***BENUTZER**SERVICE*

Kommunikation & Datenverarbeitung

18.05.93

Einbindung von PostScript-Graphik in LaTeX-Dokumente

G.Vinel

Inhalt

Literaturempfehlungen:

- dvips-Dokumentation an der `hera`:
  - *DVIPS: A TeX Driver*, Dokumentation von Tomas Rockicki, in der FVTK-Annahme erhältlich,
  - *man pages* (Aufruf: `man dvips`)
- LaTeX *Eine Einführung* von Helmut Kopka, Addison-Wesley Deutschland, 1992.
- *TeX at Cern* von Michel Goossens und Alexander Samarin, CERN Local Guide, 1992, CN/US/136.
- *PostScript Language Reference Manual*, Addison-Wesley, 1992.

1 LaTeX-Mittel zur Graphikeinbindung

1.1 figure-Umgebung

Die `figure`-Umgebung ist für gleitende Bilder vorgesehen, d.h. die Bildplazierung wird von LaTeX entsprechend der Bildgröße, dem vorausgehenden Text und den im Befehl angegebenen Positionierungsparametern vorgenommen. Letztere können mehrere Varianten in Prioritätsreihenfolge enthalten.

Umgebungssyntax:

`\begin{figure}[`*wohin*`]` *bild* `\end{figure}`

`\begin{figure*}[`*wohin*`]` *bild* `\end{figure*}`

Die *Form ist für zweispaltige Seitenformatierung vorgesehen. (`twocolumn`-Option im `documentstyle`-Befehl).

bild	-	steht hier für ein mit `picture`-Umgebung erzeugtes LaTeX-Bild, oder für ein fremdes Graphikfile, dessen Einbindung weiter unten erklärt wird.
wohin	-	Positionierungsparameter der gewünschten Bildplazierungen:

`h`	(*Here*)	:	an der Stelle im Text, wo `figur`-Umgebung aufgerufen wird, (für *Form nicht erlaubt)
`t`	(*Top*)	:	am Seitenbeginn,
`b`	(*Bottom*)	:	am Seitenende,
`p`	(*Page of floats*)	:	die gleitenden Bilder werden gemeinsam auf Bildseiten angeordnet

Beispiel für Positionierungsparameterkombination `wohin`:

`htp`, mit erster Priorität `h`.
`tbp` , entspricht der in LaTeX eingestellten Parameterkombination.

Die tatsächliche Positionierung entsprechend der Prioritätsfolge wird von LaTeX nach bestimmten logischen Regeln und eingestellten Stilparametern an der frühest möglichen Stelle festgelegt.

Für Bildbeschriftungen (weniger als 300 Zeichen) ist ein gesonderter Befehl erforderlich:

`\caption[`*kurzform*`]` {*beschriftungstext*},

der innerhalb der `figure`-Umgebung als normaler Absatztext entsprechend gewünschter Über- oder Unterschrift vor bzw. nach `bild` steht.

In deutschen Dokumenten erscheint die Beschriftung als :

`Abbildung Nr.`: *beschriftungstext*

mit automatischer Numerierung (`Nr.`) durch LaTeX .

Beschriftungstext kürzer als eine Zeilenlänge erscheint **zeilenzentriert**. Eine spezielle Beschriftungsanordnung kann über einen `\parbox`-Befehl (s. Beispiel im Abschnitt 4.1) erzeugt werden.

Der optionale Parameter *kurzform* ist als Text für ein mögliches Bildverzeichnis vorgesehen. In diesem Fall muss ein `\listoffigures`-Befehl im Vorspann stehen.

Mit den Befehlspaaren:

`\label`{*bezugstext*}

`\ref`{*bezugstext*}

oder

`\label`{*bezugstext*}

`\pageref`{*bezugstext*}

kann im LaTeX-Dokument auf eine Abbildung, die durch einen *bezugstext* markiert ist, Bezug genommen werden. Für die Markierung ist der \ `caption`-Befehl wie folgt zu schreiben:

`\caption[`*kurzform*`]` {`\label`{*bezugstext*} *beschriftungstext*}

Wie bereits erwähnt, existiert im LaTeX eine Reihe von Stilparametern zur Anordnung gleitender Objekte im Dokument, die vom Benutzer geändert werden können (siehe H. Kopka, LaTeX *Eine Einführung*).

1.2 Vertikale Boxen

Ein Bild an eine feste Stelle im Dokument einzubauen ist mit Hilfe der bekannten (s. LaTeX *an der hera (Briefe, Boxen)*) `minipage`- oder `parbox`-Umgebung möglich. Allerdings gibt es in diesem Fall keinen speziellen LaTeX-Befehl für die Bildbeschriftung. Letztere ist mit Hilfe üblicher LaTeX-Mittel zu organisieren.

Befehlssyntax:

`\begin{minipage}[`*pos*`]`{*breite*} *bild* `\end{minipage}`

`\parbox[`*pos*`]`{*breite*} { *bild* }

pos - Positionierungsparameter, der folgende Werte annehmen kann:

`b` - die unterste Zeile der Box ist mit der laufenden Zeile ausgerichtet,
`t` - die oberste Zeile der Box ist mit der laufenden Zeile ausgerichtet.

Bei fehlendem Positionierungsparameter ist die Box vertikal zentriert zur laufenden Zeile.

breite - Maßangabe in einer der Längeneinheiten.
bild -steht für die Definition des einzurahmenden Graphikfiles.

Es sind verschiedene Kombinationen und Verschachtelungen von Boxbefehlen erlaubt. Man kann auch **Boxbefehle innerhalb einer Gleitumgebung** verwenden, um Bildbeschriftungen von LaTeX erzeugen zu lassen.

2 DVIPS-Treiber und special-Befehl

Ein LaTeX Formatierungsergebnis ist ein geräteunabhängiger (übertragbar auf einen beliebigen Rechner) **DVI-File** (.dvi). Seine Bearbeitung für die Ausgabe auf einem konkreten Drucker (oder Bildschirm) übernehmen spezielle Treiber.
An der `hera` sind dafür zwei Treiber installiert:

> **dvips** - Umwandlung von DVI-Files in PostScript-Files,
> (Version 5.4 von Tomas Rokicki, letzte Revision 1990)
>
> **xdvi** - Bildschirmausgabe von DVI-Files.
> (Version, May 1990 von mehreren Autoren)

Die Einbindung von anderen graphischen Formaten oder TeX-fremden Files in LaTeX-Dokumente geschieht in der Regel über Drucker-Treiber (**dvips** an der `hera`). Dafür ist im LaTeX ein `\special`-Befehl vorgesehen.

Befehlssyntax:

> `\special`{*Treibertext*}

Der *Treibertext*, der im `\special`-Befehl enthalten ist, wird von LaTeX selbst bei der Formatierung des Dokuments ignoriert. Von dem DVIPS-Treiber wird dieser Befehl jedoch als auszuführender erkannt und entsprechend der Information in den geschweiften Klammern abgearbeitet.

Die Einbindung von PostScript-Graphik oder PostScript-Texten (Programmen) kann folglich mit Hilfe des DVIPS-Treibers behandelt und in DVI-Files eingebaut werden.

Der DVIPS-Treiber erzeugt ein sogenanntes Encapsulated PostScript File (EPS-File).

Vom **xdvi**-Treiber werden PostScript-Informationen im `\special`-Befehl nicht behandelt.

