

# $\text{\TeX}$ und $\text{\LaTeX}$

## Eine Einführung

Lars-Chr. Schulze, DC0BM

Version 1.0.2

# Übersicht I

- 1 Allgemeines
- 2 Wie arbeitet T<sub>E</sub>X
- 3 Texteingabe
- 4 Formeln
- 5 Was ich weggelassen habe
- 6 Referenzen

- T<sub>E</sub>X (gesprochen „tech“, im Englischen „tek“) ist ein Text**sat**zsystem.
- Es wurde in den 70'ern und 80'ern des letzten Jahrhunderts von Prof. Donald Knuth von der Universität Stanford, CA, entwickelt.
- Angeblich als Reaktion auf die erbärmliche Qualität des ersten Versuchs, seine Buchreihe *The Art Of Computer Programming* mit eben diesen zu setzen ☺.
- T<sub>E</sub>X steht dabei für die drei griechischen Buchstaben Tau, Epsilon und Chi ( $\tau\epsilon\chi$ ) und soll die Fähigkeit zum Setzen wissenschaftlicher Texte inkl. Formeln symbolisieren.
- T<sub>E</sub>X ist ein Markenzeichen der *American Mathematical Society*.
- Es gibt eine Test-Suite für T<sub>E</sub>X. Nur ein Programm, welches diese besteht, darf sich T<sub>E</sub>X nennen.
- Wo T<sub>E</sub>X draufsteht, ist also auch T<sub>E</sub>X drin.

- Die erste Version war das sog.  $\text{T}_{\text{E}}\text{X}78$ .
- Diese wurde dann weiterentwickelt zum sog.  $\text{T}_{\text{E}}\text{X}82$ .
- Auch wenn es inzwischen ein paar Ergänzungen und Erweiterungen gab, ist das im Kern immer noch die Version, die heute verwendet wird.
- Daher muß man bei allem, was einem bei  $\text{T}_{\text{E}}\text{X}$  heute merkwürdig oder evtl. umständlich vorkommt, bedenken, welchen Stand die Computertechnik zu der Zeit hatte (oder nicht hatte ☺).
- Aber durch die Arbeit vieler begeisterter Anwender, welche jede Menge Zusatzpakete („Bibliotheken“) geschrieben haben, und die Fortschritte der Computertechnik, kann  $\text{T}_{\text{E}}\text{X}$  heute für Dinge eingesetzt werden, für die es nie gedacht war.

# Allgemeines

## Vorteile von T<sub>E</sub>X I

- T<sub>E</sub>X ist frei verfügbar.
- T<sub>E</sub>X ist plattformunabhängig und läuft auf **jedem** Betriebssystem.
- Der Austausch der Quelldateien zwischen den Systemen ist problemlos möglich (Kodierung beachten).
- T<sub>E</sub>X bietet eine hohe Druckqualität nach den Regeln des professionellen Buchdrucks, z. B.
  - Blocksatz (der Satz mit Randausgleich) inkl. Silbentrennung
  - Ligaturen, z. B. ff ffi fi ffi  $\Leftrightarrow$  ff ffi fi ffi.
  - Kerning (Unterschneidung), z. B. Va Te fe  $\Leftrightarrow$  Va Te fe,
- Professioneller Formelsatz.
- Viele wissenschaftliche Bücher sind in T<sub>E</sub>X gesetzt.

# Allgemeines

## Vorteile von T<sub>E</sub>X II

- Moderater Ressourcenverbrauch.  
T<sub>E</sub>X lief seinerzeit auf einem 286'er-PC mit 20 MHz Takt und 640 kB RAM oder einem Atari-Mega-ST mit 8 MHz und 4 MB RAM.
- T<sub>E</sub>X besitzt teilweise Markup-Eigenschaften (Auszeichnungseigenschaften), ähnlich wie HTML oder XML.
- Allerdings ist ein T<sub>E</sub>X-Text **wesentlich** besser lesbar als HTML.
- Hohe Rückwärts-kompatibilität.  
Alte (L<sup>A</sup>T<sub>E</sub>X)-Dokumente können meistens mit geringem oder gar keinem Anpassungsaufwand mit neueren Versionen bearbeitet werden.
- Sehr einfache Erstellung automatischer Dokumente.

