

Technische Universität Ilmenau  
Fakultät für Mathematik  
und Naturwissenschaften  
Institut für Mathematik

Postfach 10 0565  
98684 Ilmenau  
Germany  
Tel.: 03677/692652  
Fax: 03677/691241  
Telex: 33 84 23 tuil d.  
email: W.Neundorf@mathematik.tu-ilmenau.de

Preprint No. M 7/96

*Mathematica* - PostScript -  
**TEX**

Werner Neundorf

April 1996

---

‡MSC (1991): 65-01, 65D17, 68U05

# 1 Einleitung

Graphische Komponenten erweisen sich als wirksame und moderne didaktische Werkzeuge. Sie findet man natürlich in Graphikprogrammen, in Computer Algebra Systemen, CAD-Programmen, Editoren, Textverarbeitungssystemen und vielen anderen Softwareprodukten.

Die Erstellung von Graphiken ist die eine Seite, aber die Einbeziehung dieser in unterschiedliche Systeme, ihre Anpassung an das dortige Layout kann durchaus zu einigen Problemen führen.

Ziel ist es hier, die mit dem CA-System *Mathematica* erstellte Graphiken in sinnvoller Weise in das Computer-Programm T<sub>E</sub>X im Rahmen des Setzens und Druckens von Texten, mathematischen Formeln und Bildern zu übernehmen, und dazu geeignete Mechanismen vorzustellen.

Bei L<sup>A</sup>T<sub>E</sub>X ist die Situation insofern etwas unbequem, da der Autor beim Schreiben des Eingabefiles in der Regel noch nicht sieht, wie der Text oder die Graphik nach dem Formatieren aussehen wird. Er kann aber durch Aufruf des entsprechenden Menüpunktes oder nach Erzeugung eines Postscript-Files jederzeit eine Probe-Ansicht/Ausdruck seines Dokuments machen und danach sein Eingabefile entsprechend korrigieren und die Arbeit fortsetzen.

Der Bedarf an Rechenzeit und Speicherplatz am Computer ist höher als bei primitiveren Textverarbeitungsprogrammen. Die Ausgabe der Schriftstücke sollte vorzugsweise an Laser-Printern, Graphik-Bildschirmen und Plottern, nicht aber an einfachen zeilenorientierten Druckern erfolgen.

Im Gegensatz dazu steht ein optischer Entwurf von Dokumenten/Notebooks mit Textverarbeitungsprogrammen wie z.B. WordStar, Works, WordPerfect, Excel, AmiPro, Scientific Word. In diesem Fall legt der Autor das Layout des Textes bei der interaktiven Eingabe fest. Dabei sieht er am Bildschirm genau das, was auch auf der gedruckten Seite stehen wird. Solche Systeme, die optische Entwürfe unterstützen, werden auch WYSIWYG-Systeme ("what you see is what you get") genannt.

## 2 Systeme, Konfigurationen, Programme

Folgende Hardwarekomponenten und Softwaretools wurden in die Betrachtungen einbezogen.

- **3 Varianten von *Mathematica***

- Mathematica* unter Windows auf PC

- Enhanced Version 2.2.1

- Enhanced Version 2.2.3

- Mathematica* Version 2.2.1 unter Unix auf Workstation (SUN)

- **T<sub>E</sub>X auf PC**

- TeXShell V2.6 mit L<sup>A</sup>T<sub>E</sub>X 2.09
- TeXShell V2.7.1 $\beta$  mit L<sup>A</sup>T<sub>E</sub>X 2e, L<sup>A</sup>T<sub>E</sub>X 2.09 (*Optionen*  $\Rightarrow$  *Formate*)

- **Drucker**

- EPSON LQ-400 24-Nadeldrucker
- EPSON Stylus 400, Tintenstrahldrucker  
ESC/P2 Postscript-Files werden i.a. an das Programm Ghostscript (gs) weitergereicht und dort mittels **GS Drucker** ausgegeben
- HP LaserJet 4M Laserdrucker  
besonders geeignet, insbesondere für PS-Files

### 3 Drucken in *Mathematica*

Hierbei soll insbesondere auf die Art und Weise der Ausgabe von graphischen Anteilen im Notebook eingegangen werden.

Folgende Graphik-Befehle bilden die Grundlage der Betrachtungen.

```
p1 = Plot[Sin[x^2], {x,-Pi,Pi}]
p2 = ParametricPlot[{4 Sin[t],Cos[t]}, {t,0,2Pi}]
p3 = ParametricPlot3D[{Sin[t],Cos[t],t/4}, {t,0,2Pi}]
p4 = ParametricPlot3D[{u Sin[t],u Cos[t],t/4}, {t,0,2Pi},{u,-1,1}]
p5 = Plot3D[Sin[x+Cos[y]], {x,-3,3},{y,-3,3}]
p6 = ContourPlot[Sin[x+Cos[y]], {x,-3,3},{y,-3,3}]
p7 = DensityPlot[Sin[x+Cos[y]], {x,-3,3},{y,-3,3}]
s10 = Plot3D[x y, {x,0,1},{y,0,1}]
```

