



TeX - LaTeX

Prof. Dr. Karsten Urban,
Dipl. Math. Katharina Becker-Steinberger,
Dipl. Math.oec. Sebastian Kestler
Institut für Numerische Mathematik, Universität Ulm

Numerik 1: $\text{T}_{\text{E}}\text{X}$ - $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ -Einführung

Ulm, 24.10.2012

Was sind T_EX und L^AT_EX?

Das erste L^AT_EX-File

Dokument-Klassen und das Einbinden von Paketen

Elementarer Text

Mathematische Formeln

Referenzen

Makros

Zähler

Umgebungen

Tabellen

Bilder

Literaturverzeichnis

T_EX

- T_EX ist Programmiersprache für Textverarbeitung (Textsatzsystem)
 - entwickelt '77-'82 von Prof. Donald Knuth, Stanford University
 - Befehlsumfang etwa 300 Befehle
- T_EX ist Freeware, aber eingetragenes Warenzeichen
 - Abkürzung für griechisch $\tau\epsilon\chi\nu\eta$ - Fähigkeit, Kunstfertigkeit, Handwerk
 - entweder T_EX oder Tex schreiben!
 - Versionsnummer konvergiert gegen π , aktuelle Version 3.1415926 (März 2008)
 - bei Knuths Tod wird Weiterentwicklung gestoppt und Versionsnummer auf π gesetzt
- T_EX gilt als fehlerfreie Software
 - jeder gefundene Fehler wird belohnt
- T_EX erlaubt eigenes schreiben von Makros
 - Makros \approx Funktion
 - genauer:
 - Makro = Abkürzung für gewisse Befehlsfolge
 - Interpreter ersetzt beim Übersetzen Abkürzung durch vollständigen Code
 - entspricht der inline-Funktion in C

Abbildung: Donald Knuth, Quelle: <http://www-cs-faculty.stanford.edu/~uno/>

Makro-Pakete für T_EX

- '82 veröffentlichte die American Mathematical Society eine Makro-Sammlung **amstex** für T_EX
- '85 veröffentlichte Leslie Lamport die Makro-Sammlung **L^AT_EX**
 - heute **de facto Standard in der Mathematik**
 - '89-'03 Entwicklung von L^AT_EX₃
(unvollendet, Projekt als abgeschlossen erklärt)
 - aktuelle Version: L^AT_EX_{2_ε} (2003)



Abbildung: Leslie Lamport, Quelle: <http://research.microsoft.com/en-us/um/people/lamport/>

Vor- und Nachteile von L^AT_EX

Vorteile von L^AT_EX

- L^AT_EX ist Freeware und für alle gängigen Systeme vorhanden (rechner- und betriebssystemunabhängig)
 - Output-Dokumente sehen auf jedem System identisch aus
- produziert professionelles Layout
 - Layout-Vorlagen für Artikel/Bücher/Folien
- nur wenige Befehle für die logische Strukturierung eines Schriftstücks notwendig
- mathematische Formeln können gut umgesetzt werden
- Dokumente lassen sich problemlos erweitern, automatische Aktualisierung von
 - Layout
 - Querverweisen
 - Referenzen
 - Inhalts- und Stichwortverzeichnis
- direkte Schnittstelle zu ps/pdf
- WYSWYM = What you see is what you mean

Nachteile von L^AT_EX

- Einarbeitungszeit
- nicht klickbar
- nicht WYSWYG = What you see is what you get
- eigene Layout-Vorlagen sind vergleichsweise kompliziert zu schreiben

Literatur

Bücher:

- Goossen, M., Mittelbach, F. et a. (2010): *Der L^AT_EX-Begleiter*
2. Aufl., Addison Wesley, München
- Kopka, H. (2002): *L^AT_EX, Bd. 1: Einführung*
3. überarb. Aufl., Addison Wesley, München
- Kopka, H. (2002): *L^AT_EX, Bd. 2: Ergänzungen. Mit einer Einführung in METAFONT*
3. überarb. Aufl., Addison Wesley, München
- Kopka, H. (2002): *L^AT_EX, Bd. 3: Erweiterung: BD3)*
Korrigierter Nachdruck der 2. Aufl., Addison Wesley, München
- Braune, K., Lammarsch, J. u. M. (2006): *L^AT_EX-Basissystem, Layout, Formelsatz*
Springer, Berlin Heidelberg

Internetquellen:

- Oetiker, T., Partl, H., Hyna, I. et al. (2003): *L^AT_EX₂_ε-Kurzbeschreibung*,
www.dante.de/CTAN/info/lshort/german/l2kurz2.pdf, 16.10,2012
- Jürgens, M., Feuerstack, T. (2012): *L^AT_EX - eine Einführung und ein bisschen mehr...* Hrsg. v.d. FernUniversität Hagen,
http://www.fernuni-hagen.de/imperia/md/content/zmi_2010/a026_latex_einf.pdf, 16.10,2012

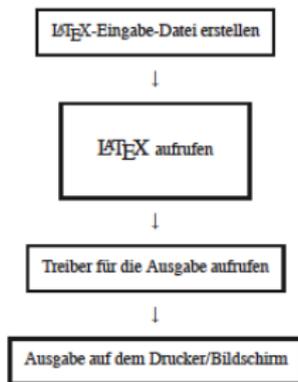
Was braucht man, um mit L^AT_EX zu arbeiten?

L^AT_EX-Software

- arbeitet im Hintergrund
- Bestandteile: T_EX/L^AT_EX-Programme, Schriften, Skripte ...
- Einfachster Installationsweg: Distribution
 - MiKTeX, TeX Live (Unix/Linux/Windows/Mac), MacTeX (Mac OS X)

Eingabe-/Steuerungssoftware (Entwicklungsumgebung)

- Texteditor
 - gedit (GNOME Editor, frei), Texmaker (Plattformunabhängig, freier LaTeX-Editor), Eclipse (Erweiterung TeXlipse), TeXShop (Mac OS X, frei), WinShell (Windows, frei)



1. Eingabefile schreiben (Textfile)
2. File mit L^AT_EX bearbeiten
⇒ erzeugt Datei, die gesetzten Text in **geräteunabhängigem Format** (DVI, PDF, PS) enthält
3. Probeausdruck auf Bildschirm anzeigen (Preview)
4. wenn nötig zurück zu Schritt 2 und Eingabe korrigieren

Abbildung: Vorgehensweise beim Arbeiten mit L^AT_EX, Quelle: Jürgens, M., Feuerstack, T.

