Wissenschaftliche Arbeiten mit LaTeX an der Fakultät NXT

Eine Einführung in die Klasse nxtthesis

Fakultät NXT Nachhaltigkeit und Technologie
Hochschule Reutlingen

24. September 2026

Vorbemerkung

Diese Anleitung richtet sich an Studierende der Fakultät NXT Nachhaltigkeit und Technologie der Hochschule Reutlingen (HSRT), die ihre Abschlussarbeit mit der Klasse nxtthesis schreiben. Sie ist keine allgemeine LaTeX-Einführung; davon gibt es gute und umfangreiche (Lamport, 1994; Mittelbach & Fischer, 2023). Sie beschreibt, was die Klasse mitbringt, wie man es benutzt und warum die Dinge so geregelt sind, wie sie geregelt sind. Die vollständige technische Dokumentation der Klasse selbst steht in `nxtlatex.pdf` (Reichenberger, 2026).

Die Anleitung ist selbst mit nxtthesis gesetzt. Jede Abbildung, jede Tabelle, jede Formel und jeder Verzeichniseintrag in diesem Portable Document Format (PDF) ist mit denselben Mitteln entstanden, die hier beschrieben werden. Wer wissen möchte, wie eine bestimmte Stelle gemacht ist, findet die Antwort in der Quelldatei `ThesisanleitungNXT.tex` oder in den Kapiteldateien unter `Kapitel/`.

Ein zweites, ausführlicheres Beispiel liegt dem Paket unter `Beispielthesis/` bei. Es zeigt eine vollständige, plausible Arbeit mit Rechnung, Abbildungen und Anhang. Wo diese Anleitung ein Verfahren nur skizziert, ist dort die ausgeführte Fassung zu finden.

INHALTSVERZEICHNIS

TABELLENVERZEICHNIS

ABBILDUNGSVERZEICHNIS

ABKÜRZUNGSVERZEICHNIS

APA	American Psychological Association
DOI	Digital Object Identifier
HSRT	Hochschule Reutlingen
MSE	Mean Squared Error
NXT	NXT Nachhaltigkeit und Technologie
PDF	Portable Document Format
VSC	Visual Studio Code

SYMBOLVERZEICHNIS

α	Abkühlfaktor
$C_{\max}$	Gesamtdauer eines Ablaufplans (min)
E	Energiebedarf (kW h)

1 Erste Schritte

1.1 Was LaTeX ist – und was es nicht ist

LaTeX ist kein Textverarbeitungsprogramm. In Word sieht man beim Schreiben das fertige Layout und bearbeitet es direkt. In LaTeX schreibt man eine reine Textdatei, in der neben dem Inhalt Anweisungen stehen, die die *Bedeutung* der Textteile beschreiben: `\chapter` sagt „hier beginnt ein Kapitel“, nicht „setze diese Zeile in 16 pt fett“. Ein Übersetzungsprogramm macht daraus eine PDF-Datei. Wie ein Kapitel aussieht, entscheidet die Dokumentklasse – hier nxtthesis.

Diese Trennung ist der eigentliche Gewinn. Sie bedeutet, dass man beim Schreiben schreibt und nicht formatiert, dass das Layout im ganzen Dokument gleich bleibt, und dass die Vorgaben der Fakultät an einer einzigen Stelle stehen statt in tausend einzelnen Absätzen. Sie bedeutet auch, dass sich Querverweise, Verzeichnisse und Literaturangaben selbst verwalten: Wer ein Kapitel verschiebt, muss keine Nummer nachziehen.

Der Preis ist eine Umstellung. Man sieht das Ergebnis erst nach dem Übersetzen, und Fehlermeldungen sind gewöhnungsbedürftig. Für eine Abschlussarbeit mit Formeln, Abbildungen, Tabellen, Literatur und Verzeichnissen zahlt sich der Preis in aller Regel aus.

LuaLaTeX ist die Variante von LaTeX, die nxtthesis voraussetzt. Sie versteht *UTF-8* unmittelbar – Umlaute schreibt man einfach als Umlaute – und kann die Schriften des Betriebssystems verwenden. Genau darauf beruhen die Schriftoptionen der Klasse. Die älteren Programme `pdflatex` und `xelatex` funktionieren mit nxtthesis *nicht*.

1.2 Installation

LaTeX kommt als Distribution. Empfohlen ist *TeX Live*, die es für alle Systeme gibt:

- **Windows:** TeX Live von `tug.org/texlive`, oder ersatzweise MiKTeX.
- **macOS:** MacTeX (TeX Live mit macOS-Zubehör) von `tug.org/mactex`.
- **Linux:** TeX Live über die Paketverwaltung oder direkt von `tug.org`. Die Distributionspakete sind oft veraltet; die Installation von `tug.org` ist meist die bessere Wahl.

Wählen Sie die vollständige Installation. Sie ist einige Gigabyte groß, erspart aber das Nachinstallieren einzelner Pakete an dem Abend, an dem die Arbeit fertig werden soll.

Die Klasse selbst liegt auf *CTAN* und ist in einer vollständigen Installation von TeX Live oder MiKTeX bereits enthalten. Wer eine neuere Fassung aus den Quellen braucht, kopiert sie mit `make install` in den lokalen `texmf`-Baum. Die NXT-Logos bringt das Paket selbst mit, als Zeichen im NXT Logo Font; herunterladen muss man nichts (Abschnitt 11.2).

1.3 Das Minimaldokument

Eine Arbeit mit nxtthesis beginnt so:

```
\documentclass[german,stix]{nxtthesis}

\title{Titel der Arbeit}
\author{Vorname Nachname}
\akademischergrad{Bachelor of Science}
\studiengang{Nachhaltige Technologie}
\matrikelnummer{1234567}
\abgabedatum{31. Maerz 2027}
\erstpruef{Prof. Dr. Erika Mustermann}
\zweitpruef{Prof. Dr. Max Mustermann}

\begin{document}
\maketitle
\tableofcontents
\hauptteil

\chapter{Einleitung}
Hier beginnt die Arbeit.

\eigenständigkeitserklärung[ki=keine]
\end{document}
```

Listing 1.1: Ein lauffähiges Minimaldokument

Mehr braucht es nicht für ein vollständiges, gesetztes Dokument mit Titelseite, Inhaltsverzeichnis, Seitenzahlen und der Erklärung zur wissenschaftlichen Arbeit. Kapitel 3 geht die einzelnen Angaben durch.

1.4 Übersetzen

Aus der Quelldatei wird die PDF-Datei mit `lualatex`.

1.4.1 Wo man das eintippt

Die folgenden Zeilen sind keine LaTeX-Befehle und gehören nicht in die Quelldatei. Es sind Programmaufrufe, und sie werden in der *Kommandozeile* eingegeben – also in dem Fenster, in dem man Programme durch ihren Namen startet statt durch einen Doppelklick:

- **Windows:** *Eingabeaufforderung* oder *PowerShell* (Startmenü, dann den Namen tippen).
- **macOS:** *Terminal* (Programme → Dienstprogramme).
- **Linux:** das Terminal der Arbeitsumgebung.

Zuerst wechselt man in das Verzeichnis, in dem die Quelldatei liegt. Das geschieht mit `cd` (*change directory*), gefolgt vom Pfad:

```
cd Dokumente/Abschlussarbeit
```

Listing 1.2: In das Verzeichnis der Arbeit wechseln

Ein Verzeichnis zurück geht mit `cd ..`; `ls` (Windows: `dir`) zeigt, was im aktuellen Verzeichnis liegt. Wer sich den Pfad sparen will, zieht den Ordner aus dem Dateimanager ins Terminalfenster – der Pfad steht dann da.

In den meisten Fällen tippt man das allerdings gar nicht selbst: Ein Editor mit LaTeX-Unterstützung erledigt die ganze Folge auf Knopfdruck. Abschnitt 1.5 nennt die gängigen, Kapitel 11 zeigt die Einrichtung für Visual Studio Code im Einzelnen. Wissen sollte man trotzdem, was dabei geschieht – spätestens dann, wenn es klemmt und die Fehlermeldung im Protokoll steht.

1.4.2 WARUM MEHRERE LÄUFE

Für eine vollständige Arbeit reicht ein Lauf nicht:

```
lualatex  ThesisanleitungNXT     # Text setzen, Hilfsdateien schreiben
biber     ThesisanleitungNXT     # Literaturverzeichnis erzeugen
makeindex ThesisanleitungNXT     # Index sortieren
lualatex  ThesisanleitungNXT     # Literatur und Index einbauen
lualatex  ThesisanleitungNXT     # Verweise und Seitenzahlen richtigstellen
```

Listing 1.3: Die volle Folge von Programmläufen

Der Grund für die Wiederholung liegt in der Arbeitsweise von LaTeX. Das Programm liest die Datei einmal von vorne nach hinten. Wenn auf Seite 3 `\ref{kap:ergebnisse}` steht und das Kapitel auf Seite 57 beginnt, kann der erste Lauf die Nummer noch nicht kennen. Er schreibt sie stattdessen in eine Hilfsdatei (`.aux`). Der zweite Lauf liest diese Hilfsdatei und setzt die Nummer ein – wodurch sich der Text minimal ändert, was den Seitenumbruch verschieben kann, weshalb ein dritter Lauf nötig sein kann. Dasselbe gilt für das Inhaltsverzeichnis: Es steht vorne, kennt aber die Seitenzahlen von hinten.

`biber` und `makeindex` sind eigene Programme. `biber` liest, welche Quellen zitiert wurden, sucht sie in der `.bib`-Datei und erzeugt das fertig formatierte Literaturverzeichnis; `makeindex` sortiert die Indexeinträge. Beide brauchen die Hilfsdateien des ersten Laufs, und ihr Ergebnis kann erst der nächste `lualatex`-Lauf einbauen.

Das *Abkürzungsverzeichnis* ist die Ausnahme: Das Paket acro, das die Klasse fertig eingerichtet mitbringt, braucht kein eigenes Programm. Zwei `lualatex`-Läufe genügen.

Meldet LaTeX am Ende „Rerun to get cross-references right“, fehlt noch ein Lauf. Die Beispielthesis bringt ein `Makefile` mit, das die Folge automatisiert; ein einfaches `make` genügt dann. Alternativ nimmt einem `latexmk -lualatex` die Buchführung ab.

1.5 EDITOR

LaTeX schreibt man in einem Texteditor, nicht in einer Textverarbeitung. Verbreitet sind:

- **TeXstudio** – fertig eingerichtet, plattformübergreifend, mit eingebautem PDF-Betrachter. Der bequemste Einstieg.

- **VS Code** mit der Erweiterung *LaTeX Workshop* – wer ohnehin damit programmiert, bleibt in einer Umgebung.
- **TeXShop** (macOS), **Overleaf** (im Browser).

Drei Einstellungen sind wichtig, unabhängig vom Editor. Erstens: Als Übersetzungsprogramm muss `lualatex` eingestellt sein, nicht `pdflatex`. Zweitens: Die Dateien müssen als *UTF-8* gespeichert werden – das ist heute meist Voreinstellung, aber ein Blick lohnt, sonst werden aus Umlauten kaputte Zeichen. Drittens: Der *SyncTeX*-Sprung zwischen Quelle und PDF (meist Doppelklick oder `Strg`+Klick) spart bei einer hundertseitigen Arbeit sehr viel Zeit.

Ein letzter Hinweis, der keine Editor-Einstellung ist, aber genauso wichtig: Legen Sie die Arbeit von Anfang an in eine *Versionsverwaltung*, etwa Git. Eine Abschlussarbeit läuft über Monate, und die Frage „was stand vorletzte Woche in Abschnitt 4.2?“ kommt zuverlässig. Die Quelldateien sind reiner Text und dafür bestens geeignet – ein weiterer Vorteil des Verfahrens.

2 GRUNDLAGEN VON LaTeX

Das Minimaldokument aus Kapitel 1 übersetzt. Damit ist die Technik erledigt, nicht aber die Sprache. Dieses Kapitel stellt die Bauteile vor, aus denen jede LaTeX-Quelldatei besteht: Befehle, Umgebungen, Gruppen, Verweise – und die Handvoll Regeln, an denen Anfänger regelmäßig auflaufen. Wer sie einmal verstanden hat, liest den Rest dieser Anleitung ohne Mühe.

Es lohnt sich, dieses Kapitel wirklich zu lesen und nicht zu überfliegen. Fast alles, was in einer Abschlussarbeit später schiefgeht, geht hier schief.

2.1 BEFEHLE

Ein *Befehl* ist eine Anweisung an LaTeX. Er beginnt immer mit einem *Backslash* \, gefolgt vom Namen des Befehls:

```
Stuttgart, den \today
```

Stuttgart, den 24. September 2026

Listing 2.1: Der einfachste Fall: ein Befehl ohne Argument

`\today` setzt das aktuelle Datum ein – heute also den 24. September 2026. Das ist der ganze Aufbau: Backslash, Name, fertig.

2.1.1 ARGUMENTE

Die meisten Befehle brauchen etwas, worauf sie wirken. Dieses *Argument* steht in geschweiften Klammern unmittelbar hinter dem Namen:

```
Das Verfahren ist \textbf{robust} gegen Ausreißer.
```

Das Verfahren ist **robust** gegen Ausreißer.

Listing 2.2: Ein Befehl mit Argument

`\textbf` bekommt „robust“ als Argument und setzt es fett. Manche Befehle nehmen mehrere Argumente; dann folgt eine geschweifte Klammer auf die andere, ohne Komma und ohne Leerzeichen dazwischen. `\frac{1}{2}` etwa ergibt in einer Formel $\frac{1}{2}$: erst der Zähler, dann der Nenner. Die Reihenfolge ist für jeden Befehl festgelegt und muss stimmen.

Daneben gibt es *optionale Argumente*. Sie stehen in eckigen Klammern, kommen vor den geschweiften und dürfen weggelassen werden:

```
\includegraphics{ablaufplan}            % Bild in Originalgroesse
\includegraphics[width=5cm]{ablaufplan} % Bild auf 5cm Breite skaliert
```

Listing 2.3: Ein optionales Argument

Beide Zeilen sind richtig. Ohne die eckige Klammer greift der Vorgabewert des Befehls, hier also die Originalgröße. Optionale Argumente sind der übliche Weg, Feinheiten einzustellen, ohne dass der Normalfall umständlich wird; man begegnet ihnen überall in dieser Anleitung.

2.1.2 Groß- und Kleinschreibung

LaTeX unterscheidet in Befehlsnamen streng zwischen Groß- und Kleinschreibung. `\LaTeX` ist ein Befehl, `\latex` und `\Latex` sind es nicht – sie führen zu einer Fehlermeldung. Ebenso ist `\tableofcontents` etwas anderes als `\TableOfContents`. Wer eine Fehlermeldung über einen unbekannten Befehl bekommt, hat sich in vier von fünf Fällen genau hier vertippt.

2.1.3 Wo ein Befehl endet

Ein Befehlsname besteht aus Buchstaben. Er endet dort, wo das erste Zeichen steht, das kein Buchstabe ist – eine Ziffer, ein Satzzeichen, eine geschweifte Klammer oder eben ein Leerzeichen. Woher soll LaTeX sonst wissen, ob `\todaymorgen` den Befehl `\today` meint oder einen Befehl namens `\todaymorgen`?

Diese Regel hat eine Nebenwirkung, die jeden Anfänger einmal erwischt: Das Leerzeichen, das den Namen beendet, wird verbraucht. Es erscheint nicht im Text.

```
Stand \today und damit aktuell.

Stand \today{} und damit aktuell.
```

Stand 24. September 2026und damit aktuell.
Stand 24. September 2026 und damit aktuell.

Listing 2.4: Das verschluckte Leerzeichen

In der ersten Zeile beendet das Leerzeichen den Namen `today` und ist danach aufgebraucht – im PDF steht das Datum direkt am „und“ geklebt. In der zweiten Zeile beenden die leeren geschweiften Klammern den Namen, das Leerzeichen dahinter ist ein gewöhnliches Leerzeichen und überlebt. Deshalb schreibt man Befehle ohne Argument im Fließtext grundsätzlich mit angehängten leeren Klammern: `\today{}`, `\LaTeX{}`, `\ldots{}`.

Vor einem Satzzeichen braucht man sie nicht, denn ein Punkt ist kein Buchstabe und beendet den Namen von selbst. Man darf sie trotzdem setzen; sie schaden nie. Wer sich die Klammern gleich angewöhnt, spart sich die Suche nach zusammengeklebten Wörtern in der Endfassung. Kapitel 9 kommt bei den eigenen Makros auf denselben Punkt zurück.

2.1.4 STERNFORMEN

Viele Befehle haben eine Zwillingsform, deren Name auf ein Sternchen endet. Sie tut fast dasselbe, aber in einer abgewandelten Variante: `\section*` setzt eine Überschrift ohne Nummer und ohne Eintrag ins Inhaltsverzeichnis, `\caption*` eine Bildunterschrift ohne Nummer. Was der Stern genau bewirkt, hängt vom Befehl ab; ein Muster gibt es aber: Der Stern schaltet die automatische Zählung ab. Abschnitt 2.4 zeigt den wichtigsten Fall.

2.2 UMGEBUNGEN

Ein Befehl wirkt auf ein Argument. Wenn etwas auf einen ganzen Textabschnitt wirken soll – eine Aufzählung, ein Zitat, eine Tabelle –, wären geschweifte Klammern über zwanzig Zeilen unübersichtlich. Dafür gibt es die *Umgebung*. Sie hat einen Anfang und ein Ende:

```
\begin{quote}
Alles, was hier steht, wird eingerückt gesetzt.
\end{quote}
```

> Alles, was hier steht, wird eingerückt gesetzt.

Listing 2.5: Der Aufbau einer Umgebung

`\begin` und `\end` bekommen denselben Namen als Argument. Zwischen ihnen gelten die Regeln der Umgebung. Auch Umgebungen können Argumente haben, verpflichtende wie optionale; sie stehen hinter dem `\begin`.

Umgebungen sind das zweite Grundgerüst der Sprache, und man begegnet ihnen ständig: `itemize` und `enumerate` für Aufzählungen, `quote` für Zitate, `tabular` für Tabellen, `equation` für Formeln, `figure` und `table` für Abbildungen und Tabellen, die sich eine Stelle im Umbruch suchen. Sogar das Dokument selbst ist eine: Alles zwischen `\begin{document}` und `\end{document}` ist der Text der Arbeit.

2.2.1 AUFZÄHLUNGEN

Zwei Umgebungen braucht man sofort. `itemize` erzeugt eine Aufzählung mit Spiegelstrichen, `enumerate` eine mit Nummern. Die einzelnen Punkte beginnen jeweils mit `\item`:

```
\begin{itemize}
  \item Der Datensatz stammt aus der Fertigung.
  \item Er umfasst zwölf Monate.
  \item Fehlende Werte wurden entfernt.
\end{itemize}
```

- Der Datensatz stammt aus der Fertigung.
- Er umfasst zwölf Monate.
- Fehlende Werte wurden entfernt.

Listing 2.6: Aufzählung mit Spiegelstrichen

```
\begin{enumerate}
  \item Daten einlesen
  \item Ausreißer entfernen
  \item Modell schätzen
\end{enumerate}
```

1. Daten einlesen
2. Ausreißer entfernen
3. Modell schätzen

Listing 2.7: Nummerierte Aufzählung

Die Nummern in `enumerate` vergibt LaTeX selbst. Wer nachträglich einen Punkt einschiebt, muss nichts nachziehen – dasselbe Prinzip wie bei den Kapitelnummern. Ob die Spiegelstriche Punkte, Striche oder Kreise sind und wie weit eingerückt wird, entscheidet die Klasse. Man kann es ändern; man sollte es in einer Abschlussarbeit lassen.

2.2.2 VERSCHACHTELN

Umgebungen dürfen ineinander stehen. LaTeX passt die Einrückung und die Markierung der inneren Ebene selbst an:

```
\begin{enumerate}
  \item Vorverarbeitung
    \begin{itemize}
      \item Ausreißer entfernen
      \item Werte normieren
    \end{itemize}
  \item Modellbildung
\end{enumerate}
```

1. Vorverarbeitung
    - Ausreißer entfernen
    - Werte normieren
2. Modellbildung

Listing 2.8: Eine Aufzählung in einer Aufzählung

Dabei gilt eine Regel ohne Ausnahme: Umgebungen müssen sauber ineinander liegen, wie Schachteln. Die innere muss geschlossen sein, bevor die äußere schließt. `\begin{enumerate}` … `\begin{itemize}` … `\end{enumerate}` … `\end{itemize}` ist falsch und führt zu einer Fehlermeldung, die selten dort steht, wo der Fehler ist.

Und: Jedes `\begin` braucht sein `\end` mit exakt demselben Namen. Ein vergessenes `\end{itemize}` merkt LaTeX oft erst am Ende des Dokuments und meldet dann etwas über `\end{document}` – eine der irreführendsten Meldungen überhaupt. Die meisten Editoren setzen das `\end` deshalb automatisch, sobald man das `\begin` getippt hat. Nehmen Sie die Hilfe an.

2.3 AUSZEICHNEN IM TEXT

Ein einzelnes Wort hervorzuheben ist so alltäglich, dass man selten darüber nachdenkt. In LaTeX lohnt es sich, genau hier einmal nachzudenken, denn es gibt zwei Sorten von Befehlen dafür, und sie unterscheiden sich in einem Punkt, der die ganze Arbeitsweise von LaTeX auf ein Wort zusammenzieht.

Die eine Sorte beschreibt das *Aussehen*:

- `\textbf{…}` – **fett**
- `\textit{…}` – *kursiv*
- `\texttt{…}` – `Schreibmaschine`
- `\textsf{…}` – serifenlos

Die andere Sorte beschreibt die *Bedeutung*. Sie besteht im Wesentlichen aus einem Befehl:

- `\emph{…}` – *hervorgehoben*

`\textit` sagt „setze das kursiv“. `\emph` sagt „das hier ist wichtig“ und überlässt es der Dokumentklasse, wie sie das zeigt. Im deutschen Buchsatz ist die Antwort kursiv, weshalb beide auf dem Papier gleich aussehen. Der Unterschied liegt nicht im Ergebnis, sondern in der Aussage – und genau das ist der Gedanke aus Abschnitt 1.1, angewendet auf eine einzelne Textstelle: Man schreibt hin, was etwas *ist*, nicht, wie es aussehen soll.

Das ist keine Prinzipienreiterei. Es hat zwei handfeste Folgen.

Die Klasse behält die Kontrolle. Wenn die Fakultät morgen entscheidet, Hervorhebungen sollen gesperrt statt kursiv gesetzt werden, ändert sich eine Zeile in nxtthesis und jede Hervorhebung der Arbeit folgt. Jedes `\textit` müsste dagegen von Hand nachgezogen werden – und man fände nicht alle, weil man `\textit` nicht ansieht, ob es eine Hervorhebung, ein Werktitel oder ein fremdsprachiges Wort war.

`\emph` kann verschachteln, `\textit` nicht. Steht eine Hervorhebung in einem ohnehin kursiven Text, hilft mehr Kursivschrift nicht weiter. `\emph` weiß das und schaltet zurück auf aufrecht:

```
\emph{Das Verfahren ist \emph{immer} zulässig.}

\textit{Das Verfahren ist \textit{immer} zulässig.}
```

Das Verfahren ist immer *zulässig.*
Das Verfahren ist immer zulässig.

Listing 2.9: Hervorhebung in der Hervorhebung

Im gesetzten Ergebnis sieht man den Unterschied: In der ersten Zeile steht das innere „immer“ aufrecht und sticht dadurch heraus. In der zweiten Zeile bleibt alles kursiv – die innere Hervorhebung verpufft. `\textit` kann nichts dafür; es hat getan, was es sollte, nämlich kursiv setzen.

Die Regel für eine Abschlussarbeit ist deshalb kurz: Zum Hervorheben nehmen Sie `\emph`, immer. `\textbf` und `\textit` bleiben den Fällen vorbehalten, in denen es wirklich um die Schriftform geht und nicht um Betonung – `\texttt` für einen Dateinamen oder ein Stück Quelltext ist so ein Fall.

Dazu kommt eine Vorgabe: Die Richtlinie der Fakultät erlaubt zum Hervorheben im Fließtext ausdrücklich nur Kursivschrift. Fettdruck und Unterstreichung sind nicht zulässig. `\emph` liefert genau das Erlaubte, ohne dass man darüber nachdenken muss – ein weiterer Grund, es zur Gewohnheit zu machen. Unterstreichen ist ohnehin ein Erbe der Schreibmaschine, die nichts anderes konnte; im Bleisatz hat es nie jemand getan.

2.4 GLIEDERUNG

Die Überschriften einer Arbeit setzt man nicht, man erklärt sie. LaTeX kennt eine feste Hierarchie:

```
\chapter{Ergebnisse}          % Ebene 1, beginnt eine neue Seite
\section{Laufzeiten}          % Ebene 2
```

```
\subsection{Kleine Instanzen}    % Ebene 3
\subsubsection{Sonderfaelle}     % Ebene 4
```

Listing 2.10: Die Ebenen der Gliederung

Man schreibt nur, was die Überschrift *ist* – ein Kapitel, ein Abschnitt –, und der Rest ergibt sich: Schriftgröße, Abstand davor und danach, Seitenumbruch, Nummer, Eintrag ins Inhaltsverzeichnis, Eintrag in die PDF-Lesezeichen.

Die *Nummerierung* verwaltet LaTeX vollständig selbst. Verschiebt man ein Kapitel, ändern sich alle Nummern dahinter, alle Einträge im Inhaltsverzeichnis und alle Verweise im Text – ohne Zutun. Das ist der Punkt, an dem Word bei einer hundertseitigen Arbeit üblicherweise Ärger macht.

Deshalb die wichtigste Regel dieses Abschnitts: *Formatieren Sie eine Überschrift niemals von Hand.* Eine Zeile

```
\textbf{\large 4.2 Laufzeiten}
```

Listing 2.11: So nicht

sieht im PDF vielleicht aus wie eine Überschrift, ist aber keine. Sie steht nicht im Inhaltsverzeichnis, sie bekommt kein Lesezeichen, sie zählt nicht mit, man kann nicht auf sie verweisen, und die „4.2“ wird falsch, sobald davor etwas dazukommt. Man hat sich mit einer Zeile aus allem herausgeschrieben, wofür man LaTeX überhaupt benutzt.

2.4.1 Sternformen

Manche Überschriften sollen keine Nummer tragen – ein Vorwort etwa. Dafür gibt es die Sternform:

```
\chapter*{Vorwort}
```

Listing 2.12: Eine Ueberschrift ohne Nummer

Das Kapitel erscheint dann ohne Nummer und *auch nicht* im Inhaltsverzeichnis. Letzteres überrascht regelmäßig, ist aber folgerichtig: Ein Verzeichniseintrag ohne Nummer wäre schwer auffindbar. Wo ein unnummeriertes Kapitel trotzdem ins Verzeichnis gehört – beim Literaturverzeichnis etwa –, erledigt das die Klasse bereits. Für eine gewöhnliche Arbeit brauchen Sie die Sternform selten.

2.4.2 Das optionale Argument

Manche Überschriften sind lang. Im Inhaltsverzeichnis und in der Kopfzeile stört das:

```
\chapter[Simulationsergebnisse]{Ergebnisse der Simulation unter
  Beruecksichtigung stochastischer Bearbeitungszeiten}
```

Listing 2.13: Kurzfassung fuers Verzeichnis

Das optionale Argument in eckigen Klammern ist die Kurzfassung; sie wandert ins Inhaltsverzeichnis und in die Kopfzeile. Das verpflichtende Argument in geschweiften Klammern bleibt die Überschrift im Text. Fehlt das optionale Argument, wird die lange Fassung überall verwendet.

Kapitel 3 geht die Gliederungsbefehle im Zusammenhang einer ganzen Arbeit noch einmal durch – mitsamt der Frage, wie tief man sinnvollerweise gliedert.

2.5 VERWEISEN: \LABEL UND \REF

Wenn es ein einzelnes Argument für LaTeX gäbe, dann dieses. Ein *Querverweis* in einer Arbeit – „siehe Abbildung 4.3 auf Seite 57“ – ist von Hand nicht zu halten. Es gibt in einer Abschlussarbeit leicht hundert davon, und jede eingeschobene Abbildung, jedes verschobene Kapitel, jeder gewachsene Absatz macht ein paar davon falsch. Falsche Verweise fallen in der Verteidigung auf.

LaTeX löst das mit zwei Befehlen. `\label` setzt eine Marke, `\ref` holt die Nummer der Marke:

```
\chapter{Ergebnisse der Simulation}
\label{kap:ergebnisse}
...
Die Auswertung in Kapitel~\ref{kap:ergebnisse} auf
Seite~\pageref{kap:ergebnisse} zeigt, dass ...
```

Listing 2.14: Marke setzen, Nummer holen

`\ref` liefert die Nummer, `\pageref` die Seitenzahl. Der Name in den geschweiften Klammern ist frei wählbar; er erscheint nirgends im PDF und dient nur LaTeX als Adresse. Was dabei herauskommt, ist immer richtig, weil LaTeX beide Enden desselben Verweises kennt.

Marken kann man an alles hängen, was gezählt wird: Kapitel, Abschnitte, Abbildungen, Tabellen, Gleichungen, Quelltexte, Fußnoten. Das Verfahren ist überall dasselbe.

2.5.1 DIE REIHENFOLGE – DER HÄUFIGSTE ANFÄNGERFEHLER

Bei Abbildungen und Tabellen kommt eine Regel dazu, an der fast jeder einmal scheitert: *`\label` muss hinter `\caption` stehen.*

```
\begin{figure}
  \centering
  \includegraphics[width=0.7\textwidth]{auslastung}
  \caption{Auslastung der Maschinen ueber den Planungshorizont}
  \label{abb:auslastung}
\end{figure}
```

Listing 2.15: Richtig: label nach caption

Der Grund liegt in der Arbeitsweise der Befehle. `\label` merkt sich nicht, wo es steht, sondern was zuletzt gezählt wurde. In einer `figure`-Umgebung ist es `\caption`, das die Abbildungsnummer weiterzählt. Steht `\label` davor, gibt es noch keine neue Nummer – die Marke greift dann die letzte Nummer davor ab, oft die der vorigen Abbildung oder die Nummer des Abschnitts. Das Tückische: Es gibt keine Fehlermeldung. Das Dokument übersetzt klaglos, im Text steht nur die falsche Zahl. Wer statt „Abbildung 4.3“ plötzlich „Abbildung 4.2“ oder „Abbildung 4“ liest, hat genau dieses Problem.

Dieselbe Regel gilt für Tabellen. Bei Überschriften und Gleichungen ist die Lage entspannter, weil `\chapter` und `equation` selbst zählen; man gewöhnt sich das `\label` trotzdem am besten grundsätzlich an die Stelle direkt hinter dem an, was gezählt wird. Kapitel 5 und Kapitel 4 zeigen das im Zusammenhang.

2.5.2 PRÄFIXE

Den Namen einer Marke dürfen Sie frei wählen – LaTeX schreibt ihn nicht vor. Trotzdem sollten Sie die Namen nicht wahllos vergeben, sondern sich an eine feste Regel halten. Bewährt hat sich ein Präfix, das die Art des Ziels nennt:

- `kap:` für Kapitel
- `sec:` für Abschnitte
- `abb:` für Abbildungen
- `tab:` für Tabellen
- `gl:` für Gleichungen
- `lst:` für Quelltexte

Der Nutzen ist doppelt. Erstens sieht man an `\ref{abb:auslastung}`, dass eine Abbildung gemeint ist, und schreibt das passende Wort davor – der falsche Verweis „siehe Tabelle `\ref{abb:auslastung}`“ entsteht gar nicht erst. Zweitens dürfen eine Abbildung und der Abschnitt, in dem sie steht, denselben Namen tragen, ohne zu kollidieren: `abb:auslastung` und `sec:auslastung` sind zwei verschiedene Marken. Diese Anleitung hält sich daran, die Beispielthesis auch.

