

L^AT_EX-Kurs

Teil 01

Stefan Held

30. Juli 2013

Inhaltsverzeichnis

1	Einleitende Worte	2
2	Installation	2
2.1	Ziele	2
2.2	Benötigte Software	2
2.3	Reihenfolge	3
2.4	Test	3
3	Header	4
4	Dokumentenstruktur	7
5	Titelseite	9
6	Hauptdokument	12
6.1	Formatierung	12
6.1.1	Silbentrennung	12
6.1.2	Absätze und Abstände	13
6.1.3	Abschnitte und Kapitel	13
6.1.4	Textattribute	14
6.1.5	Textausrichtungen	16
6.2	Befehle	18
6.2.1	Mathematik	18
6.2.2	Tabellen	21
6.2.3	Grafiken	23
6.2.4	Fußnoten	27
6.2.5	Zitate	28
6.2.6	Eigene Befehle definieren	29
6.2.7	Plotten mit L ^A T _E X	30

1 Einleitende Worte

Bei diesem Dokument handelt es sich um ein begleitendes Nachschlagewerk zu einem L^AT_EX-Kurs. Es wird daher laufend erweitert und verbessert. Über Verbesserungsvorschläge bezüglich weiterer Kapitel, Reihenfolge, Verständnis, sowie natürlich Rechtschreibfehler, bin ich daher sehr dankbar. Wegen der eben genannten eventuellen Unvollständigkeit bitte ich darum, dieses Dokument nicht weiterzugeben. Ich wünsche euch viel Spaß beim L^AT_EX lernen und hoffe alles verständlich vermitteln zu können.

2 Installation

2.1 Ziele

Das Installations-How-To gliedert sich in folgende Abschnitte:

- **WAS** benötigt man?
- **WELCHE** Reihenfolge ist bei der Installation von entscheidender Bedeutung?
- **WIE** testet man, ob alles funktioniert?

Nach Durcharbeiten dieses How-To's sollte es möglich sein ...

- ✓ ... den Texmaker zu nutzen,
- ✓ ... L^AT_EX-Dateien zu kompilieren,
- ✓ ... in diesen L^AT_EX-Kurs einzusteigen.

2.2 Benötigte Software

Benötigt wird:

- MiKTeX (Hier downloaden)



- Ghostscript und Ghostview (Hier downloaden), genauer gesagt: GSView und Ghostscript



- Texmaker (Hier downloaden)



Für alle Mac-User: statt MiKTeX, Ghostscript und Ghostview muss man MacTeX (Hier downloaden) installieren. Den Texmaker allerdings dann wieder ganz normal installieren.

2.3 Reihenfolge

Wichtig ist es nun zuerst MiKTeX, dann Ghostscript und Ghostview und zum Schluss den Texmaker zu installieren. Nur so werden die Verlinkungen zu den zuvor installierten Programmen automatisch richtig gesetzt.

2.4 Test

Um nun zu testen, ob die Installation richtig geklappt hat, hier ein kurzes Musterdokument:

```
\documentclass[12pt,a4paper]{scrartcl}
\usepackage{ngerman}
\usepackage{blindtext}
\begin{document}
\blindtext
\end{document}
```

Listing 1: ein paar Zeilen code

Einfach per Copy & Paste in ein neues TeX-Dokument einfügen, dann die Datei in einem Ordner speichern und abschließend F1 drücken. Es kann nun sein, dass der Texmaker zusätzliche Packages (*.sty - Dateien) benötigt - diese einfach installieren. Nun sollte auf der rechten Seite das fertige Dokument erscheinen. Evtl. muss man unter „Ansicht“ und unter den „Optionen“ erst das integrierte PDF-Anzeige Fenster aktivieren (unter dem Menüpunkt „embed“, bzw. „eingebettet“).

3 Header

Zunächst befassen wir uns mit dem Header. Hier werden alle relevanten Voreinstellungen getroffen, also alles das was ihr aus klassischen Textverarbeitungsprogrammen unter „Seite einrichten“ kennt, und vieles darüber hinaus. Kommen wir also zum ersten Kommando:

```
\documentclass[12pt,a4paper]{scrartcl}
```

Listing 2: documentclass

Hier definieren wir die globalen Einstellungen unseres Dokuments. 12pt bestimmt die Schriftgröße, a4paper legt die Seitengröße fest und scrartcl den Dokumententyp. Für letzteres hat sich wie bereits genannt die Variante scrartcl bewährt, weil es sehr vielseitig ist und die klassische Kapitelnummerierung bereitstellt.

Als nächstes folgt schon das erste usepackage. Usepackages stellen in L^AT_EX quasi vorgefertigte Bibliotheken dar, mit deren Hilfe man ganz bestimmte Einstellungen leichter erledigen kann, bestimmt Formatierungen erreicht, etc. . . .

```
\usepackage[] {geometry}  
\geometry{a4paper, hmargin={2cm,2cm},vmargin={2.5cm,2.5cm}}
```

Listing 3: geometry

Das usepackage geometry ermöglicht uns die Seitenränder einzustellen. Man spricht von margin - h steht dabei immer für horizontal, v für vertical. Die erste Zahl ist der linke Abstand, die zweite der rechte.

Kommentare leitet man mit einem % Zeichen ein. Sie erleichtern die Übersichtlichkeit bei komplexeren Dokumenten ungemein. Es empfiehlt sich Trennzeilen oder Abschnittsbezeichnungen als Kommentar einzufügen.

```
%-----  
%      Ab      hier      Formel      zum      Thema      ...
```

Listing 4: Kommentare

Um nun auch Formeln schreiben zu können benötigen wir diverse usepackages:

```
\usepackage{amsmath}  
\usepackage{amsfonts}  
\usepackage{amssymb}  
\usepackage{cancel}
```

Listing 5: Mathematikbefehle

Die erste drei gewährleisten eine korrekte Darstellung der Formeln und Symbole in einer zum übrigen Dokument passenden Schriftart. Das vierte ermöglicht es uns einzelne Formelteile zu streichen, was z.B. Kürzen symbolisiert. Jedes Package bringt dabei eigene Befehle mit sich, wie wir später noch sehen werden.

Um Seiten im Dokument auch im Querformat verwenden zu können benötigen wir:

```
\usepackage{landscape}
```

Listing 6: Querformat

Dabei kann man ohne Probleme einzelne Textabschnitte im Querformat definieren, welche von den übrigen Seiten abgegrenzt sind. So hat man bei nachträglichen Änderungen nicht das Problem, dass, wie bei klassischer Textverarbeitung, etwas verrutscht und plötzlich das Falsche quer ist.