Es ist zu beachten:

> Bei der Einbindung eines LaTeX-fremden Files mittels `\special`-Befehl muß eine entsprechende Platzreservierung und Platzzuweisung im LaTeX-Dokument vorgenommen werden.

Das geschieht mit den speziellen LaTeX-Stilfiles (DVI-Postprozessoren)

> `epsf.sty` oder `psfig.sty`,

bzw. durch ergänzende LaTeX-Umgebungen:

> `figure`, `minipage`, `parbox`.

3 Encapsulated PostScript-File (EPS-File)

3.1 EPS-File und "Boundig-Box"-Angaben

Das EPS-Format ist ein Satz von Regeln für in PostScript-Sprache geschriebene Programme. Beispiel der Anfangsstruktur des EPS-Files (AVS Ausgabe - 'image to postscript'):

```
%!PS-Adobe-3.0 EPSF-3.0
%%BoundingBox: 36 91 576 735
%%Creator: image to postscript
%%CreationDate: Tue Apr 13 13:43:34 1993
%%EndComments
%%Pages: 1
%%EndProlog
%%Page: 1 1

/bitdump % stk: width, height, iscale
................................
```

Am Anfang stehen meistens mehrere META-Kommentare, die mit den Steuerzeichen %% beginnen und verschiedene Information über das PostScript-File enthalten (Herkunft, Erstellungsdatum u.s.w.). Folgende Regeln sind für ein EPS-File bindend:

- die erste Zeile muß mit den zwei Steuerzeichen `%!` starten, gefolgt von einem Text, wie `PS-Adobe-3.0 EPSF-3.0`.

  In diesem Text sind die Versionsnummern der *Struktur-Konventionen (DSC)* von 'adobe' und des *Encapsulated PostScript-Formats (EPSF)* angegeben. In unserem Beispiel entsprechen beide der aktuellen Version 3.0.

  Das saubere Einhalten der *DSC* von verschiedenen Anwendungsprogrammen ist notwendig, um die Austauschbarkeit (export-import) von EPS-Files zu ermöglichen (s. *Postscript Language Reference Manual*)

- Der META-Kommentar

  ```
  %%BoundingBox: 36 154 576 672
  ```

  muß in einem EPS-File unbedingt vorhanden sein. In unserem Beispiel steht er an zweiter Stelle (was nicht bindend ist). In der `BoundigBox` sind die PostScript-Koordinaten des EPS-Files spezifiziert.

Der DVIPS-Treiber sucht immer zuerst nach dem `BoundigBox-Kommentar`, um in der Lage zu sein, PostScript-Graphik (Text) zu behandeln und in seine Ausgabe einzubinden.

Die BoundingBox enthält vier PostScript-Koordinaten eines das EPS-File umschließenden Rechtecks, die seine äußeren Grenzen auf der Ausgabeseite markieren. Die untere linke Ecke der Seite ist der

```
PostScript Koordinatenanfang (0,0).
```

Anordnung und Bedeutung der BoundingBox-Parameter:

`%%BoundingBox: llx lly urx ury`

`llx`	-	untere linke Ecke: x-Koordinate
`lly`	-	untere linke Ecke: y Koordinate
`urx`	-	obere rechte Ecke: x-Koordinate
`ury`	-	obere rechte Ecke: y-Koordinate

Alle Koordinatenwerte müssen in sogenannten Big-Points (**bp**) angegeben werden.
Es gilt:

$$\begin{aligned} 72\text{ bp} &= 1\text{ inch } (1\text{ inch} = 2.54\text{cm}) \\ 1\text{bp} &\cong 0.0353\text{cm} \\ 1\text{cm} &= 28.35\text{ bp} \end{aligned}$$

Die BoundingBox, die z.B. einer A4-Seite entspricht (210*297mm), lautet:

`%%BoundingBox: 0 0 595 842`

Das EPS-Format wird immer mehr als Standard für graphische PostScript-Ausgaben anerkannt. EPS-Files können leicht in LaTeX-Dokumente eingebunden werden.
Viele Anwendungsprogramme, wie **AVS, PAW**, erlauben es, ihre Ergebnisse als EPS-File abzulegen. Andere, wie **GKS**, haben die Möglichkeit, ihr Ausgabe-Format mit Hilfe spezieller Konvertierungsprogramme in ein EPS-File zu überführen.

3.2 Quellen der PostScript-Files auf den zentralen Servern

Auf den zentralen Servern `hera` und `rita` gibt es folgende Möglichkeiten, PostScript-Files (`ps`) bzw. Encapsulated PostScript-Files (`eps`) zu erzeugen:

1. **AVS** (Release 3.9 der `hera` bzw. 4.0 der `rita`):
   - Modul output_postscript: `ps` (`hera + rita`)
   - Modul image_to_postscript: `ps`, `eps` (`hera + rita`)
   - Graph Viewer: `ps` (`hera + rita`; keine Farbunterstützung)
2. **XWindow Dump**:
   `xwd / xwd2ps: eps` (`hera + rita, rita`, ggw. nur `xwd` möglich)
3. **GKS-CGI-Treiber**: `ps`, wie `eps` handhabbar (nur `hera`,).

Nähere Angaben zu 1. und 2. findet man im Schrifttum der Nutzerseminare zur Graphik vom 23. 03. 1993.

Die vom Graph Viewer und vom GKS der `hera` erzeugten PostScript-Files sind keine Pixelbilder, sondern echte Liniengraphiken entsprechend Plotterzeichnungen (ohne aliasing-Effekte) mit der Genauigkeit des Laserdruckers. Dabei werden vom Graph Viewer nur eine Farbe, vom GKS sieben Farben unterstützt. Das GKS der `rita` (wie auch das darüberliegende graPHIGS) unterstützen keine PostScript-Ausgabe, sondern HPGL-Files (ebenfalls druckbar bzw. in PostScript umformbar).

3.3 Schritte zur Einbindung eines Graphikfiles

Um sauber PostScript-Graphik in ein Dokument einzubinden, sind üblicherweise folgende Vorbereitungen notwendig:

1. Das Bild (Graphikfile) mit dem **ghostview**-Programm auf dem X-Terminal anschauen, um seine Größe und Lage auf der Seite festzustellen.
   Aufruf :

   `gv` *filename[.ps]* oder
   `ghostview -a4` *filename[.ps]*

   Die vier Koordinaten der BoundingBox-Parameter ('reichlich', d.h. 1-2mm zugeben) bezüglich der unteren linken Seitenecke des Bildes in **bp**-Maßeinheiten aufschreiben. (Im linken oberen Feld des Ghostview-Fensters werden nach dem `filename` und `datum` die `x` und `y` Koordinaten des Bildschirmzeigers angezeigt, wenn er sich im Ghostview-Betrachtungsfeld befindet.)

2. Mit einem beliebigen Editor oder Unix-Befehl das Graphikfile anschauen, um festzustellen, ob es ein EPS-File ist. Ist das der Fall, dann müssen die BoundigBox-Angaben mit den im vorhergehenden Schritt festgelegten übereinstimmen. Ist das nicht der Fall, dann muss evtl. die Bilderzeugung wiederholt werden, um die 'leere' Bildumrahmung zu reduzieren.

3. Es ist wichtig das LaTeX-Seitenformat zu kennen, insbesondere Textbreite und Texthöhe (s.g. 'Rumpf'). Diese sind abhängig von:

   - dem gewählten LaTeX- Hauptstil (`article, book, letter, u.s.w.`),
   - der Wahl der Schriftgröße für das Dokument (`10pt, 11pt, 12pt`)
   - den ausgewählten optionalen Stilparametern (z.B. `a4.sty`).

