

**P<sup>4</sup>**  
**(PPower4)**  
**PDF PRESENTATION POST PROCESSOR**  
Präsentationen mit L<sup>A</sup>T<sub>E</sub>X und Acrobat Reader

Klaus Guntermann  
TU Darmstadt  
Fachbereich Informatik  
Fachgebiet Systemprogrammierung  
Bericht TI-14/99  
  
September 1999

## Inhaltsverzeichnis

<b>1</b>	<b>Überblick</b>	<b>2</b>
<b>2</b>	<b>Anforderungen und Umfeldanalyse</b>	<b>2</b>
2.1	PowerPoint etc. . . . .	2
2.2	xdvi . . . . .	3
2.3	Acrobat Reader . . . . .	3
2.3.1	Anzeigeeigenschaften . . . . .	3
2.3.2	Erzeugung des pdf-Dateiformats . . . . .	4
<b>3</b>	<b>Aufgaben von PPower4</b>	<b>4</b>
3.1	Implementierung allein über den Textformatierer . . . . .	5
3.2	Nachbearbeitung . . . . .	7
3.3	Implementierung . . . . .	8
<b>4</b>	<b>Erweiterungen</b>	<b>9</b>
4.1	Ein- und Ausblenden . . . . .	9
4.2	Animationen . . . . .	9
4.3	Hintergründe . . . . .	9
<b>5</b>	<b>Benutzeranleitung</b>	<b>10</b>
5.1	Installation des Postprozessors . . . . .	10
5.2	Einbau im Dokument . . . . .	10
5.3	Aufruf des Postprozessors . . . . .	11
5.4	Beispiele . . . . .	11
<b>6</b>	<b>Zusammenfassung</b>	<b>15</b>

# 1 Überblick

Immer mehr Vorlesungsräume werden mit einem Datenprojektor („Beamer“) ausgestattet, so daß man auf die Vorbereitung von realen Folien für einen Overheadprojektor verzichten kann. Das führt aber zu der Frage, wie man die darzustellenden Inhalte geeignet aufbereitet und welche Software für die Vorführung benutzt wird.

In diesem Bericht wird ein Werkzeug beschrieben, das es erlaubt, auch Inhalte mit einem hohen Anteil von mathematischen Formeln in für  $\text{T}_{\text{E}}\text{X}$ -Benutzer gewohnter Weise aufzubereiten und in guter Darstellungsqualität zu präsentieren. Dabei wird zunächst beleuchtet, warum man nicht mit bereits vorhandenen Hilfsmitteln auskommen kann. Schließlich wird dargelegt, welche Kompromisse man weiterhin machen muß.

## 2 Anforderungen und Umfeldanalyse

Für die Projektion einer Präsentation (zu denen man auch Vorlesungsinhalte zählen muß), bieten sich die folgenden Programme an:

- spezielle Präsentationsprogramme wie PowerPoint<sup>1</sup> [Koh94] (für Windows) oder MagicPoint (für Linux/Unix),
- das Preview-Programm `xdvi`, das für das Korrekturlesen der Ausgaben des Textsatzprogramms  $\text{T}_{\text{E}}\text{X}$  [Knu86] unter Unix verwendet wird, und
- Acrobat Reader für `pdf`-Dateien [Ado99].

Wenn man in einer Präsentation nicht nur farbige Bilder und Text vermitteln will, sondern auf die nahtlose Einpassung von Formeln angewiesen ist, bleibt nicht viel Auswahl. Bei vereinzelt Formeln mag man sich noch auf die Einbindung von Graphikelementen für jede Formel einlassen. Diese Vorgehensweise ist jedoch bei einer mathematisch orientierten Text mit vielen Formeln zu umständlich.

Im Einzelnen sind bei den verwendbaren Programmen folgende Eigenschaften zu berücksichtigen:

### 2.1 PowerPoint etc.

MagicPoint ist als freie Software für Unix verfügbar. Die Eingabe kann mit einem beliebigen Texteditor erstellt werden, läßt aber keine Formeln zu. Allerdings können Graphiken eingebunden werden. Aufzählungen von Stichpunkten sind leicht zu erstellen. Dabei kann auch die Präsentation innerhalb einer Seite unterbrochen und nach einer Interaktion fortgesetzt werden. Farbige gestaltete Hintergründe und einfache animierte Texte werden ebenfalls unterstützt.

PowerPoint ist als kommerzielle Software für die Benutzung auf einem PC unter Windows bekannt und hat dort eine starke Verbreitung gefunden. Formeln können jedoch wiederum nur als Bilder eingebunden werden. Die Bearbeitung von Präsentationen kann nur im System selbst erfolgen. Dies schränkt

---

<sup>1</sup>Warenzeichen werden in diesem Text ohne besondere Kennzeichnung verwendet. Daraus darf nicht geschlossen werden, daß alle verwendeten Bezeichnungen frei von Rechten der Inhaber von Warenzeichen sind.

die Portabilität ein, erlaubt jedoch eine breite Auswahl von Spezialeffekten, die über Menüs abrufbar sind. Die Auswahl geht von Hintergrunddefinitionen bis zu Animationen und erlaubt z. B. auch, Textbereiche einer Seite dynamisch ein- und auszublenden bzw. je nach Fortschreiten der Präsentation hervorzuheben oder in den Hintergrund treten zu lassen.

