

# Advances in new technologies concerning recentness and related research areas

## MASTERARBEIT

KIT – KARLSRUHER INSTITUT FÜR TECHNOLOGIE  
FRAUNHOFER IOSB – FRAUNHOFER-INSTITUT FÜR OPTRONIK,  
SYSTEMTECHNIK UND BILDAUSWERTUNG

**Hans Mustermann**

42. Hintester 2038

Verantwortlicher Betreuer:	Prof. Dr.-Ing. habil. Jürgen Beyerer
Betreuende Mitarbeiter:	Dr.-Ing. Dieter Doktorgrad Michael Mastergrad, M.Sc.



## Erklärung der Selbstständigkeit

Hiermit versichere ich, dass ich die Arbeit selbständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, die wörtlich oder inhaltlich übernommenen Stellen als solche kenntlich gemacht habe und die Satzung des Karlsruher Instituts für Technologie zur Sicherung guter wissenschaftlicher Praxis in der gültigen Fassung beachtet habe.

Karlsruhe, den 42. Hintester 2038

---

(Hans Mustermann)



# Abstract

If the thesis is written in English, the abstract has to be present *both* in English and as a German translation.



## **Kurzfassung**

Falls die Abschlussarbeit auf Deutsch geschrieben wird, genügt die deutsche Kurzfassung.





# Notation

## Allgemeine Bezeichner

$\alpha, \dots, \omega$	Skalare
$a, \dots, z$	Skalar, Vektor, Funktionssymbol (oder Realisierung einer Zufallsvariablen)
$\mathbf{a}, \dots, \mathbf{z}$	Zufallsvariable (skalar bzw. vektoriell)
$\hat{\mathbf{a}}, \dots, \hat{\mathbf{z}}$	Schätzer für jeweilige Variable als Zufallsgröße
$\hat{a}, \dots, \hat{z}$	Realisierter Schätzer für jeweilige Variable
$A, \dots, Z$	Matrix
$\mathbf{A}, \dots, \mathbf{Z}$	Matrix als Zufallsgröße
$\mathcal{A}, \dots, \mathcal{Z}$	Menge
$\mathfrak{A}, \dots, \mathfrak{Z}$	Mengensystem

## Spezielle Bezeichner

$t$	Spezielle Bezeichner mit konkreter Bedeutung in dieser Arbeit, z. B. $t$ Zeitindex
-----	------------------------------------------------------------------------------------

## Allgemeine Mengen

$\mathbb{C}$	Menge der komplexen Zahlen
$\mathbb{H}$	Poincaré Halbebene
$\mathbb{N}$	Menge der natürlichen Zahlen (ohne Null)
$\mathbb{N}_0$	Menge der natürlichen Zahlen mit Null
$\mathbb{Q}$	Menge der rationalen Zahlen
$\mathbb{Q}^{>0}, \mathbb{Q}^{<0}$	Menge der positiven bzw. negativen rationalen Zahlen
$\mathbb{R}$	Menge der reellen Zahlen
$\mathbb{R}^{>0}, \mathbb{R}^{<0}$	Menge der positiven bzw. negativen reellen Zahlen
$\mathbb{Z}$	Menge der ganzen Zahlen

## Spezielle Symbole

$\mathfrak{N}(\mu, \sigma^2)$	Normalverteilung mit Erwartungswert $\mu$ und Varianz $\sigma$
$\mathfrak{F}_{r,s}$	Fisher-Verteilung mit $r$ Zähler- und $s$ Nennerfreiheitsgraden
$t_s$	Student- $t$ -Verteilung mit $s$ Freiheitsgraden
$\delta_\xi$	Ein-Punkt-Maß an der Stelle $\xi$
$\chi_s^2$	$\chi^2$ -Verteilung mit $s$ Freiheitsgraden

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Struktur der Vorlage</b>	<b>3</b>
<b>3</b>	<b>Programme und Werkzeuge</b>	<b>5</b>
3.1	Zeichenkodierung und Umlaute . . . . .	5
3.2	Kommandozeilentools . . . . .	5
3.3	Ubuntu Pakete . . . . .	5
3.4	IDE Configuration . . . . .	6
3.5	TeXnicCenter – Integrierte Entwicklungsumgebung für Windows . . . . .	6
3.6	TeXStudio unter Linux . . . . .	9
3.7	TikzEdt . . . . .	9
3.8	Best Practices MiKTeX . . . . .	9
3.8.1	File 'inconsolata.sty' not found . . . . .	10
3.8.2	Biber und Miktex2.9 x64 . . . . .	10
3.8.3	Windows API error 5: Access is denied . . . . .	10
3.8.4	package xparse error: support package l3kernel too old . . . . .	10
<b>4</b>	<b>Etwas LaTeX</b>	<b>13</b>
4.1	Dokumentationsquellen . . . . .	13
4.2	Referenzen . . . . .	13
4.3	Fließumgebungen (Floats) . . . . .	14
4.4	Tabellen . . . . .	15
4.5	Bilder, Grafiken und Diagramme . . . . .	16
4.6	Theoreme und Listings . . . . .	22
4.7	Literaturangaben . . . . .	23
4.8	Stichwortverzeichnis (Index) und Glossar . . . . .	24
4.9	Mathematik . . . . .	25
4.10	Sonstiges (URLs, Anführungszeichen, Sprache und Randnotizen) . . . . .	26
4.10.1	URLs . . . . .	26

4.10.2	Anführungszeichen . . . . .	26
4.10.3	Sprachumschaltung (Deutsch, Englisch, etc.) . . . . .	26
4.10.4	Randnotizen . . . . .	27
<b>Literatur</b>		<b>29</b>
<b>Tabellenverzeichnis</b>		<b>31</b>
<b>Abbildungsverzeichnis</b>		<b>33</b>
<b>Theoremverzeichnis</b>		<b>35</b>
<b>Listings</b>		<b>37</b>
<b>Index</b>		<b>39</b>
<b>Glossar</b>		<b>43</b>

# 1 Einleitung

Am Lehrstuhl für Interaktive Echtzeitsysteme (IES) des Karlsruhe Instituts für Technologie (KIT) wurde von Philipp Wook die erste LaTeX-Vorlage zur Erstellung von wissenschaftlichen Arbeiten erstellt. Diese basierte ihrerseits auf der Vorlage von Matthias Pospiech von der Leibniz Universität Hannover. Die von Matthias Pospiech und durch Philipp Wook stark erweiterte Vorlage, hatte den Anspruch die „eierlegende Wollmilchsau“ zu sein und möglichst alle Anwendungsfälle abzudecken.

Leider hatte diese Vorlage – da sie ständig um zusätzliche Pakete erweitert wurde – auch zwei meiner Meinung nach entscheidende Nachteile:

- Es setzte auf den alten BibLatexs-Paketen auf anstatt des neueren bibLatex-Ökosystems. Dadurch war eine durchgängige UTF-8-Unterstützung nicht möglich und die ein oder andere Konstellation von Umlauten hat immer mal wieder „geknallt“.
- Zum Erstellen von Vektorgrafiken mit einer hohen Druck- und Typografiequalität gibt es TikZ. Die alte Vorlage unterstützte zwar grundlegendes TikZ, aber bei vielen TikZ-Zusatzpaketen kam es zu Inkompatibilitäten mit anderen Paketen.

Diese Vorlage ist quasi ein Neudesign der Vorlage von Philipp Wook. Allerdings hat sie intern nicht mehr allzu viele Gemeinsamkeiten. Im Wesentlichen wurde versucht das Layout nachzuahmen, um eine halbwegs konsistentes Erscheinungsbild zwischen Dokumenten mit alter und neuer Vorlage zu erreichen.

Natürlich ist so ein Neustart nicht ohne Nachteile möglich. Gegenwärtig ist diese Vorlage noch sehr „schlank“ und viele Möglichkeiten der alten Vorlage sind noch nicht wieder nachgebaut. Dies wird in Zukunft und bei Bedarf geschehen.

