatata tatata tatata tatata tatata ta\-ta\-ta\-ta

\typeout{done.}

\sloppy
\typeout{Second with \string\sloppy\space and underfull \string\hbox}

tatata tatata tatata tatata tatata tatata ta\-ta\-tata
tatata tatata tatata tatata tatata tatata tata\-tata
tatata tatata tatata tatata ta\-tatatatt\-ta
tatata tatata tatata tatata tatata tatata ta\-ta\-ta\-ta

\typeout{done.}
\end{document}
```

Quelle: Message-ID: <8557097.gEimXdBtjU@ID-107054.user.dfncis.de>

Tabelle 2: *Von (n)german.sty bzw. babel.sty mit der Option (n)german definierte Makros*

Makroname	Original Definition	deutsche Ausgabe
`\prefacename`	Preface	Vorwort
`\refname`[a]	References	Literatur
`\abstractname`	Abstract	Zusammenfassung
`\bibname`[b]	Bibliography	Literaturverzeichnis
`\chaptername`	Chapter	Kapitel
`\appendixname`	Appendix	Anhang
`\contentsname`	Contents	Inhaltsverzeichnis
`\listfigurename`	List of Figures	Abbildungsverzeichnis
`\listtablename`	List of Tables	Tabellenverzeichnis
`\indexname`	Index	Index
`\figurename`	Figure	Abbildung
`\tablename`	Table	Tabelle
`\partname`	Part	Teil
`\enclname`	encl	Anlage(n)
`\ccname`	cc	Verteiler
`\headtoname`	To	An
`\pagename`	Page	Seite
`\seename`	see	siehe
`\alsoname`	see also	siehe auch

[a] Nur in den `article`-Klassen
[b] Nur in den `report`- und `book`-Klassen

3.5 Die `\*name`-Makros

Da in `de.comp.text.tex` von Zeit zu Zeit danach gefragt wird, wie man zum Beispiel „Literatur“ in „Quellenverzeichnis“ ändern kann, habe ich in Tabelle 2 die entsprechenden Makros zusammengestellt. Sie sind aus *german.sty* entnommen.

Um beispielsweise das „Abbildungsverzeichnis“ in „Abbildungen“ umzubenennen, benutzt man folgenden Befehl:

```
\renewcommand*{\listfigurename}{Abbildungen}
```

Die anderen Makros lassen sich analog umbenennen. Bei der Verwendung von *babel.sty* muss man mit `\addto` arbeiten. Siehe auch die De-TeX-FAQ [7, Punkt 8.5.9].

```
\addto{\captionsngerman}{%
  \renewcommand*{\listfigurename}{Abbildungen}}
```


1.6 Abgesetzte Formeln mit `$$...$$`

Bitte nicht! `$$...$$` ist ein plainTeX-Befehl und sollte in LaTeX vermieden werden, da dadurch die vertikalen Abstände bei abgesetzten Formeln inkonsistent werden (siehe auch Abschnitt 3.3 auf Seite 14, insbesondere die Warnung bezüglich `displaymath` im Zusammenhang mit *amsmath.sty*). Ferner funktioniert die Klassenoption fleqn nicht mehr.

Ersetze: `$$...$$` durch `\[...\]`

oder

```
\begin{displaymath}
...
\end{displaymath}
```

1.7 `\def` vs. `\newcommand`

Makros sollte man *immer* mittels `\newcommand{\<name>}{...}` definieren und *nicht* mit `\def\<name>{...}`.[7]

Das Hauptproblem von `\def` ist, dass keine Überprüfung auf die Existenz eines Makros durchgeführt wird. Es wird deshalb ggf. ohne Fehlermeldung/Warnung überschrieben.

Bereits existierende Makros können mit `\renewcommand{\<name>}{...}` umdefiniert werden.

Wer genau weiß, *warum* er `\def` benötigt, weiß auch um dessen Nach- bzw. Vorteile und kann diesen Unterpunkt getrost ignorieren.