Bei wissenschaftlichen Texten müssen Formeln leicht und in gewohnter Weise beschreibbar sein. Dies ist bei den Präsentationsprogrammen PowerPoint und MagicPoint nicht gegeben. Damit wird die Benutzung dieser beiden Programme zu unhandlich, wenngleich sie bei der Präsentation mit ihren dynamischen Effekten das Entwickeln von Gedankengängen visuell unterstützen können und eine Vielzahl von weiteren Gestaltungsmöglichkeiten bieten.

## **2.2 xdvi**

Das Programm `xdvi` ist entstanden als Hilfsmittel zum Korrekturlesen. Seine Fähigkeiten als Präsentationsprogramm sind daher eingeschränkt. So sind dynamische Effekte (z. B. das Aufrollen eines Textes) nicht abrufbar und auch ein Ganzseitenmodus muß durch explizite Konfigurierung des Programms und des Windowmanagers simuliert werden. Dabei bleibt problematisch, daß wegen der Verwendung von Zeichensätzen im Bitrasterformat die Anpassung an unterschiedliche Bildschirmauflösungen nur näherungsweise möglich ist. Es ist zu befürchten, daß man bei Verwendung eines fremden Rechners bei einem externen Vortrag nicht problemlos die gewünschte Skalierung für seine Präsentation erreicht. Auch kann die Aufbereitung von PostScript-Graphiken im Hintergrund durch ein externes Programm (GhostScript) erhebliche Verzögerungen bei der Präsentation hervorrufen. Den Eindruck eines augenblicklichen Seitenwechsels erreicht man nur auf sehr schnellen Rechnern und wenn keine Graphiken eingebunden werden. Da sich die präsentierten Dokumente noch auf externe Graphik- und Schriftdateien beziehen, muß man für eine Präsentation eine Vielzahl von Dateien vorhalten.

## **2.3 Acrobat Reader**

Als reines Anzeigeprogramm ist der Acrobat Reader kostenfrei für die gängigsten Rechnerplattformen verfügbar. Leider liegt kein Quelltext des Programms vor, so daß man bei Fehlern auf die Reparatur durch den Hersteller angewiesen ist. Für die Erzeugung von Dateien für den Acrobat Reader ist man nicht mehr auf die kostenpflichtige Programm Distiller angewiesen, sondern es gibt inzwischen eine Auswahl auch an freien Programmen, die das von Acrobat Reader verarbeitete Dateiformat erzeugen können.

### **2.3.1 Anzeigeeigenschaften**

Bei Präsentationen mit dem Acrobat Reader muß man auf das Aufrollen eines Textes leider verzichten, weil nur das Umblättern zwischen verschiedenen Seiten und das Verfolgen von Hypertext-Verknüpfungen möglich ist. Ein Vollbildmodus wird vom Programm angeboten und paßt sich bei Verwendung skalierbarer Schriften an beliebige Bildschirmgrößen ohne Benutzereingriffe in guter Qualität an. Das Umblättern kann im Vollbildmodus mit einem augenblicklichen

Wechsel erfolgen. Es werden jedoch auch verschiedene andere Übergänge angeboten, die das Bild z. B. streifenweise ersetzen. Über “hot spots” kann im Dokument navigiert werden, ggf. auch dynamisch zurück, wenn mehrere Links auf eine Seite verweisen. Jedes Dokument enthält typischerweise auch die darin verwendeten Schriften und die eingebundenen Bilder, so daß es reicht, eine Datei zu transportieren.

### 2.3.2 Erzeugung des pdf-Dateiformats

Wer in seinen Dokumenten ein Vielzahl von Formeln benötigt und diese auf verschiedenen Rechnerplattformen bearbeiten will, kommt um die Verwendung von  $\text{T}_{\text{E}}\text{X}$  bzw.  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  [Lam95] nicht herum. Für die Umsetzung eines mit  $\text{T}_{\text{E}}\text{X}$  formatierten Textes kann man entweder auf einen Konverter vom  $\text{dvi}$ -Format nach  $\text{pdf}$  zurückgreifen, zuerst aus der  $\text{dvi}$ -Datei eine PostScript-Datei erzeugen und diese mit dem (kostenpflichtigen) Distiller bearbeiten oder den nachfolgend beschriebenen direkten Weg beschreiten.

In neueren Versionen der Unix- $\text{T}_{\text{E}}\text{X}$ -Distribution  $\text{t}_{\text{E}}\text{X}$  ist mit  $\text{pdftex}$  eine Variante von  $\text{T}_{\text{E}}\text{X}$  verfügbar, mit der ohne weitere Zwischenschritte eine präsentationsfähige Datei erzeugt werden kann. Graphiken können dabei jedoch nur im JPEG-[Wal91], MetaPost-[Hob92] oder  $\text{pdf}$ -Format eingebunden werden. Falls Bilder mit dem Programm  $\text{xfig}$  [SS] erzeugt wurden, können diese durch ein entsprechendes Backend auch in das MetaPost-Format umgewandelt werden. Wenn Präsentationen neu erstellt werden und keine größeren Mengen an Bildern im PostScript-Format eingebunden werden müssen, ist dies der einfachste Weg. Darauf setzt daher auch das in diesem Bericht beschriebene Werkzeug PPower4 auf.

## 3 Aufgaben von PPower4

Der Textformatierer ist mit seinen Vorgaben dafür entworfen, gedruckte Dokumente zu gestalten. Die Druckausgaben sind jedoch statisch und für die Verwendung im Rahmen eines Vortrags nur bedingt geeignet. So wünscht man sich an manchen Stellen, die einzelnen Punkte einer Liste schrittweise aufdecken zu können. In vielen Fällen ist es zwar hilfreich, wenn die Zuhörer dem Gedankengang des Vortragenden etwas voraussehen können, indem sie die vollständige Folie bereits lesen und die Gesamtperspektive gleich erkennen. Aber manchmal möchte man auch eine unerwartete Entwicklung zeigen oder einen Überraschungseffekt ausnutzen. Dies sollte zur Erhaltung des Gesamteindrucks nicht auf einer neuen Folie stehen, aber auch nicht im Voraus sichtbar werden.