Aufbau anhand eines ersten L_AT_EX-Files

Quelldatei (L_AT_EX)

```
1 %Helloworld.tex
2 \documentclass[a4paper,11pt]{article}
3 \usepackage{fullpage}
4
5 \begin{document}
6 Hello World!
7 \end{document}
```

Ausgabe-Datei (PDF)

Hello World!

- Jedes L_AT_EX-Programm (Quelldatei.tex) besitzt die Zeilen 2 ,5, 7
- Übersetzung stets sequentiell von oben nach unten
- Zeilen vor `\begin{document}` bilden den L_AT_EX-Kopf, -Vorspann oder die Präambel
 - Zeile 2: legt Layout des Dokuments fest
 - Zeile 3: bindet Makro-Pakete ein
 - Definition von eigenen Makros
- `\begin{document} ... \end{document}` beinhaltet eigentliches Dokument (Inhalt des Schriftstücks)
- Zeile 1: **Kommentarzeile**, eingeleitet durch %
- L_AT_EX-Befehle beginnen immer mit `\`
 - `\documentclass`, `\usepackage`, `\begin`, `\end`
 - Optionale Parameter immer in `[...]`
 - Obligatorische Parameter immer in `{...}`
- **Beispiel 1: Übersetzen einer Tex-Datei**

Dokument-Klassen

- `\documentclass[options]{documenttyp}`
- Standard-Dokumenttypen in L^AT_EX
 - `article` = wiss. Publikation
 - `report` = kurze Bücher, Bachelor-, Masterarbeiten
 - `book` = Bücher
 - `beamer` = Folien, Präsentationen

Optionale Parameter für `article`

- `10pt`, `11pt`, `12pt` = Schriftgröße für Standardtext
- `a4paper` immer wählen! (Papiergröße)
 - Standard ist `letterpaper` = US-Maße
- `fleqn` = Formeln linksbündig statt zentriert
- `leqno` = Formeln rechtsbündig statt zentriert
- `titlepage` = neue Seite nach Titel/Autor etc.
 - Standard ist `notitlepage`
- `twocolumn` = zweispaltig statt einspaltig
 - Standard ist `onecolumn`
- `twoside` = zweiseitiges Dokument statt einseitig
 - Standard ist `oneside`
- `landscape` = Querformat statt Hochformat
- [Beispiel 2: Dokument-Klasse article](#), [Beispiel 3: Optionale Parameter](#)
- weiteres Beispiel ist das vorliegende Dokument (`beamer`)

Optionale Parameter für `report` und `book`

Wie bei `article`, Ausnahmen sind:

- `notitlepage` = keine neue Seite nach Titelseite
 - Standard ist `titlepage`
- `twocolumn` = zweispaltig statt einspaltig
 - Standard ist `onecolumn`
- `oneside` = einseitiges Dokument
 - Standard ist `twoside`
- `openany` = neue Kapitel beginnen auf neuer Seite
 - Standard ist `openright` = neue Kapitel beginnen stets auf der nächsten rechten Seite

Einbinden von Paketen

- `\usepackage[options]{packagename}`
- Einbinden vom Erweiterungspaket (Makropaket) `packagename`
- übergibt gewisse optionale Parameter
- `fullpage` = minimiert Randbereiche
- `inputenc` = erlaubt direkte Verwendung von Sonderzeichen (Zeichenkodierung)
 - Option `latin1` (Unix) bzw. `applemac` (Mac) oder `utf8` für deutsche Sonderzeichen
 - z.B.: ä ü, ö, ß
 - Vergessen: Sonderzeichen werden weggelassen d.h.
- `babel` = Wahl der Sprache des Dokuments
 - Option `ngerman` - Neue deutsche Rechtschreibung
 - beeinflusst automatische Silbentrennung
 - „Kapitel“ statt „Chapter“, etc.

Quelldatei (L^AT_EX)

```
1 %Helloworld.tex
2 \documentclass[a4paper,11pt]{article}
3 \usepackage{fullpage}
4 \usepackage[latin1]{inputenc}
5 \usepackage[ngerman]{babel}
6
7 \begin{document}
8 Hello Wörlld!
9 \end{document}
```

Ausgabe-Datei (PDF)

Hello Wörlld!

- Beispiel 4: Einbinden von Paketen, deutsche Sonderzeichen

Leerzeichen

- L^AT_EX interpretiert folgendes als ein Leerzeichen:
 - ein oder mehrere Leerzeichen
 - ein oder mehrere Tabulator-Einrückungen
 - ein Zeilenumbruch im Dokument
- Manuelles Leerzeichen mittels Tilde ~ oder Backslash \
 - Tilde verhindert Zeilenumbruch
- L^AT_EX interpretiert folgendes als Absatzende:
 - ein oder mehrere Leerzeichen
- Leerzeichen am Zeilenanfang wird übergangen

Quelldatei (L^AT_EX)

```

1 %Absatz.tex
2 \documentclass[a4paper,11pt]{article}
3 \usepackage{fullpage}
4 \usepackage[latin1]{inputenc}
5 \usepackage[ngerman]{babel}
6
7 \begin{document}
8 Dieser Text      steht in einer
9 Zeile! Man sieht, dass
10 mehrere Leerzeichen und Zeilenumbrüche
11 ignoriert werden.
12
13 Und jetzt folgt
14 ein neuer
15 Absatz.
16 \end{document}

```

Ausgabe-Datei (PDF)

Dieser Text steht in einer Zeile! Man sieht, dass mehrere Leerzeichen und Zeilenumbrüche ignoriert werden.

Und jetzt folgt ein neuer Absatz.

- [Beispiel 5: Absatz](#)

Leerzeichen nach Befehlen

- Leerzeichen nach parameterlosen Befehlen werden übergangen (nur als Befehlsende gedeutet)
 - `\LaTeX` ist super = L^AT_EX ist super
 - `\LaTeX{}` ist super = L^AT_EX ist super
 - `\LaTeX\` ist super = L^AT_EX ist super
 - `\LaTeX~` ist super = L^AT_EX ist super

Sonderzeichen

- Standard-ASCII wird 1:1 zeichenweise ausgegeben:
 - Ausnahme: #, \$, %, &, -, {, }, \, ~
 - Diese Zeichen haben spezielle Funktionen in L^AT_EX: T_EX-Steuerzeichen
 - Stattdessen: `\#`, `\$`, `\%`, `\&`, `\-`, `\{`, `\}`, `\textbackslash`, `\textasciitilde`
- Anführungszeichen "vermeiden:
 - stattdessen `"Text in Anführungszeichen"` oder `\glqq Text in Anführungszeichen\grqq` (dt.)