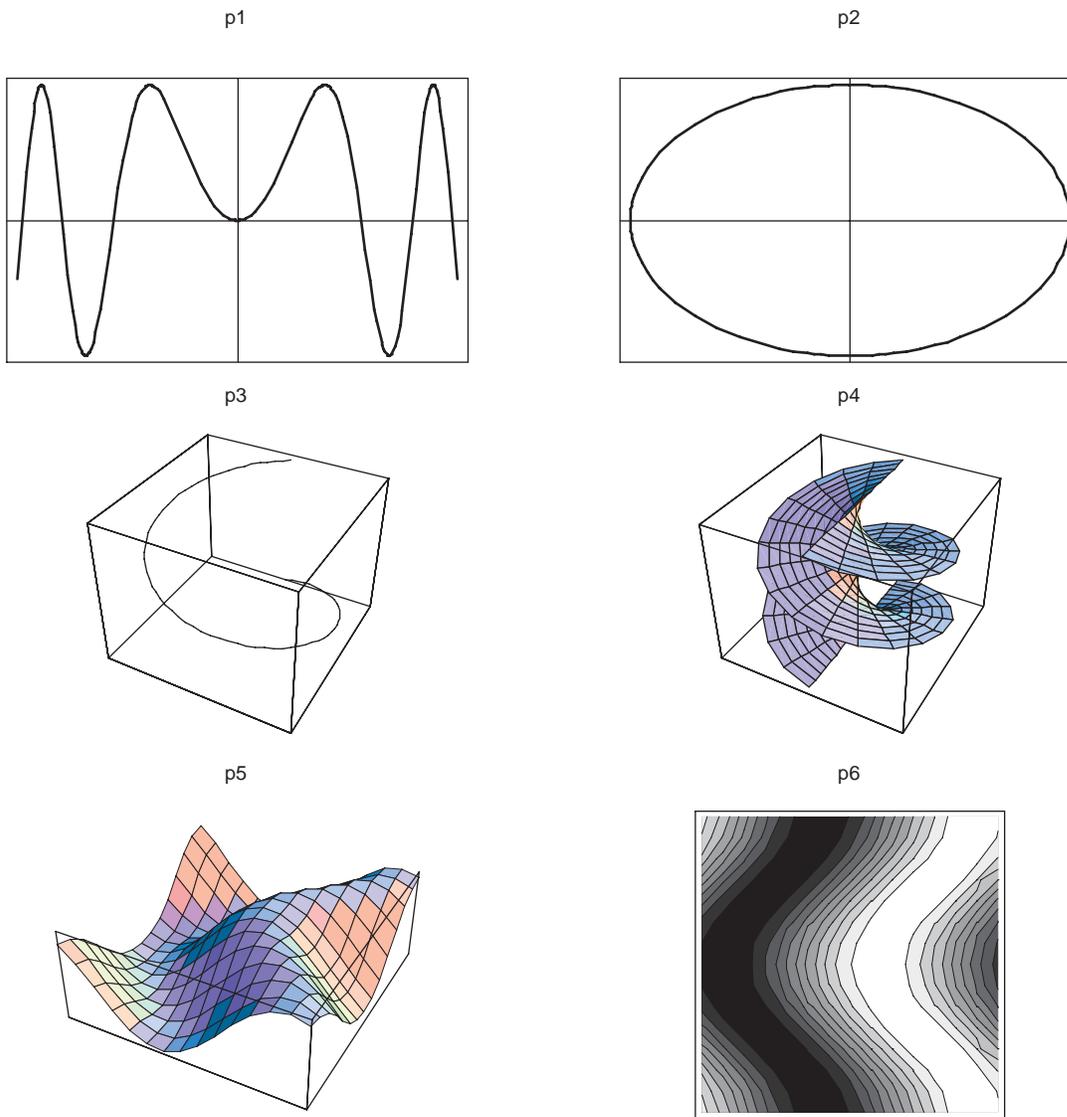
Die ersten 6 Bilder wurden dann nochmal als GraphicsArray *p* zusammengefaßt. Die Größe der Titel über den Bildern erfolgt mit dem Zeichenfont *Courier* = 4.

```
$DefaultFont = {"Courier",4}
p1 = Plot[Sin[x^2], {x,-Pi,Pi},
  Ticks -> None, Frame -> True, FrameTicks -> None,
  FrameLabel -> {"","","p1",""}]
$DefaultFont = {"Courier",4}
p2 = ParametricPlot[{4 Sin[t],Cos[t]}, {t,0,2Pi},
  Ticks -> None, Frame -> True, FrameTicks -> None,
  FrameLabel -> {"","","p2",""}]
p3 = ParametricPlot3D[{Sin[t],Cos[t],t/4}, {t,0,2Pi},
  Ticks -> None,
  PlotLabel -> FontForm["p3",{"Courier",4}]]
p4 = ParametricPlot3D[{u Sin[t],u Cos[t],t/4}, {t,0,2Pi},{u,-1,1},
  Ticks -> None,
  PlotLabel -> FontForm["p4",{"Courier",4}]]
p5 = Plot3D[Sin[x+Cos[y]], {x,-3,3},{y,-3,3},
  Ticks -> None,
  PlotLabel -> FontForm["p5",{"Courier",4}]]
```

```

p6 = ContourPlot[Sin[x+Cos[y]], {x,-3,3},{y,-3,3},
                FrameTicks -> None,
                PlotLabel -> FontForm["p6",{"Courier",4}]]
p = GraphicsArray[{{p1,p2},{p3,p4},{p5,p6}}]
Show[p]

```



**Abb.1.** Graphik-Tableau von 6 Funktionen,

    Datei `sk02_a64.eps` mit *Mathematica* V2.2.3  $\Rightarrow$  PS-File  $\Rightarrow$  T<sub>E</sub>X

Die T<sub>E</sub>X-Befehle zur Darstellung dieses Bildes unter Verdopplung des Originals der Dimension  $232.25 * 219.00$  pt und kleiner Verschiebung nach links sind

```

\epsfysize438.00pt      % 219.00pt ist Originalhoehe
\epsfbox[10 0 242.25 219.00]{sk02_a64.eps}

```

### 3.1 Druck-Format in *Mathematica* unter Windows

Nach Erzeugen der Graphik (Initialisierung des Notebooks) ist diese zu markieren und kann nun ausgegeben werden.

*File* ⇒ *Print...* ⇒ *Printer*

ev. *Setup*, um anderen Drucker auszuwählen

*PrintQuality*: ..., um geringere Auflösung (dpi) einzustellen

Dazu sollte man beachten, daß in *Mathematica* die Bildgröße voreingestellt ist und durch

*Options* ⇒ *Graphics...* ⇒ *Graphics Settings*

zu erkennen ist.

Diese Einstellung

*Default Graphic Size*

Width :	<input type="text" value="300"/>	pixels	Height :	<input type="text" value="300"/>	pixels
X	<input type="text" value="100"/>	pixels	Y	<input type="text" value="0"/>	pixels

heißt, daß die gedruckte Graphik auf der A4-Seite ab der aktuellen Druckposition horizontal zentriert erscheint, wobei ihre Breite ungefähr die halbe Seitenbreite ausmacht und ihre Höhe i.a. etwas kleiner als die Breite ist.

Diese Einstellungen befinden sich auch in der INI-Datei von *Mathematica*

**MATH22.INI** und heißen dort

```
;      Default Graphics Settings, in Pixels
defgraphx=100
defgraphy=0
defgraphw=300
defgraphh=300
```

Für die verschiedenen Versionen von *Mathematica* werden beim Druck trotz einheitlichen *Default Graphics Settings* nicht gleichgroße Bilder erzeugt.

Für die **Version 2.2.1** ist bei 300\*300 Pixel das Fenster maximal 80\*80mm groß. Veränderte Graphiken, z.B. nach Zoomen, sollen hier nicht weiter untersucht werden.