2.5.3 VERWEISE AUF FORMELN

Gleichungen verweist man nicht mit `\ref`, sondern mit `\eqref`:

```
Die Zielfunktion \eqref{gl:makespan} minimiert die Gesamtdauer.
```

Listing 2.16: Verweis auf eine Gleichung

Der Unterschied ist klein und lohnt sich trotzdem: `\ref` liefert „3“, `\eqref` liefert „(3)“ – mit den Klammern, in denen die Nummer auch an der Gleichung selbst steht. So schreibt man „Gleichung (3)“ statt „Gleichung 3“, und beides passt zusammen. `\eqref` stammt aus amsmath, das die Klasse ohnehin lädt; Kapitel 6 vertieft das.

2.5.4 WARUM ZWEI LÄUFE

Der Grund steht in Abschnitt 1.4, sei hier aber kurz wiederholt, weil er die Verweise unmittelbar betrifft: LaTeX liest die Datei einmal von vorn nach hinten. Ein Verweis auf Seite 3 auf ein Kapitel, das auf Seite 57 beginnt, kann im ersten Lauf nicht aufgelöst werden – die

Nummer gibt es noch nicht. Der erste Lauf schreibt deshalb alle Marken in die Hilfsdatei `.aux`, der zweite liest sie und setzt die Nummern ein.

Praktisch heißt das: *Nach jeder Änderung an Marken oder Verweisen zweimal übersetzen.* Wer nur einmal übersetzt, sieht ein veraltetes Ergebnis – und wundert sich über einen Verweis, den er gerade korrigiert hat.

2.5.5 WENN EIN VERWEIS NICHT AUFGEHT

Ein Verweis auf eine Marke, die LaTeX nicht kennt, erscheint im PDF als zwei Fragezeichen:

> Die Auswertung in Kapitel **??** zeigt, dass …

Im Protokoll steht dazu:

```
LaTeX Warning: Reference `kap:ergebnise' on page 3 undefined on input line 42.
LaTeX Warning: There were undefined references.
```

Listing 2.17: Die Warnung im Protokoll

Dafür gibt es genau drei Ursachen, und in dieser Reihenfolge prüft man sie:

1. Es fehlt ein Lauf. Das ist der Normalfall beim ersten Übersetzen eines neuen Verweises – einfach noch einmal übersetzen.
2. Der Name ist vertippt. Im Beispiel oben steht `kap:ergebnise` statt `kap:ergebnisse`. LaTeX vergleicht Zeichen für Zeichen und kennt keine Nachsicht.
3. Das `\label` fehlt ganz, oder das Kapitel, in dem es steht, ist gerade auskommentiert.

Nehmen Sie die Warnung ernst und geben Sie keine Arbeit ab, in der noch Fragezeichen stehen. Sie sind im fertigen PDF leicht zu übersehen und im Druck peinlich. Ein Blick ins Protokoll vor der Abgabe kostet eine Minute.

2.6 LEERZEICHEN, ABSÄTZE, SONDERZEICHEN

Hier liegen die Regeln, die von Word am weitesten wegführen – und deshalb die, die man sich am bewusstesten aneignen muss.

2.6.1 LEERZEICHEN UND ZEILENUMBRÜCHE

LaTeX liest den Weißraum der Quelldatei nicht wörtlich, sondern normalisiert ihn:

- Mehrere Leerzeichen hintereinander gelten als *ein* Leerzeichen.
- Ein Zeilenumbruch in der Quelldatei gilt ebenfalls als Leerzeichen.
- Leerzeichen am Zeilenanfang werden ignoriert.

Die folgenden drei Fassungen ergeben deshalb im PDF exakt dasselbe:

```
Der Datensatz umfasst zwoelf Monate.

Der   Datensatz    umfasst     zwoelf Monate.

Der Datensatz
umfasst zwoelf
Monate.
```

Listing 2.18: Drei Schreibweisen, ein Ergebnis

Das ist keine Schlamperei, sondern ein Dienst: Wo die Zeile im gesetzten Text umbricht, entscheidet LaTeX anhand der Zeilenbreite, der Silbentrennung und des Blocksatzes – nicht anhand der Zeilen in Ihrer Quelldatei. Nutzen Sie die Freiheit und halten Sie die Quellzeilen kurz, etwa 70 Zeichen. Der Quelltext bleibt lesbar, und eine Versionsverwaltung zeigt bei einer Änderung die geänderte Zeile statt eines ganzen Absatzes.

2.6.2 ABSÄTZE

Ein neuer *Absatz* entsteht durch eine *Leerzeile*. Das ist der einzige Weg:

```
Der Datensatz umfasst zwoelf Monate der Fertigung.

Die Vorverarbeitung entfernt zunaechst die Ausreisser.
```

Listing 2.19: Zwei Absaetze

Mehrere Leerzeilen wirken wie eine; man kann also großzügig Luft in den Quelltext lassen. Wo ein Absatz beginnt, rückt LaTeX ein oder setzt einen Abstand – was davon, entscheidet die Klasse.

Und nun der Fehler, der es in fast jede erste LaTeX-Arbeit schafft: *Ein Absatz wird nicht mit \\ gemacht.* `\\` erzeugt einen *Zeilenumbruch*, und das ist etwas völlig anderes. Es bricht die Zeile ab und schreibt in der nächsten weiter – ohne Einzug, ohne Absatzabstand, und der Blocksatz der abgebrochenen Zeile wird obendrein zerrissen. Das Ergebnis sieht aus wie ein Absatz, der schiefgegangen ist, und ist es auch.

```
Der Datensatz umfasst zwoelf Monate. \\
Die Vorverarbeitung entfernt die Ausreisser.   % falsch

Der Datensatz umfasst zwoelf Monate.

Die Vorverarbeitung entfernt die Ausreisser.   % richtig
```

Listing 2.20: Falsch und richtig

Wer aus Word kommt, hat die Eingabetaste jahrelang für beides benutzt und muss sich hier umgewöhnen. Die Faustregel: `\\` gehört in eine Tabelle, in eine mehrzeilige Formel, in eine Adresse – also überall dorthin, wo eine Zeile wirklich eine Zeile ist. Im Fließtext einer Abschlussarbeit hat es nichts zu suchen. Wenn Ihnen der Abstand zwischen zwei Absätzen nicht gefällt, ist das eine Frage an die Klasse und nicht an `\\`.

2.6.3 GESCHÜTZTES LEERZEICHEN

Die Tilde ~ setzt ein *geschütztes Leerzeichen*: ein Leerzeichen, an dem die Zeile nicht umbrechen darf. Man braucht sie überall dort, wo zwei Dinge zusammengehören und ein Umbruch dazwischen den Leser stolpern ließe:

```
Abbildung~\ref{abb:auslastung} zeigt die Auslastung.
Prof.~Dr.~Erika Mustermann betreut die Arbeit.
Der Wert liegt bei 42~Prozent.
Siehe Kapitel~3, Seite~57.
```

Listing 2.21: Wo die Tilde hingehoert

Ohne die Tilde kann „Abbildung“ am Zeilenende stehen und die „4.3“ allein am Anfang der nächsten – unschön und unnötig, denn die Tilde kostet ein Zeichen. Gewöhnen Sie sie sich gleich mit an: vor jeder Verweisnummer, zwischen abgekürztem Titel und Namen, zwischen Zahl und Einheit.

2.6.4 RESERVIERTE ZEICHEN

Zehn Zeichen haben in LaTeX eine eigene Bedeutung und können deshalb nicht einfach so getippt werden. Diese *Sonderzeichen* und ihre Ersatzschreibweise:

Zeichen	Bedeutung in LaTeX	So schreibt man es
%	Kommentar	\%
$	Mathematikmodus	\$
&	Spaltentrenner	\&
#	Makro-Argument	\#
_	Tiefstellung	_
{ }	Gruppe	\{ \}
~	geschütztes Leerzeichen	\textasciitilde
^	Hochstellung	\textasciicircum
\	Befehlsanfang	\textbackslash

Die ersten sechs bekommen einfach einen Backslash vorangestellt. Die letzten drei nicht: \\ ist bereits als Zeilenumbruch vergeben, und \~ und \^ sind Akzentbefehle. Deshalb die ausgeschriebenen Namen.

2.6.5 DAS PROZENTZEICHEN

Ein Zeichen verdient eine eigene Warnung. Das *Prozentzeichen* leitet einen *Kommentar* ein: LaTeX überliest alles vom % bis zum Zeilenende, samt dem Zeilenumbruch. Wer im Text schreibt

```
Das Verfahren ist in 50 % der Fälle schneller.
```

Das Verfahren ist in 50

Listing 2.22: Der Klassiker

bekommt im PDF: „Das Verfahren ist in 50“. Der Rest des Satzes ist weg – ohne Fehlermeldung, ohne Warnung, denn aus Sicht von LaTeX ist alles in Ordnung: Sie haben einen Kommentar geschrieben. Richtig ist:

```
Das Verfahren ist in 50\,\% der Fälle schneller.
```

Das Verfahren ist in 50 % der Fälle schneller.

Listing 2.23: So ist es richtig

Das `\,` dazwischen ist ein schmales Leerzeichen – zwischen Zahl und Prozentzeichen gehört im deutschen Satz ein schmaler Abstand, kein voller und schon gar keiner. Für Zahlen mit Einheiten gibt es dafür siunitx; Kapitel 6 zeigt es.

Ansonsten ist der Kommentar ein nützliches Werkzeug. Man notiert damit Erinnerungen an sich selbst, erklärt eine unübersichtliche Tabelle oder schaltet einen Absatz vorübergehend ab, ohne ihn zu löschen:

```
% Diese Zahl noch mit dem Betreuer klären.
Der Ertrag steigt um 12 Prozent.

% Der folgende Absatz kommt vielleicht wieder rein:
% Ein alternativer Ansatz wäre ein genetischer Algorithmus.
```

Der Ertrag steigt um 12 Prozent.

Listing 2.24: Kommentare

2.6.6 ANFÜHRUNGSZEICHEN

Das Zeichen `"` auf der Tastatur ist kein *Anführungszeichen*, sondern ein Zollzeichen. Typografisch richtige Anführungszeichen sehen anders aus, und wie sie aussehen, hängt von der Sprache ab: „so“ im Deutschen, anders im Englischen, wieder anders im Französischen.

Die Klasse lädt deshalb csquotes. Man schreibt:

```
Der Betreuer nannte das Ergebnis \enquote{überraschend robust}.
```

Der Betreuer nannte das Ergebnis „überraschend robust“.

Listing 2.25: Anführungszeichen

`\enquote` setzt die Anführungszeichen, die zur eingestellten Sprache passen – bei `german` also die deutschen, bei `english` die englischen. Wer die Sprachoption der Arbeit umstellt, muss nichts anfassen. Auch das Verschachteln stimmt von selbst: Ein `\enquote` in einem `\enquote` liefert die halben Anführungszeichen, „so ‚hier' zum Beispiel". Dasselbe Muster wie bei `\emph` – man sagt, was gemeint ist, und das Paket kennt die Regeln.

2.7 GRUPPEN UND DER GELTUNGSBEREICH

Ein letzter Baustein, den man kennen muss, um Fehler zu verstehen, die sonst rätselhaft bleiben.

Geschweifte Klammern bilden in LaTeX nicht nur Argumente, sondern auch eine *Gruppe*. Eine Gruppe ist ein abgegrenzter Bereich: Alles, was darin eingestellt wird, gilt nur darin und ist danach wieder wie vorher. Das nennt man den *Geltungsbereich* einer Einstellung.

```
Der Text ist normal, {\large hier größer}, und danach wieder normal.
```

Der Text ist normal, hier größer, und danach wieder normal.

Listing 2.26: Eine Gruppe begrenzt die Wirkung

`\large` schaltet auf eine größere Schrift und hat kein Argument – es wirkt einfach ab der Stelle, an der es steht. Die schließende Klammer beendet die Gruppe und damit die Wirkung.

Umgebungen bilden ebenfalls eine Gruppe. Was zwischen `\begin` und `\end` eingestellt wird, gilt nur dort. Deshalb kann man in einer `quote`-Umgebung die Schrift umstellen, ohne dass der Rest der Arbeit betroffen ist – und deshalb ist eine Umgebung oft die sauberere Wahl als eine Klammer über zwanzig Zeilen.

Warum das wichtig ist, zeigt der Gegenfall:

```
\bfseries Diese Überschrift soll fett sein.

Und dieser Absatz auch? Und der nächste? Und das ganze Kapitel?
```

Diese Überschrift soll fett sein.
Und dieser Absatz auch? Und der nächste? Und das ganze Kapitel?

Listing 2.27: Eine vergessene Gruppe

`\bfseries` schaltet auf Fettschrift – und schaltet nicht wieder zurück, weil es niemand darum bittet. Alles ab dieser Stelle bis zum Ende der umgebenden Gruppe ist fett. Steht die Zeile im Fließtext, ist die umgebende Gruppe das `document` – und der Rest der Arbeit ist fett. Der Fehler ist nicht `\bfseries`; der Fehler ist die fehlende Gruppe:

```
{\bfseries Nur diese Zeile ist fett.}
```

Nur diese Zeile ist fett.

Listing 2.28: Mit Gruppe

Dieselbe Falle stellen `\itshape`, `\ttfamily`, `\color` und alle anderen Schalter dieser Art. Die Regel: Befehle mit Argument (`\textbf{…}`) bringen ihre Gruppe mit, Schalter (`\bfseries`) nicht. Nehmen Sie im Zweifel die Form mit Argument. Und in einer Abschlussarbeit brauchen Sie beides ohnehin kaum – das Layout ist Sache der Klasse (Abschnitt 2.3).

2.8 WENN ETWAS SCHIEFGEHT

LaTeX meldet Fehler auf eine Weise, die niemand als freundlich bezeichnen würde. Ein paar Regeln machen den Umgang damit erträglich.

2.8.1 DIE ERSTE MELDUNG ZÄHLT

LaTeX bricht bei einem Fehler nicht ab, sondern versucht weiterzumachen. Es rät, was gemeint gewesen sein könnte, und arbeitet mit dieser Annahme weiter – was oft in den nächsten Fehler führt, und in den übernächsten. Am Ende stehen dreißig Meldungen im Protokoll, von denen neunundzwanzig Folgeschäden der ersten sind.

Deshalb: *Immer die erste Fehlermeldung lesen und nur die.* Fehler beheben, neu übersetzen, wieder die erste lesen. Wer die letzte Meldung zu verstehen versucht, sucht an einer Stelle, an der nie ein Fehler war. Das ist mit Abstand der nützlichste Rat dieses Kapitels.

2.8.2 DIE ZEILENNUMMER

Eine Fehlermeldung nennt die Zeile, in der LaTeX gestolpert ist:

```
! Undefined control sequence.
l.42 Das Ergebnis ist \textbdf
                              {robust}.
```

Listing 2.29: Eine typische Fehlermeldung

Das `l.42` heißt „line 42“. Die Zeile ist an der Stelle umgebrochen, an der LaTeX nicht weiterwusste: Alles vor dem Umbruch hat es noch gelesen, alles danach nicht mehr. Damit steht die Fehlerstelle ziemlich genau da, wo der Text abbricht – hier bei `\textbdf`.

Zwei Einschränkungen. Erstens ist die genannte Zeile die, in der der Fehler *aufgefallen* ist, nicht immer die, in der er *gemacht* wurde: Ein fehlendes `\end{itemize}` in Zeile 40 fällt vielleicht erst in Zeile 120 auf. Zweitens zählt die Nummer in der Datei, die den Fehler enthält – bei einer Arbeit aus mehreren Kapiteldateien also nicht unbedingt in der Hauptdatei. Welche Datei gemeint ist, steht weiter oben im Protokoll.

2.8.3 Zwei Meldungen, die jeder kennenlernt

`! Undefined control sequence.` LaTeX kennt den Befehl nicht. Fast immer ein Tippfehler – `\textbdf` statt `\textbf`, oder ein Befehl in falscher Groß- und Kleinschreibung. Die zweithäufigste Ursache: Der Befehl ist richtig geschrieben, aber das Paket, das ihn bereitstellt, ist nicht geladen. Wer `\includegraphics` ohne graphicx benutzt, bekommt genau diese Meldung.

`! Missing $ inserted.` Etwas, das nur in Formeln erlaubt ist, steht im normalen Text. Die Hauptverdächtigen sind _ und ^: Ein Dateiname wie `daten_roh.csv`, direkt so getippt, löst sie zuverlässig aus, denn _ bedeutet „tiefstellen“ und Tiefstellung gibt es nur in der Mathematik. Richtig ist `daten\_roh.csv` – oder besser `\texttt{daten\_roh.csv}`, weil ein Dateiname ohnehin in Schreibmaschinenschrift gehört.

2.8.4 Und wenn das nicht reicht

Diese Abschnitte reichen für die Fehler der ersten Wochen. Kapitel 11 behandelt die Fehlersuche im Einzelnen: wie man das Protokoll liest, was die Warnungen über zu volle Zeilen bedeuten, wie man einen Fehler durch Halbieren des Dokuments einkreist und was zu tun ist, wenn eine Hilfsdatei veraltet ist und LaTeX Dinge meldet, die längst korrigiert sind.

Zwei Bücher lohnen sich, wenn Sie mehr wissen wollen, als eine Anleitung leisten kann: Lamports Handbuch (Lamport, 1994) ist knapp und stammt vom Erfinder; der LaTeX Companion (Mittelbach & Fischer, 2023) ist das Nachschlagewerk, in dem alles steht. Wer wissen will, was unter LaTeX eigentlich arbeitet, findet es in Knuths TeXbook (Knuth, 1986) – nötig ist das für eine Abschlussarbeit nicht.

Damit sind die Grundlagen beisammen. Kapitel 3 setzt sie zu einer vollständigen Arbeit zusammen.

3 Aufbau einer Arbeit

Eine Abschlussarbeit hat einen festen äußeren Aufbau: Titelseite, Verzeichnisse, Hauptteil, Literaturverzeichnis, Anhang, Erklärung zur wissenschaftlichen Arbeit. Die Klasse nxtthesis nimmt einem davon so viel ab, wie sich sinnvoll abnehmen lässt.

Die Optionen in eckigen Klammern hinter `\documentclass` steuern Sprache, Schrift und Layout der ganzen Arbeit. Sie stehen in der ersten Zeile und werden deshalb hier zuerst behandelt.

3.1 Die Klassenoptionen

Die Optionen stehen in eckigen Klammern vor dem Klassennamen, durch Kommata getrennt:

```
\documentclass[german,stix,richtlinie]{nxtthesis}
```

Listing 3.1: Klassenoptionen angeben

Hier sind es drei: die Sprache `german`, die Schrift `stix` und das Layout `richtlinie`. Die Optionen lassen sich frei kombinieren, sinnvollerweise eine Sprache, eine Schrift und gegebenenfalls `richtlinie`. Die folgenden Abschnitte gehen sie durch.

3.1.1 Sprache

`german` setzt Deutsch als Dokumentsprache, `english` setzt Englisch. Die Sprache bestimmt die automatisch erzeugten Überschriften („Inhaltsverzeichnis“ gegenüber „Contents“), die Namen des Abkürzungs- und Symbolverzeichnisses, die Texte der Titelseite („Vorgelegt von“ gegenüber „Submitted by“) und die Trennregeln. Ohne Angabe gilt Deutsch.

Eine Ausnahme ist die Erklärung zur wissenschaftlichen Arbeit (Abschnitt 3.6): Ihren Wortlaut gibt die Richtlinie nur auf Deutsch vor, deshalb bleibt sie auch in einer englischen Arbeit deutsch.

3.1.2 Schrift

Zwölf Optionen stellen je eine abgestimmte Kombination aus Brotschrift, serifenloser Schrift und Mathematikschrift ein. Die Mathematikschrift ist dabei das Entscheidende: Sie muss zur Textschrift passen, weil in einer technischen Arbeit auf jeder zweiten Seite eine Formel mitten im Satz steht.

Bringt eine Schrift einen Semibold-Schnitt mit, wird dieser statt des fetten verwendet – ein fetter Schnitt fällt gegenüber dem Grundtext oft zu schwer aus. Eine Besonderheit hat `times`: Times New Roman bringt keine echten Kapitälchen mit. Die Klasse setzt Abkürzungen dann

Option	Brotschrift	Serifenlos	Mathematik
`bookman`	TeX Gyre Bonum	TeX Gyre Heros	Bonum Math
`charter`	XCharter	Source Sans 3	XCharter Math
`concrete`	CMU Concrete	CMU Bright	Euler Math
`fira`	Fira Sans	Fira Sans	Fira Math
`garamond`	EB Garamond	Source Sans 3	Garamond Math
`gyre`	TeX Gyre Pagella	TeX Gyre Heros	Pagella Math
`libertinus`	Libertinus Serif	Source Sans 3	Libertinus Math
`newtx`	TeX Gyre TermesX	TeX Gyre Heros	Termes Math
`segoeui`	Segoe UI	Segoe UI	Euler Math
`stix`	STIX Two Text	Source Sans 3	STIX Two Math
`times`	Times New Roman	Arial	Termes Math
`utopia`	Erewhon	Source Sans 3	Erewhon Math

Tabelle 3.1: Die Schriftoptionen der Klasse nxtthesis und die Schriften, die sie einstellen.

im normalen Schnitt, statt sie aus Versalien herunterzurechnen. Wer Times möchte und Kapitälchen braucht, nimmt `newtx`.

Ohne Schriftoption stellt die Klasse keine Schrift ein; LuaLaTeX setzt dann in Latin Modern. Das funktioniert, sieht aber nach Standard-LaTeX aus, nicht nach NXT.

Die Entscheidung für eine Schrift trifft man am besten am Beispiel. Anhang A zeigt für jede Option dieselben drei Seiten: die Titelseite, eine Seite mit Grundtext, Kapitälchen, Mathematik und einer Tabelle sowie eine dicht gesetzte Textseite. Nebeneinander betrachtet fällt die Wahl leicht – und die dritte Seite ist die wichtigste, weil sich eine Schrift erst über einen ganzen Absatz hinweg beurteilen lässt.

3.1.3 `logo`

`logo` setzt das Logo der Fakultät mit dem Zusatz „Reutlingen University“ zentriert über den Titel der Titelseite. Ohne die Option bleibt die Titelseite ohne Logo. Mit `richtlinie` steht es ohnehin dort, weil die Richtlinie der Fakultät das Hochschullogo vorschreibt; die Option ist dann wirkungslos. Das Logo kommt aus dem NXT Logo Font, der dem Paket beiliegt (Abschnitt 11.2).

3.1.4 `kapitaelchen`

`kapitaelchen` setzt Titel und Überschriften in *Kapitälchen* statt fett. Kapitälchen sind Großbuchstaben auf der Höhe der Kleinbuchstaben – so sehen sie aus. Der Befehl dafür ist `\textsc`:

```
Ein Wort in \textsc{Kapitälchen}
sieht ruhiger aus als in
\textbf{Fettdruck}.
```

Ein Wort in Kapitälchen sieht ruhiger aus als in **Fettdruck**.

Listing 3.2: Kapitälchen gegen Fettdruck

Das wirkt ruhiger als Fettdruck, weil Kapitälchen den Grauwert der Seite kaum stören; diese Anleitung ist so gesetzt. Bringt die gewählte Schrift keine echten Kapitälchen mit – das betrifft nur `times` –, werden die Überschriften stattdessen serifenlos gesetzt; Titel und Abkürzungen bleiben in der Serifenschrift, und die Klasse gibt eine Warnung aus.

Auch eine Schrift mit Kapitälchen hat sie nicht in jedem Schnitt; Latin Modern etwa kennt keine fetten. Steht `\textsc{nasa}` in einer fetten Überschrift, erscheint dort schlicht „nasa“ in Kleinbuchstaben. `\kapitaelchen{nasa}` prüft das und setzt in diesem Fall Großbuchstaben. Für Namen, die auch in Überschriften vorkommen, ist es deshalb die bessere Wahl.

3.1.5 `richtlinie`

Die Fakultät gibt für Abschlussarbeiten ein Layout vor. Die Option `richtlinie` schaltet es ein. Sie bewirkt im Einzelnen:

- Schriftgröße 11 pt, Zeilenabstand 1,5-zeilig;
- Flattersatz statt Blocksatz;
- Seitenränder links 3 cm, sonst 2,5 cm;
- Überschriften in normaler Größe und fett, in der Brotschrift;
- Beschriftungen von Tabellen und Abbildungen in 10 pt fett, zentriert, einzeilig; die Fußnoten in 10 pt einzeilig;
- fortlaufende Nummerierung von Tabellen, Abbildungen und Formeln über die ganze Arbeit statt kapitelweise;
- Seitenzahl außen in der Fußzeile, auch auf Kapitelanfangsseiten.

Ohne die Option setzt die Klasse die „schöne“ Variante: Blocksatz, Überschriften in der Serifenschrift, kapitelweise Nummerierung. Diese Anleitung und die Beispielthesis sind so gesetzt.

Der Rat: Schreiben Sie die Arbeit ohne `richtlinie` – das liest sich beim Korrekturlesen angenehmer – und schalten Sie die Option für die Abgabefassung ein. Da nur eine Option in der ersten Zeile umzustellen ist, kostet das nichts. Prüfen Sie danach aber, ob Tabellen und Abbildungen noch passen: Der linke Rand von 3 cm macht den Textblock schmaler.

Befehl	Inhalt	Vorbelegung
`\title`	Titel der Arbeit	–
`\subtitle`	Untertitel (optional)	leer
`\author`	Verfasserin, Verfasser	–
`\adresse`	Anschrift	leer
`\matrikelnummer`	Matrikelnummer	leer
`\akademischergrad`	angestrebter Grad	leer
`\studiengang`	Studiengang	leer
`\abgabedatum`	Datum der Abgabe	leer
`\ort`	Ort	Reutlingen
`\erstpruef`	Erstprüfende(r)	leer
`\erstpruefbezeichnung`	Bezeichnung dazu	Erstprüfer
`\zweitpruef`	Zweitprüfende(r)	leer
`\zweitpruefbezeichnung`	Bezeichnung dazu	Zweitprüfer
`\fakultaet`	Fakultät	NXT Nachhaltigkeit und Technologie
`\hochschule`	Hochschule	Hochschule Reutlingen

Tabelle 3.2: Die Felder der Titelseite und ihre Vorbelegung.

3.2 DIE TITELSEITE

Die Titelseite wird nicht geschrieben, sondern ausgefüllt. Man teilt der Klasse die einzelnen Angaben mit, und `\maketitle` setzt daraus die Seite. Der Vorteil: Das Layout ist überall gleich und entspricht den Vorgaben, ohne dass man Abstände von Hand einstellt.

Alle Angaben stehen in der Präambel, also vor `\begin{document}`:

```
\title{Optimierung des Mitarbeitereinsatzes in der Lohnfertigung}
\subtitle{Ein Vergleich exakter und heuristischer Verfahren}
\author{Jean Dujardin}
\adresse{Wegstraße 123, 72764 Reutlingen}
\matrikelnummer{1234567}
\akademischergrad{Bachelor of Science}
\studiengang{Nachhaltige Technologie}
\abgabedatum{24. Dezember 2026}
\ort{Reutlingen}
\erstpruef{Prof.~Dr.~Katharina Erstmal}
\erstpruefbezeichnung{Erstprüferin}
\zweitpruef{Prof.~Dr.~Kornelius Hurtig}
\zweitpruefbezeichnung{Zweitprüfer}
```

Listing 3.3: Die Angaben für die Titelseite

Die beiden Bezeichnungsfelder gibt es, damit die Titelseite die richtige Form annimmt – „Erstprüferin“ statt „Erstprüfer“, oder „Betreuer“ statt „Prüfer“, wenn die Arbeit so geführt wird. Sie sind mit den männlichen Formen vorbelegt, weil eine Vorbelegung nötig ist; ändern

Sie sie.

`\fakultaet` und `\hochschule` sind bereits richtig vorbelegt. Man muss sie nur anfassen, wenn die Klasse an einer anderen Einrichtung eingesetzt wird – dafür sind sie da.

Die übrigen Felder sind leer vorbelegt. Das ist Absicht: Ein vergessenes Feld lässt die Titelseite an dieser Stelle einfach leer, statt den Lauf mit einer Fehlermeldung abzubrechen, die nicht verrät, welche Angabe fehlt. Prüfen Sie die Titelseite deshalb nach dem ersten Lauf mit den Augen – LaTeX beschwert sich nicht über eine fehlende Matrikelnummer.

`\maketitle` setzt die Seite und schaltet zugleich auf römische Seitenzahlen um, beginnend bei ii. Die Titelseite selbst trägt keine Nummer, zählt aber als Seite i. So verlangt es die Richtlinie.

Wie die Seite aussieht, hängt von der Option `richtlinie` ab. Ohne sie steht der Titel groß über einer Seite mit Angaben zu Grad, Fakultät, Hochschule und Studiengang. Mit ihr gilt die Vorgabe der Fakultät: Logo, Titel in 14 pt fett, alles Übrige in 12 pt.

3.3 Vorspann und Hauptteil

Zwischen Titelseite und erstem Kapitel liegt der Vorspann: Kurzfassung, Inhaltsverzeichnis, Tabellen-, Abbildungs-, Abkürzungs- und Symbolverzeichnis. Er trägt römische Seitenzahlen. Der Hauptteil beginnt danach mit arabischer 1.

Diese Umschaltung besorgt `\hauptteil`:

```
\begin{document}
\maketitle              % ab hier roemische Seitenzahlen

\chapter*{Kurzfassung}
\addcontentsline{toc}{chapter}{Kurzfassung}
Die Arbeitsvorbereitung der ...

\tableofcontents
\listoftables
\listoffigures
\printacronyms

\hauptteil              % ab hier arabische Seitenzahlen, beginnend bei 1

\input{Kapitel/01-einleitung}
```

Listing 3.4: Der Übergang vom Vorspann zum Hauptteil

Der Befehl beginnt eine neue Seite, schaltet die Nummerierung auf arabische Ziffern um und setzt den Zähler auf 1. Er ist genau einmal aufzurufen, unmittelbar vor dem ersten Kapitel.

Die Kurzfassung ist mit `\chapter*` als unnummeriertes Kapitel gesetzt – sie bekommt keine Kapitelnummer. Der Stern hat aber die Nebenwirkung, dass der Eintrag im Inhaltsverzeichnis fehlt; deshalb die Zeile mit `\addcontentsline`. Dasselbe Muster gilt für jedes unnummerierte Kapitel, das trotzdem im Verzeichnis stehen soll.

3.4 Kapitel auf Dateien verteilen

Eine Abschlussarbeit in einer einzigen Datei wird schnell unhandlich. Die Lösung ist `\input`: Die Hauptdatei enthält nur Präambel und Gerüst, jedes Kapitel steht in einer eigenen Datei.

```
\input{Kapitel/01-einleitung}
\input{Kapitel/02-grundlagen}
\input{Kapitel/03-modell}
```

Listing 3.5: Ein Kapitel je Datei

`\input` fügt die Datei an dieser Stelle ein, als stünde ihr Inhalt direkt dort. Die Endung `.tex` kann entfallen. Die Kapiteldatei selbst enthält kein `\documentclass` und kein `document`-Umgebung, sondern beginnt schlicht mit `\chapter`.