Einige Möglichkeiten der alten Vorlage werden aber nie wieder ihren Weg in diese Vorlage finden. Dies betrifft alle LaTeX-Pakete die in irgendeiner Weise PostScript benötigen. Um das volle Potential von UTF-8, BibLaTeX und TikZ ausnutzen zu können, ist die Verarbeitungskette auf

LaTeX-Quellcode  $\xrightarrow{\text{pdf\texttt{latex}}}$  PDF

reduziert worden. Umwege wie

$$\text{LaTeX-Quellcode} \xrightarrow{\text{latex}} \text{DVI} \xrightarrow{\text{dvi2ps}} \text{PS} \xrightarrow{\text{ps2pdf}} \text{PDF}$$

sind ausgeschlossen. Das bedeutet insbesondere, dass alle Optionen, die das bekannte Paket `pstricks` bietet nicht mehr zur Verfügung stehen. Allerdings bietet hier TikZ immer eine Ersatzlösung an. Das einzige, was definitiv gar nicht mehr funktioniert und auch nicht durch TikZ nachgestellt werden kann, ist das direkte Einbinden von PostScript- bzw. EPS-Abbildungen. Diese müssen nun zunächst durch externe Tools in PDF konvertiert werden.

## 2 Struktur der Vorlage

Die Vorlage ist wie aus Tabelle 2.1 zu entnehmen aufgebaut.

Datei/Verzeichnis	Bedeutung	Benutzerinteraktion
./thesis.tcp	TeXnicCenter-Projektdatei	indirekt
./main.tex	TeX-Hauptdatei	nur input-Direktiven
./literature.bib	BibLaTeX-Quelldatei	ja
./figures-src/*	TikZ-Zeichnungen	ja
./images/*	Binärformat-Bilder	ja
./content/*	LaTeX-Kapitel	ja
./preamble.tex	Paketkonfigurationen	nein
./preamble-final-setup.tex	Paketkonfigurationen	nein
./preamble/*	Paketkonfigurationen	nein
./logos/*	Institutslogos	nein
./figures-compiled/*	Temporäre Kompilate	nur löschen

Tabelle 2.1: Dateien und Verzeichnisse der Vorlage

Die Datei ./thesis.tcp ist ein TeXnicCenter-Projekt und sollte in TeXnicCenter geladen werden.

Jedes Kapitel der Arbeit wird im Verzeichnis ./content/ als separate Datei gespeichert. Es empfiehlt sich als Dateiname das Schema nn-name.tex zu verwenden, wobei nn die Nummer des Kapitels ist, sodass die Dateien in der semantisch richtigen Reihenfolge sortiert angezeigt werden. Per input-Direktive werden diese Dateien in der Datei ./main.tex eingebunden. Zudem kann das Titelblatt durch Verwendung von content/00a-titlepage-hska und content/00b-declaration-hska auf das Layout der Hochschule angepasst werden. Andere Veränderungen sollte man an der Datei ./main.tex unterlassen.

Bilder bzw. Zeichnungen werden auf zwei Arten eingebunden. Handelt es sich um Bilder, die mit einem externen Programm erzeugt und in einem üblichen Binärformat abgespeichert wurden (PNG, JPEG, TIFF, PDF, etc.) so werden diese im Verzeichnis ./images/ abgelegt. Werden (Vektor-)Zeichnungen per TikZ erzeugt, so wird die entsprechende Quelldatei im Verzeichnis ./figures-src/ gespeichert. Werden des Kompilierens werden für jede TikZ-Zeichnung im

Verzeichnis `./figures-compiled/` mehrere Dateien erzeugt. Der Inhalt der Verzeichnisses kann gefahrlos gelöscht werden. **Bitte beachten: Das Verzeichnis `./figures-compiled/` muss einmalig manuell erzeugt werden.**



## 3 Programme und Werkzeuge

### 3.1 Zeichenkodierung und Umlaute

Bei Erstellen neuer Dateien bzw. Öffnen und Speichern bereits vorhandener Dateien ist darauf zu achten, dass stets UTF-8 als Zeichenkodierung verwendet wird. Dies gilt insbesondere auch für Quellen des Literaturverzeichnisses (BIB-Dateien).

Umlaute und Zeichen mit Akzent werden in den Quelldateien direkt als solche eingegeben, also direkt Ä, ä, ß, é, usw. nicht "A, "a, "ss, 'e.

### 3.2 Kommandozeilentools

Es empfiehlt sich eigentlich die Benutzung einer integrierten Entwicklungsumgebung, wie z. B. TeXnicCenter unter Windows. Diese wird im folgenden Abschnitt besprochen. Sollte man doch mal in die Verlegenheit kommen direkt mit den Kommandozeilentools des TeX-Ökosystems arbeiten zu müssen, so werden nur noch die beiden Tools `pdflatex` und `biber` benötigt. Letzteres erstellt das Literaturverzeichnis. Die Aufrufreihenfolge ist

```
#> pdflatex main.tex
#> pdflatex main.tex
#> biber main
#> pdflatex main.tex
#> pdflatex main.tex
```

Wenn kein Fehler aufgetreten ist, sollte nach dem vierten Durchlauf von `pdflatex` die Warnung `Please re-run latex` verschwunden sein.

### 3.3 Ubuntu Pakete

Unter 17.04 sollte es funktionieren, wenn folgende Pakete installiert wurden. Vermutlich reichen auch weniger Pakete:

```
texlive-bibtex-extra texlive-lang-german texlive-fonts-extra texlive
```

fonts-linuxlibertine biber texmaker (Texmaker ist eine Latex IDE, die aber vermutlich über dependencies schon einige Pakete mitbringt)

### 3.4 IDE Configuration

Wenn das Build-Kommando von der IDE aus angestoßen werden soll, muss diese entsprechend konfiguriert werden. TikZ, welches die schönen Graphen malt, möchte gerne Shell Zugriff, dies muss ihm mit dem Parameter `--enable-write18` (Windows), bzw. `--shell-escape` (Linux) erlaubt werden. Außerdem muss statt `"biblatex %.aux"` für die Referenzen `"biber %"` genutzt werden.

### 3.5 TeXnicCenter – Integrierte Entwicklungsumgebung für Windows

Um nicht obige Kommandos immer per Hand tippen zu müssen, empfiehlt sich die Verwendung einer integrierten Entwicklungsumgebung; unter Windows z. B. TeXnicCenter. Für diese Entwicklungsumgebung existiert bereits eine entsprechende Projektdatei `thesis.tcp`. Als PDF-Betrachter empfiehlt sich insbesondere SumatraPDF, da dieser sehr gut mit TeXnicCenter interagiert und per Doppelklick ein direktes Hin- und Herspringen zwischen Quellcode und (schadhafter) Stelle im fertigen PDF erlaubt. Unter Linux eignet sich die Entwicklungsumgebung Kile.