Bei der Präsentation mit Acrobat Reader kann man durch den augenblicklichen Wechsel der Seite den Eindruck erwecken, daß man ein und dieselbe Seite dynamisch aufbaut. Der Inhalt der Seite wird also auf mehreren Seiten nacheinander schrittweise zusammengesetzt. Man muß nur vermeiden, daß jede der neuen Zwischenseiten eine neue Seitennummer bekommt, wenn man diese zur generellen Orientierung auf den Folien angebracht hat.

### 3.1 Implementierung allein über den Textformatierer

Für eine erste Lösung zu diesem Problem kann man auf der Ebene des Textformatierers  $\text{T}_{\text{E}}\text{X}$  ansetzen. Wenn man einen Zwischenschritt für eine Präsentationsseite erreicht hat, dann kann der aktuelle Seiteninhalt mit Leerraum aufgefüllt und ausgegeben werden. Außerdem wird der Inhalt ohne den letzten Leerraum auf die nächste Seite übernommen ohne den Seitenzähler zu erhöhen.

Fast genauso einfach funktioniert letztendlich auch die Implementierung dieser Strategie. Die Ausgabeaktion kann unabhängig von der aktuell verwendeten Seitenaufbereitung angestoßen werden. Dies muß natürlich im Text vermerkt werden durch Aufruf eines Makros, das den den Namen `\pause` bekommen hat. Seine Implementierung ist in Abb. 1 angegeben.

Dieser Makroaufruf kann bei Aufzählungen problemlos eingestreut werden, wie das nachfolgende Beispiel zeigt.

```
\begin{enumerate}
\item hier erklären wir den ersten Fall,\pause
\item hier den zweiten\pause
\item und hier den letzten.
\end{enumerate}
```

Das Ergebnis demonstrieren wir hier am Inhalt von drei Minipages, die bei der Vorführung den Eindruck einer dynamisch aufgebauten Seite erwecken.

1. hier erklären wir den ersten Fall,	1. hier erklären wir den ersten Fall,  2. hier den zweiten	1. hier erklären wir den ersten Fall,  2. hier den zweiten  3. und hier den letzten.
--	---	--

Diese Strategie hat aber einige Nachteile:

1. Wenn Aufzählungen die Höhe einer Folie vollständig füllen, versucht  $\text{T}_{\text{E}}\text{X}$ , bei Erreichen des Randes im Rahmen der Formatierungsparameter noch Zeilenzwischenraum zu stauchen, damit das Gesamtergebnis besser wird. Dies führt jedoch dazu, daß beim Erscheinen des letzten Punkts der Aufzählungen die darüberliegenden Punkte etwas nach oben springen. Man könnte diesen unschönen Effekt durch explizites Verbiegen der Dehn- und Stauchbarkeit der Zeilenzwischenräume vermeiden.
2. Man kann nicht innerhalb einer Zeile anhalten. Der Auffüllmechanismus beendet die Zeile und setzt vertikal Leerraum ein. Das ist besonders ärgerlich, weil bei der Projektion durch die eingeschränkte Auflösung (in der Regel zur Zeit maximal XGA, also  $1024 \times 768$  Pixel) sowieso relativ wenig Platz auf jeder Folie zur Verfügung steht. Wenn man zu häufig eine neue Zeile beginnen muß, wird das Ergebnis unruhig und auch dadurch unleserlich.
3. Nicht einmal die Entwicklung einer Rechnung, bei der die Formelzeilen ausgerichtet sein sollen, kann zeilenweise angezeigt werden.

```

%% texpause.sty
%%-----
%% This is a quick hack to enable repeated pages with incremental
%% contents e.g. for displaying slides uncovering step by step.
%%
%% It depends on pdftex to be activated, i.e. for normal (La)TeX we
%% would ignore the command.
%%
%% The initial version (numbered 1.0) was written 07 May 99.
%% Version 1.1 was created 25 May 99 and fixed a naming problem.
%%
%% Possible extensions: make the page numbering optional (maybe using
%% subnumbering)
%% Make action optional (for printed versions via pdf).
%%
%% Plan of attack (should work with TeX and LaTeX):
%% - Get some resources, i.e. one counter, one token register and one
%%   box.
%% - When activated save the current page count in the counter and the
%%   output routine in the token register.
%% - Setup a new output routine, which saves away a copy of the
%%   current page.
%% - Trigger this output routine to save the cumulated page contents.
%% - Restore the former output routine and run it with the restored
%%   saved contents.
%% - Reset the page count and reinsert the contents once again,
%%   removing the last glue item on the page.
%%
\ifx\pdfoutput\undefined
\let\pause=\relax\expandafter\endinput
\fi
\newbox\p@uses@vebox
\newtoks\p@uses@veoutput
\newcount\p@uses@vepage
\def\pause{\global\p@uses@vepage=\count0\relax %save pagenumber
\p@uses@veoutput=\output % make backup copy of output routine
\output={\global\setbox\p@uses@vebox=\box255}% copy current contents
% only, when triggered
\vfill\eject %trigger now
\output=\p@uses@veoutput % restore output routine
\unvcopy\p@uses@vebox % insert contents
\eject % now really show the output
\global\count0=\p@uses@vepage\relax %restore page number
\unvbox\p@uses@vebox\vskip-\lastskip % and insert again for next turn
}%

```

Abbildung 1: Implementierung des inkrementellen Seitenaufbaus mit den Mitteln von  $\TeX$

4. Weitere Strukturen, in denen eine Unterbrechung der Präsentation auf diese Weise nicht möglich ist, ließen sich anführen.