Das hat mehrere Vorteile. Man findet sich wieder; die Versionsverwaltung zeigt sinnvolle Änderungen statt eines Riesen-Diffs; und beim Umstellen der Gliederung verschiebt man eine Zeile statt tausend. Die Nummerierung der Dateinamen (`01-`, `02-`) sorgt nebenbei dafür, dass die Dateiliste im Editor in der Reihenfolge der Arbeit steht.

Diese Anleitung selbst ist so gebaut, ebenso die Beispielthesis. Ein Blick in beide Hauptdateien zeigt das Muster.

3.5 Gliederung

Die Klasse beruht auf der Standardklasse report und kennt deshalb die übliche Hierarchie:

```
\chapter{Analyse der Ergebnisse}      % Kapitel, beginnt neue Seite
\section{Die gefundene Zuordnung}     % Abschnitt
\subsection{Auslastung}               % Unterabschnitt
\subsubsection{Einzelfaelle}          % Unterunterabschnitt
\paragraph{Ein Sonderfall}            % Absatz mit Ueberschrift
```

Listing 3.6: Die Gliederungsbefehle

Mehr als drei Ebenen braucht eine Abschlussarbeit selten. Wer `\subsubsection` regelmäßig benutzt, sollte prüfen, ob die Gliederung nicht zu fein geraten ist – eine Ebene, die im Inhaltsverzeichnis nicht mehr erscheint, hilft der Leserin nicht.

Hinter jede Überschrift, auf die verwiesen werden soll, gehört ein `\label`. Bewährt hat sich ein Präfix nach Art des Ziels: `kap:` für Kapitel, `sec:` für Abschnitte, `tab:`, `abb:`, `gl:` für Tabellen, Abbildungen und Gleichungen. Dann sieht man schon am Verweis, worauf er zeigt.

```
\chapter{Analyse der Ergebnisse}
\label{kap:ergebnisse}
...
Die Auswertung in Kapitel~\ref{kap:ergebnisse} zeigt, dass ...
```

Listing 3.7: Label und Verweis

Das Tilde-Zeichen vor `\ref` ist ein geschütztes Leerzeichen: Es verhindert, dass zwischen „Kapitel“ und der Nummer ein Zeilenumbruch fällt. Gewöhnen Sie es sich an; es kostet nichts und sieht besser aus.

3.6 Die Erklärung zur wissenschaftlichen Arbeit

Am Ende der Arbeit steht die Erklärung zur wissenschaftlichen Arbeit. Sie ist nicht frei formulierbar: Die Richtlinie der Fakultät gibt den Wortlaut vor, und die Klasse bringt ihn mit. Zu wählen ist nur, ob und wie Sie Werkzeuge der künstlichen Intelligenz genutzt haben:

```
\eigenständigkeitserklärung[ki=keine]
\end{document}
```

Listing 3.8: Die Erklärung am Ende der Arbeit

Für `ki` gibt es drei Werte, entsprechend den drei Optionen der Richtlinie:

`keine` Sie haben keine inhaltsgenerierenden KI-Werkzeuge verwendet.

`kennzeichnung` Die prüfende Person hat die Nutzung ausdrücklich erlaubt, und die verwendeten Werkzeuge sind der Arbeit nach den Vorgaben der Fakultät angefügt. Die Beispielthesis zeigt, wie das aussehen kann.

`ohnekennzeichnung` Die Nutzung ist erlaubt, eine Kennzeichnung aber nicht verlangt.

Fehlt die Angabe, übersetzt das Dokument trotzdem, gibt aber eine Warnung aus – die Erklärung wäre sonst unvollständig.

Wird die Arbeit nach der Prüfungsordnung Ihres Studiengangs auch an einer Partnerhochschule eingereicht, verlangt die Richtlinie einen anderen Absatz. Die Klasse setzt ihn, sobald Sie die Partnerhochschule nennen:

```
\eigenständigkeitserklärung[
  ki                = kennzeichnung,
  partnerhochschule = {Université Exemple},
  partnerordnung    = {des Bachelorstudiengangs Engineering},
]
```

Listing 3.9: Erklärung bei Einreichung an einer Partnerhochschule

Der Befehl beginnt eine neue Seite mit der Überschrift „Erklärung zur wissenschaftlichen Arbeit“, setzt den Text und darunter Ort und Abgabedatum sowie eine Linie für die Unterschrift mit Ihrem Namen. Der Name des Befehls enthält Umlaute; das funktioniert unter LuaLaTeX ohne Weiteres, solange die Datei als UTF-8 gespeichert ist (Abschnitt 1.5).

Vergessen Sie nicht, das ausgedruckte Exemplar auch tatsächlich zu unterschreiben. LaTeX nimmt Ihnen viel ab, aber das nicht.

4 Tabellen

Tabellen sind die Stelle, an der in Abschlussarbeiten am meisten schiefgeht. Das liegt selten an LaTeX und meist daran, dass man versucht, das Aussehen einer Tabellenkalkulation nachzubauen. Dieses Kapitel beschreibt, wie es besser geht.

4.1 Keine senkrechten Linien

Die Klasse lädt das Paket booktabs. Es stellt drei Linien bereit: `\toprule` über der Tabelle, `\midrule` unter dem Kopf und `\bottomrule` am Ende.

```
\begin{table}[tb]
  \centering
  \begin{tabular}{llr}
    \toprule
    \textbf{Methode} & \textbf{Beschreibung} & \textbf{Aufwand}\\
    \midrule
    Interviews      & Qualitative Befragung        & hoch\\
    Fragebogen      & Quantitative Erhebung        & mittel\\
    Dokumentanalyse & Auswertung vorhandener Daten & gering\\
    \bottomrule
  \end{tabular}
  \caption[Forschungsmethoden]{Übersicht der Forschungsmethoden.}
  \label{tab:methoden}
\end{table}
```

Listing 4.1: Eine Tabelle nach den Regeln von booktabs

Das Ergebnis zeigt Tabelle 4.1.

Was fehlt, ist ebenso wichtig wie das, was da ist: keine senkrechten Linien, keine Linie zwischen den Zeilen, kein Rahmen. Das ist keine Geschmacksfrage, sondern Handwerk. Eine Linie ist ein starkes gestalterisches Mittel; sie soll etwas trennen, das ohne sie verwechselt würde. Spalten trennt aber schon der Weißraum zwischen ihnen. Ein Gitternetz fügt

Methode	Beschreibung	Aufwand
Interviews	Qualitative Befragung	hoch
Fragebogen	Quantitative Erhebung	mittel
Beobachtung	Direkte Verhaltensanalyse	hoch
Dokumentanalyse	Auswertung vorhandener Daten	gering

Tabelle 4.1: Übersicht der Forschungsmethoden. Drei waagerechte Linien, keine senkrechten – so sieht eine Tabelle nach booktabs aus.

der Tabelle deshalb nur Rauschen hinzu und zerhackt die Zeilen, die das Auge eigentlich entlanglaufen soll. Senkrechte Linien in einer Tabelle sind ein Erbe der Schreibmaschine, in der Weißraum nicht steuerbar war.

Wer meint, eine Tabelle brauche mehr Linien, hat meist eine Tabelle, die zu viel auf einmal will. Die Abhilfe ist, sie zu teilen – nicht, sie zu umzäunen. Die Befehle `\cmidrule{2-3}` für eine Linie über einzelne Spalten und `\addlinespace` für eine kleine Lücke zwischen Blöcken decken die verbleibenden Fälle ab.

Der zweite Grundsatz betrifft die Beschriftung: Sie steht *unter* der Tabelle. So gibt es die Fakultät vor. In manchen anderen Regelwerken steht die Tabellenbeschriftung oben – hier gilt die Vorgabe der Fakultät, und zwar für Tabellen wie für Abbildungen gleichermaßen.

Technisch entscheidet darüber allein die Reihenfolge im Quelltext: Die `table`-Umgebung setzt ihre Bestandteile so, wie sie geschrieben sind. `\caption` und `\label` gehören deshalb *hinter* das `\end{tabular}`. Stünden sie davor, erschiene die Beschriftung über der Tabelle. Das ist der häufigste Fehler an dieser Stelle, und LaTeX warnt nicht davor – es tut genau das, was dasteht.

`\label` steht dabei stets *nach* `\caption`, nie davor. Der Grund: `\label` merkt sich die zuletzt vergebene Nummer, und die Nummer der Tabelle vergibt erst `\caption`. Ein `\label` davor zeigt auf die Kapitelnummer – ein Fehler, der lange unbemerkt bleibt, weil der Verweis ja irgendeine Zahl liefert.

4.2 Zahlen ausrichten

Zahlen in einer Tabelle richten sich am Dezimalkomma aus. Alles andere ist unlesbar: Man vergleicht Größenordnungen mit dem Auge, und dazu müssen die Kommata untereinanderstehen. Die Richtlinie verlangt genau das.

Von Hand ist das mühsam. Das Paket siunitx (Wright, 2024) erledigt es mit dem Spaltentyp `S`:

```
\usepackage{siunitx}
\sisetup{
  output-decimal-marker = {,},   % deutsches Dezimalkomma
  group-digits          = integer,
  group-minimum-digits  = 4,     % ab 10000 wird gruppiert
  group-separator       = {\,},  % schmales Leerzeichen als Trenner
  detect-all,                    % Schrift der Umgebung uebernehmen
}

\begin{tabular}{lS[table-format=3.2]S[table-format=2.1]}
  \toprule
  {Verfahren} & {Dauer in min} & {Zeit in s}\\
  \midrule
  Greedy    & 412.50 & 0.1\\
  Exakt     &  87.25 & 12.4\\
  Heuristik &  91.05 & 48.7\\
  \bottomrule
\end{tabular}
```

Listing 4.2: Ausrichtung am Dezimalkomma mit S-Spalten

Verfahren	Dauer in min	Zeit in s
Greedy	412,50	0,1
Exakt	87,25	12,4
Heuristik	91,05	48,7

Tabelle 4.2: Zahlen in S-Spalten: Die Kommata stehen untereinander, unabhängig von der Zahl der Vor- und Nachkommastellen.

Tabelle 4.2 zeigt das Ergebnis. Drei Dinge sind bemerkenswert. Erstens: Im Quelltext steht ein Punkt, im Satz erscheint ein Komma. siunitx rechnet um; man schreibt die Zahlen also so, wie sie aus der Rechnung kommen, und die Einstellung `output-decimal-marker` entscheidet über die Ausgabe. Zweitens: `table-format=3.2` gibt an, wie viele Stellen vor und nach dem Komma zu erwarten sind; daraus berechnet siunitx die Spaltenbreite. Drittens: Text in einer S-Spalte – also auch die Kopfzeile – muss in geschweifte Klammern, sonst versucht siunitx, ihn als Zahl zu lesen.

Die Einheit gehört nicht in jede Zelle, sondern in den Spaltenkopf. „Dauer in min“ ist richtig; „412,50 min“ in dreißig Zeilen ist Wiederholung, die nur Platz kostet.

4.3 Tabellen über mehrere Seiten

Eine `table`-Umgebung ist ein Gleitobjekt: LaTeX sucht sich eine passende Stelle und schiebt sie dorthin. Auf eine Seite muss sie aber passen. Für längere Tabellen – eine vollständige Messreihe im Anhang etwa – gibt es longtable. Die Klasse lädt das Paket bereits.

```
\begin{longtable}{lS[table-format=3.2]}
  \caption{Alle Messwerte der Reihe.}\label{tab:messwerte}\\
  \toprule
  {Nummer} & {Wert}\\
  \midrule
  \endfirsthead                          % Kopf auf der ersten Seite
  \multicolumn{2}{l}{\textit{Fortsetzung von der vorigen Seite}}\\
  \toprule
  {Nummer} & {Wert}\\
  \midrule
  \endhead                               % Kopf auf allen Folgeseiten
  \midrule
  \multicolumn{2}{r}{\textit{Fortsetzung auf der naechsten Seite}}\\
  \endfoot                               % Fuss auf allen Seiten ausser der letzten
  \bottomrule
  \endlastfoot                           % Fuss auf der letzten Seite
  1 & 12.30\\
  2 & 14.05\\
  % ... viele weitere Zeilen
\end{longtable}
```

Listing 4.3: Eine Tabelle über mehrere Seiten

Eine `longtable` gleitet nicht – sie steht dort, wo sie im Quelltext steht, und bricht am Seitenende um. Die vier `\end...`-Befehle legen fest, was am Anfang und Ende jeder Teiltabelle wiederholt wird. Der Aufwand lohnt sich: Ohne wiederholten Kopf ist die Fortsetzung auf Seite 3 nicht mehr lesbar.

`\caption` und `\label` stehen bei longtable innerhalb der Umgebung und werden mit `\\` abgeschlossen.

4.4 Breite Tabellen im Querformat

Manche Tabellen sind zu breit. Bevor Sie zum Querformat greifen, prüfen Sie die einfacheren Auswege: Kürzere Spaltenköpfe, weniger Nachkommastellen (drei Stellen sind fast immer zwei zu viel), oder schlicht weniger Spalten. Oft ist eine breite Tabelle eine Tabelle, die zwei Aussagen gleichzeitig macht und besser zwei Tabellen wäre.

Hilft das nicht, dreht man die Seite. Das Paket pdflscape liefert die Umgebung `landscape`:

```
\usepackage{pdflscape}
...
\begin{landscape}
  \begin{table}
    \centering
    ... % sehr breite Tabelle
  \end{table}
\end{landscape}
```

Listing 4.4: Eine einzelne Seite im Querformat

Der Unterschied zwischen lscape und pdflscape ist klein, aber wichtig: pdflscape dreht die Seite zusätzlich im PDF, sodass sie im Betrachter aufrecht erscheint und man den Kopf nicht drehen muss. Nehmen Sie pdflscape.

Für Tabellenköpfe, die nur wegen langer Beschriftungen breit sind, gibt es einen sanfteren Weg: `\rotatebox{90}{...}` aus dem Paket rotating dreht einzelne Zellen. Die Beispielthesis macht das an einer Stelle vor.

4.5 Verweise

Jede Tabelle bekommt ein `\label` und wird im Text angesprochen. Eine Tabelle, auf die nirgends verwiesen wird, gehört nicht in die Arbeit – entweder sie trägt etwas bei, dann sagt man wozu, oder sie tut es nicht, dann kann sie weg oder in den Anhang.

```
Tabelle~\ref{tab:siunitx} zeigt, dass die Kommata untereinanderstehen.
```

Listing 4.5: Auf eine Tabelle verweisen

Beachten Sie die Doppelform von `\caption`. Das optionale Argument ist der Kurztext für das Tabellenverzeichnis, das obligatorische die volle Beschriftung unter der Tabelle:

```
\caption[Ausrichtung am Dezimalkomma]{Zahlen in S-Spalten: Die Kommata
  stehen untereinander, unabhängig von der Zahl der Vor- und
```

```
  Nachkommastellen.}
```

Listing 4.6: Kurz- und Langfassung der Beschriftung

Die Beschriftung darf und soll erklären, was zu sehen ist – das Verzeichnis soll nur benennen. Ohne die Kurzfassung stünde der ganze Erklärungstext im Tabellenverzeichnis und machte es unbrauchbar.

5 Abbildungen

Abbildungen kommen auf drei Wegen in die Arbeit: als fertige Datei, die eingebunden wird; als Zeichnung, die LaTeX selbst erzeugt; oder als Grafik, die ein anderes Programm berechnet. Dieses Kapitel zeigt alle drei und sagt, wann welcher taugt.

5.1 Fertige Dateien einbinden

Der Normalfall ist `\includegraphics` aus dem Paket graphicx. Die Klasse lädt es bereits mit – über nxtlatexlogos, das die Logo-Makros bereitstellt –, ein eigenes `\usepackage{graphicx}` schadet aber nicht und macht die Absicht deutlich.

```
\begin{figure}[tb]
  \centering
  \includegraphics[width=0.6\textwidth]{Abbildungen/auslastung.pdf}
  \caption[Auslastung je Mitarbeiter]{Auslastung der Mitarbeitenden
    nach dem Verfahren. Der Makespan ist die Höhe des höchsten Balkens.}
  \label{abb:auslastung}
\end{figure}
```

Listing 5.1: Eine Abbildung einbinden

Abbildung 5.1 zeigt das Ergebnis mit einem Platzhalter.

Wie bei Tabellen gilt: `\caption` und dahinter `\label`, beides *unter* der Abbildung. Bei Abbildungen ist die Beschriftung unten ohnehin die verbreitete Konvention; die Vorgabe der Fakultät bestätigt sie hier nur.

Zur Breite: Geben Sie sie relativ an, als Bruchteil von `\textwidth`, nicht in Zentimetern. Dann passt die Abbildung noch immer, wenn Sie am Ende auf `richtlinie` umschalten und der Textblock wegen des linken Rands von 3 cm schmaler wird. Ohne Größenangabe wird die Grafik in ihrer natürlichen Größe eingesetzt – was bei einer Datei aus einem Zeichenprogramm selten die gewünschte ist.

Zum Format: Nehmen Sie PDF, wo immer es geht. PDF ist ein Vektorformat – die Grafik besteht aus Linien und Flächen, nicht aus Bildpunkten, und bleibt beim Vergrößern scharf. Für Diagramme, Zeichnungen und Schaubilder ist das der einzig richtige Weg. Alle gängigen Programme können PDF ausgeben, auch matplotlib (`savefig("bild.pdf")`) und Excel über den Umweg „Drucken als PDF“. PNG und JPEG bleiben den Fotos und Bildschirmfotos vorbehalten, für die sie gemacht sind. Ein Bildschirmfoto eines Diagramms ist immer die schlechtere Lösung als das Diagramm selbst.

Die Dateiendung darf man weggelassen; LaTeX sucht sich dann eine passende Datei. Beim Dateinamen gilt: keine Leerzeichen, keine Umlaute. Und Pfade werden relativ zur *Hauptdatei* angegeben, nicht relativ zur Kapiteldatei, in der der Befehl steht – eine Stolperstelle, sobald man Kapitel auf Dateien verteilt.

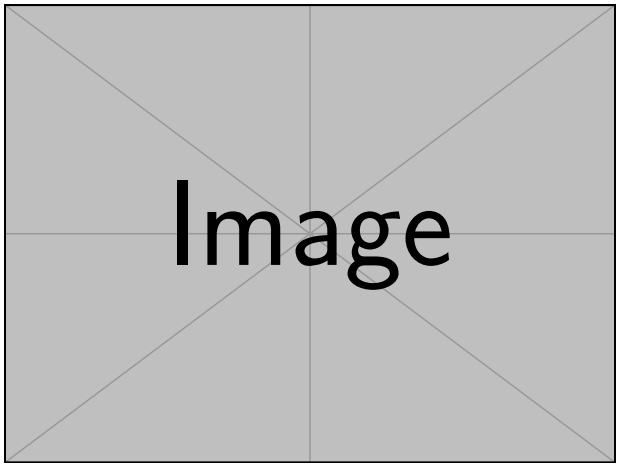

Abbildung 5.1: Eine mit `\includegraphics` eingebundene Datei. Die Beschriftung steht unter der Abbildung, das `\label` hinter der Beschriftung.

5.2 ZEICHNEN MIT TIKZ

Manche Abbildungen gibt es nicht als Datei, weil sie erst zu zeichnen sind: ein Blockschaltbild, ein Ablauf, eine Skizze des Modells. Dafür gibt es *TikZ* (Tantau, 2024). Man beschreibt die Zeichnung im Quelltext, und LaTeX setzt sie.

Das klingt umständlicher, als es ist, und hat einen entscheidenden Vorteil: Die Zeichnung benutzt die Schrift und die Mathematik des Dokuments. Beschriftungen sehen also nicht aus wie eingeklebt, sondern gehören dazu, und $C_{\max}$ in der Zeichnung ist dasselbe $C_{\max}$ wie im Text. Bei einer aus einem Zeichenprogramm exportierten Grafik ist das nie der Fall.

Zusammen mit den Farben aus nxtlatexcolors entsteht so eine Abbildung, die zum Rest der Arbeit passt:

```
\usepackage{tikz}
\usetikzlibrary{positioning}
...
\begin{tikzpicture}[
    kasten/.style={draw=nxtgreen, line width=.8pt, rounded corners=2pt,
                   fill=nxtgreen!8, minimum height=9mm, minimum width=24mm},
    pfeil/.style={-latex, draw=nxt.gray4},
  ]
  \node[kasten] (daten)   {Daten};
  \node[kasten, right=14mm of daten] (modell) {Modell};
  \node[kasten, right=14mm of modell] (loesung) {Lösung};
  \draw[pfeil] (daten)  -- (modell);
  \draw[pfeil] (modell) -- (loesung);
\end{tikzpicture}
```

Listing 5.2: Eine TikZ-Zeichnung mit den NXT-Farben

Abbildung 5.2 zeigt das Ergebnis.

Der Stil-Mechanismus (`kasten/.style={...}`) ist der Schlüssel zu lesbaren TikZ-Bildern:

Abbildung 5.2: Eine mit TikZ gezeichnete Abbildung in den Farben der Fakultät. Schrift und Mathematiksatz sind dieselben wie im Fließtext.

Das Aussehen wird einmal oben festgelegt, die Zeichnung darunter enthält nur noch, was wo steht. Wer alle Zeichenbefehle einzeln ausschmückt, bekommt Code, den er nach zwei Wochen selbst nicht mehr versteht.

Längere Zeichnungen gehören in eine eigene Datei und werden mit `\input` eingebunden – so hält man die Kapiteldatei lesbar. Die Beispielthesis macht das unter `Abbildungen/` vor.

5.3 Rechnen und Zeichnen mit MetaPost

MetaPost ist eine eigene Sprache für Grafiken, älter als TikZ und mit einer anderen Stärke: Sie rechnet. Ein Funktionsverlauf, eine Kurvenschar, eine Konstruktion, die sich aus Schnittpunkten ergibt – das schreibt sich in MetaPost natürlicher als in TikZ.

Das Paket luamplib (Hoekwater et al., 2024) macht MetaPost aus LuaLaTeX heraus verfügbar. Ein eigener `mpost`-Lauf entfällt; die Grafik wird während des `lualatex`-Laufs berechnet, und die Beschriftungen zwischen `btex` und `etex` erben Schrift und Mathematiksatz des Dokuments.

```
\usepackage{luamplib}
...
\begin{mplibcode}
  beginfig(1);
    numeric b, h; b := 200; h := 90;
    drawarrow (0,0) -- (b+10, 0) withcolor nxt_gray4;
    drawarrow (0,0) -- (0, h+10) withcolor nxt_gray4;
    % Exponentielle Abkuehlung, ueber die normierte Laufvariable
    path kurve;
    kurve = (0, h) for i = 1 upto 60:
              .. (b*i/60, h*mexp(-256*2.5*i/60))
            endfor;
    draw kurve withcolor nxtgreen withpen pencircle scaled 1.2pt;
    label.rt(btex \footnotesize $T_k = T_0\,\alpha^k$ etex, (b*0.3, h*0.5));
  endfig;
\end{mplibcode}
```

Listing 5.3: MetaPost aus LuaLaTeX heraus

Die Farben stehen hier als `nxtgreen` und `nxt_gray4` zur Verfügung, weil die Beispielthesis sie aus nxtlatexcolors.sty in eine MetaPost-Datei (`Abbildungen/nxtfarben.mp`) übersetzt und diese mit `input` einliest. Dieselbe Farbdefinition speist so LaTeX, TikZ, MetaPost und matplotlib – ein Aufwand, der sich lohnt, sobald mehr als zwei Abbildungen im Spiel sind.

MetaPost hat zwei Eigenheiten, über die jeder einmal stolpert. Ein Name darf nicht auf einer Ziffer enden: `T0` liest MetaPost als `T` mit dem Suffix `0`. Und Zahlen sind auf rund

Die Abbildung ist …	**…dann nehmen Sie**
ein Diagramm aus Messdaten	matplotlib, R, gnuplot → PDF
ein Foto, ein Bildschirmfoto	PNG oder JPEG
ein Schema, ein Ablauf, eine Skizze	TikZ
ein gerechneter Verlauf, eine Konstruktion	MetaPost über luamplib
aus einem fremden Werkzeug	als PDF exportieren

Tabelle 5.1: Anhaltspunkte für die Wahl des Werkzeugs. Die Grenzen sind fließend; entscheidend ist, dass das Ergebnis ein Vektorformat ist und die Beschriftung lesbar bleibt.

32 768 begrenzt, weshalb α^{60000} überläuft und man Verläufe über eine normierte Laufvariable rechnet. Die Datei `Abbildungen/abkuehlung.mp` der Beispielthesis erklärt beides im Kommentar.

5.4 WELCHES WERKZEUG WANN?

Tabelle 5.1 fasst die Anhaltspunkte zusammen. Die wichtigste Regel steht nicht darin: Erzeugen Sie Abbildungen aus Daten niemals von Hand. Wenn die Messreihe sich ändert – und sie ändert sich –, soll ein Programmlauf genügen, um Diagramm und Zahlen im Text richtigzustellen. Die Beispielthesis treibt das konsequent: Ihre Abbildungen *und* die Zahlen im Fließtext entstehen aus den Programmen unter `Code/`, sodass im Text keine Zahl stehen kann, die von der Rechnung abweicht. Kapitel 9 zeigt, wie das geht.

Die Beispielthesis führt alle drei Wege vor: eingebundene PDF-Dateien aus matplotlib unter `Abbildungen/`, zwei TikZ-Zeichnungen (`tikz-maschinenmodell.tex`, `tikz-nachbarschaft.tex`) und eine MetaPost-Grafik (`abkuehlung.tex`). Wer eine Vorlage sucht, findet sie dort.

5.5 GLEITOBJEKTE

Das `[tb]` hinter `\begin{figure}` ist eine Bitte an LaTeX: „setze die Abbildung oben (*top*) oder unten (*bottom*) auf eine Seite“. LaTeX entscheidet dann selbst, welche Seite. Das ist gewollt: Eine Abbildung mitten im Absatz zerreißt den Text, und eine halb leere Seite, nur damit die Abbildung an genau dieser Stelle steht, sieht schlecht aus.

Deshalb schreibt man auch nie „in der folgenden Abbildung“ oder „die Tabelle unten“, sondern immer „Abbildung 5.2 zeigt …“. Wo die Abbildung landet, weiß man vorher nicht – und nach der nächsten Textänderung ist es ohnehin anders.

Wenn am Ende alle Abbildungen im Anhang zu landen scheinen, liegt das meist daran, dass zu viele auf einmal warten. Dann hilft `\clearpage` am Kapitelende, das alle Wartenden ausgibt. Die Option `[h]` für *here* ist dagegen fast immer der falsche Weg – sie bittet LaTeX um etwas, das es aus gutem Grund nicht tun will.

6 MATHEMATIK

Formelsatz ist die Disziplin, in der LaTeX allen Alternativen überlegen ist. Dieses Kapitel beschreibt, was die Klasse dafür mitbringt und worauf man im deutschen Satz zu achten hat.

6.1 WAS DIE KLASSE LÄDT

Um Formeln muss man sich bei dieser Klasse kaum kümmern. Sie lädt amsmath – die Grundlage jedes ernsthaften Formelsatzes – von sich aus; man muss es *nicht* in der Präambel angeben. Mit einer Schriftoption (Abschnitt 3.1.2) kommt außerdem unicode-math und die passende Mathematikschrift dazu: bei `stix` etwa STIX Two Math, bei `gyre` TeX Gyre Pagella Math. Die Formeln passen so zur Brotschrift, weil beide aus derselben Familie stammen.

unicode-math erlaubt außerdem, griechische Buchstaben und mathematische Zeichen unmittelbar einzugeben, sofern der Editor sie hergibt. Man kann in einer Formel also das Zeichen α tippen, statt `\alpha` zu schreiben. Beide Eingaben – das getippte α und der Befehl `\alpha` – ergeben im Satz dasselbe α. Ob man davon Gebrauch macht, ist Geschmackssache: Die Befehlsnamen sind unmissverständlich, die getippten Zeichen kürzer.

6.2 FORMELN IM TEXT UND ABGESETZT

Eine Formel im laufenden Text steht zwischen Dollarzeichen: `$C_{\max} = 87{,}25$` ergibt $C_{\max} = 87{,}25$. Für Formeln mit Brüchen, Summen oder Integralen ist der Fließtext zu eng – sie gehören abgesetzt.

Abgesetzte Formeln ohne Nummer schreibt man zwischen `\[` und `\]`:

```
\[
  \int_{-\infty}^{+\infty} e^{-x^2}\,\mathrm{d}x = \sqrt{\pi}
\]
```

Listing 6.1: Eine abgesetzte Formel ohne Nummer

$$\int_{-\infty}^{+\infty} e^{-x^2}\,\mathrm{d}x = \sqrt{\pi}$$

Verwenden Sie *nicht* `$$...$$`. Das ist reines TeX, kein LaTeX, und führt zu falschen Abständen.

Soll die Formel eine Nummer bekommen, nimmt man `equation`:

```
\begin{equation}
  \label{gl:makespan}
  C_{\max} = \max_{i \in M} \sum_{j \in J} p_{ij}\, x_{ij}
```

```
\end{equation}
```

Listing 6.2: Eine nummerierte Formel

$$C_{\max} = \max_{i \in M} \sum_{j \in J} p_{ij}\, x_{ij} \tag{6.1}$$

Nummerieren Sie nur, worauf Sie auch verweisen. Eine Nummer ist ein Versprechen an die Leserin, dass die Formel später noch einmal gebraucht wird; wird sie es nicht, ist die Nummer Ballast. Für unnummerierte abgesetzte Formeln gibt es `\[...\]` und die Sternform `equation*`.

Wie die Nummer aussieht, entscheidet die Option `richtlinie`: Ohne sie wird kapitelweise gezählt (5.1, 5.2, …), mit ihr fortlaufend über die ganze Arbeit (1, 2, …). Dasselbe gilt für Tabellen und Abbildungen.

6.3 MEHRZEILIGE FORMELN

Für mehrere Formeln, die aneinander ausgerichtet werden sollen, gibt es `align` aus amsmath. Das Zeichen `&` markiert die Stelle, an der ausgerichtet wird – üblicherweise das Gleichheits- oder Relationszeichen:

```
\begin{align}
  \min\ & C_{\max} \label{gl:ziel}\\
  \text{u.\,d.\,N.}\quad
        & \sum_{i \in M} x_{ij} = 1 && \text{für alle } j \in J \label{gl:zuordnung}\\
        & x_{ij} \in \{0, 1\}       && \text{für alle } i \in M,\ j \in J \label{gl:
    binaer}
\end{align}
```

Listing 6.3: Mehrere ausgerichtete Formeln

$$\begin{aligned}
&\min C_{\max} && && (6.2)\\
\text{u. d. N.}\quad &\sum_{i \in M} x_{ij} = 1 && \text{für alle } j \in J && (6.3)\\
&x_{ij} \in \{0,1\} && \text{für alle } i \in M,\ j \in J && (6.4)
\end{aligned}$$

Jede Zeile bekommt eine eigene Nummer und darf ihr eigenes `\label` tragen. Ein doppeltes `&&` beginnt eine weitere Ausrichtungsspalte – hier für die Nebenbedingungen rechts. Soll eine einzelne Zeile keine Nummer bekommen, setzt man `\nonumber` ans Zeilenende; soll keine Zeile eine Nummer bekommen, nimmt man `align*`.

