

Die T_EXnische Komödie

DANTE
Deutschsprachige
Anwendervereinigung T_EX e.V.

14. Jahrgang Heft 1/2002 März 2002

1/2002

Impressum

„Die T_EXnische Komödie“ ist die Mitgliedszeitschrift von DANTE e.V. Der Bezugspreis ist im Mitgliedsbeitrag enthalten. Namentlich gekennzeichnete Beiträge geben die Meinung der Schreibenden wieder. Reproduktion oder Nutzung der erschienenen Beiträge durch konventionelle, elektronische oder beliebige andere Verfahren ist nur im nicht-kommerziellen Rahmen gestattet. Verwendungen in größerem Umfang bitte zur Information bei DANTE e.V. melden.

Beiträge sollten in Standard-L^AT_EX-Quellcode unter Verwendung der Dokumentenklasse `dtk` erstellt und an untenstehende Anschrift geschickt werden (entweder per E-Mail oder auf Diskette). Sind spezielle Makros, L^AT_EX-Pakete oder Schriften dafür nötig, so müssen auch diese mitgeliefert werden. Außerdem müssen sie auf Anfrage Interessierten zugänglich gemacht werden.

Diese Ausgabe wurde mit Hilfe folgender Programme erstellt: **TeX**, Version 3.14159 (Web2C 7.3.3.1), **LaTeX2e** (2000/06/01), **xdvi(k)** 22.40b für die Bildschirmdarstellung und **dvips(k)** 5.86e für Korrektur und Belichtung. Die Schriften zur Belichtung wurden mit dem METAFONT-Modus `linoone` (1270 dpi) berechnet.

Erscheinungsweise: vierteljährlich

Erscheinungsort: Heidelberg

Auflage: 2700

Herausgeber: DANTE, Deutschsprachige Anwendervereinigung T_EX e.V.
Postfach 10 18 40
69008 Heidelberg

E-Mail: dante@dante.de
dtk-redaktion@dante.de (Redaktion)

Druck: Konrad Triltsch Print und digitale Medien GmbH
Johannes-Gutenberg-Straße 1–3, 97199 Ochsenfurt-Hohe Stadt

Redaktion: Gerd Neugebauer (verantwortlicher Redakteur)

Luzia Dietsche	Gert Ingold	Volker RW Schaa
Rudolf Herrmann	Rolf Niepraschk	Herbert Voß
Moritz	Günter Partosch	
Hoffmann-Axthelm	Bernd Raichle	

Redaktionsschluss für Heft 2/2002: 1. April 2002

ISSN 1434-5897

Editorial

Liebe Leserinnen und Leser,

in der letzten Ausgabe habe ich zur Mitarbeit bei der Erstellung der Mitgliederzeitschrift „Die T_EXnische Komödie“ aufgerufen. Zu meiner Freude haben sich auch einige Freiwillige gemeldet und schon für diese Ausgabe ihre Arbeit aufgenommen. Ich möchte die „Neuen“ herzlich begrüßen und hoffe, dass sie lange so viel Spaß bei der Arbeit haben wie ich.

Diese Ausgabe ist nur noch sehr knapp im Zeitplan und auch nicht so dick, wie sie sein könnte. Daran ist wieder einmal der Mangel an Beiträgen schuld. Einige interessante Dinge sind in der Vorbereitung, aber es ist unklar, wann wir diese dann auch wirklich abdrucken können. Deshalb hat die Mitgliederversammlung in Erlangen diese Ausgabe überholt. Die Protokolle werden in der Ausgabe 02/2002 abgedruckt. Ich hoffe, dass dadurch nicht zu viel Verwirrung entsteht.

Ich habe eine Vision: Ich würde gerne eine Seite – oder vielleicht auch nur eine halbe Seite – jeder Ausgabe dafür spendieren, die wesentlichen Beiträge der nächsten Ausgabe anzukündigen. Dazu müßte die nächste Ausgabe aber wenigstens so weit vorbereitet sein, dass diese Information schon vorliegt. Leider passiert es aber gelegentlich, dass bei Redaktionsschluss nur ein Beitrag fertig ist. Ich würde mir wünschen, dass möglichst viele Leser mir dabei helfen und mit kleinen oder großen Artikeln dazu beitragen.

In diesem Zusammenhang möchte ich auf die neue Version der Klasse `dtk` hinweisen, die im CTAN zu finden ist. Auch dieses Mal haben wir den Jahreswechsel zum Anlass genommen, einige Kleinigkeiten an dem Layout der Vereinszeitschrift zu verbessern. Vielleicht könnte man es als Suchspiel für die Interessierten ansehen, herauszufinden, worin diese bestehen. An Lob, Kritik, Anregungen und Mithilfe sind wir jederzeit interessiert.

Mit T_EXnischen Grüßen

Ihr Gerd Neugebauer

Hinter der Bühne

Vereinsinternes

Grußwort

Liebe Mitglieder,

wie die Teilnehmer der 26. Mitgliederversammlung von DANTE e.V. in Erlangen bereits wissen, habe ich mich entschlossen, bei der dortigen Wahl zum Vorstand nicht mehr als Vorsitzender zu kandidieren. Meine berufliche Belastung läßt leider ein sinnvolles Engagement für den Verein nicht mehr zu. Schon im letzten Jahr fand ich nicht mehr genug Zeit, um mich in einem Maße einzubringen, wie es dem Titel des Vorsitzenden gerecht würde. Ich habe mich zunehmend darauf verlassen, dass die anfallende Arbeit von meinen Kollegen im Vorstand übernommen wurde. Und ich konnte mich auch darauf verlassen. Dafür danke ich ihnen sehr.

Mein Dank gilt außerdem dem segensreichen Wirken von Frau Dornacher, die in ihrer umsichtigen Art dafür sorgt, dass der Betrieb und die Verwaltung des Vereins vollständig, korrekt und termingerecht erfolgt. Ohne ihren wertvollen Beitrag wäre sicherlich einiges in Unordnung.

Ich blicke gerne auf die Amtszeit als Vorsitzender zurück. Wir haben im Vorstand in der Zeit seit 1998 sehr gut zusammengearbeitet und vieles erreicht, um die Unterstützung der Mitglieder von DANTE e.V. und die Entwicklung von T_EX weiter zu fördern. Der Verein arbeitet Dank des Einsatzes von vielen Aktiven sehr erfolgreich.

An dieser Stelle möchte ich allen danken, die ihre Zeit und ihr Engagement einbringen. Die einzelnen Bereiche der Arbeit wie „Die T_EXnische Komödie“, der CTAN-Server und die CD-ROM-Distributionen, die Tagungsorganisation, die Arbeitskreise und viel Weiteres, was nicht in festen Vereinsstrukturen eingebunden ist, wären ohne den Einsatz dieser Helfer nicht möglich. Ich möchte jeden dazu einladen, sich zu beteiligen und an der Entwicklung und Förderung von T_EX mitzuarbeiten. Ein konkretes Feld liegt uns zur Zeit besonders am Herzen: der WWW-Server des Vereins. Hier werden dringend

Leute gesucht, die *für einen längeren Zeitraum kontinuierlich* an der Pflege des Web-Auftritts von DANTE e.V. mitarbeiten, da die Arbeit umfangreich und die Zeit der Beteiligten begrenzt ist. Interessenten können sich an den Schriftführer Günter Partosch wenden.

Für ein Engagement in den einzelnen Bereichen ist immer Platz und es bieten sich vielfältige Betätigungsmöglichkeiten – machen Sie mit, besuchen Sie die Tagungen und nehmen Sie an den Gesprächen und Diskussionen teil! Ich kann nur sagen, es lohnt sich. Dasselbe gilt auch für die internationale Zusammenarbeit der T_EX-Gemeinschaft. Mit den Entwicklungen der letzten Zeit, wie ϵ -T_EX, pdfT_EX, $\mathcal{N}\mathcal{T}\mathcal{S}$ und den Entwicklungen rund um Omega, L^AT_EX3 sowie ConT_EXt zeigt sich, dass in T_EX genug Leben ist, um Weiterentwicklungen zu inspirieren, auch wenn sich die Projekte durch die meist ehrenamtliche Natur der Arbeit schon mal etwas hinziehen.

Selbst wenn man in letzter Zeit zunehmend den Eindruck hat, dass die Verwendung von T_EX zurückgeht und selbst das Bewusstsein schwindet, dass es Textsatz jenseits von WinWord gibt, bin ich voller Zuversicht, dass die Qualität und die Stärken von T_EX *and friends* ihren Platz in der Landschaft der Rechner und des Textsatzes behaupten werden.

Bei den Neuwahlen des Vorstands in Erlangen hat auch der Schatzmeister Horst Szillat sein Amt niedergelegt, wie er es schon auf der 25. Mitgliederversammlung im Herbst 2001 in Kerkrade angekündigt hatte. Dass DANTE e.V. eine stabile Finanzverwaltung vorweisen kann, ist zu großen Teilen sein Verdienst. Er nahm sich für die Umstellung der Buchhaltung und der Mitgliederverwaltung viel Zeit und betreute zudem mit großem Einsatz die Infrastruktur im Büro von DANTE e.V. in Heidelberg. Er hat sich dankenswerterweise bereit erklärt, diesen organisatorischen Teil seiner Tätigkeit auch weiterhin auszuüben. Ich möchte ihm meinen Dank und den Dank des Vorstandes für seine geleistete und zukünftige Arbeit aussprechen.

Seinem Nachfolger im Amt des Schatzmeisters, Tobias Sterzl, gilt unser Dank für seine Bereitschaft, diese Aufgabe zu übernehmen. Er ist seit 1994 Mitglied bei DANTE e.V. und plant Mitte nächsten Jahres seine Promotion am Forschungsinstitut „caesar“ in Maschinenbau abzuschließen.

Als neuen Beisitzer im Vorstand darf ich Bernd Raichle herzlich willkommen heißen. Eine Vorstellung erübrigt sich im Kreis der, nicht nur deutschen, T_EX-Gemeinschaft, da er allen durch die Weiterentwicklung von *german* und ϵ -T_EX ebenso bekannt sein dürfte wie durch die T_EX-FAQ und seine ständige

Hilfsbereitschaft, bei Problemen mit und um $\text{\TeX}$ mit Rat und Tat zur Seite zu stehen.

Erhalten bleibt uns als Schriftführer Günter Partosch, der diese Aufgabe bereits seit vielen Jahren erfüllt und in dessen Händen die aufwändige und zeitraubende Kommunikation des Vereins liegt. Ebenso, mit neuer Aufgabe (und zusätzlicher Verantwortung), bleibt uns Klaus Höppner erhalten, der als stellvertretender Vorsitzender dem neuen Vorstand weiterhin angehören wird.

Meinem Nachfolger im Amt des Vorsitzenden, Volker RW Schaa, möchte ich für die Fortführung seines großen Einsatzes in neuer Position rund um DANTE e.V. und $\text{\TeX}$ alles Gute wünschen und kann nur hoffen, dass er die gleiche Unterstützung findet, wie er sie immer großzügig zu geben bereit war und ist. Ich werde versuchen, meinen Teil nun als Beisitzer im Vorstand beizutragen.

Mit freundlichen Grüßen,

Thomas Koch
(Ex-Präsident; Beisitzer)

Grußwort, 2. Teil

Liebe Mitglieder,

Sie fragen sich nun sicherlich, was das für eine neue Mode ist, in einer Ausgabe von „Die $\text{\TeX}$ nische Komödie“ zwei Grußworte unterzubringen? Unser Chefredakteur hätte auch lieber frühzeitig ein Grußwort als derer zwei – und zudem noch so lange. Aber nachdem wir uns 1998 in Eichstätt als Team zur Wahl gestellt hatten und als eine kleine Neuerung das Grußwort gemeinsam verfassten (und unser Chefredakteur damit zwei Leuten nachlaufen musste), wollten wir diese gemeinsame Tradition auch für den letzten Auftritt nutzen. Wenn Sie nun mit dem Lesen bis hierher gekommen sind, werden Sie verstehen, dass Thomas ein persönliches Grußwort geschrieben hat, bei dem ich schlechterdings nicht meinen Namen hinzumogeln kann. Zudem hat er mir keine Möglichkeit gelassen, den alten und neuen Mitstreitern zu danken; es wäre nur eine müde Wiederholung.