Grundsätzlich bleibt festzuhalten, daß dieser Ansatz nur für eine sehr beschränkte Interaktion tauglich ist. Er hat aber den Vorteil, daß er eigenständig genutzt werden kann, daß er mit jedem Makropaket kooperieren kann, daß er ohne weitere Nachbearbeitung auskommt und daß er nicht auf die Benutzung von `pdf` festgelegt ist.

## 3.2 Nachbearbeitung

Eine Unterbrechung an beliebigen Stellen läßt sich dagegen nur erreichen, wenn man die Ausgabe von `TEX` nachbearbeitet. Wenn man das geräteunabhängige Format `dvi` erzeugt, muß eine solche Nachbearbeitung für das gewählte Ausgabegerät sowieso erfolgen. Im `dvi`-Format sind besondere Befehle ("`\special`") für die Kommunikation des Formatierers mit dem Nachbearbeitungsprogramm vorgesehen, die entsprechende Anweisungen und Parameter aufnehmen können. Bei der Verarbeitung mit `pdftex` entfällt aber dieser Schritt, weil von `TEX` direkt das Format `pdf` erzeugt wird, das keine solchen Erweiterungen kennt. Allerdings erlaubt die Definition von `pdf` auch die Angabe von Kommentaren, die von den Anzeigeprogrammen ignoriert werden. Diese Kommentare kann man nutzen, um Anweisungen an einen Postprozessor zu transportieren, der die Aufteilung jeder schrittweise aufzurollenden Seite in mehrere Seiten übernimmt.

Für den Einsatz des Postprozessors sind folgende Anforderungen zu formulieren:

1. Es soll vermieden werden, daß man in der Entwicklungsphase jedesmal den Postprozessor bemühen muß, um zu sehen, wo die Unterbrechungen eingestreut sind.
2. Die Ausgabe einer Fußzeile (mit Seitennummer o.ä.) sollte auch schon auf der ersten Teilseite erfolgen. Die Verwendung von Fußnoten auf Folien ist wegen der Schriftgrößen und der bereits angesprochenen Platzprobleme zu vermeiden.
3. Hyperlink-Verweise auf Seiten sollen auf die vollständige Seite gehen.
4. "hot spots" müssen auch von den Teilseiten aus funktionieren.

Die Anforderungen können wie folgt erfüllt werden:

1. Man kann bei der Erstellung der Seite eine Markierung zur Visualisierung der Unterbrechung anbringen, ohne daß dadurch das Layout beeinflußt wird. Diese Markierung kann vom Postprozessor wieder entfernt werden und erlaubt ohne Nachbearbeitung, den richtigen Präsentationsablauf zu beurteilen.
2. Bei der Formatierung muß die Fußzeile zusammen mit der Kopfzeile ausgegeben werden, bevor mit der Ausgabe des Texttrumpfes begonnen wird. Dies erfordert allerdings einen Eingriff in die verwendeten Makropakete. Es wäre wünschenswert, wenn diese Reihenfolge der Ausgabe grundsätzlich benutzt würde oder per Option abrufbar wäre.

3. Bei der Bearbeitung der Datei müssen alle vollständigen Seiten ihren Platz behalten bzw. die Verweise darauf müssen aktualisiert werden. Das kann in der Struktur von `pdf` durch Beibehalten der vorgegebenen Seitenstruktur erreicht werden. Die neuen Seiten werden in dieser Struktur hinzugefügt.
4. Die Aktivierung der “hot spots” auf allen Seiten kann man erreichen, indem man alle von der vollständigen Seite auf die Teilseiten übernimmt. Es ist jedoch Vorsicht geboten, wenn diese aktiven Stellen durch Umrahmungen gekennzeichnet werden. Dies ist jedoch bei einer Präsentation nur bedingt hilfreich, weil diese Kennzeichnung die Aufmerksamkeit der Zuhörer unnötig auf sich zieht. Die Markierung ist natürlich dann sinnvoll, wenn ein fremder Leser durch das Dokument navigiert. Aber wenn man als Autor selbst präsentiert, braucht man diese Hilfestellung in der Regel nicht und kann die “hot spots” unmarkiert/unsichtbar lassen. Da der Acrobat Reader den Typ des Mauszeigers über einem “hot spot” ändert, hat man dennoch eine gute Chance, die Stelle bei der Präsentation richtig zu treffen.

### 3.3 Implementierung

Für die Verarbeitung der `pdf`-Dateien wird ein Programm benötigt, das eine `pdf`-Datei einlesen, umbauen und wieder ausgeben kann. Für die Bearbeitung von Dateien im `pdf`-Format sind seit 1999 einige Bibliotheken verfügbar, die jedoch häufig nur Teile dieses Anforderungsspektrums abdecken (z.B. nur lesende oder nur schreibende Bearbeitung) oder nicht als “Open Source” vorliegen. Eine Ausnahme bildet die `Java`-Bibliothek `pj` von Etymon Systems, Inc [Nas98]. Diese wurde daher für die Erstellung eines Prototyps von PPower4 benutzt. Die Verwendung von `Java` hat den Vorteil, daß diese Software auf einem breiten Spektrum von Maschinen ohne besonderen Portierungsaufwand eingesetzt werden kann. Wenn man berücksichtigt, daß der Verarbeitungsschritt nur selten ausgeführt werden muß, sind Bedenken wegen schlechten Durchsatzes bei Wahl dieser Sprache eher zurückzustellen.

