

# Erstellung von leistungsfähigen PDF-Dokumenten mit L<sup>A</sup>T<sub>E</sub>X und den Paketen **hyperref** sowie **thumbpdf**

Sascha Beuermann

[beuermann@ibnm.uni-hannover.de](mailto:beuermann@ibnm.uni-hannover.de)

Institut für Baumechanik und Numerische Mechanik  
Universität Hannover

Version 2.10 vom 15. Januar 2002

## Zusammenfassung

In diesem Artikel werden die Möglichkeiten aufgezeigt, die die Pakete **hyperref** und **thumbpdf** bieten, um mit Hilfe von L<sup>A</sup>T<sub>E</sub>X leistungsfähige Dokumente im Portable Document Format (PDF) zu erzeugen.

Nach einer **Einführung** wird in **Kapitel 2** dargestellt, wie Bitmap-Zeichensätzen in PDF-Dokumenten vermieden werden können. In **Kapitel 3** wird ausführlich auf das **Erzeugen von Lesezeichen** (Bookmarks), auf **PDF-spezifische Anzeigoptionen** sowie auf das **Anlegen von Hypertext-Strukturen** (Verweise auf andere Stellen in demselben Dokument und auch in anderen PDF-Dokumenten sowie Verweise auf beliebige URLs im Internet) mit Hilfe des **hyperref**-Pakets eingegangen. **Kapitel 4** geht auf die Generierung der Seitenvorschau (Piktogramme oder Thumbnails) mit dem **thumbpdf**-Paket ein. Mit der Optimierung und Verschlüsselung von PDF-Dokumenten mit Hilfe des Tools **pdlin** beschäftigt sich das **Kapitel 5**. Abschließend wird in **Kapitel 6** demonstriert, wie aus bestehenden L<sup>A</sup>T<sub>E</sub>X-Dateien einfach und ohne großen Aufwand leistungsfähige PDF-Dokumente erstellt werden können.

Dieser Artikel ist auf der Homepage des Instituts für Baumechanik und Numerische Mechanik (<http://www.ibnm.uni-hannover.de/>) in der aktuellsten Version veröffentlicht.

## Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>4</b>
1.1	Dokumentformate zur digitalen Veröffentlichung . . . . .	4
1.2	Erzeugung von PDF-Dokumenten aus L <sup>A</sup> T <sub>E</sub> X-Dateien . . . . .	5
<b>2</b>	<b>Verwendung von Typ 1 Zeichensätzen</b>	<b>5</b>
2.1	Allgemeine Hinweise zu den verwendeten Zeichensätzen . . . . .	5
2.2	Vorgehensweise bei der Erstellung von PDF-Dokumenten . . . . .	6
<b>3</b>	<b>Das hyperref-Paket</b>	<b>7</b>
3.1	Allgemeine Konfiguration von hyperref . . . . .	7
3.2	Dokumentzusammenfassung . . . . .	7
3.3	Festlegung von Link-Darstellungen . . . . .	9
3.4	Acrobat-Lesezeichen (Bookmarks) . . . . .	11
3.4.1	Zugehörige Paketooptionen . . . . .	11
3.4.2	Erstellung von individuellen Lesezeichen mit dem Befehl <code>\pdfbookmark</code> . . . . .	11
3.4.3	Ersetzen von L <sup>A</sup> T <sub>E</sub> X-Code mit dem Befehl <code>\texorpdfstring</code> . . . . .	12
3.4.4	PD1-Kodierung . . . . .	12
3.4.5	Ersetzen von L <sup>A</sup> T <sub>E</sub> X-Code mit dem hypbmsec-Paket . . . . .	12
3.5	PDF-spezifische Anzeigoptionen . . . . .	13
3.6	Zusätzliche Benutzermakros für Hyperlinks . . . . .	15
3.6.1	Verweise innerhalb des Dokuments . . . . .	15
3.6.2	Verweise auf beliebige URLs im Internet . . . . .	16
3.6.3	Verweise auf andere (PDF-)Dokumente . . . . .	17
3.7	Zugriff auf Menüoptionen des Acrobat Readers . . . . .	17
<b>4</b>	<b>Das thumbpdf-Paket</b>	<b>18</b>
<b>5</b>	<b>Optimierung und Verschlüsselung von PDF-Dokumenten</b>	<b>19</b>
<b>6</b>	<b>Erstellung von PDF-Dokumenten aus bestehenden L<sup>A</sup>T<sub>E</sub>X-Dateien</b>	<b>21</b>
	<b>Literatur</b>	<b>23</b>

## Abbildungsverzeichnis

1	Dokumentinformation, Piktogramme und Lesezeichen . . . . .	4
2	Von $\text{\LaTeX}$ zu PDF . . . . .	5
3	Darstellung von PDF-Dokumenten mit unterschiedlichen Zeichensätzen . . . . .	6
4	Verwendete Zeichensätze im PDF-Dokument . . . . .	6
5	Dokumentzusammenfassung im Acrobat Reader . . . . .	8
6	Verwendung der Option <code>colorlinks</code> . . . . .	10
7	Verwendung der Option <code>pagebackref</code> . . . . .	10
8	Lesezeichen im Acrobat Reader . . . . .	11
9	Piktogramme im Acrobat Reader . . . . .	19
10	Sicherheitsinformation im Acrobat Reader . . . . .	20
11	Von $\text{\LaTeX}$ zu PDF mit dem Paket <code>thumbpdf</code> . . . . .	21

## Tabellenverzeichnis

1	Erweiterungsoptionen für <code>colorlinks</code> . . . . .	9
2	PD1-Kodierung von Sonderzeichen . . . . .	13
3	PDF-spezifische Anzeigoptionen . . . . .	14
4	Makronamen für den <code>\autoref</code> -Befehl von <code>hyperref</code> . . . . .	16
5	Bezeichnungen für die Acrobat-Menüoption-Verknüpfungen . . . . .	18
6	Optionen des Optimierungs- und Verschlüsselungstools <code>pdlin</code> . . . . .	20

# 1 Einführung

## 1.1 Dokumentformate zur digitalen Veröffentlichung

Viele Autoren müssen ihre (z.B. mit  $\text{\LaTeX}$  erstellten) Dokumente sowohl in gedruckter Form als auch in digitaler Form (beispielsweise im Internet) veröffentlichen. Da es auf Dauer nicht nur sehr aufwendig, sondern auch fehlerträchtig ist, zwei Erscheinungsformen ein und desselben Dokuments getrennt zu erstellen und zu pflegen, ist es sinnvoll, auf der Grundlage derselben Dokumentdatei beide Arten von Veröffentlichungsdateien zu erstellen. Für die Druckausgabe bietet sich das PS-Format (PostScript) an, zur Veröffentlichung im Internet die Formate HTML (Hypertext Markup Language) oder PDF (Portable Document Format).

Zur Übertragung der  $\text{\LaTeX}$ -Datei in das HTML-Format kann der  $\text{\LaTeX}$ 2HTML-Übersetzer verwendet werden.  $\text{\LaTeX}$ 2HTML ist ein Konverter, der eine  $\text{\LaTeX}$ -Datei in mehrere miteinander verbundene HTML-Dateien konvertiert. Er ist zwar in der Lage, auch kompliziertere und komplexere  $\text{\LaTeX}$ -Strukturen (wie Gleichungen, Verzeichnisse, Numerierungen, Querverweise, Bilder und Tabellen) korrekt zu bearbeiten. Die meisten der heutigen Browser können jedoch keine mathematische Formeln darstellen. Deshalb wandelt  $\text{\LaTeX}$ 2HTML Zeichen, die nicht zum Standardzeichensatz gehören (mathematischen Zeichen, griechische Buchstaben usw.) in Bitmaps um, die in die HTML-Dateien eingebunden werden. Jede erzeugte HTML-Datei wird zum leichteren Navigieren mit entsprechenden Links versehen.<sup>1</sup> Leider hat der Autor keinen direkten Einfluss auf das äußere Erscheinungsbild seines Dokuments im Browser des Betrachters.

Das Layout eines PDF-Dokuments hingegen ist mit allen typografischen und grafischen Feinheiten unabhängig vom Ausgabemedium. Aus diesem Grund wird das PD-Format insbesondere dann eingesetzt, wenn Dokumente veröffentlicht werden sollen, bei denen das Layout wichtig ist und deren Layout vom Browser des Lesers nicht mehr verändert werden soll. Zudem ist die Dateigröße verhältnismäßig klein, was sich positiv auf den Speicherplatzbedarf und die Übertragungsdauer im Internet auswirkt. Über die einfache Lesefunktion hinaus bietet das PD-Format die Möglichkeit, Hypertext-Strukturen anzulegen: Verweise auf andere Stellen im Dokument, auf andere PDF-Dokumente und auf beliebige Dokumente im Internet. Zusätzlich können Lesehilfen generiert werden: Dokumentinformation, Lesezeichen (Bookmarks) und Seitenübersichtsbilder (Thumbnails oder Piktogramme) wie in [Abbildung 1](#) dargestellt.