1.8 Verwendung von `\sloppy`

Der Schalter `\sloppy` sollte nicht verwendet werden, schon gar nicht global in der Präambel. Wenn man in einzelnen Absätzen Probleme mit dem Umbruch hat, gilt:

1. Überprüfen, ob die entsprechenden Trennmuster (z. B. mittels *(n)german.sty*) und T1-Schriften geladen sind (siehe auch FAQ [7, Punkt 5.3 ff.]).
2. Umformulieren. Man muss nicht unbedingt den Satz, in dem das Umbruchproblem auftritt umformulieren, oft reicht es schon einen der vorhergehenden oder nachfolgenden Sätze umzuformulieren/umzustellen.
3. Moderate Anpassung der Parameter, die TeX bei der Berechnung der Zeilenumbrüche verwendet. Axel Reichert hat in `de.comp.text.tex` einmal seine persönliche Anpassung[8] gepostet[9], mit der sich die meisten Umbruchprobleme bei weiterhin gutem Layout vermeiden lassen. (Man sollte hierbei allerdings beachten, dass nun auftretende Warnungen *wirklich* zu beherzigen und durch Umformulieren zu beseitigen sind.):

[7] Siehe auch [4, Abschnitt 2.7.2] und [6, Abschnitt 3.4]
[8] Man kann diese Werte natürlich nach dem persönlichen Geschmack ändern, aber man sollte vorallem bei `\emergencystretch` aufpassen. Sonst erhält man einen löchrigen Blocksatz, wie bei einem sehr bekannten Textverarbeitungsprogramm.
[9] Zu finden unter der Message-ID: `<a84us0$plqcm$7@ID-30533.news.dfncis.de>`

```
\tolerance 1414
\hbadness 1414
\emergencystretch 1.5em
\hfuzz 0.3pt
\widowpenalty=10000
\vfuzz \hfuzz
\raggedbottom
```

Erst wenn diese Punkte nicht geholfen haben, kann man versuchen, mit der `sloppypar`-Umgebung den nachfolgenden Absatz „lockerer“ zu setzen.

tatata tatata tatata tatata tatata tatata tatata tatata tatata tatata tatata tatata tatata tatata tatata tatatatata tatata tatata tatata tatata tatata tatatatatta tatata tatata tatata tatata tatata tatata tatata

Abbildung 1: *Beispiel mit LaTeXs Standardwerten*

tatata tatata tatata tatata tatata tatata tatata tatata tatata tatata tatata tatata tatata tatata tatata tatatatata tatata tatata tatata tatata tatata tatatatattta tatata tatata tatata tatata tatata tatata tatata

Abbildung 2: *Beispiel mit* `\sloppy`

In den Abbildungen 1 und 2 habe ich versucht, den Effekt von `\sloppy` darzustellen. Bei der hier verwendeten „Times“ ist die negative Auswirkung von `\sloppy` auf Grund der sehr geringen Laufweite nicht so extrem, wie beispielsweise bei der „Computer Modern Roman“. Der prinzipielle Effekt sollte aber dennoch erkennbar sein.

Markus Kohm hat in `comp.text.tex` ein Beispiel veröffentlicht, welches den Effekt sehr deutlich zeigt. Ich habe es mit seiner Erlaubnis angehängt (siehe Anhang A auf Seite 17).

`align*` ersetzt werden. Für eine vollständige Übersicht der Möglichkeiten von *amsmath.sty* verweise ich auf die Dokumentation amsldoc [1].

