

Einfach(e) Dokumente mit L^AT_EX erzeugen (unter MS-Windows)¹

Wolfgang Ludwig-Mayerhofer
Universität Siegen
Philosophische Fakultät
57068 Siegen
ludwig-mayerhofer@soziologie.uni-siegen.de

2. Mai 2016

¹ Gegenüber der letzten Fassung (September 2014) etwas geänderte und erweiterte Version Mai 2016. Beachten Sie: Das meiste, was hier steht, gilt auch für Personen, die andere Betriebssysteme nutzen. Es gibt aber für jedes Betriebssystem ein paar Besonderheiten, die hier eben nicht behandelt werden.

Inhaltsverzeichnis

1	Warum \LaTeX?	3
2	Worauf muss man beim Arbeiten mit \LaTeX achten?	4
3	Über diesen Text	5
4	Was unterscheidet das Arbeiten mit \LaTeX von der gewohnten Art und Weise, Texte am PC zu erzeugen?	6
5	\LaTeX installieren sowie Hilfen zum Einstieg	6
6	Mit \LaTeX arbeiten – die Grundzüge	8
6.1	Prinzipielles	8
6.2	Der Kopf des Dokuments („Präambel“)	9
6.3	Hauptelemente eines \LaTeX Dokuments	10
6.3.1	Titelseite	11
6.3.2	Text	11
6.3.3	Etwas mehr über die Gestaltung von Text	12
6.3.4	Spezielle Zeichen	13
6.3.5	Silbentrennung	14
6.3.6	Fester Zeilen- oder Seitenumbruch	15
6.4	Besonderheiten der deutschen Sprache und Feinheiten des Satzes	15
6.4.1	Umlaute und ß	15
6.4.2	Anführungszeichen	15
6.4.3	Gedankenstriche	16
6.4.4	Besondere Probleme von Trennungsstrichen	16
6.4.5	Zitate freistellen	16
6.4.6	Den Schriftgrad wechseln	17
6.4.7	Zwischenräume/Abstände	17
6.4.8	Mediävalziffern	18
6.4.9	Text zusammenhalten	19
6.4.10	Punkte zur Signalisierung von Auslassungen	19
7	Mehr	19
7.1	Kommentare	19
7.2	Graphiken	20
7.3	Inhaltsverzeichnis	22
7.4	Verweise im Text	22
7.5	Fußnoten	22
7.6	Tabellen	23
7.7	Textboxen (umrahmter Text)	25

8	Gesamtdokument, Seitenlayout	25
8.1	Andere Dokumentklassen	27
8.2	Das Seitenlayout	28
8.3	Andere Schriften (Fonts)	29
8.4	Andere Schriften mit XeLaTeX	30
9	Stuff not easily found	32
9.1	Fallunterscheidungen in mathematischen Formeln	32
9.2	„Typographische“ Brüche	33
9.3	Das Komma als Dezimalzeichen in mathematischen Umgebungen	33
9.4	Mathematik im Fließtext und Zeilenabstand	33
9.5	Durchstreichen	34
10	Englisch	34

1 Warum L^AT_EX?

Stellen Sie sich vor, Sie schreiben einen Text von mittlerer Länge, etwa eine studentische Haus- oder Abschlussarbeit oder einen vergleichbaren Aufsatz. Dieser Text wird Kapitel oder Abschnitte (und Unterkapitel bzw. Unterabschnitte) haben, vielleicht gibt es dort auch gelegentlich listenförmige Aufzählungen oder vergleichbare Elemente.

Derzeit gängige Textverarbeitungssoftware kann Gliederungen durchaus unterstützen. Die Unterstützung ist jedoch nicht systematisch und wird Ihnen auch nicht aktiv von Seiten des Programms angeboten; Sie müssen sie sich ‚holen‘. Nehmen wir z. B. Kapitelüberschriften: Die meisten Menschen wissen, dass man diese vom Fließtext absetzen soll. Sie werden also irgendwie anders formatiert (etwa mit einem größeren Font, evtl. Fett- oder Kursivschrift). Sollen sie auch nummeriert werden? Und wie geht das? Gar nicht so einfach, außer man macht es per Hand, was fehleranfällig ist. Oder die Aufzählungen: Heute verfährt die Software meist so, dass sie von selbst versucht zu erkennen, ob sie eine Aufzählung darbieten wollen oder nicht. Sprich, die Software macht, was sie will, und nicht, was Sie wollen. Alle diese Dinge kann man mit Formatvorlagen in den Griff bekommen, die allerdings wenige Menschen kennen und nutzen. Und wenn sie sie kennen und nutzen, bekommen sie trotzdem nicht immer, was sie wollen, und die Bedienung ist weit davon entfernt, eingängig zu sein.

Wäre es da nicht einfacher, eine Software zu haben, die wie folgt funktioniert:

- Wenn Sie eine Kapitelüberschrift haben wollen, schreiben Sie einfach „Kapitelüberschrift“ und danach die Kapitelüberschrift in Klammern.
- Wenn Sie eine Aufzählung machen möchten, schreiben Sie einfach „Beginn der Aufzählung“, dann die Aufzählung, und zuletzt „Ende der Aufzählung“.
- Wenn Sie ein Inhaltsverzeichnis haben wollen, schreiben Sie einfach „Inhaltsverzeichnis“.
- Usw.

Nun, diese Software gibt es, dieser Text handelt von ihr. Sie heißt L^AT_EX (gesprochen Latech). Der einzige kleine Unterschied zu dem, was ich gerade beschrieben habe, liegt darin, dass Sie die Befehle auf Englisch schreiben. Aber das kann heutzutage ja jeder.

Man kann noch weitere Gründe für L^AT_EX angeben, etwa die folgenden:

- L^AT_EX erzeugt ein ordentliches Schriftbild und generell ein ordentliches Aussehen von Texten. Damit unterscheidet es sich ziemlich stark von verbreiteter Textverarbeitungssoftware.

- \LaTeX ist genau und logisch und ermuntert so die, die es benutzen, zu genauem und logischem Arbeiten.
- \LaTeX kostet nichts, obwohl es ein äußerst leistungsfähiges Programm ist.
- \LaTeX -Texte haben von Haus aus Seitenzahlen, Blocksatz mit Silbentrennung (also ohne die hässlichen Löcher im Text, die man heute immer häufiger sieht, weil auf die Silbentrennung verzichtet wird) und korrekt nummerierte Überschriften. Wer viel mit von Studierenden oder auch Kolleginnen und Kollegen verfassten Texten zu tun hat, kann sich schon deshalb kaum etwas sehnlicher wünschen, als alle Welt mit \LaTeX arbeiten zu sehen.

Nur ganz kurz sei auf folgendes hingewiesen: Hinter oder besser gesagt unter \LaTeX steckt noch eine andere Software, nämlich \TeX , das viel komplizierter ist. \LaTeX hilft also eigentlich nur dabei, mit \TeX zu arbeiten. Tatsächlich kann man sich auch nicht einfach \LaTeX auf den Rechner holen, sondern spezifische Aufbereitungen von \LaTeX und \TeX . Für Windows-User ist heute MiKTeX die Implementation der Wahl. Aber an MiKTeX kommt man wiederum am leichtesten, wenn man sich eine entsprechend aufbereitete Distribution holt, siehe dazu Abschnitt 5.

2 Worauf muss man beim Arbeiten mit \LaTeX achten?

Viele Leute sagen: „Man muss da so viel lernen“. Gut, wenn Sie zu den Menschen gehören, die die Vorstellung, etwas zu lernen, perhorreszieren, sollten Sie vielleicht die Finger davon lassen. Wer an Lernen grundsätzlich mit der Assoziation herangeht, man „müsse“ es tun, sollte übrigens auch nicht studieren. Aber das nur nebenbei.

Tatsächlich ist für die meisten Menschen der Lernaufwand, um erfolgreich mit \LaTeX arbeiten zu können, *gering*. \LaTeX kann durchaus kompliziert sein: Es erlaubt einen perfekten Formelsatz, aber auch wenn der Formelsatz ähnlich einfach geht wie andere Elemente (benötigen Sie einen Bruch? Schreiben Sie einfach „Bruch“ oder genauer gesagt das englische Äquivalent „fraction“ bzw. die Abkürzung „frac“), sind komplexe Formeln, wenig überraschend, etwas unübersichtlich. Wenn Sie große (!) Tabellen schreiben wollen oder müssen, kann es auch etwas schwierig werden. Aber die meisten Menschen benötigen derlei gar nicht. Und die, die komplexe Formeln schreiben wollen, lernen sowieso kurz über lang \LaTeX , weil es anders kaum geht (außer, man nimmt enorme Abstriche hinsichtlich Lesbarkeit in Kauf).

Auch die Installation ist nach meinen Erfahrungen heute kein Problem mehr (in früheren Versionen dieses Textes wurde sie als mögliche Problem-

quelle erwähnt). Ich kann das nur für Windows sagen und nur für den Fall, dass Ihr Rechner nicht allzu alt ist; aber unter dieser Bedingung sollte bei Verwendung eines der weiter unten erwähnten Pakete eigentlich nichts schief gehen.

Gleichwohl: Gerade Einsteiger können in die Verzweiflung getrieben werden, wenn sie Fehlermeldungen erhalten, deren Ursache sie nicht kennen und die auch nicht immer transparent sind – und das wird unweigerlich eintreten (genauer verstehen werden Sie das nach Lektüre von Abschnitt 4). Zunächst also der Hinweis, dass die meisten Fehlermeldungen einen Hinweis auf die Stelle erhalten, an der der Fehler aufgetreten ist. Das hilft schon mal. Um was für Fehler handelt es sich aber? Nach meiner Erfahrung sind die zwei wichtigsten Fehlerquellen:

1. *Ungenauigkeit beim Arbeiten:* Wie schon oben an einem Beispiel verdeutlicht, haben nicht wenige Anweisungen die Struktur „Beginn . . . Ende“. Nun kann es vorkommen, dass man den abschließenden Befehl vergisst – und schon ist \LaTeX verwirrt. Ähnlich werden häufig Elemente in geschweiften Klammern eingeschlossen; wird nun die abschließende Klammer vergessen, gibt es Wirrwarr. Übrigens erzeugen solche Fehler oft eine große Zahl von Fehlermeldungen, auch wenn der Fehler nur ein- oder zweimal auftaucht. Also: Mit der Fehlersuche und -behebung *immer* vorne beginnen und einen Fehler nach dem anderen abarbeiten!
2. *Spezielle Zeichen:* In \LaTeX kann man einige Zeichen der Schreibmaschinentastatur nicht einfach im Text verwenden. Außer bei Buchstaben, Zahlen und den einfachen Satzzeichen kann man sich nicht darauf verlassen, dass ein auf Ihrer Tastatur enthaltenes Zeichen so erscheint, wie es soll. Und noch schlimmer: Manche Zeichen haben bei \LaTeX eine besondere Bedeutung. Der Unterstrich `_` beispielsweise bedeutet in \LaTeX , dass ein Zeichen oder eine Zeichenfolge tiefgestellt werden soll! Benötigt man also einen Unterstrich im Text, muss man wissen, wie man diesen erzeugt. Ein wichtiger Tipp: Wie Sie noch sehen werden, beginnen alle \LaTeX -Befehle mit einem Backslash, also dem Zeichen `\`. Den Unterstrich erzeugt man also zum Glück einfach durch den Befehl `_`. Und wie erzeugt man einen Backslash? Nicht durch `\\` – das würde sogar ein intelligentes Programm wie \LaTeX verwirren. Mehr dazu finden Sie in Abschnitt 6.3.4 auf Seite 13, den Sie auf jeden Fall im Auge behalten sollten, wenn Sie zu schreiben beginnen.