Deshalb möchte ich einen kurzen Blick zurückwerfen auf die Tagung in Erlangen, wo wir Gäste des regionalen Rechenzentrums waren. Das Protokoll und die Handouts der Mitgliederversammlung und einen Tagungsbericht werden Sie in der Ausgabe 2/2002 von „Die T_EXnische Komödie“ finden. Jetzt möchte ich möglichst zeitnah den lokalen Organisatoren und hier stellvertretend Hans Cramer und Sigmar Lingner danken für die hervorragende Planung und Durchführung der Tagung. Es hat uns viel Spaß gemacht, da uns neben einem weitgefächerten interessanten T_EXnischen Programm auch noch eine exzellente Versorgung geboten wurden. Sowohl das Spezialarrangement mit der Mensa wie auch die ununterbrochene Bereitstellung von Getränken und Keksen zwischen 8.30 Uhr und 18.30 Uhr waren überzeugend! Damit die T_EXnische Diskussion beim geselligen Beisammensein nicht zu kurz kam, hatte Hans Cramer mit seinen Mannen für die vier Abende auch noch nette Lokale in der Altstadt von Erlangen mit fränkischer Küche und lokalen Spezialitäten ausgewählt, in denen die Abende ausklangen. Nochmals vielen Dank!

Bevor ich nun zum Ende komme, möchte ich Sie alle einladen, die oben geschilderte Erfahrung selbst zu machen. Hier haben Sie nun die Auswahl: Sie können selbst eine Tagung ausrichten. Wir suchen dringend Veranstalter für die nächsten Termine – das Organisationsteam von DANTE e.V. hilft Ihnen tatkräftig bei Ausrichtung, Programmgestaltung und Planung. Das wäre aus unserer Sicht die beste Wahl.

Oder Sie besuchen eine der verschiedenen T_EX-Tagungen. Hierfür möchte ich Ihnen die EuroBachoT_EX-Tagung besonders ans Herz legen. In der Zeit vom 29. April bis zum 3. Mai 2002 findet in Bachotek/Polen die diesjährige europäische T_EX-Tagung statt. Der Veranstaltungsort liegt an einem schönen See, man wohnt zu zweit oder zu dritt in gemütlichen „Holzbungalows“ in einem Kiefernwald, die Umgebung lädt zum Urlaubmachen ein. Die niederländische T_EX-Gruppe NTG sponsert einen Bus, der von Utrecht über Osnabrück, Hannover und Berlin nach Polen fährt und für alle Teilnehmer kostenlos ist. Das Programm, das Sie auf unseren Veranstaltungsseiten finden, liest sich sehr vielversprechend und deckt wieder das gesamte Spektrum der T_EXnischen Entwicklung ab. Sie sollten es sich gönnen!

Bis zum nächsten Grußwort in anderer Besetzung verbleibe ich nun mit freundlichen Grüßen, Ihr

Volker RW Schaa
(Ex-Vizepräsident; Präsident)

Bretter, die die Welt bedeuten

Erstellung von leistungsfähigen PDF-Dokumenten

Sascha Beuermann

In diesem Artikel werden die Möglichkeiten aufgezeigt, die die Pakete `hyperref` und `thumbpdf` bieten, um mit Hilfe von $\text{\LaTeX}$ leistungsfähige Dokumente im Portable Document Format (PDF) zu erstellen. Dabei wird insbesondere auf das Erzeugen von Lesezeichen (Bookmarks), auf PDF-spezifische Anzeigeeoptionen sowie auf das Anlegen von Hypertext-Strukturen mit Hilfe des Pakets `hyperref` und auf die Generierung der Seitenvorschau (Piktogramme oder Thumbnails) mit dem Paket `thumbpdf` eingegangen. Abschließend wird die Linearisierung und Verschlüsselung von PDF-Dokumenten behandelt.

Einleitung

Bei der Veröffentlichung von Dokumenten im Internet findet das Portable Document Format (PDF) besonders für öffentliche Dokumente zunehmend Verbreitung, deren Layout vom Browser des Lesers nicht mehr verändert werden soll. Über die einfache Lesefunktion hinaus bietet PDF die Möglichkeit, Hypertext-Strukturen anzulegen und Lesehilfen (Lesezeichen und Seitenübersichtsbilder) zu generieren.

In [1] wurden bereits unterschiedliche Möglichkeiten aufgezeigt, aus $\text{\LaTeX}$ -Quellen PDF-Dokumente zu erstellen. Im Unterschied zu [1] widmet sich dieser Artikel der Erzeugung von PDF-Dokumenten mittels $\text{\LaTeX}$, `dvips` und `ps2pdf` (bzw. Distiller). Auf diese Weise ist es ohne große Umstände möglich, auch aus bereits bestehenden $\text{\LaTeX}$ -Quellen leistungsfähige PDF-Dokumente zu erzeugen.

Bereits bei der Erstellung der PostScript-Datei sollten die verwendeten Zeichensätze eingebunden werden, damit diese dann vom PDF-Viewer nicht als

Bitmap mit einer schlechten Qualität wiedergegeben werden. Dies geschieht beispielsweise mit `dvips` und der Option `-Ppdf`, siehe sonst aber auch [8]. Bei der Konvertierung der PostScript-Datei in eine PDF-Datei mit `ps2pdf` ist darauf zu achten, dass `ghostscript` in der Version 6.0 oder höher verwendet wird, da frühere Versionen die Einbindung der Zeichensätze in die PDF-Datei nicht unterstützen. Falls keine entsprechende Version vorhanden ist, kann auf Online-Distiller (beispielsweise von Rechenzentren wie dem RRZN unter <http://www.rrzn.uni-hannover.de/pdf/>) zurückgegriffen werden.

Das Paket `hyperref`

Das Paket `hyperref` von Sebastian Rahtz und Heiko Oberdiek [2] erweitert (automatisch) die Funktionen aller $\text{\LaTeX}$ -Querverweisbefehle (einschließlich Inhaltsverzeichnis, Literaturverzeichnis usw.) und erzeugt `\special`-Befehle, die ein Treiber wie `dvips` mit der Option `-z` oder `-Ppdf` in Hypertext-Links umwandeln kann. Außerdem verfügt das Paket über neue Befehle, die es dem Benutzer ermöglichen, zusätzliche Hypertext-Links zu produzieren, einschließlich Verknüpfungen auf externe Dokumente und beliebige Dokumente im Internet.

Da die grundlegenden Angaben zu `hyperref` im Vorspann der $\text{\LaTeX}$ -Datei gemacht werden, ist es ohne weiteres möglich, aus bestehenden $\text{\LaTeX}$ -Quellen leistungsfähige PDF-Dokumente zu erstellen.

Die Beschreibung des Paketes `hyperref` [6] ist aus dem Jahre 1998 und damit etwas veraltet. Eine aktuelle Beschreibung von `hyperref` mit allen seinen Funktionen findet sich beispielsweise in [7] Kapitel 2.3 oder [3] Kapitel 5.

Die wichtigsten Funktionen von `hyperref` werden in den folgenden Abschnitten behandelt. Auf die Möglichkeit, mit `hyperref` PDF-Formulare zu erzeugen, soll hier jedoch nicht eingegangen werden. Insofern wird auf [6] Kapitel 6 und [7] Kapitel 2.3.6 verwiesen.

Allgemeine Konfiguration von `hyperref`

Das Paket muss im Vorspann der $\text{\LaTeX}$ -Datei angefordert werden. Da es viele $\text{\LaTeX}$ -Befehle neu definiert, sollte es als letztes Paket geladen werden. Die Paketoptionen 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 ausgewählt werden:

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

Mit der Treiber-Option **draft** werden alle Hypertext-Optionen ausgeschaltet, so dass auch Pakete, die den Inhalt der Makros `\label` und `\ref` setzen – wie **showkeys** –, wieder funktionieren.

Dokumentzusammenfassung

Mit Hilfe des Pakets **hyperref** können folgende Einträge in der Dokumentzusammenfassung, die im Acrobat-Reader über das Menü **Datei – Dokumentinfo – Allgemein** aufgerufen werden kann, erstellt werden: Titel (`pdftitle`), Thema (`pdfsubject`), Verfasser (`pdfauthor`), Stichwörter (`pdfkeywords`), erstellt mit (`pdfcreator`) und erzeugt mit (`pdfproducer`).

Leider gibt es keine Möglichkeit, Erstellungs- und Modifikationszeitpunkt mit dem Paket **hyperref** einzustellen. Deshalb müssen 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.

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, die Hypertext-Optionen vorübergehend auszuschalten, was wohl aber nur in Ausnahmefällen notwendig ist.

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 eventuell selbst Hand angelegt werden muss, um Links zu erstellen, die auf dasselbe Ziel verweisen. Für das Inhaltsverzeichnis

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

Tabelle 1: Erweiterungsoptionen für `colorlinks`

kann auf die Option `linktocpage` ausgewichen werden, die dafür sorgt, dass nicht der Text, sondern die Seitennummer eines Eintrags im Inhaltsverzeichnis als Hyperlink aktiviert wird.

Mit der Option `colorlinks` wird der Text von Verknüpfungen und Ankern eingefärbt. Die Wahl der Farbe hängt dabei von der Verknüpfungsart ab, kann jedoch mit den in Tabelle 1 dargestellten Erweiterungsoptionen verstellt werden, wobei auch in L^AT_EX definierte Farben verwendet werden können. Das zur farbigen Darstellung der Links erforderliche Paket `color` wird bei der Verwendung von `hyperref` automatisch dazugeladen.

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

Ferner gibt es im Paket `hyperref` die Option, im Literaturverzeichnis die einzelnen Einträge mit „rückwärtigen“ Verknüpfungen zu versehen. Die Option `backref` fügt am Ende eines Eintrags eine Rückverknüpfung als Liste von Abschnittsnummern hinzu, die Option `pagebackref` als eine Liste von Seitennummern.

Der Eintrag dieser beiden Optionen sollte im optionalen Argument des Befehls `\usepackage` und nicht im Befehl `\hypersetup` stehen. 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 B_IB_TE_X der Fall ist.

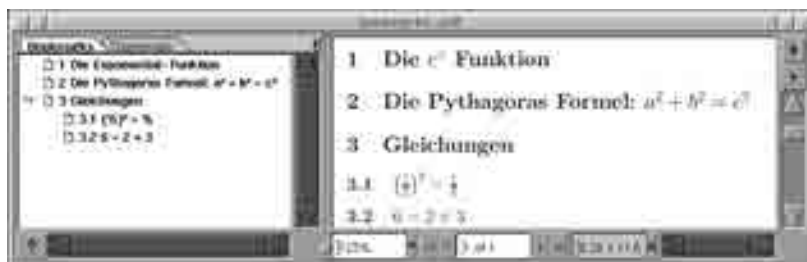


Abbildung 1: Lesezeichen im Acrobat-Reader

Acrobat-Lesezeichen (Bookmarks)

Zugehörige Paketoptionen

Lesezeichen (oder Bookmarks) sind eine Navigationshilfe im Acrobat-Reader (siehe Abbildung 1). Mit der Paketoption `bookmarks` werden Acrobat-Lesezeichen in ähnlicher Weise wie das Inhaltsverzeichnis erstellt. Das Paket `hyperref` schreibt automatisch den Lesezeichen-Code für die Gliederungsbefehle `\section`, `\subsection` usw. in eine `.out`-Datei. Mit der (alternativen) Option `bookmarksopen` werden bei der Anzeige der Lesezeichen alle Ebenen angezeigt. Die Tiefe, bis zu der die Lesezeichen geöffnet werden, kann jedoch auch mit `bookmarksopenlevel={number}` vorgegeben werden. Durch die (zusätzliche) Option `bookmarksnumbered` werden die Lesezeichen zusammen mit den Abschnittsnummern angezeigt. Abbildung 1 wurde beispielsweise mit den Paketoptionen `bookmarksopen` und `bookmarksnumbered` erstellt.

Erstellung individueller 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}{}` eine *Marke* erstellt werden, auf die sich dann das mit

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

erstellte Lesezeichen bezieht. Mit dem optionalen 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 beispielsweise in der Dokumentenklasse `article` mit

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

das Inhaltsverzeichnis zu den Lesezeichen hinzugefügt werden. Die Marke `toc` wird – wie auch die Marken `lof` und `lot` für das Abbildungs- bzw. Tabellenverzeichnis – automatisch gesetzt.

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 beispielsweise 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{$e^x$}{Exponential-}Funktion}
```

ist in Abbildung 1 gezeigt.

PD1-Kodierung

Mit dem Paket `pd1enc`, 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{$a^2+b^2=c^2$}}%
```

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>
$\frac{1}{2}$	<code>\textonehalf</code>
$\frac{1}{4}$	<code>\textonequarter</code>

Tabelle 2: PD1-Kodierung von Sonderzeichen

```
{a\texttwosuperior\ + b\texttwosuperior\ %
= c\texttwosuperior\ }}
```

das in Abbildung 1 dargestellte Lesezeichen. Kodierungen für andere gebräuchliche Sonderzeichen sind in Tabelle 2 zusammengestellt. Genauere Informationen zu der Zeichenkodierung in PDF sind beispielsweise in [3] Kapitel 2 und 3 zu finden.