Natürlich ist es auch möglich Bilder einzufügen, einzelne Textabschnitte, Wörter, oder nur einzelne Buchstaben farbig darzustellen.

```
\usepackage{graphicx}  
\usepackage{color}  
\usepackage[usenames,dvipsnames]{xcolor}
```

Listing 7: Farben und Bilder

Graphicx ist für die Bildereinbindung zuständig. Dabei ist es möglich mit dem Einfügen-Befehl nicht nur Höhe und Breite festzulegen - darüber hinaus kann man die Grafik zuschneiden, drehen, etc. Die beiden anderen stellen die Textfarben zur Verfügung. Da der Umfang der möglichen Farben mit dem color-package recht gering ist empfiehlt es sich auch noch xcolor einzubinden, welches deutlich mehr Farbvarianten zulässt. Grundsätzlich ist es auch möglich selbst definierte RGB-Farben zu verwenden.

Oft macht es Sinn, Links innerhalb eines Dokuments zu setzen. Da die Ausgabe des fertig kompilierten Dokuments grundsätzlich als PDF erfolgt, macht es diese sehr übersichtlich und einfach in der Navigation. Wenn man ein Inhaltsverzeichnis eingefügt sind zudem alle Überschrif-

ten automatisch mit den zugehörigen Textpassagen verknüpft.

```
\usepackage{hyperref}
```

Listing 8: hyperref

Entgegen den gewohnt blauen Hyperlinks stellt L^AT_EX die Links mit einem roten Rahmen dar.

Für's erste genügen diese usepackages. Man kann jedoch jederzeit weitere hinzufügen.

4 Dokumentenstruktur

Nachfolgend ist exemplarisch die Dokumentenstruktur dargestellt. Allerdings sind der Übersichtlichkeit halber nicht alle oben genannten usepackages enthalten¹.

```
%%%%%%%%%%
%DOCUMENTCLASS%
%%%%%%%%%%

%Gibt_vor,_um_welchen_Typ_Dokument_es_sich_handelt
\documentclass[12pt,a4paper]{scrartcl}

%%%%%%%%%%
%USEPACKAGES%
%%%%%%%%%%

%hiermit_lassen_sich_die_Seitenraender_manuell_einstellen
%h_wie_horizontal,_v_wie_vertical
\usepackage[]{geometry}
\geometry{a4paper,hmargin={2cm,2cm},vmargin={2.5cm,2.5cm}}
%deutsche Sprache und Zeichen (Umlaute etc.)
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[ngerman]{babel}

%%%%%%%%%%
%TITELSEITE%
%%%%%%%%%%

%fuer die Titelseite
\title{\LaTeX{}-Kurs\ Teil 01}
\author{Stefan Held}

%%%%%%%%%%
%DOKUMENT%
%%%%%%%%%%
\begin{document}
\maketitle %erzeugt Titelseite
\tableofcontents %erzeugt Inhaltsangabe
\newpage %neue Seite
\section{erstes Kapitel}
Hier koennte Ihre Werbung stehen
\end{document}
```

Listing 9: Dokumentenstruktur

¹Damit die Kopierbarkeit des Quellcodes (d.h. Copy & Paste) möglich ist, werden im folgenden einige Leerzeichen mit _ , sowie Umlaute ersetzt. Beim Kopieren einfach wieder manuell rückgängig machen.

Also der Reihe nach die notwendige Struktur:

1. Definition des Dokumententyps: vgl. Kapitel 3, S. 4
2. Einbinden der usepackages: vgl. Kapitel 3, S. 4
3. Einstellungen für die Titelseite. vgl. Kapitel 5, S. 9
4. Hauptdokument: vgl. Kapitel 6, S. 12, für spezielle Infos zu den einzelnen Befehlen vgl. Kapitel 6.2, S. 18

5 Titelseite

Die Einstellungen für die Titelseite finden sich nach den eingebundenen usepackages und enthalten je nach Bedarf folgende Befehle:

- für den Dokumententitel (mehrzeilige Titel → Zeilenumbruch mit \\)

```
\title{Testdokument}
```

- für das anzuzeigende Datum (wird dieser Befehl weggelassen, wird das Datum automatisch auf das Datum des letzten Kompilierens, d.h. Quellcode-Übersetzens, aktualisiert)

```
\date{01.07.2013}
```

- für den Namen des Autors:

```
\author{Max Mustermann}
```

Alles was davon nicht benötigt wird, einfach weglassen. Jeden Befehl sollte man hier in eine eigene Zeile schreiben. Kompiliert sieht das so aus:



Innerhalb des Dokuments erzeugt man nun die gewünschte Titelseite mit Hilfe des Befehls:

```
\maketitle
```

Das reicht für einfachere Dokumente meist aus. Allerdings ist für Semesterarbeiten meist eine komplizierter aufgebaute Titelseite notwendig. Da diese jedoch in der Regel als Muster verfügbar ist, gebe ich nachfolgend nur den Beispielcode an, ohne genauer ins Detail zu gehen. Am Ende dieses Kurses sollten sich jedoch die meisten der verwendeten Befehle klären.

Kompliziertere Titelseiten kann man also mit Hilfe selbst definierter Titelseiten aufbauen. Dazu ein Beispielcode:

```
\documentclass[12pt,a4paper]{scrartcl}
\usepackage[]{geometry}
\geometry{a4paper,hmargin={2cm,2cm},vmargin={2.5cm,2.5cm}}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[ngerman]{babel}
\usepackage{graphicx}
\usepackage{color}
\begin{document}
%%%%%%%%%%
%%Titelseite%%
%%%%%%%%%%
\begin{titlepage}
\thispagestyle{empty}
\begin{minipage}[c]{3cm}\hfill
\includegraphics[width=3cm]{baer.jpg}
\end{minipage}\hfill
\begin{minipage}[c]{9cm}
\begin{center}
\textsc{Universitaet\\Fakultaet\\Lehrstuhl}\\
\end{center}
\end{minipage}\hfill
\begin{minipage}[c]{3cm}
\includegraphics[width=3cm]{baer.jpg}
\end{minipage}
\begin{center}
\line(1,0){350}
\end{center}{\color{white}.}\\[1cm]
\begin{center}\textbf{\LARGE{Arbeit}}\end{center}
{\color{white}.}\\[8mm]
\begin{center}\textbf{Titel}\end{center}{\color{white}.}\\[8mm]
\begin{center}
\includegraphics[width=\textwidth]{baer.jpg}
\end{center}
\begin{flushleft}
\begin{tabular}{lll}
\textbf{Verfasser:}&&Max_Mustermann_Mat.-Nr.:1000017\\&&\\
\textbf{ausgegeben_am:}&&01.07.2013\\
\textbf{eingereicht_am:}&&01.07.2013\\
\textbf{Betreuer:}&&Max_Mustermann\\
\end{tabular}
\end{flushleft}
\end{titlepage}
%%Ende_Titelseite%%
\end{document}
```

Kompiliert sieht das wiederum so aus:



UNIVERSITÄT
FAKULTÄT
LEHRSTUHL



Arbeit

Titel



Verfasser: Max Mustermann <Mat.-Nr.: 1000017>

ausgegeben am: 01.07.2013

eingereicht am: 01.07.2013

Betreuer: Max Mustermann

6 Hauptdokument

Im Hauptdokument wird sämtlicher Text der zwischen `\begin{document}` und `\end{document}` steht eins zu eins in das Enddokument übernommen. Voreinstellung ist dabei der Blocksatz. Durch unsere eingebundenen usepackages zur deutschen Sprache weiß der Compiler, an welchen Stellen ein Wort getrennt werden kann. Wie man zusätzliche Trennstellen erzeugen kann, wird auf Seite 12, Abschnitt 6.1.1 näher erläutert.