Silbentrennung

- Silbentrennung erfolgt i.d.R. automatisch
 - `\usepackage[ngerman]{babel}`
- Manchmal manuelle Silbentrennung nötig, weil
 - L^AT_EX falsch trennt
 - L^AT_EX nicht weiß, wie es trennen soll
 - ⇒ Text über Rand hinaus, im LOG-File: `Overfull hbox`
 - `\-` gibt L^AT_EX optionale Trennung an, z.B.: `Sil\-\ben\-\tren\-\nung`
- [Beispiel 7: Automatische Silbentrennung](#)
- [Beispiel 8: Manuelle Silbentrennung](#)

Zeilenumbruch

- Manuell mittels `\\` oder `\newline` oder `\linebreak`
 - Zeile links-bündig bei `\\` oder `\newline`
 - Zeile links-rechts-bündig bei `\linebreak`
- [Beispiel 9: Zeilenumbruch](#)

Seitenumbruch

- Manuell mittels `\newpage` oder `\clearpage`
 - `\clearpage` ist rigoroser (später genauer!)

Textausrichtung

- Standardmäßig verwendet L^AT_EX Blocksatz
- `center`-Umgebung: zentriert Text
- `flushright`-Umgebung: Rechtsbündig
- `flushleft`-Umgebung: Linksbündig
- [Beispiel 10: Textausrichtung](#)

Lieber aus ganzem Holz
eine Freundschaft
als eine geleimte Feindschaft (Friedrich Nietzsche)

Lieber aus ganzem Holz
eine Freundschaft
als eine geleimte Feindschaft (Friedrich Nietzsche)

Lieber aus ganzem Holz
eine Freundschaft
als eine geleimte Feindschaft (Friedrich Nietzsche)

Abbildung: Ausgabe-Datei Beispiel 10

Kleine Schriftkunde

Hervorhebungen

normal:	<code>\textrm{text}</code> oder <code>{\rm text}</code>
fett:	<code>\textbf{text}</code> oder <code>{\bf text}</code>
<i>kursiv:</i>	<code>\textit{text}</code> oder <code>{\it text}</code>
<i>hervorgehoben:</i>	<code>\emph{text}</code> oder <code>{\em text}</code>
sans-serif	<code>\textsf{text}</code> oder <code>{\sf text}</code>
typewriter:	<code>\texttt{text}</code> oder <code>{\tt text}</code>
Kapitälchen:	<code>\textsc{text}</code> oder <code>{\sc text}</code>
<u>unterstrichen:</u>	<code>\underline{text}</code>

- **ACHTUNG:** Es ist nicht alles kombinierbar.

Schriftgröße

- stets relativ zur Schriftgröße des Dokuments
- Schriftgrößen der Größe nach geordnet:
 - `\tiny`, `\scriptsize`, `\footnotesize`, `\small`
 - `\normalsize` gemäß `\documentclass`
 - `\large`, `\Large`, `\LARGE`, `\huge`, `\Huge`

Gliederungsebenen - Unterteilung eines Dokuments in Kapitel und hierarchische Unterabschnitte

- In **report** und **book** gibt es standardmäßig folgende Gliederungsebenen (inkl. Nummerierung und Überschriften):
 - `\chapter{title}`
 - `\section{title}`
 - `\subsection{title}`
 - `\subsubsection{title}`
 - `\paragraph{title}`
 - `\subparagraph{title}`
- **Wichtig:** Hierarchie der Befehle muss eingehalten werden!
- Bei **article** entfällt `\chapter`
- Will man nur Überschriften ohne Nummerierung, verwende
 - `\chapter*{title}`, `\section*{title}`, `\subsection*{title}`, etc.
 - kein Eintrag in das Inhaltsverzeichnis (siehe nächste Folie)
- **Beispiel 11:** Gliederung
- **Beispiel 12:** Gliederung und mehr

Inhaltsverzeichnis

- Mittels `\tableofcontents` wird **automatisch** ein Inhaltsverzeichnis erstellt
 - Erzeugt zusätzliche TOC-Datei (Table of Contents)
 - Wird beim nächsten L^AT_EX-Durchlauf automatisch eingebunden
 - Benötigt 2× L^AT_EX-Durchläufe, um aktuell zu sein
- Reine Überschriften werden nicht eingetragen
 - `\chapter*{title}`, `\section*{title}`, `\subsection*{title}`, etc.

Was sind T_EX und L^AT_EX?

Das erste L^AT_EX-File

Dokument-Klassen und das Einbinden von Paketen

Elementarer Text

Mathematische Formeln

Referenzen

Makros

Zähler

Umgebungen

Tabellen

Bilder

Literaturverzeichnis

Absatzlayout

- `\setlength{\parindent}{0pt}`
 - Einrückung der ersten Absatzzeile auf 0pt
 - Alternativ `\noindent` vor Absatz schreiben
- `\setlength{\baselineskip}{1.5\baselineskip}`
 - Zeilenabstand auf $1\frac{1}{2}$ setzen
- `\setlength{\parskip}{2pt}`
 - Abstand zwischen zwei Absätzen festlegen

Manuelle Einrückungen

- horizontal:
 - `\hspace{5mm}` = 5mm horizontaler Abstand
 - horizontale Abstände relativ zur Schriftgröße: `\quad`, `\quad`, `\,`
 - `\hfill` = Zeile auffüllen
- vertikal:
 - `\vspace{5mm}` = 5mm vertikaler Abstand
 - vertikale Anstände relativ zur Schriftgröße: `\smallskip`, `\medskip`, `\bigskip`
 - `\vfill` = Seite auffüllen