Auf der `TeX`-Ebene wird bei Benutzung der Markierung `\pause` ein kleiner farbiger Block (ohne Breite und Höhe für die Formatierung) eingesetzt. In der `pdf`-Datei wird dieser Block von einem eindeutigen Kommentar eingeschlossen, der durch den Postprozessor mitsamt dem Kommentar entfernt wird. Bei einer Verarbeitung ohne Erzeugung von `pdf` ist dieser Makroaufruf wirkungslos.

Wir ersparen uns hier, das gesamte Programm für die Umsetzung anzugeben. Das würde den Rahmen dieses Berichts sprengen. Folgende Erläuterungen sollten ausreichen:

- Der Postprozessor besteht aus
  - einem Steuerprogramm, das die Verarbeitung der Kommandozeilenparameter übernimmt und die übrigen Komponenten aktiviert,
  - einer Verarbeitungsmaschinerie, die die Eingabedatei durchläuft und nach den zu bearbeitenden Kommentaren durchsucht, Methoden zum Verändern des Seiteninhalts und zum Erzeugen neuer Seiten anbietet, und



- den spezifischen Kommentarbearbeitungen, die die spezifischen Aktionen abwickeln.
- Die Aktionen auf der pdf-Datei sind in der Verarbeitungsmaschinerie gekapselt, so daß sich bei Verwendung einer anderen Bibliothek für das Parsen und Erzeugen der pdf-Datei in den übrigen Komponenten keine Änderungen ergeben.
- Es ist leicht möglich, weitere Kommentarbearbeitungen nachzurüsten und in das Gesamtsystem einzuklinken. Beispiele dafür finden sich unten.
- Das Gesamtsystem (ohne die Bibliothek zum Parsen von pdf) umfaßt etwa 1000 Zeilen Java-Code.

Die Makros, die benötigten Bibliotheken und das Programm findet man unter

<http://www-sp.iti.informatik.tu-darmstadt.de/software/ppower4/>

## 4 Erweiterungen

Für eine ausgewachsene Präsentation benötigt man einige weitere Funktionen, die nur zum Teil realisierbar sind.

### 4.1 Ein- und Ausblenden

Es wäre wünschenswert, ganze Textabschnitte ein- und ausblenden zu können. Bei einer textuellen Erfassung des Dokuments ist dazu jedoch zunächst eine geeignete Benutzerschnittstelle zu definieren. Mit L<sup>A</sup>T<sub>E</sub>X bietet sich an, hierzu Minipages zu benutzen. Dies ist jedoch relativ aufwendig, weil pdf<sub>t</sub>ex zur Zeit seine Positionierungen in der pdf-Datei als relative Angabe spezifiziert. Damit ist die Reihenfolge der Ausgaben festgelegt und kann nur durch mühsames Rückrechnen der aktuellen Position verändert werden.

### 4.2 Animationen

Animierte Effekte sind wegen der fehlenden Unterstützung durch Acrobat Reader nicht realisierbar. Diese müssen bei Bedarf über externe Programm nachgerüstet werden. Wie weit durch die Implementierung von JavaScript in Acrobat Reader vielleicht doch Lösungen greifbar werden, wurde bisher nicht untersucht.

### 4.3 Hintergründe

Zur Unterstützung der Gesamtwirkung einer Präsentation kann auch Farbe in stärkerem Maße herangezogen werden, als dies in gedruckten Werken der Fall ist, für deren Erstellung T<sub>E</sub>X normalerweise benutzt wird. Dabei ist es nützlich, auch farbige Hintergründe definieren zu können. Als die Überlegungen in diese Richtung begonnen wurden, war eine entsprechende Unterstützung auf der Ebene von T<sub>E</sub>X im Zusammenspiel mit der Ausgabe von pdf nicht gegeben. Für monochrome Hintergründe gibt es inzwischen aus den Internet-Diskussionslisten

einen Implementierungsvorschlag, der für die meisten Anwendungen eine ausreichende Lösung ohne Nachbearbeitung ermöglicht. Bei der Bearbeitung durch den Postprozessor ist es jedoch einfach, auch diese Aufgabe zu erledigen.

Man muß sich bewußt sein, daß man ohne den Einsatz des Postprozessors zum Einsetzen des Hintergrunds nicht mehr viel erkennen kann, wenn man eine dunkle Hintergrundfarbe wählt und darauf mit heller Schrift schreibt.

## 5 Benutzeranleitung

In diesem Abschnitt soll die Installation und die Benutzung des Postprozessors kurz erläutert werden.

### 5.1 Installation des Postprozessors

Neben der Implementierung des Postprozessors benötigen man zum Betrieb die Java-Bibliotheken `pj` (getestet mit Version 0.22) und `gnu.getopt`. Beides unterliegt der GPL und ist frei verfügbar. Diese Bibliotheken können mit dem Rest des Systems in einem Java-Archiv zusammengestellt werden. Für die Benutzung unter Unix empfiehlt sich die Installation eines kleinen Skripts, das für die Routinen den Aufruffad setzt.