Drucker	ungefähre Breite*Höhe des Bildes in mm				
	p1, p2 Graphics	p3, p4 Graphics3D	p5, s10 SurfaceGr.	p6, p7 ContourGr. DensityGr.	p Gr.Array
EPSON LQ-400	65*40	65*65	65*50	65*65	
EPSON Stylus 400	80*50	80*80	80*65	80*80	
HP LaserJet 4M	80*50	80*80	80*65	80*80	75*75

**Tab.1.** Druckgröße der Originalgraphiken  $p_1, \dots, p_7, p, s_{10}$  mit *Mathematica* V2.2.1

### Achtung:

Der Druck mittels HP LaserJet 4M kann zu einigen Komplikationen führen.

1. Falls die zu druckende Datenmenge sehr umfangreich ist (viele Graphiken), erscheinen manche Teile nicht in der Druckausgabe. Man versuche, die Auflösung (dpi, Quality) zu verkleinern.
2. Sobald im markierten Notebookteil eine Graphik dabei ist, müßte sich der Drucker automatisch auf HP LaserJet 4M Postscript einrichten. Das ist hier per Hand vorzunehmen.  
Außerdem muß eine Abstimmung mit dem Windows PS Header erfolgen. Sonst unterbleibt die Ausgabe und man erhält nur die Nachricht „Windows PS Header has not been downloaded. Please download the header.“
3. Wird das Notebook oder das evaluierte Notebook (mit Graphiken) ausgegeben, erscheint eine zusätzliche Kopfzeile mit dem Filenamen und der Seitennummer sowie eventuell auch eine Fußzeile mit horizontaler Linie. Gibt man nur die markierte Graphik aus, dann entfallen diese Zusätze.

Für die **Version 2.2.3** erhält man folgende Ausgabefenster.

Drucker	ungefähre Breite*Höhe des Bildes in <i>mm</i>				
	p1, p2	p3, p4	p5,s10	p6, p7	p
alle 3 genannten Drucker	65*40	65*65	65*50	65*65	60*60

**Tab.2.** Druckgröße der Originalgraphiken  $p_1, \dots, p_7, p, s_{10}$  mit *Mathematica V2.2.3*

### Achtung:

1. Sobald eine Graphik auszugeben ist, wird der Drucker HP LaserJet 4M Postscript sinnvollerweise automatisch ausgewählt.  
Stellt man selbst auf den Drucker HP LaserJet 4M (**ohne** Postscript) um, ist dies i.a. problemlos für die Ausgabe. Nur bei umfangreichen Dokumenten kann es dann wieder passieren, daß Graphiken nur teilweise erscheinen.
2. Egal ob Notebook, evaluiertes Notebook oder markierte Zelle oder Graphik, die Ausgabe enthält stets eine Kopfzeile mit horizontaler Linie und Filenamen sowie eine Fußzeile ebenfalls mit horizontaler Linie und Seitennummer.

## **3.2 Ausgabe in *Mathematica V2.2.1* unter Unix**

- Die Druckerausgabe von erzeugten Graphiken erfolgt nur über den Umweg von Postscript-Files.  
Dazu markieren der Graphik oder Zelle, dann  
Menü *File*  $\Rightarrow$  *Print Selection...*  
**Print**-Dialogfenster : *Print...*

- Druck des Postscript-Files mittels postscriptfähigen Drucker HP LaserJet 4M  
*Programs ⇒ Print Tool... ⇒ Printer ⇒ Postscript*

Die Ausgabefenster sind etwas größer als bei der Windows-Version.

Drucker	ungefähre Breite*Höhe des Bildes in mm				
	p1, p2	p3, p4	p5, s10	p6, p7	p
HP LaserJet 4M Postscript	100*65	100*100	100*80	100*100	95*95

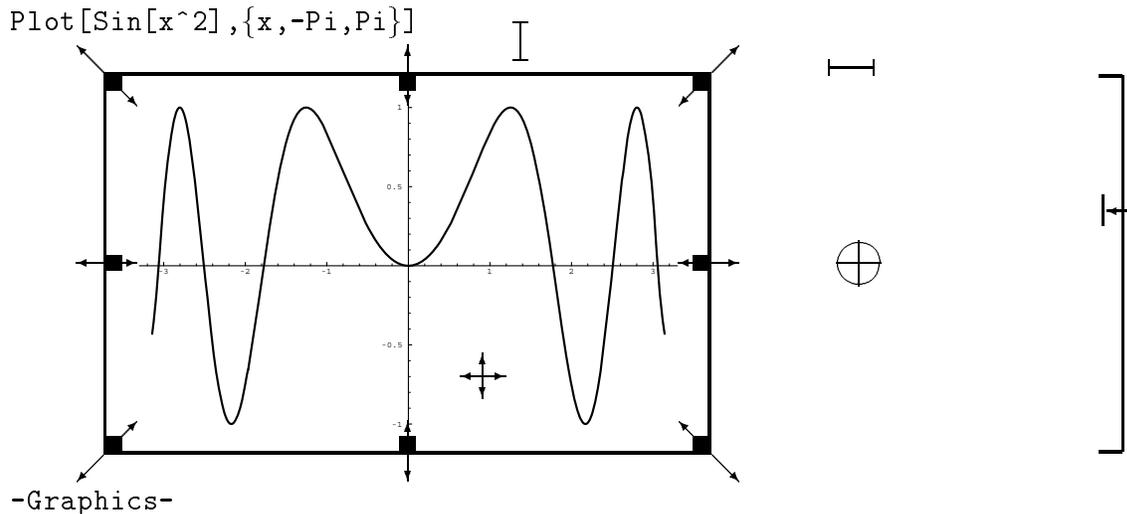
**Tab.3.** Druckgröße der Originalgraphiken  $p1, \dots, p7, p, s10$  mit V2.2.1 unter Unix

**Achtung:**

Das PS-File enthält in jedem Fall eine Kopfzeile mit horizontaler Linie sowie Filenamen links und Seitennummer rechts.