Ersetzen von $\LaTeX$ -Code mit dem Paket `hybmsec`

Eine andere (und bequemere) Möglichkeit $\LaTeX$ -Code zu ersetzen bietet die Erweiterung der Gliederungsbefehle durch das Paket `hybmsec`[4]: Ein zweites (optionales) Argument in eckigen Klammern wird nicht nur für den Eintrag im Inhaltsverzeichnis und gegebenenfalls 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` [3]:

```

\section{Inhaltsverzeichnis/Seitenkopf = Lesezeichen}{Text}
\section{Lesezeichen}[Inhaltsverzeichnis/Seitenkopf]{Text}
\section{Lesezeichen}{Inhaltsverzeichnis/Seitenkopf = Text}
\section{Inhaltsverzeichnis/Seitenkopf = Lesezeichen = Text}
\section{Inhaltsverzeichnis/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 Paket `hyperref` 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 1 gezeigt.

PDF-spezifische Anzeigooptionen

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 [7] Kapitel 2.3.8 dargestellt.

Zusätzliche Benutzermakros für Hyperlinks

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

Verweise innerhalb des Dokuments

Mit dem Makro

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

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.	/I (Invertierung)* /N (kein Effekt) /O (Umrandung) /P („gedrückter Kopf“)
<code>pdfmenubar</code>	Legt fest, ob die Menüleiste 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 Abbildung 2. 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 [7] 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 (*: Voreinstellung) [7]

wird der *Text* als Link zu einem Punkt aktiviert, der mit dem normalen L^AT_EX-Befehl `\label` mit dem symbolischen Namen *Marke* erstellt wurde. Da dieser Befehl nur nach einem Gliederungsbefehl oder in den Umgebungen `equation`, `eqnarray`, `figure`, `table` oder `enumerate` 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 Befehl `\ref` und fügt vor jedem Verweis eine verknüpfungsabhängige Kennung ein. Der Unterschied wird bei den folgenden Verweisen auf Tabelle 4 deutlich: `\ref{Marke}` erzeugt als Verknüpfung „4“, während `\autoref{Marke}` „Tabelle 4“ erzeugt. Im ersten Fall besteht der Verweis nur aus der Zahl, im zweiten Fall ist die Kennung der Marke (hier: Tabelle) Teil der Verknüpfung. Die zweite Variante ist wegen der größeren Verknüpfungsfläche benutzerfreundlicher als die erste. Die Kennung wird von `hyperref` mit Hilfe der in Tabelle 4 aufgeführten Makros aus dem Kontext des ursprünglichen Befehls `\label` 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{Makroname}{Bezeichnung}
```

Sollen eigene Verknüpfungstexte – und nicht nur mit `\ref` oder `\pageref` erzeugte 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]{Tab.~\ref*{Marke}, S.~\pageref*{Marke}}
```

erzeugt „Tab. 4, S. 18“, wobei `\ref*{Marke}` und `\pageref*{Marke}` die richtige Nummer, jedoch keine Verknüpfung generieren. Diese wird mit dem Befehl `\hyperref` erzeugt.

<i>Makroname</i>	<i>Bezeichnung</i>
<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 Befehl `\autoref` von `hyperref` [7]

Mit Hilfe des Pakets `nameref`, 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 Umgebung `NoHyper` verwendet werden:

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

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“ 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“ und

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

„mailto:beuermann@ibnm.uni-hannover.de“.

Verweise auf andere (PDF-)Dokumente

Das Ergänzungspaket `xr-hyper` ermöglicht es zusammen mit `hyperref`, auf Markierungen, die in externen $\text{\LaTeX}$ -Dateien mit `\label` erstellt wurden, zu verweisen. In diesen Dateien muss jedoch ebenfalls das Paket `hyperref` geladen werden.

Um auf andere PDF-Dokumente zu verweisen, muss die `hyperref`-Option `extension=pdf` 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 beispielsweise wie folgt aus,

```
\usepackage{xr-hyper}
\usepackage[ps2pdf,extension=pdf]{hyperref}
\externaldocument[A-]{name1}
\externaldocument[B-]{name2}
```

kann auf die Markierung *Marke* im Dokument `name1.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 L^AT_EX-Dateien mit dem Befehl `\hypertarget` erstellt wurden, ist auch ohne Verwendung des Pakets `xr-hyper` mit Hilfe von `\href` möglich. Auf die Markierung *Marke* im Dokument `name1.pdf` kann mit

```
\href{name1.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{name2.pdf#subsection.2.1}{Text}
```

wird auf Abschnitt 2.1 in der Datei `name2.pdf` verwiesen. Auskünfte über die Bezeichnungen der Gliederungspunkte wie *subsection.2.1* gibt die jeweilige `.out`-Datei des externen Dokuments.

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. Eine Liste aller zur Verfügung stehenden Menüoptionen ist in [6] Kapitel 4 und [7] Kapitel 2.3.4 zu finden.

Beispielsweise würde der Befehl

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

folgende Schaltfläche erzeugen:

Dokumentzusammenfassung

Das Verhalten der Verknüpfungsschaltflächen beim Anwählen kann mit der Option `pdfhighlight` festgelegt werden, siehe Tabelle 3.

Werden Schaltflächen aus dem Acrobat-Menü *Bearbeiten* auf jeder Dokumentseite erzeugt, lässt sich beispielsweise für *Bildschirm-Präsentationen* eine komfortable Navigationsleiste im Dokument erstellen.

<i>Acrobat-Menü</i>	<i>Option</i>	<i>Menüoption</i>
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 [7]

Für eine Bildschirmpräsentation sollte das PDF-Dokument im Querformat erstellt werden, was durch Einfügung folgender `\special`-Befehle und entsprechende Seitenformatierungsanweisungen in den Vorspann der $\text{\LaTeX}$ -Datei möglich ist:

```
\special{landscape}
\special{! TeXDict begin /landplus90{true}store end }
```

Das Paket thumbpdf

Das Paket `thumbpdf` von Heiko Oberdiek [5] erzeugt mit Hilfe des Perl-Programms `thumbpdf`, das wiederum `ghostscript` verwendet, Thumbnails

(Piktogramme) für L^AT_EX-Dateien, die zu PDF-Dateien konvertiert werden [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, wie der Acrobat-Reader.

Zur Generierung der Bilder ist **ghostscript** mindestens in der Version 5.50 erforderlich, zum Einbinden der Bilder mit **ps2pdf** mindestens die Version 6.0. Für die Benutzung mit **ps2pdf** muss das Paket **thumbpdf** in die L^AT_EX-Datei wie folgt eingebunden werden:

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

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

Zunächst ist aus der L^AT_EX-Datei (**name.tex**) wie gewöhnlich eine (jedoch nur vorläufige) PDF-Datei (**name.pdf**) zu erstellen. Anschließend muss das **perl**-Programm **thumbnail** mit

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

aufgerufen werden, das die Thumbnails erstellt und in der Datei **name.tpt** speichert. Abschließend muss L^AT_EX erneut aufgerufen und mittels **dvips -Ppdf** und **ps2pdf** eine neue (die endgültige) PDF-Datei erstellt werden. Abbildung 2 zeigt ein (fertiges) PDF-Dokument mit Piktogrammen, das mit der Option **pdfpagelabels** erstellt wurde (vgl. Tabelle 3).

Es ist auch möglich, anstelle der verkleinerten Seitenbilder andere Bilder als Piktogramme zu verwenden, siehe dazu [3] Kapitel 8.4.

Linearisierung und Verschlüsselung von PDF-Dokumenten

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

Das Linearisieren – oft auch Optimieren genannt – einer PDF-Datei bedeutet jedoch nicht, dass die Dateigröße kleiner wird. Im Gegenteil, sie wird durch die Linearisierung eher größer.



Abbildung 2: Piktogramme im Acrobat-Reader



Abbildung 3: Sicherheitsinformation im Acrobat-Reader

Beispielsweise können mit Hilfe des Tools `pdlin` der Glance AG linearisierte Dateien aus einer bestehenden PDF-Datei erstellt werden. Es ist außerdem möglich, verschlüsselte Dateien als Eingabe-Datei zu lesen beziehungsweise die linearisierte Ausgabe-Datei entsprechend der Standard-PDF-Verschlüsselung zu verschlüsseln. Unter <http://pdf.glance.ch/eval/> steht das Tool `pdlin` in der Version 2.02 (für Windows, Linux und Solaris) zum freien Download zur Verfügung. Die freie Verwendung beschränkt sich auf Evaluationszwecke, für den produktiven Betrieb ist das Tool zu lizenzieren. Bei größeren PDF-Dateien treten jedoch Probleme beim Optimieren mit `pdlin` auf (Lesezeichen und Links funktionieren nicht mehr).

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 (pcsa): p: Drucken (printing) nicht zulässig, c: Änderungen (changing) nicht zulässig, s: Auswählen (selecting) nicht zulässig, a: Hinzufügen/Ändern (adding) nicht zulässig
<code>-p password</code>	Kennwort für die einzulesende Datei

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

In der *Sicherheitsinformation*, die im Acrobat-Reader über das Menü *Datei – Dokumentinfo – Sicherheit* aufgerufen werden kann, 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 3. Diese Einstellungen können mit dem Tool `pdlin`, das beispielsweise unter Linux mit

```
pdlin [Optionen] eingabe.pdf ausgabe.pdf
```

aufgerufen wird, und seinen Optionen entsprechend Tabelle 6 gesetzt werden.

Beispielsweise erzeugt

```
pdlin -user pw1 eingabe.pdf ausgabe.pdf
```

aus der Datei `eingabe.pdf` die mit dem Kennwort `pw1` lesegeschützte und linearisierte Datei `ausgabe.pdf`. Mit

```
pdlin -user pw1 -owner pw2 -perm p eingabe.pdf ausgabe.pdf
```

ist die Datei `ausgabe.pdf` zusätzlich nicht druckbar, siehe Abbildung 3. Die Angabe des Kennworts (`pw2`) ist zum Setzen der Sicherheitsoptionen zwingend erforderlich.

Einen hundertprozentigen Schutz bieten die Sicherheitsoptionen nicht; eine erneute Anwendung von `pdlin` auf die PDF-Datei `ausgabe.pdf` mittels