Dies kann folgende Form haben:

```
#!/bin/sh
CLASSPATH=/common/Java/lib/ppower4/pp4.jar \
    java de.tu_darmstadt.de.sp.pp4.PPower4 "$@"
```

### 5.2 Einbau im Dokument

Für die Benutzung des Makros `\pause` muß die entsprechende Definitionsdatei `pause.sty` eingelesen werden. Diese ist in dem von  $\LaTeX$  durchsuchten Bereich bereitzustellen. Für die Hintergrunddefinitionen sind zusätzliche Angaben aus `background.sty` erforderlich.

Folgende Hintergrunddefinitionen sind spezifizierbar:

`\pagecolor{farbe}` Einfarbiger Hintergrund in der angegebenen Farbe.

`\hpagecolor[farbe1]{farbe2}` Hintergrund horizontal wechselnd von `farbe1` nach `farbe2` bzw., wenn der optionale Parameter fehlt, Hintergrund von der angegebenen `farbe2` ausgehend horizontal heller werdend.

`\vpagecolor[farbe1]{farbe2}` Hintergrund vertikal wechselnd von `farbe1` nach `farbe2` bzw., wenn der optionale Parameter fehlt, Hintergrund von der angegebenen `farbe2` ausgehend vertikal heller werdend.

Die Farbangaben müssen auf der  $\LaTeX$ -Ebene definiert sein. Dabei kann man auf die in `color.sty` vordefinierte Farben zurückgreifen oder mit `\definecolor` eigene Farben definieren. Wenn ein Farbverlauf gewünscht wird, müssen beide Farbwerte in demselben Farbmodell angegeben werden (rgb oder cmyk). Graustufen werden nicht unterstützt.

Bei Seiten, die durch `\pause` schrittweise aufgebaut werden, muß die Hintergrundangabe vor der ersten Unterbrechung erfolgen, ansonsten wird sie ignoriert. Hintergrundangaben gelten ab der Seite, auf der sie gemacht werden, bis zu einer Neudefinition.

### 5.3 Aufruf des Postprozessors

Nach der Verarbeitung des Dokuments mit `pdftex/pdflatex` muß der Postprozessor aufgerufen werden. Beim Aufruf sind mindestens die Namen einer Eingabe- und einer Ausgabedatei anzugeben. Darüberhinaus kann das Verhalten mit den folgenden Optionen gesteuert werden:

- `-v` Erhöhen der Testausgabestufe (kann mehrfach angegeben werden). Diese Option wird in der Regel zum Test von neuen Erweiterungen nützlich sein.
- `--verbose=n` Setzen der Testausgabestufe auf den Wert  $n$ .
- `-n` Verhindern der Komprimierung der Ausgabedatei (für Test- oder Demonstrationszwecke).
- `--nocompress` wie `-n`
- `-h` Ausgabe der Parameterinformationen
- `--help` wie `-h`
- `-?` wie `-h`

### 5.4 Beispiele

Zunächst soll ein minimales Beispiel unter Verwendung der Klasse `foils` den Aufbau einer Seite mit einer schrittweise zu entwickelnden Formel zeigen. Aus Platzgründen beschränken wir uns hier auf wenige Schritte.

```
\documentclass[30pt,landscape]{foils}
\usepackage[pdftex]{geometry}
\geometry{headsep=3ex,hscale=0.9}
\usepackage[pdftex]{color} % for the colored block
\usepackage{pause}
\rightfooter{} % no more page numbers bottom right
\MyLogo{} % no logo bottom left
\begin{document}
\foilhead{Eine schrittweise aufgebaute Formel}
\[
Q(n) = \sum_{i=1}^n i^2 =
\frac{1}{3}n(n+\frac{1}{2})(n+1)\pause
\]
\begin{eqnarray*}
\left|\frac{1}{3}n^3 + \frac{1}{2}n^2+\frac{1}{6}n\right|
&\leq&\left|\frac{1}{3}n^3\right|+\left|\frac{1}{2}n^2\right|+
&\quad\left|\frac{1}{6}n\right|\pause\\
&\leq&\frac{1}{3}|n^3|+\frac{1}{2}|n^2|+\frac{1}{6}|n^3|\backslash\backslash
\end{eqnarray*}
```

<p><b>Eine schrittweise aufgebaute Formel</b></p> $Q(n) = \sum_{i=1}^n i^2 = \frac{1}{3}n(n + \frac{1}{2})(n + 1)$ $\left  \frac{1}{3}n^3 + \frac{1}{2}n^2 + \frac{1}{6}n \right  \leq \left  \frac{1}{3}n^3 \right  + \left  \frac{1}{2}n^2 \right  + \left  \frac{1}{6}n \right $ $\leq \frac{1}{3} n^3  + \frac{1}{2} n^2  + \frac{1}{6} n^3 $ $=  n^3 $	<p><b>Eine schrittweise aufgebaute Formel</b></p> $Q(n) = \sum_{i=1}^n i^2 = \frac{1}{3}n(n + \frac{1}{2})(n + 1)$
<p><b>Eine schrittweise aufgebaute Formel</b></p> $Q(n) = \sum_{i=1}^n i^2 = \frac{1}{3}n(n + \frac{1}{2})(n + 1)$ $\left  \frac{1}{3}n^3 + \frac{1}{2}n^2 + \frac{1}{6}n \right  \leq \left  \frac{1}{3}n^3 \right  + \left  \frac{1}{2}n^2 \right  + \left  \frac{1}{6}n \right $	<p><b>Eine schrittweise aufgebaute Formel</b></p> $Q(n) = \sum_{i=1}^n i^2 = \frac{1}{3}n(n + \frac{1}{2})(n + 1)$ $\left  \frac{1}{3}n^3 + \frac{1}{2}n^2 + \frac{1}{6}n \right  \leq \left  \frac{1}{3}n^3 \right  + \left  \frac{1}{2}n^2 \right  + \left  \frac{1}{6}n \right $ $\leq \frac{1}{3} n^3  + \frac{1}{2} n^2  + \frac{1}{6} n^3 $ $=  n^3 $

Abbildung 2: Zwischenschritt und Ergebnis des ersten Beispieldokuments

```

&=&|n^{3}|
\end{eqnarray*}
\end{document}

```

Als Ausgabe von `pdflatex` erscheint die Seite vollständig in einem Dokument, das aus einer Seite besteht. Durch den Postprozessor wird dann daraus ein Dokument mit drei Seiten. Die Ergebnisse sind in Abbildung 2 verkleinert dargestellt.