   Folgendes einfache LaTeX-Textbeispiel kann behilflich sein, die Seitengröße (Breite und Höhe) für den zu formatierenden Text zu bestimmen:

```
\documentstyle[12pt,a4,twoside]{article}  % gewuenschte Stilparameter
\pagestyle{headings}                      % gewuenschte Seitenstil
%\pagestyle{empty}
%\pagestyle{plain}
\begin{document}
   Text Text........................      % eine reichliche Seite
   .................................      % mit 'Text' auffuellen
\end{document}
```

   Nach der LaTeX-Formatierung und Druckerausgabe der entsprechenden Seite sind sowohl Seitengrösse als auch 'Rumpf'-Lage auf der Seite durch 'Text ..'-Worte markiert.

4 epsf.sty

4.1 Befehle

Mögliche Aktivierung im LaTeX :

`\input epsf`

`\documentstyle[...,epsf,...]`{*hauptstilparameter*}

Syntax des Befehlsaufrufs an der Graphikeinbindungsstelle im LaTeX-Text:

`\epsfbox`{*psfilename.ps*}

oder mit BoundigBox-Angaben, als optionale Parameter

`\epsfbox[llx lly urx ury]{psfilename.ps}`

`epsf.sty` hat folgende Funktionen bei der Graphikeinbindung :

- Einlesen der Koordinaten eines PostScript-Files, die im BoundigBox-Kommentar angegeben sind,
- entsprechend diesen Koordinaten wird an der aktuellen Stelle (Aufrufstelle) im LaTeX eine vertikale Box (`vbox`) mit reichlicher Platzreservierung gesetzt,
- Behandlung auch von 'Non-EPS-File' bei optionaler BondingBox-Koordinatenangabe möglich. Sind die optionalen Werte `llx, lly, urx, ury` nicht angegeben, gelten die für BoundigBox im `epsf.sty` eingestellten Werte `72 72 540 720`.
- Ermöglicht eine gewünschte proportionale Bildskalierung bei Aufruf eines der zwei Befehle

  `\epsfxsize=`*längenmaß*
  `\epsfysize=`*längenmaß*

  Werden beide aufgerufen, gilt der kleinere Skalierungsfaktor für beide Achsen.

  Achtung: Der Wert *längenmaß* ist nur für den laufenden Aufruf von `\epsfxsize`-Befehl gültig und wird danach auf "natürliche Größe " der BoundingBox gesetzt.
- Aufruf des `\special`-Befehls und Übergabe des Graphikfiles zur weiteren Bearbeitung an den DVIPS-Treiber.

Querformat-Ausgabe eines DVI-Files kann mit dem DVIPS-Befehl

`\special{landscape}`

angefordert werden.

Achtung:

Dabei sind die Angaben der LaTeX-Befehle `\textheight` und `\textwidth` zu vertauschen (s. entsprechendes Beispiel im nächsten Abschnitt).
Da der `\special{landscape}`-Befehl nach seinem Aufruf nicht zurücksetzbar ist, sind bei Querformatausgabe einiger Seiten diese als Einzeldokumente zu erstellen.

4.2 Beispiele der Graphikeinbindung mit dem epsf.sty

Ein einfaches Beispiel: AVS EPS-File

```
\vskip 0.5cm
\epsfxsize=8cm
\begin{figure}[h]
\begin{center}
% Bild mit \mbox einrahmen
\mbox{\epsfbox{mono7.ps}}
\end{center}
\caption{\bf `mono7'- skaliert und zentriert}
\end{figure}
```

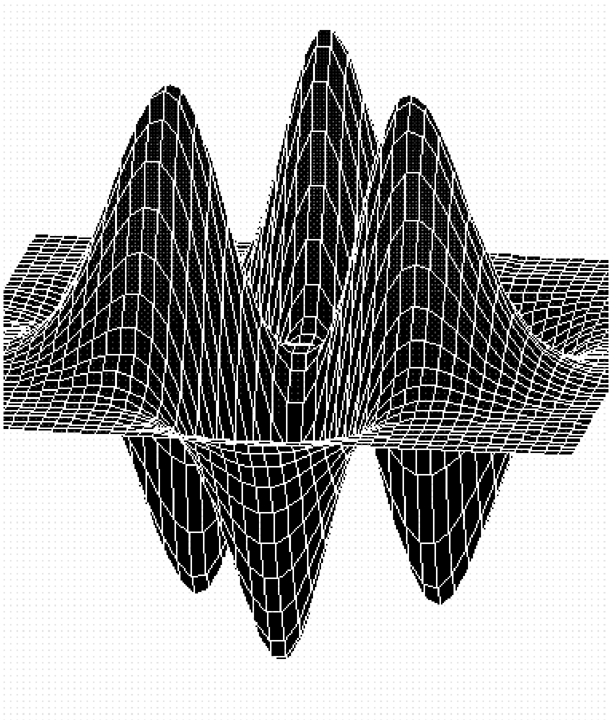


Abbildung 1: **'mono7'- skaliert und zentriert**

Beschriftungsmöglichkeiten, AVS-Files

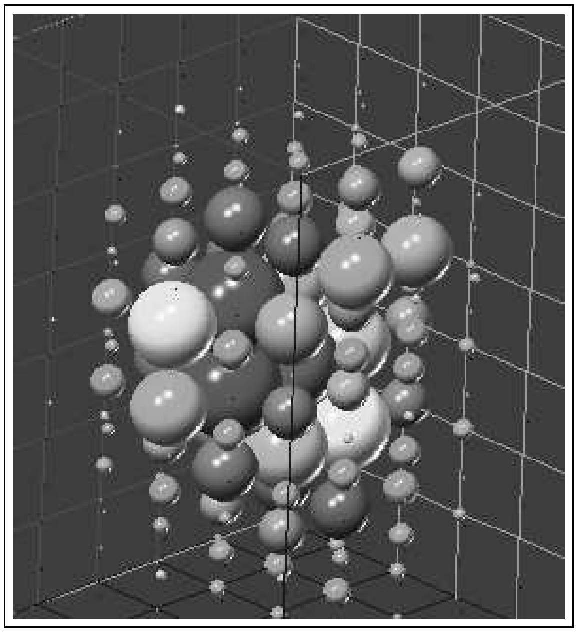
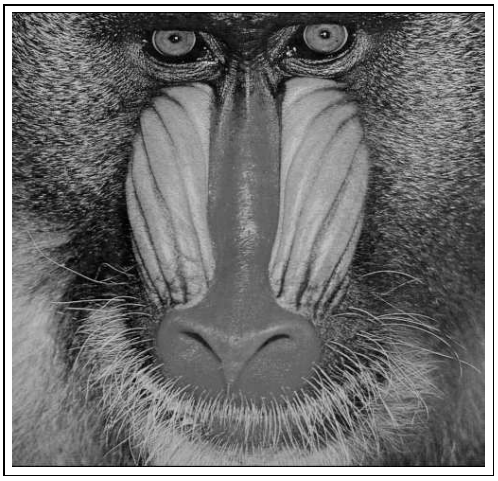