3 Über diesen Text

\LaTeX kann enorm viel. Dieser Text erzählt Ihnen in recht autoritärer Manier maximal ein halbes Prozent davon – das halbe Prozent, das ein Einsteiger oder eine Einsteigerin (sagen wir: Oberstufenschüler oder jemand, der gerade

ein Studium an der Universität aufnimmt) benötigt, um eine Seminararbeit oder einen vergleichbaren (auch kürzeren) Text zu erstellen. (Wenn Sie Mathematik oder Physik studieren, kommen Sie an \LaTeX ohnehin nicht vorbei, weil keine Software so gut Formeln schreiben kann. Das wird hier aber nicht behandelt. Auch Ökonomen, vor allem wenn sie ökonometrisch arbeiten, verwenden \LaTeX gerne.)

Fangen Sie einfach mit dem an, was Sie hier lernen. Für aufwändigere Texte (Abschlussarbeiten etc.) können Sie dann Ihr Wissen erweitern. Solange Sie nicht ganz spezielle Dinge brauchen, fehlt für solche Arbeiten, beherrschen Sie das hier Vorgestellte, gar nicht mehr viel, falls überhaupt etwas.

4 Was unterscheidet das Arbeiten mit \LaTeX von der gewohnten Art und Weise, Texte am PC zu erzeugen?

Textverarbeitung heißt auch Textgestaltung. Übliche Textverarbeitungssoftware tut dies mittels Formatierungsbefehlen, die für die Nutzer intransparent sind: Man klickt auf einen Button (etwa ein F für Fettdruck) o. ä. und die Software gestaltet den Text entsprechend – wenn man Glück hat. Wo genau welcher Befehl im Dokument steht, lässt sich nur höchst vage erschließen.

In \LaTeX werden sämtliche Befehle zur Textgestaltung in Klartext in den Text geschrieben. Das ist auf den ersten Blick ungewohnt, da wir heute von WYSIWYG verwöhnt sind. Aber WYSIWYG heißt in der Praxis fast immer auch, dass man etwas bekommt, was man nicht gewollt hat. Bei \LaTeX sieht man erst hinterher, was man bekommt. Aber es ist, vor allem bei einiger Vertrautheit mit dem Programm, nahezu immer genau das, was man will – und wenn nicht, ist es in der Regel das, was man wollen sollte.

Ein weiterer Vorteil dieses Vorgehens ist: Da man für die Formatierungsbefehle Klartext verwendet, also normale Schriftzeichen und nicht irgendwelchen Geheimcode, sind mit \LaTeX erzeugte Texte plattformunabhängig. \LaTeX kann auch plattformunabhängigen Output erzeugen, das ist Output im PDF- oder PostScript-Format.

5 \LaTeX installieren sowie Hilfen zum Einstieg

- Ein nutzerfreundliches Paket von \LaTeX auf den eigenen Rechner downloaden und installieren. Empfehlenswert für Menschen, die unter Windows arbeiten, ist proTeXt, siehe <http://www.tug.org/protext/>, oder TeXLive, das auch für andere Betriebssysteme verfügbar ist, siehe <http://www.tug.org/texlive/>. Alternativ kann man beide Distributionen auch auf DVD bekommen, das kostet aber etwas Geld. ProTeXt basiert auf dem weiter oben erwähnten MikTeX, und ist daher vielleicht

für Einsteiger vorzuziehen, weil man hierzu wohl leichter im Internet Informationen und Hilfe findet.

- Die genannten Pakete enthalten auch sogenannte „Entwicklungsumgebungen“. Das sind Editoren, mit denen Sie Ihren Text schreiben (und dazu oft Unterstützung erhalten) und dann in ein les- oder druckbares Dokument verwandeln können. Man kann sie auch unabhängig von den Paketen im Internet finden und herunterladen. Aktuell werden TeXnicCenter oder TeXstudio (früher TeXMakerX genannt) favorisiert, aber es gibt zahlreiche andere Umgebungen, betriebsunabhängige wie solche, die auf spezifische Betriebssysteme zugeschnitten sind (letzteres gilt etwa für TeXnicCenter). Ich sage das nur der Vollständigkeit halber, weil der Inhalt der obigen Pakete auf jeden Fall genügt. TeXnicCenter bietet z. B. die Möglichkeit, Texte, die sich aus mehreren Dateien zusammensetzen, zu „Projekten“ zusammenzufassen, was beim Erstellen größerer Werke sehr hilfreich sein kann; auch wird die Bibliographie, sofern man sie mit BibTeX erzeugt (ja, das gibt es auch noch) automatisch eingebunden, während man sie in vielen anderen Umgebungen separat generieren muss. Dafür bieten andere Umgebungen vielleicht mehr Möglichkeiten, auch Alternativen zu L^AT_EX einzusetzen.¹
- Ich gehe hier davon aus, dass Sie PDF-Dateien erzeugen werden. Um diese zu betrachten, benötigt man den verbreiteten Acrobat Reader, den aber ohnehin jeder und jede auf seinem bzw. ihren Rechner hat (manche der eben erwähnten Entwicklungsumgebungen verfügen auch über eigene PDF-Betrachter, was im Alltagsbetrieb Vorteile hat – man muss die PDF-Datei nicht schließen, bevor man eine neue Version des Textes erzeugt). Das *Erzeugen* von PDF-Dateien erledigt Ihre Entwicklungsumgebung (mit Hilfe des Programms pdfLaTeX, das immer mitgeliefert wird). Für Kenner: Auch Postscript-Dateien können erzeugt werden (für deren Anzeige man dann Ghostview oder Ähnliches benötigt).
- Einführendes zu L^AT_EX (einschließlich Literaturhinweisen und Links) findet man unter <http://de.wikipedia.org/wiki/LaTeX>.²

¹ Ja, auch das gibt's, etwa LuaTeX oder XeTeX. Vielleicht gehört diesen beiden die Zukunft, weil sie noch universeller sind (es scheint bspw. leichter, andere – auch nicht-westliche – Schriftarten einzubinden). Aber L^AT_EX wird noch lange, lange existieren. Wer sich noch genauer damit befassen will: <http://texwelt.de/wissen/fragen/5868/was-ist-der-unterschied-zwischen-latex-pdflatex-lualatex-und-xelatex> (Achtung – für Einsteiger wohl unverständlich).

² Liebe Leserin, lieber Leser, falls Sie Soziologe oder Sozialwissenschaftler/in sind: Die auf diese Disziplin bezogenen Einträge in Wikipedia sind leider meist ziemlich erbärmlich. Das gilt aber nicht grundsätzlich, und der Eintrag zu L^AT_EX ist wirklich brauchbar; Sie erfahren dort u. a. auch mehr über die verschiedenen Betriebssysteme, was ich hier nicht leisten kann.

- Sehr viel Info bei der Deutschsprachigen Anwendervereinigung TeX e.V. unter <http://www.dante.de/>
- Hilfreiches steht auch in der folgenden Datei: <http://www.ctan.org/tex-archive/info/lshort/german/l2kurz.pdf>.
- Ich selbst verwende auch ein Buch zum Nachschauen: Kopka, Helmut: \LaTeX Band 1: Einführung, Bonn: Addison & Wesley, 2. Auflage 1996. Damit komme ich sehr weit. Es gibt aber bestimmt noch andere gute Bücher. Achtung: Die zweite Auflage dieses Buches sollte es schon sein, weil sich diese auf \LaTeX 2 ϵ bezieht, die seit 1994 offizielle Version.
- Ansonsten finden Sie eigentlich zu allem und jedem schnell Hilfe, wenn Sie in eine Suchmaschine Ihrer Wahl „Latex (weitere Stichworte)“ eingeben („Latex“ alleine kann je nach Suchmaschine und Ihrer Suchhistorie auch ganz andere Ergebnisse zeitigen ...). Gerade auf StackExchange gibt es viele Threads zu \LaTeX ; probieren Sie es also mal mit Englisch. Wegen dieser guten Möglichkeiten verzichte ich in diesem Text auch sehr weitgehend auf Nachweise; zu allem, was ich hier erwähne, finden Sie fix die nötige Information im Internet. Ich weiß, das ist nicht wirklich sauberer Stil, aber im Rahmen dieser informellen Einführung sei mir gestattet, mich mehr auf die Inhalte zu konzentrieren.

6 Mit \LaTeX arbeiten – die Grundzüge

6.1 Prinzipielles

- Starten Sie Ihre Entwicklungsumgebung.
- Schreiben Sie Ihren Text nach den unten vorgestellten Regeln in das Textfenster.
- Vermutlich finden Sie in Ihrer Entwicklungsumgebung einen Button, der für die Verwandlung in eine PDF-Datei sorgt (da ich nicht alle Umgebungen kenne, kann ich keine allgemeine Aussage machen). Schauen Sie in das Log-Fenster unten am Bildschirm, ob da was von „fatal error“ steht; wenn das der Fall ist, haben Sie irgendwo etwas falsch gemacht. Inspizieren Sie die in diesem Zusammenhang genannte Textstelle. Sie können sich darauf verlassen – in mindestens 99,9 Prozent der Fälle liegt das Problem bei Ihnen.

Hat alles geklappt, endet der Log mit folgendem Text: Output written on Dateiname.pdf (x pages, xxx bytes). Transcript written on Dateiname.log. Jetzt können sie auf den Button PDF klicken und die Datei wird angezeigt (im TeXnic-Center können sie auch veranlassen, dass die Datei automatisch nach gelungener Kompilation angezeigt

wird). Achten Sie aber darauf, vor dem Erzeugen einer neuen Version des PDF-Dokumentes die alte Version im Acrobat Reader zu schließen, sonst gibt's eine Fehlermeldung (oder Sie finden die Stelle in den Optionen Ihrer Entwicklungsumgebung, wo Sie diese veranlassen, eine offene PDF-Datei vor dem Kompilieren zu schließen). Übrigens: Wenn Sie ein Inhaltsverzeichnis oder automatische Verweise haben, sollten Sie die Verwandlung Ihrer Rohdatei in die PDF-Datei zwei- oder sogar dreimal durchführen, bevor Sie die Datei betrachten.³

6.2 Der Kopf des Dokuments („Präambel“)

Am Anfang des Dokumentes müssen (die ersten vier Zeilen) bzw. sollten (die nächsten beiden) – wenn Sie, wie ich hier unterstelle, einen Text in deutscher Sprache und zum Ausdruck in Deutschland (wo man Papier der Größe A4 verwendet) verfassen – folgende Befehle stehen (Sie müssten diese am Rechner direkt aus dieser Datei in Ihre Datei kopieren können):

```
\documentclass[a4paper,11pt,titlepage]{article}
\usepackage[ngerman]{babel}
\usepackage[ansinew]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{microtype}
\usepackage{eurosym}
```

Diese (und vergleichbare) Befehle werden als *Präambel* bezeichnet. Wie die Präambel eines Gesetzes oder eines Vertrages steht sie vor dem eigentlichen Inhalt. Weiter unten werden mögliche Erweiterungen dieser Angaben angeführt, die sich aus speziellen Bedürfnissen ergeben können. Die erste Zeile ist essentiell; sie definiert die Dokumentenklasse. Neben „article“ gibt es auch andere Klassen (z. B. „book“); diese beeinflussen vorrangig den Aufbau des Dokuments. Man kann mit „article“ aber sehr gut Arbeiten mittlerer Länge wie Studienabschlussarbeiten schreiben, und die folgenden Angaben beziehen sich nur hierauf. Etwa mehr dazu finden Sie in Abschnitt 8.1. Beachten Sie: Die Angabe `11pt` bezieht sich auf den Schriftgrad der Browserschrift (also der Schrift, in der der Fließtext gesetzt wird); wenn Sie diese weglassen,

³ Bei der Gelegenheit ein Hinweis: Beim Erzeugen der PDF-Datei werden neben dieser noch mehrere weitere Dateien angelegt, die alle den gleichen Namen wie Ihr Dokument, aber eine andere Endung (Extension) haben. Verzeichnisse, die \LaTeX Dokumente enthalten, können also recht schnell unübersichtlich werden. Sie können aber alle Dokumente außer Ihrem Originaldokument (das die Endung „.tex“ aufweist) jederzeit löschen. Allerdings kann es sinnvoll sein dies erst dann zu tun, wenn das Dokument wirklich fertig ist, denn zumindest bei komplexeren Texten sind einige dieser Dateien wertvolle Hilfsdateien, die beispielsweise schon die Struktur des Inhaltsverzeichnisses oder des Index' enthalten. Aber keine Sorge: Das einzige „Problem“, das entstehen kann, wenn Sie sämtliche zusätzlichen Dateien gelöscht haben, ist, dass Sie das Kompilieren Ihrer Datei ein paar Mal wiederholen müssen, bis Verzeichnisse, Querverweise etc. wieder stimmen.

wird als Schriftgrad 10 Punkt gewählt, was für ältere Menschen ein klein wenig leseunfreundlich ist. Sie können auch 10pt explizit angeben, oder 12pt. Andere Schriftgrade können an dieser Stelle nicht angegeben werden, sind für Texte auch nicht erforderlich. Der Zusatz `titlepage` bewirkt, dass die Angaben zum Titel (mehr dazu unten) auf einer eigenen Seite (natürlich der ersten im Dokument) ausgegeben werden.