An einem ausführlicheren Beispiel sollen die Details etwas genauer betrachtet werden:

```

1 \documentclass[30pt,landscape]{foils}
2 \usepackage[english,german]{babel} % language support
3 \usepackage[latin1]{inputenc} % allow Latin1 characters
4 \usepackage[pdftex]{color}
5 \usepackage[pdftex]{geometry}
6 \geometry{headsep=3ex,hscale=0.9}
7 \usepackage{hyperref}
8 \hypersetup{pdftitle={PPower4Beispiel},
9 pdfsubject={Ein Beispiel zur Anwendung von PPower4},
10 pdfauthor={Klaus Guntermann, FG Systemprogrammierung,
11 TU Darmstadt <guntermann@iti.informatik.tu-darmstadt.de>},
12 pdfkeywords={pdftex, acrobat},
13 pdfpagemode={FullScreen}
14 }

```

```

15 \usepackage{pause}
16 \usepackage{background}
17 \usepackage{pdfslide}
18 \begin{document}
19 \definecolor{bgblue}{rgb}{0.04,0.39,0.53}
20 \vpagecolor{bgblue}
21
22 \foilhead{Ein Beispiel mit verlaufendem Hintergrund}
23 \begin{itemize}
24 \item Aufzählungen\pause
25   \begin{itemize}
26   \item werden geschachtelt\pause
27   \item und mit Symbolen markiert
28     \hypertarget{Anfang}{} %Markierung irgendwo auf der Seite
29     \begin{itemize}
30     \item auch in dieser Tiefe\pause
31     \end{itemize}
32   \item auf verschiedenen Ebenen\pause
33   \end{itemize}
34 \item mit Formeln wie  $\sum_{i=0}^{\infty} a_i \cdot x^i$ 
35
36 \end{itemize}
37
38 \foilhead{Andere Seitenübergänge}
39 \hypersetup{pdfpagetransition=Dissolve}
40 \begin{itemize}
41 \item Anderen Seitenübergang einschalten\pause
42 \item Aber auch explizit wieder ausschalten
43 \end{itemize}
44
45 \foilhead[2ex]{Das war's schon}
46 \hypersetup{pdfpagetransition=R}
47 \begin{center}
48 Das soll als Beispiel reichen.
49
50 Auf Wunsch geht es \hyperlink{Anfang}{zurück} zum Anfang.
51 \end{center}
52 \end{document}

```

Das Dokument verwendet besonders große Schrift und eine den Beamerdimensionen angepaßte Orientierung, die mittels des Pakets `geometry` konfiguriert wird (Zeilen 5 und 6). Für den Einsatz von Farben wird `color` verwendet (Zeilen 4 und 19). Für die Aufbereitung von Zusatzinformationen, des Seitenübergangs (spezifisch für Acrobat Reader) und für das Navigieren in der Datei wird `hyperref` eingesetzt (Zeilen 7, 8, 28, 39, 46 und 50). Für die Details zu diesen Paketen sei auf deren mitgelieferte Dokumentation verwiesen. Die für PPower4 benötigten Pakete `pause` und `background` werden ebenfalls eingelesen (Zeilen 15 und 16). In `pdfslide` werden besondere Voreinstellungen und Konfigurationen der anderen Pakete vorgenommen. Daher sollte dieses als letztes geladen werden (Zeile 17). Im Detail werden in dieser Datei die Vorgaben von

foils so umgesetzt, daß keine Fußzeile mehr entsteht und es werden die Farben für die verschiedenen Anteile definiert, was folgendermaßen erfolgt:

```
1 %% pdfslide.sty 30 Aug 99
2 %%-----
3 %% Adapt foiltex to be used to prepare slides in pdf format
4 %% using backgrounds and partial builds.
5 %%
6 %% Disable some pdf commands, when not used in pdf(la)tex.
7 %%
8 %% Special (PDF) effects:
9 %% - make background blue, write normally in white and
10 %% headings in yellow.
11 %% - use colorful labels in itemized lists.
12 %% - avoid footline usage, because that would interfere with
13 %% partial builds of a page. Move the page number to the
14 %% upper right corner. This means also suppression of Logo
15 %% etc.
16 %% - make sure that links are displayed in text color, not
17 %% with frames.
18 %% - Do not use paragraph indentation and justified text on
19 %% slides (shouldn't that be the default?)
20 \rightfooter{} % no more page numbers bottom right
21 \MyLogo{} % no logo bottom left
22 \rightheadert{\rlap{\quad\textsf{\tiny\thepage}}}% % page number
23 \parindent Opt % do not indent paragraphs
24 \rightskip Opt plus 1fil % allow ragged right
25 %% select colors
26 \RequirePackage{color}
27 %% for the frames/page numbers etc.
28 \renewcommand\Black{\color{white}}
29 %% for the headline etc.
30 \renewcommand\normalcolor{\color{yellow}}
31 %% for the background
32 \pagecolor{blue}
33 %% for the text
34 \color{white}
35 %% choose some colored item labels
36 \renewcommand{\labelitemi}{\textcolor{red}{$\bullet$}}
37 \renewcommand{\labelitemii}{\textcolor{yellow}{$\star$}}
38 \renewcommand{\labelitemiii}{\textcolor{magenta}{$\ast$}}
39 \renewcommand{\labelitemiv}{\textcolor{cyan}{$\circ$}}
40 % Make hyperlinks colored, not framed, if hypersetup is used
41 \ifx\hypersetup\undefined\relax\else
42 \hypersetup{colorlinks=true}
43 \fi
```