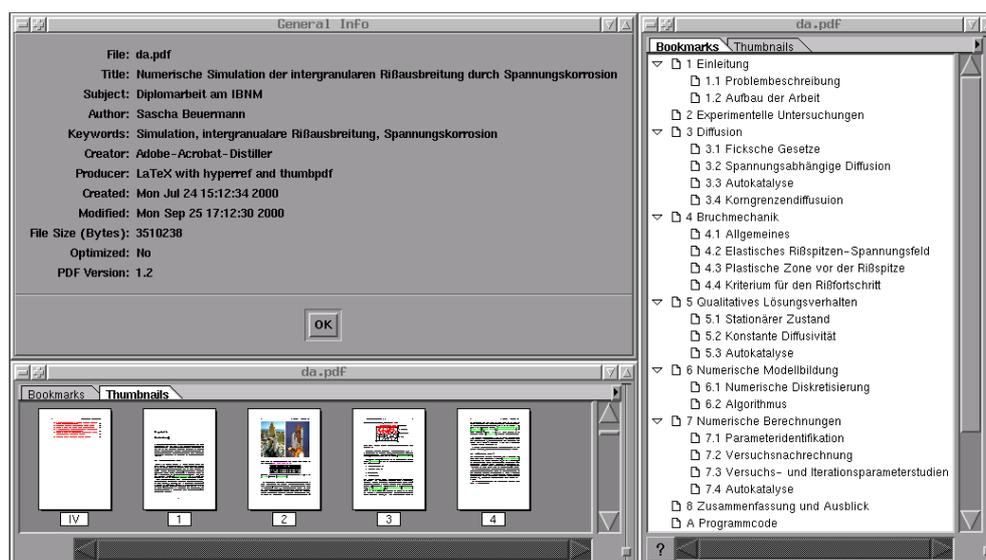


Abbildung 1: Dokumentinformation, Piktogramme und Lesezeichen im Acrobat Reader

<sup>1</sup> Weitere Informationen zu  $\text{\LaTeX}$ 2HTML befinden sich z.B. in [1] Kapitel 3.

Da das PD-Format ein weltweiter und geräteunabhängiger Standard ist, kann das PDF-Dokument überall mit einem geeigneten Viewer wie z.B. mit der Gratissoftware Acrobat Reader von [Adobe](#)<sup>2</sup> betrachtet und gedruckt werden. Über Plugins können PDF-Dokumente auch in den wichtigsten Web-Browsern angezeigt werden.

## 1.2 Erzeugung von PDF-Dokumenten aus L<sup>A</sup>T<sub>E</sub>X-Dateien

Die Erzeugung eines Dokuments mit L<sup>A</sup>T<sub>E</sub>X von der Texteingabe bis zur Druckausgabe ist standardmäßig ein zweistufiger Vorgang, an dessen Ende eine druckfähige PostScript-Datei steht. Zur Erstellung des PD-Formates schließt sich ein weiterer Schritt an, in dem mit einem sogenannten „Distiller“ (wie z.B. dem (kommerziellen) Acrobat Distiller von Adobe, aber auch Ghostscript über `ps2pdf`) aus der PostScript- eine PDF-Datei erzeugt wird. Die einzelnen Verarbeitungsschritte zeigt [Abbildung 2](#).

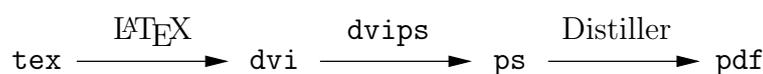


Abbildung 2: Von L<sup>A</sup>T<sub>E</sub>X zu PDF

**Bemerkung zu pdfT<sub>E</sub>X:** Da die Erzeugung in drei Schritten relativ umständlich ist, wurde pdfT<sub>E</sub>X entwickelt, das PDF-Dateien in einem einzigen Durchgang erzeugt.<sup>3</sup>

Ein großes Problem von pdfT<sub>E</sub>X ist jedoch die Einbettung von Grafiken. pdfT<sub>E</sub>X unterstützt nur die Bitmap-Formate PNG, JPEG und TIFF sowie die Vektor-Formate PDF und MPS. Ohne Umweg ist es nicht möglich, (E)PS-Dateien einzubinden, was den Standardfall für L<sup>A</sup>T<sub>E</sub>X-Anwender darstellt. Diese Dateien können zwar relativ einfach in PDF-Dateien umgewandelt werden, es ist aber nur mit Aufwand möglich, mit `Xfig` erzeugte, im Combined PS/L<sup>A</sup>T<sub>E</sub>X-Modus exportierte Bilder einzubinden.

Der größte Nachteil von pdfT<sub>E</sub>X ist jedoch, dass die Erstellung von leistungsfähigen PDF-Dokumenten aus bestehenden L<sup>A</sup>T<sub>E</sub>X-Dateien nur mit großem Aufwand möglich ist.

## 2 Verwendung von Typ 1 Zeichensätzen

### 2.1 Allgemeine Hinweise zu den verwendeten Zeichensätzen

Dokumente, die von L<sup>A</sup>T<sub>E</sub>X über die Formate DVI (DeVice Independent) und PS (PostScript) mittels eines Distillers in eine PDF-Datei (Portable Document Format) konvertiert werden, enthalten i.d.R. Bitmap-Zeichensätze (Typ 3), die nicht geräte- und auflösungsunabhängig wie die skalierbaren Typ 1 Zeichensätze sind und daher vom PDF-Viewer nur in einer schlechten Qualität wiedergegeben werden können. Die unterschiedlichen Darstellungen sind in [Abbildung 3](#) dargestellt.

Besonders wenn PDF-Dokumente auch am Bildschirm gelesen werden sollen, empfiehlt sich wegen der klareren Darstellung die Verwendung von eingebetteten Typ 1 Zeichensätzen, obwohl durch die Einbindung die PDF-Datei größer wird.

Welche Zeichensätze in einem PDF-Dokument verwendet werden, lässt sich durch Anzeige der Dokumentschriften (bzw. Schriftinformation) des PDF-Viewers<sup>4</sup> kontrollieren. In [Abbildung 4](#) sind diese links für ein Dokument mit eingebetteten Typ 1 Zeichensätzen und rechts für ein Dokument mit Typ 3 Bitmap-Zeichensätzen dargestellt.

<sup>2</sup> Kostenlos zu beziehen unter <http://www.adobe.de/products/acrobat/readstep.html>

<sup>3</sup> Informationen über pdfT<sub>E</sub>X sind unter <http://www.tug.org/applications/pdftex/> zu finden.

<sup>4</sup> beim Acrobat Reader im Menü Datei > Dokumenteigenschaften > Schriften

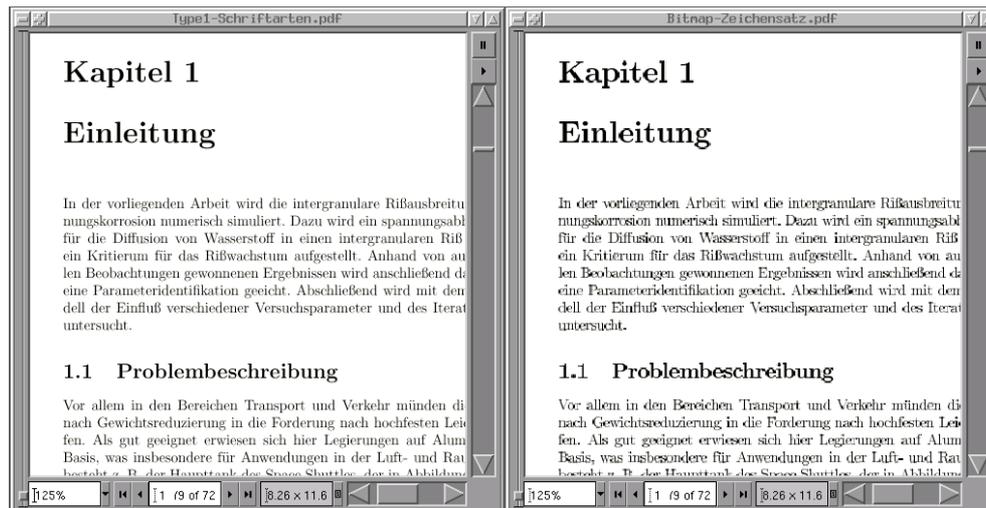


Abbildung 3: Darstellung eines PDF-Dokuments mit eingebettetem Zeichensatz (links) und mit einem Bitmap-Zeichensatz (rechts)

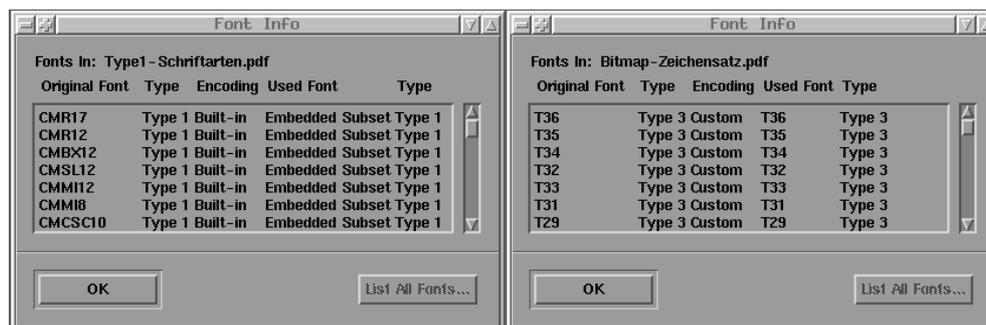


Abbildung 4: Dokumentschriften eines PDF-Dokuments mit eingebettetem Zeichensatz (Typ 1, links) und mit einem Bitmap-Zeichensatz (Typ 3, rechts)

## 2.2 Vorgehensweise bei der Erstellung von PDF-Dokumenten

Zunächst müssen die Zeichensätze in die PS-Datei eingebunden werden. Dies geschieht mit `dvips` und der Option `-Ppdf`. Falls einige Zeichensätze nicht vorhanden sind oder die Option `-Ppdf` nicht unterstützt wird, müssen die Zeichensätze lokal installiert werden. Siehe dazu [10], [9] Kapitel 3, [1] Kapitel 2.2.2 oder [8]. An den Mitteilungen von `dvips` lässt sich erkennen, ob und welche Zeichensätze (.pfd Dateien) in die PS-Datei eingebunden werden:

```
beuermann@ibnm:~ > dvips -Ppdf da.dvi
This is dvips(k) 5.86 Copyright 1999 Radical Eye Software
(www.radical-eye.com)
' TeX output 2001.08.14:1029' -> da.ps
<tex.pro><alt-rule.pro><texc.pro><8r.enc><texps.pro><special.pro>
<color.pro>. <msam10.pfb><cm1110.pfb><cm119.pfb><line10.pfb> ...
<cmr12.pfb><cmr17.pfb>[1<IBNM-logo.color.eps>] [2] [3] [4] [1] ...
```

Anschließend muss die PS-Datei noch mit einem Distiller (z.B. Acrobat Distiller von Adobe oder Ghostscript mit `ps2pdf`) in eine PDF-Datei konvertiert werden. Bei der Verwendung von `ps2pdf` ist darauf zu achten, dass Ghostscript in der Version 6.0 oder höher verwendet wird,

da frühere Versionen die Einbindung der Schriften in die PDF-Datei nicht unterstützen. Falls keine entsprechende Version vorhanden ist,<sup>5</sup> kann auf Online-Distiller zurückgegriffen werden.<sup>6</sup> Aber auch einige Rechenzentren bieten online Konvertierungsmöglichkeiten an.<sup>7</sup>

### 3 Das hyperref-Paket

Das `hyperref`-Paket<sup>8</sup> von Sebastian Rahtz und Heiko Oberdiek erweitert (automatisch) die Funktionalität aller  $\LaTeX$ -Querverweisbefehle (einschließlich Inhaltsverzeichnis, Literaturverzeichnis usw.) und erzeugt `\spezial`-Befehle, die ein Treiber (z.B. `dvips` mit der Option `-z`<sup>9</sup>) in Hypertext-Links umwandeln kann. Außerdem verfügt das Paket über neue Befehle, die es dem Benutzer ermöglichen, Hypertext-Links zu produzieren, einschließlich Verknüpfungen auf externe Dokumente und beliebige Dokumente im Internet.