Da standardmäßig TeXnicCenter von der Verwendung von BibLatex und nicht von BibLaTeX ausgeht, muss die Konfiguration angepasst werden. Über den Menüpunkt Ausgabe → Ausgabeprofile definieren den gleichnamigen Dialog aufrufen. Am einfachsten ist es, dass vorhandene Profil LaTeX ⇒ PDF zu kopieren und z. B. das Duplikat LaTeX ⇒ SumatraPDF + Biber zu nennen. Die Einstellungen sind entsprechend Abbildungen 3.1 und 3.2 zu ändern. Im Register (La)TeX ist das Compilerargument

```
-synctex=-1 -interaction=nonstopmode -max-print-line=140 "%pm" --enable-write18
```

im Register Viewer ist der Anwendungspfad

```
SumatraPDF.exe -inverse-search "\"TeXnicCenter.exe\" /ddecmd \"[goto('%f','%l')]\\""
```

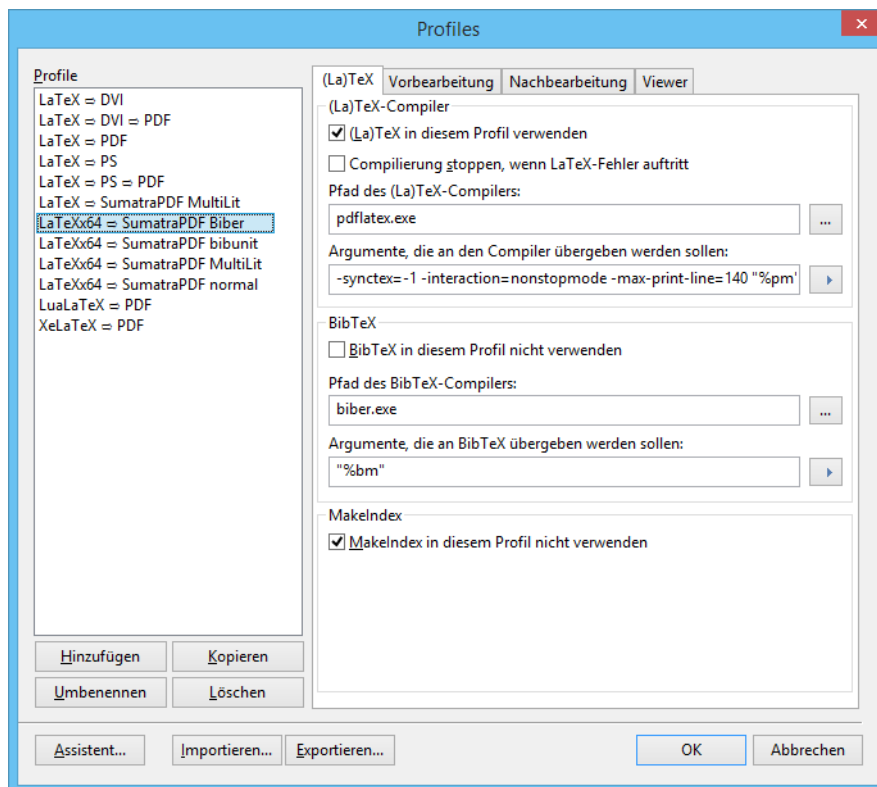


Abbildung 3.1: Neues Ausgabeprofil in TeXnicCenter (1 von 2)

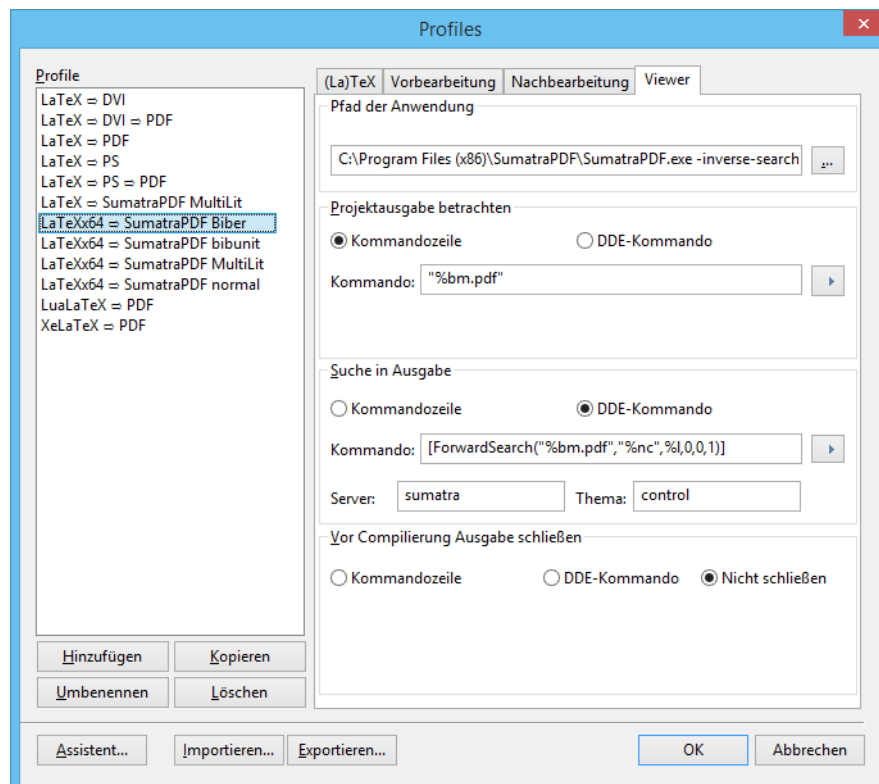


Abbildung 3.2: Neues Ausgabeprofil in TeXnicCenter (2 von 2)

## 3.6 TeXStudio unter Linux

Zur Konfiguration von TeXStudio muss TexLive manuell installiert werden. Siehe dazu [https://wiki.ubuntuusers.de/TeX\\_Live/#Manuell](https://wiki.ubuntuusers.de/TeX_Live/#Manuell). Ansonsten sind die packages nicht kompatibel aufgrund von falschen Versionen. Bei der small Installation werden noch einige Pakete benötigt, siehe Liste im Git. Dabei muss ausschliesslich tlmgr verwendet werden, um Versionskonflikte zu vermeiden. Um damit installierte Programme zu exportieren und damit aus der Konsole nutzen zu können:

```
1 $ export PATH=/usr/local/texlive/2016/bin/x86_64-linux:$PATH
```

Im Git liegt außerdem eine tex-profile fraunhofer--vorlage--thesis-.txsprofile. Dann lässt sich mit dem Kompilieren (einfacher grüner Pfeil) Button das Dokument komplett durchkompilieren inklusive Literaturverzeichnis. Zum Bauen ohne Literatur und direktem Anzeigen dient der Button Erstellen und Anzeigen (doppelter grüner Pfeil). Um das Profil als Standard einzustellen, muss Texstudio als sudo gestartet werden, das Profil unter dem Reiter Optionen geladen werden und anschliessend unter Optionen Aktuelle Einstellungen Speichern gewählt werden.

## 3.7 TikzEdt

Hochqualitative Zeichnungen können mit TikZ erstellt werden. TikZ ist jedoch (analog zu LaTeX) dafür entwickelt worden, den zugehörigen Code per Hand zu schreiben. Einige TikZ-Beispiele befinden sich auch in diesem Dokument. Mit TikzEdt gibt jedoch einen rudimentären WYSIWYG-Editor für Windows. Dieser unterstützt bietet nur eine minimale Unterstützung der TikZ-Kommandos ist jedoch hilfreich, um schnell mit optischen Feedback die Kontrollpunkte im Koordinatensystem festzulegen, bevor dann die weitere Arbeit per Hand erfolgt.

## 3.8 Best Practices MiKTeX

Unter Windows wird üblicherweise MiKTeX zur Verwaltung von LaTeX-Paketen verwendet. Dies funktioniert in den meisten Fällen einwandfrei. Da die Vorlage jedoch, für LaTeX Verhältnisse, sehr neue und moderne Pakete verwendet, kann es manchmal zu Problemen mit Abhängigkeiten kommen. Hier sollen Probleme gesammelt werden und auch wie man sie lösen kann.

### 3.8.1 File 'inconsolata.sty' not found

In der Vorlage wird die Schriftart *inconsolata* verwendet. In MiKTeX 2.9 ist diese Schriftart unter dem Namen *zi4* bekannt. Es besteht die Hoffnung, dass zukünftig die Schriftart wieder unter dem korrekten Namen verfügbar sein wird. In der Zwischenzeit hilft es die Datei `preamble/01-fonts.tex` anzupassen.

```
\usepackage[varqu,varl]{inconsolata}
```

ersetzen durch

```
\usepackage[varqu,varl]{zi4}
```

### 3.8.2 Biber und Miktex2.9 x64

Die Vorlage verwendet BibLaTeX und benötigt dafür das Programm `biber.exe`. In der 64 bit Version ist leider aktuell kein Biber enthalten. Man muss hier<sup>1</sup> die für MiKTeX 2.9 die Version 1.9 runterladen und in den Ordner `Program Files/MiKTeX 2.9/miktex/bin/x64` kopieren.