## 4 Erzeugen von Postscript-Files in *Mathematica*

Zunächst gibt es zwischen den beiden Versionen einen wichtigen Unterschied. Die V2.2.1 orientiert in der Struktur dieser Files auf Graphiken, die auf ganzen A4-Seiten untergebracht werden. Hingegen wird bei der V2.2.3 für eine Graphik nur der Platz auf der Druckseite vorgesehen, der unter Berücksichtigung eines kleinen Randes wirklich gebraucht wird. Des weiteren finden bei den Windows-Varianten die bei Direktdruck in *Mathematica* erscheinenden Kopf- und Fußzeilen im PS-File **keine** Berücksichtigung. In jedem Fall ist die Graphik zu markieren (anklicken). Ihr Zoomen hat auf das spätere Postscript keinen Einfluß, wohl aber die eventuell vorher durchgeführten Einstellungen in *Graphics Settings*.



**Abb.2.** Markieren einer Graphik im Notebook mit Funktion  $p1$ , Symbolik des Mauszeigers bei Markieren und Zoomen

Zum Verständnis nur kurz eine Erläuterung zur Erstellung der Abbildung 2, da im folgenden alles genauer beschrieben wird.

Das PS-File `zi02_1.ps` zum Graphen  $p_1$  wurde mit *Mathematica* V2.2.1 erzeugt. Es hat die ersten Zeilen

```

%!PS-Adobe-2.0 EPSF-2.0
%%BoundingBox: 0.000 0.000 612.000 792.000
%%Creator: Mathematica
%%CreationDate:
%%EndComments
/Mathdict 150 dict def
Mathdict begin
/Mlmarg          0.000 72 mul def
/Mrmarg          0.000 72 mul def
/Mbmarg          0.000 72 mul def
/Mtmarg          0.000 72 mul def
/Mwidth          8.500 72 mul def
/Mheight         11.000 72 mul def
.....

```

Der erfaßte Bereich (BoundingBox) ist sehr groß. Er hat die Dimensionen

$$\text{Breite} \times \text{Höhe} = 612 * 792 \text{ pt} \approx 8.5 * 11.0 \text{ in} \approx 215 * 279 \text{ mm}.$$

Darin liegt ungefähr vertikal zentriert die eigentliche Graphik mit der Ausdehnung  $612 * 420 \text{ pt}$  (Achtung! Graph  $\gg 80 * 50 \text{ mm}$  bei Druck in *Mathematica* V2.2.1). Diese gilt es durch eine Box zu erfassen und auf eine Höhe von  $140 \text{ pt} \approx 50 \text{ mm}$  zu stauchen, damit sie dann noch nach rechts verschoben in die Framebox einer Picture-Umgebung mit `\framebox(80.00,50.00)` eingebettet werden kann.

Dazu die ausgewählten Befehle der Input-Datei `math1.pic` .

```

% MATH1.PIC
\unitlength=1.00mm
\special{em:linewidth 0.4pt}
\linethickness{0.4pt}
\begin{picture}(165.00,67.00)
\put(25.00,10.00){\framebox(80.00,50.00)[cc]{} }
\put(24.83, 9.83){\framebox(80.33,50.33)[cc]{} }
.....
\epsfysize140pt      % Hoehe 420=520-100 auf 140 gestaucht
\epsfbox[-250 100 500 520]{zi02_1.ps}
%
%moeglich ist auch :
%\put(44.00,25.00)
%   {\makebox(0,0)[l]{\epsfbox[-250 100 500 520]{zi02_1.ps}}}
\end{picture}

```

## 4.1 Die Vorgehensweisen

Für die 3 Varianten von *Mathematica* gibt es voneinander abweichende Wege mit einigen Besonderheiten.

- **V2.2.1 (Windows)**

1. Mit Markieren der Graphik erscheint im Fenster Formatvorlage 

Postscript
------------
2. Anwählen : *File* ⇒ *Save As/Export...*
3. Auswählen im Dialogfenster **Save As/Export**

File Name :	<input type="text"/>		
File Type :	<table border="1"><tr><td>Enc. PostScript (*.eps)</td></tr><tr><td>PostScript (*.ps)</td></tr></table>	Enc. PostScript (*.eps)	PostScript (*.ps)
Enc. PostScript (*.eps)			
PostScript (*.ps)			

Eintragen des Namens, des Verzeichnisses und dergleichen, sowie eines der beiden PS-File-Typen, dann <OK>-Taste.

4. Gegebenenfalls Entfernen von nicht ASCII-Code am Anfang und Ende des \*.eps-Files, d.h.

vor     %!PS-Adobe-2.0 EPSF-2.0  
nach    showpage

Damit sind aber das \*.eps-File und \*.ps-File identisch (was die Frage der Sinnfälligkeit einer Unterscheidung aufwirft).

- **V2.2.3 (Windows)**

1. Mit Markieren der Graphik erscheint im Fenster Formatvorlage 

Postscript
------------
2. Anwählen : *File* ⇒ *Export...*
3. Auswählen im Dialogfenster **Export**

File Name :	<input type="text"/>	
File Type :	<table border="1"><tr><td>Encapsulated Postscript File (*.eps)</td></tr></table>	Encapsulated Postscript File (*.eps)
Encapsulated Postscript File (*.eps)		

Eintragen des Namens, des Verzeichnisses und dergleichen, dann <OK>.

- **V2.2.1 (Unix)**

1. Mit Markieren der Graphik
2. Anwählen : *File* ⇒ *Print Selection...*
3. Auswählen im Print-Dialogfenster : *Print Destination* ⇒ *File...(\*.ps)*  
Eintragen des Namens, des Verzeichnisses und dergleichen, dann <OK>.
4. Zwecks Kontrolle sollte man sich das PS-File \*.ps anzeigen lassen. Mit seiner Auswahl (Ikone) wird automatisch ein PS-File-Betrachter gestartet :
  - (a) auf SUN *Newton* : **PAGE VIEW V3.0**
  - (b) auf SUN *Gauss* : **Image Tool V3.4**

Unter **View** erhält man zum File die Image Information der Form

Height	:	11,0 in	/	27,9 cm	/	913 pix
Width	:	8,5 in	/	21,6 cm	/	705 pix
Colors	:	N/A				
File Size	:	... K Bytes				
File Format	:	PostScript				

Man erkennt, daß die Basis des PS-Dokuments die A4-Seite (210 \* 297mm) ist. Ist das Dokument länger, kann man vor und zurückblättern. Jede Seite beginnt mit einer Kopfzeile.