6.1 Formatierung

Die grundlegende Formatierung übernimmt \LaTeX selbst, nach den gängigen Typographierichtlinien. Ziel ist dabei ein sauberes und ausgewogenes Textbild, welches zudem den Lesefluss unterstützen soll. So sind beispielsweise nie einzelne Worte eines Abschnitts auf einer separaten Seite. \LaTeX modifiziert dabei die Wortabstände im Blocksatz so, dass zumindest ein zusammengehöriger Satz(teil) zusammen umbricht. Auch sind keine doppelten Leerzeichen möglich; oft tippt man aus Versehen zwei Leerzeichen hintereinander, welche ungleiche Wortabstände produzieren. Um dies zu vermeiden ignoriert \LaTeX jedes zusätzliche Leerzeichen. Um nun trotzdem bei Bedarf Kontrolle über diese und ähnliche Punkte zu haben folgen hier einige Formatierungsvorschriften für den Compiler.

6.1.1 Silbentrennung

Ein Wort, welches **nur** an einer bestimmten Stelle umgebrochen werden soll/ darf definiert man so:

```
Problemorientierungs\interdisziplinaritaetsindikator
```

Listing 10: ausschließliche Trennstelle

Ist ein Wort nicht im vorinstallierten Wörterbuch enthalten, so kann es passieren, dass der Compiler nicht weiß, an welcher Stelle er trennen soll. Um dies zu beheben kann man mögliche Trennstellen einfügen:

```
Problem"-orientierungs"-inter"-disziplinaritaets"-indikator
```

Listing 11: Trennstellen vorschlagen

Will man nun schließlich verhindern, dass ein Wort getrennt wird kann man dies auf einfache Weise verhindern:

```
\mbox{Problemorientierungsinterdisziplinaritaetsindikator}
```

Listing 12: Trennen verhindern

Will man einen geschützten Bindestrich erzeugen, an dem nicht getrennt werden soll, verwendet man:

```
Max"~Mustermann"~Strasse
```

Listing 13: Trennen verhindern

6.1.2 Absätze und Abstände

Mit `\\` kann man einen Zeilenumbruch produzieren. Dabei kann man bei Bedarf zusätzlich den Abstand angeben, beispielsweise:

```
\\[1cm]
```

Listing 14: Zeilenumbruch

Dabei sind mögliche Abstandsangaben:

- $1 \text{ sp} = 65536 \text{ pt}$
- $12 \text{ pt} = 1 \text{ pc}$
- $1238 \text{ pt} = 1157 \text{ dd}$
- $12 \text{ dd} = 1 \text{ cc}$
- $72.27 \text{ pt} = 1 \text{ in} = 72 \text{ bp}$
- $1 \text{ in} = 25,4 \text{ mm} = 2,54 \text{ cm}$
- Einheit 1 ex : entspricht der Höhe des Buchstabens x
- 1 em : entspricht der Breite des Buchstaben M

Eine neue Seite beginnt man mit:

```
\newpage
```

Listing 15: neue Seite

6.1.3 Abschnitte und Kapitel

Um einen Text zu gliedern bedarf es Kapitel und Unterkapitel. Diese in \LaTeX zu erzeugen ist sehr praktisch. Größe, Nummerierung und Eintrag im Inhaltsverzeichnis werden dabei automatisch ergänzt. Es gibt folgende Gliederungsarten:

- `\chapter` (nur für Dokumentenklasse *book*)
- `\section`
- `\subsection`
- `\subsubsection`

Konkret sieht das so aus:

```
\section{Ueberschrift}  
\subsection{Unterueberschrift}  
\subsubsection{nochmal_eins_darunter}
```

Listing 16: Abschnitte

Zusätzlich ist es möglich ein „label“, d.h. einen Querverweis zu setzen dazu schreibt man direkt unter die Überschrift:

```
\section{Ueberschrift}  
\label{Verweis_auf_Ueberschrift}
```

Listing 17: Labels

, wobei der Text innerhalb der geschweiften Klammern des label-Befehls praktisch die ID des Verweises ist. Es gibt zwei Arten von Verweisen:

- **\pageref**: gibt die Seitenzahl der Überschrift aus **\ref**: gibt die Kapitelnummer der Überschrift aus

```
\section{Ueberschrift}  
\label{Verweis_auf_Ueberschrift}
```

Listing 18: Labels

Z.B. kann man auf Seite 9, Kapitel 5 verweisen, wo mehr über Titelseiten zu finden ist. Wichtig ist dabei, dass beide Befehle nur eine Nummer ausgeben. Die Wörter „Seite“, oder „Kapitel“ müssen im Text ergänzt werden. Schön ist diese Art der Querverweise vor allem in Kombination mit dem hyperref-usepackage. Die Verweise dienen dann zusätzlich als Sprungmarke innerhalb des PDF's.

Nun bleibt noch anzumerken, dass diese Art des Verweises nicht nur auf Überschriften beschränkt ist, auch Bilder, Tabellen, Fußnoten, etc. können in analoger Weise referenziert werden.

6.1.4 Textattribute

Fett gedruckten Text erzeugt man mit:

```
{\bf fettgedruckter_Text}
```

Listing 19: fettgedruckter Text

, wobei der fettgedruckte Text innerhalb der geschweiften Klammern stehen muss. Will man einen längeren Text fettgedruckt darstellen eignet sich die dafür passende Umgebung:

```
\textbf{Das_ist_ein_fettgedruckter_Text}
```

Listing 20: längerer fettgedruckter Text

In ähnlicher Weise kann ein Textabschnitt in einer anderen Farbe eingefärbt werden:

```
{\color{red} Das_ist_ein_roter_Text}
```

Listing 21: farbiger Text

Dabei sind folgende Farben möglich:

- black
- blue
- brown
- cyan
- darkgray
- gray
- green
- lightgray
- magenta
- orange
- purple
- red
- violet
- white
- yellow

Das sind nur die gängigsten Farben, mit den beiden Optionen `usenames` und `dvipsnames` beim Einbinden des `xcolor`-packages sind weit mehr Farben möglich.