Abbildung 2: 'mono2', Unterschrift mittels minipage-Umgebung

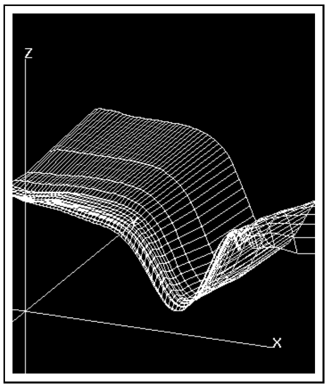
```
\epsfysize=8cm
\begin{figure}[h]
\begin{minipage}[t]{0.45\textwidth}
\fbox{\epsfbox{mono2.ps}}
\end{minipage}
\hfill \mbox{} \hfill
\begin{minipage}[b]{0.45\textwidth}
\caption{‘mono2’, Unterschrift mittels ...}
\vskip 4cm
\vfill
\mbox{}
\end{minipage}
\end{figure}
```

Abbildung 3: 'Mandrill', Beschriftung links



```
\epsfysize=6cm
\begin{figure}[h]
\begin{minipage}[b]{0.45\textwidth}
\caption{`Mandrill', Beschriftung links}
\vskip 3cm \vfill \mbox{}
\end{minipage}
 \hfill \mbox{} \hfill
\begin{minipage}[b]{0.45\textwidth}
\fbox{\epsfbox{mono6.ps}}
\end{minipage}
\end{figure}
```

Abbildung 4: AVS-File 'mono4', Beschriftung breiter als das Bild



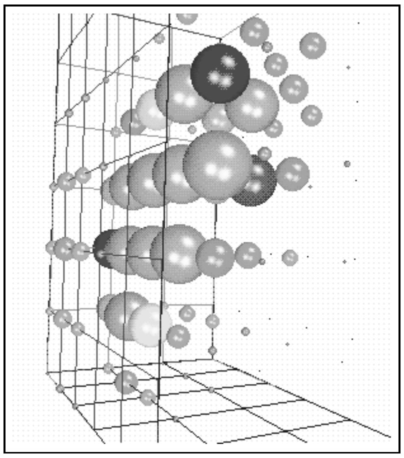


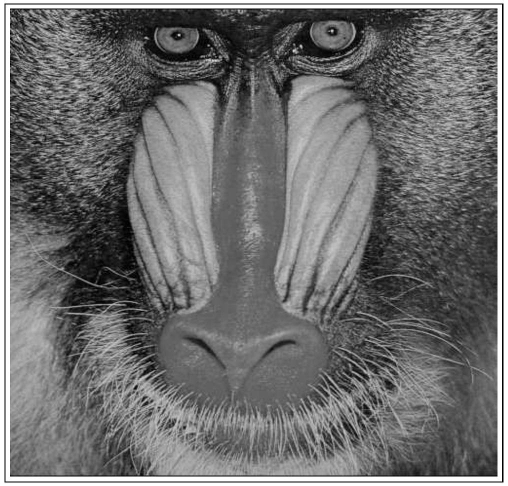
Abbildung 5: AVS-File 'mono1', Beschriftungbreite an die Breite des Bildes angepaßt

Auszug aus dem LaTeX-Text des letzten Beispiels:

```
........................
\fbox{\epsfbox{mono4.ps}}
\vfill\mbox{}\vfill
\parbox{5cm}{\caption{AVS-File ‘mono1’, .....}}
......................
```

Unser Mandrill, Landscape-Ausgabe

```
\textheight6.5in
\textwidth10in
\special{landscape}
\epsfysize=10cm
\begin{figure}[h]
\begin{minipage}[b]{14cm}
 ... \LaTeX-Text ...
\end{minipage}
 \hfill
\begin{minipage}[b]{10cm}
\fbox{\epsfbox{mono6.ps}}
\end{minipage}
\end{figure}
```



Zwei EPS-Files nebeneinander

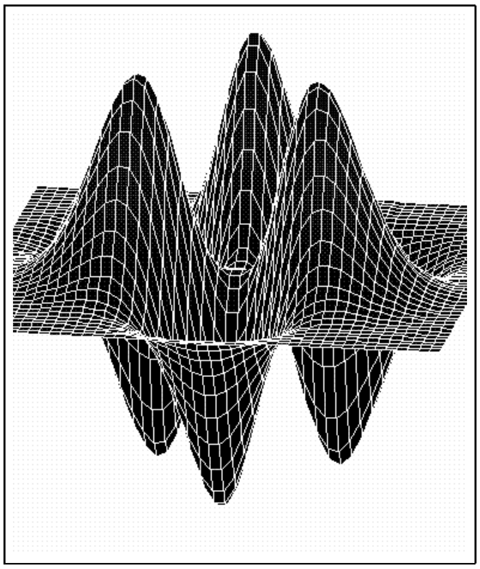


Abbildung 6: ‘mono7’

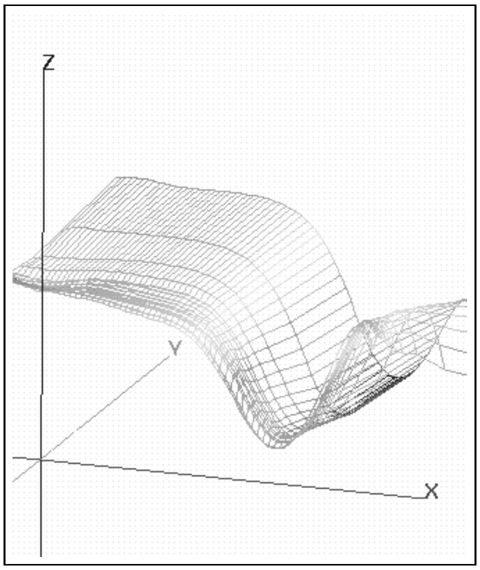


Abbildung 7: ‘mono3’

```
\begin{figure}[h]
\begin{minipage}[b]{6.5cm}
\epsfxsize=6cm
\begin{center}
\fbox{\epsfbox{mono7.ps}}  % einrahmen mit \fbox
\caption{‘mono7’}
\end{center}
\end{minipage}
\hfill \mbox{} \hfill
\begin{minipage}[b]{6.5cm}
\epsfxsize=6cm
\begin{center}
\fbox{\epsfbox{mono3.ps}}  % einrahmen mit \fbox
\caption{‘mono3’}
\end{center}
\end{minipage}
\end{figure}
```