Wichtig ist `\text{...}` für Wörter innerhalb einer Formel – so wie das „für alle“ in der Nebenbedingung oben. Ohne `\text` setzt LaTeX jeden Buchstaben einzeln als kursive Variable und multipliziert sie: Aus `alle` würde $a\,l\,l\,e$, mit störenden Abständen zwischen den Buchstaben. Genau das verhindert `\text`.

6.4 AUF FORMELN VERWEISEN

Formelnummern spricht man mit `\eqref` an, nicht mit `\ref`. `\eqref` setzt die Klammern gleich mit:

```
Die Zielfunktion~\eqref{gl:ziel} minimiert die Gesamtdauer, während
Nebenbedingung~\eqref{gl:zuordnung} sicherstellt, dass jede Aufgabe
genau einmal vergeben wird.
```

Listing 6.4: Verweis auf eine Formel

Das liest sich dann so: Die Zielfunktion (6.2) minimiert die Gesamtdauer, während Nebenbedingung (6.3) sicherstellt, dass jede Aufgabe genau einmal vergeben wird. Mit `\ref` stünde dort eine nackte Zahl, was unüblich ist.

Auch hier gilt: `\label` nach der Formel beziehungsweise am Ende der Zeile, nie davor.

6.5 ZAHLEN IM DEUTSCHEN SATZ

Im Deutschen trennt ein *Dezimalkomma* die Nachkommastellen, kein Punkt. Das ist kein Detail: „1.500“ bedeutet im deutschen Satz eintausendfünfhundert, im englischen eineinhalb. In einer Arbeit über Messwerte ist das ein Unterschied, der zählt.

Im Mathematikmodus behandelt LaTeX das Komma als Trennzeichen und setzt eine Lücke dahinter – `$3,14$` ergibt $3,14$ mit einem zu großen Abstand. Abhilfe schafft eine Klammer: `$3{,}14$` ergibt $3{,}14$.

Das merkt sich niemand zuverlässig. Deshalb ist siunitx die bessere Lösung, das schon in Kapitel 4 die Spalten ausgerichtet hat. Es kennt die Regeln und wendet sie überall gleich an:

```
\usepackage{siunitx}
\sisetup{
  output-decimal-marker = {,},   % deutsches Dezimalkomma
  group-digits          = integer,
  group-minimum-digits  = 4,     % ab 10000 gruppieren
  group-separator       = {\,},  % schmales Leerzeichen als Trenner
  detect-all,
}
...
Das Verfahren braucht \num{12.4} Sekunden und liefert
\qty{87.25}{\minute}. Der Bedarf sinkt auf \qty{1250.5}{\kilo\watt\hour},
bei \num{1250000} Teilen im Jahr.
```

Listing 6.5: Zahlen und Einheiten mit siunitx

Gesetzt ergibt das: Das Verfahren braucht 12,4 Sekunden und liefert 87,25 min. Der Bedarf sinkt auf 1 250,5 kW h, bei 1 250 000 Teilen im Jahr.

Man schreibt im Quelltext also den Punkt, wie er aus der Rechnung kommt, und siunitx setzt das Komma. Das ist genau richtig herum: Die Zahl bleibt so, wie das Programm sie ausgegeben hat, und die Darstellung ist Sache des Satzes.

`\qty` setzt Zahl und Einheit zusammen und kümmert sich um den Abstand dazwischen – ein schmales, geschütztes Leerzeichen, wie es die Norm verlangt, sodass 87,25 min nie über den Zeilenumbruch zerrissen wird. Steht die Einheit allein, etwa im Spaltenkopf, nimmt man

`\unit{\minute}`: min. Und die Einheiten schreibt man als Befehle (`\kilo\watt\hour`), nicht als Text: Dann sind sie überall gleich gesetzt, aufrecht und im richtigen Abstand – 1 250,5 kW h statt „1250,5 kWh“.

7 ZITIEREN

Zitieren ist keine Formalie, sondern der Kern wissenschaftlichen Arbeitens: Es macht überprüfbar, was Sie übernommen haben und was Ihr eigener Beitrag ist. Formal falsch zitieren kostet Punkte; gar nicht zitieren kostet den Abschluss. LaTeX nimmt einem die Form ab, den Rest nicht.

7.1 BIBLATEX UND BIBER

Die Klasse bringt kein Literaturpaket mit, weil die Wahl des Stils der Arbeit gehört. Empfohlen und in beiden Beispielen benutzt ist biblatex (Lehman et al., 2024) mit dem Programm `biber` und dem Stil `apa`:

```
\usepackage[backend=biber,style=apa]{biblatex}
\addbibresource{literatur.bib}
```

Listing 7.1: Literaturverzeichnis einrichten

Am Ende der Arbeit, vor dem Anhang, steht dann:

```
\printbibliography[heading=bibintoc]
```

Listing 7.2: Das Verzeichnis ausgeben

Das `heading=bibintoc` sorgt dafür, dass das Literaturverzeichnis auch im Inhaltsverzeichnis auftaucht – ohne die Angabe fehlt es dort.

Ein Wort zur Werkzeugwahl: Man liest in älteren Anleitungen von natbib und `bibtex`. Beides funktioniert noch, kann aber weniger, insbesondere im Umgang mit Umlauten und nichtenglischen Titeln. Für eine neue Arbeit gibt es keinen Grund dazu. Achten Sie darauf, dass Ihr Editor tatsächlich `biber` aufruft und nicht `bibtex` – das ist die häufigste Ursache für ein leer bleibendes Literaturverzeichnis.

Der Stil `apa` folgt den Regeln der American Psychological Association (APA). Er ist in den Ingenieur- und Naturwissenschaften verbreitet, zitiert im Text mit Autor und Jahr und sortiert das Verzeichnis alphabetisch – beides verlangt die Richtlinie. Fragen Sie im Zweifel Ihre Prüferin, welcher Stil erwartet wird.

7.1.1 DEN ZITIERSTIL WECHSELN

Der Stil ist das Wort hinter `style=` in der `\usepackage`-Zeile. Ihn zu wechseln heißt: dieses eine Wort austauschen. Der Rest der Arbeit – alle `\cite`-Befehle, die `.bib`-Datei – bleibt unangetastet.

```
\usepackage[backend=biber,style=numeric]{biblatex}   % [1], [2], ...
\usepackage[backend=biber,style=ieee]{biblatex}      % IEEE-Norm
\usepackage[backend=biber,style=alphabetic]{biblatex}% [Mül23]
\usepackage[backend=biber,style=authoryear]{biblatex}% Müller 2023
```

Listing 7.3: Ein anderer Zitierstil

biblatex bringt Dutzende Stile mit; die häufigsten sind `numeric` (Zahlen in eckigen Klammern, in der Technik verbreitet), `ieee` (die Norm vieler Ingenieurzeitschriften), `apa` (Autor und Jahr, sozial- und naturwissenschaftlich) und `alphabetic` (Kürzel aus Name und Jahr). Der Wechsel ändert zweierlei zugleich: wie die Verweise im Text aussehen *und* wie das Literaturverzeichnis formatiert ist – beide gehören zum selben Stil und passen deshalb immer zusammen.

Nach einem Stilwechsel muss die Arbeit einmal vollständig neu übersetzt werden, `biber` eingeschlossen (Abschnitt 1.4), sonst mischt LaTeX alte und neue Formatierung. Und maßgeblich bleibt, was die Prüfung verlangt: Die Richtlinie der Fakultät schreibt Autor und Jahr vor, also `apa`. Wer ohne Rücksprache auf `numeric` umstellt, setzt formal sauber, aber an der Vorgabe vorbei.

7.2 DIE .BIB-DATEI

Die Literatur steht in einer eigenen Datei, üblicherweise `literatur.bib`. Jeder Eintrag hat einen Typ, einen Schlüssel und Felder:

```
@book{mittelbach2023companion,
  author    = {Frank Mittelbach and Ulrike Fischer},
  title     = {The LaTeX Companion},
  edition   = {3},
  publisher = {Addison-Wesley},
  address   = {Boston},
  year      = {2023},
  isbn      = {978-0-201-36299-2},
}

@article{graham1969bounds,
  author  = {R. L. Graham},
  title   = {Bounds on Multiprocessing Timing Anomalies},
  journal = {SIAM Journal on Applied Mathematics},
  volume  = {17},
  number  = {2},
  pages   = {416--429},
  year    = {1969},
  doi     = {10.1137/0117039},
}
```

Listing 7.4: Einträge in der .bib-Datei

Der Schlüssel (`mittelbach2023companion`) ist der Name, unter dem Sie die Quelle im Text ansprechen. Das Muster *NachnameJahrStichwort* hat sich bewährt: Es ist eindeutig, sprechend und man erkennt beim Lesen des Quelltexts, was gemeint ist.

Die wichtigsten Typen sind `@book` für Bücher, `@article` für Aufsätze in Zeitschriften, `@inproceedings` für Beiträge in Tagungsbänden, `@phdthesis` für Dissertationen und `@online` für Webseiten. Der Typ entscheidet, welche Felder verlangt sind und wie der Eintrag im Verzeichnis erscheint.

Zwei Hinweise, die immer wieder gebraucht werden. Erstens: Mehrere Autoren werden mit `and` verbunden, nie mit Komma. Zweitens: Geben Sie den Digital Object Identifier (DOI) an, wo es einen gibt – er ist die dauerhafte Adresse der Quelle, im Gegensatz zu einer *URL*, die in drei Jahren ins Leere zeigt. Bei Online-Quellen ohne DOI gehört ein `urldate` dazu, das Datum des Abrufs.

Eine Besonderheit betrifft die Groß- und Kleinschreibung: Manche Stile schreiben Titel klein. Was groß bleiben muss – Eigennamen, Abkürzungen –, schützt man mit geschweiften Klammern: `title = {Ein Vergleich von {MILP}-Verfahren}`.

7.3 LITERATURVERWALTUNG

Die `.bib`-Datei von Hand zu tippen, ist Zeitverschwendung und fehleranfällig. Nehmen Sie ein Verwaltungsprogramm:

- **Zotero** – kostenlos, mit Browser-Erweiterung, die eine Quelle mit einem Klick aus einer Datenbank oder einer Verlagsseite übernimmt. Exportiert nach BibTeX; die Erweiterung *Better BibTeX* hält die Datei automatisch aktuell und vergibt stabile Schlüssel.
- **JabRef** – arbeitet unmittelbar auf der `.bib`-Datei, ohne Export. Für wen LaTeX ohnehin das Werkzeug ist, oft die geradlinigere Wahl.
- **Citavi** – über die Hochschule verfügbar, im deutschsprachigen Raum verbreitet, exportiert ebenfalls nach BibTeX.

Welches Sie nehmen, ist gleich. Dass Sie *eines* nehmen, und zwar vom ersten gelesenen Aufsatz an, ist es nicht. Wer die Literatur erst am Ende zusammensucht, verbringt eine Woche mit der Frage, wo dieses eine Zitat noch einmal herkam.

Prüfen Sie die übernommenen Daten trotzdem. Verlagsseiten liefern erstaunlich oft kaputte Umlaute, fehlende Seitenzahlen oder eine falsche Auflage.

7.4 ZITIEREN IM TEXT

biblatex kennt zwei Grundformen, und der Unterschied ist grammatikalisch:

```
Der Makespan ist das gebräuchlichste Gütemaß im Scheduling
\parencite{pinedo2016scheduling}.

\textcite{graham1969bounds} zeigt, dass die Schranke scharf ist.
```

Listing 7.5: \parencite und \textcite

`\parencite` setzt die Quelle vollständig in Klammern. Man benutzt es, wenn die Quelle nur belegt, was im Satz steht – der Satz liest sich ohne sie vollständig.

`\textcite` macht den Autor zum Satzglied. Man benutzt es, wenn der Autor selbst Gegenstand der Aussage ist: wer etwas gezeigt, behauptet oder vorgeschlagen hat. Nur das Jahr steht dann in Klammern.

Der Fehler, den man häufig sieht, ist `\parencite` mit angeflanschtem Autornamen: „Wie Graham (Graham, 1969) zeigt“ – der Name steht doppelt. Wenn Sie den Autor im Satz nennen wollen, nehmen Sie `\textcite`.

Bei wörtlichen Zitaten und beim Verweis auf eine bestimmte Stelle gehört die Seitenzahl dazu. Sie steht im optionalen Argument:

```
\parencite[S.~42]{pinedo2016scheduling}
\parencite[vgl.][S.~42--45]{pinedo2016scheduling}
\parencite{pinedo2016scheduling,graham1969bounds}
```

Listing 7.6: Seitenangabe und mehrere Quellen

Bei zwei optionalen Argumenten ist das erste der Text davor („vgl.“), das zweite die Stelle. Mehrere Quellen trennt man mit Komma innerhalb *eines* `\parencite` – nicht durch mehrere Befehle nebeneinander, sonst stehen zwei Klammerpaare da.

Wörtliche Zitate setzt man mit `\enquote` aus csquotes, nicht mit von Hand gesetzten Anführungszeichen:

```
\usepackage{csquotes}
...
Scheduling ist \enquote{the allocation of resources to tasks over given
time periods} \parencite[S.~1]{pinedo2016scheduling}.
```

Listing 7.7: Ein wörtliches Zitat

csquotes setzt die Zeichen, die zur Dokumentsprache passen – im Deutschen „so“, im Englischen anders –, und richtig verschachtelt. Von Hand getippte Zeichen sind fast immer die falschen. biblatex lädt csquotes ohnehin gern; laden Sie es explizit.

7.5 WAS DIE RICHTLINIE VERLANGT

Zwei Punkte der Richtlinie erledigt der Stil `apa` von selbst, man sollte sie aber kennen.

Alphabetisch sortiert. Das Verzeichnis ist nach Nachnamen sortiert, nicht nach Reihenfolge des Auftretens und nicht nach Erscheinungsjahr. biblatex sortiert selbst; schreiben Sie die Einträge in die `.bib`-Datei in beliebiger Reihenfolge.

Nur zitierte Literatur. Ins Verzeichnis kommt, was im Text zitiert wird – und nur das. Kein „weiterführendes Schrifttum“, keine Bücher, die man gelesen, aber nicht verwendet hat. Auch das erledigt biblatex von selbst: Die `.bib`-Datei darf ruhig hundert Einträge enthalten, ins Verzeichnis kommen nur die zitierten. Das ist ein weiterer Grund, eine einzige, wachsende `.bib`-Datei zu pflegen, statt sie am Ende auszudünnen.

Der Befehl `\nocite{...}` nimmt eine Quelle ins Verzeichnis auf, ohne sie zu zitieren. Für eine Abschlussarbeit an dieser Fakultät ist er damit genau der Befehl, den Sie nicht brauchen.

8 VERZEICHNISSE

Eine Abschlussarbeit trägt fünf Verzeichnisse vor sich her: Inhalt, Tabellen, Abbildungen, Abkürzungen und Symbole; dazu am Ende der Index. Alle fünf erzeugt LaTeX selbst – vorausgesetzt, man hat beim Schreiben die richtigen Befehle benutzt. Genau darum geht es in diesem Kapitel.

8.1 INHALT, TABELLEN, ABBILDUNGEN

Die ersten drei sind mit je einer Zeile erledigt:

```
\tableofcontents      % Inhaltsverzeichnis
\listoftables         % Tabellenverzeichnis
\listoffigures        % Abbildungsverzeichnis
```

Listing 8.1: Die Standardverzeichnisse

Sie stehen im Vorspann, also nach `\maketitle` und vor `\hauptteil`. Die Reihenfolge gibt die Richtlinie vor: Inhalt, Tabellen, Abbildungen, danach Abkürzungen und Symbole.

Der Inhalt speist sich aus den Gliederungsbefehlen, die beiden anderen aus den `\caption`-Befehlen – weshalb die Kurzfassung im optionalen Argument von `\caption` (Abschnitt 4.5) so wichtig ist. Alle drei brauchen zwei Läufe: Der erste sammelt ein, der zweite setzt.

Ein Tabellen- oder Abbildungsverzeichnis mit zwei Einträgen ist übrigens keins. Wenn Ihre Arbeit drei Abbildungen hat, fragen Sie, ob das Verzeichnis wirklich verlangt ist.

8.2 ABKÜRZUNGEN

Für Abkürzungen bringt die Klasse das Paket acro fertig eingerichtet mit. Ein eigener Programmlauf ist nicht nötig; zwei `lualatex`-Läufe genügen.

Deklariert wird in der Präambel mit `\akronym`:

```
\akronym{MSE}{Mean Squared Error}
\akronym{MILP}{Mixed Integer Linear Programming}
\akronym{CNC}{Computerized Numerical Control}
```

Listing 8.2: Abkürzungen deklarieren

Das erste Argument schreibt man groß, so wie die Abkürzung gelesen wird. Im Text spricht man sie dann über den *klein* geschriebenen Schlüssel an:

```
Der \ac{mse} misst die mittlere quadratische Abweichung. Ein hoher
\ac{mse} deutet auf systematische Fehler hin.
```

Listing 8.3: Abkürzungen benutzen

Beim ersten Mal erscheint Mean Squared Error (MSE), danach nur noch MSE. Die Buchführung darüber, was schon genannt wurde, übernimmt acro – und zwar richtig auch dann, wenn Sie später ein Kapitel verschieben. Genau das ist der Grund, es nicht von Hand zu machen.

Neben `\ac` gibt es `\acs` für die Kurzform allein (MSE), `\acl` für die Langform allein (Mean Squared Error), `\acf` für die volle Form unabhängig davon, ob sie schon genannt wurde, und `\acp` für den Plural (DOIs).

Weitere Schlüssel von acro reicht das optionale Argument durch, etwa für eine abweichende Pluralform:

```
\akronym[long-plural-form = Gradientenabstiegsverfahren]{GD}{Gradient Descent}
```

Listing 8.4: Eine abweichende Pluralform

8.2.1 KAPITÄLCHEN

Kurzformen erscheinen in *Kapitälchen* – MSE, nicht MSE. Das ist Absicht: Vier Großbuchstaben mitten im Satz sind ein optischer Fleck, Kapitälchen fügen sich ein.

Technisch ist das heikler, als es aussieht. Echte Kapitälchen entstehen nur aus *Klein*buchstaben; deshalb schreibt `\akronym` das erste Argument intern klein. Damit beim Kopieren aus dem PDF und bei der Volltextsuche trotzdem „MSE“ herauskommt und nicht „mse“, hinterlegt `\akronym` die Großschreibung zusätzlich für die PDF-Lesezeichen und – über accsupp – für den kopierbaren Text. Darum muss man sich nicht kümmern; man sollte nur wissen, dass `\textsc{MSE}` *nicht* funktioniert (auf Großbuchstaben bewirkt `\textsc` nichts) und `\textsc{mse}` zwar Kapitälchen liefert, beim Kopieren aber „mse“.

8.2.2 ABKÜRZUNGEN IM EIGENEN LANGNAMEN

Manche Abkürzungen stecken in ihrem eigenen Langnamen. Die Fakultät NXT heißt vollständig „NXT Nachhaltigkeit und Technologie“ – die übliche erste Nennung „NXT Nachhaltigkeit und Technologie (NXT)“ wäre doppelt gemoppelt. Sauber wird das so:

```
\akronym[first-style = long]{NXT}{\acs*{nxt} Nachhaltigkeit und Technologie}
```

Listing 8.5: Eine Abkürzung, die in ihrem Langnamen steckt

In der Langform steht die Kurzform als `\acs*{nxt}`, damit sie in Kapitälchen erscheint und beim Kopieren trotzdem „NXT“ liefert; der Stern verhindert, dass die Nennung mitgezählt wird. `first-style = long` sorgt dafür, dass bei der ersten Nennung nur der Langname ohne Klammerzusatz erscheint.

8.2.3 VORDEFINIERTE ABKÜRZUNGEN

Die Klasse deklariert `nxt` und `hsrt` bereits selbst, passend zu den Vorbelegungen von `\fakultaet` und `\hochschule`. `\ac{nxt}` und `\ac{hsrt}` stehen also ohne Zutun zur Verfügung – diese Anleitung benutzt sie. Da nur benutzte Abkürzungen ins Verzeichnis kommen, stören sie nicht, wenn eine Arbeit sie nicht braucht.

8.2.4 DAS VERZEICHNIS AUSGEBEN

```
\printacronyms
```

Listing 8.6: Das Abkürzungsverzeichnis

Der Befehl steht im Vorspann, üblicherweise nach den anderen Verzeichnissen. Er erscheint als eigenes, unnummeriertes Kapitel und enthält nur die Abkürzungen, die tatsächlich benutzt wurden. Die Überschrift richtet sich nach der Sprachoption: „Abkürzungsverzeichnis“ bei `german`, „List of Abbreviations“ bei `english`.

Dass nur Benutztes erscheint, ist bequem: Man darf großzügig deklarieren, auch auf Vorrat, ohne das Verzeichnis aufzublähen.

8.3 SYMBOLE

Eine Arbeit mit Formeln braucht ein Symbolverzeichnis. Die Klasse stellt dafür `\Symbol` bereit:

```
\Symbol[min]{cmax}{C_{\max}}{Gesamtdauer eines Ablaufplans}
\Symbol{alpha}{\alpha}{Abkühlfaktor}
\Symbol[\unit{\kilo\watt\hour}]{ebedarf}{E}{Energiebedarf}
```

Listing 8.7: Symbole deklarieren

Der Befehl nimmt vier Angaben: die Einheit (optional, in eckigen Klammern), ein Kürzel, das Symbol selbst und seine Bedeutung. Das Symbol schreibt man ohne Dollarzeichen – `\Symbol` setzt es selbst in den Mathematikmodus, weshalb `C_{\max}` genügt.

Ausgegeben wird das Verzeichnis mit:

```
\printsymbols
```

Listing 8.8: Das Symbolverzeichnis

Das Symbolverzeichnis dieser Anleitung, gleich hinter dem Abkürzungsverzeichnis, zeigt das Ergebnis. Ein Unterschied zum Abkürzungsverzeichnis ist wichtig: `\printsymbols` gibt *alle* deklarierten Symbole aus, nicht nur die benutzten. Ein Symbolverzeichnis soll ja nachschlagbar machen, was in den Formeln vorkommt, und dass ein Symbol nirgends mit einem eigenen Befehl angesprochen wird, heißt nicht, dass es nicht in einer Formel steht. Deklarieren Sie hier also sparsamer als bei den Abkürzungen.

Technisch sind Symbole in derselben Maschinerie untergebracht wie Abkürzungen – acro verwaltet beide. Die Klasse hält sie über ein Merkmal auseinander, sodass Symbole nicht im Abkürzungsverzeichnis auftauchen und umgekehrt. Man muss davon nichts wissen; es erklärt nur, warum beide Verzeichnisse gleich aussehen.

8.4 DER INDEX

Ein *Index* lohnt sich nicht für jede Arbeit, wohl aber für längere Texte, die auch als Nachschlagewerk dienen. Die Klasse richtet ihn fertig ein: Sie lädt makeidx und ruft `\makeindex` auf; man muss beides nicht selbst tun.

Es gibt zwei Befehle. `\index` trägt ein Wort in den Index ein, ohne im Text etwas zu ändern. `\Index` tut dasselbe und setzt das Wort zusätzlich kursiv:

```
Der \Index{Makespan} ist die Höhe des höchsten Balkens.
Der Makespan\index{Makespan} ist die Höhe des höchsten Balkens.
```

Listing 8.9: Indexeinträge

`\Index` ist für die Stelle gedacht, an der ein Begriff *eingeführt* wird – dort steht er im Deutschen kursiv, und genau dorthin soll der Index zeigen. Kommt das Wort später wieder vor, ohne dass es erklärt wird, nimmt man `\index` oder gar nichts.

Beachten Sie: `\Index` nimmt genau *ein* Argument. Ein optionales Argument für einen abweichenden Sortiereintrag gibt es nicht. Wer das braucht, nimmt `\index` und schreibt kursiv von Hand.

Unterpunkte trennt ein Ausrufezeichen:

```
\index{Verfahren!exaktes}
\index{Verfahren!heuristisches}
```

Listing 8.10: Ober- und Unterpunkte im Index

Im Index steht dann „Verfahren“ als Oberpunkt mit den eingerückten Unterpunkten „exaktes“ und „heuristisches“ darunter. Mehr als zwei Ebenen braucht man praktisch nie.

Ausgegeben wird der Index am Ende der Arbeit, nach dem Anhang:

```
\printindex
```

Listing 8.11: Den Index ausgeben

Anders als das Abkürzungsverzeichnis braucht der Index einen eigenen Programmlauf: `makeindex` sortiert die Einträge, die der `lualatex`-Lauf gesammelt hat, und erst der nächste `lualatex`-Lauf baut das Ergebnis ein (Abschnitt 1.4). Wer den Index leer vorfindet, hat meist `makeindex` vergessen.

Ein Rat zum Schluss: Legen Sie den Index nicht am Ende an. Ein Index, den man in der letzten Woche über den fertigen Text stülpt, wird entweder lückenhaft oder beliebig. Setzen Sie `\Index` beim Schreiben, dort, wo Sie einen Begriff einführen – dann ist er nebenbei fertig und zeigt auf die richtigen Stellen.

9 EIGENE MAKROS

Ein *Makro* ist ein selbst definierter Befehl. Man schreibt einen Namen und hinterlegt, wofür er steht. Das klingt nach einer Spielerei für Fortgeschrittene, ist aber das Mittel, mit dem eine lange Arbeit konsistent bleibt.

9.1 \NEWCOMMAND

Die einfachste Form definiert eine Abkürzung für einen festen Text:

```
\newcommand{\Firma}{Erpolino Metallverarbeitung GmbH}
...
Die Arbeitsvorbereitung der \Firma{} verteilt die Arbeitsgänge nach
Erfahrung. Damit steht \Firma{} vor demselben Problem wie viele
Lohnfertiger.
```

Listing 9.1: Ein Makro ohne Argumente

Die Definition steht in der Präambel, der Aufruf im Text. Die leeren geschweiften Klammern hinter `\Firma` sind wichtig: Ohne sie frisst LaTeX das folgende Leerzeichen, und aus „…GmbH verteilt“ wird „…GmbHverteilt“. Der Grund ist, dass LaTeX Leerzeichen nach einem Befehlsnamen als dessen Ende liest, nicht als Text. Die leeren Klammern beenden den Namen und retten das Leerzeichen.

9.2 ARGUMENTE

Ein Makro kann bis zu neun Argumente entgegennehmen. Die Zahl steht in eckigen Klammern, die Argumente selbst heißen `#1` bis `#9`:

```
\newcommand{\Quelltext}[1]{\texttt{Code/#1}}
...
Die Rechnung steht in \Quelltext{sa.py}.
```

Listing 9.2: Ein Makro mit Argument

Ein Argument darf optional sein und einen Vorgabewert bekommen. Der steht in einer zweiten eckigen Klammer:

```
\newcommand{\Wert}[2][min]{\qty{#2}{#1}}
...
\Wert{87.25}          % ergibt 87,25 min
\Wert[\second]{12.4}  % ergibt 12,4 s
```

Listing 9.3: Ein optionales Argument mit Vorgabewert

Das erste Argument ist dann das optionale; die übrigen folgen in geschweiften Klammern.

`\renewcommand` arbeitet genauso, definiert aber einen *bestehenden* Befehl um. Der Unterschied ist eine Sicherung: `\newcommand` bricht mit einer Fehlermeldung ab, wenn der Name schon vergeben ist, `\renewcommand` bricht ab, wenn er es nicht ist. So kann man weder versehentlich einen LaTeX-Befehl überschreiben noch ins Leere umdefinieren. Nehmen Sie im Zweifel `\newcommand` und geben Sie dem Makro einen anderen Namen – die Fehlermeldung ist ein Dienst, keine Schikane.

9.3 WOZU DAS GUT IST

Der offensichtliche Nutzen ist Tipparbeit. Der eigentliche ist ein anderer.

Fachbegriffe bleiben einheitlich. Eine Arbeit läuft über Monate. Über diese Zeit heißt dieselbe Sache mal „Makespan“, mal „Gesamtdauer“, mal „$C_{\max}$“; eine Firma heißt mal mit, mal ohne „GmbH“. Ein Makro macht die Entscheidung einmal. Und wenn sie sich ändert – der Industriepartner besteht in der letzten Woche darauf, anders genannt zu werden –, ändert man eine Zeile statt vierzig Fundstellen, von denen man drei übersieht.

Notation bleibt einheitlich. Dasselbe gilt für Symbole:

```
\newcommand{\Makespan}{C_{\max}}
\newcommand{\Zuordnung}[2]{x_{#1#2}}
...
\[ \Makespan = \max_{i \in M} \sum_{j \in J} p_{ij}\, \Zuordnung{i}{j} \]
```

Listing 9.4: Makros für die Notation

Wer sich in Kapitel 6 entscheidet, die Zuordnungsvariable doch y_{ij} zu nennen, ändert eine Zeile – statt in fünfzig Formeln zu suchen und in zweien das x stehen zu lassen.

Zahlen bleiben richtig. Das ist der wichtigste Fall. Eine Zahl, die im Text steht, stammt aus einer Rechnung. Die Rechnung ändert sich: ein Datensatz kommt dazu, ein Fehler wird gefunden, ein Parameter wird angepasst. Dann muss die Zahl im Text mitwandern – und zwar überall, in der Kurzfassung, im Ergebniskapitel, in der Zusammenfassung. Von Hand geht das schief. Zuverlässig.

Der Ausweg: Das Programm, das rechnet, schreibt seine Ergebnisse als Makros in eine Datei, und die Arbeit liest sie ein.

```
% Tabellen/kennzahlen.tex -- erzeugt von Code/tabellen.py, nicht von Hand ändern
\newcommand{\Optimum}{87,25}
\newcommand{\MakespanGreedy}{412,50}
\newcommand{\ZahlAufgaben}{24}
```

Listing 9.5: Von der Rechnung erzeugte Makros

```
\input{Tabellen/kennzahlen}
```

Listing 9.6: Die Makros in der Hauptdatei einlesen

Im Text steht dann keine Zahl mehr, sondern:

```
Für das betrachtete Los aus \ZahlAufgaben{} Arbeitsgängen liefert das
exakte Verfahren eine optimale Gesamtdauer von \Optimum{} Minuten,
```

```
während die bisherige Praxis \MakespanGreedy{} Minuten benötigt.
```

Listing 9.7: Zahlen im Text

Ein Lauf des Rechenprogramms, ein Lauf von `lualatex` – und jede Zahl in der Arbeit stimmt wieder. Die Beispielthesis macht das konsequent: Sie erzeugt ihre Kennzahlen mit `Code/tabellen.py` nach `Tabellen/kennzahlen.tex`, und im Fließtext steht keine Zahl, die von der Rechnung abweichen könnte. Der Aufwand ist ein Nachmittag; er zahlt sich beim ersten Mal aus, wenn ein Betreuer eine Woche vor der Abgabe eine Annahme in Frage stellt.

Dasselbe Muster taugt für Tabellen: Das Programm schreibt den kompletten Tabellenrumpf in eine Datei, die Arbeit bindet ihn mit `\input` ein.

9.4 WANN MAN ES LASSEN SOLLTE

Makros können auch schaden. Ein Makro, das Formatierung versteckt, ist meist ein Zeichen dafür, dass man gegen die Klasse arbeitet statt mit ihr:

```
\newcommand{\wichtig}[1]{\textbf{\color{red}#1}}
```

Listing 9.8: Ein Makro, das man nicht schreiben sollte

Wenn etwas hervorzuheben ist, gibt es `\emph`; wenn die Klasse etwas anders setzen soll, als Sie möchten, ist das eine Frage an die Klasse und nicht an ein Makro. Und ein Makro, das nur einmal vorkommt, ist eine Sichtblende: Es verbirgt, was dasteht, ohne etwas zu sparen.