Ersetze:
```
\begin{eqnarray}
  a &=& b \\
  b &=& c \\
  a &=& c
\end{eqnarray}
```
durch
```
\begin{align}
  a &= b \\
  b &= c \\
  a &= c
\end{align}
```

3.4 Die Verwendung von `\graphicspath`

Das beliebte Makro `\graphicspath` sollte aus folgenden Gründen vermieden und durch die Umgebungsvariable `TEXINPUTS` ersetzt werden[17]:

1. Die Verzeichnistrenner sind *nicht* plattformunabhängig (Windows/Unices: `/`, Macs: `:`).
2. Die TeX-Suche dauert länger, als diese Aufgabe durch die kpathsea-Bibliothek lösen zu lassen. (Bei heutigen Prozessoren fällt das nicht mehr so ins Gewicht.)
3. Jedes Bild verbraucht einen Teil des begrenzten TeX-Speichers *und* gibt ihn während des gesamten Kompilierens nicht mehr frei.

Bei einer Bourne-Shell kann man z. B. folgenden Aufruf

```
$ TEXINPUTS=Bildverz:$TEXINPUTS latex datei.tex
```

verwenden, oder in der `~/.profile` zusätzlich

```
export TEXINPUTS=./Bildverz:$TEXINPUTS
```

eintragen. Bei letzterem werden die Dateien im `Bildverz` unterhalb des aktuellen Arbeitsverzeichnisses gefunden.
Unter MS Windows ($\leq$ 98) setzt man die Variable mittels

```
set TEXINPUTS=.\Bildverz;%TEXINPUTS%
```

in der `autoexec.bat`. Bei Windows NT4 und Windows 2000 kann man Umgebungsvariablen über Systemsteuerung → System → Umgebung und seit Windows XP über Systemsteuerung → System → Erweitert → Umgebungsvariablen setzen.

Die hier aufgeführten Vorgehensweisen sind Beispiele. Mir ist durchaus bewusst, dass man `TEXINPUTS` auch auf andere Art und Weise bzw. in anderen Dateien anpassen kann. Genaueres ist der Dokumentation des verwendeten Betriebsystems bzw. der TeX-Distribution zu entnehmen.

[17] Vgl. David Carlisles Antwort auf Markus Kohms „Bug-Report“: `http://www.latex-project.org/cgi-bin/ltxbugs2html?pr=latex/2618`

3 Verschiedenes

Dieser Abschnitt enthält – mit Ausnahme von 3.2 – eher allgemeine Tipps und Hinweise als „Sünden“.

3.1 Gleitumgebungen – „figure“, „table“

Um den Inhalt einer Gleitumgebung zu zentrieren, sollte man `\centering` an Stelle der `\begin{center}`-`\end{center}`-Umgebung verwenden, da diese zusätzlichen vertikalen Abstand einfügt, der meistens nicht erwünscht ist.

Ersetze:
```
\begin{figure}
  \begin{center}
  \includegraphics{bild}
  \end{center}
\end{figure}
```
durch
```
\begin{figure}
  \centering
  \includegraphics{bild}
\end{figure}
```

Anmerkung Wenn man innerhalb des Fließtextes oder der `titlepage`-Umgebung einen Bereich zentrieren möchte, kann dieser zusätzliche Abstand natürlich durchaus erwünscht sein.

3.2 Der Anhang

Der Anhang wird mit dem *Schalter* `\appendix` eingeleitet. Er ist *keine* Umgebung.

Ersetze:
```
\begin{appendix}
\section{Blub}
\end{appendix}
```
durch
```
\appendix
\section{Blub}
```

3.3 Mathematiksatz

Generell sollte man für komplizierteren Mathematiksatz *amsmath.sty* benutzen. Es bietet neue Umgebungen, die vor allem `eqnarray` ersetzen sollen. Die Vorteile des Paketes:

- Abstände innerhalb und außerhalb von Umgebungen sind konsistenter.
- Gleichungsnummern werden so positioniert, dass sie nicht mehr überdruckt werden.
- Die neuen Umgebungen (z. B. `split`) ermöglichen es, lange Gleichungen einfacher zu umbrechen.
- Einfache Möglichkeit, neue Operatoren (ähnlich wie `\sin` usw.) mit sauberen Zeichenabständen zu definieren.