Vier AVS EPS-Files

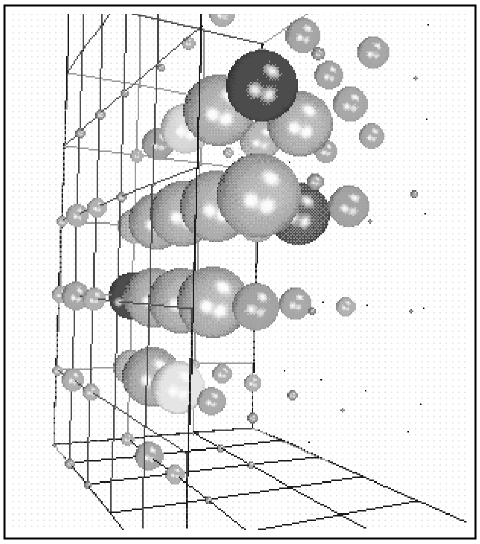


Abbildung 8: mono1

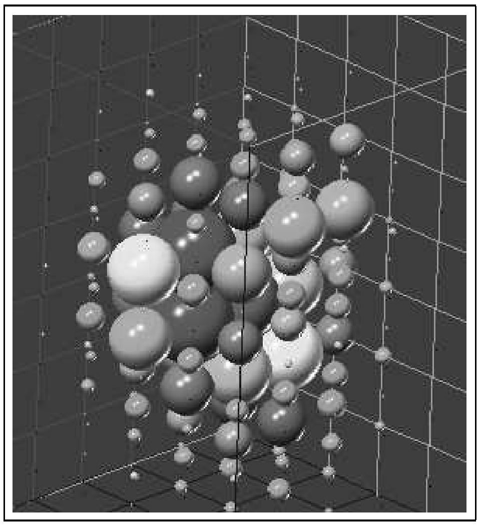


Abbildung 9: mono2

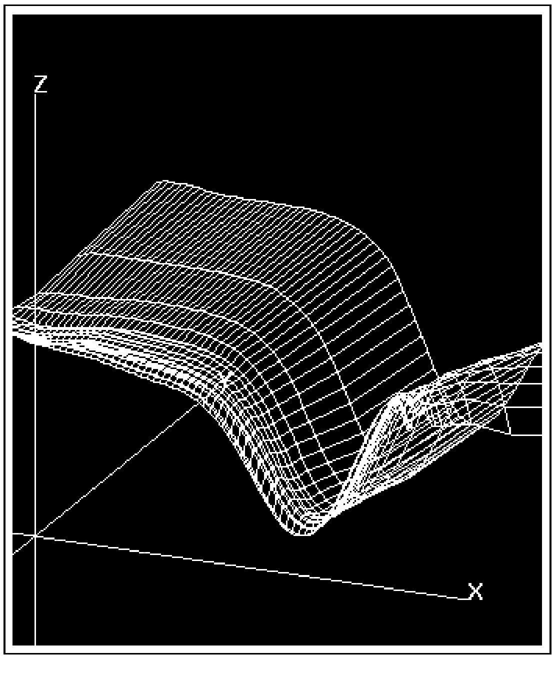


Abbildung 10: mono4

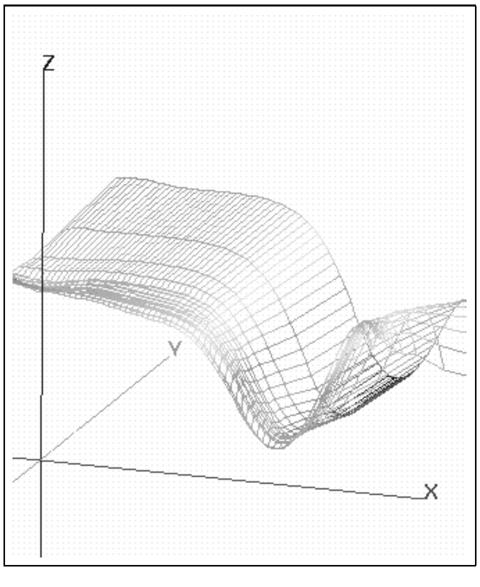


Abbildung 11: mono3

5 psfig.sty

5.1 Befehle

Mögliche Aktivierung im LaTeX :

`\input psfig`

`\documentstyle[...,psfig,...]{`*hauptstilparameter*`}`

Syntax des Befehlsaufrufs an der Graphikeinbindungsstelle im LaTeX-Text:

```
\psfig{file=, height=, width=,
     bbllx=, bblly=, bburx=, bbury=, angle=, clip=}
```

Angabe aller Parameter außer des `file`-Parameters ist optional.

`file` - ist der *filename*

`width` - gewünschtes *breitenmaß* des Bildes
`height` - gewünschtes *höhenmaß* des Bildes

Ohne Angabe von Breite und Höhe gelten die BoundingBox-Größen (`BB`-Größen)

`bbllx, bblly, bburx, bbury` - BoundingBox-Angaben *llx, lly, urx, ury* in **bp**

Ohne Angabe dieser BoundingBox-Größen gelten die BB-Angaben des EPS-Files

`angle` - spezifiziert Rotationswinkel in Grad
`clip=` - Option, die dafür sorgt, daß kein Bildteil außerhalb der BoundingBox erscheint. Aufruf: `clip=` (ohne Wertzuweisung).

`psfig.sty` hat gleiche Funktionen bei der Graphikeinbindung, wie epsf.sty.
Unterschiede bzw. zusätzliche Möglichkeiten:

- Bei gleichzeitiger Angabe von *höhenmaß* und *breitenmaß* werden beide buchstäblich genommen. D.h. Bildverzerrung (unterschiedliche Skalierungsfaktoren für x und y) kann erzwungen werden.
- Der `\psdraft`-Befehl ermöglicht eine Umstellung in die zeitsparende Skizzenmode. D.h. Ausdruck nur des Rahmens des einzubindenden Graphikfiles an der geforderten Stelle, mit geforderter Größe und Namensangabe (im Rahmen) des Files.
- Der `\psfull`-Befehl schaltet die Skizzenmode aus.

5.2 Beispiele der Graphikeinbindung mit dem psfig.sty

\psdraft- und \psfull -Befehl

```
\begin{figure}[b]
\psdraft                                   % \psdraft
\begin{minipage}[b]{5cm}
\begin{center}
\mbox{\psfig{file=mono2.ps,width=5cm}}  % \mbox{...} !!!
\end{center}
\end{minipage}
\psfull                                    % \psfull
\hfill \mbox{} \hfill
\begin{minipage}[b]{5cm}
\begin{center}
\mbox{\psfig{file=mono2.ps,width=5cm}} % \mbox{...} !!!
\end{center}
\end{minipage}
\caption{‘mono2’, als {\tt $\backslash$psdraft}- und
 {\tt $\backslash$psfull}-Ausgabe}
\end{figure}
```

Abbildung 12: ‘mono4’, als `\psdraft`- und `\psfull`-Ausgabe

Bildverzerrung

```
\begin{figure}[h]
\begin{minipage}[b]{4cm}
\fbox{\psfig{file=mono7.ps, \mbox{}\hfill\mbox{}
height=8cm,width=4cm}}
\caption{\bf‘mono7’, height=8cm, width=4cm}
\end{minipage}
\hfill
\begin{minipage}[b]{8cm}
\fbox{\psfig{file=mono7.ps, \mbox{}\hfill\mbox{}
height=4cm,width=8cm}}
\caption{\bf ‘mono7’, height=4cm, width=8cm}
\end{minipage}
\end{figure}
```

zwei verzerrte Bilder nebeneinander

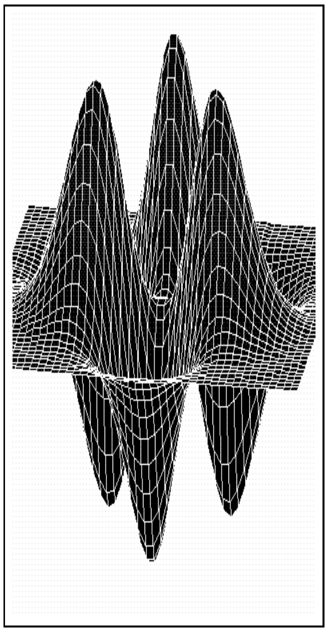


Abbildung 13: **‘mono7’, height=8cm, width=4cm**

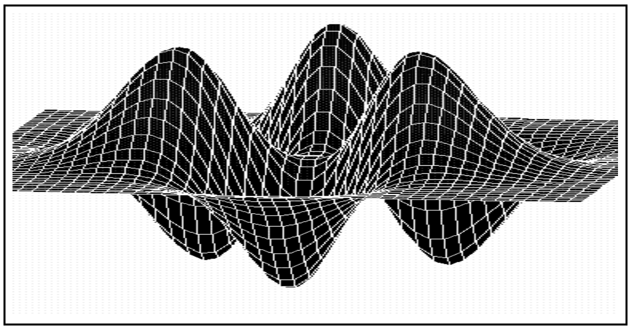


Abbildung 14: **‘mono7’, height=4cm, width=8cm**

Einbindung von NON-EPS-Files

PAW-Ausgabe

```
\begin{figure}[h]
\begin{center}
\mbox{
% Achtung!!!
% ‘space’ oder ‘newline’ zwischen
%  psfig-Parametern nicht erlaubt
%
\psfig{file=paw.ps,width=10cm,bbllx=16bp,%
bblly=140bp,bburx=576bp,bbury=701bp}}
\end{center}
\caption{\bf ‘paw.ps’}
\end{figure}
```