```
pdlin -p pw1 ausgabe.pdf neu.pdf
```

hat nämlich zur Folge, dass alle Sicherheitsoptionen in der Datei `neu.pdf` auf *Zulässig* gesetzt werden.

Zusammenfassung

Der vorliegende Artikel will aufzeigen, dass es mit $\text{\LaTeX}$ und den Paketen `hyperref` und `thumpdf` ohne viel Aufwand möglich ist, die zusätzlichen Möglichkeiten zu nutzen, die PDF bietet. Das Paket `hyperref` erlaubt unter anderem das Anlegen von Hypertext-Strukturen und Lesezeichen. Mit Hilfe des Pakets `thumpdf` können die Seitenübersichtsbilder erstellt und eingebunden werden.

Die Verwendung von `pdfTeX` (siehe auch [1]), für das die vorliegende Beschreibung des Pakets `hyperref` natürlich auch gilt, ist also nicht unbedingt notwendig, um leistungsfähige PDF-Dokumente zu erstellen. Aus diesem Grunde stellt es kein Problem dar, aus bestehenden $\text{\LaTeX}$ -Quellen leistungsfähige PDF-Dokumente zu erstellen, nur eine kurze Ergänzung im Vorspann ist notwendig. Der Nachteil der etwas umständlichen Erzeugung des PDF-Dokuments über eine DVI- und PostScript-Datei mit $\text{\LaTeX}$ im Vergleich zur direkten Erzeugung mit `pdfTeX` kann durch die Verwendung eines kleinen Skriptes ausgeglichen werden.

Literatur

- [1] Thomas Feuerstack: *Einführung in pdfTeX*; *Die T_EXnische Komödie*; 2/2001, S. 41–54; 2001.
- [2] Sebastian Rahtz und Heiko Oberdiek: *hyperref*; Version 6.71e; 2001; CTAN: `tex-archive/macros/latex/contrib/supported/hyperref/`.
- [3] Heiko Oberdiek: *PDF information and navigation elements with hyperref, pdfTeX, and thumpdf*; EuroT_EX'99 Proceedings; 1999; CTAN: `tex-archive/macros/latex/contrib/supported/hyperref/doc/`.
- [4] Heiko Oberdiek: *The hypbmsec package*; Version 2.1; 2000; CTAN: `tex-archive/macros/latex/contrib/supported/oberdiek/`.
- [5] Heiko Oberdiek: *Project thumpdf*; Version 2.10; 2001; CTAN: `tex-archive/support/thumpdf/`.

- [6] Sebastian Rahtz: *Hypertext marks in L^AT_EX: the hyperref package*; Juni 1998; CTAN: tex-archive/macros/latex/contrib/supported/hyperref/doc/manual.pdf.
- [7] Michael Goossens und Sebastian Rahtz: *Mit L^AT_EX ins Web*; Addison-Wesley Publishing Company; Bonn; 2000.
- [8] Kendall Whitehouse: *Creating quality Adobe PDF files from T_EX with DVIPS*; z. B. <http://gemini.iti.informatik.tu-darmstadt.de/~kehr/doc/TeX2PDF/adobe-guide.html>.

Ansprechende technische Illustration mit METAPOST

Uwe Siart

METAPOST ist eine Grafik-Beschreibungssprache zur Erstellung qualitativ hochstehender Vektorzeichnungen. Obwohl der Compiler in praktisch allen modernen T_EX-Distributionen enthalten ist, findet METAPOST noch relativ wenig Beachtung. METAPOST ist besonders geeignet zur Erstellung von Strichzeichnungen, wie sie in Naturwissenschaft und Technik häufig auftreten. Dabei erzeugt es besonders schlanke Bilddateien, deren Format praktisch identisch zu EPS ist. Allerdings werden Ressourcen, die das T_EX-Dokument ohnehin einbindet, einfach weggelassen. Die angeführten Beispiele stammen – wie auch der Autor – aus dem ingenieurwissenschaftlich-technischen Bereich. Sie sollen einige Fähigkeiten von METAPOST aufzeigen und den Leser ermutigen, seine eigenen Anwendungsgebiete für METAPOST zu suchen.

Einführung

Das Erstellen von naturwissenschaftlichen oder technischen Strichzeichnungen ist durch Eingabe mit einem Zeigegerät zwar möglich, erfordert aber meistens sehr viel Geduld. Trotz beachtlichen Zeitaufwands ist das grafische Ergebnis nicht selten von nur mäßiger Qualität. Es treten Positionierungsaufgaben auf, die mit Mauseingabe nur durch Augenmaß gelöst werden können. Änderungen oder Zuweisungen, die auf diese Weise gemacht wurden, sind meist nicht reproduzierbar. Und benötigt man eine Serie, deren Bilder sich

eigentlich nur in einem Parameter unterscheiden, so bedeutet dies nicht selten die mehrfache Zeichenarbeit. Schließlich sind die erzeugten EPS-Dateien häufig durch ungenutzte PostScript-Ressourcen aufgebläht, was bei der PostScript-Datei eines Buches mit mehreren hundert Abbildungen durchaus ins Gewicht fällt. Den Benutzern von pdf(LA)TeX wird angenehm auffallen, dass sie die Bilddateien von METAPOST *nicht* in PDF umwandeln müssen. Sie können von pdf(LA)TeX direkt konvertiert werden, sodass hier ein Arbeitsschritt entfallen kann.

Die Arbeitsweise von METAPOST ist identisch mit derjenigen von TeX. Das Programm liest eine Eingabedatei mit Anweisungen, die das Bild beschreiben, und legt nach erfolgreichem Lauf eine oder mehrere Bilddateien im Arbeitsverzeichnis ab. Der Name, unter dem METAPOST aufgerufen wird, hängt von der jeweiligen TeX-Distribution ab. In MiKTeX lautet er beispielsweise

```
mp dateiname.mp
```

wobei die Endung `.mp` beim Aufruf weggelassen werden kann. Neben `mp` lautet das Kommando auch häufig `mpost`. In den meisten Fällen dürfte eine dieser beiden Varianten zutreffen. Innerhalb der Eingabedatei befinden sich die Befehle für ein Bild zwischen den Anweisungen

```
beginfig(1);  
...  
endfig;
```

Dieses Anweisungspaar zusammen mit einem Zeichenbefehl ist daher auch der Mindestinhalt einer METAPOST-Datei, um ein Bild zu erzeugen. Die Eingabedatei darf aber auch die Anweisungen für mehr als ein Bild enthalten. Die umschließenden Anweisungen `beginfig(n)` und `endfig` dürfen deshalb auch mehrfach auftreten, eben einmal je Bild. Nach erfolgreichem Durchlauf hat METAPOST Dateien mit den Bezeichnungen `dateiname.1`, `dateiname.2`, usw. erzeugt, wobei die Nummer im Argument von `beginfig` die Endung der zugehörigen Ausgabedatei bestimmt. Dieses sind die Bilddateien, die direkt in L^ATeX-Dokumente eingebunden werden können. Beim Erzeugen von Beschriftungen ist in METAPOST jedes beliebige TeX-Konstrukt zulässig. Wenn METAPOST zum Setzen von Beschriftungen anstatt TeX lieber L^ATeX aufrufen soll, so kann dieses entweder mit dem Aufruf

```
mp --tex=latex dateiname
```

oder durch Setzen der Umgebungsvariablen `TEX` auf den Wert `latex` erreicht werden. An den Beginn der METAPOST-Eingabedatei ist dann ein gewöhnlicher $\text{\LaTeX}$ -Vorspann in der Form

```
verbatimtex
\documentclass{...}
  eventuell weitere Pakete
\begin{document}
etex
```

zu setzen. Das sonst notwendige `\end{document}` entfällt, weil dieses von METAPOST automatisch erzeugt wird. Abhängig von der jeweiligen $\text{\TeX}$ -Distribution kann die Anweisung zum Aufruf von $\text{\LaTeX}$ auch in den Vorspann der METAPOST-Datei geschrieben werden. Hierzu sollte sofort nach `verbatimtex` das Kommando `%\latex` eingefügt werden. Welche dieser Methoden funktioniert und welche nicht, kann man – falls nicht in der Dokumentation des eigenen $\text{\TeX}$ -Paketes angegeben – durch Probieren herausfinden.

Im Vorspann müssen wie in einem $\text{\LaTeX}$ -Dokument alle Pakete geladen werden, auf deren Funktionen innerhalb der METAPOST-Quelldatei zurückgegriffen wird. Um Konflikte beim Laden von Schriften oder unpassende Schriftgrößen zu vermeiden, ist es sinnvoll, wenn die METAPOST-Datei denselben Vorspann aufweist, wie das Dokument, in das die Bilder eingefügt werden.

Ausgewählte Grundfunktionen

Punkte, Geraden, Polygone, Kurven

METAPOST kennt mehrere Variablentypen. Die für diese Kurzeinführung wichtigsten sind Koordinatenpaare (`pair`), Zahlen oder Längenangaben (`numeric`) und Pfade (`path`). Soll eine solche Variable verwendet werden, so muss sie vorher folgendermaßen deklariert werden.

```
pair      <bezeichnung1>, <bezeichnung2>, ...;
numeric   <bezeichnung1>, <bezeichnung2>, ...;
path      <bezeichnung1>, <bezeichnung2>, ...;
```

Wird dabei der Name in der Form `name[]` deklariert, so können später numerische Indizes verwendet werden. Es sind dann alle Variablennamen der Form `name1`, `name2`, ... zulässig. Die Namen `x`, `y` und `z` sind vordefiniert und dürfen ohne vorherige Deklaration mit numerischen Indizes verwendet werden. Dabei ist `z` vom Typ `pair`, `x` und `y` vom Typ `numeric`. Wenn der Punkt

$z1$ bekannt ist, dann sind auch die Variablen $x1$ und $y1$ belegt, und zwar mit den Koordinaten des Punktes $z1$. Punkte können beispielsweise durch

$$z1 = (\langle dim1 \rangle, \langle dim2 \rangle);$$

festgelegt werden. Die Angabe von

$$x1 = \langle dim1 \rangle; \quad y1 = \langle dim2 \rangle;$$

ist äquivalent.

Ausgehend von Punkten können Pfade definiert werden, die als Polygone oder geglättete Kurven die Punkte verbinden. Die Form von geglätteten Kurven kann in METAPOST auf vielfältige Weise beeinflusst werden. In den meisten Fällen wird METAPOST aber ohne weitere Parameter eine gut geformte Kurve erzeugen. Gehen wir davon aus, dass **pair**-Variablen mit den Namen $z[]$ bereits belegt sind (entweder durch direkte Eingabe oder aus vorangegangenen Berechnungen), dann kann ein Polygonzug durch die Anweisung

$$\langle Pfad \rangle = z0--z1--z2--z3;$$

und eine geglättete Kurve durch die Anweisung

$$\langle Pfad \rangle = z0..z1..z2..z3;$$

definiert werden. Die Anweisungen `--` und `..` zum Verbinden zweier Punkte durch ein Geradenstück oder durch eine Kurve dürfen innerhalb einer solchen Pfaddefinition auch gemischt auftreten. In diesem Zusammenhang gibt es noch die Anweisung `---`, die bewirkt, dass beim Zusammentreffen mit einer `..`-Anweisung ein gerader und ein gebogener Pfadabschnitt glatt (also ohne Knick) ineinander übergehen.

Um den Verlauf einer von METAPOST erzeugten Kurve steuern zu können, gibt es zusätzliche Anweisungen, die man am besten [3] entnimmt, weil ihre Beschreibung hier zu weit führen würde. So gibt es etwa die Möglichkeit, Bézier-Punkte explizit anzugeben, die Krümmung und die Steigung der Kurve beim Erreichen oder Verlassen eines Stützpunktes festzulegen oder einen Spannungsparameter zu setzen.

Stifte

Zum Zeichnen von Pfaden mit den Befehlen **draw**, **drawarrow** usw. verwendet METAPOST Stifte, deren Spitzenform in Variablen vom Typ **pen** abgelegt

werden kann. Durch die Anweisung `pickup` wird ein bestimmter Stift aktiviert und bis zu einer erneuten `pickup`-Anweisung verwendet. Ein Stift mit kreisrunder Spitze und dem Durchmesser 1 ist unter dem Namen `pencircle` vordefiniert. Durch Skalierung kann mit der Anweisung

```
pickup pencircle scaled <dim>;
```

jede beliebige Strichstärke eingestellt werden. In L^AT_EX wird als Basisstrichstärke 0,4 pt verwendet. Es ist daher empfehlenswert, dieses in Zeichnungen beizubehalten. Für besondere Effekte ist die Definition eigener Stiftformen möglich. Das Vorgehen wird in [3] ausführlich beschrieben.

Durch den Zusatz `withpen` kann für einzelne Zeichenanweisungen ein individueller Stift verwendet werden. Wurde beispielsweise durch die Anweisung

```
pickup pencircle scaled 0.4pt;
```

eine globale Strichstärke gewählt, so kann etwa durch

```
draw <Pfad> withpen pencircle scaled 0.85pt;
```

ein einzelner Pfad mit größerer Strichstärke gezeichnet werden.

Richtungsangaben

Die Richtung, unter der ein Pfad einen Stützpunkt erreicht oder verlässt, kann durch Vor- oder Nachstellen einer `pair`-Variablen in geschweiften Klammern festgelegt werden. Die Richtung der Verbindung vom Koordinatenursprung zum Punkt $\langle pair \rangle$ entspricht dann der Tangentialrichtung des Pfades beim Eintreffen oder beim Verlassen des jeweiligen Stützpunktes. Besonders nützlich ist hierbei die `dir <Winkel>`-Anweisung. Sie erzeugt ein `pair` mit den Koordinaten $(\cos \langle Winkel \rangle, \sin \langle Winkel \rangle)$. Die Anwendung innerhalb einer Pfaddefinition geschieht dann beispielsweise in der Form

```
<Pfad> = z0{dir 30}..{dir -50}z1;
```

wodurch erzwungen wird, dass $\langle Pfad \rangle$ den Punkt `z0` unter 30° verlässt und im Punkt `z1` unter -50° ankommt. Zusätzlich sind die Angaben `{up}`, `{down}`, `{left}` und `{right}` vordefiniert, welche die gleiche Wirkung haben wie `{dir 90}`, `{dir -90}`, `{dir 180}` und `{dir 0}`.

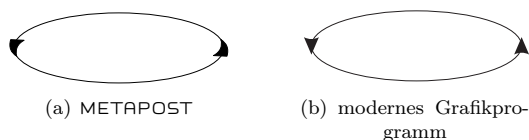


Abbildung 1: Vergleich von Pfeilen verschiedener Zeichenprogramme

Pfeile und Linientypen

Besonders angenehm fällt METAPOST durch seine Fähigkeit auf, schöne Pfeile auch an stark gekrümmte Kurven zu zeichnen. Die Pfeilspitzen sind nicht einfach Dreiecke, die auf der Linie in tangentialer Richtung orientiert sind ohne deren Verlauf zu beachten. METAPOST passt Pfeilspitzen in ihrer Form der lokalen Krümmung der Kurve an. Ohne diese Anpassung würden die Pfeilspitzen bereits bei mäßiger Krümmung von der Kurve „abreißen“, anstatt ihr zu folgen. Selbst modernste Grafikpakete weisen diese Unzulänglichkeit immer noch auf. Der Vergleich von Abbildung 1(a) und (b) zeigt diese Fähigkeit von METAPOST.

Zum Zeichnen von Pfeilen dienen die Anweisungen