Warnung Bei der Verwendung von *amsmath.sty* sollte man die Umgebungen `displaymath`, `eqnarray` und `eqnarray*` *keinesfalls* weiterverwenden, da diese von *amsmath.sty* nicht unterstützt werden und deshalb wieder inkonsistente Abstände ergäben.

`\[...\]` wird von *amsmath.sty* korrekt angepasst und kann an Stelle von `displaymath` genutzt werden. `eqnarray` und `eqnarray*` kann in erster Näherung durch `align` bzw.

2 Veraltete Befehle, Klassen und Pakete

Markus Kohm hat ein Perl-Script geschrieben, mit dem man online auf `http://kohm.de.tf/markus/texidate.html` seine Dateien auf die häufigsten Fehler überprüfen kann.

Allerdings ist zu beachten, dass es keinen vollständigen TEX-Parser enthält und deshalb nur die „offensichtlichen“ Fehler überprüfen kann. Erst testen, dann posten.

2.1 Befehle

2.1.1 Änderung des Schriftstils

In Tabelle 1 sind die alten und aktuellen Befehle zur Änderung des Schriftstils gegenüber gestellt. Die als „lokal“ bezeichneten Makros wirken nur auf ihr Argument, wohingegen die als „global/Schalter“ bezeichneten, sich auf den gesamten folgenden Text bis zum Ende der aktuellen Gruppe auswirken.

Tabelle 1: *Befehle zur Änderung des Schriftstils*

veraltet	Ersatz in LaTeX 2ε	
	lokal	global/Schalter
`{\bf ...}`	`\textbf{...}`	`\bfseries`
—	`\emph{...}`	`\em`[a]
`{\it ...}`	`\textit{...}`	`\itshape`
—	`\textmd{...}`	`\mdseries`
`{\rm ...}`	`\textrm{...}`	`\rmfamily`
`{\sc ...}`	`\textsc{...}`	`\scshape`
`{\sf ...}`	`\textsf{...}`	`\sffamily`
`{\sl ...}`	`\textsl{...}`	`\slshape`
`{\tt ...}`	`\texttt{...}`	`\ttfamily`
—	`\textup{...}`	`\upshape`

[a] Nützlich in Makrodefinitionen. Innerhalb des Texts sollte nicht `{\em ...}`, sondern `\emph{...}` verwendet werden.

Warum sollte man die alten Befehle nicht verwenden? Die alten Befehle unterstützen nicht die Schriftverwaltung[10] von LaTeX 2ε. `{\bf foo}` zum Beispiel setzt alle schon vorhandenen Schriftattribute zurück, bevor es „foo“ fett druckt. Das führt dazu, dass man nicht einfach einen fett-kursiven Stil durch `{\it \bf Test}` definieren kann. (Die angegebene Definition erzeugt: **Test**.). Die aktuellen Befehle `\textbf{\textit{Test}}` hingegen verhalten sich wie erwartet und erzeugen: ***Test***. Ferner gibt es bei den alten Befehlen keine „Italic-Korrektur“, z. B. *fünf*hundert (`{\it fünf}hundert`) und *fünf*hundert (`\textit{fünf}hundert`).

[10] NFSS: New Font Selection Scheme. Einen Überblick über das NFSS gibt der fntguide [5].

2.1.2 Mathematische Brüche (`\over` vs. `\frac`)

Der Befehl `\over` sollte vermieden werden. `\over` ist ein TEX-Befehl, der durch die von LATEX abweichende Syntax schlechter bzw. nicht parsbar ist. Insbesondere das Paket *amsmath.sty* definiert `\frac{}{}` um und man erhält Fehlermeldungen bei der Verwendung von `\over`. Ein weiteres Argument für die Verwendung von `\frac{}{}` ist die für den Benutzer – vorallem bei komplexeren Brüchen – einfachere Zuordnung von Zähler und Nenner.