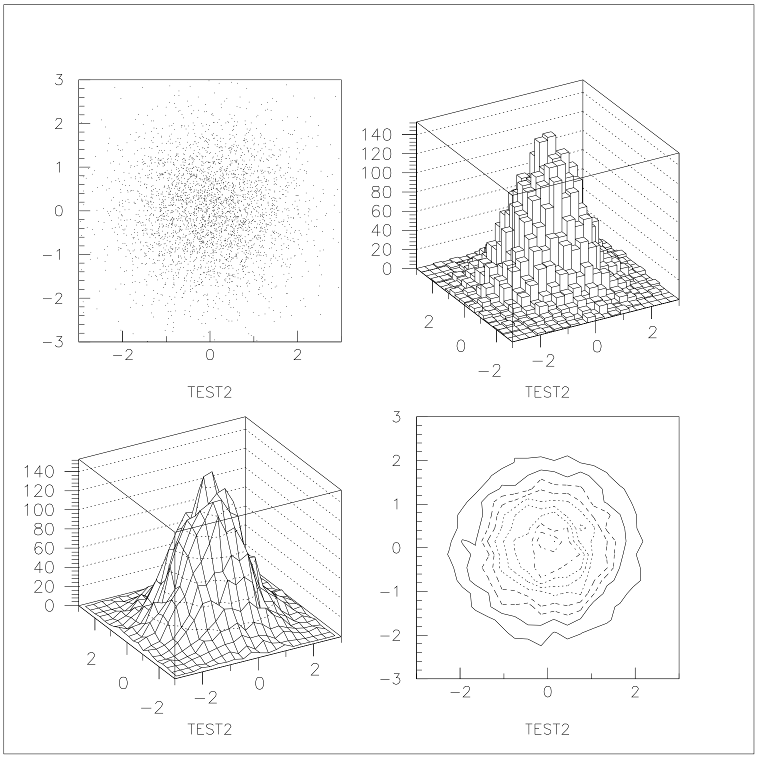
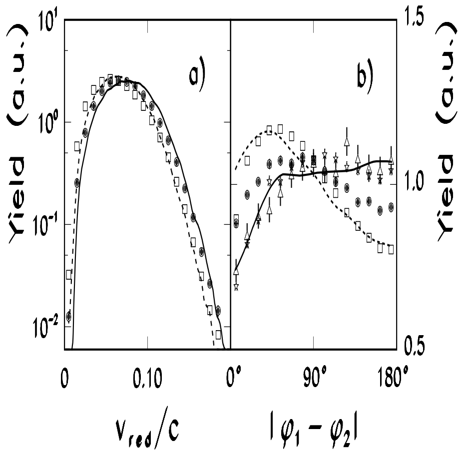
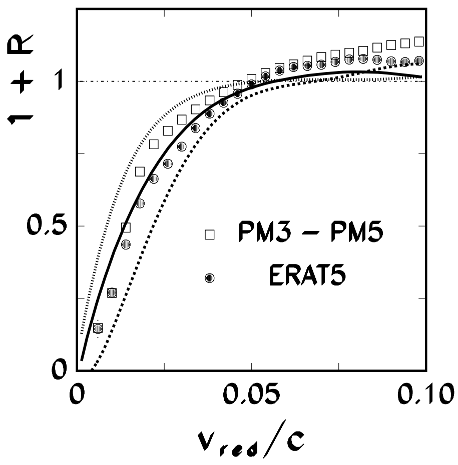


Abbildung 15: **‘paw.ps’**

Beispiel mit `tabular`-Umgebung



'fig1'



'fig3'

Zwischen optionalen Parametern beim dem Aufruf von `\psfig` dürfen keine 'spaces' oder 'newline' auftreten.

```
\begin{figure}[h] \begin{center}
\begin{tabular}{||c|c||}
\hline \hline &\\
\psfig{file=fig1.ps,width=6cm,height=6cm,%
bbllx=30bp,bblly=39bp,bburx=564bp,bbury=308bp}
\hspace*{0.2cm}&\hspace*{0.2cm}
\psfig{file=fig3.ps,width=6cm,height=6cm,%
bbllx=50bp,bblly=39bp,bburx=549bp,bbury=520bp}\\ &\\
\bf 'fig1'\hspace*{0.2cm}&\hspace*{0.2cm}\bf  'fig3'\\
&\\ \hline\hline
\end{tabular} \end{center}
\end{figure}
```

6 Einfache Makros zur Graphikeinbindung

Die in diesem Abschnitt vorgestellten Makros basieren auf Befehlen von **epsf.sty** .

- Es wird intern der `\epsfbox`-Befehl aufgerufen.
- Bildskalierung wird mittels der Angabe des entsprechenden Parameters für x- oder y-Richtung durchgeführt (`\epsfxsize`- oder `\epsfysize`-Befehl).
- Bei der Einbindung in LaTeX-Dokumente wird die `\figure`-Umgebung mit dem Positionierungsparameter `h` benutzt.

Aktivierung im 'LaTeX :

`\input epsf` (s. Abschnitt 4)

`\input usermac`

Makros zur Einbindung eines EPS-Files:

Aufruf:

`\onefigx`{*bildbreite*}{*filename.ps*}{*beschriftungstext*} oder

`\onefigy`{*bildhöhe*}{*filename.ps*}{*beschriftungstext*}

Wird Bildumrahmung gewünscht, können die Makros

`\onefigxr` und `\onefigyr`

mit dem gleichen Parameteraufruf verwendet werden. Das am Ende des Makronamens angefügte `r` bedeutet Bildumrahmung mit dem `\fbox`-Befehl.

Parameterbedeutung:

bildbreite	-	Maßangaben in einer der Längeneinheiten,
bildhöhe	-	Maßangaben in einer der Längeneinheiten,
filename.ps	-	muß immer angegeben sein,
beschriftungstext	-	erscheint als Unterschrift.

Werden die Angaben von *bildbreite* (*bildhöhe*) oder *beschriftungstext* nicht gewünscht, muß für die entsprechenden Parameter bei dem Makroaufruf {} stehen. Bei den fehlenden Angaben für *bildbreite* (*bildhöhe*) werden die Graphikfiles in ihren 'naturellen' Größen (BondingBox-Angaben) eingebunden.

Makros zur Einbindung zwei nebeneinander stehender EPS-Files:

`\twofigx`{*bildbreite1*}{*filename1.ps*}{*beschriftungstext1*}
{*bildbreite2*}{*filename2.ps*}{*beschriftungstext2*} oder

`\twofigy`{{*bildhöhe1*}{*filename1.ps*}{*beschriftungstext1*}
{*bildhöhe2*}{*filename2.ps*}{*beschriftungstext2*}

Wird Bildumrahmung gewünscht, müssen folgende Makronamen verwendet werden:

`\twofigxr` und `\twofigyr`

Bei diesen Makros haben die Parameter die gleiche Bedeutung wie bei `\onefigx` und `\onefigy`.