- Etwas höherer Einarbeitungsaufwand.  
Die ersten 2 bis 3 Schritte sind etwas schwieriger.
- Die Erstellung neuer Layouts kann etwas aufwendiger sein als bei den „WYSIWYG“-Systemen („What you see is what you get“, manchmal jedenfalls).
- Inzwischen ist aber eine Vielzahl von Zusatzpaketen für fast alle denkbaren Layout-Wünsche und Anpassungen verfügbar.

# Wie arbeitet T<sub>E</sub>X

## Grundsätzliche Arbeitsweise I

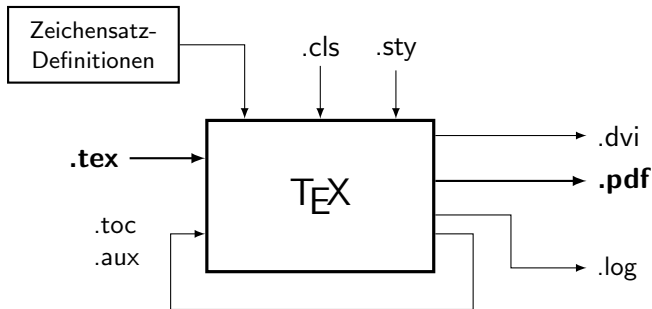


Abbildung: Die Struktur von T<sub>E</sub>X (stark vereinfacht).



# Wie arbeitet T<sub>E</sub>X

## Grundsätzliche Arbeitsweise II

- T<sub>E</sub>X ist im Prinzip ein Compiler.
- T<sub>E</sub>X liest die Eingabedatei (\*.tex) sowie die evtl. angeforderten Zusatzpakete (\*.cls, \*.sty) und erzeugt daraus das sog. *Device Independant File* (\*.dvi, früher).

**In dieser DVI-Datei war die gesamte Formatierung des Dokumentes unveränderbar festgelegt.**

- Diese DVI-Datei konnte dann entweder über entspr. Druckertreiber ausgedruckt und/oder in andere Formate konvertiert werden.
- Wer schon mal programmiert hat, wird sich mit dieser Arbeitsweise schnell zurechtfinden.
- Heute wird meist eine Variante von T<sub>E</sub>X verwendet, das sog. pdfT<sub>E</sub>X, welche direkt eine .pdf-Datei erzeugt.

# Wie arbeitet T<sub>E</sub>X

T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X und Pakete

- **Reines T<sub>E</sub>X**

Programmierung in Maschinencode „zu Fuß“.

- **Makropaket plainT<sub>E</sub>X** → „T<sub>E</sub>X“

Programmierung in Maschinencode mit Assembler.

- **Makropaket L<sup>A</sup>T<sub>E</sub>X** → „L<sup>A</sup>T<sub>E</sub>X“

Programmiert von Leslie Lamport.

„Hochsprache“ mit strukturierten Kommandos.

Hier gibt es eine *Begriffsüberlappung*: Man sagt T<sub>E</sub>X, meint aber eigentlich L<sup>A</sup>T<sub>E</sub>X. Kaum jemand erstellt Dokumente in plainT<sub>E</sub>X [9, 16].

- **Zusatzpakete und Styles** Erweiterungsbibliotheken für spezielle Aufgaben oder Layouts.

# Texteingabe

## Ein minimales L<sup>A</sup>T<sub>E</sub>X-Dokument

Es wurde ja schon erwähnt, daß T<sub>E</sub>X mit sog. **Auszeichnungen** arbeitet. Dies läßt sich am besten an einem Beispiel zeigen.

Ein minimales L<sup>A</sup>T<sub>E</sub>X-Dokument sähe z. B. folgendermaßen aus<sup>1</sup>:

```
\documentclass[12pt]{article}    % Allgemeines Layout

\usepackage[ngerman]{babel}      % Deutsche Silbentrennung
\usepackage[utf8]{inputenc}      % Deutsche Umlaute erlauben
\usepackage[T1]{fontenc}

\begin{document}
Dies ist mein erster \TeX-Text.
\end{document}
```

---

<sup>1</sup>Das Prozentzeichen „%“ leitet einen Kommentar ein.