Da die grundlegenden Angaben zu `hyperref` im Vorspann der  $\LaTeX$ -Datei gemacht werden, ist es ohne weiteres möglich, aus bestehenden  $\LaTeX$ -Dateien leistungsfähige PDF-Dokumente zu erzeugen, wie das Beispiel in [Kapitel 6](#) zeigt.

Die Beschreibung des `hyperref`-Paketes [6] ist aus dem Jahre 1998 und damit etwas veraltet. Eine Beschreibung von `hyperref` mit allen seinen Funktionen findet sich z.B. in [1] Kapitel 2.3 oder [3] Kapitel 5.

Die Funktionen von `hyperref` werden in den folgenden Abschnitten behandelt. Ein ausführlicher Katalog aller Paketooptionen findet sich in [1] Kapitel 2.3.8. Die wichtigsten Optionen daraus werden im Folgenden beschrieben.

Auf die Möglichkeit, mit `hyperref` PDF-Formulare zu erzeugen, soll hier jedoch nicht eingegangen, sondern insofern auf [6] Kapitel 6 und [1] Kapitel 2.3.6 verwiesen werden.

#### 3.1 Allgemeine Konfiguration von hyperref

Das Paket muss im Vorspann der  $\LaTeX$ -Datei angefordert werden. Da es viele  $\LaTeX$ -Befehle neu definiert, sollte es als letztes Paket geladen werden.

Die Paketooptionen von `hyperref` können entweder im optionalen Argument des Befehls `\usepackage` oder in dem (neuen) Befehl `\hypersetup{...}` angegeben werden. Im optionalen Argument sollte aber auf jeden Fall der verwendete Treiber<sup>10</sup> ausgewählt werden:

```
\usepackage[ps2pdf]{hyperref}
```

Mit der Treiber-Option `draft` werden alle Hypertext-Optionen ausgeschaltet, so dass auch Pakete, die den Inhalt der `label`- und `ref`-Makros setzen – wie z.B. `showkeys` –, wieder funktionieren. Mit der Treiber-Option `debug` wird die `log`-Datei um zusätzliche Diagnose-Informationen erweitert.

#### 3.2 Dokumentzusammenfassung

Mit Hilfe des `hyperref`-Paketes können folgende Einträge in der Dokumentzusammenfassung (bzw. Allgemeinen Information)<sup>11</sup> erstellt werden: Titel (`pdftitle`), Thema (`pdfsubject`), Verfasser (`pdfauthor`), Stichwörter (`pdfkeywords`), erstellt mit (`pdfcreator`) und erzeugt mit

<sup>5</sup> wie z.B. bei der Standardinstallation von SuSE 7.0

<sup>6</sup> beispielsweise <http://www.ps2pdf.com/>

<sup>7</sup> z.B. das RRZN unter <http://www.rrzn.uni-hannover.de/pdf/>

<sup>8</sup> hier verwendete Version: `hyperref.sty`, 2000/01/22, v6.69c; das Paket in seiner neusten Version kann von folgender Internetseite bezogen werden:

<http://www.ctan.org/tex-archive/macros/latex/contrib/supported/hyperref/>

<sup>9</sup> Bei Verwendung der Option `-Ppdf` (siehe [Abschnitt 2.2](#)) ist die Option `-z` nicht mehr notwendig.

<sup>10</sup> sowie ggfs. die Option zur „rückwärtigen“ Verknüpfung des Literaturverzeichnisses `backref` bzw. `pagebackref` (siehe [Abschnitt 3.3](#)), und die Optionen `pdfpagelabels` (siehe [Tabelle 3](#)) sowie `extension` (siehe [Unterabschnitt 3.6.3](#)).

<sup>11</sup> beim Acrobat Reader im Menü Datei > Dokumenteigenschaften > Allgemeines

(pdfproducer). **Abbildung 5** zeigt die Dokumentzusammenfassung, die mit der folgenden Eingabe im  $\text{\LaTeX}$ -Dokumentvorspann erzeugt wurde:

```
\hypersetup{%
  pdftitle      = {Numerische Simulation der intergranularen
                  Rißausbreitung durch Spannungskorrosion},
  pdfsubject    = {Diplomarbeit am IBNM},
  pdfauthor     = {Sascha Beuermann},
  pdfkeywords   = {Simulation, intergranulare Rißausbreitung,
                  Spannungskorrosion},
  pdfcreator    = {Adobe-Acrobat-Distiller},
  pdfproducer   = {LaTeX with hyperref and thumpdf}
}
```

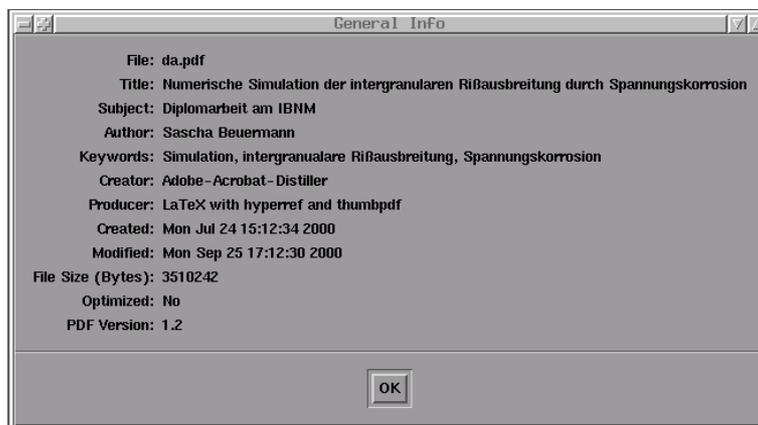


Abbildung 5: Dokumentzusammenfassung im Acrobat Reader

**Bemerkung zu den verwendeten Zeichen:** PDF verwendet eine eigene Zeichenkodierung (PDFDocEncoding) und kann keine  $\text{\LaTeX}$ -Abkürzungen wie beispielsweise `--` für `–` erkennen. Daher sollten bei der Verwendung von Sonderzeichen, Formeln etc. in den Eingaben die Informationen in den **Unterabschnitten 3.4.3 ff** beachtet werden.

**Bemerkung zum Erstellungs- und Änderungszeitpunkt mit psp2df:** Leider gibt es keine Möglichkeit, Erstellungs- und Modifikationszeitpunkt mit dem `hyperref`-Paket zu einstellen. Deshalb muss manuell vor der Erstellung der PDF-Datei die PostScript-Datei editiert und vor dem Eintrag `/Keywords` die folgenden Ausdrücke eingefügt werden, um als Erstellungszeitpunkt beispielsweise den 24. Juli 2000, 15.12:34 Uhr und als Änderungszeitpunkt den 25. September 2000, 17.12:30 Uhr zu erhalten:

```
/CreationDate (D:20000724151234+00'00')
/ModDate (D:20000925171230+00'00')
```

Wird kein Änderungszeitpunkt vorgegeben, wird automatisch der Erstellungszeitpunkt für diesen Eintrag gewählt.

Unter Linux können mittels des folgenden Shell-Skripts `pdftime` Änderungs- und Erstellungszeitpunkt auch automatisch auf die aktuelle Systemzeit gesetzt werden, so dass eine manuelle Bearbeitung nicht mehr notwendig ist [2].

```

#! /bin/bash
if test "$1" = ""; then
    echo 'Usage: pdftime ps-file'
    exit 1
fi
cdate='date +/CreationDate' '\(D:20%y%m%d%H%M%S+00\,'00\,\)'
echo $cdate
mv $1 $1.tmp
sed "s|/Keywords|$cdate /Keywords|" $1.tmp > $1
rm $1.tmp
exit 0

```

Das Skript muss mit `pdftime LaTeXDateiname.ps` aufgerufen werden.

### 3.3 Festlegung von Link-Darstellungen

Das Paket `hyperref` bietet mehrere Optionen, die das Aussehen der erzeugten Verweise beeinflussen.

Im Gegensatz zur Treiberoption `draft`, mit der alle Hypertext-Optionen ausgeschaltet werden, ist es mit der neuen Umgebung `NoHyper` möglich, alle Hypertext-Optionen nur vorübergehend auszuschalten, was wohl aber nur in Ausnahmefällen notwendig ist.<sup>12</sup>

Die Option `breaklinks` ermöglicht einen Zeilenumbruch innerhalb von Verknüpfungstexten. Leider unterstützt der Treiber `dvips` das Umbrechen von Links nicht: Es werden zwar Umbrüche erzeugt, die Link-Bereiche stimmen jedoch nicht mehr, so dass ggfs. selbst Hand angelegt werden muss und zwei Links erstellt werden müssen, die auf dasselbe Ziel verweisen.<sup>13</sup>

Die Option `linktocpage` sorgt dafür, dass nicht der Text, sondern die Seitennummer eines Eintrags im Inhaltsverzeichnis als Hyperlink aktiviert wird (so auch im `Inhaltsverzeichnis` dieses Dokuments).