```
drawarrow    <Pfad>;
drawdblarrow <Pfad>;
```

wobei im ersten Fall $\langle Pfad \rangle$ eine Spitze an seinem Ende erhält. Die zweite Anweisung zeichnet $\langle Pfad \rangle$ mit einer Pfeilspitze am Anfang und am Ende. Pfade in METAPOST haben einen Anfang und ein Ende und damit eine Richtung, die bei der Pfaddefinition festgelegt wird. Durch die Anweisung **reverse** wird allgemein die Richtung eines Pfades umgedreht. Das Pfadende, welches die Pfeilspitze erhalten soll, kann deshalb durch Hinzufügen oder Weglassen der **reverse**-Anweisung ausgewählt werden.

Verschiedene Linientypen werden innerhalb eines **draw**-Befehls durch die Zusatzanweisung **dashed** mit nachfolgender Angabe eines Strichmusters ausgewählt. Als vordefinierte Muster stehen bereits **evenly** und **withdots** zur Verfügung. Eine einfache Anweisung, einen Pfad gestrichelt zu zeichnen, wäre etwa

```
draw <Pfad> dashed evenly;
```

Durch Transformationen (s. u.) können die vordefinierten Muster auch einfach verändert werden. Die Strichlänge von `evenly` kann durch Skalieren verändert werden, sodass

```
draw <Pfad> dashed evenly scaled 0.5;
```

den Pfad mit einem engeren Strichmuster zeichnet. Für umfangreichere Aufgaben gibt es zudem den Befehl `dashpattern`, mit dem eigene Muster definiert werden können.

Beschriftungen

Mit Hilfe der Anweisung

```
label.<Suffix> (btex <Anweisung> etex,<Pair>);
```

können Punkte mit Beschriftungen versehen werden. Als Ort für die Beschriftung stehen dabei die acht Himmelsrichtungen O, NO, N, NW, W, SW, S und SO rund um den zu beschriftenden Punkt zur Verfügung. Die Auswahl der Platzierung geschieht dabei über `<Suffix>`, welches als mögliche Werte

```
rt, urt, top, ulft, lft, llft, bot und lrt
```

annehmen kann. Dabei steht `rt` für ‚right‘, `lft` für ‚left‘, `ulft` für ‚upper left‘ usw. Als `<Anweisung>` ist praktisch jede beliebige $\text{\LaTeX}$ -Anweisung zulässig, wenn sie nur durch die im Kopf geladenen Pakete abgedeckt ist.

Schnittpunkte

Schnittpunkte zwischen Pfaden können mit den Befehlen

```
<Paar> = <Pfad1> intersectiontimes <Pfad2>;
<Pfad> = <Pfad1> cutafter <Pfad2>;
<Pfad> = <Pfad1> cutbefore <Pfad2>;
```

bestimmt werden. Dabei erhält man entweder die zwei Längenparameter, bei denen sich der Schnittpunkt auf den beiden Pfaden befindet oder denjenigen Teil von `<Pfad1>`, der sich vor oder nach dem Schnittpunkt mit `<Pfad2>` befindet. Schwierigkeiten ergeben sich nur, wenn es mehrere Schnittpunkte gibt. METAPOST wählt dann einen aus, der möglicherweise nicht der gewünschte ist. In dieser Situation kann man zunächst versuchen, einen der beteiligten Pfade durch `reverse` umzukehren. Liefert METAPOST auch dann nicht das

gewünschte Ergebnis, so müssen den obenstehenden Makros Teilpfade übergeben werden, die sich sicher in nur einem Punkt schneiden. Mit etwas Übung versteht man sehr schnell, wie METAPOST in solchen Fällen „denkt“ und man wird die Pfade von Anfang an so anlegen, dass die gewünschten Ergebnisse vorliegen.

Transformationen

Besonders mächtig und nützlich ist die Möglichkeit, durch elementare Transformationen neue Punkte oder Pfade zu berechnen. Transformationen können auf Punkte, Pfade oder sogar ganze Bilder angewendet werden. Die verfügbaren Transformationen sind

$\langle \text{Objekt} \rangle$ rotated	$\langle \text{Numeric} \rangle$
$\langle \text{Objekt} \rangle$ scaled	$\langle \text{Numeric} \rangle$
$\langle \text{Objekt} \rangle$ shifted	$\langle \text{Pair} \rangle$
$\langle \text{Objekt} \rangle$ slanted	$\langle \text{Numeric} \rangle$
$\langle \text{Objekt} \rangle$ xscaled	$\langle \text{Numeric} \rangle$
$\langle \text{Objekt} \rangle$ yscaled	$\langle \text{Numeric} \rangle$
$\langle \text{Objekt} \rangle$ zscaled	$\langle \text{Pair} \rangle$
$\langle \text{Objekt} \rangle$ rotatedaround	$(\langle \text{Pair} \rangle, \langle \text{Numeric} \rangle)$
$\langle \text{Objekt} \rangle$ reflectedabout	$(\langle \text{Pair} \rangle, \langle \text{Pair} \rangle)$

Die Namen der Transformationen sind weitestgehend selbsterklärend, detailliertere Informationen sind in [3] zu finden. Es gibt einerseits die Möglichkeit, ein Objekt mittels **scaled** in beiden kartesischen Koordinatenrichtungen um den gleichen Faktor zu skalieren oder andererseits mit **xscaled** und **yscaled** verschiedene Faktoren auszuwählen. Eine Ellipse mit den Halbachsen $a = 40$ mm und $b = 20$ mm, die außerdem um 60° gegen den Uhrzeigersinn geneigt ist, würde man also durch den Befehl

```
 $\langle \text{Pfad} \rangle = \text{fullcircle xscaled 80mm yscaled 40mm rotated 60};$ 
```

erhalten. Bei der Verwendung mehrerer Transformationen innerhalb einer Anweisung ist entweder auf die richtige Reihenfolge oder auf richtige Klammerung zu achten. So liefert

```
 $\langle \text{Objekt} \rangle \text{ shifted } \langle \text{Pair} \rangle \text{ rotated } \langle \text{Numeric} \rangle;$ 
```

ein anderes Resultat als

```
 $\langle \text{Objekt} \rangle \text{ rotated } \langle \text{Numeric} \rangle \text{ shifted } \langle \text{Pair} \rangle;$ 
```

weil im ersten Fall zuerst $\langle Pair \rangle$ gedreht wird und dann $\langle Objekt \rangle$ um das hieraus resultierende Ergebnis verschoben wird.

Einbindung in L^AT_EX-Dokumente

Das Einbinden von METAPOST-Bildern in L^AT_EX-Dokumente geschieht wie bei allen anderen Bildformaten durch das **graphicx**-Paket. Nachdem das Paket mit

```
\usepackage[\(Treiber\)]{graphicx}
```

geladen wurde, können METAPOST-Bilder einfach durch

```
\includegraphics{\(Dateiname\)}
```

eingebunden werden. Dabei ist lediglich zu beachten, dass der verwendete DVI-Treiber in der Lage ist, EPS-Grafiken zu verarbeiten. An dieser Stelle sei auch darauf hingewiesen, dass die METAPOST-Bilddateien im EPS-Format vorliegen, aber dennoch keine selbstständigen Grafikdateien in dem Sinne sind, dass sie in beliebiger Weise weiterverarbeitet oder mit einem PostScript-Viewer betrachtet werden können. Sie sind sozusagen darauf angewiesen, in ein L^AT_EX-Dokument eingebunden und vom DVI-Treiber mit Fonts versorgt zu werden. Hinsichtlich einer kleinen Datenmenge ist dieses Vorgehen sehr sinnvoll. Es muss nicht jede Bilddatei von neuem Fonts mitbringen, die vom Rest des Dokuments ohnehin angefordert werden. Bei der Entwicklung von METAPOST-Zeichnungen erfordert dieses aber einen kleinen Kunstgriff, will man nicht zum Betrachten jedes Mal das gesamte Dokument übersetzen. Es ist hierzu ratsam, einfach ein kleines L^AT_EX-Dokument bereitzuhalten, welches nur die zur Einbindung einer Grafikdatei notwendigen Anweisungen enthält, und dieses zum Betrachten von einzelnen METAPOST-Bildern zu verwenden.

Für die Anwender von pdf(L^A)T_EX ist besonders angenehm, dass die METAPOST-Bilder zu diesem Zweck nicht erst in PDF konvertiert werden müssen, sondern von pdf(L^A)T_EX direkt eingelesen und umgewandelt werden. Hierzu ist außer der Option `pdftex` für das **graphicx**-Paket lediglich der Befehl

```
\DeclareGraphicsRule{*}{mps}{*}{} 
```

im Vorspann notwendig.

Beispiele

Die folgenden Beispiele sollen zeigen, auf welche Weise mit Hilfe von METAPOST ansprechende Zeichnungen erstellt werden können. Die notwendigen Anweisungen sind fast vollständig angegeben, sodass der Leser die Beispiele leicht nachvollziehen kann. Es soll dadurch auch deutlich werden, dass oft nur wenige Programmzeilen notwendig sind, um eine vollständige Zeichnung zu erzeugen.

Graph eines Fachwerks

Als erstes Beispiel soll der in Abbildung 2 gezeigte Graph eines Fachwerks mit METAPOST gezeichnet werden. Festgelegt sind dabei die Knotenpunkte z_0 , z_1 und z_2 . Die Knoten z_3 und z_4 sollen exakt in der Mitte zwischen den Verbindungen z_0 - z_1 und z_0 - z_2 liegen. Außerdem sollen die Stäbe z_3 - z_5 und z_4 - z_6 auf den Verbindungen z_0 - z_1 und z_0 - z_2 jeweils senkrecht stehen.

Nach beliebiger Zuweisung der Einheitslänge u werden mit

```
z0 = origin;
z1 = (-4u,-2u);
z2 = ( 4u,-2u);
```

zunächst die Eckpunkte festgelegt. Die Anweisungen

```
z3 = 0.5[z0,z1];
z4 = 0.5[z0,z2];
```

berechnen die Punkte z_3 und z_4 , sodass sie in der Mitte zwischen z_0 und z_1 bzw. z_2 liegen. Aus den Bedingungen

```
z5 = whatever[z1,z2];
z6 = whatever[z1,z2];
(z1-z0) dotprod (z5-z3) = 0;
(z2-z0) dotprod (z6-z4) = 0;
```

kann METAPOST schließlich die Punkte z_5 und z_6 berechnen.

Die Stäbe werden durch die Anweisungen mit einem etwas dickeren Stift gezeichnet.

```
pickup pencircle scaled 0.85pt;
draw z0--z1--z2--cycle;
draw z3--z5--z0;
```

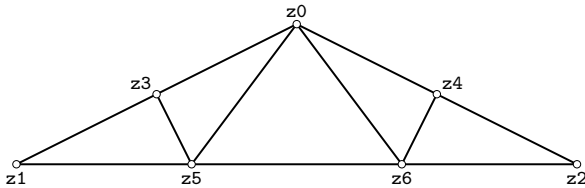


Abbildung 2: Ein Fachwerk aus elf Stäben und sieben Knoten

```
draw z4--z6--z0;
```

Die Markierung der Knoten durch kleine Kreise geschieht durch

```
pickup pencircle scaled 0.4pt;
for a = 0 upto 6:
    unfill fullcircle scaled 3pt shifted z[a];
    draw fullcircle scaled 3pt shifted z[a];
endfor
```

Ringhybridkoppler

Die Aufgabe, einen Ringhybridkoppler zu zeichnen, zeigt auch sehr schön die Überlegenheit von METAPOST gegenüber einer Grafikeingabe durch ein Zeigergerät. Ein Ringhybridkoppler ist ein Bauelement aus der Hochfrequenztechnik. Dabei handelt es sich um eine kreisförmige Leitung, von der unter den Winkeln 0° , 60° , 120° und 180° weitere Leitungen abzweigen. Die Verlängerungen der Leitungskanten gehen dabei nicht durch den Mittelpunkt der Ringleitung und die Abzweigleitungen sollen die gleiche Breite haben wie die Ringleitung (Abbildung 3).

Von der Außenkante der Ringleitung sind nur die Bögen zwischen den Kanten der abzweigenden Leitungen zu zeichnen. Es ist daher zweckmäßig, durch