# Texteingabe

## Ein minimales L<sup>A</sup>T<sub>E</sub>X-Dokument I

- Formatierungen und Strukturanweisungen werden durch entsprechende Steuersequenzen in den Text mit eingegeben.
- Die Steuersequenzen beginnen mit „\“ (Backslash).
- Das grundsätzliche Layout des Textes (Seitenformat, Kopf- und Fußzeilen, Schriftart) wird durch die *Klassen* (\*.cls-Dateien) bestimmt.
- Die *Styles* (\*.sty-Dateien) machen dann die Feinjustierung.
- Einfache Kommandos bestimmen Schriftattribute wie **fette** oder *kursive* Schrift oder die **Schriftgröße**.

# Texteingabe

## Beispiel für Formatierungen I

Dieser Text soll verdeutlichen, wie `\TeX` arbeitet. Mehrere Leerzeichen werden wie eines behandelt.

Eine kurze Zeile bedeutet für `\TeX` `\textbf{nicht}` das Ende eines Absatzes. `\TeX` liest immer einen ganzen Absatz ein und bricht diesen unter typografischen Gesichtspunkten möglichst optimal um, inkl. Silbentrennung.

Daher stimmt der Umbruch in der Eingabedatei auch nicht mit dem der Ausgabe überein. Am `\textbf{\textit{Inhalt}}` des Textes ändert sich dadurch aber nichts.

Erst eine bzw. mehrere `\textit{Leerzeilen}` bedeuten für `\TeX` einen neuen Absatz. Dieser wird dann mit dem vordefinierten Abstand zum vorherigen Absatz gesetzt. `{\Large Eigentlich ganz intuitiv,}` oder nicht?

# Texteingabe

## Beispiel für Formatierungen II

Dieser Text soll verdeutlichen, wie T<sub>E</sub>X arbeitet. Mehrere Leerzeichen werden wie eines behandelt. Eine kurze Zeile bedeutet für T<sub>E</sub>X **nicht** das Ende eines Absatzes. T<sub>E</sub>X liest immer einen ganzen Absatz ein und bricht diesen unter typografischen Gesichtspunkten möglichst optimal um, inkl. Silbentrennung. Daher stimmt der Umbruch in der Eingabedatei auch nicht mit dem der Ausgabe überein. Am *Inhalt* des Textes ändert sich dadurch aber nichts.

Erst eine bzw. mehrere *Leerzeilen* bedeuten für T<sub>E</sub>X einen neuen Absatz. Dieser wird dann mit dem vordefinierten Abstand zum vorherigen Absatz gesetzt. Eigentlich ganz intuitiv, oder nicht?

# Texteingabe

## Beispiel für Formatierungen III

- T<sub>E</sub>X liest immer einen **ganzen** Absatz und versucht, diesen, nach typografischen Gesichtspunkten optimal umzuberechnen.
- Dabei arbeitet T<sub>E</sub>X wie früher die Setzer und reiht Zeichen für Zeichen aneinander, bis eine Zeile gefüllt ist.
- Dann beginnt T<sub>E</sub>X mit der nächsten Zeile.
- Die einzelnen Zeilen werden dann mit einem definierten Abstand zueinander gesetzt.
- Ist der Absatz formatiert, wird das Ergebnis in ein Ausgaberegister geschrieben.
- Wenn eine Seite komplett ist, wird sie ausgegeben.
- T<sub>E</sub>X macht **keine** Optimierung des Layouts **über mehrere Seiten** hinweg.

- T<sub>E</sub>X verlangt einen strukturierten Dokumentenaufbau. Dadurch entsteht ggf. ein etwas höherer Aufwand.
- Dazu werden sog. *Umgebungen* definiert.
- Umgebungen stellen bestimmte Funktionen und/oder Formatierungen zur Verfügung.
- Sie werden durch `\begin{<name>} ... \end{<name>}` eingeschlossen.
- Eine Umgebung haben wir schon gesehen: die Umgebung `document`.
- Beispiele für Umgebungen sind:
  - 1 Mathematische Umgebungen (`displaymath`, `equation`, ...)
  - 2 Aufzählungen (`itemize`, `enumerate`)
  - 3 Literaturverzeichnis (`thebibliography`)