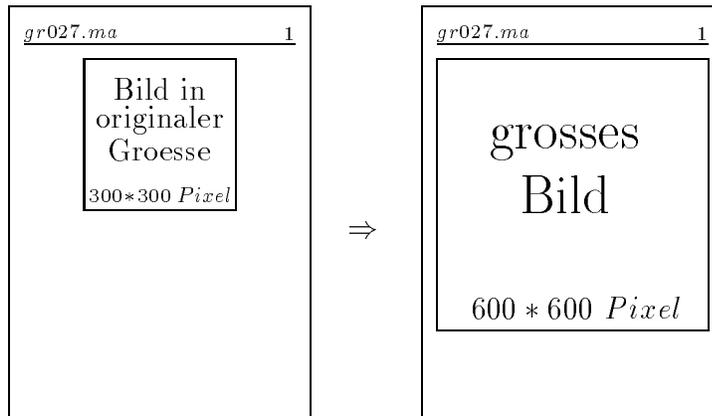
**Achtung:**

Bezieht man das PS-File in T<sub>E</sub>X ein, so wird dieser Seitenwechsel jedoch nicht wirksam. Beim Ansehen eines mehrseitigen Dokuments kommen alle Seiten übereinander zu liegen. Somit sollte man unter Unix umfangreiche Datenmengen und Graphiken seitenweise in mehrere PS-Files zerlegen.

Außerdem kann die über die ganze Seite verlaufende Kopfzeile stören, wenn die Graphik nur halb so breit ist. Beim Einbinden des Files in T<sub>E</sub>X und der Vergrößerung/Verschiebung der Graphik, geht die Kopfzeile dann über den Rand hinaus. In einem solchen Fall hat man zwei Möglichkeiten.

- (a) Graphik in ihrer Breite nach der Kopfzeile ausrichten.

Das geschieht mittels *Style* ⇒ *Edit Styles ...*. Im Info/Dialogfenster **Styles for ...** ist das Graphik-Format **Default Graphics Size** auf gewünschte Größe zu zoomen (auf ca. das Doppelte verbreitern, um auf Kopfzeilenlänge zu kommen). Anschließend dort *File* ⇒ *Close* ausführen. Nunmehr wird die neue Größe sowohl im Notebook gezeigt als auch im PS-File berücksichtigt.



**Abb.3.** Vergrößerung des Graphen bei Kopfzeile (V2.2.1. Unix)

- (b) Die Kopfzeile ganz unterdrücken durch Anwählen von

*File* ⇒ *Printing Settings* ⇒ *Headers and Footers* ⇒ *Setting Up the Starting Page*  **No headers or footers on first page**

## 4.2 Zur Struktur der PS-Files

Die 3 Varianten liefern abweichende Formen ihrer Postscript-Dateien. Das bezieht sich sowohl auf den Inhalt als auch auf die Dateigröße.

Zunächst geben wir die ersten Zeilen der jeweiligen PS-Files anhand des Beispiels der Funktionsgraphik *p1* an.

- *Mathematica* V2.2.3 Windows : `sk02_1.eps`

```
%!PS-Adobe-3.0 EPSF-3.0
%%Creator: ImageMark Software Labs
%%For: () ()
%%Title: O:\STUD_M93\VERSCHIE\PS\EPS_TEX\SK02_1.EPS
%%CreationDate:
%%BoundingBox: 0.0000 0.0000 276.2500 170.2500
%%DocumentProcessColors: Black
%%DocumentFonts: Helvetica
.....
```

- *Mathematica* V2.2.1 Windows : `zi02_1.ps`  $\equiv$  `zi02_1.eps` (nach „Säubern“)

```
%!PS-Adobe-2.0 EPSF-2.0
%%BoundingBox: 0.000 0.000 612.000 792.000
%%Creator: Mathematica
%%CreationDate:
%%EndComments
/Mathdict 150 dict def
Mathdict begin
/Mlmarg          0.000 72 mul def
/Mrmarg          0.000 72 mul def
/Mbmarg          0.000 72 mul def
/Mtmarg          0.000 72 mul def
/Mwidth         8.500 72 mul def
/Mheight        11.000 72 mul def
.....
```

- *Mathematica* V2.2.1 Unix : `ws02_1.ps` (mit Kopfzeile)  
                          `wg02_1.ps` (ohne Kopfzeile)

```
%!PS-Adobe-2.0
%%Title: /home/imath/neundorf/mathematica/grafik/gr023.ma
%%Creator: Mathematica
%%BoundingBox: 0 0 612 792
%%EndComments
.....
```

Des weiteren zeichnen sich die 3 *Mathematica*-Varianten durch unterschiedlichen Speicherbedarf der entsprechenden PS-Dateien aus.

Graphik n=1,...,6	Windows		Unix	
	V 2.2.3	V 2.2.1	V 2.2.1	
	sk02_n.eps	zi02_n.ps	ws02_n.ps	wg02_n.ps
p1	35137	21416	33171	31712
p2	31693	19870	31625	30166
p3	35372	26352	38110	36651
p4	260114	87228	98986	97527
p5	115107	36147	47905	46446
p6	70209	39152	50898	49439
p	sk02_a64.eps 396403	zi02_a64.ps 130642	ws02_a64.ps 141800	wg02_a64.ps 140340

**Tab.4.** Speicherbedarf in Byte der PS-Files von  $p1, \dots, p6, p$

Als nächstes fassen wir die in den PS-File und ihren Eigenschaften angegebenen Bereichsgrößen für die Graphiken (BoundingBox) und die Bildgrößen zusammen. Der Umrechnung zwischen den Maßstäben ist  $1'' = 1\text{ in} = 25.4\text{ mm} = 72.27\text{ pt}$ .

Graphik n=1,...,6	Windows		Unix
	V 2.2.3 sk02_n.eps	V 2.2.1 zi02_n.ps	V 2.2.1 ws02_n.ps
	eingebettet in Bereich $612 * 792\text{ pt}$ $\approx 8.5 * 11.0\text{ in} \approx 215 * 279\text{ mm}$		
	$pt$	$pt$	$Pixel$
p1, p2	$276.25 * 170.25$ $\approx 97 * 86\text{ mm}$	$612 * 377$	$300 * 185$
p3, p4	$276.25 * 274.25$ $\approx 97 * 96\text{ mm}$	$612 * 607$	$300 * 298$
p5	$276.25 * 226.25$ $\approx 97 * 80\text{ mm}$	$612 * 501$	$300 * 246$
p6	$276.25 * 226.25$ $\approx 97 * 97\text{ mm}$	$612 * 612$	$300 * 300$
p	sk02_a64.eps $232.25 * 219.00$ $\approx 82 * 77\text{ mm}$	zi02_a64.ps $612 * 572$	ws02_a64.ps $252 * 238$

**Tab.5.** Bildgrößen Breite\*Höhe von Graphiken  $p1, \dots, p6, p$

Die Pixeleinstellung für die Graphikgröße (siehe Kap. 3.1) wirkt sich aus auf die Bereichsgrenzen der Box im PS-File (Kasten, BoundingBox).