Verständlicherweise dürfen jetzt Angaben für die entsprechende *bildbreite* oder *bildhöhe* nicht fehlen ({}-Angabe nicht erlaubt), und ihre Werte sollen nicht größer als 0.5`\textwidth` sein.

Beispiele für EPS-Files

```
\onefigxr{6cm}{hund.ps}
{\bf ‘ghostview Window-Dump’ mit xwd|xwd2ps erstellt}
```

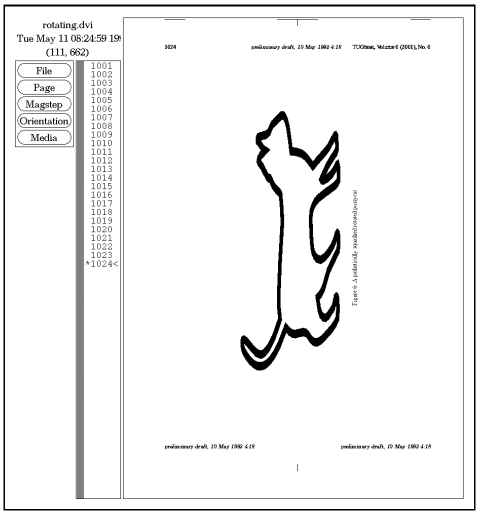


Abbildung 16: **‘ghostview Window-Dump’ mit** `xwd|xwd2ps` **erstellt**

```
\twofigyr{4cm}{mono4.ps}{‘mono4’}{6cm}{mono3.ps}{}
```

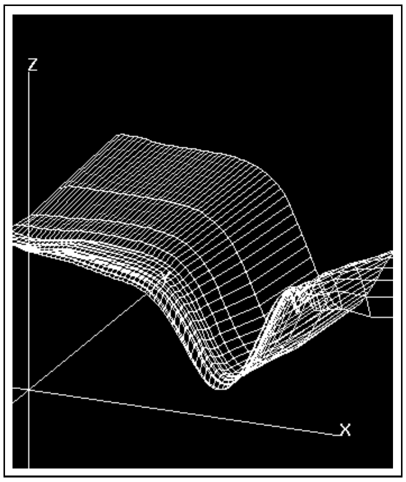
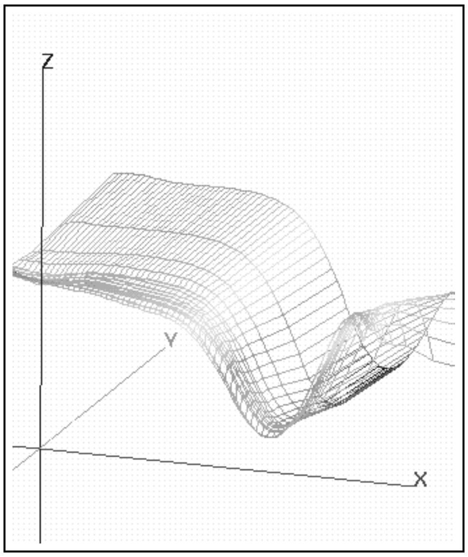


Abbildung 17: ‘mono4’



Beispiele für NON-EPS-Files

```
\twofigyr{6cm}{paw.ps}{PAW-‘paw.ps’}
        {6cm}{fig3.ps}{PAW-‘fig3.ps’}
```

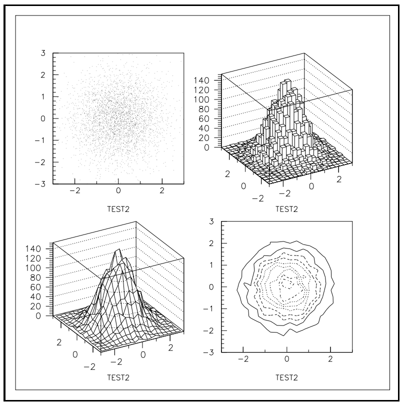


Abbildung 18: PAW-‘paw.ps’

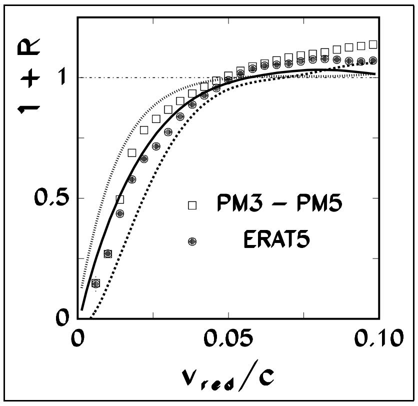


Abbildung 19: PAW-‘fig3.ps’

```
\onefigx{6cm}{iso4.ps}{AVS-‘iso4.ps’}
```

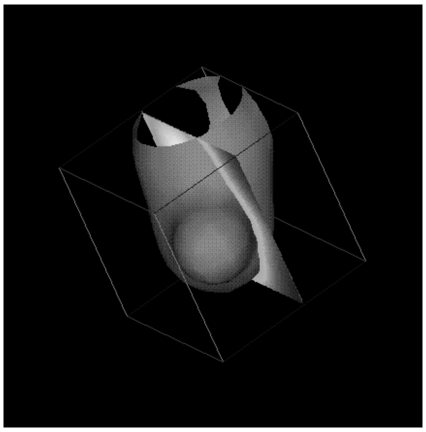


Abbildung 20: AVS-‘iso4.ps’

Achtung:

Probleme bei der Einbindung von NON-EPS-Files sind nicht ausgeschlossen.

7 Einbindungsmöglichkeiten direkt über den special-Befehl, Einbindungsprobleme

In der DVIPS-Dokumentation sind die Einbindungsmöglichkeiten von PostScript-Graphikfiles und PostScript-Befehlsfolgen (PS-Texten) über den `\special`-Befehl gegeben. Aus folgenden Gründen müssen diese Möglichkeiten mit Vorsicht benutzt werden:

- Bei der Verwendung der dort angegebenen Befehle muß der Nutzer selbst für eine Plazierung auf eine gewünschte Stelle im Dokument (z.B. LaTeX-Aufrufstelle) sorgen, was auch problematisch sein könnte.
- Ein sicherer Umgang mit dieser Einbindungstechnik (insbesondere für PostScript-Texte) bedarf bestimmter Erfahrungen sowohl in LaTeX-Formatierungstechnik als auch in PostScript-Programmierung.
- Einbindungen von Graphik-Files aus anderen Anwendungsprogrammen können viel einfacher und komfortabler mit den Befehlen von `epsf.sty` oder `psfig.sty` vorgenommen werden. In beiden Styles wird automatisch für eine richtige Plazierung des einzubindenden Files gesorgt und danach der `\special`-Befehl zur Übergabe des Graphikfiles an den DVIPS-Treiber aufgerufen. Entstehen dabei **Probleme**, liegen die Ursachen meistens an den einzubindenden Graphikfiles selbst, oder an den **häufig unkorrekten BoundingBox-Angaben** in den entsprechenden Befehlen.

An dieser Stelle sollen einige Regeln für einzubindende Graphikfiles erwähnt werden.