- Umgebungen müssen korrekt verschachtelt sein.
- Ca. 40 Umgebungen sind standardmäßig vordefiniert.
- Zusätzliche Umgebungen werden durch Zusatzpakete bereitgestellt oder selbst definiert.
- Darüberhinaus erlaubt  $\text{\TeX}$  Blockbildungen mit lokalen Eigenschaften und Variablen, ähnlich wie in höheren Programmiersprachen.
- Blöcke werden durch geschweifte Klammern `{...}` eingeschlossen.
- Änderungen von Eigenschaften innerhalb eines Blockes haben nach außen keine Wirkung.
- Umgebungen wirken gleichzeitig wie Blöcke, d. h. Änderungen von Parametern innerhalb eines Blocks haben nach außen keine Wirkung.

# Texteingabe

## Blöcke und Umgebungen III

```
\begin{itemize}
...
\item Umgebungen stellen bestimmte Funktionen und/oder Formatierungen
zur Verfügung, \zB:
\begin{enumerate}
\item Mathematische Umgebungen (displaymath, equation, \ldots)
\item Aufzählungen (itemize, enumerate)
\item Literaturverzeichnis (thebibliography)
\end{enumerate}
...
\item Umgebungen müssen korrekt verschachtelt sein.

\item Ca.\ 40 Umgebungen sind standardmäßig vordefiniert.

\item Zusätzliche Umgebungen werden durch Styles bereitgestellt oder
selbst definiert.
...
\end{itemize}
```

# Texteingabe

## Gliederungen

- Die Gliederung des Textes erfolgt über Kommandos wie `\chapter`, `\section`, `\subsection`, `\subsubsection` etc.
- Beispiel: `\section{Einleitung}`
- Der eingeschlossene Text wird damit als Überschrift 1. Ordnung gekennzeichnet („Markup“).
- Die tatsächlich Formatierung hängt dann vom der gewählten Dokumentenklasse und ggf. dem Style oder der Umgebung ab.
- T<sub>E</sub>X übernimmt dann die (fehlerfreie !) Numerierung der Kapitel und Unterkapitel entspr. dem gewählten Style.
- Mit der Anweisung `\tableofcontents` kann dann an einer beliebigen Stelle das Inhaltsverzeichnis eingefügt werden.
- Lädt man im Vorspann mittels `\usepackage{hyperref}` das Paket Hyperref dazu, sind alle Einträge im Inhaltsverzeichnis und sonstige Referenzen Hyperlinks.

# Texteingabe

## Boxen – Keine Sportart I

- Intern arbeitet T<sub>E</sub>X mit sog. **Boxen**, in denen der Text untergebracht wird.
- Im wesentlichen unterscheidet man hier *horizontale* und *vertikale* Boxen (`\hbox{...}` und `\vbox{...}`).
- Wenn T<sub>E</sub>X eine Zeile mit Zeichen zusammenstellt, sieht das für T<sub>E</sub>X so aus:



- Man erkennt hier die beiden horizontalen Boxen für die beiden Zeilen, welche dann mit dem definierten Zeilenabstand in einer vertikalen Box übereinander gesetzt werden.

- Man kann aber auch explizit mit den Boxen arbeiten.

Hier ein kleines Beispiel für einen improvisierten 2-spaltigen Satz. In dieser Spalte ist der überschüssige Platz mit

Leerraum aufgefüllt. Die Texte „kleben“ daher oben und unten an der Box.

Und dies ist ein Text in der zweiten Spalte. Sie ist normal formatiert. Die Rahmen zeigen die verwendeten Boxen an.

- Hiermit muß man sich aber nur beschäftigen, wenn man eigene Layouts erstellen will.
- Bei  $\text{\LaTeX}$  sind inzwischen Pakete für fast alle denkbaren Layout-Wünsche vorhanden.