Die Faustregel: Makros für Inhalte, die sich wiederholen oder ändern könnten. Nicht für Layout – dafür ist die Klasse da.

9.5 \NEWDOCUMENTCOMMAND

Für die meisten Fälle reicht `\newcommand`. An seine Grenzen stößt es bei mehreren optionalen Argumenten, bei Sternformen wie `\section*` oder bei Argumenten in eckigen statt geschweiften Klammern. Dafür gibt es den moderneren Befehl `\NewDocumentCommand`:

```
\NewDocumentCommand{\Kennzahl}{s O{min} m}{%
  \IfBooleanTF{#1}{\textbf{#3}}{#3}\,\unit{#2}%
}
...
\Kennzahl{87,25}              % 87,25 min
\Kennzahl[\second]{12,4}      % 12,4 s
\Kennzahl*{87,25}             % 87,25 min, Zahl fett
```

Listing 9.9: Ein Makro mit Sternform und optionalem Argument

Die Zeichenkette `s O{min} m` beschreibt die Argumente: `s` für eine Sternform, `O{min}` für ein optionales Argument mit Vorgabewert, `m` für ein gewöhnliches. Die Klasse selbst benutzt diesen Mechanismus, um `\akronym` und `\Symbol` zu definieren.

Für eine Abschlussarbeit brauchen Sie das selten. Wenn Sie merken, dass `\newcommand` nicht mehr ausreicht, wissen Sie nun, wo Sie weitersuchen – der LaTeX Companion (Mittelbach & Fischer, 2023) beschreibt es ausführlich.

10 Das Kolloquium mit Beamer

Nach der Arbeit kommt der Vortrag. Für die Folien gibt es die Klasse beamer (Tantau et al., 2024) und dazu das Thema NXT, das dem Paket beiliegt. Wer die Arbeit mit nxtthesis gesetzt hat, findet sich sofort zurecht: Es ist dieselbe Sprache, dieselben Farben, dasselbe Erscheinungsbild.

10.1 Ein Gerüst

Folien sind ein eigenes Dokument mit eigener Klasse:

```
\documentclass[aspectratio=169]{beamer}
\usetheme{NXT}
\usepackage[ngerman]{babel}

\title{Optimierung des Mitarbeitereinsatzes}
\subtitle{Kolloquium zur Bachelorarbeit}
\author{Jean Dujardin}
\date{\today}

\begin{document}

\begin{frame}[plain]
  \titlepage
\end{frame}

\begin{frame}{Ausgangslage}
  \begin{itemize}
    \item Arbeitsgänge werden nach Erfahrung verteilt
    \item Die Gesamtdauer schwankt stark
  \end{itemize}
\end{frame}

\end{document}
```

Listing 10.1: Das Gerüst einer Präsentation

Das Thema heißt `NXT` und wird mit `\usetheme{NXT}` geladen – großgeschrieben. Eigene Optionen hat es nicht; es ist fertig eingestellt.

`aspectratio=169` ist eine Option der Klasse beamer, kein Thema-Schalter. Sie ist zu empfehlen: Beamer und Bildschirme sind heute im Format 16:9, und das NXT-Thema ist darauf ausgelegt. Ohne die Option bekommen Sie schwarze Balken links und rechts.

Auch die Folien setzt `lualatex`, nicht `pdflatex`: Das Thema benutzt *Segoe UI* als Schrift, Euler-Math für die Formeln und den NXT Logo Font für die Aufzählungszeichen und die Logos auf Titelfolie und Kopfzeile (Abschnitt 11.2).

Ein vollständiges Beispiel liegt dem Paket als `BeispielfolienNXT.tex` bei. Es führt alles vor, was hier beschrieben ist, und ist der schnellste Einstieg: kopieren, ausprobieren, überschreiben.

10.2 FOLIEN

Jede Folie ist eine `frame`-Umgebung; ihr Titel steht in geschweiften Klammern dahinter:

```
\begin{frame}{Ergebnisse}
  \begin{itemize}
    \item Das exakte Verfahren löst in 12,4 Sekunden
    \item Die Heuristik braucht länger und wird nicht besser
  \end{itemize}
\end{frame}
```

Listing 10.2: Eine Folie

Zwei Optionen braucht man regelmäßig. `[plain]` unterdrückt Kopf- und Fußzeile – das ist für die Titelfolie gedacht, deren Fußzeile das Thema ohnehin unterdrückt. `[fragile]` ist nötig, sobald `verbatim`, `lstlisting` oder Ähnliches auf der Folie steht; ohne die Option bricht der Lauf mit einer rätselhaften Meldung ab. Wer Quelltext zeigt, braucht sie.

Zwei Spalten macht man mit `columns`:

```
\begin{columns}[T]
  \begin{column}{0.55\textwidth}
    \begin{itemize}
      \item Der Makespan sinkt um 21 Prozent
      \item Die Auslastung wird gleichmäßiger
    \end{itemize}
  \end{column}
  \begin{column}{0.4\textwidth}
    \includegraphics[width=\textwidth]{Abbildungen/auslastung.pdf}
  \end{column}
\end{columns}
```

Listing 10.3: Zwei Spalten

Das `[T]` richtet die Spalten oben aus. Ohne es zentriert Beamer sie senkrecht, was fast nie gewollt ist.

Für Hervorhebungen gibt es `block`, `alertblock` und `exampleblock` – ein Kasten mit Überschrift für Hinweise, Warnungen und Beispiele. Und `\emph` setzt das Thema in NXT-Grün und fett, statt kursiv wie im Fließtext: Kursives liest sich auf einer Folie aus zehn Metern nicht.

10.3 GLIEDERUNG

`\section` erzeugt auf Folien keine Überschrift, sondern nur einen Eintrag in der Gliederung. Sichtbar wird sie durch `\tableofcontents`:

```
\begin{frame}{Überblick}
  \tableofcontents
\end{frame}

\section{Ausgangslage}
\begin{frame}{Wie bisher geplant wird}
  ...
\end{frame}

\section{Modell}
```

Listing 10.4: Eine Gliederungsfolie

Bei einem Kolloquium von 20 Minuten reicht eine Gliederungsfolie am Anfang. Zwischengliederungen vor jedem Abschnitt sind bei fünf Abschnitten fünf verlorene Folien.

10.4 OVERLAYS

Eine Folie kann schrittweise erscheinen. Am einfachsten ist `\pause`:

```
\begin{frame}{Das überraschende Ergebnis}
  \begin{itemize}
    \item Die Heuristik ist im Mittel schlechter
    \pause
    \item Und sie braucht mehr Rechenzeit
    \pause
    \item Erst ab 60 Arbeitsgängen kehrt sich das um
  \end{itemize}
\end{frame}
```

Listing 10.5: Schrittweise aufdecken

Feiner steuert `\onslide` mit einer Angabe, auf welchen Schritten etwas sichtbar sein soll:

```
\onslide<1->{Immer sichtbar}
\onslide<2->{Ab dem zweiten Schritt}
\onslide<3>{Nur im dritten Schritt}
```

Listing 10.6: Gezielte Overlays

Dieselbe Angabe versteht auch `\item`: `\item<2->{Ab Schritt 2}`.

Overlays sind ein Werkzeug, kein Selbstzweck. Sie sind richtig, wenn die Reihenfolge zur Sache gehört – ein Verfahren, das schrittweise aufgebaut wird, oder eine Pointe, die vorweggenommen würde. Sie sind falsch, wenn eine Liste nur deshalb einzeln aufpoppt, damit etwas passiert. Das Publikum liest schneller, als Sie klicken, und merkt es.

Bedenken Sie auch: Jeder Overlay-Schritt ist eine eigene Seite im PDF. Aus zwanzig Folien werden schnell sechzig Seiten – was bei der Frage „gehen Sie bitte noch einmal zurück zu dem Diagramm“ lästig wird.

10.5 Grafiken auf Folien

Grafiken bindet man ein wie in der Arbeit, mit `\includegraphics` (Abschnitt 5.1). Auch TikZ funktioniert, und die NXT-Farben stehen zur Verfügung – das Thema lädt nxtlatexcolors, dieselbe Palette wie die Thesis-Klasse. Eine TikZ-Zeichnung aus der Arbeit lässt sich also unverändert auf eine Folie legen.

Nur: Meistens sollte man es nicht. Eine Abbildung aus der Arbeit ist für eine A4-Seite gemacht, die man aus 40 cm liest. Auf der Leinwand sitzt die hinterste Reihe zehn Meter weg. Achsenbeschriftungen, die auf Papier gut lesbar sind, sind dort unsichtbar.

Für den Vortrag heißt das: neu zeichnen, nicht vergrößern. Weniger Kurven, größere Schrift, dickere Linien, keine Legende mit sieben Einträgen. Wenn die Abbildung aus einem Programm kommt (Kapitel 5), ist das ein zweiter Aufruf mit anderen Größen – ein weiterer Grund, Abbildungen nicht von Hand zu bauen.

Die Logo-Makros aus Abschnitt 11.2 stehen auch auf Folien zur Verfügung. Das Thema setzt `\nxtlogo` selbst in die Kopfzeile jeder Folie und `\nxthsrtlogo` auf die Titelfolie; von Hand braucht man sie selten. Wer andere Logos will, definiert die Makros nach `\usetheme` um:

```
\renewcommand{\nxthsrtlogo}[1][]{\includegraphics[#1]{mein-logo.pdf}}
```

Listing 10.7: Ein eigenes Logo auf der Titelfolie

10.6 Farben

Die Palette aus nxtlatexcolors ist dieselbe wie in der Arbeit: `nxtgreen` (RGB 0, 139, 146) als Hausfarbe, dazu `nxtgreen.2` bis `nxtgreen.4`, die Akzente `nxt.violet`, `nxt.blue`, `nxt.orange` und die Grautöne `nxt.gray0` bis `nxt.gray8`. `nxtcolor` ist ein Alias für `nxtgreen`. Die Folie „Farben“ in `BeispielfolienNXT.tex` zeigt alle nebeneinander.

Das Thema setzt Titel, Folienüberschriften und Aufzählungszeichen bereits in NXT-Grün. Wenn Sie selbst Farbe einsetzen – in einem Diagramm etwa –, nehmen Sie diese Namen und keine eigenen. Und nehmen Sie sparsam: Farbe hebt hervor, solange sie selten ist.

10.7 Zum Vortrag

Zum Schluss ein paar Hinweise, die mit LaTeX nichts zu tun haben, aber über das Kolloquium mehr entscheiden als jedes Overlay.

Rechnen Sie mit zwei Minuten je Folie. Für 20 Minuten sind das etwa zehn bis fünfzehn Folien. Wer dreißig mitbringt, hetzt und wird abgebrochen.

Die Folie ist nicht die Arbeit. Niemand liest Fließtext von der Leinwand, während Sie sprechen – man kann nicht gleichzeitig lesen und zuhören. Stichworte, eine Abbildung, eine Zahl. Was Sie sagen, steht nicht auf der Folie; deshalb sagen Sie es ja.

Zeigen Sie Ihr Ergebnis. Prüfende kennen den Stand der Technik. Was sie interessiert, ist, was Sie gemacht haben, warum es richtig ist und was dabei herauskam. Der Grundlagenteil, der in der Arbeit zwanzig Seiten braucht, ist im Vortrag eine Folie.

Ein negatives Ergebnis ist ein Ergebnis. Dass sich die Heuristik in der betrachteten Größenordnung nicht lohnt, ist eine Erkenntnis, sofern sie sauber begründet ist. Sie zu verstecken, ist der Fehler – nicht, sie zu haben.

Üben Sie laut, mit Uhr. Nicht im Kopf. Der Unterschied zwischen „das passt schon“ und der Wirklichkeit beträgt zuverlässig fünf Minuten.

Bereiten Sie Reservefolien vor. Für Fragen, die Sie erwarten: die Herleitung, die Sie übersprungen haben, die Tabelle mit allen Läufen, die Parameter des Verfahrens. Hinter der letzten Folie, ohne Nummer im Vortrag. Eine Frage mit „dazu habe ich etwas vorbereitet“ zu beantworten, wirkt.

Nehmen Sie das PDF auf einem Stick mit. Und prüfen Sie es vorher auf einem fremden Rechner. PDF ist gerade deshalb das richtige Format: Es sieht überall gleich aus. Aber nur, wenn Sie es dabeihaben.

11 WERKZEUGE FÜR DIE TÄGLICHE ARBEIT

Kapitel 1 hat gesagt, was installiert sein muss und dass man einen Editor braucht. Dieses Kapitel wird konkret: Wie richtet man sich so ein, dass ein einziger Tastendruck das fertige PDF erzeugt? Und welche Programme rund um LaTeX nehmen einem Arbeit ab, von der man nicht wusste, dass man sie hat?

Der Schwerpunkt liegt auf *Visual Studio Code*, weil das Paket dafür eine fertige Konfiguration mitbringt. Wer lieber mit einem anderen Editor arbeitet, findet in Abschnitt 11.4 die Alternativen; die Grundsätze aus Abschnitt 1.5 gelten dort unverändert.

11.1 VISUAL STUDIO CODE MIT EINEM KLICK

11.1.1 DIE ERWEITERUNG

Visual Studio Code kann von Haus aus kein LaTeX. Es braucht die Erweiterung *LaTeX Workshop* von James Yu. Sie ist im Erweiterungsmarkt zu finden – suchen Sie nach „LaTeX Workshop“, der Rest ist ein Klick auf „Install“. Die Erweiterung bringt Syntaxhervorhebung, Vervollständigung von Befehlsnamen, einen PDF-Betrachter und vor allem die Steuerung der Programmläufe mit.

Eine LaTeX-Distribution ersetzt sie nicht. `lualatex`, `biber` und `makeindex` müssen installiert und über den Suchpfad erreichbar sein (Abschnitt 1.2); die Erweiterung ruft sie nur auf.

11.1.2 WARUM EIN EINZELNER LAUF NICHT GENÜGT

Abschnitt 1.4 hat den Grund schon genannt: LaTeX liest die Datei einmal von vorne nach hinten und kann deshalb beim ersten Lauf nichts wissen, was weiter hinten steht. Querverweise, Seitenzahlen und das Inhaltsverzeichnis gehen den Umweg über die Hilfsdateien. `biber` und `makeindex` sind eigene Programme, die dazwischen laufen müssen, und die Verzeichnisse von acro brauchen ihrerseits zwei Durchgänge, ehe sie vollständig sind.

Genau deshalb konfiguriert man den Editor nicht mit einem Befehl, sondern mit einem *Rezept* aus fünf Schritten:

```
lualatex  -> Text setzen, .aux/.bcf/.idx schreiben
biber     -> Literaturverzeichnis erzeugen
makeindex -> Index sortieren
lualatex  -> Literatur und Index einbauen
lualatex  -> Verweise, Seitenzahlen, acro-Verzeichnisse richtigstellen
```

Listing 11.1: Was hinter dem Knopf steckt

Es ist dieselbe Folge, die auch das `Makefile` abarbeitet. Der Unterschied ist nur, wer sie anstößt.

11.1.3 DIE KONFIGURATION

Die Einstellungen stehen in der Datei `.vscode/settings.json` im Wurzelverzeichnis des Pakets. Sie gelten für alles, was in diesem Verzeichnis liegt – die Anleitung, die Beispielthesis, die Folien –, ohne dass man an den globalen Einstellungen von Visual Studio Code (VSC) etwas ändern muss. Wer die Klasse für die eigene Arbeit benutzt, kopiert die Datei in das eigene Projektverzeichnis und ist fertig.

Zuerst die Rezepte. Das erste ist die Voreinstellung und steht oben im Menü:

```
"latex-workshop.latex.outDir": "%DIR%",

// Das erste Rezept ist die Voreinstellung und steht oben im Menü.
"latex-workshop.latex.recipes": [
    {
        "name": "nxtthesis: vollständig (lualatex, biber, makeindex, 2x lualatex)",
        "tools": ["lualatex", "biber", "makeindex", "lualatex", "lualatex"]
    },
    {
        "name": "nxtthesis: schnell (ein lualatex-Lauf)",
        "tools": ["lualatex"]
    },
    {
        "name": "nxtthesis: make (rechnet bei der Beispielthesis auch die Daten)",
        "tools": ["make"]
    }
],
```

Listing 11.2: Die Rezepte aus `.vscode/settings.json`

Ein Rezept nennt nur Namen. Was diese Namen bedeuten, steht getrennt davon in der Liste der Werkzeuge:

```
"latex-workshop.latex.tools": [
    {
        "name": "lualatex",
        "command": "lualatex",
        "args": [
            "-synctex=1",
            "-interaction=nonstopmode",
            "-file-line-error",
            "%DOC%"
        ],
        "env": {}
    },
    {
        "name": "biber",
        "command": "biber",
        "args": ["%DOCFILE%"],
        "env": {}
    },
```

```
    {
        "name": "makeindex",
        "command": "makeindex",
        "args": ["%DOCFILE%"],
        "env": {}
    },
    {
        // Ruft das Makefile im Verzeichnis des Dokuments auf.
        "name": "make",
        "command": "make",
        "args": [],
        "env": {}
    }
],
```

Listing 11.3: Die Werkzeuge aus `.vscode/settings.json`

Die Trennung hat einen Sinn: `lualatex` ist einmal beschrieben und wird im ersten Rezept dreimal benutzt. Wer eine Option ändern will – etwa `-file-line-error` herausnehmen –, ändert sie an einer Stelle.

Die Platzhalter setzt die Erweiterung ein. `%DOC%` ist das Hauptdokument ohne Endung, `%DOCFILE%` dasselbe ohne Verzeichnis (`biber` und `makeindex` wollen es so), und `%DIR%` bei `outDir` sorgt dafür, dass das PDF neben der Quelldatei landet und nicht in einem Unterverzeichnis – sonst findet das `Makefile` es nicht mehr.

Die drei Optionen von `lualatex` lohnen eine Erklärung. `-synctex=1` erzeugt die Datei, auf der der Sprung zwischen Quelle und PDF beruht (Abschnitt 11.1.5). `-interaction=nonstopmode` hält bei einem Fehler nicht an und fragt, sondern läuft durch und schreibt alles ins Protokoll – was man will, wenn fünf Läufe hintereinander abzuarbeiten sind. `-file-line-error` setzt vor jede Meldung Datei und Zeilennummer, sodass VSC sie anklickbar macht und direkt an die richtige Stelle springt.

11.1.4 BAUEN

Gebaut wird mit `Strg+Alt+B` (Windows, Linux) beziehungsweise `Cmd+Alt+B` (macOS). Wer die Tastenkombination nicht mag: In der Seitenleiste erscheint nach der Installation ein TEX-Symbol; darunter steht „Build LaTeX project“. Beides startet das erste Rezept – also den vollen Ablauf.

Ein anderes Rezept wählt man über „Recipe: …“ in derselben Leiste. Das lohnt sich beim Schreiben: Wer nur wissen will, ob der Absatz umbricht, den er gerade getippt hat, nimmt „nxtthesis: schnell“ und wartet einen Lauf statt fünf. Verweise stehen dann als `??` im Text und die Literatur fehlt – das ist der Preis und meist verschmerzbar. Vor dem Ansehen des Ergebnisses, spätestens vor der Abgabe, muss das volle Rezept laufen.

Das PDF zeigt `llatex-workshop.view.pdf.viewer: ttab"` in einem Reiter neben der Quelle. Man kann es auch im Browser oder in einem externen Betrachter öffnen; der Reiter hat den Vorteil, dass alles in einem Fenster bleibt und der Sprung zwischen Quelle und PDF ohne Umweg funktioniert.

11.1.5 SyncTeX

SyncTeX ist die Einstellung, die man nach einer Woche nicht mehr missen will. Ein `Strg`- beziehungsweise `Cmd`-Klick auf eine Stelle im PDF springt in der Quelldatei an die Zeile, die diese Stelle erzeugt hat. Umgekehrt springt derselbe Klick in der Quelle an die passende Stelle im PDF.

Bei einer hundertseitigen Arbeit ist das der Unterschied zwischen „da war doch irgendwo dieser Satz“ und einem Klick. Voraussetzung ist allein `-synctex=1` in den Argumenten von `lualatex` – das steht in der Konfiguration bereits. Fehlt die Option, wird die Datei `.synctex.gz` nicht erzeugt, und der Klick tut nichts.

11.1.6 Warum nicht automatisch gebaut wird

Zwei Einstellungen wirken auf den ersten Blick unbequem und sind es nicht:

```
"latex-workshop.latex.autoBuild.run": "never",
```

Listing 11.4: Kein Bau beim Speichern

LaTeX Workshop baut in der Voreinstellung bei jedem Speichern. Das ist bei einem einseitigen Brief angenehm und bei einer Abschlussarbeit eine Plage: Jedes `Strg+S` löst fünf Durchgänge aus, und da man beim Schreiben alle paar Sekunden speichert, rechnet der Rechner dauernd und der Lüfter läuft. Deshalb steht `never` – gebaut wird, wenn Sie es sagen.

Die Erweiterung räumt dafür nach jedem Bau auf. `autoClean.run` steht auf `onBuilt`, und `clean.fileTypes` zählt auf, was verschwinden darf: `*.aux`, `*.bbl`, `*.toc`, `*.bcf` und die übrigen Hilfsdateien. Das Verzeichnis bleibt übersichtlich, und veraltete Hilfsdateien – eine der häufigsten Fehlerursachen (Abschnitt 11.6) – können sich gar nicht erst ansammeln.

11.1.7 LuaLaTeX ist Pflicht

In allen drei Rezepten steht `lualatex`, nirgends `pdflatex`. Das ist keine Geschmacksfrage. Die Klasse nxtthesis lädt fontspec und unicode-math, um die Schriften aus Abschnitt 3.1.2 einzustellen. Beide Pakete setzen eine Unicode-fähige Maschine voraus und brechen unter `pdflatex` sofort ab – mit einer Meldung wie „Fontspec error: The fontspec package requires either XeTeX or LuaTeX“, die immerhin sagt, was zu tun ist.

Wer die Konfiguration aus einem anderen Projekt übernimmt, muss deshalb zuerst hier nachsehen. Die Voreinstellung von *LaTeX Workshop* selbst ist `pdflatex`, und der Fehler tritt dann in der ersten Minute auf.

11.1.8 Das dritte Rezept: make

Ein *Makefile* ist eine Textdatei mit dem Namen `Makefile`, die neben dem Dokument liegt und die Arbeitsschritte auflistet, aus denen ein fertiges PDF entsteht – im einfachsten Fall genau die fünf Läufe von oben. Das Programm `make` liest diese Datei und arbeitet die Schritte ab. Es stammt aus der Softwareentwicklung und ist auf macOS und Linux vorhanden; unter Windows muss man es nachinstallieren.

Makro	Logo
`\nxtzeichen`	nur das Zeichen
`\nxtlogo`	Zeichen und NXT („kurz“)
`\nxtlogonut`	mit „Nachhaltigkeit und Technologie“
`\nxthsrtlogo`	dazu „Reutlingen University“
`\nxtlogostandard`	Zeilen oben, NXT unten

Tabelle 11.1: Die Logo-Makros aus nxtlatexlogos.

Das braucht nicht jeder. Wer im Editor auf den Bau-Knopf drückt oder `latexmk` nutzt (Abschnitt 11.5.1), kommt ohne `Makefile` aus – der Knopf tut dasselbe. Nützlich wird es erst, wenn zum Setzen mehr gehört als LaTeX-Läufe. Genau das ist bei der Beispielthesis der Fall.

Das Rezept „nxtthesis: make“ ruft nur `make` auf, ohne Argument, im Verzeichnis des Dokuments. Was dann passiert, entscheidet das `Makefile`, das dort liegt.

Für die Anleitung ist das nichts Besonderes: Ihr `Makefile` arbeitet dieselben fünf Schritte ab wie das erste Rezept. Bei der *Beispielthesis* ist es etwas anderes. Dort heißt `make` „erst rechnen, dann setzen“: Eine Reihe von Python-Skripten erzeugt die Probleminstanz, löst sie mit *GLPK* und mit dem Verfahren aus der Arbeit, schreibt die Tabellen als `.tex`-Dateien und zeichnet die Abbildungen – und erst danach läuft `lualatex`. Kapitel 5 beschreibt den Gedanken dahinter: Zahlen und Abbildungen werden nicht von Hand übertragen, sondern erzeugt.

Wann braucht man das? Immer dann, wenn sich an den Daten oder an der Rechnung etwas geändert hat. Wer nur am Text schreibt, nimmt das erste Rezept und spart die Rechenzeit; wer einen Parameter des Verfahrens verstellt hat, muss über `make` gehen, sonst setzt LaTeX die alten Zahlen. Im Zweifel ist `make` die sichere Wahl – es baut nur neu, was neu zu bauen ist.

11.2 DIE LOGOS

Die Logos der Fakultät stecken im *NXT Logo Font*, der dem Paket beiliegt. Sie sind damit immer da: Herunterladen muss man nichts, und die Titelseite zeigt das Logo, sobald die Option `logo` oder `richtlinie` gesetzt ist. Die Formen stammen unverändert aus den Vorlagen des Corporate Design.

11.2.1 DIE MAKROS

Das Paket nxtlatexlogos – die Klasse lädt es selbst – stellt fünf Makros bereit:

Ohne Angabe ist das Zeichen 1 cm hoch. `width` oder `height` bestimmen die Größe, `schema` das Farbschema des Corporate Design: `farbig` (die Vorgabe), `schwarz`, `weiss`, `weisseTextmarke` und `weisseZeile` für dunkle Hintergründe.

```
\nxtlogo[width=4cm]
\nxthsrtlogo[height=1cm, schema=schwarz]
```

```
\colorbox{black}{\nxtlogonut[height=1cm, schema=weisseZeile]}
```

Listing 11.5: Logos einbinden

Für den Druck in anderen Farben – etwa in CMYK – lassen sich die Farben auch einzeln setzen: `zeichenfarbe`, `nxtfarbe` und `zeilenfarbe`, oder `farbe` für alle drei.

Das Logo der Titelseite (`\nxthsrtlogo`) muss man *nicht* selbst einbinden – die Klasse setzt es dort von sich aus. Von Hand braucht man die Makros nur, wenn ein Logo an einer eigenen Stelle stehen soll; auch das Beamer-Thema bringt seine Logos selbst mit (Abschnitt 10.5).

11.2.2 HOCHSCHULLOGO UND SKYLINE

Zwei Grafiken sind nicht im Font: das Logo der Hochschule, das das Landeswappen enthält, und die Reutlinger Skyline aus dem Brieffuß. Sie gehören der Hochschule und liegen dem Paket deshalb nicht bei. Gebraucht werden sie nur für Brief und Folien; für eine Abschlussarbeit spielen sie keine Rolle.

Zwei beiliegende Skripte laden sie von der Hochschul-Website und installieren sie in den lokalen *texmf*-Baum. In TeX Live liegen die Skripte bei der Dokumentation des Pakets, im Verzeichnis `doc/latex/nxtlatex` des Baums, den `kpsewhich -var-value TEXMFDIST` nennt:

```
sh getnxtlogos                                          # macOS, Linux

powershell -ExecutionPolicy Bypass -File getnxtlogos.ps1  # Windows
```

Listing 11.6: Hochschullogo und Skyline installieren

Anschließend rufen sie `mktexlsr` auf – TEX sucht Dateien nicht bei jedem Lauf im Dateisystem, sondern in einer Datenbank, die aktualisiert werden muss. Unter MiKTeX heißt das Programm `initexmf --update-fndb`; das Windows-Skript erkennt selbst, welches gebraucht wird. Fehlen die Dateien, übergehen `\hsrtlogograu` und `\hsrtskyline` sie stillschweigend.

11.3 BARRIEREFREIE PDFS

Ein barrierefreies PDF lässt sich mit einem Bildschirmleser lesen: Es kennt seine Sprache, seinen Titel und seine Struktur – was Überschrift ist, was Tabelle, was Abbildung. Die Klasse setzt Sprache, Titel und Autor selbst. Die Struktur schreibt LATEX ins PDF, wenn die erste Zeile der Hauptdatei, noch vor `\documentclass`, lautet:

```
\DocumentMetadata{lang=de, pdfversion=2.0,
                  pdfstandard=ua-2, tagging=on}
\documentclass[german,stix]{nxtthesis}
```

Listing 11.7: Ein getaggtes, barrierefreies PDF

Jede Abbildung braucht dann einen Alternativtext, der sagt, was sie zeigt. Er steht im Schlüssel `alt` von `\includegraphics`, etwa `alt={Balkendiagramm der Auslastung je Person}`; die Beispielthesis macht es vor – sie ist selbst so gesetzt. Quelltext mit listings geht

ebenfalls: Das Paket verträgt sich von sich aus nicht mit dem Tagging, die Klasse lädt aber die Anpassung, die das LaTeX-Projekt dafür bereitstellt. Ob das Ergebnis den Standard erfüllt, prüft das Programm *veraPDF*.

Bei der Sprache genügt `lang=de`. Mit `de-DE` warnt biblatex, weil es den Sprachnamen nicht kennt, den babel daraus macht.

Ohne Tagging, nur für die Archivierung, genügt `pdfstandard=a-2b` – ein PDF/A.

11.4 Andere Editoren

VSC ist eine Empfehlung, keine Vorschrift. Die Klasse ist ein gewöhnliches LaTeX-Paket und funktioniert überall.

TeXstudio ist plattformübergreifend, kostenlos und bringt einen PDF-Betrachter mit eingebautem *SyncTeX* mit. Für den Einstieg ist es die bequemste Wahl, weil man nichts installieren muss außer dem Programm selbst. Der Preis: Die Voreinstellung ist `pdflatex`, und der volle Ablauf mit `biber` und `makeindex` ist in den Einstellungen von Hand einzurichten („Optionen“ → „TeXstudio konfigurieren“ → „Erzeugen“). Wer das einmal gemacht hat, hat dieselbe Bequemlichkeit wie in VSC.

TeXShop liegt MacTeX bei und ist damit auf macOS schon da. Schlank, schnell, mit gutem Betrachter. Auch hier ist die Maschine auf `lualatex` umzustellen.

Emacs mit *AUCTeX* und **Vim beziehungsweise Neovim mit *VimTeX*** sind die Wahl für alle, die diese Editoren ohnehin benutzen. Beide können den vollen Ablauf, beide beherrschen SyncTeX, beide sind mächtiger als alles bisher Genannte – und beide lohnen sich nicht, wenn man sie für die Thesis erst lernen muss. Eine Abschlussarbeit ist nicht der Moment, einen neuen Editor anzufangen.

Overleaf läuft im Browser. Kein Setup, keine Installation, und mehrere Leute können gleichzeitig an derselben Datei schreiben – zum Ausprobieren und für gemeinsames Schreiben ist das schwer zu schlagen. Drei Einschränkungen gehören aber dazu:

- Die Klasse ist dort nicht installiert. `nxtthesis.cls` mit `nxtlatexcolors.sty`, `nxtlatexlogos.sty` und `NXT_Logo.otf` müssen ins Projekt hochgeladen werden, neben die `.tex`-Datei. Dasselbe gilt für das Beamer-Thema, wenn man Folien setzt.
- Der Compiler ist umzustellen: unter „Menu“ → „Compiler“ auf `LuaLaTeX`. Mit der Voreinstellung `pdfLaTeX` bricht der Lauf sofort ab.
- Die *Beispielthesis* lässt sich dort *nicht* vollständig bauen. Sie rechnet ihre Zahlen und zeichnet ihre Abbildungen mit Python und *GLPK* (Abschnitt 11.1.8), und beides gibt es auf Overleaf nicht. Man kann die erzeugten Abbildungen und Tabellen hochladen und das Dokument setzen – der Kreislauf aus Rechnen und Setzen, der den eigentlichen Punkt des Beispiels ausmacht, bleibt aber lokal.