### 3.8.3 Windows API error 5: Access is denied

Wenn man sich entschieden hat, MiKTeX in der Single-User Version zu installieren (was empfohlen wird, um Probleme mit `l3kernel` zu vermeiden), sind anfänglich die Rechte nicht korrekt gesetzt. Deshalb bricht jeder Versuch die MiKTeX Pakete zu aktualisieren mit der Fehlermeldung: „Windows API error 5: Access is denied“ ab. Um den Fehler zu beheben muss dem eigenen Benutzer Vollzugriff auf den Ordner geben, in dem MiKTeX installiert wurde.

### 3.8.4 package xparse error: support package l3kernel too old

Dies ist wohl der unerfreulichste Fehler weil die genaue Ursache nicht geklärt ist. Im Internet finden sich folgende Lösungsideen:

- Update von MiKTeX über den Paketmanager
- Installation des Pakets „`l3kernel`“
- Installation des Pakets „`l3experimental`“
- Neuinstallation von MiKTeX
- **Neuinstallation von MiKTeX im Single-User Mode**

---

<sup>1</sup> <http://sourceforge.net/projects/biblatex-biber/>

---

Bisher konnte das Problem nur durch eine Neuinstallation von MiKTeX im Single-User Mode gelöst werden. Dabei bitte auch den Punkt **Windows API error 5: Access is denied** beachten.





## 4 Etwas LaTeX

In diesem Kapitel sind einige LaTeX-Dinge zusammengesammelt, die immer wieder auch von Personen verkehrt gemacht werden, die bereits längere Zeit mit LaTeX arbeiten. Dies soll hier also keine Einführung in LaTeX werden, sondern spiegelt nur meine Erfahrung der häufigsten Fehler wieder. Außerdem soll gezeigt werden, wie bestimmte Dinge innerhalb dieser Vorlage gemacht werden.

Grundsätzlich sollte man bei Problemen nie, nimmer, niemals einfach nach einer Lösung im Internet suchen. 95 % der Lösungen im Internet sind bestenfalls falsch, aber eigentlich der größte Mist für den die jeweiligen Autoren angespitzt in den Boden gerammt, im eigenen Saft gegart, gevierteilt und anschließend in Beton gegossen gehören. Aber eigentlich ist es schade um den guten Beton.

### 4.1 Dokumentationsquellen

Statt wahllos im Internet nach Lösungen zu suchen, sucht man direkt auf <http://www.ctan.org/tex-archive/> nach dem Paketnamen und verwendet die dortige Originaldokumentation des Paketautors selbst. Dort zu findende Warnungen sollte man ernst nehmen und nicht machen, auch wenn irgendwo anders behauptet wird, es würde so funktionieren. Es funktioniert NICHT oder nur scheinbar.

Im Rahmen dieser Vorlage sind insbesondere die Dokumentationen aus dem Literaturverzeichnis wärmstens zu empfehlen.

### 4.2 Referenzen

Querverweise sollten nicht mit dem Befehl `\ref{...}` gesetzt werden sondern mit `\cref{...}` und verwandten Befehlen aus dem Paket `cleveref` [Cub13]. Diese Befehle haben den Vorteil nicht nur die Nummer zu referenzieren, sondern auch den Typ mit anzugeben. Hinzukommt eine intelligente Verwendung der Pluralform und Sortierung bei Mehrfachaufzählungen auch unterschiedlichen Typs. Will man bspw. auf zwei Abbildungen und eine Tabelle mit den Marken („Labels“)

- `fig:texniccenter-new-profile-1`
- `tab:files-dirs-of-template`
- `fig:texniccenter-new-profile-2`

verweisen, so schreibt man einfach per Komma getrennt

```
1 \cref{fig:texniccenter-new-profile-1,
2     tab:files-dirs-of-template,
3     fig:texniccenter-new-profile-2}
```

Listing 4.1: Cleveres Referenzieren

und erhält als Resultat „Abbildungen 3.1 und 3.2 sowie Tabelle 2.1“.

### 4.3 Fließumgebungen (Floats)

Fließumgebungen sind bei LaTeX blockbildende Elemente, die nicht an der Stelle erscheinen, an der sie im Quellcode definiert sind sondern aus optischen Gründen an umhergeschoben werden können. Typische Beispiele sind Tabellen, Bilder, längere Codeausschnitte und ähnliche Dinge. Später wird noch auf Bilder und Tabellen im Detail eingegangen, aber an diese Stelle sollen vier Todsünden in Bezug auf Fließumgebungen abgehandelt werden.

Todsünde Nummer eins ist die Verwendung der Platzierungsangabe `H`, also bspw.

```
1 \begin{figure}[H]
2 \end{figure}
```

Listing 4.2: Verbot von `H` als Platzierungsangabe

um zu erzwingen, dass ein Fließobjekt an dieser Stelle (engl. „here“) passiert, wenn `h` nicht genügt. Wenn `h` nicht genügt, dann liegt der Fehler bereits woanders und man sollte in das Log schauen, warum LaTeX die Umgebung nicht platzieren kann und das originäre Problem lösen. Alles andere macht es nur schlimmer.

Todsünde Nummer zwei ist die Verwendung von Leerzeilen innerhalb der Fließumgebung oder auch das Abrücken der Fließumgebung mit einer Leerzeile von dem Text der die Fließumgebung referenziert. Leerzeilen sind bei LaTeX Absätze und damit potentielle Stellen für Seitenumbrüche. Korrekt ist also folgendes:

```
1 Ein Text der auf die \cref{fig:my-fig} verweist
2 \begin{figure}[htbp]
3   \centering
4   \includegraphics{./images/my-image.png}
```

```

5  \caption{Eine tolle Abbildung}
6  \label{fig:my-fig}
7  \end{figure}
8  und ohne Leerzeile an der figure-Umgebung dransteht.
9
10 Dies ist nun ein neuer Absatz.

```

#### Listing 4.3: Verbot von Leerzeilen

Wie man erkennen kann, steht die `figure`-Umgebung sogar mitten im Satz zu der sie gehört. Der häufigste Grund, warum die Platzierungsoptionen `t`, `b`, `h` und `p` nicht so verhalten, wie man erwartet, ist, dass die `figure`-Umgebung aus Gründen der Übersicht mit Leerzeilen abgesetzt wird, sodass diese dann für LaTeX einen eigenen Block bildet.

Todsünde Nummer drei ist die Verwendung von `\begin{center}` und `\end{center}` statt von `\centering` innerhalb der Fließumgebung. Ersteres erzeugt wieder einen internen Absatz und damit einen eigenen Block. Dies ist also genauso schlimm wie Leerzeilen.

Todsünde Nummer vier ist die falsche Reihenfolge von `\caption{...}` und `\label{...}`. Die Reihenfolge ist *immer* das Objekt selbst (also `\includegraphics` oder `tabular`, usw.), dann folgt `\caption{...}` und zum Schluss `\label{...}`.

## 4.4 Tabellen

Typografisch gute Tabellen haben *niemals* vertikale Trennlinien sondern nur wenige horizontale Linien. Ferner haben sie eine trennende Linie ganz oben und ganz unten. Hierfür stellt das Paket `booktabs` die Befehle

- `\toprule`
- `\midrule`
- `\bottomrule`

zur Verfügung. Der Befehl `\hline` ist tabu. Für eine ausführliche Erläuterung auch über gute und schlechte Tabellen siehe die Dokumentation des `booktabs`-Pakets [Feao5].