Mathematische Formeln

Quelldatei (L^AT_EX)

```

1 formeln.tex
2 \documentclass[a4paper, 12pt]{article}
3 \usepackage{fullpage}
4
5 \begin{document}
6
7 \noindent
8 Per Induktion zeigt man
9  $\frac{n(n+1)}{2} = \sum_{j=1}^n j$ .
10 Als abgesetzte Formel liest sich das
11 \begin{equation}
12 \frac{n(n+1)}{2} = \sum_{j=1}^n j
13 \end{equation}
14 Ein elementares mehrzeiliges Beispiel:
15 \begin{eqnarray*}
16 \sum_{j=1}^2 j &=& 3, \\
17 \sum_{j=1}^3 j &=& 6, \\
18 \sum_{j=1}^4 j &=& 10. \\
19 \end{eqnarray*}
20 \end{document}

```

Beispiel 13: Mathematische Formeln

Ausgabe-Datei (PDF)

Per Induktion zeigt man

$$\frac{n(n+1)}{2} = \sum_{j=1}^n j.$$

Als abgesetzte Formel liest sich das

$$\frac{n(n+1)}{2} = \sum_{j=1}^n j \quad (1)$$

Ein elementares mehrzeiliges
Beispiel:

$$\sum_{j=1}^2 j = 3,$$

$$\sum_{j=1}^3 j = 6,$$

$$\sum_{j=1}^4 j = 10.$$

Mathematische Formeln

- Formeln im Text `$formel$` oder `math`-Umgebung
- Einzeilige, abgesetzte Formeln mit Nummer
 - z.B. `equation`-Umgebung
- Einzeilige, abgesetzte Formeln ohne Nummer
 - z.B. `displaymath`-Umgebung oder `$$formel$$`
- Mehrzeilige, abgesetzte Formeln mit Nummer
 - z.B. `align`-Umgebung oder `eqnarray`-Umgebung
 - zusätzlicher Stern `*` unterdrückt Nummerierung
 - Ausrichtung der Formeln mit `&`

Klammern

- Etliche Varianten, z.B.
 - runde Klammern (...) mittels ()
 - eckige Klammern [...] mittels []
 - geschwungene Klammern {...} mittels \{...\}
 - Absolutbetrag |...| mittels | oder \vert ...\vert
 - Norm ||...|| mittels \|...\|
- größere Größe der Klammern händisch wählbar
 - Präfix \big, \Big, \bigg, \Bigg vor Klammer
 - z.B. $\big((x+1)(x-1)\big)^2 = (x^2-1)^2$
 $\implies ((x+1)(x-1))^2 = (x^2-1)^2$
- oder Größe automatisch von L^AT_EX wählbar
 - Präfix \left, \right vor Klammer
 - jeder \left braucht ein \right
 - ggf. nur \right, falls nur links Klammer sein soll

Mathematische Sonderzeichen

- De facto alles vorhanden (Pakete einbinden!)
 - `\usepackage{latexsym}` `\usepackage{amssymb}`

Exponenten und Indizes

- $a^{x+y} \neq a^x + y \neq a^{x+y}$ $\implies a^x + y \neq a^{x+y}$
- $x_{\ell+1} := x_{\ell} + x_{\ell-1}$ $\implies x_{\ell+1} := x_{\ell} + x_{\ell-1}$

Brüche und Wurzeln

- $\frac{1}{n+1} \neq \frac{1}{n(n+1)}$ $\implies \frac{1}{n+1} \neq \frac{1}{n(n+1)}$
- $\frac{\partial f}{\partial x_j}$ $\implies \frac{\partial f}{\partial x_j}$
- $(\sqrt{x})^{1/3} = x^{1/6} = \sqrt[6]{x}$ $\implies (\sqrt{x})^{1/3} = x^{1/6} = \sqrt[6]{x}$

Mengen

- $y \in \{f(x) \mid x > 0\}$ $\implies y \in \{f(x) \mid x > 0\}$
- $\in, \ni, \cup, \bigcup, \cap, \bigcap$
- \backslash
- $\subset, \subseteq, \subsetneq$
- $\supset, \supseteq, \supsetneq$

Gleichheit und Ungleichheit

- $=, <, >, \neq, \leq, \leqneq, \geq, \geqneq$

Mathematische Funktionen

- `\exp`, `\log`, `\ln`, `\arg`
- Trigonometrische Funktionen z.B. `\sin`, `\arccos`, `\sinh`
- `\sup`, `\max`, `\inf`, `\min`
- `\lim`, `\limsup`, `\liminf`

- $\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$

- $\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$

- $$\lim_{x \rightarrow 0} \frac{\sin x}{x} = 1$$

Summe, Produkt, Integral

Code-Auszug (L^AT_EX)

```

 $\sum_{j=1}^n j = \frac{n(n+1)}{2}$  bzw.
 $\sum_{j=1}^n j = \frac{n(n+1)}{2}$ 

\vspace{1cm}

 $\prod_{j=1}^{\infty} j = 1 \cdot 2 \cdot 3 \cdots$  bzw.
 $\prod_{j=1}^{\infty} j = 1 \cdot 2 \cdot 3 \cdots$ 

\vspace{1cm}

 $\int_0^{\pi/2} \cos(x) dx = 1$  bzw.
 $\int_0^{\pi/2} \cos(x) dx = 1$ 

```

Ausgabe-Datei (PDF)

$$\sum_{j=1}^n j = \frac{n(n+1)}{2} \text{ bzw.}$$

$$\sum_{j=1}^n j = \frac{n(n+1)}{2}$$

$$\prod_{j=1}^{\infty} j = 1 \cdot 2 \cdot 3 \cdots \text{ bzw.}$$

$$\prod_{j=1}^{\infty} j = 1 \cdot 2 \cdot 3 \cdots$$

$$\int_0^{\pi/2} \cos(x) dx = 1 \text{ bzw.}$$

$$\int_0^{\pi/2} \cos(x) dx = 1$$

Kalligraphische Großbuchstaben

- \mathcal{A} , \mathcal{B} , \mathcal{C} etc.

Griechische Symbole

- α , β , γ , δ , ϵ etc.
- Γ , Δ

Logische Quantoren

- $\forall x > 0: x^2 > 0 \implies \forall x > 0: x^2 > 0$
- $\forall T \text{ Topf } \exists D \text{ Deckel} \implies \forall T \text{ Topf } \exists D \text{ Deckel}$

Blackboard-Großbuchstaben

- `\usepackage{amssymb}` erforderlich!
- \mathbb{N} , \mathbb{Z} , \mathbb{R} , \mathbb{C} etc.