Die Packages sind gewissermaßen Ergänzungen, teilweise auch Aktualisierungen, des Grundpakets. Ergänzungen sind beispielweise die auf Angabe der Dokumentenklasse folgenden drei Pakete. Sie machen aus dem US-amerikanischen $\text{T}_{\text{E}}\text{X}$ ein Programm, das Texte in deutscher Sprache verarbeiten kann. Die Zeile `\usepackage[ngerman]{babel}` sorgt dafür, dass die Silbentrennung nach den Regeln der deutschen Sprache geschieht. Die darauffolgende Zeile macht, dass $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ deutsche Umlaute und das β versteht. Die nächste Zeile sorgt dann dafür, dass auch ein entsprechender Zeichensatz verwendet wird. Das Package `microtype` hingegen ist optional – aber es lohnt sich auch in dieser einfachen Form, weil der Textsatz dadurch noch etwas schöner wird.

Um eine Aktualisierung handelt es sich bei der letzten Zeile: Diese ist dann (aber auch nur dann) erforderlich, wenn Sie das früher nicht existierende Symbol für den Euro, also €, erzeugen wollen (siehe dazu Abschnitt 6.3.4).

Packages gibt es sehr viele; aber die genannten reichen für die einfachsten Zwecke. Für mathematischen Formelsatz ist es jedoch zweckmäßig (eigentlich geradezu erforderlich), weitere Packages zu laden, siehe dazu Abschnitt 9.1). Auch zum Einbinden von Graphiken benötigt man ein eigenes Paket. Ich weise jeweils darauf hin, wenn weitere Packages benötigt werden.

6.3 Hauptelemente eines $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ Dokuments

Der eigentliche Text beginnt mit `\begin{document}` und endet (wenig überraschend) mit `\end{document}`. Der erstgenannte Befehl folgt also auf die Präambel (hier: die eben unter 6.2 genannten Befehle), oder besser gesagt: Die Präambel endet dort, wo `\begin{document}` steht. Der zweite Befehl steht ganz am Ende der Datei. Genauer gesagt: Alles, was auf diesen Befehl folgt, wird von $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ignoriert.

Innerhalb des so aufgespannten Rahmens muss nun ihr Text geschrieben werden. Dieser besteht normalerweise aus

- einem Titel und der Nennung des Autor/der Autorin (diese können entweder einfach so am Beginn des Dokuments stehen oder auf einer eigenen Titelseite);
- einzelnen Abschnitten (Kapiteln) und evtl. Unterabschnitten (Unterkapiteln);
- dem eigentlichen Text.

6.3.1 Titelseite

Die Titelseite dieses Dokuments wurde durch folgende Befehle erzeugt:

```
\title{Einfach(e) Dokumente mit \LaTeX\ erzeugen\thanks{Gegenüber  
der letzten Fassung ... leicht erweiterte Version ... }}  
\author{Wolfgang Ludwig-Mayerhofer\\Universität Siegen\\  
Philosophische Fakultät\\57068 Siegen\\  
ludwig-mayerhofer@soziologie.uni-siegen.de}  
\maketitle
```

Diese Befehle folgen auf den Befehl `\begin{document}`. (Sie können in die nächste Zeile geschrieben werden, aber Sie können genauso gut ein paar Zeilen leer lassen zwecks besserer Übersicht). Die Zeichen innerhalb der auf `\author` folgenden Klammer können fortlaufend geschrieben werden, sie sind hier nur zur Verdeutlichung auf eine größere Zahl von Zeilen verteilt worden. Die doppelten Backslashes erzeugen neue Zeilen im fertiggestellten Dokument. Sie sehen, dass die Fußnote mit dem Hinweis durch den Zusatz `\thanks{}` erzeugt wird (das gilt nur für die Fußnote zum Titel!); als Datum wird automatisch das heutige Datum eingesetzt, wenn kein Datumsbefehl angegeben wird. Soll gar kein Datum angegeben werden, fügen Sie den Angaben zum Titel noch die Zeile `\date{}` hinzu.

Wenn Sie keine eigene Titelseite möchten, lassen Sie einfach in der Präambel beim Befehl `\documentclass` die Angabe `titlepage` weg; dann werden Titel, Autor etc. einfach oben auf der ersten Seite wiedergegeben und der Text schließt sich dann unmittelbar (natürlich mit einem gewissen Abstand) an.

6.3.2 Text

Ein Text gliedert sich in Abschnitte und evtl. Unterabschnitte. Die Überschrift für einen Abschnitt wird durch den Befehl

```
\section{Hier steht die Überschrift}
```

erzeugt. Für einen Unterabschnitt lautet der Befehl

```
\subsection{Hier steht die Überschrift}
```

Wenn der Text noch tiefer gegliedert sein soll (so wie dieser), gibt es noch `\subsubsection`. Unterhalb hiervon gibt es noch zwei weitere Ebenen, nämlich `\paragraph` und `\subparagraph`; hier wird die Überschrift aber nicht freigestellt, und sie erhält auch keine Nummer.⁴ Wollen Sie, dass die Überschriften der „Sektionen“ nicht nummeriert werden, müssen Sie `\section*`

⁴ Wenn Sie noch tiefer gegliederte Texte schreiben wollen, benötigen Sie ein anderes Format als „article“. Sie sollten aber auch prüfen, ob eine so tiefe Gliederung wirklich nötig ist. Meistens ist das nicht der Fall; ist ihr Bedürfnis nach tiefer Gliederung groß, sind Sie entweder ein Jurist oder aber ein Pedant.

bzw. `\subsection*` usw. schreiben. Es können nummerierte und nicht nummerierte Überschriften kombiniert werden. Beispielsweise erhält das Literaturverzeichnis üblicherweise keine Gliederungsnummer. Man kann es also das Literaturverzeichnis wie folgt einleiten: `\section*{Literaturverzeichnis}`.⁵ (Beachten Sie: Es gibt auch eine Funktion `\appendix`, auf die ich hier nicht eingehen kann.)

Der Fließtext wird einfach nach den Überschriften geschrieben. Er bedarf keiner besonderen Kennzeichnung. *Wichtig:* Ein neuer Absatz entsteht einfach durch Einfügen einer Leerzeile.

Fußnoten werden durch den Befehl `\footnote{Hier der Text}` erzeugt. Per Voreinstellung sehen die Fußnotenziffern zu Beginn der Fußnote leider nicht ganz so aus, wie sie sollen. Solche Dinge lassen sich ändern – aber erst, wenn man tiefer in \LaTeX (und seine Grundlage, \TeX) eingestiegen ist. Einen Vorschlag dazu finden Sie weiter unten (Seite 22). Für den Einstieg können Sie aber mit der Voreinstellung arbeiten.

6.3.3 Etwas mehr über die Gestaltung von Text

Hervorhebung innerhalb des Fließtextes erfolgt durch *Kursivdruck*. Dazu schreibt man, um im Beispiel zu bleiben, `\emph{Kursivdruck}`. Die Kursive der Standardschrift (Computer Modern) ist besonders apart, nicht zuletzt die Ziffern, namentlich 2, 4, 7.

Der erste Absatz nach einer Überschrift hat keinen Erstzeileneinzug, die weiteren Absätze schon. Dafür sorgt \LaTeX automatisch. Will man den Erstzeileneinzug eines Folgeabsatzes aus bestimmten Gründen unterbinden, beginnt man den Absatz mit dem Befehl `\noindent`.

Eine Aufzählung mit „bullet points“, wie ich sie weiter oben mehrfach verwendet habe, wird wie folgt erzeugt:

```
\begin{itemize}
\item Hier steht der Text zum ersten bullet point.
\item Hier steht der Text zum zweiten bullet point.
\item usw.
\end{itemize}
```

Soll die Aufzählung statt durch bullet points durch eine Nummerierung gegliedert sein, rahmt man die Items durch die Klammern `\begin{enumerate}` und `\end{enumerate}`. Eine solche Struktur, die mit `\begin{...}` anfangen

⁵ Für den bibliographischen Apparat Ihres Textes gibt es ein eigenes Programm (BibTeX), das ebenfalls in \LaTeX integriert ist. Man kann das Literaturverzeichnis aber auch von Hand machen. Der Vorteil von BibTeX ist, dass die Verweise im Text ebenso wie das Verzeichnis einheitlich gestaltet werden, und indem man sich zwingt, alles komplett auf BibTeX einzustellen, kann man keinen Titel „vergessen“ (bzw. wird man darauf aufmerksam gemacht, wenn das geschehen ist). Aber das kann ein normaler Mensch genau so gut „einfach so“ erledigen.

und `\end{...}` abgeschlossen wird, nennt man übrigens „Umgebung“ (nicht zu verwechseln mit der eingangs angesprochenen Entwicklungsumgebung).

6.3.4 Spezielle Zeichen

Bevor wir weitermachen, möchte ich Sie noch einmal daran erinnern, dass es einige Zeichen gibt, die in \LaTeX eine spezielle Bedeutung haben. Wenn man diese Zeichen also als Textelemente verwenden möchte, muss man zu speziellen Befehlen greifen. Tabelle 1 zeigt links das Zeichen, das angezeigt werden soll, und rechts den Befehl, der das bewirkt.

Tabelle 1: \LaTeX -Befehle für besondere Zeichen

Zeichen	\LaTeX -Code
§	<code>\S</code>
\$	<code>\\$</code>
%	<code>\%</code>
&	<code>\&</code>
{	<code>\{</code>
}	<code>\}</code>
\	<code>\textbackslash</code>
#	<code>\#</code>
~	<code>\textasciitilde</code>
^	<code>\textasciicircum</code>
_	<code>_</code>

Bitte beachten Sie: Diese, ebenso wie viele andere \LaTeX -Befehle, enden einfach durch ein Leerzeichen. Mit anderen Worten: Ein Leerzeichen, das auf diese Befehle folgt, wird nicht als Leerzeichen, sondern „nur“ als Befehlsende interpretiert. Wenn tatsächlich ein Leerzeichen folgen soll, müssen Sie diese Befehle entweder mit einem backslash (also mit `\`) oder mit zwei geschweiften Klammern (also mit `{}`) abschließen. Dann wird das nächste Leerzeichen als Leerzeichen wiedergegeben. (Es verfinde nicht, einfach zwei Leerzeichen zu schreiben – \LaTeX kennt nämlich die Unart menschlicher Wesen, zu viele Leerzeichen einzugeben, und macht aus mehreren Leerzeichen immer eines!).