Auch kann der Text in Kapitälchen geschrieben werden mittels

```
\textsc{Das_ist_ein_Text_in_Kapitaelchen}
```

Listing 22: Text in Kapitälchen

... oder kursiv mittels

```
\textit{Das_ist_ein_kursiver_Text}
```

Listing 23: kursiver Text

,wobei auch hier, ähnlich wie bei `\bf`, ein Kurzbehl `\it` existiert.

Zusätzlich gibt es die Möglichkeit einzelne Textabschnitte farbig zu hinterlegen, ähnlich einer Textmarker-Option:

```
\hl{Das_ist_ein_hinterlegter_Text}
```

Listing 24: Text mit Hintergrundfarbe

Dazu muss allerdings das passende `usepackage` eingebunden werden:

```
\usepackage{soul}
```

Listing 25: soul

Im Überblick sind das nun folgende Formatierungsmöglichkeiten:

Das ist ein fettgedruckter Text.
Das ist ein roter Text.
DAS IST EIN TEXT IN KAPITÄLCHEN.
Das ist ein kursiver Text.
Das ist ein hinterlegter Text.

6.1.5 Textausrichtungen

In \LaTeX kann man fast alles mittels des `center`-Befehls zentrieren. Dazu zählen nicht nur Text und Bilder, sondern auch Tabellen, oder ähnliches. Alles was zentriert werden soll muss dabei in der `center`-Umgebung stehen:

```
\begin{center}  
Das_ist_zentriert.  
\end{center}
```

Listing 26: zentrierter Text

Analog erhält man links ausgerichteten Text:

```
\begin{flushleft}  
Das_ist_links_ausgerichtet.  
\end{flushleft}
```

Listing 27: links ausgerichteter Text

... sowie rechts ausgerichteten Text:

```
\begin{flushright}  
Das_ist_rechts_ausgerichtet.  
\end{flushright}
```

Listing 28: rechts ausgerichteter Text

Mittels `minipages` ist es zudem möglich Text in Spalten nebeneinander zu schreiben. Dazu ein kurzes Codebeispiel in Kombination mit den genannten Ausrichtungsmöglichkeiten:


```

\begin{minipage}{0.33\textwidth}
\begin{flushleft}
TEST
\end{flushleft}
\end{minipage}
%%%%%%%%
\begin{minipage}{0.33\textwidth}
\begin{center}
TEST
\end{center}
\end{minipage}
%%%%%%%%
\begin{minipage}{0.33\textwidth}
\begin{flushright}
TEST
\end{flushright}
\end{minipage}

```

Listing 29: minipages

Wobei `\textwidth` der Parameter für die aktuelle beschreibbare Seitenbreite ist. Kompiliert sieht das so aus:

TEST	TEST	TEST
------	------	------

Zum Abschluss dieses Kapitels noch einige Worte bezüglich der minipage Umgebung:

Diese ist ein beliebtes Mittel zum „layouten“ per Hand. Es bewirkt eine „Seite in der Seite“ und damit beispielsweise eine A4 Seite mit mehreren Spalten, wie dies ja beispielsweise in Zeitungsartikeln üblich ist. Zu beachten ist hierbei, dass die minipages wie Zeichen im Text verankert sind, d.h. vor, bzw. nach einer minipage muss auch ein Zeilenumbruch folgen. Dieser erfolgt, im Gegensatz zu einer align Umgebung (vgl. Seite 18, Kapitel 6.2.1), nicht automatisch und führt so häufig zu Problemen. Will man beispielsweise 2 Spalten nebeneinander haben so sieht das im Quellcode so aus:

```

\begin{minipage}{0.48\textwidth}
\blindtext
\end{minipage}
%%%%%%%%
\hspace{0.04\textwidth}
%%%%%%%%
\begin{minipage}{0.48\textwidth}
\blindtext
\end{minipage}

```

Listing 30: minipages

... und kompiliert:

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Dies hier ist ein Blindtext zum Testen von Textausgaben. Wer diesen Text liest, ist selbst schuld. Der Text gibt lediglich den Grauwert der Schrift an. Ist das wirklich so? Ist es gleichgültig, ob ich schreibe: „Dies ist ein Blindtext“ oder „Huardest gefburn“? Kjift – mitnichten! Ein Blindtext bietet mir wichtige Informationen. An ihm messe ich die Lesbarkeit einer Schrift, ihre Anmutung, wie harmonisch die Figuren zueinander stehen und prüfe, wie breit oder schmal sie läuft. Ein Blindtext sollte möglichst viele verschiedene Buchstaben enthalten und in der Originalsprache gesetzt sein. Er muss keinen Sinn ergeben, sollte aber lesbar sein. Fremdsprachige Texte wie „Lorem ipsum“ dienen nicht dem eigentlichen Zweck, da sie eine falsche Anmutung vermitteln.

Der `\hspace` Befehl erzeugt hierbei den definierten horizontalen Abstand zwischen den Spalten. Natürlich dürfen minipages auch ineinander verschachtelt sein. Die Breitenangabe kann zudem auch in cm, o.Ä. erfolgen.

6.2 Befehle

6.2.1 Mathematik

Jeder Mathebefehl steht in einer Matheumgebung. Dabei muss man unterscheiden, ob die Formel im Fließtext auftauchen soll, oder separat stehen soll im Fließtext stehen Formeln zwischen Dollarzeichen:

```
Das_ist_ein_Beispieltext_in_dem
 $\alpha$ als_griechischer_Buchstabe_auftaucht.
```

Listing 31: Formel im Text

Die separate Umgebung produziert auch die schöneren Formeln und sieht so aus:

```
\begin{align*}
\alpha
\end{align*}
```

Listing 32: Formelumgebung

Der Stern hinter align verhindert dabei eine Durchnummerierung der Formeln. Ist dies jedoch gewünscht, die beiden Sterne einfach weglöschen.