Mit der Option `colorlinks` wird der Text von Verknüpfungen und Ankern eingefärbt (siehe `Abbildung 6`). Die Wahl der Farbe hängt dabei von der Verknüpfungsart ab, kann jedoch mit den in `Tabelle 1` dargestellten Erweiterungsoptionen verstellt werden, wobei auch in `LATEX` definierte Farben verwendet werden können.<sup>14</sup>

Die Option `frenchlinks` ermöglicht es, den Verknüpfungstext nicht farbig hervorzuheben, sondern in `KAPITÄLCHEN` zu setzen.

Option	Standardfarbe	Farbe für
<code>linkcolor</code>	<code>red</code>	einfache interne Verknüpfungen
<code>anchorcolor</code>	<code>black</code>	Ankertext
<code>citecolor</code>	<code>green</code>	Verweise auf Literaturverzeichnis- einträge im Text
<code>filecolor</code>	<code>magenta</code>	Verknüpfungen, die lokale Dateien öffnen
<code>menucolor</code>	<code>red</code>	Acrobat-Menüpunkte
<code>pagecolor</code>	<code>red</code>	Verknüpfungen zu anderen Seiten
<code>urlcolor</code>	<code>cyan</code>	verknüpfte URLs in einem Netzwerk

Tabelle 1: Erweiterungsoptionen für `colorlinks`

<sup>12</sup> siehe in `Unterabschnitt 3.6.1`

<sup>13</sup> Für das Inhaltsverzeichnis kann auf die Option `linktocpage` ausgewichen werden.

<sup>14</sup> Das zur farbigen Darstellung der Links erforderliche `color`-Paket wird bei der Verwendung von `hyperref` automatisch dazugeladen.

Ferner gibt es im `hyperref`-Paket die Option, im Literaturverzeichnis die einzelnen Einträge mit „rückwärtigen“ Verknüpfungen zu versehen.<sup>15</sup> Die Option `backref` fügt am Ende eines Eintrags eine Rückverknüpfung als Liste von Abschnittsnummern hinzu, die Option `pagebackref` als Liste von Seitennummern. **Abbildung 7** zeigt die Verwendung dieser Funktion.

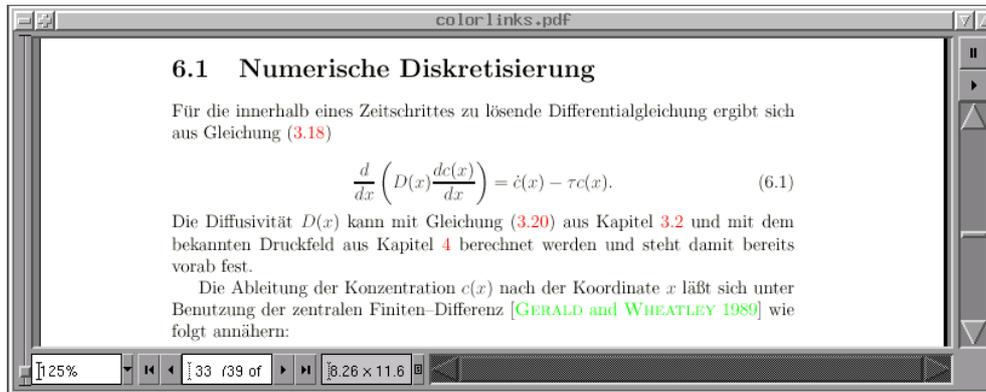


Abbildung 6: Verwendung der Option `colorlinks`

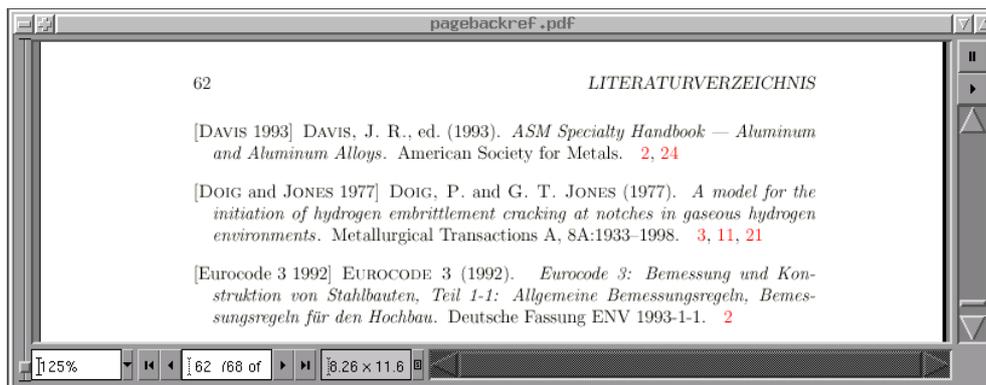


Abbildung 7: Verwendung der Option `pagebackref`

**Bemerkungen zur Option `backref` und `pagebackref`:** Der Eintrag der Option sollte im optionalen Argument des Befehls `\usepackage` und nicht im Befehl `\hypersetup` stehen.<sup>16</sup> Die beiden Optionen arbeiten nur dann korrekt, wenn nach jedem `\bibitem`-Eintrag in der `bb1`-Datei eine Leerzeile eingefügt wird, was bei der Erstellung mit `BibTeX` der Fall ist.

**Bemerkung zur Verwendung von `BibTeX` und dem `url`-Paket:** `BibTeX` erzeugt bei langen URLs z.T. Zeilenumbrüche mit dem  $\LaTeX$ -Kommentarzeichen `%`, was bei der Umwandlung zu Hyperlinks in PDF mit dem (automatisch zugeladenen) Paket `url` zu Fehlern führt. Deshalb ist die Datei `url.sty`<sup>17</sup> bis zum Erscheinen einer entsprechenden Version um die folgenden Zeilen zu ergänzen:<sup>18</sup>

<sup>15</sup> hier verwendete Version: `backref.sty`, 2000/01/19, v1.19

<sup>16</sup> siehe dazu auch **Abschnitt 3.1**

<sup>17</sup> hier verwendete Versionen: `BibTeX` (Web2C 7.3.1) 0.99c; `url.sty`, 1999/03/02, ver 1.4

<sup>18</sup> Mitteilung von Donald Arseneau, dem Autor des `url`-Pakets, vom 2. August 2001 in der [Newsgroup comp.text.tex von Google Groups](#)

```

\begingroup
\makeatletter
\g@addto@macro{\UrlSpecials}{%
  \endlinechar=13 \catcode\endlinechar=12
  \do\%{\Url@percent}\do\^~M{\break}}
\catcode13=12 %
\gdef\Url@percent{\@ifnextchar^~M{\@gobble}{%
  \mathbin{\mathchar'\%}}}%
\endgroup %

```

### 3.4 Acrobat-Lesezeichen (Bookmarks)

#### 3.4.1 Zugehörige Paketooptionen

Lesezeichen (oder Bookmarks) sind eine Navigationshilfe im Acrobat Reader (siehe [Abbildung 8](#)). Mit der Paketooption `bookmarks` werden Acrobat-Lesezeichen in ähnlicher Weise wie das Inhaltsverzeichnis erstellt. Das `hyperref`-Paket schreibt automatisch Lesezeichen-Code für die Gliederungsbefehle `\section`, `\subsection` usw. in die Hilfsdatei `LaTeXDateiname.out`, die bei Bedarf auch von Hand nachbearbeitet werden kann.<sup>19</sup> Mit der (alternativen) Option `bookmarksopen` werden bei der Anzeige der Lesezeichen alle Ebenen angezeigt. Die Tiefe, bis zu der die Lesezeichen geöffnet werden, kann aber auch mit `bookmarksopenlevel={number}` vorgegeben werden. Durch die (zusätzliche) Option `bookmarksnumbered` werden die Lesezeichen zusammen mit den Abschnittsnummern angezeigt. [Abbildung 8](#) wurde beispielsweise mit den Paketooptionen `bookmarksopen` und `bookmarksnumbered` erstellt.

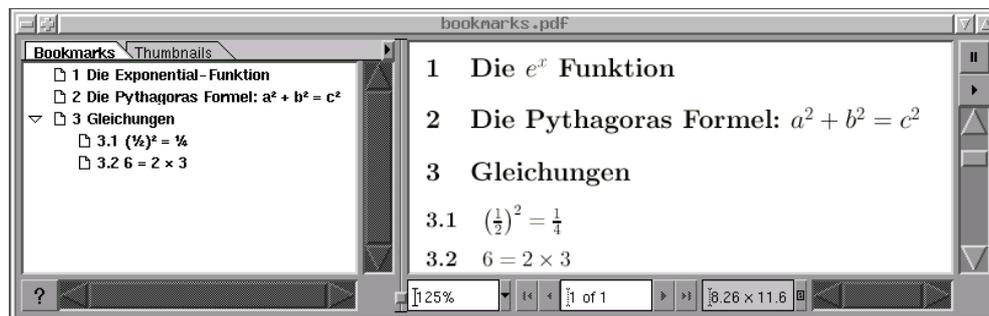


Abbildung 8: Lesezeichen im Acrobat Reader

#### 3.4.2 Erstellung von individuellen Lesezeichen mit dem Befehl `\pdfbookmark`

Der Befehl `\pdfbookmark` macht es möglich, eigene Lesezeichen unabhängig von Gliederungsbefehlen zu erstellen [3]. Dazu muss an den entsprechenden Stellen mit dem Befehl `\hypertarget{Marke}{}` (siehe [Unterabschnitt 3.6.1](#)) eine Marke erstellt werden, auf die sich dann das mit

```
\pdfbookmark[level]{Lesezeichentext}{Marke}
```

erstellte Lesezeichen bezieht. Mit dem optionalem Argument `level` ist es möglich, die Gliederungstiefe des Lesezeichens mit Werten zwischen `-1` und `5` vorzugeben. Der Wert `-1` entspricht der Gliederungsebene `\part`, `0` entspricht `\chapter`, `1` entspricht `\section` usw.