Ein spezielles Problem ist das Euro-Zeichen €. Ich habe weiter oben (Abschnitt 6.2) bereits den Befehl `\usepackage{eurosym}` für die Präambel angeführt. Dieses Paket sollte in den meisten aktuellen Distributionen vorhanden sein. (Wenn nicht, erhalten Sie beim Kompilieren des Dokuments eine Fehlermeldung). Dann können Sie im Text u. a. die in Tabelle 2 aufgeführten Befehle verwenden. Der dort gezeigte zweite Befehl – hier *exemplarisch* für einen Preis von 10 € verwendet, Sie können natürlich jede andere Zahl einsetzen.

zen – sorgt dafür, dass zwischen die Zahl und das Euro-Zeichen automatisch der korrekte etwas kleinere Abstand (vgl. Abschnitt 6.4.7) gesetzt wird.

Tabelle 2: Das Euro-Zeichen in \LaTeX

Zeichen	\LaTeX -Code
€	<code>\euro</code>
10 €	<code>\EUR{10}</code>

Sollte das Paket „Eurosymb“ tatsächlich nicht installiert sein, können Sie es mal mit dem Befehl `\texteuro` versuchen. Aber Paket `T1` (das aus anderen Gründen sinnvoll ist) versteht diesen Befehl nicht.

6.3.5 Silbentrennung

Die Silbentrennung wird von \LaTeX automatisch durchgeführt, es bedarf also keines eigenen Befehls dazu. Es wird also die heute weit verbreitete Unart, Texte mit Blocksatz, aber ohne Silbentrennung – mit der Folge leseunfreundlicher und abgrundtief hässlicher Lücken zwischen den Worten – zu publizieren, von vornherein unterbunden.

Manchmal fällt es \LaTeX schwer, Worte zu trennen. Es erscheinen dann bei der Erzeugung der PDF-Datei im Log-Fenster entsprechende Meldungen, oder man sieht es beim Betrachten der PDF-Datei. Auch teilt \LaTeX mit Microsoft Word die Unart, Worte, die einen Trennungsstrich enthalten, nur beim Trennungsstrich zu trennen. Hier ist es zweckmäßig, bedingte Trennstriche einzufügen. Hat \LaTeX etwa Probleme mit dem Wort Zeichenkombination, kann man schreiben: `Zei\~chen\~kom\~bi\~na\~tion`. Es wird dann die beste Stelle für die Trennung gesucht. Man kann auch nur einige dieser bedingten Trennstriche eingeben; dann kann das Wort nur an diesen Stellen getrennt werden. Bei längeren Texten könnte es sich empfehlen, solche Trennungshinweise ein für allemal für das ganze Dokument anzugeben. Dazu schreibt man noch *vor* `\begin{document}` beispielsweise: `\hyphenation{Zei-chen-kom-bi-na-tion}` (also Trennungsstriche ohne Backslashes!). Diese Liste kann erweitert werden, allerdings nicht grenzenlos (habe die Grenzen gerade nicht im Kopf).

Kann oder soll ein Wort, welches über den Seitenrand hinaus steht, nicht getrennt werden, schreibt man unmittelbar vor dieses Wort `\linebreak[4]`. Dann wird das Wort in die nächste Zeile verschoben. In der Zeile zuvor wird ein Randausgleich durchgeführt.

6.3.6 Fester Zeilen- oder Seitenumbruch

In der Regel werden Zeilen- und Seitenumbruch automatisch erledigt. In bestimmten Fällen will man aber vielleicht eingreifen.

Eine *feste neue Zeile* (innerhalb eines Absatzes – also nicht, wenn ein neuere Absatz beginnen soll), erhält man durch die Befehle `\` oder `\newline`. In der Regel ist ersterer empfehlenswert; es gibt auch einige Optionen dafür, die man leicht mit Hilfe einer Suchmaschine ermitteln kann.

Eine *feste neue Seite* erzeugt man hingegen durch `\newpage`. In Verbindung mit Graphiken oder anderen Gleitobjekten (siehe S. 20) kann auch der Befehl `\clearpage` nötig werden; suchen Sie danach, wenn Ihre Tabellen und/oder Graphiken partout nicht dort auftauchen, wo sie benötigt werden. (Besser ist es nach meiner Erfahrung allerdings, in solchen Fällen am Text selbst zu arbeiten, sprich, diesen in geeigneter Weise zu kürzen, umzustellen oder was auch immer.)

6.4 Besonderheiten der deutschen Sprache und Feinheiten des Satzes

6.4.1 Umlaute und ß

Umlaute und das sz („scharfes S“, „ß“) können problemlos verwendet werden, wenn Sie (wie empfohlen) die Befehle `\usepackage[ansinew]{inputenc}` und `\usepackage[T1]{fontenc}` in die Präambel aufnehmen.

Alternativ kann man die Umlaute der deutschen Sprache durch `\"A`, `\"a` usw. erzeugen, ein ß durch `\"s`. Viele Entwicklungsumgebungen unterstützen nach entsprechender Einstellung die automatische Erzeugung dieser Zeichen, wenn man auf der Tastatur die Umlaute bzw. das sz eingibt. Der Vorteil ist, dass Ihr Dokument dann auch auf anderen Betriebssystemen lesbar ist und somit einer der großen Vorzüge von \LaTeX , die grundsätzliche Plattformunabhängigkeit, erhalten bleibt. Allerdings wird der (Roh-)Text dadurch auch schwerer lesbar.

6.4.2 Anführungszeichen

Für richtige Anführungszeichen (in der deutschen Sprache) gibt es mehrere Möglichkeiten.

Hat man in der Präambel die von mir empfohlenen Pakete aufgerufen, kann man doppelte Anführungen einfach durch `"`` und `"'` erzeugen. (Beachten Sie: `'` ist das Apostroph-Zeichen, das auf der Tastatur rechts neben dem `Ä` liegt [ebenfalls in Verbindung mit der Umschalttaste]). Ihre Entwicklungsumgebung erlaubt möglicherweise auch, dass die Befehle für die Anführungszeichen automatisch beim Drücken von `"` erzeugt werden. Einfache Anführungszeichen erhält man durch die Zeichen `,` (also ein Komma) für das öffnende und ``` (also den Gravis) für das schließende Anführungszeichen.

Alternativ kann man für doppelte Anführungen die Befehle `\glqq` und `\grqq` und für einfache Anführungen `\glq` und `\grq`. Dabei ist zu beachten: Folgt auf das schließende Anführungszeichen ein Leerzeichen, so wird dieses als Ende des Befehls `\grqq` bzw. `\grq` interpretiert. Also muss man das Ende des Befehls durch einen nachgestellten Backslash anzeigen, wenn auch im Text ein Leerzeichen erscheinen soll. Dies gilt aber nicht vor Satzzeichen, denn diese folgen ja unmittelbar auf das Zeichen. Um also „Anführungszeichen“ zu schreiben, gibt man ein `\glqq Anführungszeichen\grqq\`, sofern – wie hier – kein Satzzeichen folgt. Ist letzteres der Fall, schreibt man `\glqq Anführungszeichen\grqq`. Das Satzzeichen kann sich ohne Zwischenraum an den Befehl `\grqq` anschließen.

Will man doppelte und einfache Anführungszeichen kombinieren (das kann z. B. erforderlich sein, wenn man etwas zitieren möchte, was im Original in Anführungsstrichen steht), dann kommt einfach nach jeden Befehl ein Leerzeichen.

6.4.3 Gedankenstriche

Gedankenstriche erzeugt man durch zwei Bindestriche hintereinander, also so: --.

6.4.4 Besondere Probleme von Trennungsstrichen

Fallkonstellation 1: Manchmal will man einem Wort noch einen Zusatz voransetzen, der aber nicht grundsätzlich gelten soll; etwa wenn man (Teil-) Privatisierung schreibt, um zum Ausdruck zu bringen, dass es in dem fraglichen Bereich gelegentlich zu vollständiger, manchmal aber eben auch zu nur Teilaspekten umfassender Privatisierung gekommen ist. Es handelt sich hier also um einen Bindestrich, der aber nicht auch als Trennungsstrich fungieren soll. Schreibt man aber einfach einen solchen Strich, kann es sein, dass am Zeilenende Teil- stehen bleibt und die nächste Zeile mit)Privatisierung beginnt. In diesem Fall kann man den Bindestrich als Zeichenfolge (normale) doppelte Anführungszeichen plus Tilde schreiben, also "~". Hinzu kommt noch ein zweites Problem: Es könnte sein, dass sich gerade nach (Teil-) ein Zeilenumbruch anbietet; in diesem Fall sollte aber auf die Klammer kein zweiter Trennungsstrich folgen. Die Trennung eines Wortes am Zeilenende ohne Trennungsstrich erzeugt man durch zwei aufeinanderfolgende doppelte Hochkommata, also durch "". Das vollständige Beispiel lautet also (Teil"~)""Privatisierung.

Fallkonstellation 2: Eine ähnliche Konstellation hat man, wenn ein Trennungsstrich am Anfang eines Wortes steht, etwa wenn man schreiben will: Trennungsstriche und -hilfen. Auch hier kann bei Bedarf die eben genannte Tastenkombination Abhilfe schaffen; im Beispiel also: "~hilfen.

6.4.5 Zitate freistellen

Längere Zitate soll man üblicherweise im Text freistellen. Dies geschieht, indem man sie mit `\begin{quote}` und `\end{quote}` rahmt.

6.4.6 Den Schriftgrad wechseln

Manchmal will man einzelnen Absätze oder sonstwie definierte Textteile in kleinerer oder größerer Schrift setzen. Einige Befehle zum Verkleinern sind `\small` und `\footnotesize` (letzterer Befehl führt zu einer stärkeren Verkleinerung), zum Vergrößern kann man etwa `\large`, `\Large` und `\LARGE` verwenden. Grundsätzlich sollte man mit diesem Gestaltungsmittel in wissenschaftlichen Texten zurückhaltend sein; ich selbst gebrauche allenfalls ab und zu `\small`, um weniger Wichtiges kenntlich zu machen, oder damit Tabellen weniger aufdringlich/raumgreifend sind.

Der gewünschte Befehl steht einfach an der Stelle, ab der die geänderte Schriftgröße gelten soll. Zurück zum voreingestellten Schriftgrad wechselt man, indem man an der betreffenden Stelle den Befehl `\normalsize` einfügt. Betreffen die Änderungen nur kurze Passagen, schreibt man etwa wie folgt: `{\small Kleiner Text}`. Dann gilt die Änderung des Schriftgrades nur innerhalb der geschweiften Klammer.

6.4.7 Zwischenräume/Abstände

Zwischenräume zwischen Absätzen: Will man einen Absatz durch etwas Leerraum vom nächsten Absatz abheben, kann man am Ende dieses Absatzes je nach Wunsch `\smallskip`, `\medskip` oder `\bigskip` einfügen. Die exakte Größe wird dann vom Programm festgelegt. Exakte Abstände kann man definieren durch `\vspace{Xcm}`. Für X ist die gewünschte Zahl einzusetzen; man beachte, dass als Dezimalzeichen der Punkt verwendet werden muss.

Zwischenräume zwischen einzelnen Zeichen: Zwischen die Bestandteile von Datumsangaben, Abkürzungen mehrerer Worte u. a. m. kommt ein kleiner Zwischenraum. Diesen erzeugt man durch die Zeichenfolge `\.`. Vergleiche (ich beginne jeweils mit der Version ohne Zwischenräume):

d.h. versus d. h.

30.11.2007 versus 30. 11. 2007

Ein zusätzlicher Vorteil von `\.` ist, dass die entsprechenden Bestandteile immer zusammengehalten werden, es kann also nicht passieren, dass bei d. h. das d. am Ende der Zeile steht und das h. am Beginn der nächsten Zeile, oder dass Daten oder Zahlen auf zwei Zeilen verteilt werden.

Man beachte: Es sieht auch gut aus, wenn man sehr große Zahlen (fünf Stellen und mehr) durch einen oder mehrere Zwischenräume unterteilt. Hierfür sollte man das Paket und den Befehl `\numprint` verwenden (siehe dazu Abschnitt 9.3 auf S. 33). Ein Beispiel: 10 675 300.