```
path    k[], innenkreis, aussenkreis;
numeric a;

innenkreis = fullcircle scaled 10u;
ausсенkreis = fullcircle scaled 12u;
k1         = (0,-0.5u)--(9u,-0.5u);
```

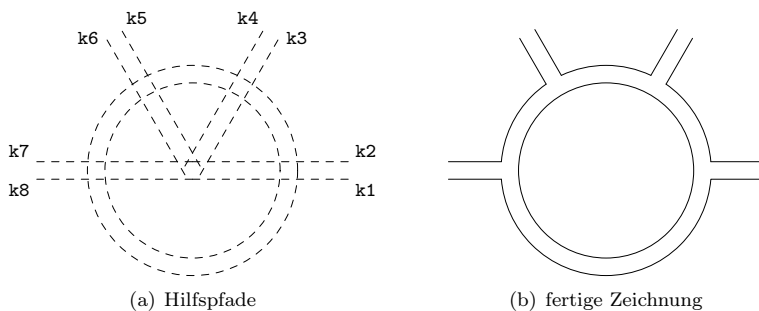


Abbildung 3: Zeichnung eines Ringhybridkopplers

```

k2      = (0, 0.5u)--(9u, 0.5u);
k3      = k1 rotated 60;
k4      = k2 rotated 60;
k5      = k1 rotated 120;
k6      = k2 rotated 120;
k7      = k1 rotated 180;
k8      = k2 rotated 180;

```

die Pfade in Abbildung 3a zu definieren. Durch Anwendung von `cutbefore` und `cutafter` werden dann die Teilbögen gezeichnet, sodass durch

```

pickup pencircle scaled 0.4pt;

draw innenkreis;
draw aussenkreis cutbefore k2 cutafter k3;
draw aussenkreis cutbefore k4 cutafter k5;
draw aussenkreis cutbefore k6 cutafter k7;
draw aussenkreis cutbefore k8 cutafter k1;

for a = 1 upto 8:
  draw k[a] cutbefore aussenkreis;
endfor

```

die Zeichnung wie in Abbildung 3b fertiggestellt wird.

Elektrostatisches Feld einer Punktladung

Als letztes Beispiel soll hier die Zeichnung des elektrostatischen Feldes einer positiven Punktladung dienen. Das Ergebnis ist in Abbildung 4 gezeigt. Zunächst wird mit

```
pickup pencircle scaled 0.4pt;

draw fullcircle scaled 0.4u;
draw (-0.12u,0)--(0.12u,0);
draw (0,-0.12u)--(0,0.12u);
```

die Punktladung mit einem Zeichen + gezeichnet. Die radial verlaufenden Feldlinien mit einer Pfeilspitze in ihrer Mitte entstehen durch die Anweisungen

```
path feldlinie;
numeric a;

feldlinie = (0.3u,0)--(2u,0);

for a = 0 upto 11:
    drawarrow subpath(0,0.5) of feldlinie rotated (a*30);
    draw      subpath(0.5,1) of feldlinie rotated (a*30);
endfor
```

die nach dem vorher Gesagten keine weiteren Erklärungen benötigen und leicht nachvollzogen werden können.

Schlusswort

Auch hier gilt die häufige Schlussbemerkung, dass in diesem Rahmen die ganzen Möglichkeiten und Funktionen von METAPOST nicht aufgezeigt werden können. Die Absicht des Autors war auch nicht die Erstellung einer Referenzkarte, sondern vielmehr den Appetit zu wecken und die Hemmschwelle zur Benutzung einer Beschreibungssprache für die Erstellung von Abbildungen abzubauen. Außerdem gibt es in METAPOST stets mehrere Wege, die zur selben Zeichnung führen. Hier ist sehr viel Spielraum für einen eigenen Programmierstil. Wenn die ersten Schritte einmal gewagt sind, wird man METAPOST aus seiner Grafikumgebung bald nicht mehr wegdenken können.

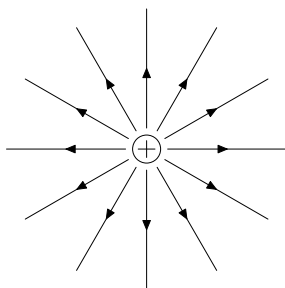


Abbildung 4: Elektrostatisches Feld einer Punktladung

Literatur

- [1] Michel Goossens, Sebastian Rahtz und Frank Mittelbach: *The L^AT_EX Graphics Companion*; Addison-Wesley Publishing Company; Reading, Mass.; 1997.
- [2] Hans Hagen: *Metafun*; Preliminary Version October 27, 2000; <http://www.pragma-ade.com>.
- [3] John D. Hobby: *A User's Manual for MetaPost*; Bell Laboratories; <http://cm.bell-labs.com/cm/cs/cstr/162.ps.gz>, siehe auch `doc/mpman.ps` in CTAN/graphics/metapost/metapost.tar.gz.
- [4] Alan Hoenig: *T_EX unbound*; Oxford University Press; Oxford; 1998.
- [5] Donald E. Knuth: *The METAFONT Book*; Bd. C of Computers and Typesetting; Addison-Wesley Publishing Company; Reading, Mass.; 1986.
- [6] Helmut Kopka: *L^AT_EX*; Bd. 2. Ergänzungen; Addison-Wesley; Bonn; 1997.

Die mathematischen Funktionen von Postscript

Herbert Voß

PostScript, faktisch genauso alt wie $\text{\TeX}$, ist im Verhältnis dazu allgemein noch weniger bekannt, wenn es darum geht zu beurteilen, was es denn nun im eigentlichen Sinne ist. Außerdem wird häufig vergessen, dass sich mit den PostScript-Funktionen viele Dinge erledigen lassen, bei denen sonst auf externe Programme zurückgegriffen wird. Dies wird im Folgenden für die mathematischen Funktionen im Zusammenhang mit dem Paket `pst-plot` gezeigt.

Einführung

PostScript, sehr häufig nur als Druckertreiber bekannt oder im Zusammenhang damit als Seitenbeschreibungssprache, ist ähnlich wie $\text{\TeX}$ eine Programmiersprache, wobei PostScript eindeutig als sogenannte Hochsprache bezeichnet werden muss.[1] Während die Ursprünge der Entwicklung bis in die Anfänge der siebziger Jahre zurückgehen, datiert die erste veröffentlichte PostScript-Version von 1982, was nicht zufällig das Jahr der Gründung von *Adobe System Incorporated* ist. Der erste Laserdrucker mit vollständiger PostScript-Implementierung war der *Apple Laserwriter*. Der einzige Computer, der PostScript auch für die Bildschirmausgabe (Display PostScript) verwendet, ist *Next*.

PostScript-Befehle

PostScript arbeitet mit dem so genannten Stack-System, welches den Benutzern von HP-Taschenrechnern geläufig und auch unter dem Namen *UPN*, *Umgekehrte Polnische Notation* (Reverse Polish Notation), bekannt ist und letztlich den internen Standard für alle Computer darstellt. Die normale Notation für die Multiplikation „ $a * b =$ “ wird zu „ $a <\text{enter}> b <\text{enter}> *$ “. Es sind immer zuerst die Parameter (Variablen) auf dem Stack abzulegen (durch `<enter>` symbolisiert), bevor eine der mathematischen Funktionen aufgerufen wird. Die hier beschriebenen Befehle beziehen sich immer auf das oberste Stack-Element oder die obersten beiden Stack-Elemente.

Die direkte Anwendung der PostScript-Befehle für die Darstellung mathematischer Zusammenhänge bringt gegenüber Programmen wie beispielsweise `gnuplot` nicht immer Vorteile in der endgültigen Druckausgabe. Auch lässt sich nicht unbedingt jedes mathematische Problem mit den PostScript-Befehlen einfach lösen. Vorteile ergeben sich durch die Übersichtlichkeit und vor allen Dingen in der Länge der Textdateien.

Anwendung mit `pst-plot`

Das umfangreiche Paket `pstricks` (CTAN:/graphics/pstricks/) ist nichts weiter als ein Frontend für $\text{\LaTeX}$ zu den PostScript-Funktionen. Das Paket `pst-plot` (CTAN:/graphics/pstricks/latex/pst-plot.sty), welches Teil des Pakets `pstricks` ist, ermöglicht mit zwei Befehlen die Anwendung der PostScript-Funktionen für das Zeichnen von mathematischen Funktionen. Mit der Anweisung `\usepackage{pst-plot}` wird automatisch das übergeordnete Paket `pstricks` geladen.

Die grundsätzliche Struktur der hier behandelten Befehle ist:

```
\psplot[Parameter]{xmin}{xmax}{Funktion f(x)}
\parametricplot[Parameter]{tmin}{tmax}{Funktionen x(t), y(t)}
```

Hierin bedeuten $[x_{min}; x_{max}]$ und $[t_{min}; t_{max}]$ das jeweilige Definitionsintervall (Start- und Endwert). Die möglichen Parameter können der Dokumentation zu `pstricks` entnommen werden, die zwar schon älteren Datums ist, jedoch völlig ausreicht [2]. Hier ist im Zusammenhang mit den Funktionen lediglich der Parameter `plotpoints` interessant, welcher die Anzahl der Stützstellen angibt und standardmäßig auf 50 gesetzt ist. Sämtliche berechneten Punkte werden intern mit `\psline` verbunden, so dass sich eine zu geringe Anzahl an Stützstellen durch einen Polygonzug bemerkbar macht. Mit Werten um 200 liegt man in den meisten Fällen auf der richtigen Seite. In diesem Zusammenhang ist unbedingt zu beachten, dass geänderte Parameter so lange ihren Wert behalten, bis innerhalb eines $\text{\LaTeX}$ -Laufs eine neue Wertzuweisung erfolgt.

Der Variablenname für `psplot` ist per Definition x und für `parametricplot` t . Beide können nicht verändert werden, was jedoch bezüglich der Anwendung keinerlei Einschränkung darstellt. Die Variablen können beliebig oft innerhalb eines Ausdrucks verwendet werden, denn erst mit der schließen der Klammer für den Funktionsausdruck wird davon ausgegangen, dass die zweite Koordinate für einen Punkt des Graphen oben auf dem Stack liegt.

Der einzige Unterschied zwischen diesen beiden Befehlen ist, dass bei `psplot` nur der oberste Stack-Wert (y) und bei `parametricplot` die obersten beiden Stack-Werte ($x; y$) als Argumente verwendet werden.

Zu beachten ist unbedingt, dass mit den beiden Befehlen keine Fehlermeldungen ausgegeben werden, was insbesondere für diejenigen mathematischen Funktionen von Interesse ist, deren Definitionsbereich nicht ganz $\mathbb{R}$ entspricht. Denn bei *einem* fehlerhaften Argument, beispielsweise $\sqrt{-1}$, wird der komplette Graph nicht gezeichnet!

Der Vollständigkeit halber sei erwähnt, dass `pst-plot` noch weitere drei Befehle zum Zeichnen von Funktionsgraphen zur Verfügung stellt, die jedoch alle externe Datensätze voraussetzen:

```
\fileplot[Parameter]{Dateiname}
\dataplot[Parameter]{Befehle}
\listplot[Parameter]{Liste}
```

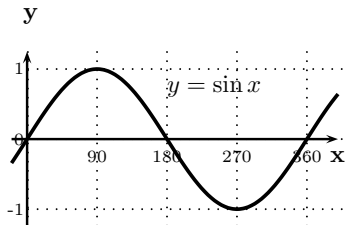
Weitere Informationen kann man der Dokumentation zu `pstricks` entnehmen [2].

Beispiele für `\psplot`

Für alle Beispiele wird jeweils die komplette `pspicture`-Umgebung angegeben, sodass eine direkte Übernahme der Beispiele möglich ist. Eine Dokumentation des `multido`-Befehls findet man unter CTAN:/macros/latex209/contrib/multido/multido.doc, alle anderen in der Dokumentation zu `pstricks` [2].

Sinusfunktion

Funktion	PostScript
$y(x) = \sin x$	<code>x sin</code>

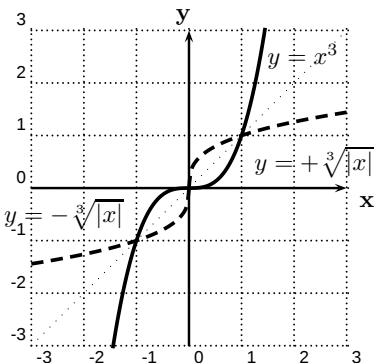