Ersetze: `$a \over b$` durch `$\frac{a}{b}$`

2.1.3 Zentrierung mit `\centerline`

Der Befehl `\centerline` ist ebenfalls ein TEX-Befehl und sollte in LATEX vermieden werden. `\centerline` ist einerseits inkompatibel zu einigen LATEX-Paketen (z. B. *color.sty*) und andererseits kann die Verwendung zu unerwünschten bzw. unerwarteten Effekten führen:

```
\begin{enumerate}
\item \centerline{Ein Punkt}
\end{enumerate}
```

Ein Punkt
1.

Ersetze: `\centerline{...}` durch `{\centering ...}`
oder
```
\begin{center}
...
\end{center}
```

Anmerkung Hinweise zur Zentrierung von Grafiken und Tabellen sind in Abschnitt 3.1 auf Seite 14 zu finden.

2.2 Klassen und Pakete

2.2.1 *scrlettr.cls* vs. *scrlttr2.cls*

Die Klasse *scrlettr.cls* aus dem KOMA-Script-Paket ist veraltet und wurde durch *scrlttr2.cls* ersetzt. Um ein *ähnliches* Layout wie die alte Klasse zu erreichen, kann man die Klassenoption KOMAold verwenden, die einen Kompatibilitätsmodus zur Verfügung stellt.

Ersetze:	durch
`\documentclass{scrlettr}`	`\documentclass[KOMAold]{scrlttr2}`

Anmerkung Für neue Vorlagen und Briefe, sollte aber die neue Schnittstelle benutzt werden, da diese deutlich flexibler ist.

Eine Erklärung der Unterschiede des Benutzerinterfaces würde den Rahmen dieser Übersicht sprengen, deshalb muss ich hier auf den scrguide [3] verweisen.

2.3.6 Aufrechte griechische Buchstaben

Die im folgenden rot markierten Passagen sind nicht veraltet im Sinne von „man soll sie nicht mehr benutzen“, aber es gibt nun mit dem Paket *upgreek.sty* eine Vereinfachung der Eingabe. Hinweise zur Benutzung bitte wie immer der Dokumentation upgreek [9] entnehmen.

Die *pifont.sty*-Tricks

Ersetze:
```
\usepackage{pifont}
\newcommand{\uppi}{\Pisymbol{psy}{112}}
\uppi
```
oder
```
\newcommand[1]{\upgreek}{%
 \usefont{U}{psy}{m}{n}#1}
\upgreek{p}
```
durch
```
\usepackage{upgreek}
$\uppi$
```

Der *babel.sty*-Trick

Ersetze:
```
\usepackage[greek,...]{babel}
\newcommand[1]{\upgreek}{%
 \foreignlanguage{greek}{#1}}
\upgreek{p}
```
durch
```
\usepackage{upgreek}
$\uppi$
```

2.3.7 *euler.sty* vs. *eulervm.sty*

Das Paket *euler.sty* sollte durch *eulervm.sty* ersetzt werden, da es Kompatibilitätsprobleme mit anderen Paketen ausräumt und einige Detailverbesserungen enthält:

- `\hbar` (`\hslash` bei dieser Schrift) funktioniert nun sauber
- Fette Mathematikschriften inklusive griechische Symbole sind möglich.

Genauere Informationen siehe eulervm [8].
Ersetze: `\usepackage{euler}` durch `\usepackage{eulervm}`

Anmerkung Der Skalierungsfaktor für *helvet.sty* in Kombination mit der Times sollte zwischen 0.90 und 0.92 liegen.

2.3.2 *mathptm.sty*

Das Paket *mathptm.sty* ist der Vorgänger von *mathptmx.sty*.