Es gibt zahlreiche weitere Möglichkeiten, die Verteilung von Zeichen, Zeilen und Absätzen zu regeln, nicht zuletzt auch Befehle für den Seitenumbruch, auf die hier nicht eingegangen werden kann.

6.4.8 Mediävalziffern

Mediävalziffern sind die Ziffern, die ich in diesem Text fast durchgängig verwende (außer bei vereinzelt mathematischen Beispielen). Sie sind uns heute im Alltag fast nicht mehr geläufig, doch sind sich Typographen einig, dass sie zum anspruchsvollen Textsatz dazugehören.⁶ Der Unterschied zu den üblichen Ziffern – meist Versalziffern genannt – zeigt sich am besten im Beispiel: 1 234 567 890 versus 1 234 567 890. Die Versalziffern haben ausschließlich Oberlängen, keine Unterlängen, genau wie Großbuchstaben (daher auch ihr Name). Die Mediävalziffern haben dagegen teilweise Unter- und nur vereinzelt Oberlängen (gelegentlich auch weder – noch), genau wie Kleinbuchstaben (daher heißen die Mediävalziffern manchmal auch Minuskelziffern).

Über die genaue Verwendung von Mediävalziffern kann man sicher diskutieren. Beispielweise mag man geteilter Meinung darüber sein, ob die Abschnittsnummern nicht besser in Versalziffern gehalten sein sollten. Egal. Hier geht es nur um die Frage, wie man Mediävalziffern erhält. Denn auch in L^AT_EX sind sie nicht standardmäßig vorgesehen. Tatsächlich sind sie auch nicht in allen Fonts verfügbar; die Times z. B. enthält, soweit ich weiß, keine.

Grob gesagt gibt es zwei Wege, Mediävalziffern zu erhalten:

Einstellung für das ganze Dokument: Will man Mediävalziffern als „Grundeinstellung“ haben, kann man für einige Schriften Packages laden. Für die (auch in diesem Dokument verwendete) Standardschrift, die Modern, ist das das Paket `crf-lm` (Befehl `\usepackage{crf-lm}` in die Präambel aufnehmen).⁷ Für die Palatino fügt man dem Aufruf dieser Schrift einfach die Option `[osf]` hinzu. Auch für die Linux Libertine existiert diese Option.

Die Darstellung von Zahlen im (in diesem Dokument nicht erläuterten!) Mathematik-Modus wird hiervon nicht berührt.

Will man dann ausnahmsweise doch einmal Versalziffern verwenden, kann man die Befehle `\lstyle{}` oder `\textl{}` verwenden; die Ziffern stehen dann in den geschweiften Klammern.

Einstellung für einzelne Ziffern(folgen): Will man Versalziffern als Voreinstellung haben, aber punktuell Mediävalziffern verwenden, kann man den Befehl `\oldstylenums{}` verwenden (`old-style numerals` oder `figures` ist

⁶ Ich nutze die Gelegenheit darauf hinzuweisen, dass ich Vieles von meinem Wissen über Typographie Alexandra Wicht verdanke. So kümmerlich und dilettantisch mein Umgang mit Textsatz und Typographie auch sein mag, für ihre zahlreichen geduldigen Belehrungen bin ich ihr zutiefst dankbar, gar nicht zu reden von einschlägiger Literatur, von der Robert Bringhursts „Elements of Typographic Style“ besondere Erwähnung verdienen.

⁷ Hinweis für Experten: Tatsächlich bewirkt dieser Befehl, dass eine Variante der Standardschrift verwendet wird: Statt der Computer Modern wird die Latin Modern eingesetzt. Auch wenn die Bildschirmdarstellung bei letzterer schlechter ist, sind jedenfalls auf dem Papier kaum Unterschiede zu bemerken.

ein gängiger englischer Ausdruck für Mediävalziffern; alternativ können sie auch text figures heißen).

6.4.9 Text zusammenhalten

Manchmal sollen zwei Einheiten – meist nicht zwei Worte, sondern ein Wort und ein Zusatz – zusammen gehalten werden, d. h. es soll nicht ein Bestandteil am Zeilenende und der andere am Anfang der nächsten Zeile stehen. Beispiele sind: Seitenangaben: S. 4; Gesetzbücher, so wie etwa SGB II (Sozialgesetzbuch II); Gesetzesabschnitte, etwa § 17 StGB. Hierfür verbindet man die beiden Bestandteile mit einer Tilde ~. Beachten Sie, dass hier meist *kein* verkleinerter Wortzwischenraum gewählt wird (siehe den vorstehenden Abschnitt; sollte doch ein solcher kleiner Zwischenraum gewünscht werden, genügt das entsprechende Zeichen).

Alternativ hilft der Befehl `\nolinebreak`. Wir schreiben also beispielsweise `\SGB \nolinebreak II`.

6.4.10 Punkte zur Signalisierung von Auslassungen

Für Auslassungen verwendet man gerne drei Punkte; denken wir etwa an Kleists berühmte „Marquise von O...“.⁸ Diese Punkte sehen ansprechend aus, wenn man sie durch den Befehl `\dots` erzeugt.

7 Mehr

7.1 Kommentare

Wenn das Zeichen `%` auftaucht, wird alles, was bis zum nächsten festen Zeilenumbruch auf dieses Zeichen folgt, von \LaTeX ignoriert. Das ist hilfreich für eigene Anmerkungen im Text. So enthält eine meiner Dateien folgenden Text in der Präambel:

```
\usepackage[normalem]{ulem} % für Durchstreichen
% Fundstelle: http://www.matheboard.de/archive/31629/thread.html
% Durchstreichen mit \sout{Hier steht das Durchzustreichende}
```

Anders als beispielsweise in gängiger WYSIWIG-Software kann man also für sich selbst die Ausgangsdatei (wenn gewünscht oder erforderlich, sehr ausführlich) kommentieren und gleichzeitig für die Mitwelt ein makellooses (unkommentiertes) PDF-Dokument erstellen. Die Kehrseite ist freilich, dass Formen des gemeinsamen Arbeitens an einem Dokument nur eingeschränkt zu realisieren sind (Kommentare lassen sich nicht so leicht bestimmen

⁸ Im Erstdruck ist die Ellipse allerdings durch vier Punkte wiedergegeben. Das lässt sich, soweit ich sehe, mit \LaTeX nicht automatisch bewerkstelligen, da müsste man wohl ein wenig herumbasteln.

Textelementen und vor allem bestimmten Autoren zuordnen; vor allem gibt es keine Funktion „Änderungen Nachverfolgen“, außer mit einem einem Zusatzpaket, welches das Arbeiten aber wiederum recht schwierig macht).

7.2 Graphiken

Graphiken können eine komplizierte Angelegenheit sein, aber solange man als Output eine PDF-Datei erzeugen möchte und keine gestalterischen Tricks braucht, ist die Sache zum Glück ziemlich einfach. Wichtig ist daran zu denken, dass nur wenige Graphik-Formate eingebunden werden können; darunter sind aber weit verbreitete Formate wie JPEG, PNG oder auch PDF.

Zunächst ist es erforderlich, der Präambel folgenden Befehl hinzuzufügen: `\usepackage{graphicx}`. Die Befehle zum Einfügen der Graphiken folgen dann dort, wo die jeweilige Graphik stehen soll. Dabei kann man zwei Konstellationen unterscheiden:

Fall 1: Die Graphik-Datei wird in den laufenden Text eingebunden, wie hier zu sehen:



Portrait of the Author as a Medium-aged Man

Die Befehle dazu lauten:

```
\begin{center}
\includegraphics{ludwigmayerhofer.jpg} \\
Portrait of the Author as a Medium-aged Man
\end{center}
```

Fall 2: In der Regel ist es besser, Graphiken als Gleitobjekte zu definieren, die von \LaTeX dort hingestellt werden, wo es am besten passt. Hierfür verwendet man die `\begin{figure}-\end{figure}`-Umgebung. Abbildung 1 habe ich erzeugt durch:

```

\begin{figure}
\centering
\includegraphics{ludwigmayerhofer.jpg}
\caption{Portrait of the Author as a Medium-aged Man
-- hier als gleitende Graphik}
\label{fig:Portrait}
\end{figure}

```



Abbildung 1: Portrait of the Author as a Medium-aged Man – hier als gleitende Graphik

Wichtig ist, dass man in diesem Fall auf die Graphik nicht etwa mit „die folgende Graphik“ verweist, denn die Graphik steht vielleicht am Beginn der Seite oder auch am Beginn der nächsten Seite. Man verweist also, wie oben geschehen, mit einer Nummer auf die Graphik. Damit dies möglich ist, erhält die Graphik ein Label durch den Befehl `\label{...}`; das Label muss nach `\caption` stehen, denn durch letzteren Befehl wird die Abbildung erst mit einer Nummer versehen. Das Verweisen auf die Graphik anhand ihres Labels wird in Abschnitt 7.4 erläutert.

Zum Befehl `\includegraphics` gibt es einige Optionen, die in eckigen Klammer vor die geschweifte Klammer mit dem Dateinamen geschrieben werden können. So kann man beispielsweise mit `[scale=1.2]` dafür sorgen, dass die Graphik um den Faktor 1,2 vergrößert wird. Man kann die Größe der Graphik auch direkt durch Angabe einer spezifischen Höhe oder Breite beeinflussen (das jeweils andere Element wird dann automatisch angepasst), also etwa durch `[height=3cm]`. Durch `[trim = 1 u r o, clip]` kann man veranlassen, dass links, unten, rechts und/oder oben etwas von der Originalgraphik abgeschnitten wird. `[trim = 1cm 0cm 0cm 0cm, clip]` bewirkt z. B., dass links ein Zentimeter abgeschnitten wird.

Die Optionen können kombiniert werden, wie dies im letzten Beispiel

schon geschah; `clip` ist eine Option, die man mit `trim` zusammen verwenden muss, damit `trim` wirksam wird. Sie werden also, getrennt durch Kommata, aneinandergereiht. Dabei ist auf die Reihenfolge zu achten; es macht einen Unterschied, ob man eine Graphik erst verkleinert und dann einen Rand (oder mehrere) abschneidet oder umgekehrt.

Mehr zu Graphiken findet man in einem ausführlichen Text von Keith Reckdahl unter <ftp://dante.ctan.org/tex-archive/info/epslatex.pdf> (ist auch als Postscript-Datei mit der Extension `.ps` verfügbar).

7.3 Inhaltsverzeichnis

Man fügt einfach ein: `\tableofcontents`, und zwar an der Stelle, an der das Inhaltsverzeichnis stehen soll. Da das Inhaltsverzeichnis in der Regel am Anfang steht, kann \LaTeX zunächst nur einen Platzhalter für das Inhaltsverzeichnis erstellen. Erst in einem zweiten Durchlauf (also: indem man einfach nochmal die PDF-Datei erzeugt) entsteht dann das Inhaltsverzeichnis.

7.4 Verweise im Text

Will man auf ein Kapitel bzw. einen Abschnitt oder Unterabschnitt verweisen, muss man diesen durch `\label{BeliebigesEtikett}` markieren (an der Stelle von `BeliebigesEtikett` sollte eine von Ihnen gewählte Zeichenfolge stehen). Dieser `label`-Befehl kann irgendwo in dem betreffenden Kapitel (Abschnitt, Unterabschnitt) stehen; es ist aber sicherlich nicht unzweckmäßig, ihn gleich auf den betreffenden `section`-Befehl (oder dgl.) folgen zu lassen. Der Verweis auf das betreffende Kapitel etc. wird an der Verweisstelle durch `\ref{BeliebigesEtikett}` erzeugt; an der Stelle von `\ref{BeliebigesEtikett}` erscheint dann die Kapitel- bzw. Abschnittsnummer. Wenn Sie dagegen `\pageref{BeliebigesEtikett}` verwenden, wird an der betreffenden Stelle die Seitennummer ausgegeben, auf der das Label steht.