```
\psset{xunit=0.0111cm,yunit=1cm}
\begin{pspicture}(-20,-1.25)(400,1.25)
\psline[linewidth=1pt]{->}(-20,0)(400,0)
\psline[linewidth=1pt]{->}(0,-1.25)(0,1.25)
\multido{\n=90}{5}{\psline[linestyle=dotted]%
(\n,-1.25)(\n,1.25)\rput(\n,-0.25)%
{\scriptsize \n}}
\multido{\n=-1+1}{3}{\psline[linestyle=dotted]%
(0,\n)(405,\n)\rput[r]{-4,\n}{\scriptsize \n}}
\psplot[plotstyle=curve,linewidth=1.5pt]%
{-20}{400}{x sin}% postscript function
\rput[l]{-5,1.75}{\mathbf{y}}
\rput[l]{390,-.25}{\mathbf{x}}
\rput[l]{180,0.75}{y=\sin x}
\end{pspicture}
```

Potenzfunktion

Dargestellt wird eine Parabel dritten Grades sowie ihre Umkehrfunktion, wobei die Intervallunterscheidung nicht zwingend ist, wenn man die Exponentialschreibweise mit $y = x^{-\frac{1}{3}}$ wählt.

Funktion	PostScript
$y(x) = x^3$	x 3 exp
$y^{-1}(x) = \begin{cases} +\sqrt[3]{x} & x > 0 \\ -\sqrt[3]{x} & x < 0 \end{cases}$	x 0.333 exp (Umkehrfunktion)



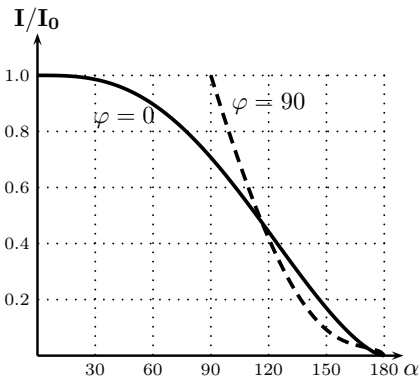
```
\begin{pspicture}(-3,-3)(3,3)
\psgrid[subgriddiv=1,griddots=10,%
gridlabels=7pt](-3,-3)(3,3)%
\psline[linewidth=1pt]{->}(-3,0)(3,0)%
\psline[linewidth=1pt]{->}(0,-3)(0,3)%
\psline[linewidth=0.5pt,linestyle=dotted]%
(-3,-3)(3,3)%
\psplot[plotstyle=curve,linewidth=1.5pt]%
{-1.4}{1.4}{x 3 exp}% postscript function
\psplot[plotstyle=curve,linewidth=1.5pt,%
linestyle=dashed]{0}{3}{x 0.333 exp}%
\psplot[plotstyle=curve,linewidth=1.5pt,%
linestyle=dashed]{-3}{0}%
{x -1 mul 0.333 exp -1 mul}%
\rput[l]{-.25,3.5}{\mathbf{y}}%
\rput[l]{3.5,-.25}{\mathbf{x}}%
\rput[l]{1.5,2.5}{y=x^3}%
\rput[l]{1.25,0.5}{y=+\sqrt[3]{|x|}}%
\rput[r]{-1.25,-0.5}{y=-\sqrt[3]{|x|}}%
\end{pspicture}
```

Beispiel aus der Leistungselektronik

Gezeigt wird die graphische Darstellung des relativen Strommittelwertes für eine Stromrichtersteuerung durch ein Thyristorpaar. Hierbei entspricht der Parameter φ der Phasenverschiebung zwischen Strom und Spannung. Die unabhängige Variable α bezeichnet den Steuerwinkel.

Funktion	$\frac{I(\alpha)}{I_0} = \begin{cases} \sqrt{1 - \frac{\alpha}{\pi} + \frac{1}{2\pi} \sin 2\alpha} & \varphi = 0 \\ \sqrt{(2 - \frac{2\alpha}{\pi})(2 + \cos 2\alpha) + \frac{3}{\pi} \sin 2\alpha} & \varphi = \frac{\pi}{2} \end{cases}$
PostScript	<pre> 1 x 180 div sub 1 6.28 div x 2 mul sin $\varphi = 0$ mul add sqrt 2 x 90 div sub x 2 mul cos 2 add mul x 2 $\varphi = \frac{\pi}{2}$ mul sin 3 3.15 div mul add sqrt </pre>

Zu beachten ist, dass PostScript die Argumente für die trigonometrischen Funktionen im Gradmaß erwartet, sodass für relative Winkel auf gleiche Einheiten zu achten ist. Der Ausdruck $\frac{\alpha}{\pi}$ ist daher durch $\frac{\alpha}{180}$ zu ersetzen.



```

\psset{xunit=0.0333cm,yunit=3cm}
\begin{pspicture}(0,-0.25)(190,1.25)
\psline[linewidth=1pt]{->}(0,0)(190,0)
\psline[linewidth=1pt]{->}(0,0)(0,1.15)
\multido{\n=30+30}{6}{\psline[%
linestyle=dotted](\n,0)(\n,1)%
\rput(\n,-0.05){\scriptsize \n}}
\multido{\n=0.2+0.2}{5}{\psline[%
linestyle=dotted](0,\n)(180,\n)%
\rput[r](-4,\n){\scriptsize \n}}
\psplot[plotstyle=curve,%
linewidth=1.5pt]{0}{180}%
{1 x 180 div sub 1 6.28 %
div x 2 mul sin mul add sqrt}
\psplot[plotstyle=curve,%
linewidth=1.5pt,linestyle=dashed]%
{90}{180}{2 x 90 div sub x 2 mul cos%
2 add mul x 2 mul sin 3 3.15%
div mul add sqrt}
\rput(0,1.2){\mathbf{I/I_0}}
\rput[1](190,-.05){\mathbf{\alpha}}
\rput(45,0.85){\varphi=0}
\rput(120,0.9){\varphi=90}