Auf diese Weise kann z.B. in der Dokumentenklasse `article` mit

<sup>19</sup> Wie zur ordnungsgemäßen Erstellung des Inhaltsverzeichnisses werden auch hier zwei Durchläufe von  $\text{\LaTeX}$  benötigt.

```
\pdfbookmark[1]{\contentsname}{toc}
\tableofcontents
```

das Inhaltsverzeichnis zu den Lesezeichen hinzugefügt werden.<sup>20</sup>

### 3.4.3 Ersetzen von $\LaTeX$ -Code mit dem Befehl `\texorpdfstring`

Das Paket `hyperref` versucht zwar, die interne Kodierung europäischer Zeichen nach PDFDocEncoding zu konvertieren, das von Acrobat in Lesezeichen verwendet wird, der Lesezeichentext wird jedoch nicht von  $\LaTeX$  bearbeitet, so dass jede Auszeichnung unverändert übernommen wird.

Einige Probleme, die durch die Einschränkung von PDFDocEncoding entstehen, wie z.B. dass keine Formeln angezeigt werden können, lassen sich mit dem Makro

```
\texorpdfstring{LaTeX-Zeichenfolge}{PDF-Zeichenfolge}
```

umgehen. Dabei wird im Gliederungsbefehl bei der Erzeugung eines Lesezeichens nicht die LaTeX-Zeichenfolge, sondern die PDF-Zeichenfolge verwendet. Die Auswirkung des Befehls

```
\section{Die \texorpdfstring{ex}-Exponential-Funktion}
```

ist in [Abbildung 8](#) gezeigt.

### 3.4.4 PD1-Kodierung

Mit dem `pd1enc`-Paket<sup>21</sup>, das von `hyperref` automatisch dazugeladen wird, ist es außerdem möglich, PDFDocEncoding-Zeichen durch Verwendung sogenannter PD1-Kodierungen zu benutzen. Beispielsweise ergibt

```
\section{Die Pythagoras Formel: \texorpdfstring{\$a^2+b^2=c^2\$}%
  {a\textttwosuperior\ + b\textttwosuperior\ %
   = c\textttwosuperior\ }}}
```

das in [Abbildung 8](#) dargestellte Lesezeichen. Kodierungen für andere gebräuchliche Sonderzeichen sind in [Tabelle 2](#) zusammengestellt.<sup>22</sup> Eine komplette Liste der Kodierungen der Sonderzeichen ist in [7] zu finden. Genauere Informationen zu der Zeichenkodierung in PDF enthalten z.B. [3] Kapitel 2 und 3.

### 3.4.5 Ersetzen von $\LaTeX$ -Code mit dem `hypbmsec`-Paket

Eine andere (und bequemere) Möglichkeit,  $\LaTeX$ -Code zu ersetzen, bietet die Erweiterung der Gliederungsbefehle durch das Paket `hypbmsec`<sup>23</sup> von Heiko Oberdiek [4], [3]: Ein zweites (optionales) Argument in eckigen Klammern wird nicht nur für den Eintrag im Inhaltsverzeichnis und ggf. im Seitenkopf, sondern auch als Zeichenfolge für das Lesezeichen verwendet. Durch ein weiteres (optionales) Argument in runden Klammern kann aber auch die Zeichenfolge für das Lesezeichen unabhängig vom Eintrag im Inhaltsverzeichnis angegeben werden. Die folgenden Beispiele verdeutlichen die Kombinationsmöglichkeiten der Optionen für die Gliederungsbefehle am Beispiel von `\section`:

<sup>20</sup> Die Marke `toc` wird – wie auch die Marken `lof` und `lot` für das Abbildungs- und Tabellenverzeichnis – automatisch gesetzt.

<sup>21</sup> hier verwendete Version: `pd1enc.def`, 1999/10/05, v0.6

<sup>22</sup> Hinweis: Der Acrobat Reader 4.05 unter Linux zeigt einige Sonderzeichen wie z.B. `\textbullet`, `\quotedblbase` und `\textquotedblleft` in den Lesezeichen nicht an.

<sup>23</sup> hier verwendete Version: `hypbmsec.sty`, 1999/04/12, v2.0

Sonderzeichen	PD1-Kodierung
•	<code>\textbullet</code>
©	<code>\textcopyright</code>
&	<code>\textampersand</code>
\	<code>\textbackslash</code>
~	<code>\textasciitilde</code>
„	<code>\quotedblbase</code>
“	<code>\textquotedblleft</code>
°	<code>\textdegree</code>
1	<code>\textonesuperior</code>
2	<code>\texttwosuperior</code>
3	<code>\textthreesuperior</code>
(	<code>\textparenleft</code>
)	<code>\textparenright</code>
±	<code>\textplusminus</code>
×	<code>\textmultiply</code>
÷	<code>\textdivide</code>
1/2	<code>\textonehalf</code>
1/4	<code>\textonequarter</code>

Tabelle 2: PD1-Kodierung von Sonderzeichen

```

\section[Inhaltsver./Seitenkopf = Lesezeichen]{Text}
\section(Lesezeichen)[Inhaltsver./Seitenkopf]{Text}
\section(Lesezeichen){Inhaltsver./Seitenkopf = Text}
\section{Inhaltsver./Seitenkopf = Lesezeichen = Text}
\section{Inhaltsver./Seitenkopf = Lesezeichen = Text}

```

Da das Paket `hypbmsec` die Syntax der Gliederungsbefehle ändert, sollte es mit

```
\usepackage{hypbmsec}
```

als letztes Paket (hinter `hyperref`) geladen werden. Es funktioniert daher auch nicht zusammen mit Paketen, die ebenfalls die Syntax der Gliederungsbefehle ändern. Wenn das `hyperref`-Paket nicht geladen ist, ignoriert `hypbmsec` die Lesezeichen, so dass derselbe Text für unterschiedliche Ausgaben verwendet werden kann.

Die Auswirkung des Befehls

```

\subsection(\textparenleft \textonehalf \textparenright
\texttwosuperior\ = \textonequarter)%
{\$\left(\frac{1}{2}\right)^2 = \frac{1}{4}$}

```

ist in [Abbildung 8](#) gezeigt.

### 3.5 PDF-spezifische Anzeigoptionen

Die wichtigsten Optionen, die zur individuellen Anpassung der Anzeige des PDF-Dokuments im Viewer dienen, sind in [Tabelle 3](#) aufgeführt. Weitere Optionen sind sehr übersichtlich in [1] Kapitel 2.3.8 dargestellt.

Option	Beschreibung	Mögliche Werte
<code>pdfcenterwindow</code>	Legt fest, ob der Viewer das Fenster zentriert, in dem das Dokument erscheint.	<code>false*</code> , <code>true</code>
<code>pdffitwindow</code>	Legt fest, ob der Viewer das Fenster, in dem das Dokument erscheint, auf die Größe der ersten angezeigten Seite des Dokuments anpasst.	<code>false*</code> , <code>true</code>
<code>pdfhighlight</code>	Legt fest, wie sich Verknüpfungsschaltflächen verhalten, wenn sie angewählt werden.	<code>/I</code> (Invertierung)* <code>/N</code> (kein Effekt) <code>/O</code> (Umrandung) <code>/P</code> („gedrückter Kopf“)
<code>pdfmenubar</code>	Legt fest, ob die Menüzeile des Viewers sichtbar ist.	<code>false*</code> , <code>true</code>
<code>pdfnewwindow</code>	Sorgt dafür, dass Verweise auf ein anderes PDF-Dokument ein neues Fenster öffnen.	<code>false*</code> , <code>true</code>
<code>pdfpagelabels</code>	Sorgt dafür, dass anstelle von physikalischen Seitennummern logische angezeigt werden, so dass die Piktogramme auch mit römischen Seitennummern versehen sind, siehe <a href="#">Abbildung 9</a> . Diese Option sollte im optionalen Argument des Befehls <code>\usepackage</code> und nicht im Befehl <code>\hypersetup</code> stehen.	<code>false*</code> , <code>true</code>
<code>pdfpagelayout</code>	Bestimmt das Layout für die Seite beim Öffnen des Dokuments.	<code>SinglePage*</code> , <code>OneColumn</code> , <code>TwoColumnLeft</code> , <code>TwoColumnRight</code>
<code>pdfpagemode</code>	Legt fest, wie die Datei in Acrobat geöffnet wird. Wenn kein Modus ausgewählt ist, aber die Option <code>bookmarks</code> gesetzt ist, wird <code>UseOutlines</code> verwendet.	<code>None*</code> , <code>UseThumbs</code> , <code>UseOutlines</code> , <code>FullScreen</code>
<code>pdfstartpage</code>	Bestimmt die Seitennummer der Seite, auf welcher die PDF-Datei geöffnet wird.	<code>1*</code> , sonstige Seitenzahlen
<code>pdfstartview</code>	Bestimmt die Anzeigegröße beim Öffnen des Dokuments.	<code>Fit*</code> , <code>FitB</code> , ... (siehe [1] Tabelle 2.1)
<code>pdftoolbar</code>	Legt fest, ob die Symbolleiste des Viewers sichtbar ist.	<code>false*</code> , <code>true</code>
<code>plainpages</code>	Zwingt Seitenanker, nicht durch die formatierten, sondern durch arabische Seitennummern benannt zu werden.	<code>false</code> , <code>true*</code>

Tabelle 3: PDF-spezifische Anzeigoptionen (\*: Standardwert) aus [1]

### 3.6 Zusätzliche Benutzermakros für Hyperlinks

Das `hyperref`-Paket stellt einige Makros für Verweise auf URLs (Universal Resource Locators) beliebiger Internetseiten und andere Dokumente sowie für Verknüpfungen innerhalb des Dokuments zur Verfügung, von denen die wichtigsten im Folgenden hier kurz erläutert werden.<sup>24</sup>

#### 3.6.1 Verweise innerhalb des Dokuments

Mit dem Makro

```
\hyperref[Marke]{Text}
```

wird der `Text` als Link zu einem Punkt aktiviert, der mit dem normalen  $\LaTeX$ -Befehl `\label` mit dem symbolischen Namen `Marke` erstellt wurde. Da dieser Befehl nur nach einem Gliederungs-befehl oder in `equation`-, `eqnarray`-, `figure`-, `table`- oder `enumerate`-Umgebungen benutzt werden kann, gibt es in `hyperref` auch die beiden folgenden Befehle:

```
\hypertarget{Marke}{Text}
\hyperlink{Marke}{Link-Text}
```

Der Befehl `\hypertarget` kann benutzt werden, um `Text` mit einer `Marke` zu markieren. Mit `\hyperlink` kann dann auf diese `Marke` mit `Link-Text` verwiesen werden. Auf mit `\label` markierte Stellen kann mit `\hyperlink` nicht verwiesen werden.