Eine gute Tabelle hat also folgenden Code

```

1  \begin{table}
2    \centering
3    \begin{tabular}{l l l}                                \toprule
4      Datei          & Bedeutung      & Benutzer \\ \midrule
5      ./main.tex    & Hauptdatei     & nein      \\

```

```

6      ./figures/ & Zeichnungen & ja      \\
7      ./content/ & Kapitel      & ja      \\
8      ./logos/   & Logos        & nein   \\ \bottomrule
9      \end{tabular}
10     \caption{Dateien der Vorlage}
11     \label{tab:files-dirs-of-template}
12 \end{table}

```

Listing 4.4: Tabellen in LaTeX

und sieht dann so aus wie Tabelle 2.1.

Sollten eine Tabelle einmal so breit sein, dass sie nicht mehr horizontal auf eine Seite passt, so ist es natürlich möglich, diese mithilfe des Pakets `rotfloat` [Somo4] statt in eine `table`- in eine `sidewaystable`-Umgebung zu setzen. Also

```

1 \begin{sidewaystable}
2   \centering
3   \begin{tabular}{...}
4     ...
5   \end{tabular}
6   \caption{Bezeichnung}
7   \label{Referenzmarke}
8 \end{sidewaystable}

```

Listing 4.5: Gedrehte Tabelle

ein Ergebnis sieht man in Tabelle 4.1.

## 4.5 Bilder, Grafiken und Diagramme

Bei Grafiken sind zwei Fälle zu unterscheiden:

- Bilder in einem Binärformat (PNG, TIFF, JPG, PDF, etc.)
- Grafiken, die im TikZ-Quellcode vorliegen

Grundlegender Unterschied ist, dass Binärformatgrafiken mit `\includegraphics` eingebunden werden während TikZ-Grafiken mit `\input` eingebunden und von `pdflatex` kompiliert werden.

Für Binärformatbilder ist ein vollständiges Beispiel also

```

1 \begin{figure}[htbp]
2   \centering

```

	Level			
	Nominal	Qualitative	Quantitative	Absolute
Empirical relation	~ Equivalence	Ordinal ~ Equivalence < Ordering	Interval ~ Equivalence < Ordering	Ratio ~ Equivalence < Ordering
Empirical operation			⊕ Addition	⊕ Addition ⊗ Multiplication
Feasible transformation	$m' = f(m)$ for $f$ bij.	$m' = f(m)$ for $f$ mon.	$m' = am + b$ for $a > 0$	$m' = am$ for $a > 0$
Examples of features	<ul style="list-style-type: none"><li>• Telephone numbers</li><li>• Postal codes</li><li>• Gender</li></ul>	<ul style="list-style-type: none"><li>• Grades</li><li>• Degree of hardness</li><li>• Wind intensity</li></ul>	<ul style="list-style-type: none"><li>• Temperatur in F°</li><li>• Calendric time</li><li>• Geographic altitude</li></ul>	<ul style="list-style-type: none"><li>• Temperatur in K</li><li>• Mass</li><li>• Length</li><li>• Electric current</li></ul>
Range of features	<ul style="list-style-type: none"><li>• Numbers</li><li>• Names</li><li>• Symbols</li></ul>	Natural numbers	Real numbers	Natural numbers
Expressiveness	low	...	...	high

Tabelle 4.1: Beispiel für eine breite, gedrehte Tabelle (hier: Taxonomie der Maßskalen)

```

3 \includegraphics{./images/my-image.png}
4 \caption{Eine tolle Abbildung}
5 \label{fig:my-image}
6 \end{figure}

```

Listing 4.6: Binärgrafik in LaTeX

Der Code für eine TikZ-Grafik ist hingegen

```

1 \begin{figure}[htbp]
2   \centering
3   \tikzsetnextfilename{my-tikz-figure}
4   \input{./figures-src/my-tikz-figure.tex}
5   \caption{Eine tolle Zeichnung}
6   \label{fig:my-figure}
7 \end{figure}

```

Listing 4.7: TikZ-Zeichnung in LaTeX

Das Kommando `\tikzsetnextfilename{...}` ist nicht unbedingt notwendig aber sehr zu empfehlen, da dies als Name für das temporäre Kompilat im Ordner `./figures-compiled/` genommen wird. Dieser sollte gleich dem Namen der Quelldatei (ohne Endung) gewählt werden. Ansonsten nimmt `pdflatex` eine hochlaufende Nummer als Dateiname, was die Fehlersuche sehr erschwert.

Bilder können auch in Unterabbildungen gesetzt und sowohl als ganzes (vgl. Abbildung 4.1) als auch einzeln (vgl. Abbildungen 4.1a to 4.1d) referenziert werden. Der Beispielcode mit dem Paket `Subfig` [CK05] hierzu ist

```

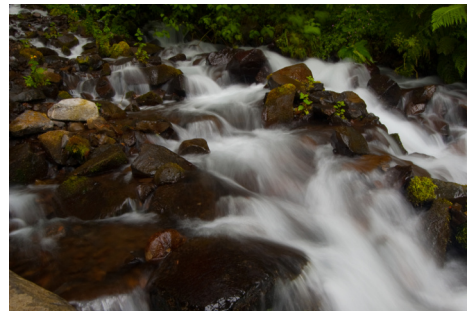
1 \begin{figure}
2   \centering
3   \subfloat[Unterbezeichnung 1]{{
4     \label{fig:my-sub-figure-1}
5     \includegraphics{./Pfad/zu/Bild/1}
6   }} \quad
7   \subfloat[Unterbezeichnung 2]{{
8     \label{fig:my-sub-figure-2}
9     \includegraphics{./Pfad/zu/Bild/2}
10  }}
11  \caption{Bezeichnung}
12  \label{fig:my-whole-figure}
13 \end{figure}

```

Listing 4.8: Unterabbildungen in LaTeX



(a) La Savoureuse, Lepuix, Frankreich (© Thomas Bresson)



(b) Bangkok, Thailand (© Prachanart Viriyaraks)



(c) Wahkeena Falls, Lincoln Park, USA (© srsly-guys)



(d) Nacionalni park Plitvička jezer, Kroatien (© Roman Bonnefoy)

Abbildung 4.1: Wasserfälle der Welt als Beispiel für Unterabbildungen

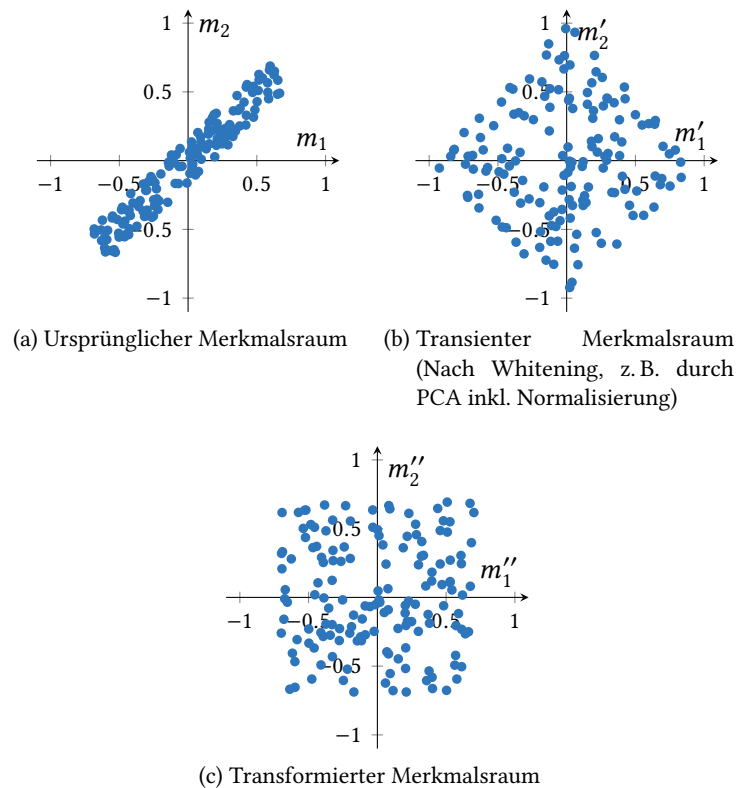


Abbildung 4.2: Diagramme mit TikZ direkt in LaTeX (hier: Die Schritte der „Independent component analysis“)

TikZ eignet sich hervorragend, um wissenschaftliche Zeichnungen, Vektorgrafiken und Diagramme direkt mithilfe von LaTeX zu setzen, sodass die Schrift direkt zum restlichen Dokument passt. Zu TikZ und dem darauf aufsetzenden PGFplots gibt es hervorragende Dokumentation [Tan13; Feu14].