Label können übrigens auch für Formeln (hier nicht erläutert), Tabellen (Abschnitt 7.6) oder Graphiken (Abschnitt 7.2) vergeben werden; an der Stelle von `\ref{BeliebigesEtikett}` erscheint dann die Nummer der Formel, Tabelle oder Graphik.

7.5 Fußnoten

Fußnoten, die entsprechend den in Deutschland üblichen typographischen Vorstellungen gestaltet sind, sind leider nicht ganz einfach zu erhalten. Auch in diesem Text sind sie nicht perfekt, denn die Fußnotenziffern sollten in den Fußnoten selbst (im Gegensatz zum Fließtext) nicht hochgestellt sein. Das lässt sich zwar im Prinzip ebenfalls bewerkstelligen, aber leider verwende ich hier ein Package, das dann zu unerwünschten Folgewirkungen führt. Ein Experte könnte diese bestimmt wiederum unterdrücken, aber mir fehlen dazu die Kenntnisse.

Zunächst also: Was habe ich gemacht? Ich verwende ganz einfach das Package `footmisc` mit der Option `hang`. Das bewirkt, dass die Fußnotenziffer links am Seitenrand steht und die Fußnote eingerückt ist (in der Voreinstellung ist die erste Zeile der Fußnote mitsamt der Fußnotenziffer eingezogen, der Rest nicht). Dazu verwende ich einen weiteren Befehl, der den Abstand zwischen Fußnotenziffer und Fußnotentext gegenüber der Voreinstellung des Pakets etwas verringert. Zusammen sieht das ganz einfach so aus (beide Zeilen stehen in der Präambel!):

```
\usepackage[hang]{footmisc}
\setlength{\footnotemargin}{1em}
```

Ist Ihnen daran gelegen, dass die Fußnotenziffern in den Fußnoten tatsächlich nicht hochgestellt sind, müssen Sie etwas im Internet recherchieren und herumprobieren. So habe ich auf Stackexchange unter der Adresse <http://tex.stackexchange.com/questions/19844/> eine Befehlssequenz gefunden, die einwandfrei funktioniert – allerdings nicht zusammen mit Mediävalziffern (siehe S. 18). Oder besser gesagt: Sie funktioniert, aber die Fußnotenziffern werden dann eben auch als Mediävalziffern gesetzt, was in meinen Augen nicht gut aussieht. – Sicherlich gibt es noch weitere Möglichkeiten, von denen die eine oder andere dieses Problem nicht hat, oder es ließe sich weiterer Code schreiben, der die Verwendung der Mediävalziffern in den Fußnoten unterdrückt, aber bei aller Liebe zu L^AT_EX: Da kümmere ich mich ein andermal drum.

7.6 Tabellen

Tabellen können ziemlich komplex sein; ich behandle hier nur ganz einfache Fälle für kurze Tabellen, die Text enthalten.

Für diesen Fall kann man die `tabular`-Umgebung verwenden. Innerhalb der Tabelle werden die verschiedenen Spalten durch das Zeichen `&` und die Zeilen durch die Zeichenfolge `\\` gekennzeichnet. Eine kleine Tabelle kann beispielsweise so aussehen:

```
ABM   Arbeitsbeschaffungsmaßnahme(n)
AFG   Arbeitsförderungsgesetz
AGHs  Arbeitsgelegenheiten
```

Die Befehle hierfür lauten:

```
\begin{tabular}{@{} l p{10.5cm}}
ABM & Arbeitsbeschaffungsmaßnahme(n) \\
AFG & Arbeitsförderungsgesetz \\
AGHs & Arbeitsgelegenheiten \\
\end{tabular}
```


`\begin{tabular}` leitet die Tabelle ein. Die Befehle innerhalb der zweiten geschweiften Klammer bedeuten Folgendes: `@{}` sorgt dafür, dass die üblicherweise automatisch gesetzte Einrückung am Beginn jeder Zeile verringert wird (soll die Tabelle überhaupt nicht eingerückt werden, muss vor `\begin{tabular}` noch ein `\noindent` gesetzt werden). `l` (der Buchstabe `l`) besagt, dass als nächstes eine Spalte folgt, deren Inhalt linksbündig angeordnet werden soll; `p{10.5cm}` schließlich definiert eine zweite Spalte von 10,5 cm Breite. Diese Variante, eine Spalte zu definieren, muss dann verwendet werden, wenn eine Spalte Text enthält, der sich über mehr als eine Zeile erstreckt. Definiert man Spalten nur mit dem Befehl `l`, wird längerer Text nicht umbrochen und reicht gegebenenfalls über den Rand hinaus. (Im vorliegenden Fall hätten freilich zwei `l` genügt, denn die rechte Spalte enthält nur kurzen Text.) Im Beispiel haben wir also zwei Spalten; daher muss in jeder Zeile des Tabelleninhalts ein `&` stehen, welches die beiden Spalten voneinander trennt. Die Breite der linken Spalte wird von \LaTeX automatisch bestimmt.

Eine so erzeugte Tabelle erscheint einfach an der Stelle, an der sie in Ihrem „Rohtext“ steht, und wird wie Fließtext behandelt. Daher kann in der Tabelle beispielsweise ein Seitenumbruch vorgenommen werden. Um das zu vermeiden, sollten Sie die Tabelle in eine Umgebung einschließen; diese lautet `\begin{table} ... \end{table}`. Die Verwendung dieser Umgebung hat einige weitere Vorteile:

- Sie können (und sollten) \LaTeX einen Tipp geben, wo die Tabelle vorzugsweise stehen sollte: Möglichst an der gleichen Stelle wie in ihrem Rohtext, oder am Seitenbeginn, oder am Seitenende, oder gar auf einer eigenen Seite. Sie können auch mehrere Positionswünsche angeben; \LaTeX versucht es dann mit ihrem ersten Wunsch, ist dieser nicht realisierbar, mit dem zweiten, usw. Eine vollständige Liste könnte beispielsweise so aussehen: `\begin{table}[htbp]`. Der Positionswunsch folgt also in eckigen Klammern. Dabei steht `h` für „here“, also „hier“, `t` für „top“, also den Seitenbeginn, `b` für „bottom“, also das Seitenende, und `p` für „page“, also eine eigene Seite. Sie können die Option aber auch weglassen; es wird dann die Voreinstellung `\begin{table}[tbp]` verwendet.
- Sie können (und sollten) die Tabellenüberschrift in der Tabelle angeben. Hierfür verwendet man `\caption{Tabellenüberschrift}`. Die Überschrift wird da platziert, wo Sie sie hinstellen, also über der Tabelle, wenn `\caption{...}` vor `\begin{tabular}`, und unter der Tabelle, wenn es nach `\end{tabular}` steht. Die Tabellen werden übrigens automatisch nummeriert.
- Eine Tabelle kann auch ein Label erhalten, so dass an anderer Stelle auf die Tabelle verwiesen werden kann. Genaueres dazu in Abschnitt 7.4

- Wichtig: Da die Tabellennummer erst durch `\caption` erzeugt wird, muss der Befehl `\label` unbedingt nach `\caption` stehen!

7.7 Textboxen (umrahmter Text)

Textboxen können geeignet sein, um beispielsweise kurze Zusammenfassungen zu bringen oder sonst einen Textblock besonders hervor- oder herauszuheben. Auch hierzu bringe ich nur ganz elementare Dinge; die Befehle lassen sich deutlich erweitern (komplizierter machen).

Ein einfaches Beispiel:

Welche Elemente benötige ich für eine Box?

- Eine längere Textbox erzeugt man wohl am besten mit der Minipage-Umgebung.
- Damit diese Box einen Rahmen erhält, muss man die Minipage-Umgebung noch in einen `fbox`-Befehl einbetten.

Hier kommt weiterer Text.

Statt der Minipage-Umgebung könnte man auch den Befehl `\parbox` verwenden, jedoch lassen sich viele Befehle nur innerhalb der Minipage-Umgebung verwenden (z. B. die Aufzählung mit Listenelementen).

Wie geht es aber nun im einzelnen? Hier sind die wesentlichen Elemente, die ich eben verwendet habe:

```
\fbox{
\begin{minipage}{11cm} Hier folgt folgt Text Text Text Text.
Innerhalb von minipage kann man weitere Gestaltungselemente wie
die \itemize-Umgebung verwenden (mit wenigen Einschränkungen).
\end{minipage}
}
```

`\fbox{}` sorgt nur dafür, dass der Rahmen gezeichnet wird, sonst hat es keine Funktion (man beachte die abschließende geschweifte Klammer in der letzten Zeile, die die `fbox` schließt). Die Angabe `{11cm}` direkt nach dem Befehl `\begin{minipage}` definiert die Breite der Box. Dann kann einfach beliebiger Text folgen. Um zwischen der Box und dem sie umgebenden Text etwas Abstand zu haben, habe ich hier den `\medskip`-Befehl verwendet; vielleicht gibt es elegantere Möglichkeiten, ich habe noch nicht danach gesucht.

8 Gesamtdokument, Seitenlayout

Wir haben jetzt die wichtigsten Dinge besprochen, mit denen man erfolgreich ein Dokument mit \LaTeX schreiben kann. Vorausgesetzt wurde dabei immer, dass es sich um ein Dokument handelt, das man sinnvollerweise mit der

Dokumentenklasse `article` bewältigen kann, etwa eine Seminararbeit oder vergleichbare Dokumente. Sie werden sich vielleicht gefragt haben: Aber wie kann ich das Dokument nun eigentlich gestalten, also die Schrift einstellen, die Seitenränder festlegen etc.?

Dazu ist zunächst einmal grundsätzlich zu sagen, dass der Zweck von \LaTeX gerade ist, Sie von solchen Fragen zu entlasten: Sie schreiben den Text, \LaTeX (bzw. \TeX) kümmert sich um das Aussehen! Dahinter steht der Anspruch, dass \LaTeX und \TeX (genauer: ihre Erfinder) wesentlich mehr als gewöhnliche Nutzer von Textsatz verstehen. Und das ist in aller Regel auch richtig. Für gut lesbare Dokumente ist wichtig, dass die Gliederung gut erkennbar ist, dass die Zeilen nicht zu lang sind, dass man gut von einer Zeile zur nächsten kommt, dass eine lesbare Schrift verwendet wird, dass der Text einen halbwegs gleichmäßigen Grauwert hat (sprich, dass die in vielen Texten zu findenden hässlichen Lücken zwischen den Worten vermieden werden), um nur die wichtigsten Punkte zu nennen.

Sie haben ja bereits bemerkt, dass der vorliegende Text anders aussieht als die Texte, die heute gängige Textverarbeitungssoftware üblicherweise produziert. Das liegt einmal an der Schrift, aber vor allem auch daran, dass die Zeilen deutlich kürzer sind, mit der Konsequenz, dass der linke und rechte Seitenrand breiter sind als üblich. Damit der Text ansprechend aussieht, sind auch die Seitenränder oben und unten verbreitert worden. Wie gesagt, alles per Voreinstellung, ich musste mich um rein gar nichts kümmern! Beachten Sie, dass auch der Zeilenabstand nicht besonders groß ist – das liest sich einfach besser als ein zu großer Abstand, weil das Auge nicht zu weit springen muss.

Soweit, so gut. In der Praxis kommt es nun immer wieder vor, dass man einfach andere Vorgaben einhalten *muss*. Beispiele könnten studentische Seminar- oder Abschlussarbeiten sein: Dort wird häufig ein bestimmter Seitenrand vorgegeben (etwa: links und rechts jeweils 3cm), meist wird auch erwartet, dass der obere und untere Seitenrand ähnlich groß bzw. klein sind. Zudem wird hier häufig ein 1,5-facher Zeilenabstand erwartet.