Der Befehl

```
\autoref{Marke}
```

dient als Ersatz für den normalen `\ref`-Befehl und fügt vor jedem Verweis eine verknüpfungs-abhängige Kennung ein. Der Unterschied wird bei den folgenden Verweisen auf **Kapitel 1** deutlich: `\ref{Marke}` erzeugt als Verknüpfung „1“, während `\autoref{Marke}` „Kapitel 1“ erzeugt. Im ersten Fall besteht der Verweis nur aus der Zahl, im zweiten Fall ist die Kennung der Marke (hier: Kapitel) Teil der Verknüpfung. Die zweite Variante ist i.d.R. benutzerfreundlicher als die erste. Die Kennung wird von `hyperref` mit Hilfe der in **Tabelle 4** aufgeführten Makros aus dem Kontext des ursprünglichen `\label`-Befehls erarbeitet. Die Makros können mit dem Befehl `\renewcommand` in Dokumenten neu definiert werden, wobei auf die Groß- und Kleinschreibung der Anfangsbuchstaben des Makros zu achten ist:

```
\renewcommand{\sectionname}{Kapitel}
```

Sollen eigene Verknüpfungstexte – und nicht nur mit `\ref` oder `\pageref` erstellte Verknüpfungen, die nur aus einer Zahl bestehen, – erstellt werden, bietet sich die Verwendung der Befehle `\ref*{Marke}` oder `\pageref*{Marke}` in Verbindung mit `\hyperref` an, wie folgendes Beispiel zeigt:

```
\hyperref[Marke]{Kap.~\ref*{Marke}, S.~\pageref*{Marke}}
```

erzeugt „Kap. 1, S. 4“, wobei `\ref*{Marke}` und `\pageref*{Marke}` die richtige Nummer, jedoch keine Verknüpfung generiert. Diese wird mit dem Befehl `\hyperref` erzeugt.

Mit Hilfe des `nameref`-Pakets<sup>25</sup>, das automatisch mit `hyperref` geladen wird, ermöglicht der Befehl

```
\nameref{Marke}
```

auf Gliederungsabschnitte unter ihrer Bezeichnung zu verweisen. Leider gibt es anders als bei `\ref` und `\pageref` keine „\*-Version dieses Befehls. Soll keine Verknüpfung hergestellt werden, muss die `NoHyper`-Umgebung verwendet werden. So wird beispielsweise mit

```
\autoref{Marke} \begin{NoHyper} \nameref{Marke} \end{NoHyper}
```

„Kapitel 1 Einführung“ erzeugt.

<sup>24</sup> Die übrigen Makros sind in [1] Kapitel 2.3.3 und [6] Kapitel 3 beschrieben.

<sup>25</sup> hier verwendete Version: `nameref.sty`, 2000/01/19, v2.17

Makronamen	Standardbezeichnung
<code>\figurename</code>	Abbildung
<code>\tablename</code>	Tabelle
<code>\partname</code>	Teil
<code>\appendixname</code>	Anhang
<code>\equationname</code>	Gleichung
<code>\Itemname</code>	Punkt
<code>\chaptername</code>	Kapitel
<code>\sectionname</code>	Abschnitt
<code>\subsectionname</code>	Unterabschnitt
<code>\subsubsectionname</code>	Unter-Unterabschnitt
<code>\paragraphname</code>	Absatz
<code>\Hfootnotename</code>	Fußnote
<code>\AMSname</code>	Gleichung
<code>\theoremname</code>	Theorem

Tabelle 4: Makronamen für den `\autoref`-Befehl von `hyperref` (aus [1])

### 3.6.2 Verweise auf beliebige URLs im Internet

Für Verweise auf URLs werden die Befehle

```
\href{URL}{Text}
\url{URL}
```

verwendet. Beim `href` wird der `Text` in einen Hyperlink auf die `URL` umgewandelt, wobei die `URL` vollständig sein muss. Die Sonderzeichen `#` und `~` in `URL` müssen nicht maskiert werden. Um gleichzeitig einen Hyperlink zu erzeugen und die Adresse anzuzeigen, dient der Befehl `\url`. So ergibt beispielsweise

```
\href{http://www.ibnm.uni-hannover.de}{Homepage des Instituts für
Baumechanik und Numerische Mechanik}
```

„[Homepage des Instituts für Baumechanik und Numerische Mechanik](http://www.ibnm.uni-hannover.de)“ und

```
\url{http://www.ibnm.uni-hannover.de}
```

„<http://www.ibnm.uni-hannover.de>“. Mit Hilfe dieser Befehle können auch Hyperlinks auf E-Mail-Adressen erzeugt werden. So ergibt

```
\href{mailto:beuermann@ibnm.uni-hannover.de}{S. Beuermann}
```

„[S. Beuermann](mailto:beuermann@ibnm.uni-hannover.de)“ und

```
\url{mailto:beuermann@ibnm.uni-hannover.de}
```

„<mailto:beuermann@ibnm.uni-hannover.de>“, wobei in diesem Dokument mit den Optionen

```
colorlinks=true,
urlcolor=webbrown
```

die Farbe für Verweise auf URLs auf die benutzerdefinierte Farbe `webbrown` umgestellt wurde, siehe auch [Abschnitt 3.3](#).

### 3.6.3 Verweise auf andere (PDF-)Dokumente

Das Ergänzungspaket `xr-hyper`<sup>26</sup> ermöglicht es, zusammen mit `hyperref` auf Markierungen, die in externen  $\text{\LaTeX}$ -Dateien mit `\label` erstellt wurden, zu verweisen.<sup>27</sup> Um auf andere PDF-Dokumente zu verweisen, muss die Option `extension=pdf` (im Vorspann, siehe [Abschnitt 3.1](#)) gesetzt werden. Außerdem müssen diese Dokumente im Vorspann mit dem Befehl

```
\externaldocument[Abkürzung]{Dateiname}
```

„angekündigt“ werden. Der (optionale) Parameter `Abkürzung` dient dazu Fehler zu vermeiden, die auftreten, wenn Markierungen in den unterschiedlichen Dateien dieselben Bezeichnungen haben.

Sieht der Vorspann z.B. wie folgt aus,

```
\usepackage{xr-hyper}
\usepackage[ps2pdf,extension=pdf]{hyperref}
\externaldocument[A-]{LaTeXDateiname1}
\externaldocument[B-]{LaTeXDateiname2}
```

kann auf die Markierung `Marke` im Dokument `LaTeXDateiname1.pdf` mit

```
\ref{A-Marke}
```

verwiesen werden. Entsprechendes gilt natürlich für die Befehle `\ref*`, `\pageref`, `\pageref*`, `\autoref` und `\hyperref`.

Auf Markierungen zu verweisen, die in externen  $\text{\LaTeX}$ -Dateien mit `\hypertarget` erstellt wurden, ist auch ohne Verwendung des `xr-hyper`-Pakets mit Hilfe des in [Unterabschnitt 3.6.2](#) eingeführten Befehls `\href` möglich. Auf die Markierung `Marke` im Dokument `LaTeXDateiname1.pdf` kann mit

```
\href{LaTeXDateiname1.pdf#Marke}{Text}
```

verwiesen werden. Das `#`-Symbol dient dabei zur Trennung von Zieldatei und Markierung. Auf ähnliche Weise kann so auch auf Gliederungspunkte verwiesen werden. Mit

```
\href{LaTeXDateiname2.pdf#subsection.2.1}{Text}
```

wird auf Abschnitt 2.1 in der Datei `LaTeXDateiname2.pdf` verwiesen. Auskünfte über die Bezeichnungen der Gliederungspunkte wie `subsection.2.1` gibt die jeweilige `.out` Datei des externen Dokuments.

## 3.7 Zugriff auf Menüoptionen des Acrobat Readers

Für Zugriffe auf Menüoptionen des Acrobat Readers steht der Befehl

```
\Acrobatmenu{Menüoption}{Text}
```

zur Verfügung. Der `Text` wird verwendet, um eine Schaltfläche zu erstellen, welche die entsprechende `Menüoption` aktiviert. In [Tabelle 5](#) sind die wichtigsten Menüoptionen aufgeführt.<sup>28</sup> Beispielsweise erzeugen die Befehle

```
\Acrobatmenu{GeneralInfo}{\fbox{Dokumentzusammenfassung}}
\Acrobatmenu{FontsInfo}{\fbox{Dokumentschriften}}
```

<sup>26</sup> hier verwendete Version: `xr-hyper.sty`, 1999/09/01, v6.00beta3

<sup>27</sup> Für die Verwendung von `hyperref` ist es notwendig, das Paket `xr-hyper` und nicht das (ursprüngliche) `xr`-Paket zu verwenden. Außerdem muss in allen verwendeten Dokumenten das `hyperref`-Paket geladen werden.