Nun zu den grundlegenden Formelbegriffen anhand einiger Beispiele:

```
\int\limits_1^2 x^2 \quad dx
```

$$\int_1^2 x^2 dx$$

```
\frac{1}{2 \cdot \alpha_G \cdot \varphi_{GH}}
```

$$\frac{1}{2 \cdot \alpha_G \cdot \varphi_{GH}}$$

```
\sum\limits_{k=0}^n \sin(x^k)
```

$$\sum_{k=0}^n \sin(x^k)$$

```
\left(\frac{a}{b}\right)^{-1}
```

$$\left(\frac{a}{b}\right)^{-1}$$

```
\sqrt{a+b}=\left\lbrace
\begin{matrix}
a\\
b\\
c
\end{matrix}
\right.
```

$$\sqrt{a+b} = \begin{cases} a \\ b \\ c \end{cases}$$

```
\left\lbrace
\frac{\exp(x)}{\bar{a} \cdot \tilde{b} \cdot \hat{c}}
\right\rangle
```

$$\left\langle \frac{\exp(x)}{\bar{a} \cdot \tilde{b} \cdot \hat{c}} \right\rangle$$

```
x^{\prime}+y^{\star}=k\in\mathbb{N}
```

$$x' + y^{\star} = k \in \mathbb{N}$$

```
100^{\circ}\mathrm{C} \quad \forall \vartheta \leq \infty
```

$$100^{\circ}\mathrm{C} \quad \forall \vartheta \leq \infty$$

Dazu einige Anmerkungen:

- Es gibt die Möglichkeit mittels einfachem \ein Leerzeichen einzufügen.
- Skalierbare Klammern (mit left und right) passen sich der Höhe des Inhalts an. Es gibt dabei folgende Klammerarten:

$$\left(\left(\left[\left(\left(\frac{a}{b}\right)\right)\right]\right)\right)$$

$$([\langle \{|\frac{a}{b}|\} \rangle])$$

ein Punkt statt des Klammertyps steht für einen Platzhalter. Das ist z.B. nützlich, wenn nur die linke Klammer benötigt wird.

- Zahlenräume wie die natürlichen Zahlen erzeugt man mit Hilfe $\backslash\mathrm{mathbb{b}}$:

$$\mathbb{N} \setminus \mathbb{Z} \setminus \mathbb{Q} \setminus \mathbb{R} \setminus \mathbb{I} \setminus \mathbb{N}_0$$

NZQRIN₀

- Matrix-Umgebungen dienen nicht nur zur Darstellung von Vektoren und Matrizen, sondern auch der Formatierung (z.B. bei Fallunterscheidungen)

```
\left|\begin{matrix}
1&2&3\\
4&5&6\\
7&8&9
\end{matrix}\right|
```

$$\begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{vmatrix}$$

```
\left(\begin{matrix}
1\\2\\3
\end{matrix}\right)
```

$$\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

$$x^k = \begin{matrix} \text{left} \\ \text{begin}\{matrix\} \\ 1 \text{ falls } k=0 \\ x \cdot \ldots \cdot \\ \text{end}\{matrix\} \\ \text{right.} \end{matrix}$$

$$x^k = \begin{cases} 1, & \text{falls } k = 0 \\ x \cdot \dots \cdot x, & \text{sonst} \end{cases}$$

- zu einigen griechischen Buchstaben gibt es eine alternative Darstellungsweise:

```
\epsilon\phi\pi\rho\sigma\theta\\
\varepsilon\varphi\varpi\rho\varsigma\vartheta
\varsigma\vartheta
```

$$\epsilon\phi\pi\rho\sigma\theta$$

$$\varepsilon\varphi\varpi\rho\varsigma\vartheta$$

- es gibt diverse Pfeile, die nur in Formeln verwendet werden können (d.h. auch für Folgepfeile im Fließtext ist eine Matheumgebung zu benutzen!):

```
\Rightarrow\Lefttrightarrow\Leftarrow\\
\rightarrow\leftrightharpoon\leftarrow\\
\longrightarrow\longleftarrow\\
\longleftarrow\\
\longmapsto\mapsto\\
\nearrow\searrow\curvearrowleft\\
\rightleftarrows
```

$$\Rightarrow\Leftrightarrow\Leftarrow$$

$$\rightarrow\leftrightarrow\leftarrow$$

$$\longrightarrow\longleftrightarrow\longleftarrow$$

$$\longmapsto\mapsto$$

$$\nearrow\searrow\curvearrowleft$$

$$\rightleftarrows$$

6.2.2 Tabellen

Tabellen erzeugt man mittels einer tabular-Umgebung:

```
\begin{tabular}{c|c|c|c|c|}
a&b&c&d\\
\hline
e&f&g&h
\hline
\end{tabular}
```

Listing 33: Tabular

Dabei stehen die c's für die Spalten. Das c bedeutet dabei nichts anderes als horizontal zentrierten Text in den Zellen. Es gibt demnach auch r und l. die vertikalen Striche stehen für die Trennstriche der Tabelle und können je nach Bedarf hinzugefügt, oder weggelassen werden. Auch doppelte Striche sind denkbar. Die & Zeichen leiten die nächste Zelle ein und wie gewohnt \\ die nächste Zeile. \hline erzeugt einen horizontalen Strich und ist analog zum vertikalen Trennstrich veränderbar.

Es empfiehlt sich die tabular-Umgebung zusätzlich in eine table-Umgebung zu verschachteln, was uns die Möglichkeit eröffnet, einen Titel für die Tabelle zu definieren:

```

\begin{table}
\begin{tabular}{c|c|c|c|}
a&b&c&d\\
\hline
e&f&g&h\\
\hline
\end{tabular}
\caption{Testtabelle}
\end{table}

```

Listing 34: Tabular

Nach dem caption-Befehl kann nun auch, wie auf Seite 14 angedeutet, ein label-Befehl stehen. Kompiliert sieht das wiederum so aus:

a	b	c	d
e	f	g	h

Tabelle 1: Testtabelle

Will man zusätzlich Zellen definieren, welche sich über mehrere Spalten erstrecken muss man den Quellcode wie folgt abändern:

```

\begin{table}
\begin{tabular}{c|c|c|c|}
a&b&c&d\\
\hline
e&\multicolumn{2}{|c|}{Text}&h\\
\hline
\end{tabular}
\caption{Testtabelle}
\end{table}

```

Listing 35: Multicolumn

Dabei steht das erste Argument des multicolumn-Befehls für die Anzahl der zu zusammenfassenden Spalten, das zweite für die Textausrichtung (vgl. Seite 21) und das dritte enthält den Text selbst. Das kompilierte Ergebnis ändert sich zu:

a	b	c	d
e	Text		h

Tabelle 2: Testtabelle

Fast vollkommen analog funktioniert das auch für mehrere Zeilen (allerdings ist es meist überflüssig die Textausrichtung zu definieren, ein \star steht anstatt dessen als neutraler Platzhalter):

```

\begin{table}
\begin{tabular}{c||c|c|c|}
a&\multirow{2}{*}{\rotatebox[origin=c]{-90}{Text}}&c&d\\
\cline{1-1}\cline{3-4}
e&g&h\\
\hline
\end{tabular}
\caption{Testtabelle}
\end{table}

```

Listing 36: Multirow

Um allerdings den multirow-Befehl nutzen zu können benötigen wir - im Gegensatz zum multicolumn-Befehl - ein zusätzliches usepackage:

```

\usepackage{multirow}