Ersetze: `\usepackage{mathptm}` durch `\usepackage{mathptmx}`

2.3.3 *pslatex.sty*

Das Paket *pslatex.sty* arbeitet intern wie *mathptm.sty* + *helvet.sty* (skaliert), wobei allerdings eine zu eng laufende Courier gewählt wird. Der Hauptnachteil von *pslatex.sty* ist, dass es *nicht* mit T1- und TS1-Encoding funktioniert.
Ersetze: `\usepackage{pslatex}` durch
```
\usepackage{mathptmx}
\usepackage[scaled=.90]{helvet}
\usepackage{courier}
```

Anmerkung zu allen Times/Helvetica-Kombinationen Man kann auch als Schreibmaschinenschrift bei der `cmtt` bleiben, also auf das Laden von *courier.sty* verzichten.

2.3.4 *palatino.sty*

Das Paket *palatino.sty* verhält sich wie *times.sty* (außer das natürlich `\rmdefault` auf „Palatino“ gesetzt wird) und sollte deshalb nicht mehr benutzt werden.
Ersetze: `\usepackage{palatino}` durch
```
\usepackage{mathpazo}
\usepackage[scaled=.95]{helvet}
\usepackage{courier}
```

Anmerkung Der Skalierungsfaktor für *helvet.sty* in Kombination mit der Schrift Palatino sollte 0.95 betragen.

Die „Helvetica“ ist *nicht* die optimale serifenlose Schrift in Kombination mit der „Palatino“, aber die beste *freiverfügbare*. Wer eine (auch ältere) CorelDraw®-CD besitzt, kann die „Palatino“ auch sehr gut mit den Schriften „Frutiger“[14] oder „Optima“[15] kombinieren. Walter Schmidt hat auf seiner Homepage[16] die entsprechenden TEX-Anpassungen veröffentlicht.

2.3.5 *mathpple.sty*

Dieses Paket ist der Vorläufer von *mathpazo.sty*. Ihm fehlen einzelne Zeichen, die Schriften werden aus den Euler-Fonts genommen, andere Zeichen passen nicht gut zu Palatino und die Zeichenabstände sind zum Teil falsch. Genaueres siehe psnfss2e [10].

[14] Bitstream „Humanist 777“, bfr
[15] Bitstream „Zapf Humanist“, bop
[16] Schriften für TEX: `http://home.vr-web.de/was/fonts`

2.2.2 *epsf.sty*, *psfig.sty*, *epsfig.sty* vs. *graphics.sty*, *graphicx.sty*

Die Pakete *epsf.sty* und *psfig.sty* sind durch *graphics.sty* oder *graphicx.sty* zu ersetzen. *epsfig.sty* ist nur ein Wrapper[11] um *graphicx.sty* für alte Dokumente, die mit *psfig.sty* erstellt wurden.

Da *epsfig.sty* intern *graphicx.sty* benutzt, *kann* man es noch verwenden, sollte aber für neu erstellte Dokumente auf das modernere *graphics.sty* oder *graphicx.sty* umsteigen. Die Syntax der beiden letztgenannten Pakete ist deutlich flexibler. Das Paket *epsfig.sty* wird hauptsächlich aus Kompatibilitätsgründen mitgeliefert.

Zu den Unterschieden zwischen den Paketen *graphics.sty* und *graphicx.sty* siehe grfguide [2]. Hinweise zur Zentrierung von Grafiken siehe Abschnitt 3.1 auf Seite 14.
Ersetze: `\usepackage{psfig}` durch `\usepackage{graphicx}`
`\psfig{file=Bild,...}` `\includegraphics[...]{Bild}`

2.2.3 *doublespace.sty* vs. *setspace.sty*

Um den Durchschuss zu ändern, sollte man das Paket *setspace.sty* verwenden. *doublespace.sty* ist veraltet und wurde durch *setspace.sty* ersetzt. Siehe auch Abschnitt 1.4 auf Seite 3.
