

LaTeX im Studium

**Noch viel fortgeschrittenere Präsentationen
mit LaTeX**

Jörn Clausen

joern@TechFak.Uni-Bielefeld.DE

Übersicht

- portable Grafikformate
- Grafiken und PPower4
- Folien für Präsentation und Druck

Abbildungen mit xfig

- Zeichen-Programm, nicht Mal-Programm
- langjährige Entwicklung
- zahlreiche Ausgabeformate:
 - (Encapsulated) PostScript, PDF
 - PNG, GIF, JPEG, PCX
 - \LaTeX , epic, eepic, MetaFont, MetaPost
 - HPGL, AutoCAD, tpic, ...
- Kommandozeilen-Konverter `fig2dev`
- später: Kombination mit PPower4

Aufgaben

- Im Archiv `uebung4.tar.gz` befindet sich eine Folien-Präsentation, in der die Abbildung `drawing.fig` eingebunden werden soll. Eine EPS-Version der Abbildung erhält man mit

```
$ fig2dev -L eps drawing.fig drawing.eps
```

Erzeuge diese Datei und übersetze die Präsentation mit \LaTeX .

- Welche Möglichkeiten gibt es, die Abbildung durch $\text{pdf}\LaTeX$ einbinden zu lassen?

Die direkte Erzeugung von PDF ist im prinzip die naheliegendste. Leider ist das Bild dann im Vergleich zur EPS-Variante im 90° gedreht, d.h. die selbe Datei erzeugt beim Übersetzen mit \LaTeX und $\text{pdf}\LaTeX$ unterschiedliche Ergebnisse. PNG/JPEG sollten nicht verwendet werden, weil das Bild dann auf eine bestimmte Auflösung festgelegt ist. Der Umweg über epstopdf liefert das beste Ergebnis.

- Man kann mit `-L png` die Abbildung in eine PNG-Datei umwandeln und direkt einbinden. Entsprechend könnte man auch das Format JPEG verwenden.

```
$ fig2dev -L pdf drawing.fig drawing.pdf
```

- Man kann die `fig`-Datei mit
- Man kann die EPS-Datei mit Hilfe von epstopdf nach PDF konvertieren.

Man kann die Abbildung auf verschiedene Weisen einbinden:

MetaPost

- Grafik-Sprache, angelehnt an METAFONT
- erzeugt PostScript statt Bitmaps
- kann „von Hand“ programmiert werden
- oder: `fig2dev -L mp ...`
- Weiterverarbeitung:

```
$ mpost figure.mp
```
- erzeugt EPS-Datei `figure.0`
- enthält keine Font-Definitionen

Aufgaben

- Ändere die Regeln im Makefile ab, um
 - mit `fig2dev` eine MetaPost-Datei zu erzeugen
 - diese mit `mpost` nach PostScript zu übersetzen
 - der PostScript-Datei die Endung `.eps` zu geben

Übersetze anschließend erneut die \LaTeX -Datei. Was hat sich geändert?

- Starte `xfig` mit folgenden Optionen:

```
$ xfig -specialtext -latexfonts -startlatexFont default
```

und lade die Zeichnung. Füge die Formel

```
$x^2 + y^2 = z^2$
```

in das Bild ein (samt Dollar-Zeichen). Übersetze die \LaTeX -Datei erneut.

- Die Formel wird als \LaTeX -Formel gesetzt. Entsprechend funktionieren auch andere \LaTeX -Befehle, wenn man den Text als "special" markiert.

```
%.eps :  
fig2dev -L mp $> *.mp  
mpost $*.mp  
mv $*.0 $@
```

- EPS-Datei mit korrekter Dateierzeugung erzeugen:

Aufgaben

- Lösche die Datei `drawing.pdf` (falls vorhanden) und lege einen *symbolic link* an:

```
$ ln -s drawing.eps drawing.mps
```

Nun übersetze die \LaTeX -Datei mit `pdf \LaTeX` . Was passiert?

Die Datei `drawing.mps` (also eigentlich `drawing.eps`) wird von `pdf \LaTeX` eingebunden.

MetaPost und pdf \TeX

- pdf \TeX kann EPS nicht direkt einbinden
- mit MetaPost erzeugtes PostScript aber sehr einfach
- on-the-fly-Konversion von PostScript zu PDF
- Dateiendung: `.mps`

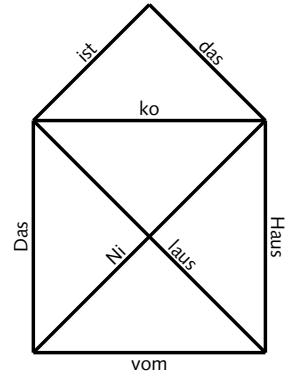
Grafik-Effekte

- schrittweise erscheinende Grafik
- xfig + MetaPost + PPower4
- Zeichnung über mehrere Ebenen verteilen
- mindestens eine „freie“ Schicht
- *Multi-MetaPost*
- MetaPost erzeugt pro Ebene ein Bild
- `mpmulti`-Paket:

```
\multiinclude{niko-anim}
```

- `graphicx`-Optionen:

```
\multiinclude[graphics={width=6cm}]{...}
```



Aufgaben

- Erzeuge mit xfig eine Grafik mit mehreren Ebenen. Speichere sie als bild-anim.fig ab.