Vektoren & Matrizen

Code-Auszug (L^AT_EX)

```
X=
\left(
  \begin{array}{ccc}
    x_{11} & x_{12} & \ldots \\
    x_{21} & x_{22} & \ldots \\
    \vdots & \vdots & \vdots
  \end{array}
\right)
```

Ausgabe-Datei (PDF)

$$X = \begin{pmatrix} x_{11} & x_{12} & \dots \\ x_{21} & x_{22} & \dots \\ \vdots & \vdots & \dots \end{pmatrix}$$

- **array**-Umgebung für Matrizen und Vektoren (= Matrix mit einer Spalte)
 - beliebig viele Zeilen
 - Zeilenumbruch jeweils mit `\\`
 - Anzahl Spalten + Ausrichtung muss angegeben werden hier: 3 Spalten mit mittiger Ausrichtung: `{ccc}`
 - Ausrichtung: mittig `{c}`, links `{l}`, rechts `{r}`
- **array**-Umgebung ist ein Teil einer mathematischen Formel
 - z.B. `$$...$$`, `equation`-Umgebung
- **array**-Umgebung auch für Fallunterscheidung
 - Verwende `\left\{` mit `\right`

Referenzen

- In mathematischen Absätzen gibt es häufig Referenzen
 - auf Formeln, z.B. siehe Formel (2.7)
 - auf Seiten, z.B. in Formel (2.7) auf Seite 10
 - auf Bilder, z.B. siehe Abbildung 2.3
 - auf Abschnitte, z.B. siehe Kapitel 3
 - auf Sätze, z.B. siehe Satz 2.4
- **Wichtig:** Referenzen werden in L^AT_EX nicht hart kodiert!
- **Hilfreich:** `\usepackage{showkeys}` zeigt Referenzen & Label an
 - zum Schreiben des Dokuments sinnvoll

Vorgehen

1. Voraussetzung: Man setzt Label
 - durch `\label{name}`
 - L^AT_EX verknüpft intern das Label `name` mit zuletzt vorausgegangener Zähler-Auswertung
2. Im Text Referenzen einfügen durch
 - `\ref{name}`: nur Zählerausgabe
 - `\eqref{ref}`: Zählerausgabe für Gleichung
 - benötigt `\usepackage{amsmath}`
 - `\pageref{name}`: Ausgabe der Seitenzahl

L_AT_EX-Warnungen

- L_AT_EX speichert Labels in AUX-Datei
- L_AT_EX erkennt, falls Referenzen neu sind
 - LOG-File endet in diesem Fall mit
LaTeX Warning: Label(s) may have changed.
Return to get cross-references right
 - Dann: L_AT_EX-File noch ein mal kompilieren
- L_AT_EX erkennt, falls Label doppelt benutzt wird
 - LaTeX Warning: Label 'X' multiply defined.
 - LOG-File endet in diesem Fall mit
LaTeX Warning: There were multiply-defined labels.
- L_AT_EX gibt Warnung, falls Label unbekannt ist
 - LaTeX Warning: Reference 'X' on page XX.
undefined on input line XXX
 - LOG-File endet in diesem Fall mit
LaTeX Warning: There were undefined references.

Definieren von Makros

- Definition eines neuen Makros mittels
 - `\newcommand{\name}[anz]{definition}`
- Obligatorisch sind
 - Name des Makros `name`
 - Befehlsfolge des Makros `definition`
- Optional ist Anzahl `anz` der obligatorischen Parameter des Makros
 - Fehlt `anz` so ist `\name` parameterlos
 - maximal 9 Parameter, intern `#1, ...#9`
- Beispiele:
 - `\newcommand{\R}{\mathbb{R}}`
 - Aufruf: `\R`
 - Ausgabe: \mathbb{R}
 - `\newcommand{\norm}[1]{\left| \#1\right|}`
 - Aufruf: `\norm{f}`
 - Ausgabe: $\|f\|$
 - `\newcommand{\set}[2]{\big\{ \#1\,,\big| \,, \#2\right\}}`
 - Aufruf: `\set{x\in\R}{f(x)=0}`
 - Ausgabe: $\{x \in \mathbb{R} \mid f(x) = 0\}$
- L^AT_EX passt auf, ob Makro-Name bereits vergeben ist
 - ! LaTeX Error: Command XXX already defined.
 - Altes Überschreiben mittels `\renewcommand`
 - Parameter/Verwendung wie bei `\newcommand`

Warum Makros?

- Lesbarkeit des Codes, insbesondere Mathematischer Formeln
 - `\big\{x\in\mathbb{R}\}`, `\big| \`, `f(x)=0\big\}`
versus
 - `\set{x\in\mathbb{R}}{f(x)=0}`
- Code wird kürzer & übersichtlicher
- Vermeidung von Tippfehlern
- Wiederverwendbarkeit in weiteren Dokumenten
- einfache Anpassung von math. Notation
 - z.B. $\{x \in \mathbb{R} \mid f(x) = 0\}$ vs. $\{x \in \mathbb{R} : f(x) = 0\}$
- Umstellung der Notation im gesamten Dokument durch Änderung einer Zeile

Was sollte man bei der Definition von Makros beachten?

- Sprechende Namen für Makros verwenden
 - `\set` , `\norm`, `\scalarproduct`
- Kurze Namen nur für reine Zeichen z.B.
 - `\bbN`, `\bbZ`, `\bbR` etc. für `mathbb`-Symbole \mathbb{N} , \mathbb{Z} , \mathbb{R}
 - `\cA`, `\cB`, `\cC` etc. für `mathcal`-Symbole \mathcal{A} , \mathcal{B} , \mathcal{C}
 - `\x`, `\y`, `\z` etc. für Vektoren \mathbb{N} , \mathbb{Z} , \mathbb{R}
- Keine Makros zur puren Abkürzung von Tipparbeit
 - Solchen Code kann man später nicht mehr lesen!