<sup>28</sup> Eine Liste aller zur Verfügung stehenden `Menüoptionen` ist in [\[6\]](#) Kapitel 4 und [\[1\]](#) Kapitel 2.3.4 zu finden.

folgende Schaltflächen:<sup>29</sup>

Dokumentzusammenfassung

Dokumentschriften

Werden Schaltflächen aus dem Acrobat-Menü Bearbeiten auf jeder Dokumentseite erzeugt, lässt sich z.B. für Präsentationen eine komfortable Navigationleiste im Dokument erstellen.<sup>30</sup>

Acrobat-Menü	Option	Menüoption
Datei	Öffnen	Open
Datei	Schließen	Close
Datei	Drucken	Print
Datei	Dokumentzusammenfassung	GeneralInfo
Datei	Dokumentschriften	FontsInfo
Datei	Seiteneinrichtung	PageSetup
Datei	Acrobat Reader beenden	Quit
Bearbeiten	Suchen im Dokument	Find
Bearbeiten	Gehe zur ersten Seite	FirstPage
Bearbeiten	Gehe zur vorherigen Seite	PrevPage
Bearbeiten	Gehe zur nächsten Seite	NextPage
Bearbeiten	Gehe zur letzten Seite	LastPage
Bearbeiten	Gehe zu Seite ...	GoToPage
Anzeige	Vollbild	FullScreen
Anzeige	Ganze Seite	FitPage
Anzeige	Originalgröße	ActualSize
Anzeige	Fensterbreite	FitWidth
Anzeige	Seitenbreite	FitVisible
Anzeige	Einzelne Seite	SinglePage
Anzeige	Fortlaufende Seiten	OneColumn
Anzeige	Fortlaufende Doppelseiten	TwoColumns
Fenster	Lesezeichen anzeigen	ShowBookmarks
Fenster	Piktogramme anzeigen	ShowThumbs
Fenster	(Nur) Seiten anzeigen	PageOnly
Fenster	Werkzeugleiste ein-/ausblenden	ShowHideToolBar
Fenster	Menüleiste ein-/ausblenden	ShowHideMenuBar

Tabelle 5: Bezeichnungen für die Acrobat-Menüoption-Verknüpfungen (aus [1])

## 4 Das thumbpdf-Paket

Das Paket `thumbpdf`<sup>31</sup> von Heiko Oberdiek erzeugt mit Hilfe des `perl`-Programms `thumbpdf`, das wiederum `Ghostscript` verwendet, Thumbnails (Piktogramme) für  $\text{\LaTeX}$ -Dateien, die zu PDF-Dateien konvertiert werden [5], [3]. Thumbnails sind eingebundene Aufnahmen der Dokumentseiten in geringer Auflösung in Größe eines Daumennagels, die die Navigation durch das Dokument vereinfachen, sofern der PDF-Viewer dies unterstützt (z.B. der Acrobat Reader).

<sup>29</sup> Das Verhalten der Verknüpfungsschaltflächen beim Anwählen kann mit der Option `pdfhighlight` festgelegt werden, siehe [Tabelle 3](#).

<sup>30</sup> Beispiele zur Erstellung von ansprechenden Schaltflächen finden sich z.B. in [9] Kapitel 6.8. Für eine Bildschirmpräsentation sollte das PDF-Dokument im Querformat erstellt werden, was durch Einfügung folgender `\special`-Befehle in die  $\text{\LaTeX}$ -Datei und entsprechende Seitenformatierungsanweisungen möglich ist: `\special{landscape}` und `\special{! TeXDict begin /landplus90{true}store end }`

<sup>31</sup> hier verwendete Version: `thumbpdf.sty`, 2001/04/02, v2.10

Zur Generierung der Bilder ist Ghostscript mindestens in der Version 5.50 erforderlich, zum Einbinden der Bilder mit ps2pdf mindestens die Version 6.0.<sup>32</sup> Für die Benutzung mit ps2pdf muss das thumbpdf-Paket in die L<sup>A</sup>T<sub>E</sub>X-Datei wie folgt eingebunden werden:

```
\usepackage[ps2pdf]{thumbpdf}
```

Die PDF-Datei mit Piktogrammen wird dann in drei Schritten erstellt:

Zunächst ist aus der L<sup>A</sup>T<sub>E</sub>X-Datei wie gewöhnlich eine (jedoch nur vorläufige) PDF-Datei zu erstellen. Anschließend muss das perl-Programm thumbnail mit

```
thumbpdf --modes=ps2pdf LaTeXDateiname.pdf
```

aufgerufen werden, das die Thumbnails erstellt und in der Datei LaTeXDateiname.tpt speichert. Abschließend muss L<sup>A</sup>T<sub>E</sub>X erneut aufgerufen werden und mittels dvips -Ppdf und ps2pdf eine neue (die endgültige) PDF-Datei erstellt werden. **Abbildung 9** zeigt ein (fertiges) PDF-Dokument mit Piktogrammen, das mit der Option pdfpagelabels erstellt wurde (siehe **Abschnitt 3.5**). Es ist auch möglich, anstelle der verkleinerten Seitenbilder andere Bilder als Piktogramme zu verwenden, siehe dazu [3] Kapitel 8.4.

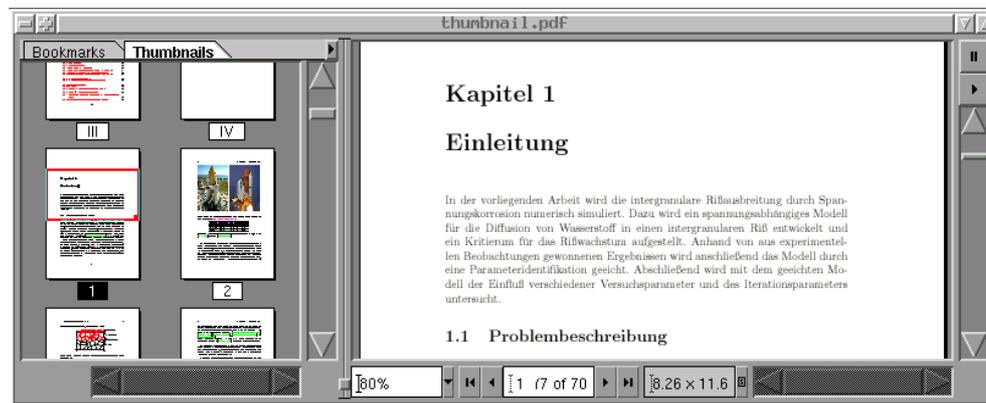


Abbildung 9: Piktogramme im Acrobat Reader

## 5 Optimierung und Verschlüsselung von PDF-Dokumenten

PDF-Dokumente, die auf Webservern veröffentlicht werden, sollten auf die Ansicht in einem Browser optimiert werden. Nicht-Optimierte Dateien müssen zunächst gänzlich heruntergeladen werden, bevor sie dargestellt werden können, während bei optimierten Dateien nur ein Teil zur Darstellung ausreicht.

Das Optimieren einer PDF-Datei bedeutet jedoch nicht, dass die Dateigröße kleiner wird. Im Gegenteil, sie wird durch die Optimierung (oder auch Linearisierung genannt) eher größer.

Z.B. mit Hilfe des Tools pdlin<sup>33</sup> von Glance können optimierte Dateien aus einer bestehenden PDF-Datei erstellt werden. Es ist außerdem möglich, verschlüsselte Dateien als Eingabe-Datei zu lesen und/oder die optimierte Ausgabe-Datei entsprechend der Standard-PDF-Verschlüsselung zu verschlüsseln.<sup>34</sup> In der Dokumentzusammenfassung des Acrobat Readers befindet sich der Hinweis, ob die PDF-Datei optimiert ist (siehe **Abbildung 5**).

<sup>32</sup> wie auch zur Einbindung der Typ 1 Zeichensätze, siehe **Abschnitt 2.2**

<sup>33</sup> Download der frei verfügbaren Version 2.02 vom 14. August 2001 (für Windows, Linux und Solaris) unter <http://pdf.glance.ch/eval/>; die freie Verwendung beschränkt sich jedoch auf Evaluationszwecke, für den produktiven Betrieb ist das Tool zu lizenzieren.

<sup>34</sup> Bei größeren PDF-Dateien treten jedoch Probleme beim Optimieren mit pdlin auf (Lesezeichen und Links funktionieren nicht mehr), so dass auch dieses Dokument nicht optimiert werden konnte.

In der Sicherheitsinformation<sup>35</sup> befinden sich Hinweise zum Sicherheitsmodus (Keiner, Standard), zu den Kennwörtern zum Öffnen der Datei und für die Sicherheitsoptionen (Ja, Nein). Die Sicherheitsoptionen (Zulässig, Nicht zulässig) sind Drucken, Dokument ändern, Auswählen von Text und Grafiken sowie Anmerkungen und Formularfelder hinzufügen/ändern, siehe [Abbildung 10](#). Diese Einstellungen können mit dem Tool `pdlin`, das beispielsweise unter Linux aufgerufen wird mit

```
pdlin [Optionen] Eingabe.pdf Ausgabe.pdf
```

und seinen Optionen entsprechend [Tabelle 6](#) gesetzt werden.