1. Am sichersten können Einbindungen von EPS-Files in LaTeX-Dokumente vorgenommen werden. Von einer Veränderung des EPS-Files (auch seiner Kommentarangaben) ist den Nutzern abzuraten.
2. In jedem EPS-File steht in der ersten Zeile die DSC-Versionsnummer. DSC-Versionen enthalten Listen zugelassener interpretierbarer Strukturkommentare und Listen von einigen in einem EPS-File verbotenen PostScript-Operatoren (s. empfohlene Literatur).

   Es gibt DSC-Versionen mit verschiedenen Entwicklungsstufen (1.0, 2.1, 3.0). AVS-EPS-Files enthalten z.B. die aktuelle DSC-Versionnummer 3.0. Der z.Z. an der `hera` installierte DVIPS-Treiber erzeugt EPS-Files mit der DSC-Version 2.0.

   Enthalten die von Anwendungsprogrammen erzeugten EPS-Files DSC-Kommentare, die von EPS-Files importierenden Programmen nicht richtig interpretiert werden, kann es evtl. bei der Einbindung oder beim Druck des einbindungsenthaltenden EPS-Files zu Fehlern kommen.
3. Aus den bisherigen Erfahrungen (AVS-EPS-Files, XWD2PS-EPS-Files, DVIPS-EPS-Files (Einzelseitenausgabe)) entstehen keine Probleme bei der Einbindung von EPS-Files in LaTeX-Dokumente.
4. Wenn aber ein NON-EPS-File mit Hilfe von DVIPS einzubinden ist, können keine Forderungen an die DVIPS-Einbindungstechnik gestellt werden. Es ist dafür zu sorgen, daß das entsprechende Anwendungsprogramm ein Interface für EPS-Files-Erzeugung in der aktuellen DSC-Version bekommt.

8 rotating.sty

Die Drehungsdienste des DVIPS-Treibers können durch Verwendung von `rotating.sty` unkompliziert benutzt werden. `rotating.sty` gehört wie auch `epsf.sty` und `psfig.sty` zu den sog. DVI-Postprozessoren.
`rotating.sty` übernimmt alle notwendigen Vorbereitungen für die Einbindung eines EPS-Files an der aktuellen Stelle (Aufrufstelle) im LaTeX-Dokument bei Berücksichtigung seiner Lage (Drehungswinkel) bezüglich der Grundlinie, und übergibt seine 'PostScript-Bearbeitung' an den DVIPS-Treiber.

`rotating.sty` ermöglicht die Durchführung verschiedener Drehungsoperationen sowohl für einzubindende **Graphikfiles** als auch für LaTeX-**Textmaterial**.

8.1 Befehle

Mögliche Aktivierung im LaTeX :

`\documentstyle[...,rotating,...]{`*hauptstilparameter*`}`

Im `rotating.sty` sind drei Umgebungen definiert:

`\begin{rotate}{angle}` *Figure(Text)* `\end{rotate}`

`\begin{turn}{angle}` *Figure(Text)* `\end{turn}`

`\begin{sideways}{angle}` *Figure(Text)* `\end{sideways}`

Dabei bedeuten:

Figure(Text)	-	Drehungsobjekt - ein Graphikfile oder
	-	eine LaTeX-Box mit Text (auch ein Wort).
`angle`	-	Winkelangabe in Grad, bezüglich Koordinaten, dessen
	-	Koordinatenanfang sich an der Aufrufstelle befindet.
`rotate`	-	führt eine Drehung ohne Reservierung
	-	des horizontalen Zwischenraums zu dem
	-	danach folgenden Text aus
`turn`	-	führt eine Drehung mit einer Reservierung
	-	des horizontalen Zwischenraums zu dem
	-	danach folgenden Text aus
`sideways`	-	ist ein Sonderfall - bedeutet eine Drehung um -90 Grad

8.2 Beispiele

Start SEMINAR Ende

```
Start
\begin{rotate}{-40}% !!!
SEMINAR
\end{rotate}
Ende
```

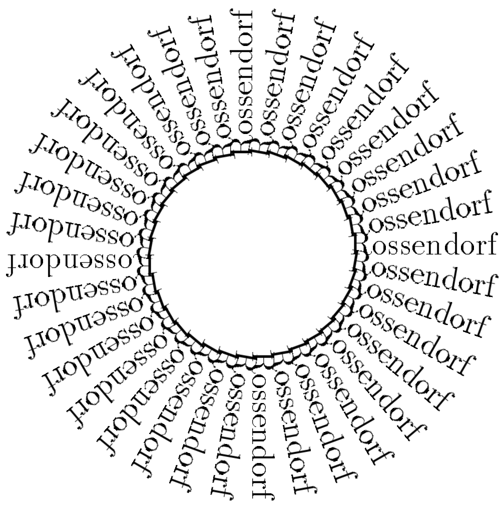
Start SEMINAR Ende

```
Start
\begin{sideways}%  !!!
SEMINAR
\end{sideways}
Ende
```

Start SEMINAR Ende

```
Start
\begin{turn}{40}%  !!!
SEMINAR
\end{turn}
Ende
```

Beispiel ohne LaTeX-Text



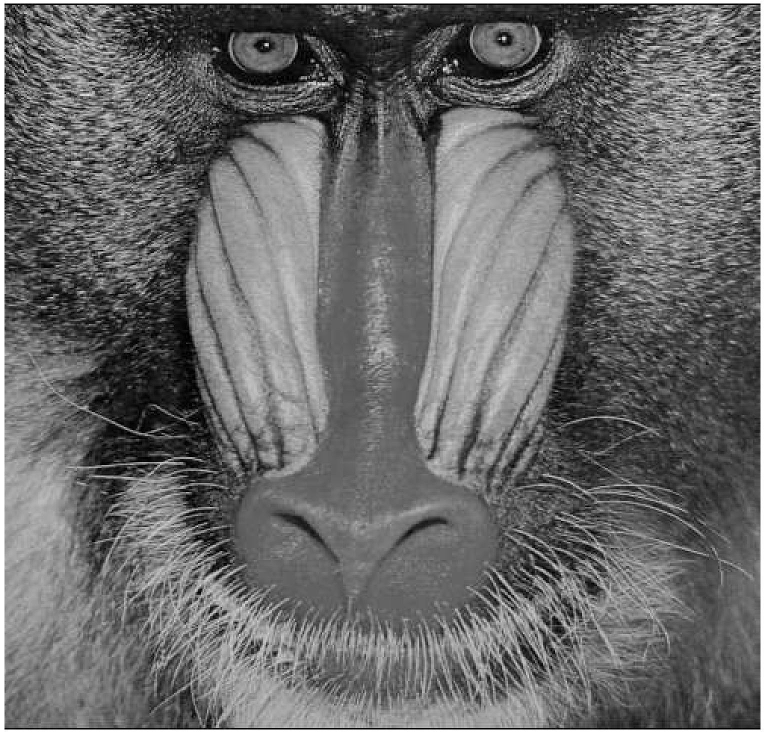


Abbildung 21: **Unser Mandrill, sideways-Umgebung**

```
\begin{figure}[h]
\begin{sideways}%
\mbox{}\hfill
\begin{minipage}[b]{0.5\textheight}
\epsfxsize=10cm
\begin{center}
\mbox{\epsfbox{mono6.ps}}
\end{center}
\vskip 0.5cm
\caption{\bf Unser Mandrill...}
\end{minipage}
\mbox{}\hfill\mbox{}
\begin{minipage}[b]{0.5\textheight}
... TEXT...
\end{minipage}
\end{sideways}
\end{figure}
```