<i>Mathematica</i> Graphics Settings	PS-File BoundingBox
Width*Heighth	
<i>Pixel</i>	<i>pt</i>
300 * 300	232.25 * 143.00
400 * 400	309.75 * 191.25
500 * 500	387.25 * 239.00
600 * 600	464.75 * 287.25
Differenz $\approx$	77.5    48.0

**Tab.6.** Pixelkoordinaten und BoundingBox für ausgewählte Graphik in *Mathematica* V2.2.3 Windows

Aus der Gegenüberstellung folgt, daß für die betrachtete Variante eine Pixeleinheit in der Breite den Wert  $0.775\text{pt}$  bzw. in der Höhe  $0.480\text{pt}$  hat.

### 4.3 Zusammenfassung

Abweichend von den Abmessungen der Graphik bei ihrem Ausdruck in *Mathematica* haben wir es beim Einbinden von PS-Files in  $\text{\TeX}$  prinzipiell mit **drei** Situationen zu tun.

PS-Files `*.eps` von V2.2.3 Windows lassen sich am bequemsten handhaben. Die Dimensionierung des Bildes ist direkt zu erkennen. Die Graphik kann in der gegebenen Größe schon benutzt werden, aber man könnte sie auch nach Wunsch geeignet verzerren und positionieren.

Schwieriger ist es mit den PS-Files `*.ps` von V2.2.1 Windows. Allein schon die Ausdehnung des Arbeitsblatts (Kasten) von  $612 * 792\text{pt} \approx 215 * 279\text{mm}$  weist darauf hin, daß die originale Graphik i.a. zu groß ausfällt und entsprechend verkleinert und verschoben werden muß.

Trotz äußerlicher Ähnlichkeiten der Kopfzeilen der PS-Dateien ist die Situation wiederum eine andere für die Variante V2.2.1 Unix.

## 5 Einbinden der Postscript-Files in T<sub>E</sub>X

### 5.1 Beispiel

Zu Beginn geben wir als Demonstration die Einbeziehung von PS-Files für die Abbildung 1 an. Dazu benutzen wir im Dokumentstil die Optionen (optionale Parameter) `epsf` bzw. `psfig`.

```
\documentstyle[12pt,german,emlines,amsfonts,epsf,psfig,dina4]{article}
```

Zum Graphiktableau *p* mit den 6 Funktionsgraphen wurde mit V2.2.3 Windows das PS-File `sk02_a64.eps` erzeugt. Dieses hat die Bildgrößen  $232.25 * 219.00 \text{ pt} \approx 82 * 77 \text{ mm}$ . Bei Verdopplung und kleiner Linksverschiebung paßt es, wie erkennbar, noch gut in den Text auf einer A4-Seite.

Die T<sub>E</sub>X-Befehle dazu sind

```
\epsfysize438.00pt    % 219.00pt normal -> verdoppeln
\epsfbox[10 0 242.25 219.00]{sk02_a64.eps}
```

Bei Anwendung der **Version 2.2.1 unter Windows** mit dem PS-File `zi02_a6f.ps` (der Zeichenfont der Titel wurde wegen der anschließend notwendigen Verkleinerung des Bildes gesetzt auf *Courier* = 12 ) sind die entsprechenden Kommandozeilen

```
% Originalgraphik der Groesse 612*572pt
%      im Bereich 612*792pt = 215*279mm
% Leerraum von 110pt (Hoehe) vor und nach der Gaphik
% Graph geht ueber rechten Rand der A4-Seite hinaus
\epsfbox[0 0 612 792]{zi02_a6f.ps}
%
% Leerraum ueber und unter Graphen entfernt
\epsfbox[0 110 612 682]{zi02_a6f.ps}
%
% Bereich der Hoehe 612pt=702-90 auf 438pt=154mm gestaucht
% damit Graph insgesamt auf etwa 2/3 verkleinert
\epsfysize438pt
\epsfbox[10 90 622 702]{zi02_a6f.ps}
```

Die letzte Graphik ist ähnlich zu Abbildung 1.

Die **Version 2.2.1 unter Unix** liefert die PS-Files

```
wg02_a66.ps  ohne Kopfzeile
ws02_a64.ps  mit Kopfzeile.
```

Die Beschriftung durch Titel wurde im ersten Fall etwas vergrößert (*Courier* = 6). Man bemerke jedoch, daß ein Eintrag von Überschriften bzw. die Vergrößerung dieser zu einer entsprechenden Verkleinerung des dazugehörigen Funktionsgraphen führt, sowie insgesamt das Graphiktableau kleiner wird.

```

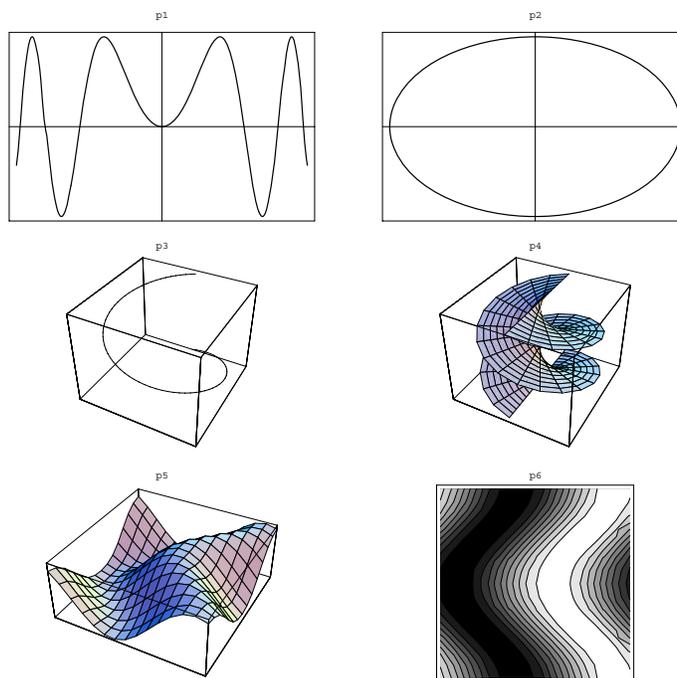
% wg02_a66.ps ohne Header und Footer
% Originalgraphik der Groesse ca 215*215pt
%      im Bereich der Ausdehnung 612*215pt = 215*75mm
\epsfbox[50 502 662 717]{wg02_a66.ps}
%
% Bereich der Hoehe 215pt=717-502 auf 438pt=154mm gestreckt
% damit Graph insgesamt etwa auf das 2-fache vergroessert
\epsfysize438pt
\epsfbox[100 502 712 717]{wg02_a66.ps}
% -----
% ws02_a64.ps mit Header
% Originalgraphik der Groesse ca 285*285pt
%      im Bereich der Ausdehnung 612*360pt = 215*126mm
% Verschiebung nach links
% wegen Kopfzeile keine Vergroesserung
\epsfbox[50 432 662 792]{ws02_a64.ps}