Makros mit optionalen Parametern

- Man kann Makros definieren deren erster Parameter optional ist
 - `\newcommand{\name}[anz] [default1]{definition}`
 - name, anz, definition wie bisher
 - Parameter #1 ist optional
 - Übergabe in eckigen Klammern [parameter1]
 - Wert default1, falls nicht gegeben
 - Parameter #2, ...#anz sind obligatorisch
 - Übergabe in Klammern {parameter}

- Beispiel:
 - `\newcommand{\norm}[2] [] {\left \| #2\right \|_{{#1}}}`
 - Aufruf: `\norm[L^2(\Omega)]{f}` Ausgabe: $\|f\|_{L^2(\Omega)}$
 - Aufruf: `\norm{f}` Ausgabe: $\|f\|$
 - `\newcommand{\set}[3] [\Big] {#1 \{\ #2, #1\ |, #3#1\}}`
 - Aufruf: `\set{x\in\mathbb{R}}{f(x)=0}` Ausgabe: $\{x \in \mathbb{R} | f(x) = 0\}$
 - Aufruf: `\set[\Big]{x\in\mathbb{R}}{f(x)=0}\norm{f}` Ausgabe: $\left\{x \in \mathbb{R} | f(x) = 0\right\}$

Vordefinierte Zähler

- Abhängig von der Dokument-Klasse gibt es Zähler für die Gliederung
 - chapter, section, subsection etc.
- Weitere Zähler sind
 - page, equation, figure, table
- Auswertung eines Zählers
 - `\arabic{counter}` = 1, 2, 3, 4 etc
 - `\roman{counter}` = i, ii, iii, iv etc.
 - `\Roman{counter}` = I, II, III, IV etc.
 - `\alpha{counter}` = a, b, c, d (counter ≤ 26)
 - `\Alpha{counter}` = A, B, C, D (counter ≤ 26)
- Zu jedem counter gehört ein Ausgabebefehl `\thecounter`, der u.a. von `\ref` aufgerufen wird
- **Beispiel:** Nummerierung der Gleichung mit Kapitel + Abschnitt + Formel
`\renewcommand{\theequation}{\arabic{chapter}.\arabic{section}.\arabic{equation}}`
- Wertzuweisung eines Zählers
 - `\setcounter{counter}{zahl}`
- Zähler um 1 erhöhen & referenzierbar machen
 - `\refstepcounter{counter}`

Eigene Zähler definieren

- `\newcounter{\newcounter}[oldcounter]`
- Falls obligatorischer Parameter `oldcounter` angegeben wird, wird `newcounter` automatisch durch `\refstepcounter{oldcounter}` auf 0 gesetzt
- **Beispiel:** Sätze kapitelweise nummerieren
 - Satz 1.1, Satz 1.2, ... Satz 2.1 etc.
- Ausgabe des Zählers festlegen
 - `\renewcommand{\thenewcounter}{...}`

Quelldatei (L^AT_EX)

```

1 % zaeler.tex
2 \documentclass[a4paper,12pt]{report}
3 \usepackage{fullpage}
4
5 \newcounter{const}
6 \renewcommand{\theconst}{\arabic{const}}
7
8 \newcommand{\newconst}[1]{%
9 \refstepcounter{const}%
10 C_{\theconst}\label{const:#1}%
11 }
12 \newcommand{\const}[1]{\ref{const:#1}}
13
14 \begin{document}
15 Seien  $\newconst{2}$ ,  $\newconst{1} > 0$ 
16 und es gelte  $\const{2} \leq \const{1}$ 
17 \end{document}

```

Ausgabe-Datei (PDF)

Seien $C_1, C_2 > 0$ und es gelte
 $1 \leq 2$

- **Beispiel 14:** Zaeler.tex

Vordefinierte Zähler bearbeiten

- Standardmäßig zählt `equation` bei der Dokument-Klasse `article` global
- Standardmäßig zählt `equation` bei der Dokument-Klasse `report` oder `book` kapitelweise
- Neu-Definition der Zählerabhängigkeit zum Zurücksetzen auf Null mittels `\numberwithin[format]{counter}{recounter}`
 - `format = \arabic, \roman, \alpha` etc.
 - Standard ist `\arabic`
 - z.B. `\numberwithin{equation}{section}`
 - Nummerierung = `\thesection.\arabic{equation}`
 - Erste Formel in neuer Section hat nun stets Nummer 1
 - benötigt `\usepackage{amsmath}`

Weitere Text-Umgebungen

- Wir kennen bereits `center`, `flushright`, `flushleft`-Umgebung
- für Zitate: `quote`-Umgebung
- als ob Schreibmaschine `verbatim`-Umgebung
- Aufzählungen: `itemize`-Umgebung
 - jeder Punkt wird mit `\item` eingeleitet
 - optional `\item[zeichen]` für anderes Symbol
- Nummerierte Aufzählung: `enumerate`-Umgebung
 - jeder Punkt wird mit `\item` eingeleitet
 - Art der Aufzählung über Zähler manipulierbar
 - `enumi`
 - `enumii`, `enumiii`, `enumiv` bei geschachtelten `enumerate`-Umgebungen
 - `\usepackage{enumerate}` hat mehr Funktionalität
 - Erweiterung der `enumerate`-Umgebungen um optionale Layout-Parameter

Warum Umgebungen?

- Viele Objekte in mathematischen Texten sollten dasselbe Layout haben
 - z.B. Sätze, Definitionen, Beweise
- Umgebungen trennen Inhalt und Layout
 - Code wird lesbarer
 - Layout wird leichter veränderbar

Definition einer Umgebung

- `\newenvironment{name}[anz]{defbegin}{defend}`
- name, anz wie bei `\newcommand`
- defbegin = Was löst `\begin{name}` aus?
- defend = Was löst `\end{name}` aus?
- `\renewenvironment` analog zu `\renewcommand`
- **Beispiel:**
 - `\newenvironment{proof}{\textbf{Beweis.}}{\hfill\textbf{q.e.d.}}`
 - **Eingabe:** `\begin{proof} ... \end{proof}` **Ausgabe:** **Beweis. ...** **q.e.d.**

Wichtige mathematische Pakete

- `amsmath` = Umgebungen, Befehle
 - im Folgenden `\usepackage{amsmath}` notwendig!
- `amsth` = Theorem-Umgebungen
- `amsfonts`, `amssymb` = Schriftarten + Symbole