Das ist keine Kritik an Overleaf, sondern eine Eigenschaft des Verfahrens: Wer rechnet, braucht eine Umgebung, in der er rechnen darf.

11.5 WERKZEUGE UM LaTeX HERUM

11.5.1 LATEXMK

latexmk beantwortet die Frage aus Abschnitt 11.1.2 anders. Statt fünf Läufe fest vorzuschreiben, sieht es nach, was sich geändert hat, und baut so oft, wie nötig ist – und ruft `biber` und `makeindex` von selbst auf, wenn es merkt, dass es sie braucht:

```
latexmk -lualatex ThesisanleitungNXT
latexmk -lualatex -pvc ThesisanleitungNXT   # baut bei jeder Aenderung neu
latexmk -C ThesisanleitungNXT               # alles Erzeugte entfernen
```

Listing 11.8: Der gesamte Ablauf in einem Aufruf

Das ist eleganter als eine feste Zahl von Durchgängen und spart Zeit, wenn nur eine Kleinigkeit geändert wurde. Die Klasse arbeitet mit beidem; das Paket setzt auf `Makefile`, weil bei der Beispielthesis ohnehin mehr zu tun ist als nur LaTeX aufzurufen (Abschnitt 11.1.8), und weil ein `Makefile` liest, was es tut. Wer nur setzt und nicht rechnet, ist mit `latexmk` gut bedient. *LaTeX Workshop* kann es übrigens auch: Ein viertes Rezept mit einem Werkzeug `latexmk` ist in `settings.json` schnell ergänzt.

11.5.2 LITERATURVERWALTUNG

Die `.bib`-Datei aus Kapitel 7 schreibt man nicht von Hand. *Zotero* sammelt Quellen aus dem Browser mit einem Klick – mitsamt DOI, Verlag und Jahr – und exportiert sie als `.bib`. Die Erweiterung *Better BibTeX* macht daraus eine Datei, die sich automatisch aktualisiert und stabile Schlüssel vergibt, sodass `\cite{mustermann2024}` nicht beim nächsten Export kaputtgeht. *JabRef* ist die Alternative für alle, die ihre Literatur lieber selbst in einer Datei verwalten; es bearbeitet die `.bib` unmittelbar.

Was beide nicht abnehmen: Die Einträge prüfen. Was aus einem Verlagsportal kommt, ist erstaunlich oft unvollständig oder falsch geschrieben, und APA zeigt jeden Fehler.

11.5.3 PRÜFPROGRAMME

chktex und *lacheck* lesen die Quelldatei und melden, was typischerweise schiefgeht: fehlende geschweifte Klammern nach einem Befehl, ein Leerzeichen zu viel vor der Fußnote, Anführungszeichen als `"`, ein `$` ohne Partner. Beide liegen TeX Live bei:

```
chktex  ThesisanleitungNXT.tex
lacheck ThesisanleitungNXT.tex
```

Listing 11.9: Die Quelle prüfen

Sie melden auch Dinge, die keine Fehler sind – man lernt schnell, was man ignorieren kann. Der eine echte Fund pro Kapitel rechtfertigt den Aufruf.

11.5.4 TEXDOC

texdoc ist das nützlichste Programm dieses Kapitels und das unbekannteste. Jedes Paket in TeX Live bringt seine Dokumentation mit, und `texdoc` öffnet sie:

```
texdoc acro          # das Paket hinter dem Abkuerzungsverzeichnis
texdoc fancyhdr      # Kolumnentitel
texdoc biblatex      # alles zum Zitieren
texdoc siunitx       # Zahlen und Einheiten
texdoc listings      # Quelltext setzen
```

Listing 11.10: Paketdokumentation lesen

Das öffnet das PDF, das die Autoren des Pakets geschrieben haben – vollständig, richtig und aktuell zu der Fassung, die installiert ist. Der Gewinn gegenüber der Suche im Netz ist genau das: Was dort steht, gilt für Ihre Installation, und nicht für eine Fassung von 2011.

11.5.5 VERSIONSVERWALTUNG MIT GIT

Abschnitt 1.5 hat es empfohlen; hier der Grund. Eine Abschlussarbeit läuft über Monate. In diesen Monaten wird ein Kapitel umgestellt, ein Abschnitt gestrichen und zwei Wochen später vermisst, und die Betreuung fragt nach der Fassung von vorletzter Woche. Ohne *Versionsverwaltung* entsteht daraus der Ordner mit `thesis_final.tex`, `thesis_final_v2.tex` und `thesis_final_v3_wirklich_final.tex` – und niemand weiß mehr, worin die sich unterscheiden.

Mit *git* beantwortet man das mit einem Befehl. Die Quelldateien sind reiner Text, und genau dafür ist git gemacht: Es zeigt, welche Zeile sich geändert hat, wann und warum. Ein `git commit` am Ende jedes Arbeitstages mit einem Satz dazu genügt vollauf; die Feinheiten von Zweigen und Rebase braucht eine Thesis nicht.

Wichtig ist, was *nicht* ins Repository gehört: alles, was beim Bauen entsteht. Das erzeugte PDF, die `.aux`-, `.log`-, `.toc`-, `.bbl`-Dateien und der ganze übrige Hilfsdateienkram ändern sich bei jedem Lauf und machen jeden Vergleich unlesbar. Sie lassen sich jederzeit neu erzeugen – das ist der Sinn der Sache. Eine Datei `.gitignore` hält sie draußen:

```
*.aux
*.log
*.toc
*.lof
*.lot
*.out
*.idx
*.ind
*.ilg
*.bbl
*.bcf
*.blg
*.run.xml
*.synctex.gz
*.pdf
```

Listing 11.11: Eine `.gitignore` für eine Thesis

Die letzte Zeile ist die umstrittene. Für das erzeugte PDF ist sie richtig. Nur: Abbildungen, die man von Hand eingebunden hat, sind auch PDF-Dateien und müssen ins Repository – die kann git nicht neu erzeugen. Entweder nimmt man sie mit `!Abbildungen/*.pdf` wieder

hinein, oder man schreibt statt `*.pdf` genauer nur den Namen des Hauptdokuments, etwa `/MeineArbeit.pdf`. Was erzeugt wird und was Quelle ist, muss man ohnehin auseinanderhalten (Kapitel 5).

11.6 WENN ES KLEMMT

11.6.1 DIE LOGDATEI LESEN

Wenn der Lauf abbricht, steht der Grund in der Datei `.log`. Die Regel dazu ist einfach und wird fast immer verletzt: **Die erste Fehlermeldung zählt, nicht die letzte.**

Der Grund ist die Arbeitsweise von LaTeX. Ein Fehler bringt das Programm aus dem Tritt, und alles Folgende ist meist Folge, nicht Ursache. Eine fehlende schließende Klammer in Zeile 40 kann in Zeile 300 eine Meldung über eine Umgebung erzeugen, die dort völlig in Ordnung ist. Wer die letzte Meldung liest, sucht am falschen Ort. Scrollen Sie nach oben, bis zur ersten Zeile, die mit `!` beginnt, und fangen Sie dort an.

`-file-line-error` aus Abschnitt 11.1.3 hilft dabei: Es setzt Datei und Zeilennummer vor die Meldung, sodass man weiß, wo „dort“ ist.

11.6.2 UNDEFINED CONTROL SEQUENCE

Die häufigste Meldung überhaupt:

```
! Undefined control sequence.
l.42 \includegraphcis
                      [width=\textwidth]{bild.pdf}
```

Listing 11.12: Die Meldung, die jeder kennt

Sie heißt: LaTeX kennt diesen Befehl nicht. Dafür gibt es genau zwei Gründe. Entweder ist der Name vertippt – im Beispiel steht `includegraphcis` statt `includegraphics` –, oder das Paket, das den Befehl definiert, ist nicht geladen. Die Zeile unter der Meldung zeigt, wo LaTeX stehen geblieben ist; der Befehl davor ist der schuldige.

Wer sich sicher ist, richtig getippt zu haben, prüft das Paket. Ein `\SI`, dessen siunitx niemand geladen hat, ist derselbe Fehler wie ein Tippfehler, nur ärgerlicher. Welches Paket einen Befehl mitbringt, sagt `texdoc` (Abschnitt 11.5.4).

11.6.3 AUFRÄUMEN

Bleibt eine Meldung unverständlich, oder passt sie nicht zu dem, was in der Quelle steht – eine Kapitelnummer, die es nicht gibt, ein Literaturverzeichnis, das eine gelöschte Quelle zeigt –, dann sind veraltete Hilfsdateien die wahrscheinlichste Ursache. Die `.aux`-Datei stammt noch vom letzten Lauf und beschreibt ein Dokument, das es nicht mehr gibt.

Die Behandlung ist immer dieselbe:

```
make clean      # alle Hilfsdateien entfernen
make            # vollstaendig neu bauen
```

Listing 11.13: Der erste Griff bei rätselhaften Fehlern

Das kostet ein paar Minuten und löst überraschend viele Fälle. Wer ohne `Makefile` arbeitet, nimmt `latexmk -C`; in VSC erledigt die Einstellung `autoClean.run` das nach jedem Bau von selbst, weshalb dieser Fehler dort seltener auftritt.

11.6.4 Das Minimalbeispiel

Bleibt der Fehler, hilft die älteste Methode: einkreisen. Legen Sie eine neue Datei an, mit derselben Klasse und denselben Paketen, aber nur der einen Stelle, die nicht funktioniert:

```
\documentclass[german,stix]{nxtthesis}
\usepackage{siunitx}
\begin{document}
Hier steht nur die eine Stelle, die klemmt:
\SI{12,4}{\second}
\end{document}
```

Listing 11.14: Ein Minimalbeispiel

Passiert der Fehler auch hier, haben Sie ihn auf zehn Zeilen eingegrenzt. Passiert er nicht, liegt er woanders – dann fügen Sie schrittweise wieder hinzu, bis er zurückkommt. Der letzte Zusatz ist die Ursache.

Die Methode wirkt umständlich und ist die schnellste, sobald man eine Viertelstunde erfolglos gestarrt hat. Sie hat einen zweiten Nutzen: Wenn Sie danach doch fragen müssen – die Betreuung, einen Kommilitonen, ein Forum –, haben Sie zehn Zeilen zum Zeigen statt zweihundert Seiten. Die allermeisten Fragen beantworten sich beim Bauen des Minimalbeispiels von selbst.

Literatur

Hoekwater, T., Roux, E., & Gesang, P. (2024). *luamplib: Use LuaTeX's MetaPost Library*. Verfügbar 21. April 2026 unter https://ctan.org/pkg/luamplib

Knuth, D. E. (1986). *The TeXbook*. Addison-Wesley.

Lamport, L. (1994). *LaTeX: A Document Preparation System: User's Guide and Reference Manual* (2. Aufl.). Addison-Wesley.

Lehman, P., Kime, P., & Wemheuer, M. (2024). *The biblatex Package: Programmable Bibliographies and Citations*. Verfügbar 21. April 2026 unter https://ctan.org/pkg/biblatex

Mittelbach, F., & Fischer, U. (2023). *The LaTeX Companion: Tools and Techniques for Computer Typesetting* (3. Aufl.). Addison-Wesley.

Reichenberger, V. (2026). *nxtlatex: LaTeX-Klassen und -Themen für die Fakultät NXT*. Verfügbar 21. April 2026 unter https://ctan.org/pkg/nxtlatex

Tantau, T. (2024). *TikZ and PGF: Manual for Version 3.1.10*. Verfügbar 21. April 2026 unter https://ctan.org/pkg/pgf

Tantau, T., Wright, J., & Miletić, V. (2024). *The beamer class: User Guide*. Verfügbar 21. April 2026 unter https://ctan.org/pkg/beamer

Wright, J. (2024). *siunitx: A comprehensive (SI) units package*. Verfügbar 21. April 2026 unter https://ctan.org/pkg/siunitx

A Schriftmuster im Vergleich

Die Klasse bringt zwölf Schriftoptionen mit. Welche zu einer Arbeit passt, lässt sich nicht an einer Aufzählung von Namen entscheiden, sondern nur am gesetzten Ergebnis. Dieser Anhang zeigt für jede Option dieselben drei Seiten: die Titelseite, eine Seite mit Grundtext, Kapitälchen, Mathematik und einer Tabelle sowie eine dicht gesetzte Textseite mit Gliederung. Erst die dritte Seite verrät den *Grauwert* einer Schrift – also den Eindruck, den eine Seite der fertigen Arbeit machen wird.

Die Muster sind verkleinert wiedergegeben und auf den Satzspiegel beschnitten; zum Beurteilen der Lesbarkeit gehört ein Blick auf eine Seite in Originalgröße. Für den Vergleich untereinander genügt der Ausschnitt: Worauf es ankommt, ist der Grauwert der Textfläche, nicht der Papierrand.

A.1 Die Schriftoptionen

Die Muster stehen in alphabetischer Reihenfolge, nicht nach Eignung – diese hängt von der Arbeit ab. Drei Fragen helfen beim Vergleich: Wie ruhig läuft der Grauwert auf der dritten Seite? Passt die Mathematik zum Text, oder fällt jede Formel auf? Und reicht die Mittellänge bei 11 pt noch aus?

Fünf der zwölf Schriften bringen einen Semibold-Schnitt mit (`libertinus`, `segoeui`, `stix`, `fira` und `garamond`); bei ihnen ersetzt er den fetten. Die übrigen sieben werden fett gesetzt. Echte Kapitälchen haben alle bis auf `times`.

Optimierung des Mitarbeitereinsatzes

Vergleichende Analyse in der metallverarbeitenden Industrie

Thesis zur Erlangung des akademischen Grades

Bachelor of Science

an der Fakultät NXT Nachhaltigkeit und Technologie der Hochschule Reutlingen

im Studiengang

Nachhaltige Technologie

Vorgelegt von:
Jean Dujardin
Wegstraße 123, 72764 Reutlingen
Matrikelnummer 1234567

Abgabedatum: 24. Dezember 2026

Erstprüfer: Prof. Dr. Katharina Erstmal
Zweitprüfer: Prof. Dr. Kornelius Hurtig

1.3 Der Grauwert im Fließsatz

Eine Schrift lässt sich nicht an einem Wort beurteilen und auch nicht an einer Zeile. Erst über einen ganzen Absatz hinweg zeigt sich, welchen Grauwert sie erzeugt, wie ruhig die Zeilen laufen und ob die Wortabstände gleichmäßig bleiben. Diese Seite ist deshalb absichtlich dicht gesetzt: Sie soll den Eindruck vermitteln, den eine Seite der fertigen Arbeit machen wird, und nicht den einer Schriftmustertafel.

Der Grauwert entsteht aus dem Zusammenspiel von Strichstärke, x-Höhe, Laufweite und Zeilenabstand. Eine Schrift mit großer x-Höhe wirkt bei gleichem Schriftgrad größer und braucht mehr Zeilenabstand, sonst verkleben die Zeilen. Eine Schrift mit geringer Strichstärke wirkt heller und verträgt engere Zeilen. Beides lässt sich nur im Satz beurteilen, nicht an einer Tabelle von Buchstaben.

1.3.1 Serifen und ihre Wirkung

Serifen sind die kleinen Abschlüsse an den Enden der Buchstaben. Ihnen wird nachgesagt, sie führten das Auge über die Zeile und erhöhten damit die Lesbarkeit im Druck. Ob das messbar stimmt, ist umstritten; sicher ist, dass Serifenschriften im Fließsatz seit Jahrhunderten den Standard bilden und dass Leserinnen und Leser sie deshalb erwarten. In den Ingenieurwissenschaften sind serifenlose Schriften verbreiteter, teils aus Gewohnheit, teils weil sie in technischen Zeichnungen und auf Bildschirmen besser bestehen.

Für eine Abschlussarbeit gilt: Beides ist zulässig, und die Wahl sollte begründet und dann durchgehalten werden. Ein Wechsel mitten im Dokument fällt auf, auch wenn er nicht bewusst wahrgenommen wird. Die Richtlinie der Fakultät schreibt deshalb ausdrücklich eine einzige Schriftart für das ganze Dokument vor, mit Ausnahme der Formeln.

1.3.2 Die Schreibmaschinenschrift

Eine dritte Schrift kommt ins Spiel, sobald Dateinamen, Befehle oder Quelltext vorkommen. Sie wird mit `\texttt{...}` gesetzt und sollte zur Textschrift passen, ohne mit ihr zu verschmelzen: Der Aufruf `lualatex Beispielthesis` muss auf den ersten Blick als Eingabe erkennbar sein, und in `Code/daten.py` darf man die Null in `p_i0 = 0.85` nicht mit dem Buchstaben `O` verwechseln. Genau daran scheitern viele Schriften. Prüfen lässt es sich an `Il1| O0` – großes I, kleines l, Eins, senkrechter Strich, großes O, Null. Wo diese Zeichen ununterscheidbar sind, taugt die Schrift nicht für Quelltext.

1 Schriftmuster

Dies ist ein Absatz im Grundtext. Er enthält Umlaute (äöüÄÖÜß), eine *Hervorhebung*, eine **fette Auszeichnung** und `Schreibmaschinenschrift`. Zahlen im Text: 1234567890. Ligaturen: Auffinden, Schifffahrt, Waffel, fünf.

Abkürzungen setzt die Klasse in Kapitälchen, sofern die Schrift echte mitbringt: Mixed Integer Linear Programming (MILP) und Simulated Annealing (SA); im zweiten Gebrauch dann MILP. Fehlen echte Kapitälchen, bleibt es beim normalen Schnitt – aus Versalien heruntergerechnete Kapitälchen wirken zu leicht und fallen im Text unangenehm auf.

1.1 Mathematik

Im Text $C_{\max} = \max_i \sum_j p_{ij} x_{ij}$ und abgesetzt:

$$\mathscr{L}(\mathbf{x}) = \frac{1}{2m} \sum_{i=1}^{m} \left(f(\mathbf{u}_i; \mathbf{x}) - y_i \right)^2, \qquad \alpha \le \frac{2}{\lambda_{\max}}. \tag{1.1}$$

1.2 Tabellensatz

Verfahren	Makespan	Zeit
Greedy	266,67	0,0
LPT	265,38	0,0
MILP	236,20	640,6

Tabelle 1.1: Zahlen im Tabellensatz

`bookman` – TeX Gyre Bonum, die freie Bookman. Kräftig und weit laufend, dadurch sehr gut lesbar, aber platzhungrig: Eine Arbeit wird damit spürbar länger als mit einer Garamond. Serifenlose ist Heros aus derselben URW-Familie. Kein Semibold. Eigene Mathematikschrift.

Abbildung A.1: Schriftmuster `nxtthesis-bookman`: Titelseite, Schriftmuster und dicht gesetzte Textseite.

Optimierung des Mitarbeitereinsatzes

Vergleichende Analyse in der metallverarbeitenden Industrie

Thesis zur Erlangung des akademischen Grades

Bachelor of Science

an der Fakultät NXT Nachhaltigkeit und Technologie
der Hochschule Reutlingen

im Studiengang

Nachhaltige Technologie

Vorgelegt von:
Jean Dujardin
Wegstraße 123, 72764 Reutlingen
Matrikelnummer 1234567

Abgabedatum: 24. Dezember 2026

Erstprüfer: Prof. Dr. Katharina Erstmal
Zweitprüfer: Prof. Dr. Kornelius Hurtig

1.3 Der Grauwert im Fließsatz

Eine Schrift lässt sich nicht an einem Wort beurteilen und auch nicht an einer Zeile. Erst über einen ganzen Absatz hinweg zeigt sich, welchen Grauwert sie erzeugt, wie ruhig die Zeilen laufen und ob die Wortabstände gleichmäßig bleiben. Diese Seite ist deshalb absichtlich dicht gesetzt: Sie soll den Eindruck vermitteln, den eine Seite der fertigen Arbeit machen wird, und nicht den einer Schriftmustertafel.

Der Grauwert entsteht aus dem Zusammenspiel von Strichstärke, x-Höhe, Laufweite und Zeilenabstand. Eine Schrift mit großer x-Höhe wirkt bei gleichem Schriftgrad größer und braucht mehr Zeilenabstand, sonst verkleben die Zeilen. Eine Schrift mit geringer Strichstärke wirkt heller und verträgt engere Zeilen. Beides lässt sich nur im Satz beurteilen, nicht an einer Tabelle von Buchstaben.

1.3.1 Serifen und ihre Wirkung

Serifen sind die kleinen Abschlüsse an den Enden der Buchstaben. Ihnen wird nachgesagt, sie führten das Auge über die Zeile und erhöhten damit die Lesbarkeit im Druck. Ob das messbar stimmt, ist umstritten; sicher ist, dass Serifenschriften im Fließsatz seit Jahrhunderten den Standard bilden und dass Leserinnen und Leser sie deshalb erwarten. In den Ingenieurwissenschaften sind serifenlose Schriften verbreiteter, teils aus Gewohnheit, teils weil sie in technischen Zeichnungen und auf Bildschirmen besser bestehen.

Für eine Abschlussarbeit gilt: Beides ist zulässig, und die Wahl sollte begründet und dann durchgehalten werden. Ein Wechsel mitten im Dokument fällt auf, auch wenn er nicht bewusst wahrgenommen wird. Die Richtlinie der Fakultät schreibt deshalb ausdrücklich eine einzige Schriftart für das ganze Dokument vor, mit Ausnahme der Formeln.

1.3.2 Die Schreibmaschinenschrift

Eine dritte Schrift kommt ins Spiel, sobald Dateinamen, Befehle oder Quelltext vorkommen. Sie wird mit `\texttt{...}` gesetzt und sollte zur Textschrift passen, ohne mit ihr zu verschmelzen: Der Aufruf `lualatex BeispielthesisNXT` muss auf den ersten Blick als Eingabe erkennbar sein, und in `Code/daten.py` darf man die Null in `p_i0 = 0.85` nicht mit dem Buchstaben `O` verwechseln. Genau daran scheitern viele Schriften. Prüfen lässt es sich an `Il1| O0` – großes I, kleines l, Eins, senkrechter Strich, großes O, Null. Wo diese Zeichen ununterscheidbar sind, taugt die Schrift nicht für Quelltext.

1.3.3 Mathematik und Text

Wo Formeln im Fließtext stehen, entscheidet sich, ob eine Schrift für eine technische Arbeit taugt. Die Mathematikschrift muss zur Textschrift passen: Ein Wert wie $\alpha = 0{,}05$ oder ein Symbol wie $C_{\max}$ steht mitten im Satz und darf dort weder größer noch kleiner noch dunkler wirken als seine Umgebung. Schriftfamilien, die eine eigene Mathematik-

1 Schriftmuster

Dies ist ein Absatz im Grundtext. Er enthält Umlaute (äöüÄÖÜß), eine *Hervorhebung*, eine **fette Auszeichnung** und `Schreibmaschinenschrift`. Zahlen im Text: 1234567890. Ligaturen: Auffinden, Schifffahrt, Waffel, fünf.

Abkürzungen setzt die Klasse in Kapitälchen, sofern die Schrift echte mitbringt: Mixed Integer Linear Programming (MILP) und Simulated Annealing (SA); im zweiten Gebrauch dann MILP. Fehlen echte Kapitälchen, bleibt es beim normalen Schnitt – aus Versalien heruntergerechnete Kapitälchen wirken zu leicht und fallen im Text unangenehm auf.

1.1 Mathematik

Im Text $C_{\max} = \max_i \sum_j p_{ij} x_{ij}$ und abgesetzt:

$$\mathscr{L}(\mathbf{x}) = \frac{1}{2m} \sum_{i=1}^{m} \bigl(f(\mathbf{u}_i; \mathbf{x}) - y_i\bigr)^2, \qquad \alpha \le \frac{2}{\lambda_{\max}}. \tag{1.1}$$

1.2 Tabellensatz

Verfahren	Makespan	Zeit
Greedy	266,67	0,0
LPT	265,38	0,0
MILP	236,20	640,6

Tabelle 1.1: Zahlen im Tabellensatz

`charter` – XCharter, die freie TeX-Fassung von Matthew Carters Charter. Für den Druck bei geringer Auflösung entworfen: kräftige Striche, große Mittellänge, sehr robust. Kein Semibold. Eigene Mathematikschrift.

Abbildung A.2: Schriftmuster `nxtthesis-charter`: Titelseite, Schriftmuster und dicht gesetzte Textseite.

Optimierung des Mitarbeitereinsatzes

Vergleichende Analyse in der metallverarbeitenden Industrie

Thesis zur Erlangung des akademischen Grades

Bachelor of Science

an der Fakultät NXT Nachhaltigkeit und Technologie
der Hochschule Reutlingen

im Studiengang

Nachhaltige Technologie

Vorgelegt von:
Jean Dujardin
Wegstraße 123, 72764 Reutlingen
Matrikelnummer 1234567

Abgabedatum: 24. Dezember 2026

Erstprüfer: Prof. Dr. Katharina Erstmal
Zweitprüfer: Prof. Dr. Kornelius Hurtig

1 Schriftmuster 2

1.3 Der Grauwert im Fließsatz

Eine Schrift lässt sich nicht an einem Wort beurteilen und auch nicht an einer Zeile. Erst über einen ganzen Absatz hinweg zeigt sich, welchen Grauwert sie erzeugt, wie ruhig die Zeilen laufen und ob die Wortabstände gleichmäßig bleiben. Diese Seite ist deshalb absichtlich dicht gesetzt: Sie soll den Eindruck vermitteln, den eine Seite der fertigen Arbeit machen wird, und nicht den einer Schriftmustertafel.

Der Grauwert entsteht aus dem Zusammenspiel von Strichstärke, x-Höhe, Laufweite und Zeilenabstand. Eine Schrift mit großer x-Höhe wirkt bei gleichem Schriftgrad größer und braucht mehr Zeilenabstand, sonst verkleben die Zeilen. Eine Schrift mit geringer Strichstärke wirkt heller und verträgt engere Zeilen. Beides lässt sich nur im Satz beurteilen, nicht an einer Tabelle von Buchstaben.

1.3.1 Serifen und ihre Wirkung

Serifen sind die kleinen Abschlüsse an den Enden der Buchstaben. Ihnen wird nachgesagt, sie führten das Auge über die Zeile und erhöhten damit die Lesbarkeit im Druck. Ob das messbar stimmt, ist umstritten; sicher ist, dass Serifenschriften im Fließsatz seit Jahrhunderten den Standard bilden und dass Leserinnen und Leser sie deshalb erwarten. In den Ingenieurwissenschaften sind serifenlose Schriften verbreiteter, teils aus Gewohnheit, teils weil sie in technischen Zeichnungen und auf Bildschirmen besser bestehen.

Für eine Abschlussarbeit gilt: Beides ist zulässig, und die Wahl sollte begründet und dann durchgehalten werden. Ein Wechsel mitten im Dokument fällt auf, auch wenn er nicht bewusst wahrgenommen wird. Die Richtlinie der Fakultät schreibt deshalb ausdrücklich eine einzige Schriftart für das ganze Dokument vor, mit Ausnahme der Formeln.

1.3.2 Die Schreibmaschinenschrift

Eine dritte Schrift kommt ins Spiel, sobald Dateinamen, Befehle oder Quelltext vorkommen. Sie wird mit `\texttt{…}` gesetzt und sollte zur Textschrift passen, ohne mit ihr zu verschmelzen: Der Aufruf `lualatex BeispielthesisNXT` muss auf den ersten Blick als Eingabe erkennbar sein, und in `Code/daten.py` darf man die Null in `p_10 = 0.85` nicht mit dem Buchstaben O verwechseln. Genau daran scheitern viele Schriften. Prüfen lässt es sich an `Il1| O0` – großes I, kleines l, Eins, senkrechter Strich, großes O, Null. Wo diese Zeichen ununterscheidbar sind, taugt die Schrift nicht für Quelltext.

1.3.3 Mathematik und Text

Wo Formeln im Fließtext stehen, entscheidet sich, ob eine Schrift für eine technische Arbeit taugt. Die Mathematikschrift muss zur Textschrift passen: Ein Wert wie $\alpha = 0{,}05$ oder ein Symbol wie $C_{\max}$ steht mitten im Satz und darf dort weder größer noch kleiner noch dunkler wirken als seine Umgebung. Schriftfamilien, die eine eigene Mathematikschrift mitbringen, sind hier im Vorteil – STIX Two, Libertinus und Pagella tun das, und die Klasse bindet sie entsprechend ein.

1 Schriftmuster 1

1 Schriftmuster

Dies ist ein Absatz im Grundtext. Er enthält Umlaute (äöüÄÖÜß), eine *Hervorhebung*, eine **fette Auszeichnung** und `Schreibmaschinenschrift`. Zahlen im Text: 1234567890. Ligaturen: Auffinden, Schifffahrt, Waffel, fünf.

Abkürzungen setzt die Klasse in Kapitälchen, sofern die Schrift echte mitbringt: Mixed Integer Linear Programming (MILP) und Simulated Annealing (SA); im zweiten Gebrauch dann MILP. Fehlen echte Kapitälchen, bleibt es beim normalen Schnitt – aus Versalien heruntergerechnete Kapitälchen wirken zu leicht und fallen im Text unangenehm auf.

1.1 Mathematik

Im Text $C_{\max} = \max_i \sum_j p_{ij} x_{ij}$ und abgesetzt:

$$\mathcal{L}(\mathbf{x}) = \frac{1}{2m} \sum_{i=1}^{m} \left(f(\mathbf{u}_i; \mathbf{x}) - y_i \right)^2, \qquad \alpha \leqslant \frac{2}{\lambda_{\max}}. \tag{1.1}$$

1.2 Tabellensatz

Verfahren	Makespan	Zeit
Greedy	266,67	0,0
LPT	265,38	0,0
MILP	236,20	640,6

Tabelle 1.1: Zahlen im Tabellensatz

`concrete` – CMU Concrete mit Euler, die Schriften aus Knuths *Concrete Mathematics*. Eine Egyptienne mit gleichmäßigen Strichstärken; Euler ist keine gewöhnliche Mathematikschrift, sondern wirkt wie handgeschrieben. Ein eigenwilliges Paar mit starkem Charakter – reizvoll, aber es fällt auf. Dazu CMU Bright und CMU Typewriter.

Abbildung A.3: Schriftmuster `nxtthesis-concrete`: Titelseite, Schriftmuster und dicht gesetzte Textseite.

Optimierung des Mitarbeitereinsatzes

Vergleichende Analyse in der metallverarbeitenden Industrie

Thesis zur Erlangung des akademischen Grades

Bachelor of Science

an der Fakultät NXT Nachhaltigkeit und Technologie
der Hochschule Reutlingen

im Studiengang

Nachhaltige Technologie

Vorgelegt von:
Jean Dujardin
Wegstraße 123, 72764 Reutlingen
Matrikelnummer 1234567

Abgabedatum: 24. Dezember 2026

Erstprüfer: Prof. Dr. Katharina Erstmal
Zweitprüfer: Prof. Dr. Kornelius Hurtig

1.3 Der Grauwert im Fließsatz

Eine Schrift lässt sich nicht an einem Wort beurteilen und auch nicht an einer Zeile. Erst über einen ganzen Absatz hinweg zeigt sich, welchen Grauwert sie erzeugt, wie ruhig die Zeilen laufen und ob die Wortabstände gleichmäßig bleiben. Diese Seite ist deshalb absichtlich dicht gesetzt: Sie soll den Eindruck vermitteln, den eine Seite der fertigen Arbeit machen wird, und nicht den einer Schriftmustertafel.