- T<sub>E</sub>X kennt darüberhinaus das Konzept des beliebig dehn- und schrumpfbaren Leerraums, den **glue** (Leim).
- Diesen *glue* kann man sich wie **Federn** vorstellen, die die umgebenden Textteile auseinanderdrücken, aber auch verbinden.
- T<sub>E</sub>X fügt den *glue* automatisch zwischen Wörtern und Absätzen und an diversen anderen Stellen ein.
- Man kann ihn aber auch explizit verwenden.
- Auf diese Weise kann man auch relativ einfach Kopf- und Fußzeilen mit einem linken, einem rechten und einem zentrierten mittleren Teil erstellen.
- Das sieht dann z. B. so aus:

DOC-ABC-123	Version 1.0	22
-------------	-------------	----

# Formeln

## Wie setze ich Formeln I

- Formeln können bei T<sub>E</sub>X im laufenden Text  $c = \sqrt{a^2 + b^2}$  oder als abgesetzte Formeln gesetzt werden:

$$c = \sqrt{a^2 + b^2}$$

- Zur Eingabe von Formeln muß man T<sub>E</sub>X zunächst in den **mathematischen Modus** schalten.
- Im laufenden Text geschieht dies dadurch, daß man die Formel durch  $\dots$  einrahmt.
- Für abgesetzte Formeln existieren verschiedene mathematische Umgebungen: *displaymath*, *equation*, *eqnarray*, ...
- Die unterschiedlichen Umgebungen bestimmen, ob eine Formel eine Nummer erhält und wie sie ausgerichtet wird.

# Formeln

## Wie setze ich Formeln II

- Formeln werden im wesentlichen so eingegeben, wie man es vom Programmieren her gewohnt ist.
- Der Pythagoras sieht in der Eingabe so aus:

```
\begin{displaymath}
  c = \sqrt{a^2 + b^2}
\end{displaymath}
```

- Bzw. in der laufenden Zeile: `$c = \sqrt{a^2 + b^2}$`.
- Über den mathematischen Formelsatz gibt es auch ein witziges Einführungsvideo von einem Prof. Weitz aus Hamburg [10].



- Über Formeln in T<sub>E</sub>X könnte man endlos referieren.
- Das Thema füllt ganze Kapitel in einschlägigen Büchern.
- Statt dessen noch kurz ein paar „Hingucker“.

Eine Matrix:

$$\mathbf{A} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix}$$

Die Standardabweichung:

$$s = \sqrt{\frac{1}{n} \sum_{i=1}^r p_i (x_i - x)^2} \quad (1)$$

Die Wärmeleitungsgleichung:

$$\frac{1}{a^2} \frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x_1^2} + \dots + \frac{\partial^2 u}{\partial x_n^2} \quad (2)$$

Das Poisson-Integral:

$$u(x, t) = \frac{1}{2a\sqrt{\pi}} \int_{-\infty}^{+\infty} \frac{\mu_0(\bar{x})}{\sqrt{t}} e^{-(x-\bar{x})^2/(4ta^2)} d\bar{x} \quad (3)$$

# Was ich weggelassen habe

- Einbinden von Bildern
- Erstellung von Bildern, z. B. mit TikZ, CircuiTikZ.
- „Fließobjekte“ (Bilder, Tabellen, etc.).
- Listen der Fließobjekte.
- Literaturverzeichnis
- Referenzen auf Bilder, Tabellen, Kapitel, Seiten, Einträge im Literaturverzeichnis ...
- Spezielle IDEs, Editoren.
- BibT<sub>E</sub>X und MakeIndex
- ...

# Referenzen I

Hier findet Ihr weitere Informationen zu den Themen  $\text{T}_{\text{E}}\text{X}$  und  $\text{\LaTeX}$ .

**Hinweis:** Beim Suchen im Internet solltet Ihr als Suchbegriff „Latex Tutorial“ angeben, sonst bekommt Ihr Ergebnisse angezeigt, die Ihr u. U. nicht sehen wollt ☺.