Praktische Umgebungen

- `matrix`-Umgebung für Vektoren und Matrizen
 - bequemer als `array`-Umgebung, weil man die Anzahl der Spalten nicht angeben muss
 - ansonsten gleiche Syntax
 - zeilenweise Angabe
 - `&` für neue Spalte
 - `\\` für neue Zeile
- `pmatrix`-Umgebung
 - `=\left(\begin{matrix}...\end{matrix}\right)`
- `case`-Umgebung
 - `=\left(\begin{array}[1l]...\end{array}\right)`
- **Beispiel 15:** `array`, `pmatrix` und `binom`

Die align-Umgebung

- Flexibler als `equation`- und `eqnarray`-Umgebung
- Mit (`align`) und ohne (`align*`) Formelnummer
- Erlaubt mehrzeilige Formeln, Zeilenumbruch `\\`
- Ordnet tabellarisch an
 - & neue Spalte
 - Spalten abwechselnd links/rechts ausgerichtet
 - Spaltenpaar links/rechts bildet jeweils Gruppe ohne Anstand
- `\tag{text}` ersetzt Formelnummer durch Text `\notag{text}` unterdrückt Ausgabe der Formelnummer
 - falls nur manche Zeilen einer mehrzeiligen Formel Nummer haben sollen
- In Verbindung mit `split`-Umgebung kann man Formelnummern mehrzeiliger Formeln vertikal zentrieren
- [Beispiel 16](#): `align`-Umgebung

Mathematisch Sätze

- Umgebung für math. Sätze etc. können leicht(!) erstellt werden, d.h. `\newenvironment` hier unnötig!
- `\newtheorem{name}[counter]{text}[supercounter]`
 - Obligatorisch:
 - **name**: Name der neuen Umgebung
 - **text**: Überschrift der neuen Umgebung, z.B. Satz, Lemma
 - Optional:
 - **counter**: falls kein neuer Zähler angelegt werden soll, sondern vorhandener mitbenutzt wird
 - **supercounter**: spezifiziert übergeordneten Zähler, z.B. **section**: wenn Section erhöht, wird **counter** auf 0 gesetzt
 - gleiche Funktion wie `\numberwithin`
- Beispiele:
 - `\newtheorem{satz}{Satz}[section]`
 - Satz-Umgebung
 - Zähler zählt in jeder section neu
 - `\newtheorem{lemma}{Satz}[Lemma]`
 - Satz & Lemma werden gemeinsam nummeriert
 - `\newtheorem{bemerkung}{Bemerkung}[section]`
 - Bemerkungen werden unabhängig nummeriert
 - Zähler zählt in jeder Section neu
- Beispiel 17: `newtheorem.tex`

Die tabbing-Umgebung

- Zur Spaltenweisen Ausrichtung von Text
- \= Markierung setzen
- \kill Zeile nicht ausgeben
 - für Definitionszeile
- \> Textposition auf nächste Markierung setzen

Quelldatei (L^AT_EX)

```
% tabbing.tex
\documentclass[a4paper,12pt]{report}
\usepackage{fullpage}
\usepackage[ngerman]{babel}
\usepackage[applemac]{inputenc}

\begin{document}
\noindent Jetzt kommt eine \texttt{tabbing}-Umgebung:
\begin{tabbing}
% Definition der Tabulator-Stops
\hspace*{25mm} \= \hspace*{3cm} \= \hspace*{5cm} \= \kill
% Der ausgerichtete Text
Spalte 1 \> Spalte 2 \> Spalte 3 \> Spalte 4\\
Text A   \> Text B   \> Text C           \\\
         \> Weiter   \> so               \> !!
\end{tabbing}
\end{document}
```

Ausgabe-Datei (PDF)

Spalte 1	Spalte 2	Spalte 3	Spalte 4
Text A	Text B	Text C	
	Weiter	so	!!

Die tabular-Umgebung

- Benutzung wie array-Umgebung
 - Anzahl Spalten angeben & Ausrichtung
 - mittig {c}, links {l}, rechts {r}
 - Blocksatz mit fester Spaltenbreite p{Breite}
 - vertikale Trennlinie mit Pipe | angeben
 - oder eigene Trennlinie mit
 - Zeilenumbruch mit \\
 - horizontale Trennlinie mit \hline
- kann Trennlinien auch in array-Umgebung nutzen
- Verwende \cline{von-bis}, falls horizontale Linie nur Spalte von bis bis
- Verwende \multicolumn{anz}{style}{text} für Eintrag text über mehrere Spalten
 - anz = Anzahl der betroffenen Spalten
 - style = analog zu tabular-Style, z.B. {l|c|}

Die tabular-Umgebung

Quelldatei (L^AT_EX)

```
% tabular.tex
\documentclass[a4paper,12pt]{report}
\usepackage{fullpage}
\usepackage[ngerman]{babel}
\usepackage[applemac]{inputenc}

\begin{document}
\begin{tabular}{|l| |c| |c| |r|}
\hline
links & mittig & rechts & rechts\\
\hline\hline
1 & 2 & 3 & 4\\
5 & 6 & 7 & 8\\
\hline
\end{tabular}
\end{document}
```

Ausgabe-Datei (PDF)

links	mittig	rechts	rechts
1	2	3	4
5	6	7	8

- [Beispiel 18: tabular.tex](#)

Die table-Umgebung

- I.d.R. soll Tabelle nicht Teil von Text sein, sondern herausgehoben mit Unterschrift und Nummer
 - verwende `table`-Umgebung
- `\caption` gibt der Tabelle eine Unterschrift
- `table`-Umgebung erzeugt eine so genanntes **float object**
 - wird von L^AT_EX automatisch platziert
 - wird intern in Liste eingetragen und sobald als möglich gesetzt
 - First-In-First-Out Prinzip
 - `\clearpage` arbeitet Float-Liste ab, danach Seitenumbruch (`\newpage` = neue Seite)
- Präferenz für Platzierung kann optional als Liste angegeben werden
 - z.B. `\begin{table}[!thpb]`
 - `!` = egal, ob es vernünftig scheint
 - `t` = top
 - `h` = here
 - `t` = page = Extraseite nur mit floats
 - `b` = bottom
 - wird in der angegebenen Reihenfolge von L^AT_EX in Erwägung gezogen
- erzeugt Tabellen-Verzeichnis
 - Einträge werden aus `\caption{...}` übernommen
 - erstes `latex name.tex` erzeugt `name.lot`
 - zweites `latex name.tex` bindet Verzeichnis ein
 - Falls Unterschrift zu lang ist, Kurztitel festlegen
 - `\caption[kurztitel]{unterschrift}\verb`