Der Grauwert entsteht aus dem Zusammenspiel von Strichstärke, x-Höhe, Laufweite und Zeilenabstand. Eine Schrift mit großer x-Höhe wirkt bei gleichem Schriftgrad größer und braucht mehr Zeilenabstand, sonst verkleben die Zeilen. Eine Schrift mit geringer Strichstärke wirkt heller und verträgt engere Zeilen. Beides lässt sich nur im Satz beurteilen, nicht an einer Tabelle von Buchstaben.

1.3.1 Serifen und ihre Wirkung

Serifen sind die kleinen Abschlüsse an den Enden der Buchstaben. Ihnen wird nachgesagt, sie führten das Auge über die Zeile und erhöhten damit die Lesbarkeit im Druck. Ob das messbar stimmt, ist umstritten; sicher ist, dass Serifenschriften im Fließsatz seit Jahrhunderten den Standard bilden und dass Leserinnen und Leser sie deshalb erwarten. In den Ingenieurwissenschaften sind serifenlose Schriften verbreiteter, teils aus Gewohnheit, teils weil sie in technischen Zeichnungen und auf Bildschirmen besser bestehen.

Für eine Abschlussarbeit gilt: Beides ist zulässig, und die Wahl sollte begründet und dann durchgehalten werden. Ein Wechsel mitten im Dokument fällt auf, auch wenn er nicht bewusst wahrgenommen wird. Die Richtlinie der Fakultät schreibt deshalb ausdrücklich eine einzige Schriftart für das ganze Dokument vor, mit Ausnahme der Formeln.

1.3.2 Die Schreibmaschinenschrift

Eine dritte Schrift kommt ins Spiel, sobald Dateinamen, Befehle oder Quelltext vorkommen. Sie wird mit `\texttt{…}` gesetzt und sollte zur Textschrift passen, ohne mit ihr zu verschmelzen: Der Aufruf `lualatex BeispielthesisNXT` muss auf den ersten Blick als Eingabe erkennbar sein, und in `Code/daten.py` darf man die Null in `p_i0 = 0.85` nicht mit dem Buchstaben `O` verwechseln. Genau daran scheitern viele Schriften. Prüfen lässt es sich an `Il1| O0` – großes I, kleines l, Eins, senkrechter Strich, großes O, Null. Wo diese Zeichen ununterscheidbar sind, taugt die Schrift nicht für Quelltext.

1.3.3 Mathematik und Text

Wo Formeln im Fließtext stehen, entscheidet sich, ob eine Schrift für eine technische Arbeit taugt. Die Mathematikschrift muss zur Textschrift passen: Ein Wert wie $\alpha = 0{,}05$

1 Schriftmuster

Dies ist ein Absatz im Grundtext. Er enthält Umlaute (äöüÄÖÜß), eine *Hervorhebung*, eine **fette Auszeichnung** und `Schreibmaschinenschrift`. Zahlen im Text: 1234567890. Ligaturen: Auffinden, Schifffahrt, Waffel, fünf.

Abkürzungen setzt die Klasse in Kapitälchen, sofern die Schrift echte mitbringt: Mixed Integer Linear Programming (MILP) und Simulated Annealing (SA); im zweiten Gebrauch dann MILP. Fehlen echte Kapitälchen, bleibt es beim normalen Schnitt – aus Versalien heruntergerechnete Kapitälchen wirken zu leicht und fallen im Text unangenehm auf.

1.1 Mathematik

Im Text $C_{\max} = \max_i \sum_j p_{ij} x_{ij}$ und abgesetzt:

$$L(\mathbf{x}) = \frac{1}{2m} \sum_{i=1}^{m} \left(f(\mathbf{u}_i; \mathbf{x}) - y_i \right)^2, \qquad \alpha \le \frac{2}{\lambda_{\max}}. \tag{1.1}$$

1.2 Tabellensatz

Verfahren	Makespan	Zeit
Greedy	266,67	0,0
LPT	265,38	0,0
MILP	236,20	640,6

Tabelle 1.1: Zahlen im Tabellensatz

`fira` – serifenlos, ursprünglich für kleine Bildschirme entworfen, dadurch sehr gut unterscheidbare Zeichen. Statt eines fetten Schnitts wird Fira Sans Medium verwendet. Fira Math als Mathematikschrift.

Abbildung A.4: Schriftmuster `nxtthesis-fira`: Titelseite, Schriftmuster und dicht gesetzte Textseite.

Optimierung des Mitarbeitereinsatzes

Vergleichende Analyse in der metallverarbeitenden Industrie

Thesis zur Erlangung des akademischen Grades

Bachelor of Science

an der Fakultät NXT Nachhaltigkeit und Technologie
der Hochschule Reutlingen

im Studiengang

Nachhaltige Technologie

Vorgelegt von:
Jean Dujardin
Wegstraße 123, 72764 Reutlingen
Matrikelnummer 1234567

Abgabedatum: 24. Dezember 2026

Erstprüfer: Prof. Dr. Katharina Erstmal
Zweitprüfer: Prof. Dr. Kornelius Hurtig

1.3 Der Grauwert im Fließsatz

Eine Schrift lässt sich nicht an einem Wort beurteilen und auch nicht an einer Zeile. Erst über einen ganzen Absatz hinweg zeigt sich, welchen Grauwert sie erzeugt, wie ruhig die Zeilen laufen und ob die Wortabstände gleichmäßig bleiben. Diese Seite ist deshalb absichtlich dicht gesetzt: Sie soll den Eindruck vermitteln, den eine Seite der fertigen Arbeit machen wird, und nicht den einer Schriftmustertafel.

Der Grauwert entsteht aus dem Zusammenspiel von Strichstärke, x-Höhe, Laufweite und Zeilenabstand. Eine Schrift mit großer x-Höhe wirkt bei gleichem Schriftgrad größer und braucht mehr Zeilenabstand, sonst verkleben die Zeilen. Eine Schrift mit geringer Strichstärke wirkt heller und verträgt engere Zeilen. Beides lässt sich nur im Satz beurteilen, nicht an einer Tabelle von Buchstaben.

1.3.1 Serifen und ihre Wirkung

Serifen sind die kleinen Abschlüsse an den Enden der Buchstaben. Ihnen wird nachgesagt, sie führten das Auge über die Zeile und erhöhten damit die Lesbarkeit im Druck. Ob das messbar stimmt, ist umstritten; sicher ist, dass Serifenschriften im Fließsatz seit Jahrhunderten den Standard bilden und dass Leserinnen und Leser sie deshalb erwarten. In den Ingenieurwissenschaften sind serifenlose Schriften verbreiteter, teils aus Gewohnheit, teils weil sie in technischen Zeichnungen und auf Bildschirmen besser bestehen.

Für eine Abschlussarbeit gilt: Beides ist zulässig, und die Wahl sollte begründet und dann durchgehalten werden. Ein Wechsel mitten im Dokument fällt auf, auch wenn er nicht bewusst wahrgenommen wird. Die Richtlinie der Fakultät schreibt deshalb ausdrücklich eine einzige Schriftart für das ganze Dokument vor, mit Ausnahme der Formeln.

1.3.2 Die Schreibmaschinenschrift

Eine dritte Schrift kommt ins Spiel, sobald Dateinamen, Befehle oder Quelltext vorkommen. Sie wird mit `\texttt{…}` gesetzt und sollte zur Textschrift passen, ohne mit ihr zu verschmelzen: Der Aufruf `lualatex Beispielthesis.nxt` muss auf den ersten Blick als Eingabe erkennbar sein, und in `Code/daten.py` darf man die Null in `p_10 = 0.85` nicht mit dem Buchstaben `O` verwechseln. Genau daran scheitern viele Schriften. Prüfen lässt es sich an `Il1| O0` – großes I, kleines l, Eins, senkrechter Strich, großes O, Null. Wo diese Zeichen ununterscheidbar sind, taugt die Schrift nicht für Quelltext.

1.3.3 Mathematik und Text

Wo Formeln im Fließtext stehen, entscheidet sich, ob eine Schrift für eine technische Arbeit taugt. Die Mathematikschrift muss zur Textschrift passen: Ein Wert wie $\alpha = 0{,}05$ oder ein Symbol wie $C_{\max}$ steht mitten im Satz und darf dort weder größer noch kleiner noch dunkler wirken als seine Umgebung. Schriftfamilien, die eine eigene Mathematikschrift mitbringen, sind hier im Vorteil – STIX Two, Libertinus und Pagella tun das, und die Klasse bindet sie entsprechend ein.

Fehlt eine passende Mathematikschrift, entsteht ein sichtbarer Bruch: Der Text ist in der einen Schrift gesetzt, jede Formel in einer anderen. Bei wenigen Formeln fällt das kaum auf, bei einer Arbeit über Optimierungsverfahren auf jeder zweiten Seite.

1 Schriftmuster

Dies ist ein Absatz im Grundtext. Er enthält Umlaute (äöüÄÖÜß), eine ***Hervorhebung***, eine **fette Auszeichnung** und `Schreibmaschinenschrift`. Zahlen im Text: 1234567890. Ligaturen: Auffinden, Schifffahrt, Waffel, fünf.

Abkürzungen setzt die Klasse in Kapitälchen, sofern die Schrift echte mitbringt: Mixed Integer Linear Programming (MILP) und Simulated Annealing (SA); im zweiten Gebrauch dann MILP. Fehlen echte Kapitälchen, bleibt es beim normalen Schnitt – aus Versalien heruntergerechnete Kapitälchen wirken zu leicht und fallen im Text unangenehm auf.

1.1 Mathematik

Im Text $C_{\max} = \max_i \sum_j p_{ij} x_{ij}$ und abgesetzt:

$$\mathcal{L}(\mathbf{x}) = \frac{1}{2m} \sum_{i=1}^{m} \left(f(\mathbf{u}_i; \mathbf{x}) - y_i \right)^2, \qquad \alpha \le \frac{2}{\lambda_{\max}}. \tag{1.1}$$

1.2 Tabellensatz

Verfahren	Makespan	Zeit
Greedy	266,67	0,0
LPT	265,38	0,0
MILP	236,20	640,6

Tabelle 1.1: Zahlen im Tabellensatz

`garamond` – EB Garamond, ein Nachschnitt der Garamond des 16. Jahrhunderts. Die leichteste Schrift im Feld: elegant, aber mit kleiner Mittellänge und deshalb bei 11 pt schon an der unteren Grenze. Statt des fetten Schnitts wird SemiBold verwendet. Eigene Mathematikschrift.

Abbildung A.5: Schriftmuster `nxtthesis-garamond`: Titelseite, Schriftmuster und dicht gesetzte Textseite.

Optimierung des Mitarbeitereinsatzes

Vergleichende Analyse in der metallverarbeitenden Industrie

Thesis zur Erlangung des akademischen Grades

Bachelor of Science

an der Fakultät NXT Nachhaltigkeit und Technologie
der Hochschule Reutlingen

im Studiengang

Nachhaltige Technologie

Vorgelegt von:
Jean Dujardin
Wegstraße 123, 72764 Reutlingen
Matrikelnummer 1234567

Abgabedatum: 24. Dezember 2026

Erstprüfer: Prof. Dr. Katharina Erstmal
Zweitprüfer: Prof. Dr. Kornelius Hurtig

1.3 Der Grauwert im Fließsatz

Eine Schrift lässt sich nicht an einem Wort beurteilen und auch nicht an einer Zeile. Erst über einen ganzen Absatz hinweg zeigt sich, welchen Grauwert sie erzeugt, wie ruhig die Zeilen laufen und ob die Wortabstände gleichmäßig bleiben. Diese Seite ist deshalb absichtlich dicht gesetzt: Sie soll den Eindruck vermitteln, den eine Seite der fertigen Arbeit machen wird, und nicht den einer Schriftmustertafel.

Der Grauwert entsteht aus dem Zusammenspiel von Strichstärke, x-Höhe, Laufweite und Zeilenabstand. Eine Schrift mit großer x-Höhe wirkt bei gleichem Schriftgrad größer und braucht mehr Zeilenabstand, sonst verkleben die Zeilen. Eine Schrift mit geringer Strichstärke wirkt heller und verträgt engere Zeilen. Beides lässt sich nur im Satz beurteilen, nicht an einer Tabelle von Buchstaben.

1.3.1 Serifen und ihre Wirkung

Serifen sind die kleinen Abschlüsse an den Enden der Buchstaben. Ihnen wird nachgesagt, sie führten das Auge über die Zeile und erhöhten damit die Lesbarkeit im Druck. Ob das messbar stimmt, ist umstritten; sicher ist, dass Serifenschriften im Fließsatz seit Jahrhunderten den Standard bilden und dass Leserinnen und Leser sie deshalb erwarten. In den Ingenieurwissenschaften sind serifenlose Schriften verbreiteter, teils aus Gewohnheit, teils weil sie in technischen Zeichnungen und auf Bildschirmen besser bestehen.

Für eine Abschlussarbeit gilt: Beides ist zulässig, und die Wahl sollte begründet und dann durchgehalten werden. Ein Wechsel mitten im Dokument fällt auf, auch wenn er nicht bewusst wahrgenommen wird. Die Richtlinie der Fakultät schreibt deshalb ausdrücklich eine einzige Schriftart für das ganze Dokument vor, mit Ausnahme der Formeln.

1.3.2 Die Schreibmaschinenschrift

Eine dritte Schrift kommt ins Spiel, sobald Dateinamen, Befehle oder Quelltext vorkommen. Sie wird mit `\texttt{...}` gesetzt und sollte zur Textschrift passen, ohne mit ihr zu verschmelzen: Der Aufruf `lualatex BeispielthesisNXT` muss auf den ersten Blick als Eingabe erkennbar sein, und in `Code/daten.py` darf man die Null in `p_i0 = 0.85` nicht mit dem Buchstaben `O` verwechseln. Genau daran scheitern viele Schriften. Prüfen lässt es sich an `Il1| O0` – großes I, kleines l, Eins, senkrechter Strich, großes O, Null. Wo diese Zeichen ununterscheidbar sind, taugt die Schrift nicht für Quelltext.

1.3.3 Mathematik und Text

Wo Formeln im Fließtext stehen, entscheidet sich, ob eine Schrift für eine technische Arbeit taugt. Die Mathematikschrift muss zur Textschrift passen: Ein Wert wie $\alpha = 0{,}05$ oder ein Symbol wie $C_{\max}$ steht mitten im Satz und darf dort weder größer noch kleiner

1 Schriftmuster

Dies ist ein Absatz im Grundtext. Er enthält Umlaute (äöüÄÖÜß), eine *Hervorhebung*, eine **fette Auszeichnung** und `Schreibmaschinenschrift`. Zahlen im Text: 1234567890. Ligaturen: Auffinden, Schifffahrt, Waffel, fünf.

Abkürzungen setzt die Klasse in Kapitälchen, sofern die Schrift echte mitbringt: Mixed Integer Linear Programming (MILP) und Simulated Annealing (SA); im zweiten Gebrauch dann MILP. Fehlen echte Kapitälchen, bleibt es beim normalen Schnitt – aus Versalien heruntergerechnete Kapitälchen wirken zu leicht und fallen im Text unangenehm auf.

1.1 Mathematik

Im Text $C_{\max} = \max_j \sum_i p_{ij} x_{ij}$ und abgesetzt:

$$\mathcal{L}(\mathbf{x}) = \frac{1}{2m} \sum_{i=1}^{m} \left(f(\mathbf{u}_i; \mathbf{x}) - y_i \right)^2, \qquad \alpha \le \frac{2}{\lambda_{\max}}. \tag{1.1}$$

1.2 Tabellensatz

Verfahren	Makespan	Zeit
Greedy	266,67	0,0
LPT	265,38	0,0
MILP	236,20	640,6

Tabelle 1.1: Zahlen im Tabellensatz

`gyre` – TeX Gyre Pagella, eine freie Palatino. Große Mittellänge, offene Formen, ruhig im Fließsatz. Kein Semibold, Überschriften werden deshalb fett gesetzt. Eigene Mathematikschrift.

Abbildung A.6: Schriftmuster `nxtthesis-gyre`: Titelseite, Schriftmuster und dicht gesetzte Textseite.

Optimierung des Mitarbeitereinsatzes

Vergleichende Analyse in der metallverarbeitenden Industrie

Thesis zur Erlangung des akademischen Grades

Bachelor of Science

an der Fakultät NXT Nachhaltigkeit und Technologie
der Hochschule Reutlingen

im Studiengang

Nachhaltige Technologie

Vorgelegt von:
Jean Dujardin
Wegstraße 123, 72764 Reutlingen
Matrikelnummer 1234567

Abgabedatum: 24. Dezember 2026

Erstprüfer: Prof. Dr. Katharina Erstmal
Zweitprüfer: Prof. Dr. Kornelius Hurtig

1.3 Der Grauwert im Fließsatz

Eine Schrift lässt sich nicht an einem Wort beurteilen und auch nicht an einer Zeile. Erst über einen ganzen Absatz hinweg zeigt sich, welchen Grauwert sie erzeugt, wie ruhig die Zeilen laufen und ob die Wortabstände gleichmäßig bleiben. Diese Seite ist deshalb absichtlich dicht gesetzt: Sie soll den Eindruck vermitteln, den eine Seite der fertigen Arbeit machen wird, und nicht den einer Schriftmustertafel.

Der Grauwert entsteht aus dem Zusammenspiel von Strichstärke, x-Höhe, Laufweite und Zeilenabstand. Eine Schrift mit großer x-Höhe wirkt bei gleichem Schriftgrad größer und braucht mehr Zeilenabstand, sonst verkleben die Zeilen. Eine Schrift mit geringer Strichstärke wirkt heller und verträgt engere Zeilen. Beides lässt sich nur im Satz beurteilen, nicht an einer Tabelle von Buchstaben.

1.3.1 Serifen und ihre Wirkung

Serifen sind die kleinen Abschlüsse an den Enden der Buchstaben. Ihnen wird nachgesagt, sie führten das Auge über die Zeile und erhöhten damit die Lesbarkeit im Druck. Ob das messbar stimmt, ist umstritten; sicher ist, dass Serifenschriften im Fließsatz seit Jahrhunderten den Standard bilden und dass Leserinnen und Leser sie deshalb erwarten. In den Ingenieurwissenschaften sind serifenlose Schriften verbreiteter, teils aus Gewohnheit, teils weil sie in technischen Zeichnungen und auf Bildschirmen besser bestehen.

Für eine Abschlussarbeit gilt: Beides ist zulässig, und die Wahl sollte begründet und dann durchgehalten werden. Ein Wechsel mitten im Dokument fällt auf, auch wenn er nicht bewusst wahrgenommen wird. Die Richtlinie der Fakultät schreibt deshalb ausdrücklich eine einzige Schriftart für das ganze Dokument vor, mit Ausnahme der Formeln.

1.3.2 Die Schreibmaschinenschrift

Eine dritte Schrift kommt ins Spiel, sobald Dateinamen, Befehle oder Quelltext vorkommen. Sie wird mit `\texttt{...}` gesetzt und sollte zur Textschrift passen, ohne mit ihr zu verschmelzen: Der Aufruf `lualatex Beispielthesis` NXT muss auf den ersten Blick als Eingabe erkennbar sein, und in `Code/daten.py` darf man die Null in `p_10 = 0.85` nicht mit dem Buchstaben `O` verwechseln. Genau daran scheitern viele Schriften. Prüfen lässt es sich an `Il1| O0` – großes I, kleines l, Eins, senkrechter Strich, großes O, Null. Wo diese Zeichen ununterscheidbar sind, taugt die Schrift nicht für Quelltext.

1.3.3 Mathematik und Text

Wo Formeln im Fließtext stehen, entscheidet sich, ob eine Schrift für eine technische Arbeit taugt. Die Mathematikschrift muss zur Textschrift passen: Ein Wert wie $\alpha = 0{,}05$ oder ein Symbol wie $C_{\max}$ steht mitten im Satz und darf dort weder größer noch kleiner noch dunkler wirken als seine Umgebung. Schriftfamilien, die eine eigene Mathematikschrift mitbringen, sind hier im Vorteil – STIX Two, Libertinus und Pagella tun das, und die Klasse bindet sie entsprechend ein.

1 Schriftmuster

Dies ist ein Absatz im Grundtext. Er enthält Umlaute (äöüÄÖÜß), eine *Hervorhebung*, eine **fette Auszeichnung** und `Schreibmaschinenschrift`. Zahlen im Text: 1234567890. Ligaturen: Auffinden, Schifffahrt, Waffel, fünf.

Abkürzungen setzt die Klasse in Kapitälchen, sofern die Schrift echte mitbringt: Mixed Integer Linear Programming (MILP) und Simulated Annealing (SA); im zweiten Gebrauch dann MILP. Fehlen echte Kapitälchen, bleibt es beim normalen Schnitt – aus Versalien heruntergerechnete Kapitälchen wirken zu leicht und fallen im Text unangenehm auf.

1.1 Mathematik

Im Text $C_{\max} = \max_i \sum_j p_{ij} x_{ij}$ und abgesetzt:

$$\mathscr{L}(\mathbf{x}) = \frac{1}{2m} \sum_{i=1}^{m} \bigl(f(\mathbf{u}_i; \mathbf{x}) - y_i \bigr)^2, \qquad \alpha \le \frac{2}{\lambda_{\max}}. \tag{1.1}$$

1.2 Tabellensatz

Verfahren	Makespan	Zeit
Greedy	266,67	0,0
LPT	265,38	0,0
MILP	236,20	640,6

Tabelle 1.1: Zahlen im Tabellensatz

`libertinus` – Nachfolger der Linux Libertine, an klassischen Renaissance-Antiqua orientiert. Etwas schmaler als Pagella, dadurch mehr Text je Zeile. Semibold vorhanden, eigene Mathematikschrift.

Abbildung A.7: Schriftmuster `nxtthesis-libertinus`: Titelseite, Schriftmuster und dicht gesetzte Textseite.

Optimierung des Mitarbeitereinsatzes

Vergleichende Analyse in der metallverarbeitenden Industrie

Thesis zur Erlangung des akademischen Grades

Bachelor of Science

an der Fakultät NXT Nachhaltigkeit und Technologie
der Hochschule Reutlingen

im Studiengang

Nachhaltige Technologie

Vorgelegt von:
Jean Dujardin
Wegstraße 123, 72764 Reutlingen
Matrikelnummer 1234567

Abgabedatum: 24. Dezember 2026

Erstprüfer: Prof. Dr. Katharina Erstmal
Zweitprüfer: Prof. Dr. Kornelius Hurtig

1 Schriftmuster 2

1.3 Der Grauwert im Fließsatz

Eine Schrift lässt sich nicht an einem Wort beurteilen und auch nicht an einer Zeile. Erst über einen ganzen Absatz hinweg zeigt sich, welchen Grauwert sie erzeugt, wie ruhig die Zeilen laufen und ob die Wortabstände gleichmäßig bleiben. Diese Seite ist deshalb absichtlich dicht gesetzt: Sie soll den Eindruck vermitteln, den eine Seite der fertigen Arbeit machen wird, und nicht den einer Schriftmustertafel.

Der Grauwert entsteht aus dem Zusammenspiel von Strichstärke, x-Höhe, Laufweite und Zeilenabstand. Eine Schrift mit großer x-Höhe wirkt bei gleichem Schriftgrad größer und braucht mehr Zeilenabstand, sonst verkleben die Zeilen. Eine Schrift mit geringer Strichstärke wirkt heller und verträgt engere Zeilen. Beides lässt sich nur im Satz beurteilen, nicht an einer Tabelle von Buchstaben.

1.3.1 Serifen und ihre Wirkung

Serifen sind die kleinen Abschlüsse an den Enden der Buchstaben. Ihnen wird nachgesagt, sie führten das Auge über die Zeile und erhöhten damit die Lesbarkeit im Druck. Ob das messbar stimmt, ist umstritten; sicher ist, dass Serifenschriften im Fließsatz seit Jahrhunderten den Standard bilden und dass Leserinnen und Leser sie deshalb erwarten. In den Ingenieurwissenschaften sind serifenlose Schriften verbreiteter, teils aus Gewohnheit, teils weil sie in technischen Zeichnungen und auf Bildschirmen besser bestehen.

Für eine Abschlussarbeit gilt: Beides ist zulässig, und die Wahl sollte begründet und dann durchgehalten werden. Ein Wechsel mitten im Dokument fällt auf, auch wenn er nicht bewusst wahrgenommen wird. Die Richtlinie der Fakultät schreibt deshalb ausdrücklich eine einzige Schriftart für das ganze Dokument vor, mit Ausnahme der Formeln.

1.3.2 Die Schreibmaschinenschrift

Eine dritte Schrift kommt ins Spiel, sobald Dateinamen, Befehle oder Quelltext vorkommen. Sie wird mit `\texttt{...}` gesetzt und sollte zur Textschrift passen, ohne mit ihr zu verschmelzen: Der Aufruf `lualatex BeispielthesisNXT` muss auf den ersten Blick als Eingabe erkennbar sein, und in `Code/daten.py` darf man die Null in `p_10 = 0.85` nicht mit dem Buchstaben `O` verwechseln. Genau daran scheitern viele Schriften. Prüfen lässt es sich an `Ill| O0` – großes I, kleines l, Eins, senkrechter Strich, großes O, Null. Wo diese Zeichen ununterscheidbar sind, taugt die Schrift nicht für Quelltext.

1.3.3 Mathematik und Text

Wo Formeln im Fließtext stehen, entscheidet sich, ob eine Schrift für eine technische Arbeit taugt. Die Mathematikschrift muss zur Textschrift passen: Ein Wert wie $\alpha = 0{,}05$ oder ein Symbol wie $C_{\max}$ steht mitten im Satz und darf dort weder größer noch kleiner noch dunkler wirken als seine Umgebung. Schriftfamilien, die eine eigene Mathematikschrift mitbringen, sind hier im Vorteil – STIX Two, Libertinus und Pagella tun das, und die Klasse bindet sie entsprechend ein.

1 Schriftmuster 1

1 Schriftmuster

Dies ist ein Absatz im Grundtext. Er enthält Umlaute (äöüÄÖÜß), eine *Hervorhebung*, eine **fette Auszeichnung** und `Schreibmaschinenschrift`. Zahlen im Text: 1234567890. Ligaturen: Auffinden, Schifffahrt, Waffel, fünf.

Abkürzungen setzt die Klasse in Kapitälchen, sofern die Schrift echte mitbringt: Mixed Integer Linear Programming (MILP) und Simulated Annealing (SA); im zweiten Gebrauch dann MILP. Fehlen echte Kapitälchen, bleibt es beim normalen Schnitt – aus Versalien heruntergerechnete Kapitälchen wirken zu leicht und fallen im Text unangenehm auf.

1.1 Mathematik

Im Text $C_{\max} = \max_i \sum_j p_{ij} x_{ij}$ und abgesetzt:

$$\mathcal{L}(\mathbf{x}) = \frac{1}{2m} \sum_{i=1}^{m} \left(f(\mathbf{u}_i; \mathbf{x}) - y_i \right)^2, \qquad \alpha \le \frac{2}{\lambda_{\max}}. \tag{1.1}$$

1.2 Tabellensatz

Verfahren	Makespan	Zeit
Greedy	266,67	0,0
LPT	265,38	0,0
MILP	236,20	640,6

Tabelle 1.1: Zahlen im Tabellensatz

`newtx` – TeX Gyre TermesX, eine überarbeitete Times aus dem Paket newtx. Times ist schmaler als die übrigen Serifenschriften und bringt damit mehr Text auf die Seite. Echte Kapitälchen, kein Semibold. Mathematik: TeX Gyre Termes Math.

Abbildung A.8: Schriftmuster `nxtthesis-newtx`: Titelseite, Schriftmuster und dicht gesetzte Textseite.

Optimierung des Mitarbeitereinsatzes

Vergleichende Analyse in der metallverarbeitenden Industrie

Thesis zur Erlangung des akademischen Grades

Bachelor of Science

an der Fakultät NXT Nachhaltigkeit und Technologie
der Hochschule Reutlingen

im Studiengang

Nachhaltige Technologie

Vorgelegt von:
Jean Dujardin
Wegstraße 123, 72764 Reutlingen
Matrikelnummer 1234567

Abgabedatum: 24. Dezember 2026

Erstprüfer: Prof. Dr. Katharina Erstmal
Zweitprüfer: Prof. Dr. Kornelius Hurtig

1 Schriftmuster 2

1.3 Der Grauwert im Fließsatz

Eine Schrift lässt sich nicht an einem Wort beurteilen und auch nicht an einer Zeile. Erst über einen ganzen Absatz hinweg zeigt sich, welchen Grauwert sie erzeugt, wie ruhig die Zeilen laufen und ob die Wortabstände gleichmäßig bleiben. Diese Seite ist deshalb absichtlich dicht gesetzt: Sie soll den Eindruck vermitteln, den eine Seite der fertigen Arbeit machen wird, und nicht den einer Schriftmustertafel.

Der Grauwert entsteht aus dem Zusammenspiel von Strichstärke, x-Höhe, Laufweite und Zeilenabstand. Eine Schrift mit großer x-Höhe wirkt bei gleichem Schriftgrad größer und braucht mehr Zeilenabstand, sonst verkleben die Zeilen. Eine Schrift mit geringer Strichstärke wirkt heller und verträgt engere Zeilen. Beides lässt sich nur im Satz beurteilen, nicht an einer Tabelle von Buchstaben.

1.3.1 Serifen und ihre Wirkung

Serifen sind die kleinen Abschlüsse an den Enden der Buchstaben. Ihnen wird nachgesagt, sie führten das Auge über die Zeile und erhöhten damit die Lesbarkeit im Druck. Ob das messbar stimmt, ist umstritten; sicher ist, dass Serifenschriften im Fließsatz seit Jahrhunderten den Standard bilden und dass Leserinnen und Leser sie deshalb erwarten. In den Ingenieurwissenschaften sind serifenlose Schriften verbreiteter, teils aus Gewohnheit, teils weil sie in technischen Zeichnungen und auf Bildschirmen besser bestehen.

Für eine Abschlussarbeit gilt: Beides ist zulässig, und die Wahl sollte begründet und dann durchgehalten werden. Ein Wechsel mitten im Dokument fällt auf, auch wenn er nicht bewusst wahrgenommen wird. Die Richtlinie der Fakultät schreibt deshalb ausdrücklich eine einzige Schriftart für das ganze Dokument vor, mit Ausnahme der Formeln.

1.3.2 Die Schreibmaschinenschrift

Eine dritte Schrift kommt ins Spiel, sobald Dateinamen, Befehle oder Quelltext vorkommen. Sie wird mit `\texttt{...}` gesetzt und sollte zur Textschrift passen, ohne mit ihr zu verschmelzen: Der Aufruf `lualatex BeispielthesisNXT` muss auf den ersten Blick als Eingabe erkennbar sein, und in `Code/daten.py` darf man die Null in `p_i0 = 0.85` nicht mit dem Buchstaben O verwechseln. Genau daran scheitern viele Schriften. Prüfen lässt es sich an `Il1| O0` – großes I, kleines l, Eins, senkrechter Strich, großes O, Null. Wo diese Zeichen ununterscheidbar sind, taugt die Schrift nicht für Quelltext.