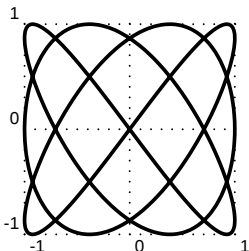
```

Beispiele für `\parametricplot`

Lissajous-Figur

Die aus der Physik oder Elektrotechnik bekannten Lissajous-Figuren sind ein typischer Anwendungsfall für Gleichungen in Parameterform. Die hier angegebene Darstellung beruht auf den Funktionen:

Funktion	PostScript
$x = \sin 1.5t$	<code>t 1.5 mul sin</code>
$y = \sin\left(2t + \frac{\pi}{3}\right)$	<code>t 2 mul 60 add sin</code>



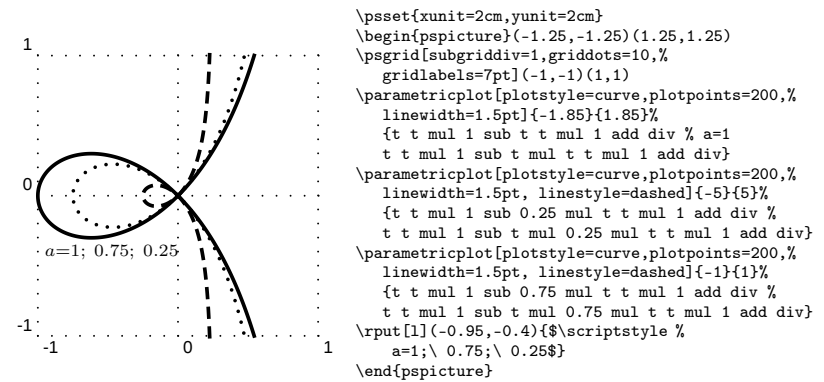
```
\psset{xunit=1.5cm,yunit=1.5cm}
\begin{pspicture}(-1,-1)(1,1)
\psgrid[subgriddiv=0,griddots=10,gridlabels=7pt]%(
-1,-1)(1,1)
\parametricplot[plotstyle=curve,linewidth=1.5pt,%
plotpoints=200]{-360}{360}{%
t 1.5 mul sin t 2 mul 60 add sin}
\end{pspicture}
```

Aufgrund der „Länge“ des Graphen wurde der Wert für `plotpoints` auf 200 gesetzt, sodass auch die „Ecken“ des Graphen mit starker Krümmung kontinuierlich erscheinen.

Strophoide

Zum Schluss wird eine sogenannte Strophoide dargestellt, die durch folgende Beziehungen gegeben ist, wobei a durch einen Zahlenwert ersetzt werden muss:

Funktion	PostScript
$x(t) = \frac{a(t^2 - 1)}{t^2 + 1}$	<code>t t mul 1 sub a mul t t mul 1 add div</code>
$y(t) = \frac{at(t^2 - 1)}{t^2 + 1}$	<code>t t mul 1 sub t mul a mul t t mul 1 add div</code>



Zusammenstellung der mathematischen PostScript-Funktionen

Die folgende tabellarische Zusammenstellung enthält bis auf die Matrizenbefehle alle mathematischen Funktionen mit ihren Eigenschaften. Hierin sind *<int>* und *<real>* die bekannten Integer und Reals, während für *<any>* jeder beliebige Typ und für *<num>* *real* oder *integer* gesetzt werden kann. Unabhängig von der Null gelten für Zahlen unter PostScript die folgenden Grenzen, wobei die Angaben für *real* betragsmäßig zu verstehen sind.

	<i>integer</i>	<i>real</i>
kleinster Wert	-2^{32}	$\pm 10^{-38}$
größter Wert	$2^{32} - 1$	$\pm 10^{38}$

<i>Name</i>	<i>Bedeutung</i>	<i>Anwendung</i>	<i>Beispiel</i>
abs	Absolutwert	<i><num></i> abs	-3 abs → 3
add	Addition ¹	<i><num1></i> <i><num2></i> add	5 7 add → 12
atan	Arcus Tangens ²	<i><real1></i> <i><real2></i> atan	2 45 atan→ 2.54
cos	Cosinus ³	<i><real></i> cos	60 cos→ 0.5
cvi	Real→Integer	<i><real></i> cvi	14.13 cvi→ 14
cvr	Integer→Real	<i><int></i> cvr	14 cvr→ 14.00
div	Division ⁴	<i><real1></i> <i><real2></i> div	100 8 div→ 12.5
dup	Dupliziere oberstes Stack-Element	<i><any></i> dup	12 dup→ 12 12

Name	Bedeutung	Anwendung	Beispiel
exch	Exchange ⁵	<any1> <any2> exch	12 13 exch → 13 12
exp	Potenz	<real1> <real2> exp	3 4 exp → 81.0
idiv	ganzzahlige Division	<int1> <int2> idiv	100 8 idiv → 12
ln	natürlicher Logarithmus	<real> ln	12 ln → 2.48491
log	Zehner- Logarithmus	<real> log	1000 log → 3.00
mod	Modulo	<int1> <int2> mod	5 3 mod → 2
mul	Multiplikation ¹	<num1> <num2> mul	5 3 mul → 15
neg	Negiere Vorzeichen	<num> neg	5 neg → -5
round	Runden	<real> round	5.7 round → 6
sin	Sinus ³	<real> sin	30 sin → 0.5
sqrt	Quadratwurzel	<real> sqrt	16 sqrt → 4.0
sub	Subtraktion ¹	<num1> <num2> sub	17 19 sub → -2
truncate	Dezimalteil abtrennen ⁶	<real> truncate	-33.33 → 33.00

Anmerkungen:

¹ Sind beide Argumente ganze Zahlen ist auch das Ergebnis vom Typ Integer.

² Entspricht der Darstellung $\alpha = \arctan \frac{\langle real1 \rangle}{\langle real2 \rangle}$.

³ Das Argument wird im Gradmaß erwartet.

⁴ Das Ergebnis ist grundsätzlich vom Typ *real*.

⁵ Die Anordnung der letzten beiden Stack-Elemente wird vertauscht.

⁶ Das Ergebnis bleibt vom Typ *real*.

Literatur

- [1] Nikolai G. Kollock: *PostScript richtig eingesetzt: vom Konzept zum praktischen Einsatz*; IWT; Vaterstetten; 1989.
- [2] Timothy Van Zandt: *PSTricks - PostScript macros for Generic TeX*;
<http://www.tug.org/application/PSTricks>; 1993.

Rezensionen

„ $\text{\LaTeX}$ echt einfach“ von Roland Willms

Knut Lickert

Jedesmal wenn ich in die Bibliothek gehe, schaue ich kurz in der EDV-Abteilung, welche $\text{\LaTeX}$ -Bücher gerade nicht verliehen sind. Diesmal stoße ich auf ein neues Einführungsbuch: „ $\text{\LaTeX}$ “ aus der Reihe *echt einfach – die kinderleichten Computer-Bücher*. Mein erster Eindruck: Das Buch ist wirklich einfach und wendet sich an $\text{\LaTeX}$, teilweise gar an Computer-Anfänger. Das Buch zeigt Schritt für Schritt, wie $\text{\TeX}$ installiert wird und wie Dokumente erstellt werden. Für eine etwas eingehendere Untersuchung leihe ich mir das Buch aus.

Die Einleitung geht kurz auf die Geschichte von $\text{\TeX}$ und $\text{\LaTeX}$ ein. Es werden die verwendeten Symbole und Schriften erläutert sowie auf die Internet-Seite <http://www.jlauncher.com> verwiesen, wo der Autor sein Programm *Jlauncher* auch als $\text{\LaTeX}$ -Umgebung empfiehlt. Etwas irritierend für den $\text{\LaTeX}$ -Kenner: kein Wort zu Betriebssystemen. Im weiteren Verlauf des Buchs wird klar, dass der Autor Windows verwendet, unter anderem, wenn er auf Unterschiede zwischen Windows 95/98/ME und Windows NT 4/2000 eingeht. Leider versäumt er zu erwähnen, dass $\text{\TeX}$ betriebssystemunabhängig ist und auf allen gängigen Plattformen läuft – eine seiner Stärken.

Auf 35 Seiten wird erklärt, wie $\text{\MiKTeX}$, Ghostscript und Acrobat Reader installiert werden. Die Programme werden auf der beiliegenden CD-ROM mitgeliefert. Problematisch erscheint mir in Hinblick auf schon vorhandene $\text{\TeX}$ -Installationen der Tipp, eine eventuelle Verknüpfung der Dateierweiterung `.tex` mit anderen Anwendungen durch Assoziation mit dem Windows-Editor `notepad` zu ersetzen.

Bereits die Installationsbeschreibung sieht die Erstellung des ersten Dokuments vor, mit dem man die Installation testen kann. In diesem Beispiel wird erklärt, wie der Windows-Explorer bedient, der Editor gestartet und $\text{\LaTeX}$

von der MS-DOS-Eingabeaufforderung aus aufgerufen wird. Im ersten Dokument gleich eine mathematische Formel einzuführen, mag die Leistungsfähigkeit von $\TeX$ demonstrieren, erscheint mir als Einstieg aber etwas gewagt. Neben $\TeX$ wird die Konvertierung von DVI-Dateien zu PostScript und PDF erläutert. Einen Hinweis auf pdf $\TeX$ konnte ich nicht finden.

Im Folgenden erläutert der Autor, wie man $\LaTeX$ -Dokumente schreibt und erklärt die wichtigsten Befehle und Umgebungen. Alle Erläuterungen beziehen sich auf die Mi $\TeX$ -Installation von der beigelegten CD-ROM. Zwar weist der Autor darauf hin, dass im CTAN viele Erweiterungen zu finden sind, mir fehlen allerdings einige wichtige Beispiele. Kennt man nur dieses Buch, glaubt man, es gebe nur die Standardklassen `letter`, `article`, `report` oder `book`. Anpassungen an europäische Verhältnisse wie die Koma-Klassen werden nicht erwähnt. Als Neuling lernt man das Paket `german` mit der Eingabemöglichkeit "a für ä kennen; die Existenz eines Pakets wie `inputenc` wird nicht erwähnt, `ngerman` auch nicht.

Es werden viele Manipulationsmöglichkeiten gezeigt (Änderung der Schriftgröße, Anpassung des Seitenlayouts); das Konzept der logischen Auszeichnung (Markup) wird jedoch nicht erläutert. Wie neue Pakete vom CTAN installiert werden, ist im Abschnitt „Textelemente drehen“ versteckt.

Interessant ist ein Abschnitt über die Möglichkeit, mittels `Imagemagick` von $\TeX$ erzeugte Textabschnitte über PostScript in Bitmaps und GIF-Bilder umzuwandeln, um sie anschließend in andere Programme einzubinden. Etwas aus dem Rahmen fällt ein Abschnitt, in dem die Einbindung des Paketes `tipa` erläutert wird. Man bekommt den Eindruck, der Autor verwendet $\LaTeX$ nur, um kurze Texte mit mathematischen Formeln oder Lautschrift zu schreiben, die er dann als Bilder in WinWord einbindet.

Das Buch ist von der Aufbereitung eher ein Windows- als ein $\LaTeX$ -Buch. Auf die $\LaTeX$ -Einstiegsprobleme von Windows-Nutzern wird durchaus eingegangen; wenn man dann aber keine weitere Literatur liest, wird der Einsteiger von der weiteren Verwendung von $\LaTeX$ womöglich eher abgeschreckt. Die Stärken von $\LaTeX$ werden nicht gezeigt, weitere Pakete, die Schwächen beheben, nicht erwähnt.

Beim Lesen war ich des Öfteren versucht, meine Kommentare in das Buch zu schreiben. Ich kann mir gut vorstellen, das Buch einem mit Windows arbeitenden Bekannten für seine ersten Schritte mit $\TeX$ zu geben. Begleitende Kommentare und fortführende Literatur sind allerdings unabdingbar,

um $\TeX$ wirklich nutzen zu können. Es würde mich freuen, wenn das Buch vor einer Neuauflage gründlich überarbeitet würde.

Roland Willms

$\LaTeX$ echt einfach

Franzis' Verlag, Pöcking, 2001

ISBN 3-7723-6599-X

15,31 €

Spielplan

Termine

- 29.4–3.5.2002** Annual Meeting of European T_EX Users Group
Bachotek (Brodnica Lake District), Poland
<http://www.gust.org.pl/EuroBachoTeX/>
Kontakt: Andrzej Borzyszkowski
“Special arrangements for DANTE members” unter
<http://www.dante.de/events/kalender/EuroBachoTeX2002.shtml>
- 12.–14.7.2002** TypeCon2002, “The Medium and the Message”
Toronto, Ontario, Kanada
<http://www.typecon2002.com>
- 3.–7.9.2002** TUG 2002 23rd annual meeting of the T_EX User Group
International Convention Center at Technopark
Trivandrum, Kerala, Indien
<http://www.tug.org.in/tug2002/>
Kontakt: Indian T_EX Users Group
- 20.–26.7.2003** TUG2003 Waikaloa Beach Resort, Big Island, Hawaii
<http://www.tug.org/tug2003/>
Kontakt: Wendy McKay

Stammtische

In verschiedenen Städten im Einzugsbereich von DANTE e.V. finden regelmäßig Treffen von $\text{\TeX}$ -Anwendern statt, die für jeden offen sind. Im WWW gibt es aktuelle Informationen unter <http://www.dante.de/events/stammtische/>.

Berlin

Rolf Niepraschk
Tel.: 0 30/3 48 13 16
niepraschk@ptb.de
Gaststätte „Bärenschenke“
Friedrichstr. 124
Zweiter Donnerstag im Monat, 19.00 Uhr

Bremen

Martin Schröder
Tel.: 04 21/2 23 94 25
martin@oneiros.de
Wechselnder Ort
Erster Donnerstag im Monat, 18.30 Uhr

Chemnitz

Ralf König
Tel.: 03 71/5 90 54 75
ralf.koenig@s1998.tu-chemnitz.de
Universitätsteil 1, Straße der Nationen 62,
Raum 1/068
Dritter Mittwoch im Monat, 18.00 Uhr

Dortmund

Stephan Lehmke
Stephan.Lehmke@cs.uni-dortmund.de
Cafe Durchblick
Universität Dortmund, Campus Nord
Zweiter Mittwoch im Monat, 20.00 Uhr

Dresden

Hilmar Preuß
hille42@gmx.de
Medien- und Kulturzentrum Pentacon,
Schandauer Str. 64
Erster Mittwoch im Monat, 19.00 Uhr

Erlangen

Walter Schmidt, Peter Seitz
was@VR-Web.de,
Gaststätte „Gambrinus“
Vierzigmannstr. 7
Dritter Dienstag im Monat, 19.00 Uhr

Freiburg

Heiko Oberdiek
Tel.: 07 61/4 34 05
oberdiek@ruf.uni-freiburg.de
Gaststätte „Aguila“
Sautierstr. 19
Dritter Donnerstag im Monat, 19.30 Uhr

Hamburg

Volker Hüttenrauch
volker_huettenrauch@hh.maus.de
Vereinsheim der Hamburger
Microcomputer-Hochschulgruppe
Grindelallee 143 (Hinterhof)
Letzter Donnerstag im Monat, 18.00 Uhr

Hannover

Mark Heisterkamp
heisterkamp@rrzn.uni-hannover.de
Seminarraum RRZN
Schloßwender Str. 5
Zweiter Mittwoch von geraden Monaten,
18.30 Uhr

Heidelberg

Luzia Dietsche
Tel.: 0 62 21/54 45 27
luzia.dietsche@urz.uni-heidelberg.de
China-Restaurant „Palast“
Lessingstr. 36
Letzter Mittwoch im Monat, 20.00 Uhr

Karlsruhe

Klaus Braune
Tel.: 07 21/6 08 40 31
braune@rz.uni-karlsruhe.de
Universität Karlsruhe, Rechenzentrum
Zirkel 2, 3. OG, Raum 316
Erster Donnerstag im Monat, 19.30 Uhr

Köln

Bruno Hopp
b.hopp@lepkes-frings.de
Institut für Kristallographie
Zülpicher Str. 49b
Letzter Mittwoch im Monat, 19.30 Uhr

Konstanz

Matthias Weisgerber, Hraban Ramm
weisgerb@fmi.uni-konstanz.de,
hraban@fief.net
Restaurant Rheingold
Spanierstrasse 3
Zweiter Donnerstag im Monat, 19.00 Uhr

München

Michael Niedermair
m.g.n@gmx.de
Gastwirtschaft „Rhaetenhaus“
Luisenstr. 27
Erster Dienstag im Monat, 19.00 Uhr

Münster

Johannes Reese
reesej@uni-muenster.de
Gaststätte „Sabroso“
Mauritzstr. 19
Erster Montag im Monat, 20.00 Uhr

Stuttgart

Marcus Schweizer
 Tel.: 07 11/6 85 44 44
schweiz@theochem.uni-stuttgart.de
Gaststätte „Alte Mira“
Büchsenstr. 24
Zweiter Dienstag im Monat, 19.30 Uhr

Wuppertal

Andreas Schrell
 Tel.: 02 02/50 63 81
schrell@wupperonline.de
Restaurant Croatia „Haus Johannisberg“
Südstr. 10
an der Schwimmooper Wuppertal-Elberfeld
Zweiter Donnerstag im Monat, 19.30 Uhr

Adressen

DANTE, Deutschsprachige Anwendervereinigung T_EX e.V.
Postfach 10 18 40
69008 Heidelberg

Tel.: 0 62 21/2 97 66 (Mo, Mi–Fr, 10⁰⁰–12⁰⁰ Uhr)

Fax: 0 62 21/16 79 06

E-Mail: dante@dante.de

Konten: Volksbank Rhein-Neckar eG

BLZ 670 900 00

Kontonummer 2310 007

Postbank Karlsruhe (Auslandsüberweisungen)

BLZ 660 100 75

Kontonummer 213 400 757

Präsidium (bis zum 23. 2. 2002)

Präsident: Thomas Koch president@dante.de

Vizepräsident: Volker RW Schaa vice-president@dante.de

Schatzmeister: Horst Szillat treasurer@dante.de

Schriftführer: Günter Partosch secretary@dante.de

Beisitzer: Klaus Höppner adviser@dante.de

Server

ftp: [ftp.dante.de](ftp://ftp.dante.de) [134.100.9.51]

E-Mail: ftpmail@dante.de

WWW: <http://www.dante.de/>

Autoren/Organisatoren

Sascha Beuermann Institut für Baumechanik und Numerische Mechanik Universität Hannover Appelstraße 9A 30167 Hannover <code>beuermann@ibnm.uni-hannover.de</code>	[8]	Wendy McKay <code>wgm@cds.caltech.edu</code>	[51]
Andrzej Borzyszkowski <code>EuroBachTeX@gust.org.pl</code>	[51]	Gerd Neugebauer Mainzer Str. 8 56321 Rhens <code>gene@gerd-neugebauer.de</code>	[3]
Indian T_EX Users Group 3rd Floor, SJP Buildings Cottons Hills, Trivandrum 695 014, India <code>tugindia@tugindia.org.in</code>	[51]	Volker RW Schaa siehe Seite 54	[6]
Thomas Koch siehe Seite 54	[4]	Uwe Siart Aribonenstr. 1 81669 München	[26]
Knut Lickert Franziskanergasse 15 73728 Esslingen <code>knut@lickert.net</code>	[48]	Herbert Voß Wasgenstr. 21 14129 Berlin <code>voss@perce.de</code>	[40]

Die T_EXnische Komödie

14. Jahrgang Heft 1/2002 März 2002

Impressum

Editorial

Hinter der Bühne

- 4 Grußwort
- 6 Grußwort, 2. Teil

Bretter, die die Welt bedeuten

- 8 Erstellung von leistungsfähigen PDF-Dokumenten
- 26 Ansprechende technische Illustration mit METAPOST
- 40 Die mathematischen Funktionen von Postscript

Rezensionen

- 48 „L^AT_EX echt einfach“ von Roland Willms

Spielplan

- 51 Termine
- 52 Stammtische

Adressen

- 55 Autoren/Organisatoren