Bilder einbinden

- Voraussetzung: `\usepackage[dvips]{graphicx}`
- Einbinden: `\includegraphics[options]{filename}`
 - Optionale Parameter sind:
 - `width = unit`: Breite festlegen (& ggf. skalieren)
 - `height = unit`: Höhe festlegen (& ggf. skalieren)
 - `scale = num`: Bild skalieren
 - `angle = unit`: Bild drehen (math. pos. Grad)

Quelldatei (L^AT_EX)

```
%includegraphics.tex
\documentclass[a4paper,12pt]{report}
\usepackage{fullpage}
\usepackage[ngerman]{babel}
\usepackage[applemac]{inputenc}
\usepackage[dvips]{graphicx}

\begin{document}
\includegraphics[width=0.75\textwidth]{Grafiken/UniLogo.pdf}

\includegraphics[width=0.75\textwidth,angle=25]{Grafiken/UniLogo.pdf}

\includegraphics[width=3cm,angle=-25]{Grafiken/TeXLogo.pdf}
\end{document}
```

Ausgabe-Datei (PDF)



ulm university universität

uulm



ulm university universität

uulm

TeX-Latex

Die figure-Umgebung

- Verwendung von `figure` analog zu `table`
- `\listoffigures` erzeugt Abbildungsverzeichnis
 - erzeugt Datei `name.laf`

Quelldatei (L^AT_EX)

```
%figure.tex
\documentclass[a4paper,12pt]{report}
\usepackage{fullpage}
\usepackage[ngerman]{babel}
\usepackage[applemac]{inputenc}
\usepackage[dvips]{graphicx}

\begin{document}
\listoffigures
\clearpage
\begin{figure}[t]
\begin{center}
\includegraphics[width=0.75\textwidth]{Grafiken/UniLogo.pdf}
\caption{Logo der Universität Ulm}
\label{fig::bsp1}
\end{center}
\end{figure}
\begin{figure}[t]
\begin{center}
\includegraphics[width=0.4\textwidth]{Grafiken/UniLogo.pdf}
\includegraphics[width=0.4\textwidth]{Grafiken/TeXLogo.pdf}
\caption{Links: Logo der Universität Ulm,
rechts: \TeX-\LaTeX-Logo}
\label{fig::bsp2}
\end{center}
\end{figure}
\end{document}
```

Ausgabe-Datei (PDF)



ulm university universität

uulm

Abbildung: Logo der Universität Ulm



ulm university universität

uulm

TeX-LaTeX

Abbildung: Links: Logo der Universität Ulm, rechts: T_EX-L^AT_EX-Logo

Export von Bildern aus Matlab

Matlab

- `print` druckt alle Figure aus
- `print(opt1, opt2, ..., name)` erzeugt File name
 - optionale Strings `opt` geben an z.B. Auflösung: `'-r200'` = 200dpi (Std. 150dpi) z.B. Dateityp:
 - `'-deps'` = EPS s/w
 - `'-depsc'` = EPS farbig
 - `'-djpeg90'` = JPG, Qualität 90
 - `'-dpdf'` = PDF

```

1 % demoprint.m
2 x = -6:.01:6
3 y = exp(-x.^2);
4 z = x.^2/30
5
6 plot(x,y, 'b--')
7 hold on
8 plot(x,z, 'r')
9 text(0,1.05, 'exp(0)=1')
10 hold off
11
12 legend('exp(-x^2)', 'x^2/30')
13 xlabel('Intervall [-6, 6]')
14 ylabel('Funktionswerte')
15 title('Ein kleines Beispiel')
16
17 print('-r600',-depsc', 'demoprint.eps')
18 print('-r600',-djpg', 'demoprint.jpg')
```

T_EX-L^AT_EX

- Ersetze Matlab-Text durch L^AT_EX-Text:


```
\usepackage{psfrag}
```
- `\psfrag{ps}[posTeX][posPS][scale][angle]{tex}`
 - ersetzt Text `ps` in eps/ps-File durch `tex` in L^AT_EX
 - Bezugspunkte `posTeX` & `posPS` im `ps`- & `tex`-Text
 - horizontal: (ℓ , r , c), vertikal (t , b , c , B) (Std. ℓB)
 - optional: Skalierung und Winkel
- vor `\includegraphics{filename}`

```

% demoprint.tex
\documentclass[a4paper,12pt]{article}
\usepackage{fullpage}
\usepackage[ngerman]{babel}
\usepackage[applemac]{inputenc}
\usepackage[dvips]{graphicx}
7
8 \begin{document}
9 \begin{figure}[t]
10 \begin{center}
11 \includegraphics[width=3cm]{demoprint.eps}
12 \caption{Wir binden \texttt{demoprint.eps}}
13 \label{fig::bsp1}
14 \end{center}
15 \end{figure}
16 \end{document}
```

Literaturverzeichnis anlegen

- thebibliography-Umgebung:
 - startet mit `\begin{thebibliography}{string}`
 - `string` gibt nur maximale Länge von Markern an
 - Einträge mittels `\bibitem[marker]{label}`
 - `label`: definiert Label zum zitieren
 - `marker`: optional, gibt Kennung für Eintrag
 - falls `marker` fehlt, wird Nummer zugewiesen
- Zitieren im Text mittels
 - `\cite[string]{referenz}`
 - `referenz = label` vom `\bibitem`
 - `string`: optional, wird zusätzlich ausgegeben, z.B. expliziter Verweis auf einen Satz
 - Listen `\cite{ref1, ref2, ...}` sind erlaubt

Grundsätzliches

- Einträge im Literaturverzeichnis einheitlich!
 - alle Vornamen abkürzen oder ausschreiben
 - gleiches Layout für alle Einträge
 - insb. einheitliche Groß-Kleinschreibung
 - am Ende jedes Eintrags ein Punkt oder nicht
- gewisse Sortierung
 - alphabetisch nach Erstautor (Autorenreihenfolge nicht ändern!)
 - chronologisch nach Veröffentlichungsjahr
 - chronologisch nach Reihenfolge des Zitierens
- korrekte Abkürzungen bei Zeitschriften (<http://www.ams.org/mathscient>)

Weitere Fragen?

Katharina.Becker-Steinberger@uni-ulm.de