Im eigentlichen Rumpf des Dokuments werden dann die Seiten der Reihe nach beschrieben. An den Stellen, wo eine Unterbrechung erfolgen soll, ist `\pause` einzusetzen (siehe Zeilen 24, 26, 30, 32 und 41 in der Beispieldatei).

Das Ergebnis hat dann die Form, die in Abbildung 3 verkleinert in den einzelnen Anzeigeschritten wiedergegeben ist.

## 6 Zusammenfassung

Mit dem vorliegenden Prozessor ist es möglich, auch wissenschaftliche Texte mit einem hohen Anteil an Formeln mit angemessenem Aufwand ansprechend zu präsentieren. Weitere Entwicklungen könnten sich anschließen, die es erlauben, die Dynamisierung der Präsentation noch zu verstärken. Allerdings muß man sich dabei etwas zurückhalten und sollte die angebotenen Effekte nur in mäßigem Umfang einsetzen, damit das Auditorium nicht von der Beschäftigung mit dem Inhalt abgelenkt wird.

## Literatur

- [Ado99] Adobe Systems Incorporated. *Portable Document Format Reference Manual*. Addison-Wesley, Reading, MA, USA, version 1.3 (march 11, 1999) edition, 1999.
- [Hob92] J. D. Hobby. A user's manual for MetaPost. Technical Report 162, AT&T Bell Laboratories, 1992.
- [Knu86] Donald E. Knuth. *The T<sub>E</sub>Xbook*, volume A of *Computers and Typesetting*. Addison-Wesley, Reading, MA, USA, 1986.
- [Koh94] Eva Kohlberg. *Microsoft powerpoint für Windows*. Microsoft Press Deutschland, Unterschleissheim, 1994.
- [Lam95] Leslie Lamport. *Das L<sup>A</sup>T<sub>E</sub>X Handbuch*. Addison-Wesley, Reading, MA, USA, 1995.
- [Nas98] Nassib Nassar. Automating pdf objects for interactive publishing. *Web Techniques Magazine*, 3(10), October 1998.
- [SS] Brian W. Smith et al. Supoj Sutanthavibul. *Xfig user manual*. online at <http://www.xfig.org>.
- [Wal91] Gregory Wallace. The JPEG still picture compression standard. *Comm. ACM*, 34(4):30–44, 1991.

<p>1</p> <p><b>Ein Beispiel mit verlaufendem Hintergrund</b></p> <ul style="list-style-type: none"> <li>• Aufzählungen</li> </ul>	<p>1</p> <p><b>Ein Beispiel mit verlaufendem Hintergrund</b></p> <ul style="list-style-type: none"> <li>• Aufzählungen <ul style="list-style-type: none"> <li>* werden geschachtelt</li> </ul> </li> </ul>
<p>1</p> <p><b>Ein Beispiel mit verlaufendem Hintergrund</b></p> <ul style="list-style-type: none"> <li>• Aufzählungen <ul style="list-style-type: none"> <li>* werden geschachtelt</li> <li>* und mit Symbolen markiert <ul style="list-style-type: none"> <li>• auch in dieser Tiefe</li> </ul> </li> </ul> </li> </ul>	<p>1</p> <p><b>Ein Beispiel mit verlaufendem Hintergrund</b></p> <ul style="list-style-type: none"> <li>• Aufzählungen <ul style="list-style-type: none"> <li>* werden geschachtelt</li> <li>* und mit Symbolen markiert <ul style="list-style-type: none"> <li>• auch in dieser Tiefe</li> </ul> </li> <li>* auf verschiedenen Ebenen</li> </ul> </li> </ul>
<p>1</p> <p><b>Ein Beispiel mit verlaufendem Hintergrund</b></p> <ul style="list-style-type: none"> <li>• Aufzählungen <ul style="list-style-type: none"> <li>* werden geschachtelt</li> <li>* und mit Symbolen markiert <ul style="list-style-type: none"> <li>• auch in dieser Tiefe</li> </ul> </li> <li>* auf verschiedenen Ebenen</li> </ul> </li> <li>• mit Formeln wie <math>\sum_{i=0}^{\infty} a_i \cdot x^i</math></li> </ul>	<p>2</p> <p><b>Andere Seitenübergänge</b></p> <ul style="list-style-type: none"> <li>• Anderen Seitenübergang einschalten</li> </ul>
<p>2</p> <p><b>Andere Seitenübergänge</b></p> <ul style="list-style-type: none"> <li>• Anderen Seitenübergang einschalten</li> <li>• Aber auch explizit wieder ausschalten</li> </ul>	<p>3</p> <p><b>Das war's schon</b></p> <p>Das soll als Beispiel reichen. Auf Wunsch geht es <b>zurück</b> zum Anfang.</p>

Abbildung 3: Inhalt des Beispieldokuments