Ersetze: `\usepackage{doublespace}` durch `\usepackage{setspace}`

2.2.4 *fancyheadings.sty*, *scrpage.sty* vs. *fancyhdr.sty*, *scrpage2.sty*

Das Paket *fancyheadings.sty* wurde durch *fancyhdr.sty* ersetzt. Eine weitere Alternative für angepasste Kopfzeilen bietet das Paket *scrpage2.sty* aus dem KOMA-Script-Bundle. Auch hier ist zu beachten, dass nicht *scrpage.sty* verwendet wird. Die Anleitung zu *scrpage2.sty* befindet sich im scrguide [3].
Ersetze: `\usepackage{fancyheadings}` durch `\usepackage{fancyhdr}`
Ersetze: `\usepackage{scrpage}` durch `\usepackage{scrpage2}`

2.2.5 Die *caption.sty*-Familie

Das Pakete *caption2.sty* sollte nicht mehr verwendet werden, da es eine neue Version (v3.x) von *caption.sty* gibt. Man sollte allerdings darauf achten, dass die neuste Version des Paketes benutzt wird. Um dies sicherzustellen, muss man das Paket folgendermaßen laden:
Ersetze: `\usepackage{caption}` durch `\usepackage{caption}[2003/12/20]`

Hatte man vorher *caption2.sty* benutzt, dann sollte man unbedingt die Dokumentation anleitung [12, Abschnitt 8] beachten.

2.2.6 *isolatin.sty*, *umlaut.sty* vs. *inputenc.sty*

Generelles Im Prinzip gibt es vier Möglichkeiten, Umlaute und andere nicht-ASCII-Zeichen einzugeben:

1. `H{\"u}lle`: Der Vorteil dieser Art der Eingabe ist, dass sie immer und auf jedem System funktioniert.

[11] Hier: Ein Stylefile, welches ein oder mehrere andere aufruft und damit Funktionen nachbildet.

Die Nachteile hingegen sind, dass das Kerning[12] zwischen den Buchstaben zerstört wird, es in einem deutschsprachigen Text äußerst umständlich ist und dass es äußerst schlecht lesbar ist.

Diese Variante sollte man – auf Grund des Kerningproblems – *immer* vermeiden.

2. Die Eingabe der Form `H\"ulle` bzw. `H\"{u}lle` hat die oben genannten Kerningprobleme nicht und ist ebenfalls auf jedem System nutzbar.

   Die Nachteile bei dieser Art, sind auch hier die aufwändige Eingabe und schlechtere Lesbarkeit.

   Diese Variante ist die sinnvollste für Makrodefinitionen und Stylefiles, da sie encoding- und paketunabhängig ist.

3. Mit *(n)german.sty* bzw. der Option (n)german beim Paket *babel.sty* kann man die Umlaute etwas einfacher (`H"ulle`) eingeben. Der Vorteil ist auch hier wieder, dass es auf allen Systemen funktioniert. Da *babel.sty* bzw. *(n)german.sty* auf allen TEX-Installationen zu finden ist, sollte es auch keine Kompatibilitätsprobleme geben.

   Die Nachteile sind auch hier die umständlichere Eingabe und schlechtere Lesbarkeit.

   Diese Variante ist für Fließtext relativ gut verwendbar. Sollte aber in Makrodefinitionen und Präambeln vermieden werden.

4. Die direkte Eingabe (`Hülle`). Die Vorteile liegen auf der Hand. Der Text ist „normal“ schreib- und lesbar.

   Der Nachteil ist, dass man LATEX mit der verwendeten Eingabekodierung bekannt machen muss und dass es beim Austausch von Dateien zwischen verschiedenen Systemen evtl. zu Problemen kommen kann. Das ist *kein* Problem für TEX bzw. LATEX selbst, aber es kann evtl. zu *Darstellungsproblemen* in den Editoren auf den verschiedenen Systemen führen. Zum Beispiel könnte ein in iso-8859-15 (latin9) kodierter € in einem Editor unter Windows (CP1252) als ¤ *dargestellt* werden.