- Erzeuge mit

```
$ fig2dev -L mmp bild-anim.fig bild-anim.mmp
```

die MetaPost-Datei. Diese wird wie gewohnt mit

```
$ mpost bild-anim.mmp
```

übersetzt. Wieviele Dateien erzeugt MetaPost?

- Füge in der Präambel der \LaTeX -Datei die Anweisung

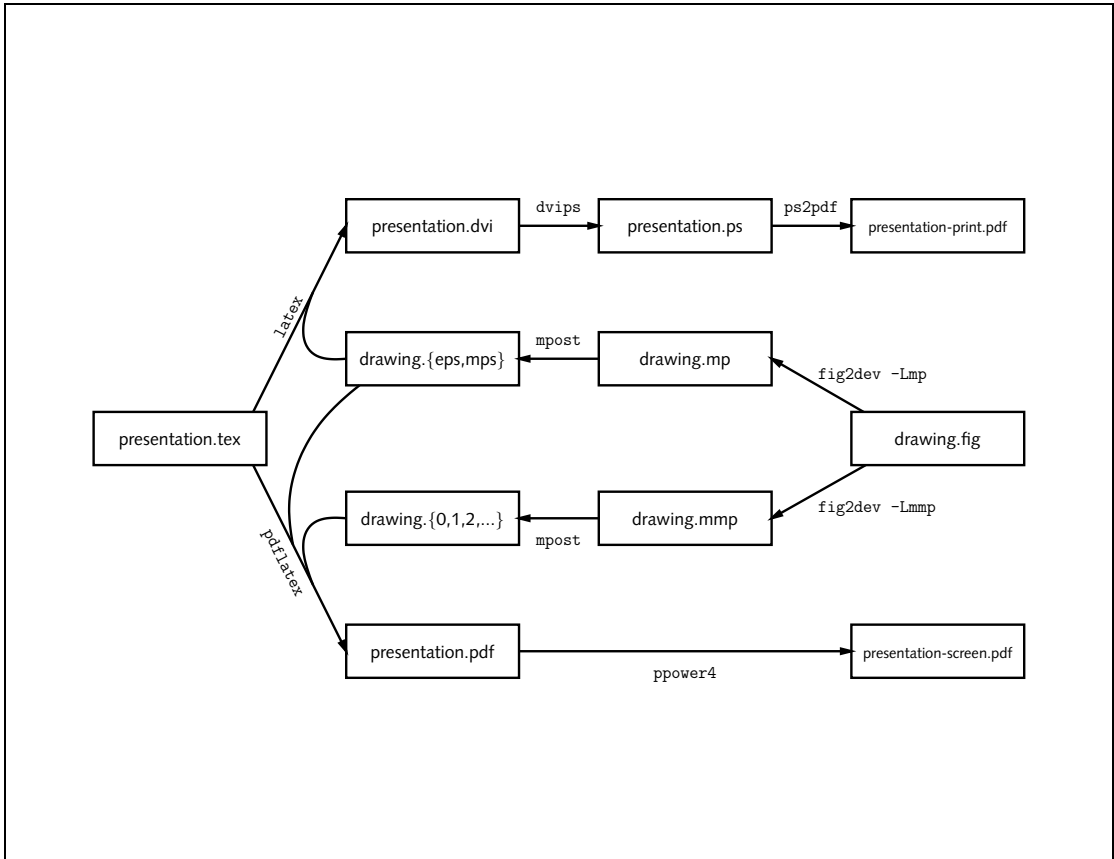
```
\DeclareGraphicsRule{*}{mps}{*}{}{}
```

ein. Binde die Zeichnung mit Hilfe des mpmulti-Pakets ein, übersetze mit pdf \LaTeX und bearbeite die PDF-Datei mit PPower4 nach.

Für jede Ebene (bzw. jede separate Folge von Ebenen) wird eine Datei erzeugt.

Aus eins mach zwei

- Ärgernis: Präsentation als .ppt-Datei verteilen
- Tonerverschwendung: helle Schrift auf dunklem Hintergrund
- besser: zwei getrennte Versionen
 - Präsentationsversion mit Farb- und Überblendeffekten
 - Druckversion ohne Effekt-Farben und mit kollabierten Folien
- beide Versionen aus einer Quelle
- Steuerung der Übersetzungen durch Makefile



entweder, oder ...

- Effekte nur bei pdf \LaTeX :

```
\usepackage{ifpdf,fixseminar,graphicx,color}
\ifpdf
  \usepackage{pause,mpmulti,background}
  \definecolor{bgcolor}{rgb}{1.00,0.97,0.86}
  \pagecolor{bgcolor}
\else
  \usepackage[ignore]{pause}
\fi
```

- Bilder:

```
\ifpdf
  \multiinclude[graphics={width=.5\textwidth}]{bild-anim}
\else
  \includegraphics[width=.5\textwidth]{bild-anim}
\fi
```

Aufgaben

- Entpacke das Archiv `example.tar.gz`. Es enthält eine einfache Präsentation mit zwei Abbildungen und nutzt PPower4. Das Makefile enthält alle nötigen Regeln, um die beiden Versionen der Folien zu erzeugen.

Folgende targets können verwendet werden:

<code>all</code>	erzeugt	<code>presentation-screen.pdf</code> und <code>presentation-print.pdf</code>
<code>quick</code>	erzeugt	<code>presentation-print.ps</code>
<code>clean</code>	löscht	temporäre Dateien
<code>realclean</code>	löscht	alles bis auf die beiden PDF-Dateien
<code>spotless</code>	löscht	alle erzeugten Dateien