1.3.3 Mathematik und Text

Wo Formeln im Fließtext stehen, entscheidet sich, ob eine Schrift für eine technische Arbeit taugt. Die Mathematikschrift muss zur Textschrift passen: Ein Wert wie $\alpha = 0{,}05$ oder ein Symbol wie $C_{\max}$ steht mitten im Satz und darf dort weder größer noch kleiner

1 Schriftmuster 1

1 Schriftmuster

Dies ist ein Absatz im Grundtext. Er enthält Umlaute (äöüÄÖÜß), eine Hervorhebung, eine **fette Auszeichnung** und `Schreibmaschinenschrift`. Zahlen im Text: 1234567890. Ligaturen: Auffinden, Schifffahrt, Waffel, fünf.

Abkürzungen setzt die Klasse in Kapitälchen, sofern die Schrift echte mitbringt: Mixed Integer Linear Programming (MILP) und Simulated Annealing (SA); im zweiten Gebrauch dann MILP. Fehlen echte Kapitälchen, bleibt es beim normalen Schnitt – aus Versalien heruntergerechnete Kapitälchen wirken zu leicht und fallen im Text unangenehm auf.

1.1 Mathematik

Im Text $C_{\max} = \max_i \sum_j p_{ij} x_{ij}$ und abgesetzt:

$$\mathcal{L}(\mathbf{x}) = \frac{1}{2m} \sum_{i=1}^{m} \left(f(\mathbf{u}_i; \mathbf{x}) - y_i \right)^2, \qquad \alpha \leqslant \frac{2}{\lambda_{\max}}. \tag{1.1}$$

1.2 Tabellensatz

Verfahren	Makespan	Zeit
Greedy	266,67	0,0
LPT	265,38	0,0
MILP	236,20	640,6

Tabelle 1.1: Zahlen im Tabellensatz

segoeui – serifenlos, die Hausschrift der Hochschule. Modern und nüchtern, im langen Fließtext anstrengender als eine Antiqua. Semibold vorhanden. Die Mathematik übernimmt Euler – ein bewusster Bruch, weil es keine passende serifenlose Mathematikschrift gibt.

Abbildung A.9: Schriftmuster `nxtthesis-segoeui`: Titelseite, Schriftmuster und dicht gesetzte Textseite.

Optimierung des Mitarbeitereinsatzes

Vergleichende Analyse in der metallverarbeitenden Industrie

Thesis zur Erlangung des akademischen Grades

Bachelor of Science

an der Fakultät NXT Nachhaltigkeit und Technologie
der Hochschule Reutlingen

im Studiengang

Nachhaltige Technologie

Vorgelegt von:
Jean Dujardin
Wegstraße 123, 72764 Reutlingen
Matrikelnummer 1234567

Abgabedatum: 24. Dezember 2026

Erstprüfer: Prof. Dr. Katharina Erstmal
Zweitprüfer: Prof. Dr. Kornelius Hurtig

1.3 Der Grauwert im Fließsatz

Eine Schrift lässt sich nicht an einem Wort beurteilen und auch nicht an einer Zeile. Erst über einen ganzen Absatz hinweg zeigt sich, welchen Grauwert sie erzeugt, wie ruhig die Zeilen laufen und ob die Wortabstände gleichmäßig bleiben. Diese Seite ist deshalb absichtlich dicht gesetzt: Sie soll den Eindruck vermitteln, den eine Seite der fertigen Arbeit machen wird, und nicht den einer Schriftmustertafel.

Der Grauwert entsteht aus dem Zusammenspiel von Strichstärke, x-Höhe, Laufweite und Zeilenabstand. Eine Schrift mit großer x-Höhe wirkt bei gleichem Schriftgrad größer und braucht mehr Zeilenabstand, sonst verkleben die Zeilen. Eine Schrift mit geringer Strichstärke wirkt heller und verträgt engere Zeilen. Beides lässt sich nur im Satz beurteilen, nicht an einer Tabelle von Buchstaben.

1.3.1 Serifen und ihre Wirkung

Serifen sind die kleinen Abschlüsse an den Enden der Buchstaben. Ihnen wird nachgesagt, sie führten das Auge über die Zeile und erhöhten damit die Lesbarkeit im Druck. Ob das messbar stimmt, ist umstritten; sicher ist, dass Serifenschriften im Fließsatz seit Jahrhunderten den Standard bilden und dass Leserinnen und Leser sie deshalb erwarten. In den Ingenieurwissenschaften sind serifenlose Schriften verbreiteter, teils aus Gewohnheit, teils weil sie in technischen Zeichnungen und auf Bildschirmen besser bestehen.

Für eine Abschlussarbeit gilt: Beides ist zulässig, und die Wahl sollte begründet und dann durchgehalten werden. Ein Wechsel mitten im Dokument fällt auf, auch wenn er nicht bewusst wahrgenommen wird. Die Richtlinie der Fakultät schreibt deshalb ausdrücklich eine einzige Schriftart für das ganze Dokument vor, mit Ausnahme der Formeln.

1.3.2 Die Schreibmaschinenschrift

Eine dritte Schrift kommt ins Spiel, sobald Dateinamen, Befehle oder Quelltext vorkommen. Sie wird mit `\texttt{...}` gesetzt und sollte zur Textschrift passen, ohne mit ihr zu verschmelzen: Der Aufruf `lualatex BeispielthesisNXT` muss auf den ersten Blick als Eingabe erkennbar sein, und in `Code/daten.py` darf man die Null in `p_10 = 0.85` nicht mit dem Buchstaben `O` verwechseln. Genau daran scheitern viele Schriften. Prüfen lässt es sich an `Il1| O0` – großes I, kleines l, Eins, senkrechter Strich, großes O, Null. Wo diese Zeichen ununterscheidbar sind, taugt die Schrift nicht für Quelltext.

1.3.3 Mathematik und Text

Wo Formeln im Fließtext stehen, entscheidet sich, ob eine Schrift für eine technische Arbeit taugt. Die Mathematikschrift muss zur Textschrift passen: Ein Wert wie $\alpha = 0{,}05$ oder ein Symbol wie $C_{\max}$ steht mitten im Satz und darf dort weder größer noch kleiner noch dunkler wirken als seine Umgebung. Schriftfamilien, die eine eigene Mathematikschrift mitbringen, sind hier im Vorteil – STIX Two, Libertinus und Pagella tun das, und die Klasse bindet sie entsprechend ein.

1 Schriftmuster

Dies ist ein Absatz im Grundtext. Er enthält Umlaute (äöüÄÖÜß), eine *Hervorhebung*, eine **fette Auszeichnung** und `Schreibmaschinenschrift`. Zahlen im Text: 1234567890. Ligaturen: Auffinden, Schifffahrt, Waffel, fünf.

Abkürzungen setzt die Klasse in Kapitälchen, sofern die Schrift echte mitbringt: Mixed Integer Linear Programming (MILP) und Simulated Annealing (SA); im zweiten Gebrauch dann MILP. Fehlen echte Kapitälchen, bleibt es beim normalen Schnitt – aus Versalien heruntergerechnete Kapitälchen wirken zu leicht und fallen im Text unangenehm auf.

1.1 Mathematik

Im Text $C_{\max} = \max_i \sum_j p_{ij} x_{ij}$ und abgesetzt:

$$\mathcal{L}(\mathbf{x}) = \frac{1}{2m} \sum_{i=1}^{m} \bigl(f(\mathbf{u}_i; \mathbf{x}) - y_i \bigr)^2 , \qquad \alpha \le \frac{2}{\lambda_{\max}} . \tag{1.1}$$

1.2 Tabellensatz

Verfahren	Makespan	Zeit
Greedy	266,67	0,0
LPT	265,38	0,0
MILP	236,20	640,6

Tabelle 1.1: Zahlen im Tabellensatz

`stix` – für den wissenschaftlichen Satz entworfen, mit sehr vollständiger Mathematik. Semibold vorhanden.

Abbildung A.10: Schriftmuster `nxtthesis-stix`: Titelseite, Schriftmuster und dicht gesetzte Textseite.

Optimierung des Mitarbeitereinsatzes

Vergleichende Analyse in der metallverarbeitenden Industrie

Thesis zur Erlangung des akademischen Grades

Bachelor of Science

an der Fakultät NXT Nachhaltigkeit und Technologie
der Hochschule Reutlingen

im Studiengang

Nachhaltige Technologie

Vorgelegt von:
Jean Dujardin
Wegstraße 123, 72764 Reutlingen
Matrikelnummer 1234567

Abgabedatum: 24. Dezember 2026

Erstprüfer: Prof. Dr. Katharina Erstmal
Zweitprüfer: Prof. Dr. Kornelius Hurtig

1.3 Der Grauwert im Fließsatz

Eine Schrift lässt sich nicht an einem Wort beurteilen und auch nicht an einer Zeile. Erst über einen ganzen Absatz hinweg zeigt sich, welchen Grauwert sie erzeugt, wie ruhig die Zeilen laufen und ob die Wortabstände gleichmäßig bleiben. Diese Seite ist deshalb absichtlich dicht gesetzt: Sie soll den Eindruck vermitteln, den eine Seite der fertigen Arbeit machen wird, und nicht den einer Schriftmustertafel.

Der Grauwert entsteht aus dem Zusammenspiel von Strichstärke, x-Höhe, Laufweite und Zeilenabstand. Eine Schrift mit großer x-Höhe wirkt bei gleichem Schriftgrad größer und braucht mehr Zeilenabstand, sonst verkleben die Zeilen. Eine Schrift mit geringer Strichstärke wirkt heller und verträgt engere Zeilen. Beides lässt sich nur im Satz beurteilen, nicht an einer Tabelle von Buchstaben.

1.3.1 Serifen und ihre Wirkung

Serifen sind die kleinen Abschlüsse an den Enden der Buchstaben. Ihnen wird nachgesagt, sie führten das Auge über die Zeile und erhöhten damit die Lesbarkeit im Druck. Ob das messbar stimmt, ist umstritten; sicher ist, dass Serifenschriften im Fließsatz seit Jahrhunderten den Standard bilden und dass Leserinnen und Leser sie deshalb erwarten. In den Ingenieurwissenschaften sind serifenlose Schriften verbreiteter, teils aus Gewohnheit, teils weil sie in technischen Zeichnungen und auf Bildschirmen besser bestehen.

Für eine Abschlussarbeit gilt: Beides ist zulässig, und die Wahl sollte begründet und dann durchgehalten werden. Ein Wechsel mitten im Dokument fällt auf, auch wenn er nicht bewusst wahrgenommen wird. Die Richtlinie der Fakultät schreibt deshalb ausdrücklich eine einzige Schriftart für das ganze Dokument vor, mit Ausnahme der Formeln.

1.3.2 Die Schreibmaschinenschrift

Eine dritte Schrift kommt ins Spiel, sobald Dateinamen, Befehle oder Quelltext vorkommen. Sie wird mit `\texttt{...}` gesetzt und sollte zur Textschrift passen, ohne mit ihr zu verschmelzen: Der Aufruf `lualatex BeispielthesisNXT` muss auf den ersten Blick als Eingabe erkennbar sein, und in `Code/daten.py` darf man die Null in `p_i0 = 0.85` nicht mit dem Buchstaben `O` verwechseln. Genau daran scheitern viele Schriften. Prüfen lässt es sich an `Il1| O0` – großes I, kleines l, Eins, senkrechter Strich, großes O, Null. Wo diese Zeichen ununterscheidbar sind, taugt die Schrift nicht für Quelltext.

1.3.3 Mathematik und Text

Wo Formeln im Fließtext stehen, entscheidet sich, ob eine Schrift für eine technische Arbeit taugt. Die Mathematikschrift muss zur Textschrift passen: Ein Wert wie $\alpha = 0{,}05$ oder ein Symbol wie $C_{\max}$ steht mitten im Satz und darf dort weder größer noch kleiner noch dunkler wirken als seine Umgebung. Schriftfamilien, die eine eigene Mathematikschrift mitbringen, sind hier im Vorteil – STIX Two, Libertinus und Pagella tun das, und die Klasse bindet sie entsprechend ein.

1 Schriftmuster

Dies ist ein Absatz im Grundtext. Er enthält Umlaute (äöüÄÖÜß), eine *Hervorhebung*, eine **fette Auszeichnung** und `Schreibmaschinenschrift`. Zahlen im Text: 1234567890. Ligaturen: Auffinden, Schifffahrt, Waffel, fünf.

Abkürzungen setzt die Klasse in Kapitälchen, sofern die Schrift echte mitbringt: Mixed Integer Linear Programming (MILP) und Simulated Annealing (SA); im zweiten Gebrauch dann MILP. Fehlen echte Kapitälchen, bleibt es beim normalen Schnitt – aus Versalien heruntergerechnete Kapitälchen wirken zu leicht und fallen im Text unangenehm auf.

1.1 Mathematik

Im Text $C_{\max} = \max_i \sum_j p_{ij} x_{ij}$ und abgesetzt:

$$\mathcal{L}(\mathbf{x}) = \frac{1}{2m} \sum_{i=1}^{m} \left(f(\mathbf{u}_i; \mathbf{x}) - y_i \right)^2, \qquad \alpha \le \frac{2}{\lambda_{\max}}. \tag{1.1}$$

1.2 Tabellensatz

Verfahren	Makespan	Zeit
Greedy	266,67	0,0
LPT	265,38	0,0
MILP	236,20	640,6

Tabelle 1.1: Zahlen im Tabellensatz

`times` – das echte Times New Roman des Betriebssystems, dazu Arial und Courier New. Die Wahl, wenn eine Arbeit wie ein Word-Dokument aussehen soll. *Einschränkung:* Times New Roman bringt als einzige Option keine echten Kapitälchen mit. Abkürzungen erscheinen deshalb im normalen Schnitt, und mit `kapitaelchen` werden die Überschriften serifenlos statt in Kapitälchen gesetzt. Wer Times möchte und Kapitälchen braucht, nimmt `newtx`.

Abbildung A.11: Schriftmuster `nxtthesis-times`: Titelseite, Schriftmuster und dicht gesetzte Textseite.

Optimierung des Mitarbeitereinsatzes

Vergleichende Analyse in der metallverarbeitenden Industrie

Thesis zur Erlangung des akademischen Grades

Bachelor of Science

an der Fakultät NXT Nachhaltigkeit und Technologie
der Hochschule Reutlingen

im Studiengang

Nachhaltige Technologie

Vorgelegt von:
Jean Dujardin
Wegstraße 123, 72764 Reutlingen
Matrikelnummer 1234567

Abgabedatum: 24. Dezember 2026

Erstprüfer: Prof. Dr. Katharina Erstmal
Zweitprüfer: Prof. Dr. Kornelius Hurtig

1.3 Der Grauwert im Fließsatz

Eine Schrift lässt sich nicht an einem Wort beurteilen und auch nicht an einer Zeile. Erst über einen ganzen Absatz hinweg zeigt sich, welchen Grauwert sie erzeugt, wie ruhig die Zeilen laufen und ob die Wortabstände gleichmäßig bleiben. Diese Seite ist deshalb absichtlich dicht gesetzt: Sie soll den Eindruck vermitteln, den eine Seite der fertigen Arbeit machen wird, und nicht den einer Schriftmustertafel.

Der Grauwert entsteht aus dem Zusammenspiel von Strichstärke, x-Höhe, Laufweite und Zeilenabstand. Eine Schrift mit großer x-Höhe wirkt bei gleichem Schriftgrad größer und braucht mehr Zeilenabstand, sonst verkleben die Zeilen. Eine Schrift mit geringer Strichstärke wirkt heller und verträgt engere Zeilen. Beides lässt sich nur im Satz beurteilen, nicht an einer Tabelle von Buchstaben.

1.3.1 Serifen und ihre Wirkung

Serifen sind die kleinen Abschlüsse an den Enden der Buchstaben. Ihnen wird nachgesagt, sie führten das Auge über die Zeile und erhöhten damit die Lesbarkeit im Druck. Ob das messbar stimmt, ist umstritten; sicher ist, dass Serifenschriften im Fließsatz seit Jahrhunderten den Standard bilden und dass Leserinnen und Leser sie deshalb erwarten. In den Ingenieurwissenschaften sind serifenlose Schriften verbreiteter, teils aus Gewohnheit, teils weil sie in technischen Zeichnungen und auf Bildschirmen besser bestehen.

Für eine Abschlussarbeit gilt: Beides ist zulässig, und die Wahl sollte begründet und dann durchgehalten werden. Ein Wechsel mitten im Dokument fällt auf, auch wenn er nicht bewusst wahrgenommen wird. Die Richtlinie der Fakultät schreibt deshalb ausdrücklich eine einzige Schriftart für das ganze Dokument vor, mit Ausnahme der Formeln.

1.3.2 Die Schreibmaschinenschrift

Eine dritte Schrift kommt ins Spiel, sobald Dateinamen, Befehle oder Quelltext vorkommen. Sie wird mit `\texttt{...}` gesetzt und sollte zur Textschrift passen, ohne mit ihr zu verschmelzen: Der Aufruf `lualatex BeispielthesisNXT` muss auf den ersten Blick als Eingabe erkennbar sein, und in `Code/daten.py` darf man die Null in `p_i0 = 0.85` nicht mit dem Buchstaben O verwechseln. Genau daran scheitern viele Schriften. Prüfen lässt es sich an `Il1| O0` – großes I, kleines l, Eins, senkrechter Strich, großes O, Null. Wo diese Zeichen ununterscheidbar sind, taugt die Schrift nicht für Quelltext.

1.3.3 Mathematik und Text

Wo Formeln im Fließtext stehen, entscheidet sich, ob eine Schrift für eine technische Arbeit taugt. Die Mathematikschrift muss zur Textschrift passen: Ein Wert wie $\alpha = 0{,}05$ oder ein Symbol wie $C_{\max}$ steht mitten im Satz und darf dort weder größer noch kleiner noch dunkler wirken als seine Umgebung. Schriftfamilien, die eine eigene Mathematikschrift mitbringen, sind hier im Vorteil – STIX Two, Libertinus und Pagella tun das, und die Klasse bindet sie entsprechend ein.

1 Schriftmuster

Dies ist ein Absatz im Grundtext. Er enthält Umlaute (äöüÄÖÜß), eine *Hervorhebung*, eine **fette Auszeichnung** und `Schreibmaschinenschrift`. Zahlen im Text: 1234567890. Ligaturen: Auffinden, Schifffahrt, Waffel, fünf.

Abkürzungen setzt die Klasse in Kapitälchen, sofern die Schrift echte mitbringt: Mixed Integer Linear Programming (MILP) und Simulated Annealing (SA); im zweiten Gebrauch dann MILP. Fehlen echte Kapitälchen, bleibt es beim normalen Schnitt – aus Versalien heruntergerechnete Kapitälchen wirken zu leicht und fallen im Text unangenehm auf.

1.1 Mathematik

Im Text $C_{\max} = \max_j \sum_i p_{ij} x_{ij}$ und abgesetzt:

$$\mathscr{L}(\mathbf{x}) = \frac{1}{2m} \sum_{i=1}^{m} \left(f(\mathbf{u}_i; \mathbf{x}) - y_i \right)^2, \qquad \alpha \leq \frac{2}{\lambda_{\max}}. \tag{1.1}$$

1.2 Tabellensatz

Verfahren	Makespan	Zeit
Greedy	266,67	0,0
LPT	265,38	0,0
MILP	236,20	640,6

Tabelle 1.1: Zahlen im Tabellensatz

`utopia` – Erewhon, die OpenType-Fassung von Robert Slimbachs Utopia. Ein ruhiger Kompromiss zwischen Charter und Garamond, für technische Texte gut geeignet. Kein Semibold. Eigene Mathematikschrift.

Abbildung A.12: Schriftmuster `nxtthesis-utopia`: Titelseite, Schriftmuster und dicht gesetzte Textseite.

A.2 DIE LAYOUTOPTIONEN

Die beiden folgenden Muster zeigen nicht eine andere Schrift, sondern ein anderes Layout – beide in `stix`, damit der Unterschied allein am Layout ablesbar ist.

Optimierung des Mitarbeitereinsatzes

Vergleichende Analyse in der metallverarbeitenden Industrie

Thesis zur Erlangung des akademischen Grades

Bachelor of Science

an der Fakultät NXT Nachhaltigkeit und Technologie
der Hochschule Reutlingen

im Studiengang

Nachhaltige Technologie

Vorgelegt von:
Jean Dujardin
Wegstraße 123, 72764 Reutlingen
Matrikelnummer 1234567

Abgabedatum: 24. Dezember 2026

Erstprüfer: Prof. Dr. Katharina Erstmal
Zweitprüfer: Prof. Dr. Kornelius Hurtig

1.3 Der Grauwert im Fließsatz

Eine Schrift lässt sich nicht an einem Wort beurteilen und auch nicht an einer Zeile. Erst über einen ganzen Absatz hinweg zeigt sich, welchen Grauwert sie erzeugt, wie ruhig die Zeilen laufen und ob die Wortabstände gleichmäßig bleiben. Diese Seite ist deshalb absichtlich dicht gesetzt: Sie soll den Eindruck vermitteln, den eine Seite der fertigen Arbeit machen wird, und nicht den einer Schriftmustertafel.

Der Grauwert entsteht aus dem Zusammenspiel von Strichstärke, x-Höhe, Laufweite und Zeilenabstand. Eine Schrift mit großer x-Höhe wirkt bei gleichem Schriftgrad größer und braucht mehr Zeilenabstand, sonst verkleben die Zeilen. Eine Schrift mit geringer Strichstärke wirkt heller und verträgt engere Zeilen. Beides lässt sich nur im Satz beurteilen, nicht an einer Tabelle von Buchstaben.

1.3.1 Serifen und ihre Wirkung

Serifen sind die kleinen Abschlüsse an den Enden der Buchstaben. Ihnen wird nachgesagt, sie führten das Auge über die Zeile und erhöhten damit die Lesbarkeit im Druck. Ob das messbar stimmt, ist umstritten; sicher ist, dass Serifenschriften im Fließsatz seit Jahrhunderten den Standard bilden und dass Leserinnen und Leser sie deshalb erwarten. In den Ingenieurwissenschaften sind serifenlose Schriften verbreiteter, teils aus Gewohnheit, teils weil sie in technischen Zeichnungen und auf Bildschirmen besser bestehen.

Für eine Abschlussarbeit gilt: Beides ist zulässig, und die Wahl sollte begründet und dann durchgehalten werden. Ein Wechsel mitten im Dokument fällt auf, auch wenn er nicht bewusst wahrgenommen wird. Die Richtlinie der Fakultät schreibt deshalb ausdrücklich eine einzige Schriftart für das ganze Dokument vor, mit Ausnahme der Formeln.

1.3.2 Die Schreibmaschinenschrift

Eine dritte Schrift kommt ins Spiel, sobald Dateinamen, Befehle oder Quelltext vorkommen. Sie wird mit `\texttt{...}` gesetzt und sollte zur Textschrift passen, ohne mit ihr zu verschmelzen: Der Aufruf `lualatex BeispielthesisNXT` muss auf den ersten Blick als Eingabe erkennbar sein, und in `Code/daten.py` darf man die Null in `p_i0 = 0.85` nicht mit dem Buchstaben O verwechseln. Genau daran scheitern viele Schriften. Prüfen lässt es sich an `Il1| O0` – großes I, kleines l, Eins, senkrechter Strich, großes O, Null. Wo diese Zeichen ununterscheidbar sind, taugt die Schrift nicht für Quelltext.

1.3.3 Mathematik und Text

Wo Formeln im Fließtext stehen, entscheidet sich, ob eine Schrift für eine technische Arbeit taugt. Die Mathematikschrift muss zur Textschrift passen: Ein Wert wie $\alpha = 0{,}05$ oder ein Symbol wie $C_{\max}$ steht mitten im Satz und darf dort weder größer noch kleiner noch dunkler wirken als seine Umgebung. Schriftfamilien, die eine eigene Mathematikschrift mitbringen, sind hier im Vorteil – STIX Two, Libertinus und Pagella tun das, und die Klasse bindet sie entsprechend ein.

1 Schriftmuster

Dies ist ein Absatz im Grundtext. Er enthält Umlaute (äöüÄÖÜß), eine *Hervorhebung*, eine **fette Auszeichnung** und `Schreibmaschinenschrift`. Zahlen im Text: 1234567890. Ligaturen: Auffinden, Schifffahrt, Waffel, fünf.

Abkürzungen setzt die Klasse in Kapitälchen, sofern die Schrift echte mitbringt: Mixed Integer Linear Programming (MILP) und Simulated Annealing (SA); im zweiten Gebrauch dann MILP. Fehlen echte Kapitälchen, bleibt es beim normalen Schnitt – aus Versalien heruntergerechnete Kapitälchen wirken zu leicht und fallen im Text unangenehm auf.

1.1 Mathematik

Im Text $C_{\max} = \max_i \sum_j p_{ij} x_{ij}$ und abgesetzt:

$$\mathcal{L}(\mathbf{x}) = \frac{1}{2m} \sum_{i=1}^{m} \bigl(f(\mathbf{u}_i; \mathbf{x}) - y_i\bigr)^2, \qquad \alpha \le \frac{2}{\lambda_{\max}}. \tag{1.1}$$

1.2 Tabellensatz

Verfahren	Makespan	Zeit
Greedy	266,67	0,0
LPT	265,38	0,0
MILP	236,20	640,6

Tabelle 1.1: Zahlen im Tabellensatz

`kapitaelchen` – Titel und Überschriften in Kapitälchen statt fett. Ruhiger als der fette Schnitt, weil Kapitälchen den Grauwert der Seite kaum stören. Diese Anleitung ist so gesetzt. Bei `times` werden die Überschriften stattdessen serifenlos gesetzt, weil die Schrift keine echten Kapitälchen mitbringt; die übrigen elf Optionen haben sie.

Abbildung A.13: Schriftmuster `nxtthesis-stix-kapitaelchen`: Titelseite, Schriftmuster und dicht gesetzte Textseite.

Optimierung des Mitarbeitereinsatzes

Vergleichende Analyse in der metallverarbeitenden Industrie

Arbeit zur Erlangung des akademischen Grades
Bachelor of Science

im Studiengang Nachhaltige Technologie
an der Hochschule Reutlingen

Vorgelegt von:

Jean Dujardin
Matrikelnummer 1234567

Vorgelegt bei:

Prof. Dr. Katharina Erstmal
Prof. Dr. Kornelius Hurtig

Reutlingen, 24. Dezember 2026

1.3 Der Grauwert im Fließsatz

Eine Schrift lässt sich nicht an einem Wort beurteilen und auch nicht an einer Zeile. Erst über einen ganzen Absatz hinweg zeigt sich, welchen Grauwert sie erzeugt, wie ruhig die Zeilen laufen und ob die Wortabstände gleichmäßig bleiben. Diese Seite ist deshalb absichtlich dicht gesetzt: Sie soll den Eindruck vermitteln, den eine Seite der fertigen Arbeit machen wird, und nicht den einer Schriftmustertafel.

Der Grauwert entsteht aus dem Zusammenspiel von Strichstärke, x-Höhe, Laufweite und Zeilenabstand. Eine Schrift mit großer x-Höhe wirkt bei gleichem Schriftgrad größer und braucht mehr Zeilenabstand, sonst verkleben die Zeilen. Eine Schrift mit geringer Strichstärke wirkt heller und verträgt engere Zeilen. Beides lässt sich nur im Satz beurteilen, nicht an einer Tabelle von Buchstaben.

1.3.1 Serifen und ihre Wirkung

Serifen sind die kleinen Abschlüsse an den Enden der Buchstaben. Ihnen wird nachgesagt, sie führten das Auge über die Zeile und erhöhten damit die Lesbarkeit im Druck. Ob das messbar stimmt, ist umstritten; sicher ist, dass Serifenschriften im Fließsatz seit Jahrhunderten den Standard bilden und dass Leserinnen und Leser sie deshalb erwarten. In den Ingenieurwissenschaften sind serifenlose Schriften verbreiteter, teils aus Gewohnheit, teils weil sie in technischen Zeichnungen und auf Bildschirmen besser bestehen.

Für eine Abschlussarbeit gilt: Beides ist zulässig, und die Wahl sollte begründet und dann durchgehalten werden. Ein Wechsel mitten im Dokument fällt auf, auch wenn er nicht bewusst wahrgenommen wird. Die Richtlinie der Fakultät schreibt deshalb ausdrücklich eine einzige Schriftart für das ganze Dokument vor, mit Ausnahme der Formeln.

1.3.2 Die Schreibmaschinenschrift

Eine dritte Schrift kommt ins Spiel, sobald Dateinamen, Befehle oder Quelltext vorkommen. Sie wird mit `\texttt{...}` gesetzt und sollte zur Textschrift passen, ohne mit ihr zu verschmelzen: Der Aufruf `lualatex BeispielthesisNXT` muss auf den ersten Blick als Eingabe erkennbar sein, und in `Code/daten.py` darf man die Null in `p_10 = 0.85` nicht mit dem Buchstaben `O` verwechseln. Genau daran scheitern viele Schriften. Prüfen lässt es sich an `Il1| O0` – großes I, kleines l, Eins, senkrechter Strich, großes O, Null. Wo diese Zeichen ununterscheidbar sind, taugt die Schrift nicht für Quelltext.

1.3.3 Mathematik und Text

Wo Formeln im Fließtext stehen, entscheidet sich, ob eine Schrift für eine technische Arbeit taugt. Die Mathematikschrift muss zur Textschrift passen: Ein Wert wie $\alpha = 0{,}05$ oder ein Symbol wie $C_{\max}$ steht mitten im Satz und darf dort weder größer noch kleiner noch dunkler wirken als seine Umgebung. Schriftfamilien, die eine eigene Mathematikschrift mitbringen, sind hier im

1 Schriftmuster

Dies ist ein Absatz im Grundtext. Er enthält Umlaute (äöüÄÖÜß), eine *Hervorhebung*, eine **fette Auszeichnung** und `Schreibmaschinenschrift`. Zahlen im Text: 1234567890. Ligaturen: Auffinden, Schifffahrt, Waffel, fünf.

Abkürzungen setzt die Klasse in Kapitälchen, sofern die Schrift echte mitbringt: Mixed Integer Linear Programming (MILP) und Simulated Annealing (SA); im zweiten Gebrauch dann MILP. Fehlen echte Kapitälchen, bleibt es beim normalen Schnitt – aus Versalien heruntergerechnete Kapitälchen wirken zu leicht und fallen im Text unangenehm auf.

1.1 Mathematik

Im Text $C_{\max} = \max_i \sum_j p_{ij} x_{ij}$ und abgesetzt:

$$\mathcal{L}(\mathbf{x}) = \frac{1}{2m} \sum_{i=1}^{m} \left(f(\mathbf{u}_i; \mathbf{x}) - y_i \right)^2, \qquad \alpha \le \frac{2}{\lambda_{\max}}. \tag{1}$$

1.2 Tabellensatz

Verfahren	Makespan	Zeit
Greedy	266,67	0,0
LPT	265,38	0,0
MILP	236,20	640,6

Tabelle 1. Zahlen im Tabellensatz

`richtlinie` – die verbindlichen Vorgaben der Fakultät: 11 pt, 1,5-zeilig, Flattersatz, Rand links 3 cm, Überschriften in Textgröße und fett, Beschriftungen 10 pt. Das Ergebnis ist deutlich nüchterner als die Vorgabe der Klasse und braucht spürbar mehr Seiten – es ist aber das, was die Prüfungsordnung verlangt. Im Zweifel gilt diese Option.

Abbildung A.14: Schriftmuster `nxtthesis-stix-richtlinie`: Titelseite, Schriftmuster und dicht gesetzte Textseite.

INDEX