- [1] Die  $\text{\LaTeX}$ -Kurzanleitung, die Anleitung zum Einstieg:  
<https://www.ctan.org/tex-archive/info/german/LaTeX2e-Kurzbeschreibung/#l2kurz.pdf>
- [2] CTAN: Das *Comprehensive  $\text{T}_{\text{E}}\text{X}$  Archive Network*: [www.ctan.org](http://www.ctan.org)  
Die offizielle Quelle für alles rund um  $\text{T}_{\text{E}}\text{X}$ .
- [3] Dante – Die *Deutschsprachige Anwendervereinigung  $\text{T}_{\text{E}}\text{X}$* : [www.dante.de](http://www.dante.de)
- [4] Die  *$\text{T}_{\text{E}}\text{X}$  User Group*: [www.tug.org](http://www.tug.org)
- [5] Ein (von vielen) Einführungen in  $\text{\LaTeX}$ :  
[https://www.grund-wissen.de/informatik/latex/\\_downloads/grundkurs-latex.pdf](https://www.grund-wissen.de/informatik/latex/_downloads/grundkurs-latex.pdf)
- [6] Helmut Kopka,  $\text{\LaTeX}$  – Eine Einführung,  
Addison Wesley 1988, ISBN 3.89319-199-2 (vermutl. nicht mehr verfügbar)
- [7] Frank Mittelbach, Ulrike Fischer: The  $\text{\LaTeX}$  Companion, Third Edition – Part I and II.  
Pearson Education, 2023. ISBN: 978-0-13-816648-9
- [8] Michael Goossens et al.: The  $\text{\LaTeX}$  Graphics Companion, Second Edition. Pearson  
Education 2008, ISBN 978-0-321-50892-8

# Referenzen II

- [9] Der  $\text{T}_{\text{E}}\text{X}$ -Code für [16] auf CTAN, in plain $\text{T}_{\text{E}}\text{X}$ :  
<https://www.ctan.org/tex-archive/systems/knuth/dist/tex/#texbook.tex>
- [10] Ein witziges Video von Prof. Weitz zum Formelsatz in  $\text{T}_{\text{E}}\text{X}$ :  
<http://mediathek.mt.haw-hamburg.de/vfaq-klein.pdfvideo/DD-2020-03-29-01-Crashkurs-LaTeX-Syntax-mathematische-Formeln-online-kommunizieren/b478640efb4ea56018b3da584c42891e>
- [11] (Off-Topic: Auf der Webseite <https://weitz.de/haw-videos/> finden sich eine ganze Reihe *sehr* interessanter Videos zu den Themen Mathematik,  $\text{T}_{\text{E}}\text{X}$  und Programmierung. Speziell hinweisen möchte ich hier auf die „Beweise“ der Riemannschen Vermutung: <http://mediathek.mt.haw-hamburg.de/video/DD-2022-08-20-01-Die-schoensten-Beweise-der-Riemannschen-Vermutung-und-noch-mehr/269685348f1d193d17310e0e2d041064>)
- [12] Eine FAQ im Internet mit vielen Hinweisen, Tips und Erläuterungen:  
<https://texfragen.de/> Geht aber teilweise etwas mehr ins Detail.
- [13] Die FAQ von [12] als pdf-Datei: <https://latex.org/detexfaq.pdf>.
- [14] Diskussionsliste rund um  $\text{\LaTeX}$ : <https://latex.org/>.
- [15] Eine umfangreiche Sammlung von Beispielen: <https://texample.net>.

# Referenzen III

Und für Liebhaber: Die fünfbändige „T<sub>E</sub>X-Bibel“.

- [16] Donald E. Knuth, The T<sub>E</sub>Xbook.  
American Mathematical Society und Addison Wesley 1986  
ISBN: 0-201-13447-0, 0-201-13448-9 (soft)
- [17] Donald E. Knuth, T<sub>E</sub>X: The Programm.  
Der vollständige und vollständig dokumentierte Quellcode von T<sub>E</sub>X.  
Addison Wesley 1986, ISBN 0-201-13437-3
- [18] Donald E. Knuth: The METAFONTbook.  
American Mathematical Society 1986  
ISBN: 0-201-13445-4, 0-201-13444-6 (soft)
- [19] Donald E. Knuth: METAFONT: The Programm.  
Der vollständige und vollständig dokumentierte Quellcode von METAFONT.  
Addison Wesley 1986, ISBN 0-201-13438-1
- [20] Donald E. Knuth: Computer Modern Typefaces.  
Der METAFONT-Quellcode zu den *Computer Modern* Zeichensätzen.  
Addison Wesley 1986, ISBN 0-201-13436-2

© Alle Rechte beim Autor. Für Ausbildungs- und Lehrzwecke frei verwendbar.  
Die gewerbliche oder kommerzielle Nutzung bedarf der schriftlichen Genehmigung.  
Nicht referenzierte Bilder vom Autor. Dokument erstellt mit L<sup>A</sup>T<sub>E</sub>X unter Verwendung der Pakete TikZ und Beamer.