   Diese Variante ist sehr gut für Fließtext verwendbar. Sollte aber in Makrodefinitionen und Präambeln vermieden werden.

Zusammenfassend kann also gesagt werden, dass man in Makros, Präambeln und Stylefiles die Form `H\"ulle` oder `H\"{u}lle` verwenden sollte und im übrigen Text entweder `H"ulle` oder `Hülle`.

Eingabekodierung Um die verwendete Kodierung LATEX bekannt zu machen, sollte man *nicht isolatin1.sty* bzw. *isolatin.sty* oder *umlaut.sty* verwenden! Diese Pakete sind veraltet bzw. nicht auf allen Systemen vorhanden.

Korrekt ist das Paket *inputenc.sty* mit folgenden Optionen zu benutzen:

latin1/latin9 für unixoide Systeme (latin1 ist auch unter MS Windows und Mac OS X verwendbar)

[12] Einfügen positiver bzw. negativer Abstände zwischen Zeichen in Abhängigkeit der Zeichenkombination

ansinew für MS Windows

applemac für Macs[13]

cp850 für OS/2

Ersetze: `\usepackage{isolatin1}` durch `\usepackage[latin1]{inputenc}`
Ersetze: `\usepackage{umlaut}` durch `\usepackage[latin1]{inputenc}`

2.2.7 *t1enc.sty* vs. *fontenc.sty*

Dieses Thema ist in der FAQ [7, Punkt 10.1 ff.] eigentlich ausreichend erörtert. Hier nur kurz der Hinweis, dass das Paket *t1enc.sty* veraltet ist und deshalb durch *fontenc.sty* ersetzt werden sollte!
Ersetze: `\usepackage{t1enc}` durch `\usepackage[T1]{fontenc}`

2.2.8 *natdin.bst* vs. *dinat.bst*

Das Stylefile *natdin.bst* wurde durch *dinat.bst* ersetzt.
Ersetze: `\bibliographystyle{natdin}` durch `\bibliographystyle{dinat}`

2.3 Schriften

Das Thema „Schriften und LATEX“ ist ein Quell ewiger „Freude“ in `de.comp.text.tex`, meistens ausgelöst durch die Frage, warum denn die Schrift im Acrobat® Reader so pixelig sei. Die häufigsten *falschen* Antworten auf diese Frage verweisen auf *times.sty* bzw. *pslatex.sty*. Durch die Nutzung dieser Pakete werden gänzlich andere Schriften eingestellt.

Um „schöne“ Schriften (Computer Modern) im AR zu erhalten, sei hiermit auf die FAQ [7, Punkte 10.1.7/10.1.8] verwiesen.

Generelle Informationen zu LATEX 2ε's New Font Selection Scheme (NFSS) gibt der fntguide [5].

2.3.1 *times.sty*

Das Paket *times.sty* ist veraltet (siehe psnfss2e [10]). Es stellt `\rmdefault` auf die Schrift „Times“, `\sfdefault` auf „Helvetica“ und `\ttdefault` auf „Courier“ um, *ohne* jedoch die passenden Mathematikschriften einzubinden. Ferner wird die Helvetica nicht korrekt skaliert und wirkt zu groß. Wenn man die Kombination Times/Helvetica/Courier benutzten möchte, dann folgendermaßen:
Ersetze: `\usepackage{times}` durch

```
\usepackage{mathptmx}
\usepackage[scaled=.90]{helvet}
\usepackage{courier}
```

[13] Bei der Verwendung von OS X sollte ebenfalls latin1/latin9 verwendet werden, da diese Kodierung besser für den Dateiaustausch mit Benutzern anderer Betriebssysteme geeignet ist als applemac. Allerdings sollte unbedingt die Eingabekodierung des verwendeten Editors beachtet werden. In Zukunft wird hoffentlich auch Unicode-Unterstützung in LATEX Einzug halten, momentan ist diese allerdings noch experimentell.