```

Listing 37: multirow-usepackage

Bei obigem Beispiel taucht zusätzlich der Befehl `\rotatebox` auf. Er bewirkt eine Drehung des zweiten Arguments um den Winkel, den das erste Argument vorgibt. In den eckigen Klammern kann man zusätzlich den Drehpunkt festlegen. Das kompilierte Ergebnis ändert sich abermals zu:

a		Text	c	d
e			g	h

Tabelle 3: Testtabelle

6.2.3 Grafiken

Um in L^AT_EX Grafiken einzufügen, wird wie auf Seite 5 Kapitel 3 erläutert, das graphicx usepackage benötigt. Wichtig ist allerdings, das Bild in den gleichen Ordner, wie die *.tex-Datei zu kopieren. Die Grundstruktur sieht nun folgendermaßen aus:

```

\includegraphics[width=10cm]{baer.jpg}

```

Listing 38: Grafiken einbinden

Wie auch schon bei den Tabellen gibt es hier eine passende Umgebung, mit Hilfe derer das Bild einen Titel, ein label, sowie eine Nummer zugeordnet bekommen kann; der `\centering` Befehl zentriert Bild und Bildunterschrift:

```

\begin{figure}
\centering
\includegraphics[width=10cm]{baer.jpg}
\caption{Das_ist_ein_gehillter_Baer_aus_dem_Muenchner_Zoo}
\label{Baer}
\end{figure}

```

Listing 39: multirow-usepackage

Das sieht dann kompiliert so aus (... wobei man sich, Dank des labels, jederzeit auf Bild 1 auf Seite 24 beziehen kann):



Abbildung 1: Das ist ein gehillter Bär aus dem Münchner Zoo

Will man mehrere Bilder nebeneinander anordnen, bietet sich je nach Konstellation eine table- oder eine minipage-Umgebung an. Die Verteilung zwei links, ein Bild rechts stellt sich als Code beispielsweise wie folgt dar (der Zeilenumbruch mit „\par“ unterbindet einen evtl. sichtbaren Abstand zwischen den beiden Bildern untereinander):


```

\begin{figure}[H]
\centering
\begin{minipage}{0.9\textwidth}
\begin{minipage}{0.6\textwidth}
\includegraphics[width=\textwidth]{baer.jpg}
\end{minipage}\begin{minipage}{0.4\textwidth}
\includegraphics[width=\textwidth]{baer.jpg}\small[-1px]
\includegraphics[width=\textwidth]{baer.jpg}
\end{minipage}
\end{minipage}
\end{minipage}
\caption{Das_sind_drei_gechillte_Baeren_aus_dem_Muenchner_Zoo}
\end{figure}

```

Listing 40: Bilderanordnung

Es bleibt anzumerken, dass im Quellcode absichtlich kein Zeilenumbruch zwischen `\end` und `\begin{minipage}` gemacht wurde, da so gewährleistet wird, dass die Bilder auch tatsächlich nebeneinander Platz finden:



Abbildung 2: Das sind drei gechillte Bären aus dem Münchner Zoo

Damit letztendlich auch die Größenverhältnisse passen, d.h. die beiden Bilder rechts die gleiche Breite besitzen und die beiden zusammen genommen die gleiche Höhe wie das Bild links, im Folgenden noch einer kleiner mathematischer Tip.

Als Demonstration werden jetzt drei unterschiedliche Bilder verwendet, welche folgende Abmaße besitzen:

- **Bild 1:** 2000×1333 Pixel
- **Bild 2:** 1214×937 Pixel
- **Bild 3:** 750×500 Pixel

Für die Breite b und die Höhe h des ersten Bildes gilt:

$$h_1 = \frac{1333}{2000} \cdot b_1$$

Für das zweite und dritte Bild gilt weiterhin:

$$b_2 = \frac{1214}{937} \cdot h_2$$

$$b_3 = \frac{750}{500} \cdot h_3$$

Die beiden Breiten rechts sollen nun identisch sein:

$$b_2 = b_3$$

$$\frac{1214}{937} \cdot h_2 = \frac{750}{500} \cdot h_3$$

$$h_3 = \frac{2428}{2811} \cdot h_2$$

Die Gesamthöhe beträgt also formelmäßig:

$$h_{ges} = h_2 + h_3$$

$$= \frac{5239}{2811} \cdot h_2$$

$$= \frac{5239}{2811} \cdot \frac{937}{1214} \cdot b_2$$

$$= \frac{5239}{3642} \cdot b_2$$

Diese soll nun gleich hoch sein, wie h_1 :

$$h_1 = h_2$$

$$\frac{1333}{2000} \cdot b_1 = \frac{5239}{3642} \cdot b_2$$

$$b_2 = \frac{78303}{169000} \cdot b_1$$

Wenn die Bilder also die gesamte Breite ausfüllen sollen ergibt sich:

$$b_1[\%] = \frac{1}{1 + \frac{78303}{169000}} = 0,68337\%$$

$$b_2[\%] = 1 - b_1[\%] = 0,31663\%$$

$$b_3[\%] = b_2[\%]$$

Zugegebenermaßen klingt diese Rechnung hier recht unnötig, liefert aber schöne Ergebnisse. Zudem besteht die Möglichkeit, falls man öfter eine derartige Anordnung benötigt, sich selbst Befehle zu definieren. Dazu in Kapitel EINFÜGEN auf Seite EINFÜGEN mehr. Hier noch das kompilierte Ergebnis obiger Rechnung:



Abbildung 3: Das sind drei gechillte Bären aus dem Münchner Zoo

6.2.4 Fußnoten

Um Fußnoten zu erzeugen existiert der Befehl `\footnote`, der sich im Fließtext anwenden lässt:

```
Das_ist_ein_Beispiel\footnote{und_das_eine_Fussnote}
```

Listing 41: Fußnoten

So kann eine Fußnote² aussehen.

Will man auch in einer Formel eine Fußnote verwenden, wird das ganze etwas komplizierter:

```
\begin{align*}
1+\left| \quad x\right| &=2\text{\texttt{\footnotemark}} \quad x=1 \\
\end{align*}
\footnotetext{$x=-1$ \quad prinzipiell_auch_moeglich}
```

Listing 42: Fußnoten in Matheumgebung

Durch das Setzen einer Fußnotenmarkierung, weiß \LaTeX , wo die Fußnote landen soll. Der zugehörige Text steht außerhalb. Das Ergebnis sieht genauso aus, wie bei einer normalen Fußnote:

²Fußnoten werden automatisch durchnummeriert und können auch Formeln x^2 (zwischen zwei $\text{\$}$ -Zeichen enthalten)

$$1 + |x| = 2 \text{ für } x = 1$$

$x = -1$ prinzipiell auch möglich

6.2.5 Zitate

Um Zitate im Text einfügen zu können bedarf es zwei Schritten:

1. *.bibtex-Datei erstellen und in die Quelldatei einbinden
2. Zitat aufrufen

Konkret erstellt man eine bibtex-Datei meistens mit einem Hilfsprogramm, z.B. Citavi, Endnote, Zotero, oder JabRef. Man sollte aber vorher prüfen, ob eine bibtex-Exportfunktion enthalten ist. Mit Citavi wurde beispielsweise der folgende Ausschnitt einer *.bibtex-Datei erzeugt:

```
@book{Gscheidle.2009,
  author={Gscheidle,      Rolf},
  year={2009},
  title={Fachkunde      Kraftfahrzeugtechnik},
  address={Haan-Gruiten},
  edition={29.,      neubearb.      Aufl.,      1.      Dr},
  publisher={Verl.      Europa-Lehrmittel      Nourney,Vollmer},
  isbn={978-3-80852-239-4},
  series={Europa-Fachbuchreihe_f{"u}r_Kraftfahrzeugtechnik}
}
```

Listing 43: *.bibtex-Schnipsel

Diese Datei muss nun in den gleichen Ordner, wie die *.tex-Datei verschoben werden. Als nächsten Schritt bedarf es einer Erweiterung im Header:

```
...
\usepackage{cite}
...
```

Listing 44: cite-usepackage

Die Zitierweise hängt dabei vom Dokumententyp und den Einstellungen im Header ab. Abschließend muss an das Ende des Dokuments, also **vor** `\end{document}`, noch ein Verweis auf die *.bibtex-Datei stehen:

```
...
\bibliography{Name_der_bibtex_Datei}
\bibliographystyle{plain}
\end{document}
```

Listing 45: Literaturdatenbank

Mögliche Abwandlungen zu plain, sind alpha, gerplain, dinat und viele weitere. Je nach Anforderungen verwenden.

Jetzt fehlt noch der Aufruf eines Zitats (mittels seines bibtex-Keys, welcher im Literaturverwaltungsprogramm definiert wurde. Evtl. muss L^AT_EX-Unterstützung aktiviert werden.):

```
Text\cite{Gscheidle.2009}
Text\cite[S. 12]{Gscheidle.2009}
```

Listing 46: Zitataufruf

Nun kann auch zitiert werden[1]. Oder mit zusätzlicher Seitenangabe[1, S. 12]. Es muss vor dem Kompilieren lediglich F11 gedrückt werden, damit sich das Literaturverzeichnis aktualisiert. Danach zweimal kompilieren.

6.2.6 Eigene Befehle definieren

Oft muss man den selben Befehl, oder die selbe Befehlsfolge sehr oft wiederholen. Das ist recht mühsam, kostet Zeit und macht zu allem Überfluss den Code unleserlich. Deshalb macht es an dieser Stelle Sinn kurz darauf einzugehen, wie man eigene Befehle definieren kann.

Die Definition eines Befehls muss im Header erfolgen und sieht z.B. folgendermaßen aus:

```
\newcommand{\nm}{\frac{\text{N}}{\text{mm}^2}}
```

Listing 47: Eigener Befehl

Dieser Befehl ermöglicht uns jetzt jederzeit mittels \nm ein $\frac{N}{mm^2}$ Formelelement einzufügen. Auch besteht die Möglichkeit Werte an den Befehl zu übergeben, z.B.:

```
\newcommand{\integral}[3]{\int_{\limits_{#1}}^{#2} \sin(x) dx}
```

Listing 48: Eigener Befehl mit Parametern

So bewirkt der Befehl \integral{1}{1-x}{\sin(x)}:

$$\int_1^{1-x} \sin(x) dx$$

Raute und Nummer stehen dabei für den jeweiligen Parameter, welcher der Reihe nach übergeben wird. In den eckigen Klammern in der Befehlsdefinition ist die Anzahl dieser Parameter festgelegt.

6.2.7 Plotten mit \LaTeX

Mit Hilfe zweier weiterer usepackages ist es möglich, Funktionen, Daten und statistische Auswertungen in Grafiken innerhalb des \LaTeX -Dokuments zu verpacken. Die besagten usepackages lauten:

```
\usepackage{tikz}
\usepackage{pgfplots}
```

Listing 49: usepackages für Plots

Fügt man zusätzlich folgenden, zugegebenermaßen komplizierten, Code (Copy&Paste reicht zum Glück) in den Header ein, definiert man sich eine eigene Pfeilspitze, welche der gewohnten Ingenieurspfeilspitze (also wie in technischen Zeichnungen) möglichst ähnlich kommt:

```
\pgfarrowsdeclare{ingenieurspfeil}{ingenieurspfeil}
{
\newdimen\len
\len=\pgfgetarrowoptions{ingenieurspfeil}
\pgfarrowslefttextend{-0.4\len}
\pgfarrowsrighttextend{0.6\len}
}
{
\newdimen\len
\len=\pgfgetarrowoptions{ingenieurspfeil}
\pgfpathmoveto{\pgfqpoint{0.6\len}{0\len}}
\pgfpathcurveto
{\pgfqpoint{0.35\len}{0.05\len}}
{\pgfqpoint{-0.1\len}{0.15\len}}
{\pgfqpoint{-0.4\len}{0.375\len}}
\pgfpathcurveto
{\pgfqpoint{-0.15\len}{0.1\len}}
{\pgfqpoint{-0.15\len}{-0.1\len}}
{\pgfqpoint{-0.4\len}{-0.375\len}}
\pgfpathcurveto
{\pgfqpoint{-0.1\len}{-0.15\len}}
{\pgfqpoint{0.35\len}{-0.05\len}}
{\pgfqpoint{0.6\len}{0\len}}
\pgfusepathqfill
}
\pgfsetarrowoptions{ingenieurspfeil}{8pt}
\pgfkeys{/tiplen/.default=8pt,
/tiplen/.code={\pgfsetarrowoptions{ingenieurspfeil}{#1}}}