Möglich sind zum Beispiel einfache Punktdiagramme wie in Abbildung 4.2 oder aufwendigere Diagramme mit mehreren Achsensystemen wie in Abbildung 4.3. Mit TikZ ist es auch möglich, kommutative Diagramme zu erstellen (vgl. Abbildung 4.4) oder Netzwerkkommunikationsgraphen (vgl. Abbildung 4.5). Natürlich darf auch eine Übersicht über die KIT-Corporate-Identity-Farben nicht fehlen (s. Abbildung 4.6).



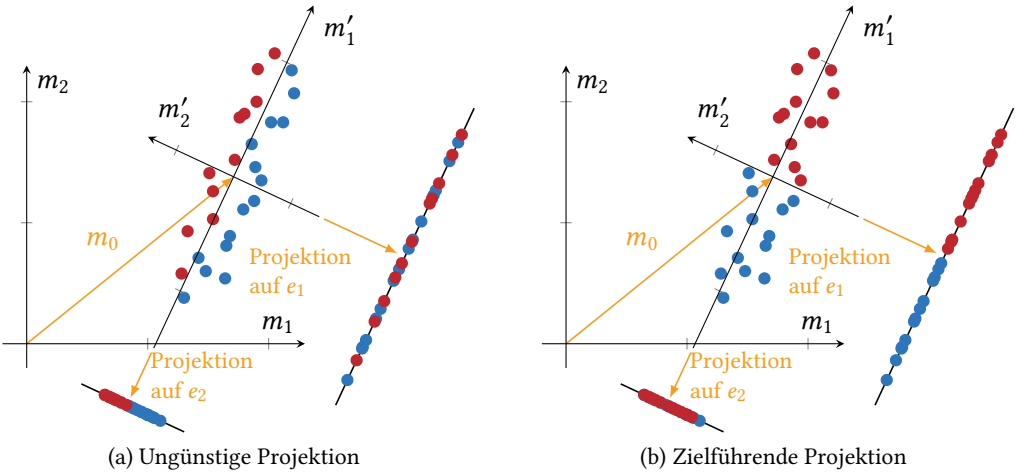


Abbildung 4.3: Aufwändiges Diagramm mit TikZ (hier: Probleme der „Principal component analysis“)

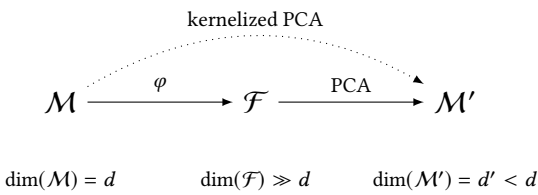


Abbildung 4.4: Kommutative Diagramm mit TikZ

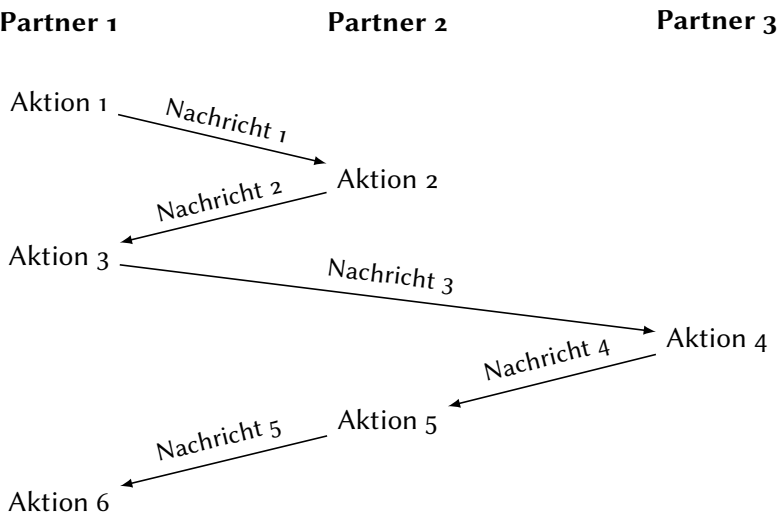


Abbildung 4.5: Netzwerkkommunikationsgraph mit TikZ

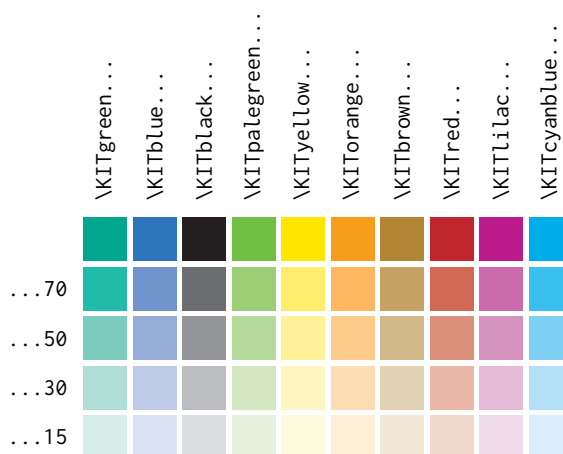


Abbildung 4.6: KIT-Corporate-Identity-Farben

## 4.6 Theoreme und Listings

Für eine mathematische Ausarbeitung gibt es LaTeX-Umgebungen, um Sätze (Theoreme), Lemma, Beispiele etc. im üblichen Stil von Mathematik-Büchern zu setzen und zu referenzieren. Vordefiniert sind die Umgebungen

- theorem für Sätze
- definition für Definitionen
- lemma für Lemma
- corollary für Korollare
- proposition für Propositionen

Die übliche Verwendung ist

```

1 \begin{theorem}[Optionaler Name]\label{thm:my-thoerem}
2 ...
3 \end{theorem}
```

Listing 4.9: Beispiel für Theorem-Umgebungen

Weitere Informationen findet man in der Dokumentation zum ntheorem-Paket [MS11]. Das Ganze sieht dann beispielsweise wie folgt aus.

**Satz 4.1 (Theorem von Arthur Dent)** *Die Antwort auf die Frage nach dem Leben, dem Universum und den ganzen Rest ist 42.*

**Proposition 4.2 (Zweifelhafte Folgerung)** *LaTeX ist schön. Beweis folgt unmittelbar aus Satz 4.1.*

Zum Einbinden und formatieren von Code-Beispielen – sog. Listings – wird das Paket listings [HMH14] unterstützt. Das Hervorheben von Schlüsselwörtern wird von LaTeX automatisch erledigt, wenn die korrekte Sprache des Listings angegeben ist. Vordefiniert sind die Umgebungen java für Java und latex für LaTeX. Ein Beispiel

```

1 \begin{java}[caption={A Java Hello-World example},label={lst:hello-
   world}]
2 public class HelloWorld {
3     public static void main( String[] args ) {
4         System.out.println( "HelloWorld" );
5     }
6 }
7 \end{java}

```

Listing 4.10: Listing-Beispiel

Das Ergebnis ist

```

1 public class HelloWorld {
2     public static void main( String[] args ) {
3         System.out.println( "HelloWorld" );
4     }
5 }

```

Listing 4.11: A Java Hello-World example

Man beachte, dass anders als bei anderen Umgebungen die Bezeichnung (caption) und die Referenzmarke (label) nicht als gesonderte Befehle sondern als optionale Argumente übergeben werden. Dies liegt daran, dass ein Listing in der Regel keine Fließumgebung ist, sondern an der Stelle im Text erscheint, an der sie im Code auch steht. Ferner folgt ein Listing den ganz normalen Seitenumbruchsregeln. Das heißt, überlanger Code wird einfach umgebrochen. Um ein Listing zu einem Fließobjekt zu machen, muss das optionale Argument float=<tbp> angegeben werden. Die Platzierungsangabe h für „hier“ ist nicht erlaubt. Denn dies ist das Standardverhalten ohne float.