Nun, in einer perfekten Welt würden Sie zu Ihrer Dozentin oder Betreuerin (oder dem männlichen Pendant) gehen und ihr erklären, dass Texte, die nach diesen Regeln produziert sind, schwer zu lesen sind und scheußlich aussehen – und die Dozentin würde sagen: Selbstverständlich, Sie haben Recht, verwenden Sie einfach die Voreinstellungen von \LaTeX . Leider leben wir aber, wie Sie wissen, nicht in einer perfekten Welt. Sie können Ihre Dozentin natürlich fragen – aber wenn Sie in der Welt der Sozial- oder Geisteswissenschaften leben, können Sie ziemlich sicher sein, dass Ihre Dozentin von \LaTeX nie etwas gehört hat – und von der Lesefreundlichkeit von Texten auch nicht. Und außerdem wird sie oder er in der Regel keine Lust haben, nur wegen eines/einer Studierenden Regeln zu ändern.

Nachfolgend also etwas mehr dazu, wie man nicht nur schreiben, sondern auch das Aussehen (mit Betonung des Seitenlayouts) beeinflussen kann.

8.1 Andere Dokumentklassen

Bislang bin ich von der Dokumentklasse `article` ausgegangen (siehe Abschnitt 6.2). Werfen wir einen kurzen Blick auf andere Klassen.

Wichtige Alternativen zu `article` sind `book` und `report`. Die Hauptunterschiede sind folgende: (1.) Im Unterschied zu `article` haben beide als höchste Gliederungseinheit Kapitel, in \LaTeX : `\chapter`. Erst darunter kommen dann `\section` usw. (2.) Die Klasse `book` geht davon aus, dass es sich wirklich um ein Buch handelt, dass also der Text auf Vorder- und Rückseite gedruckt wird. Deswegen entspricht die Breite des linken Randes einer ungeraden Seite der des rechten Randes einer geraden Seite und umgekehrt. Kapitel (also `chapter`) beginnen außerdem immer auf einer ungeraden Seite. Bei `report` ist beides nicht der Fall; die Kapitel beginnen aber auf einer neuen Seite, anders als die „Sections“, die ja die höchste Gliederungsebene bei `article` sind. (3.) Bei `book` werden außerdem in die Kopfzeile Kapitel- und Section-Überschriften geschrieben.

Nun muss man wissen (die meisten haben es sich wohl schon gedacht), dass \LaTeX (und \TeX) aus den USA stammen. Die meisten automatischen Einstellungen entsprechen dem dort vorherrschenden Stilempfinden (oder vielleicht auch dem des Schöpfers von \TeX , Donald E. Knuth). So wird vor die Überschrift von Kapiteln in Büchern grundsätzlich explizit die Zeile „Kapitel X“ (mit X für die Kapitelnummer) eingefügt. Alternative Klassen werden teilweise dem europäischen Stilempfinden etwas besser gerecht (was nicht heißt, dass sie nicht zum Ausgleich Nachteile haben). In Deutschland sehr gängig sind die KOMA-Script-Klassen.⁹ Diese sind in Analogie zu den Standard-Klassen gebildet, haben aber (natürlich) andere Namen: `scrbook` statt `book`, `scrreprt` statt `report` und `scrartcl` statt `article`. Bei diesen Klassen läuft der Text etwas breiter; auch wird für die Überschriften eine andere Schrifttype (ohne Serifen) verwendet. (Auch einige Erweiterungen setzen voraus, dass man mit dem KOMA-Script-Klassen arbeitet.)

Hingewiesen sei auch auf die Klasse `memoir`. Diese vereinigt in etwa die Eigenschaften von `book` und `report`, hat aber verschiedene Möglichkeiten der Anpassung eingebaut, die man sonst nur über Zusatzpakete bekommen kann. Eine kurze Einführung findet man etwa in einem Beitrag von Christine Römer zur \TeX nischen Komödie: http://archiv.dante.de/DTK/PDF/komoedie_2012_2.pdf (dort S. 72 finden).

Die bislang genannten Dokumentklassen passen nicht für alle Arten von Dokumenten (etwa oder Briefe oder Protokolle). Für Briefe gibt es zwar eine \LaTeX -Klasse, hier weichen aber die US-Normen besonders deutlich von den deutschen Konventionen ab. Doch auch die Koma-Klasse `scrlettr2` hilft beispielsweise in deutschen Universitäten nicht immer, weil man dort meist auch das Logo der Universität einbinden muss. Ich selbst verwende daher für Briefe doch noch die übliche Bürosoftware. Bei anderen Problemen helfen

⁹ Der Name der Klasse scheint auf deren Erfinder, Markus Kohm zurückzugehen.

allerlei nutzerdefinierte Klassen.

Für Sitzungsprotokolle gibt es beispielsweise die Klasse `meetingmins`, die voreingestellte Felder und Befehle für die An- und Abwesenden sowie den Vorsitzenden enthält, ein paar zusätzliche Möglichkeiten bietet und insgesamt in Satzspiegel und Formatierung besser dieser Art von Dokument entspricht. Allerdings ist die „Amtsprache“ dieser Klasse Englisch (was auch bei der Verwendung von `ngerman` dazu führt, dass das Protokoll an bestimmten Stellen englische Ausdrücke enthält), und einige erwartbare Angaben wie Ort, Anfang und Ende der Sitzung oder der Protokollant fehlen. Wer auch nur eine minimale Vorstellung davon hat, wie man programmiert, kann aber leicht die Klassen-Datei `meetingmins.cls` entsprechend abändern und erweitern, auch ohne genau zu verstehen, was welcher Befehl bedeutet.¹⁰ Nützlich finde ich auch die Klasse `handout`. Manchmal werden Layout-Änderungen auch über „Stile“ möglich. So hat André Miede den ebenso löblichen wie heroischen Versuch gemacht, die Gestaltungsprinzipien von Robert Bringhursts *Elements of Typographic Style* umzusetzen; dies geschieht mittels `classicthesis.sty`.

Weitere Klassen wurden geschaffen, um völlig andere typographische Konzepte umzusetzen. Beispielsweise gibt es einige Klassen, die darauf angelegt sind, die Layouts der Bücher von Edward Tufte umzusetzen (nicht nur für Bücher, sondern beispielsweise auch für Handouts!). Die nötigen Infos findet man unter <https://code.google.com/p/tufte-latex/>.

Nur hingewiesen sei auf die `beamer`-Klasse. Das ist eine phantastische Klasse für Präsentationen, also eine Alternative zu PowerPoint. Aber Präsentationen sind etwas ganz Anderes als Texte, daher gehe ich hier nicht näher darauf ein. Jeder, der Formeln in Präsentationen unterbringen muss oder möchte, sollte sich aber mit `beamer` beschäftigen (Mathematiker und Physiker tun das ohnehin ganz selbstverständlich).

8.2 Das Seitenlayout

Alles Wissen über Klassen hilft nichts, wenn Ihr Dozent darauf besteht, dass Sie Vorgaben wie die oben genannten einhalten. Nun, zum Glück ist hier die Abhilfe relativ einfach. Man kann einige Elemente des Seitenlayouts in der Präambel abwandeln; wir benötigen dafür vor allem den Befehl `\setlength`.

Seitenränder Am zweckmäßigsten ist es hier, zunächst die Breite des Textes zu definieren. Ein Seitenrand von 3 cm jeweils links und rechts führt bei DIN A4-Papier zu einer Textbreite von 15 cm. Merkwürdigerweise fällt jedenfalls bei den Artikel-Klassen der Text trotzdem etwas schmaler aus; ich muss die Breite auf 15,5 cm festlegen, damit am Ende 15 cm herauskommen

¹⁰ Je nach Rechten auf dem Rechner kann man diese Datei möglicherweise nicht am ursprünglichen Ort abspeichern. Da \LaTeX standardmäßig nach Klassen, Style-Dateien etc. zunächst in dem Verzeichnis sucht, in dem das zu erstellende Dokument steht, kann man die Datei einfach dort ablegen.

(das kann mit dem Druckertreiber zusammenhängen). Außerdem sind die Ränder der wenigsten L^AT_EX-Klassen symmetrisch; für symmetrische Ränder muss man den linken Rand umdefinieren. Kurz und gut, für die genannten Einstellungen schreibt man in die Präambel:

```
\setlength{\textwidth}{15.5cm}
\setlength{\oddsidemargin}{0.3cm}
```

Prüfen Sie das Ergebnis und variieren Sie gegebenenfalls die genannten Zahlen etwas. Das gilt auch für den Fall, dass andere Ränder als 3 cm erwartet werden.

Um auch oben und unten weniger Rand zu bekommen, wird analog die Texthöhe geändert und der obere Rand etwas verringert:

```
\setlength{\textheight}{24cm}
\setlength{\topmargin}{-1cm}
```

Zeilenabstand Wird außerdem ein 1,5-facher Zeilenabstand verlangt, kommen folgende zwei Befehle in die Präambel:

```
\usepackage{setspace}
\onehalfspacing
```

Andere Zeilenabstände sind ebenfalls möglich, dafür ist der Aufwand u. U. etwas größer. Im World Wide Web findet man rasch Auskunft.

Abstände in Aufzählungen Manchmal will oder muss man die Abstände zwischen den Elementen einer `\itemize`- oder einer `\enumerate`-Umgebung ändern. Dazu fügt man jeweils nach `\begin{itemize}` (oder einem äquivalenten Ausdruck) folgenden Befehl ein: `\setlength{\itemsep}{XXX}`, wobei für XXX eine Größenangabe steht – entweder absolut, z. B., 20pt (für 20 Punkt) oder 0.3cm, oder relativ, d. h. als Änderung des Ausgangswerts, etwa -2pt (für 2 Punkt weniger) oder +0.1cm (für 0,1 cm mehr).

8.3 Andere Schriften (Fonts)

L^AT_EX bietet von Haus aus keine so breite Schriftpalette wie die heute übliche Textverarbeitung. Es verfügt über eine Standardschrift (die Computer Modern, die Sie hier sehen) und einige wenige Alternativen, namentlich die Times, die Palatino und inzwischen auch die Linux Libertine (diese ist empfehlenswert, wenn einem die Computer Modern zu auffällig, die Times zu ausgelutscht und die Palatino zu aufregend ist). Auch serifenlose Fonts stehen zur Verfügung.

Will man die Computer Modern verwenden, muss man gar nichts tun. Für die eben genannten Fonts muss man in der Regel einfach ein entsprechendes Package laden. Dazu kann ein anderer Zeilenabstand angemessen sein, etwa bei der Palatino. Will man also diese Schrift verwenden, steht in der Präambel:

```
\usepackage{mathpazo}
\linespread{1.05}
```

Will man eine serifenlose Schrift als Ergänzung zur Palatino und außerdem noch eine Schreibmaschinenschrift benutzen, so empfehlen sich – nach Mark Trettins `l2tabu` – folgende Ergänzungen der oben zitierten `usepackage`-Anweisung:

```
\usepackage[scaled=.95]{helvet}
\usepackage{courier}
```

Es gibt noch weitere Möglichkeiten, die den Weg zu einer wesentlich breiteren Schriftenpalette eröffnen. Leider sind diese oft recht kompliziert und überschreiten den Rahmen dieser Einführung. Einen Ausweg bietet aber `XqLaTeX` das im folgenden Abschnitt kurz beschrieben wird.

8.4 Andere Schriften mit XeLaTeX

`XqLaTeX`¹¹ ist eine Software, die sich auf das ‚Verwandeln‘ von in `LaTeX` geschriebenen Texten in eine PDF-Datei bezieht. Standardmäßig wird hierfür unter Windows das Programm `pdfLaTeX` verwendet. Dieses Programm kann allerdings ohne Probleme nur die wenigen Fonts verarbeiten, die mit `TeX` mitgeliefert werden.