```

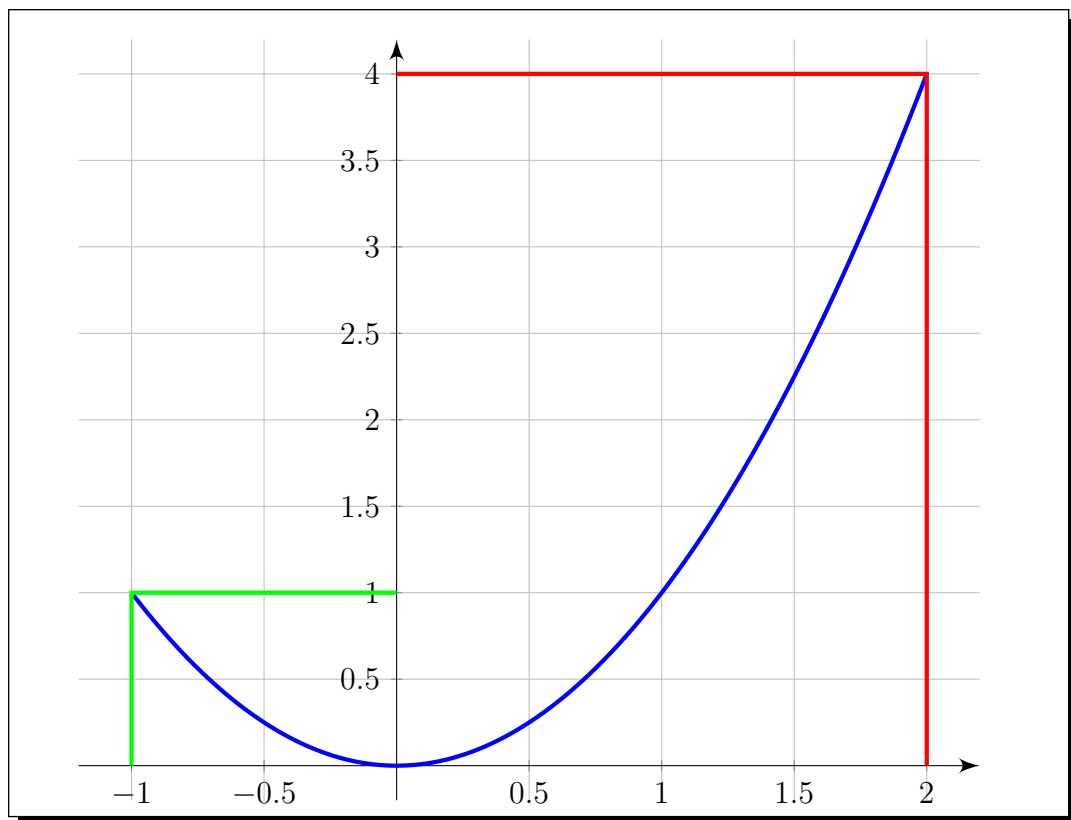
Listing 50: eigene Pfeilspitze

Ein beispielhafter 2D-Plot sieht nun folgendermaßen aus:

```
\begin{tikzpicture}
\begin{axis}[
width=15cm,
grid,
axis      x      line=center,
axis      y      line=center,
xmin=-1.2,
xmax=2.2,
ymin=-0.2,
ymax=4.2,
axis      line    style={-ingenieurspfeil}]
\addplot[domain=-1:2,color=blue,
ultra    thick,samples=500]{x^2};
\draw[red,ultra    thick](axis      cs:2,0)|-(axis      cs:0,4);
\draw[green,ultra    thick](axis      cs:-1,0)|-(axis      cs:0,1);
\end{axis}
\end{tikzpicture}
```

Listing 51: 2D-Plot

... und kompiliert:



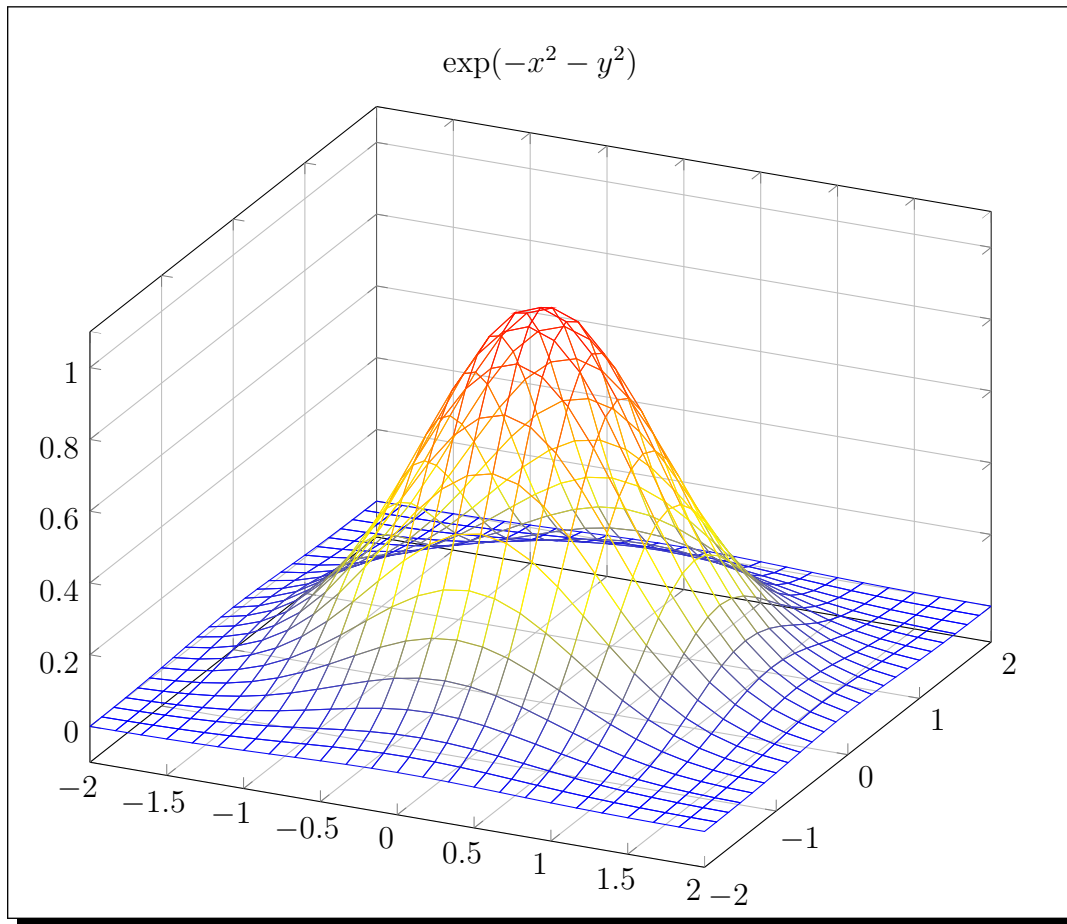
Ausführliche Erklärungen bleiben an dieser Stelle vorerst ausgespart.

Und abschließend noch ein Beispiel in 3D, als Code:

```
\begin{tikzpicture}
\begin{axis}[
width=13.5cm,
grid,
title=$\exp(-x^2-y^2)$]
\addplot3[mesh,domain=-2:2]{exp(-x^2-y^2)};
\end{axis}
\end{tikzpicture}
```

Listing 52: 2D-Plot

... und kompiliert:



Literatur

- [1] Rolf Gscheidle. *Fachkunde Kraftfahrzeugtechnik*. Europa-Fachbuchreihe für Kraftfahrzeugtechnik. Verl. Europa-Lehrmittel Nourney, Vollmer, Haan-Gruiten, 29., neubearb. Aufl., 1. dr. edition, 2009.