```

Die auf A4-Seite skalierten Abbildungen von `sk02_a64`, `zi02_a6f`, `wg02_a66` unterscheiden sich also nicht wesentlich. Deshalb sei zum Vergleich nur die Darstellung von `ws02_a64.ps` mit der zu breiten Kopfzeile angegeben.

*gr027.ma*

1



**Abb.4.** Graphik zum PS-File `ws02_a64.ps` (V2.2.1 Unix)

## 5.2 Verzerrung und Positionierung von PS-File-Graphiken

Zunächst sollte der Nutzer dazu die entsprechenden Dokumentationen und Demonstrationsprogramme studieren. Informationen findet man auch in den Dateien

**EPSF.TEX**, **EPSF.STY**, **PSFIG.STY** der T<sub>E</sub>X-Installation auf dem Rechner in den Verzeichnissen

```

LATEX 2.09 : \EMTEX\TEXINPUT
bzw.      LATEX 2e : \EMTEX\TEXINPUT\LATEX209
                    \EMTEX\TEXINPUT\DVIPS
                    \EMTEX\PS
```

Weiterhin sind gewisse Parallelen zu folgenden T<sub>E</sub>X-Strukturen zu erkennen.

### (a) Erweiterte Syntax der Picture-Umgebung

Hauptbild mit Unterbild(ern)

```
\begin{picture}(x_dimen,y_dimen)(x_offset,y_offset) bildbefehle
\end{picture}
```

In dieser Form wirkt die Picture-Umgebung so, als würden bei allen `\put`-Befehlen die Größen `x_offset` und `y_offset` von den Koordinatenangaben `x_coord` bzw. `y_coord` in *bildbefehle* subtrahiert. Die Gesamtwirkung liegt also darin, daß das ganze Unterbild um `x_offset` nach links (falls `x_offset > 0`) und `y_offset` nach unten (falls `y_offset > 0`) verschoben erscheint.

Der Fall, daß das Teilbild teilweise oder ganz außerhalb der Hauptbildgrenzen liegt, muß bei der Offset-Angabe bedacht werden. Zusätzlich kann eine Streckung mit Veränderung von `\unitlength` erfolgen.

### (b) Plazierung von Gleitobjekten

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

T<sub>E</sub>X bietet die Möglichkeit, Bilder und ähnliches, einschließlich ihrer Über- oder Unterschriften, in der beschriebenen Form gleiten zu lassen. dabei bedeutet *bild* eine Bilddefinition, erzeugt durch die Picture-Umgebung.

### (c) Absatzboxen und Teilseiten

```
\parbox[pos]{breite}[wohin] text
bzw. \begin{minipage}[pos]{breite} text
\end{minipage}
```

Beide erzeugen vertikale Boxen der Breite *breite*, mit denen die Zeilen in der entsprechenden Breite untereinander angeordnet sind.

Insgesamt sind die Beschreibungen dazu jedoch unzureichend, zum Teil verwirrend, so daß detaillierte Angaben sinnvoll und notwendig erscheinen. Im konkreten Fall ist manches noch gezielt auszutesten.

### 5.3 Anwendung des Style epsf.sty

Wir benutzen als erstes das **eps**-Format und binden das Bild ins  $\LaTeX$ -Dokument mittels `epsfbox{name.eps}` ein.

Im weiteren werden die Angaben zur Dimensionierung wahlweise in *pt* oder *mm* gemacht. Es gilt bei gerundeter Darstellung

$$\begin{aligned}
 1'' &= 1 \text{ in} &= 25.4 \text{ mm} &= 72.27 \text{ pt} \\
 1 \text{ mm} &= 2.85 \text{ pt} \\
 1 \text{ pt} &= 0.35 \text{ mm} \\
 \text{A4-Seite} &= 210 * 297 \text{ mm} = 8.3 * 11.7 \text{ in}
 \end{aligned}$$

#### 5.3.1 Mathematica V2.2.3 unter Windows (PS-File \*.eps)

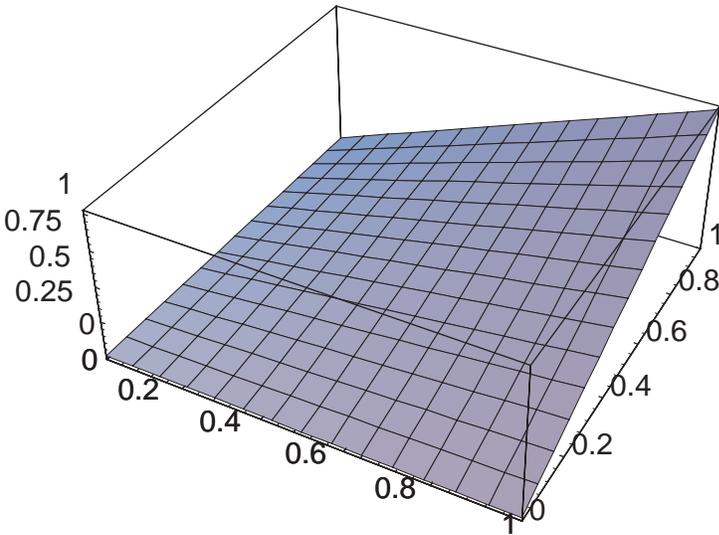
Im Kopf der Datei sind die Größen der Ausdehnung Breite\*Höhe der Graphik ersichtlich. Die Angaben dort sind in *pt* gemacht. Diesen Maßstab wollen wir möglichst beibehalten.

Zur Erläuterung der Kommandozeilen der Einbindung und deren Ergebnis beziehen wir uns auf die Graphik Plot3D *s10*,  $f(x,y) = x * y$ ,  $x, y \in [0, 1]$ , mit dem

PS-File `tbs10.eps` der Größe  $276.25 * 226.25 \text{ pt} \approx 97 * 79 \text{ mm}$   
 (Größe analog zu Graphik *p5*, PS-File `sk02_5.eps`).