Mit `XqLaTeX` kann man nun fast jede auf dem Rechner vorhandene Schrift verwenden! Zwar wurde es hauptsächlich entwickelt, um Open Type-Fonts einzubinden, aber offenbar können auch viele Fonts im True Type-Format verwendet werden. Der Clou: Die Fonts müssen nur auf dem Rechner installiert sein; `XqLaTeX` findet sie automatisch. Für normale Menschen weniger interessant, aber für Philologen von großer Bedeutung: Auf diese Weise kann man auch nicht-lateinische Schriften einbauen – einfach alles, was als Open Type verfügbar ist.

Was ist, und was brauche ich für, `XqLaTeX`? `XqLaTeX` ist einfach eine zusätzliche Software, die sich auf die Druckausgabe bezieht. Der Weg ist ganz einfach: Die meisten Entwicklungsumgebungen (also z. B. `TeXnicCenter`, jedenfalls in der neuesten Version) enthalten `XqLaTeX`. Da es um die Druckausgabe geht, müssen Sie einfach bei Ihren Ausgabemöglichkeiten nachschauen und dort entsprechend `XqLaTeX` auswählen. Im `TeXnicCenter` heißt das: Ihre Ausgabe läuft nicht von `LaTeX` nach PDF, sondern eben von `XqLaTeX` nach PDF.

Allerdings muss man in seinem `LaTeX`-Dokument ein paar Änderungen vornehmen. Man braucht (a) ein paar zusätzliche Befehle in der Präambel, muss (b) die `LaTeX`-Datei mit UTF8-Codierung speichern und (leider, leider) (c) auch beim Schreiben des Dokuments ein paar Dinge anders machen.

¹¹ So wie `LaTeX` eine Weiterentwicklung von `TeX` ist, gibt es auch `XqLaTeX` und `XqTeX`; letzteres arbeitet direkt mit in `TeX` geschriebenen Dateien.

Befehle in der Präambel Der erste wichtige Punkt ist: Löschen Sie die Befehle `\usepackage[ansinew]{inputenc}` und `\usepackage[T1]{fontenc}`. Und der zweite: Fügen Sie stattdessen ein paar andere Befehle ein. Hierzu kann man etwas unterschiedliche Angaben finden. Ich glaube aber, für den Normalfall genügt folgende beiden Befehle:

```
\usepackage{xltextra}
\setmainfont[Mapping=tex-text]{Gewünschter Font}
```

Mit dem ersten Befehl werden simultan mehrere Pakete geladen, die für X_YLaTeX hilfreich sind (darunter auch viele, die man aus anderen Gründen laden kann und vielleicht auch laden möchte; das ist aber kein Problem, weil LaTeX sich nicht daran stört, wenn ein Paket zweimal aufgerufen wird). Der zweite Befehl legt die gewünschte Schrift fest. An der Stelle von **Gewünschter Font** muss also der Name des Fonts stehen.

Speichern mit UTF8-Codierung Ihre Entwicklungsumgebung sollte diese Speicherung ermöglichen. Probieren Sie einfach mal „Speichern unter“ und schauen Sie sich die Einstellungen an. (Die Voreinstellung unter Windows-Rechnern ist normalerweise ANSI.)

Beim Schreiben aufgepasst! Leider funktionieren einige der oben genannten Befehle für besondere Zeichen nicht, vor allem Abkürzungen durch Sonderzeichen. Schon für ganz einfache Texte muss man möglicherweise folgende Befehle für Gedankenstrich und Anführungszeichen verwenden:

LaTeX	X _Y LaTeX	Anmerkung
--	<code>\textendash</code>	Gedankenstrich
“	<code>\quotedblbase</code>	Doppelte Anführung, öffnend
”	<code>\textquotedblleft</code>	Doppelte Anführung, schließend
,	,	Einfache Anführung, öffnend (unverändert)
‘	<code>\textquoteleft\</code>	Einfache Anführung, schließend

Man beachte insbesondere, dass man für die „rechten“ (also schließenden) Anführungszeichen die Befehle für die „linke“ Zeichen verwenden muss. Der Grund ist ganz einfach: Die Befehle beziehen sich auf die im angelsächsischen Sprachraum üblichen Anführungszeichen.

Wenn Sie die Kürzel aus der Spalte LaTeX verwenden, werden Sie vielleicht denken: Was soll’s, das funktioniert doch? Aber Achtung – X_YLaTeX verwendet hier (jedenfalls beim gegenwärtigen Stand der Dinge) die Anführungszeichen aus der Computer Modern! Das ist wahrscheinlich ein Bug – aber aktuell ist er noch vorhanden.

Gibt es sonst noch Nachteile? Leider ja: Das Microtype-Paket, welches Einiges dazu tut, mit LaTeX geschriebene Texte in die Nähe typographischer Vollkommenheit zu rücken, funktioniert offenbar nicht. Das ist nicht tragisch,

aber wenn man sich mal an optischen Randausgleich gewöhnt hat, ist es doch sehr schade.

9 Stuff not easily found

Dieser Abschnitt ist einer Einführungsschrift eigentlich deplatziert, denn er bezieht sich auf fortgeschrittene Dinge, für die hier noch gar nicht die Grundlagen vorhanden sind. Ich schreibe diesen Abschnitt für mich – um Dinge festzuhalten, die ich (nützlich) gefunden habe und nicht jedes Mal wieder suchen möchte. Vielleicht sind sie aber auch für Dritte hilfreich.

9.1 Fallunterscheidungen in mathematischen Formeln

Es gibt verschiedene Möglichkeiten, Formeln über mehrere Zeilen zu schreiben und dabei die Elemente untereinander zu arrangieren. Ein nicht ganz seltener Fall sind Fallunterscheidungen; diese werden aber in manchen Einführungen nicht behandelt.

Dafür verwendet man die Umgebung `\begin{cases} ... \end{cases}`. Die Elemente, die jeweils untereinander stehen sollen, werden durch das `&`-Zeichen & voneinander abgetrennt; eine neue Zeile wird (wie üblich) durch zwei Backslashes angefordert. Hier ein Beispiel:

$$f(x) = \begin{cases} P(X = x_i) = p_i & \text{wenn } x = x_i \in \{x_1, x_2, \dots, x_k, \dots\} \\ 0 & \text{sonst} \end{cases}$$

Und hier der Quellcode dazu:

```
\[f(x) =
\begin{cases}
P(X = {x_i}) = {p_i} & \text{wenn } x = {x_i} \text{ in}
\left\{ \{x_1, x_2, \dots\} \{x_k, \dots\} \right\} \ \backslash
& \text{sonst}
\end{cases}
\]
```

Voraussetzung ist – wie für alle mathematischen Dinge – das Laden des Paketes `amsmath`, am besten auch noch gleich von `amsmath` und `amssymb`, was in einem Befehl geschehen kann: `\usepackage{amsmath,amsmath,amssymb}`.

9.2 „Typographische“ Brüche

Ich weiß gar nicht, ob das die richtige Bezeichnung ist. Es geht um Brüche, die man (vorzugsweise im Fließtext) schreibt wie folgt: $\frac{1}{2}$ oder $\frac{1}{8}$.

Hier für benötigt man das Package `xfrac`, das man in der Präambel durch `\usepackage{xfrac}` lädt. Die Brüche selbst werden dann durch `\sfrac` statt `\frac` erzeugt, im Beispiel: `\sfrac{1}{2}`.

9.3 Das Komma als Dezimalzeichen in mathematischen Umgebungen

Die Original-Einstellungen von \LaTeX führen dazu, dass innerhalb mathematischer Umgebungen nach dem Komma ein wenig Platz freigelassen wird – was aber den Gewohnheiten des deutschen Schriftsatzes widerspricht, wo das Komma des Dezimalzeichen repräsentiert. Man vergleiche $1,67$ mit der korrekten Schreibung $1,67$.

Ein ad-hoc-Behelf besteht darin, das Komma in geschweifte Klammern setzen: $\$1\{, \}67\$$. Übergreifend stehen verschiedene Pakete zur Verfügung, die Abhilfe verschaffen.

- Das Paket „ziffer“, ladbar mit `\usepackage{ziffer}`, sorgt automatisch dafür, dass die Dezimalkommata richtig (also gemäß deutschen Regeln) gesetzt werden. Ähnlich ist das Paket „icomma“, mit welchem ich aber keine Erfahrung habe.
- Das Paket „numprint“ bietet zahlreiche weitere Möglichkeiten. Allerdings muss dann auch jede Zahl, für die diese Möglichkeiten in Anspruch genommen werden, mit dem numprint-Befehl gerahmt werden. Das Paket kann mit der Option „np“ geladen werden, die bewirkt, dass der numprint-Befehl durch np abgekürzt werden kann. Man lädt also `\usepackage[np]{numprint}` und schreibt beispielsweise $\$ \backslash np \{ 1,67 \} \$$. Eine weitere Möglichkeit, die numprint anbietet, ist die Gliederung großer Zahlen durch kleine Leerzeichen: $\$ \backslash np \{ 10000000 \} \$$ resultiert in 10 000 000. (Diese Möglichkeit besteht auch außerhalb des Mathematik-Modus.)

9.4 Mathematik im Fließtext und Zeilenabstand

Baut man mathematische Formeln in den Fließtext ein, wird der Zeilenabstand lokal angepasst, damit die Formel gut lesbar ist, allerdings geht dies zu Lasten eines gleichmäßigen Grauwertes des Textes und wirkt ‚unrund‘. Ein Beispiel: $2 \left(\frac{x+iy}{x} \right)$. Wir sehen, dass der Zeilenabstand nach oben und nach unten im Vergleich zum umgebenden Text etwas größer ist. Natürlich ist derlei in gewissen Grenzen auch Geschmackssache. Wie auch immer: Durch den Befehl `\smash{}` – zwischen die geschweiften Klammern kommt die Formel, einschließlich der Dollar-Zeichen, die den Wechsel in den Mathematik-Modus bewirken – kann man \LaTeX zwingen, den Zeilenabstand nicht anzupassen: $2 \left(\frac{x+iy}{x} \right)$. Jetzt wird der Zeilenabstand beibehalten. Wie man sieht, kann man sich damit aber andere Probleme einhandeln, so dass der Einsatz von `\smash` gut überlegt sein will; ansonsten sollte man die die Formel besser freistellen.

9.5 Durchstreichen

Zugegeben: Durchstreichen sieht nicht schön aus. Aber manchmal kann man es zu didaktischen Zwecken einsetzen, beispielsweise: ~~Durchstreichen sieht nicht schön aus~~.

Hierfür benötigt man `\usepackage[normalem]{ulem}` in der Präambel (die Option `[normalem]` ist erforderlich, weil andernfalls alles, was sonst kursiv wäre, unterstrichen wird – `ulem` steht tatsächlich für „underline emphasis“). Das Durchstreichen selbst geschieht vermittle des Befehls `\sout{}`, z. B. `\sout{Durchstreichen sieht nicht schön aus}`.

Das Paket `ulem` enthält noch ein paar andere Dinge, z. B. unterkringeln (mit `\uwave{}`) oder ~~radikales Ausstreichen~~ mit `\xout{}`.

10 Englisch

Bei englischen Texten entfällt der Befehl `\usepackage{ngerman}` im Kopf der Datei. Auch `[a4paper]` in der ersten Zeile kann im Grundsatz entfallen. Allerdings richtet sich das Ausgabeformat ohnehin meist nach den Drucker-Einstellungen, jedenfalls bei PDF. (In diesem Punkt muss man aber u. U. auf lokale Gegebenheiten achten.)

Englische Anführungszeichen erhält man dann durch einen (gegebenenfalls doppelten) Gravis – also das Zeichen ``` – am Beginn und durch den (gegebenenfalls doppelten) Apostroph – das Zeichen, das man auf deutschen Tastaturen durch `Shift+#` erhält – am Ende der betreffenden Textpassage.

Für dashes, also Gedankenstriche, muss man im Englischen drei Minuszeichen eingeben, also `---`.