## 4.7 Literaturangaben

Die Erstellung eines Literaturverzeichnisses geschieht mithilfe von BibLaTeX. Hier soll an dieser Stelle nicht viel Erklärung dazugegeben werden, denn die offizielle Dokumentation ist

eine der besten Dokumentation, die ich überhaupt kenne [Leh+13]. Insbesondere seien die Abschnitte „3.7 Citation Commands“ und das gesamte Kapitel „2 Database Guide“ ans Herz gelegt. Ersteres erklärt, wie im Text korrekt auf eine Zitatstelle verwiesen wird, letzteres erklärt wie die BIB-Datei auszusehen hat.

Wer nicht die Originaldokumentation liest sondern nach irgendwelche (stets total falschen) Lösungen im Internet sucht, weil er zu faul ist einfach nur mal lesen, der hat mit Fug und Recht jede nur denkbare schlechte Note auf seine Arbeit verdient.

## 4.8 Stichwortverzeichnis (Index) und Glossar

Die Vorlage unterstützt auch ein Stichwortverzeichnis und ein Glossar. Ein Stichwortverzeichnis (oder Index) ist einfach nur eine alphabetisch sortierte Liste von Begriffen mit einer Auflistung der Fundstellen im Dokument. Ein Glossar ist eine alphabetisch sortierte Liste von Begriffen mit Erklärung.

Der Index wird erzeugt, indem im Quellcode der Befehl `\index{Begriff}` eingefügt wird. Wichtig, der Begriff selbst wird dadurch nicht gedruckt, sondern muss noch einmal wiederholt werden, um auch gedruckt zu werden. Dieses Verhalten ist beabsichtigt, sodass im Index immer nur die Grundform des Wortes verwendet wird, aber im Text natürlich die richtige Deklination. Also:

```
1 Die meisten Funktionen dieser \index{Vorlage}Vorlage, werden durch
2 \index{Paket}Standardpaket bereitgestellt.
```

Listing 4.12: Beispiel für Index

Obiges Beispiel erzeugt einen Indexeintrag für „Vorlage“ der auf „Vorlage“ verweist und einen Eintrag „Paket“ der auf „Standardpaket“ verweist.

Um ein Glossar zu erzeugen, müssen die Glossarbegriffe zunächst definiert werden. Dies geschieht in der Datei `./content/00-glossary-definitions.tex`. Es gibt zwei Haupttypen von Glossarbegriffen: Abkürzungen (Akronyme) und allg. Einträge (z. B. Fachtermini).

Abkürzungen werden mit

```
1 \newacronym[%
2   shortplural={AUen},%
3   longplural={Abgasuntersuchungen}%
4 ] {au}{AU}{Abgasuntersuchung}
```

Listing 4.13: Definition von Abkürzungen

Die drei Hauptargumente sind in dieser Reihenfolge Marke, Abkürzung und Ausschreibung. Optionale Argumente sind die kurze und lange Pluralform.

Allgemeine Glossarbegriffe werden mit

```

1 \longnewglossaryentry{pkg}{%
2   name={Paket},%
3   plural={Pakete}}%
4 {%
5   Hier folgt eine lange Definition, die auch mehr als einen
6   Absatz beinhalten darf.
7 }
```

Listing 4.14: Definition von allg. Glossareinträgen

erzeugt.

Im Text werden die Einträge durch den Befehl `\gls{Marke}` verwendet. Der wesentliche Unterschied zwischen einer Abkürzung und einem allg. Glossareintrag ist, dass bei Abkürzungen bei erstmaliger Verwendung die Abkürzung gedruckt und die Langform in Klammer dahinter gesetzt wird. Bei allg. Glossareinträgen wird einfach nur der Name gesetzt. Statt `\gls{Marke}` gibt es noch viele weitere Befehle, um im Kontext des umgebenen Textes die korrekte Pluralform, Großschreibung am Satzanfang, etc. zu gewährleisten. Hierfür konsultiere man das Handbuch zum Paket `glossaries` (Pluralform, sic!) [Tal14].

## 4.9 Mathematik

Grundsätzlich gilt, was in [Ame99b; Ame99a] steht. In der Datei `./preamble/05-math.tex` sind eine Menge Kurzkommandos definiert, um eine einheitliche Typografie von Skalaren, Vektoren, Matrizen, Zufallsvariablen etc. zur vereinfachen. In diese Dateien einfach mal reinschauen, welche Kurzkommandos es gibt.

Auf zwei besondere Kommandos wird näher eingegangen, weil dies häufig falsch gemacht wird.

- Für die Matrixtransponierte gibt es das Kommando `\Tr`, also `\mA^{\Tr}` liefert  $A^T$
- Bei Integralen muss das „Differential-d“ gemäß ISO in aufrechter Schrift als Operator gesetzt sein mit einem kleinen Abstand zum Integranden. Hierfür gibt es das spezielle Kommando `\diff`. Also

$$\int_0^1 x^2 dx = \frac{1}{3} \quad (\text{falsche Typografie!}) \quad (4.1)$$

ist falsch, während `\int^1_0 x^2 \diff x = \frac{1}{3}` das Richtige liefert

$$\int_0^1 x^2 dx = \frac{1}{3} \quad (\text{richtige Typografie!}) \quad (4.2)$$

## 4.10 Sonstiges (URLs, Anführungszeichen, Sprache und Randnotizen)

Hier nur eine lockere Sammlung von kleineren Hinweisen, für die ein eigener Abschnitt zuviel gewesen wäre.

### 4.10.1 URLs

Internetadressen werden in das Kommando `\url{...}` eingefasst

### 4.10.2 Anführungszeichen

Um irgendwas in Anführungszeichen einzufassen, wird das Kommando `\enquote{...}` verwendet. Dies hat den Vorteil, dass man sich nicht um die korrekte typografische Variation der Anführungszeichen in Abhängigkeit der Sprache kümmern muss und auch verschachtelte Anführungszeichen korrekt behandelt werden. Also aus

```
1 \enquote{Beim Erreichen der Küste sprach Hamlet: \enquote{Es ist
    etwas faul im Staate Dänemark}}.
```

Listing 4.15: Behandlung von Anführungszeichen

wird „Beim Erreichen der Küste sprach Hamlet: ‚Es ist etwas faul im Staate Dänemark‘“ mit korrekt verschachtelten einfachen Anführungszeichen.

### 4.10.3 Sprachumschaltung (Deutsch, Englisch, etc.)

Um für einen Teil des Textes die Sprache zu wechseln, damit die Silbentrennung und die Auswahl der Anführungszeichen korrekt funktioniert, gibt es zwei Kommandos. Für einen kürzeren Text gibt es `\foreignlanguage[Sprache]{...}`. Dann wird für den Text in den geschweiften Klammern die angegebene Sprache verwendet. Um die Sprache bis zum nächsten Aufruf des gleichen Kommandos dauerhaft umstellen, gibt es `\selectlanguage{Sprache}`. Gegenwärtig unterstützte Sprachen sind `ngerman` für Deutsch nach neuer Rechtschreibung und `american` für Englisch nach amerikanischer Rechtschreibung.

#### 4.10.4 Randnotizen

Randnotizen werden mit dem Kommando `\marginnote{...}` gesetzt. Diese eignet sich zum Beispiel um im Text Stellen zu kennzeichnen, an denen man noch arbeiten sollte.