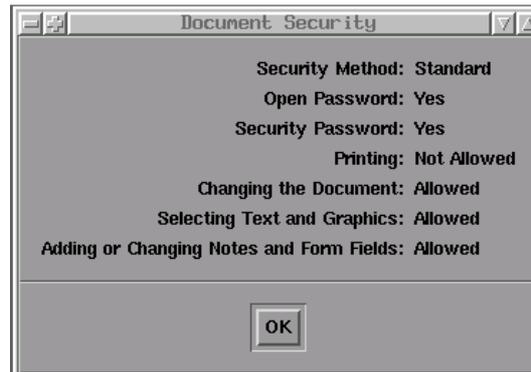


Abbildung 10: Sicherheitsinformation im Acrobat Reader

Option von <code>pdlin</code>	Beschreibung
<code>-user password</code>	Setzen des Kennworts zum Öffnen
<code>-owner password</code>	Setzen des Kennworts für die Sicherheitsoptionen
<code>-perm flags</code>	Flag für die Sicherheitsoptionen ( <code>pcsa</code> ): <p>p: Drucken (printing) nicht zulässig,  c: Änderungen (changing) nicht zulässig,  s: Auswählen (selecting) nicht zulässig,  a: Hinzufügen/Ändern (adding) nicht zulässig</p>
<code>-p password</code>	Kennwort für die einzulesende Datei

Tabelle 6: Optionen des Optimierungs- und Verschlüsselungstools `pdlin`

Beispielsweise erzeugt

```
pdlin -user password1 Eingabe.pdf Ausgabe.pdf
```

aus der Datei `Eingabe.pdf` die mit dem Kennwort `password1` lesegeschützte und optimierte Datei `Ausgabe.pdf`. Mit

```
pdlin -user password1 -owner password2 -perm p Eingabe.pdf Ausgabe.pdf
```

ist die Datei `Ausgabe.pdf` zusätzlich nicht druckbar, siehe [Abbildung 10](#). Die Angabe des Kennworts (`password2`) ist zum Setzen der Sicherheitsoptionen zwingend erforderlich.

<sup>35</sup> beim Acrobat Reader im Menü Datei > Dokumenteigenschaften > Sicherheit

## 6 Erstellung von leistungsfähigen PDF-Dokumenten aus bestehenden L<sup>A</sup>T<sub>E</sub>X-Dateien

Abschließend soll der komplette Vorgang der Erzeugung eines PDF-Dokuments mit Seitenvorschau (Thumbnails) aus einer bestehenden L<sup>A</sup>T<sub>E</sub>X-Datei, in der das `hyperref`-Paket bisher nicht verwendet wurde, beispielhaft dargestellt werden. [Abbildung 11](#) veranschaulicht diesen Erzeugungsvorgang.

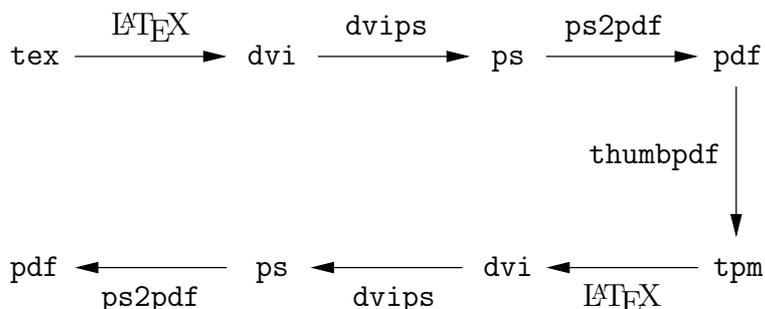


Abbildung 11: Von L<sup>A</sup>T<sub>E</sub>X zu PDF mit dem Paket `thumbpdf`

Zunächst müssen die `thumbpdf`- und `hyperref`-Pakete im Vorspann geladen und dann die gewünschten Optionen gesetzt werden. Dies geschieht, indem z.B. der folgende Text hinter das bisher zuletzt geladene Paket eingefügt wird:

```

\usepackage[ps2pdf]{thumbpdf}
\usepackage[ps2pdf,pagebackref,pdfpagelabels]{hyperref}
\hypersetup{%
  colorlinks=true,          % Einfärbung von Verknüpfungen
  pdfpagemode=UseThumbs,   % Anzeige der Piktogramme
  bookmarksopen=true,      % Anzeige aller Ebenen
  bookmarksnumbered=true,  % Anzeige der Abschnittsnummern
  pdfstartpage={1},        % Startseite
  pdftitle={Titel der Arbeit},
  pdfsubject={Thema der Arbeit},
  pdfauthor={Autor},%
  pdfkeywords={Stichwörter},
  pdfcreator={Adobe-Acrobat-Distiller},%
  pdfproducer={LaTeX with hyperref and thumbpdf}
}
  
```

Der sich anschließende Erzeugungsablauf der L<sup>A</sup>T<sub>E</sub>X-Datei `Beispiel.tex` gliedert sich (wegen der Erstellung der Seitenvorschau) in drei Schritte:

1. Erstellung einer (vorläufigen) PDF-Datei

```

latex Beispiel.tex
bibtex Beispiel
latex Beispiel.tex
latex Beispiel.tex
dvips -Ppdf Beispiel.dvi
ps2pdf Beispiel.ps
  
```

Im ersten L<sup>A</sup>T<sub>E</sub>X-Durchlauf werden wie gewohnt die Dateien `.aux`, `.dvi` sowie `.log` für Inhalts-, Abbildungs- und Tabellenverzeichnis und die Dateien `.toc`, `.lof` sowie `.lot` erstellt. Das `hyperref`-Paket erstellt wie in [Unterabschnitt 3.4.3](#) beschrieben eine `.out` Datei. Mit dem Aufruf von Bib<sub>T</sub>E<sub>X</sub> werden die Dateien `.blg` und `.bib` für das Literaturverzeichnis erstellt. Die nachfolgenden erneuten L<sup>A</sup>T<sub>E</sub>X-Bearbeitungen sind notwendig, um ggfs. die Datei `.brf` für die `backref`-Funktion (siehe [Abschnitt 3.3](#)) zu erzeugen und die Textbezüge (Quer- und Literaturverweise) zu aktualisieren. `dvips` erzeugt mit der Option `-Ppdf` eine PS-Datei, in der die verwendeten Zeichensätze eingebunden sind (siehe [Abschnitt 2.2](#)). Mittels `ps2pdf` wird eine PDF-Datei erstellt, die als Grundlage für die Seitenübersicht dient.

## 2. Erstellung der Bilder für die Seitenübersicht

```
thumbpdf --modes=ps2pdf Beispiel.pdf
```

Das perl-Programm `thumbpdf` erzeugt auf der Grundlage der im ersten Schritt erzeugten PDF-Datei die Bilder für die Seitenübersicht (Piktogramme) und speichert sie in der `.tpm` Datei.

## 3. Einbindung der Seitenübersicht in die (endgültige) PDF-Datei

```
latex Beispiel.tex
dvips -Ppdf Beispiel.dvi
pdftime Beispiel.ps
ps2pdf Beispiel.ps
```

Durch den erneuten L<sup>A</sup>T<sub>E</sub>X-Aufruf werden die Piktogramme in die `.dvi` Datei eingebunden, aus der anschließend mittels `dvips` eine PS-Datei erstellt wird. In dieser Datei kann nun ggfs. mit dem Shell-Skript `pdftime` (siehe [Abschnitt 3.2](#)) der Erstellungs- und Änderungszeitpunkt angepasst werden. Der letzte Bearbeitungsschritt ist die Erzeugung der fertigen PDF-Datei mit `ps2pdf`. Dazu muss auf jeden Fall `Ghostscript` der Version 6.0 oder höher verwendet werden (siehe [Kapitel 4](#)).

Abschließend kann das PDF-Dokument mit dem Tool `pdlin` entsprechend [Kapitel 5](#) optimiert und verschlüsselt werden.

## Literatur

- [1] GOOSSENS, MICHAEL und SEBASTIAN RAHTZ: *Mit L<sup>A</sup>T<sub>E</sub>X ins Web*. Addison-Wesley Verlag, 2000. 4, 6, 7, 13, 14, 15, 16, 17, 18
- [2] JÖCKEL, PATRICK: *How-to make a PDF-document form a L<sup>A</sup>T<sub>E</sub>X-source*. [http://dionysos.mpch-mainz.mpg.de/~joeckel/howto\\_pdf/howto.html](http://dionysos.mpch-mainz.mpg.de/~joeckel/howto_pdf/howto.html). 8
- [3] OBERDIEK, HEIKO: *PDF information and navigation elements with hyperref, pdfT<sub>E</sub>X, and thumbpdf*. EuroT<sub>E</sub>X'99 Proceedings, <http://www.ctan.org/tex-archive/macros/latex/contrib/supported/hyperref/doc/>, 1999. 7, 11, 12, 18, 19
- [4] OBERDIEK, HEIKO: *The hypbmsec package, Bookmarks in sectioning commands*. Datei hypbmsec.dtx, siehe <http://www.ctan.org/tex-archive/macros/latex/contrib/supported/oberdiek/>, 2000. 12
- [5] OBERDIEK, HEIKO: *Project thumbpdf*. Datei readme.txt, siehe <http://www.ctan.org/tex-archive/support/thumbpdf/>, 2001. 18
- [6] RAHTZ, SEBASTIAN: *Hypertext marks in L<sup>A</sup>T<sub>E</sub>X: the hyperref package*. <http://tug.org/applications/hyperref/manual.html>, June 1998. 7, 15, 17
- [7] RAHTZ, SEBASTIAN: *Hypertext marks in L<sup>A</sup>T<sub>E</sub>X*. Datei hyperref.dtx, siehe <http://www.ctan.org/tex-archive/macros/latex/contrib/supported/hyperref/>, August 2001. 12
- [8] REMPIS, PETER: *Richtlinien für das Erstellen von PostScript-Dateien aus T<sub>E</sub>X oder L<sup>A</sup>T<sub>E</sub>X für TOBIAS-lib*. [http://w210.ub.uni-tuebingen.de/dbt/doku/dvi\\_ps.html](http://w210.ub.uni-tuebingen.de/dbt/doku/dvi_ps.html). 6
- [9] STORY, D. P.: *Using L<sup>A</sup>T<sub>E</sub>X to Create Quality PDF Documents for the World Wide Web*. <http://www.math.uakron.edu/~dpstory/latx2pdf.html>, August 1999. 6, 18
- [10] WHITEHOUSE, KENDALL: *Creating quality Adobe PDF files from T<sub>E</sub>X with DVIPS*. z.B. <http://www.utdallas.edu/~cantrell/online/543e.html>. 6