Ich bin eine  
überflüssige  
Randnotiz





## Literatur

- [Ame99a] American Mathematical Society. *Sample Paper for the amsmath Package*. 1999. URL: <http://mirrors.ctan.org/macros/latex/required/amslatex/math/testmath.pdf>.
- [Ame99b] American Mathematical Society. *User's Guide for the amsmath Package*. 1999. URL: <http://mirrors.ctan.org/macros/latex/required/amslatex/math/amsldoc.pdf>.
- [CK05] Steven Cochran und Vafa Karen-Pahlav. *The Subfig Package*. 2005. URL: <http://mirrors.ctan.org/macros/latex/contrib/subfig/subfig.pdf>.
- [Cub13] Toby Cubitt. *The cleveref package*. 2013. URL: <http://mirrors.ctan.org/macros/latex/contrib/cleveref/cleveref.pdf>.
- [Fea05] Simon Fear. *Publication quality tables in LaTeX*. 2005. URL: <http://mirrors.ctan.org/macros/latex/contrib/booktabs/booktabs.pdf>.
- [Feu14] Christian Feuersänger. *Manual for Package pgfplots*. 2014. URL: <http://mirrors.ctan.org/graphics/pgf/contrib/pgfplots/doc/pgfplots.pdf>.
- [HMH14] Carsten Heinz, Brooks Moses und Jobst Hoffmann. *The Listings Package*. 2014. URL: <http://mirrors.ctan.org/macros/latex/contrib/listings/listings.pdf>.
- [Leh+13] Philipp Lehman u. a. *The biblatex Package*. 2013. URL: <http://mirrors.ctan.org/macros/latex/contrib/biblatex/doc/biblatex.pdf>.
- [MS11] Wolfgang May und Andreas Schedler. *An Extension of the LaTeX-Theorem Environment*. 2011. URL: <http://mirrors.ctan.org/macros/latex/contrib/ntheorem/ntheorem.pdf>.
- [Somo4] Axel Sommerfeldt. *The rotfloat package*. 2004. URL: <http://mirrors.ctan.org/macros/latex/contrib/rotfloat/rotfloat.pdf>.
- [Tal14] Nicola Talbot. *User Manual for glossaries.sty v4.09*. 2014. URL: <http://mirrors.ctan.org/macros/latex/contrib/glossaries/glossaries-user.pdf>.

- [Tan13] Till Tantau. *TikZ & PGF*. 2013. URL: <http://mirrors.ctan.org/graphics/pgf/base/doc/pgfmanual.pdf>.

## Tabellenverzeichnis

2.1	Dateien und Verzeichnisse der Vorlage . . . . .	3
4.1	Beispiel für eine breite, gedrehte Tabelle (hier: Taxonomie der Maßskalen) . .	17



## Abbildungsverzeichnis

3.1	Neues Ausgabeprofil in TeXnicCenter (1 von 2) . . . . .	7
3.2	Neues Ausgabeprofil in TeXnicCenter (2 von 2) . . . . .	8
4.1	Wasserfälle der Welt als Beispiel für Unterabbildungen . . . . .	19
4.2	Diagramme mit TikZ direkt in LaTeX (hier: Die Schritte der „Independent component analysis“) . . . . .	20
4.3	Aufwändiges Diagramm mit TikZ (hier: Probleme der „Principal component analysis“) . . . . .	21
4.4	Kommutative Diagramm mit TikZ . . . . .	21
4.5	Netzwerkkommunikationsgraph mit TikZ . . . . .	21
4.6	KIT-Corporate-Identity-Farben . . . . .	22



## Theoremverzeichnis

4.1	Theorem von Arthur Dent . . . . .	22
4.2	Zweifelhafte Folgerung . . . . .	23





## Listings

4.1	Cleveres Referenzieren . . . . .	14
4.2	Verbot von H als Platzierungsangabe . . . . .	14
4.3	Verbot von Leerzeilen . . . . .	14
4.4	Tabellen in LaTeX . . . . .	15
4.5	Gedrehte Tabelle . . . . .	16
4.6	Binärgrafik in LaTeX . . . . .	16
4.7	TikZ-Zeichnung in LaTeX . . . . .	18
4.8	Unterabbildungen in LaTeX . . . . .	18
4.9	Beispiel für Theorem-Umgebungen . . . . .	22
4.10	Listing-Beispiel . . . . .	23
4.11	A Java Hello-World example . . . . .	23
4.12	Beispiel für Index . . . . .	24
4.13	Definition von Abkürzungen . . . . .	24
4.14	Definition von allg. Glossareinträgen . . . . .	25
4.15	Behandlung von Anführungszeichen . . . . .	26



# Index

Abbildung, *siehe* Bild  
Achsensystem, *siehe* Diagramm  
Akzent, 5

Beispiel, *siehe* Theorem

Bild, 3

    Binär-, 3, 16

    Unterabbildung, 18

    Vektor-, 1, 3, 16

Code, *siehe* Listing

Compilerargument, 6

Diagramm, 20

    Punkt-, 20

Dokumentation, 13

Element

    blockbildend, 14

Entwicklungsumgebung

    integrierte, 6

Erklärung, 24

Fehler, 5

Fundstelle, 24

Grafik, *siehe* Bild

Integral, 25

Internetadresse, *siehe* URL

Kapitel, 3

Lemma, *siehe* Theorem

Linux, 6

Listing, 23

Lösung, 13

Matrix, 25

Platzierung, 14, 23

Problem, 13

pstricks, 2

Rechtschreibung

    amerikanisch, 26

    neue deutsche, 26

Satz, *siehe* Theorem

Schlüsselwort, 23

Silbentrennung, 26

Skalar, 25

Sortierung, 13

Sprache, 26

Theorem, 22

Vektor, 25

Warnung, 13

    Please re-run latex, 5

Windows, 6

Zeichenkodierung, 5

Zeichnung, *siehe* Bild

Zitat, 24

Zufallsvariable, 25





## Glossar

*BibLaTeX* Der Nachfolger von BibTeX zum Erzeugen von Literaturverzeichnissen in LaTeX.

Es zeichnet sich vor allem durch deutlich bessere Flexibilität bei der Gestaltung des Literaturverzeichnisses und der Art und Weise wie Zitatmarken gesetzt werden aus.

Darüber hinaus ist es vollständig UTF-8-kompatibel. 1, 6, 23

*BibLatex* Der Vorgänger von BibLaTeX. 1, 6

*EPS* Embedded Postscript. 2,

*Java* Eine von Sun Microsystems 1995 veröffentlichte, objektorientierte Programmiersprache.

23

*LaTeX* Eine von Leslie Lamport 1980 entwickelter Satz von Makros zur Erweiterung von TeX.

23

*Paket* Ein LaTeX-Paket besteht aus einer oder mehrerer Dateien, die entweder vorhandene Kernfunktionen von LaTeX umdefinieren und so das Verhalten derselbigen bzw. das Erscheinungsbild des fertigen Dokuments verändern oder die zusätzliche Befehle zur Verfügung stellen. 1

*PCA* Principal Component Analysis. 20,

*PDF* Portable Document Format. 2,

*PGFplots* Eine Sammlung von TikZ-Paketen, die ein direktes Erzeugen von Diagrammen aller Art (inkl. 3D-Diagramme) direkt aus LaTeX heraus ermöglicht. 20

*PostScript* Eine von Adobe 1984 entwickelte Seitenbeschreibungssprache. 1, 2

*TikZ* Eine Sammlung von LaTeX-Paketen, die ein direktes Erzeugen von (technischen) Zeichnungen, Diagrammen, etc. in LaTeX erlaubt. 1–3, 9, 16, 18, 20

*Umgebung* Ein Bereich im LaTeX-Code der mit `begin` eingeleitet und mit `end` beendet wird.

Umgebungen können auch verschachtelt sein. 22

*UTF-8* Ein Schema zur Kodierung von Zeichen in computerverarbeitbarer Form, die Zeichen aus allen Sprachen umfasst. 1, 5

*WYSIWYG* What you see is what you get. 9,