Sie wird in originaler Größe linksbündig und unterhalb des oberen Kastenrandes (Positionierungspunkt) eingetragen. Zur Anschaulichkeit wurden zwei horizontale Begrenzungslinien mitgezeichnet.

1 ..... 1



2 ..... 2

**Abb.5.** Graphik in Originalgröße zum PS-File `tbs10.eps` (V2.2.3 Windows)

Dazu die TeX-Datei.

```
% EPSF1.TEX
% Test von Plot3D s10
\documentstyle[12pt,german,emlines,amsfonts,epsf,psfig,dina4]{article}
\newcommand{\eins}{1\dotfill 1\}
\newcommand{\zwei}{2\dotfill 2\}
\begin{document}
\noindent
Bilder mit Style {\bf epsf}\}
Test von Plot3D TBS10.EPS : 276.25*226.25pt = 97.1*79.5mm,
Mathematica 2.2.3 unter Windows\}[1em]
\eins
% epsfbox[0 0 276.25 226.25]{TBS10.EPS}\}
\epsfbox{TBS10.EPS}\}
\zwei
--- Abbildung Plot3D TBS10.EPS im Original ---
\end{document}
```

In der Datei **EPSF.TEX** findet der Nutzer eine Beschreibung zur Einbindung, und wir notieren daraus die für uns zutreffenden wesentlichen Anteile.

TeX macros to include an Encapsulated PostScript graphic.  
Works by finding the bounding box comment, calculating the correct scale values, and inserting a vbox of the appropriate size at the current position in the TeX document.  
To use with the center environment of LaTeX (LaTeX should probably supply this itself for the center environment).

To use, simply say  
`\documentstyle[... ,epsf,...]{...}`  
% .....  
% where you want to insert a vbox for a figure  
`\epsfbox{filename.eps}`

Alternatively, you can type for supplying your own BoundingBox  
`\epsfbox[0 0 30 50]{filename.eps}`  
which will not read in the file, and will instead use the bounding box you specify.  
The effect will be to typeset the figure as a TeX box, at the point of your `\epsfbox` command. By default, the graphic will have its 'natural' width (namely the width of its bounding box, as described in filename.eps). The TeX box will have depth zero.

You can enlarge or reduce the figure by saying  
`\epsfxsize=<dimen> \epsfbox{filename.eps}`  
or  
`\epsfysize=<dimen> \epsfbox{filename.eps}`  
Then the width of the TeX box will be `\epsfxsize` and its height will be scaled proportionately  
or the height will be `\epsfysize` and its width will be scaled proportionately.  
The width (and height) is restored to zero after each use.  
Common uses include:  
- Natural scaling means, that 0 is to be used.  
- `\epsfxsize` just leave the old value alone.

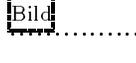
Zur Vollständigkeit und Richtigkeit muß folgendes bemerkt werden:

1. Gibt man bei TeXShell V2.6 in der Skalierung beide Parameter `\epsfxsize` und `\epsfysize` an, so wird nur die Breitenskalierung genommen. Diese hat Priorität und die Höhe wird proportional skaliert.
2. Im Gegensatz zu TeXShell V2.6 erlaubt die Version 2.7.1 $\beta$  eine unabhängige Skalierung beider Richtungen.
3. Die Bereichsangaben in der Box `[x0 y0 x1 y1]` erfolgen generell im Maßstab *pt*.
4. Die Skalierungsangaben können wahlweise in *pt* oder *mm* erfolgen. Z.B. sind folgende Notationen erlaubt

```
\epsfxsize276.25pt, \epsfxsize=276.25pt
\epsfxsize97.1mm, \epsfxsize=97.1mm
\epsfxsize0pt, \epsfxsize0mm (Originalgröße)
\epsfxsize (vorhergehenden Wert übernehmen)
```

### Varianten der Bildmanipulation

- Verzerrung eines Bildes  
Originalbildbreite= $w$ , -bildhöhe= $h$ , neue Breite= $w'$ , neue Höhe= $h'$

Boxbefehle	Verzerrung des Bildes	Bemerkung
<pre>\epsfbox[0 0 w h]{*.eps} \epsfbox{*.eps} \epsfxsize w mm \epsfysize h mm \epsfbox[0 0 w h]{*.eps}</pre>		Original, $w \geq h$
<pre>\epsfysize h' mm \epsfbox{*.eps}</pre>		Bildhöhe = $h'$ mm
<pre>\epsfxsize w' mm \epsfbox{*.eps}</pre>		Bildweite = $w'$ mm Situation bei $w' = h'$
<pre>\epsfysize h' mm \epsfbox{*.eps}</pre>		Bild auf Seitenbreite $h'=120$ mm ( $w' \approx 150$ mm)
<pre>\epsfxsize w' mm \epsfysize h' mm \epsfbox{*.eps}</pre>		Bild in $x, y$ verzerrt $w'=50$ mm, $h'=90$ mm

**Tab.7.** Auswahl von Bildverzerrungen

- Verschiebung eines Bildes  
 Bildbreite= $w$ , Bildhöhe= $h$ ,  $w' > 0$ ,  $h' > 0$

Boxbefehl <code>\epsfbox[l u r o]{*.eps}</code>	Verschiebung des Bildes	Bemerkung
<code>[ 0 0 w h ]</code>		Original
<code>[ 0 0 w h+h' ]</code>		über dem Bild Platz machen
<code>[ 0 0 w h-h' ]</code>		Bild schiebt sich auf vorherigen Text/Zeichnung
<code>[ w' 0 w+w' h ]</code> <code>[ w' 0 w h ]</code>		
<code>[ -w' 0 w-w' h ]</code>		Platz links neben dem Bild
<code>[ 0 -h' w h ]</code>		unter dem Bild Platz machen
<code>[ 0 h' w h ]</code>		nachfolgenden Text/Zeichnung ins Bild hineinziehen
<code>[ 0 h' w h+h' ]</code>		über dem Bild Platz machen, nachfolgenden Text/Zeichnung ins Bild hineinziehen

**Tab.8.** Auswahl von Bildverschiebungen

- Tableau mit 4 gleichartigen Bildern  
 Verkleinert man das Originalbild  $s_{10}$  ein wenig in der Breite gemäß  

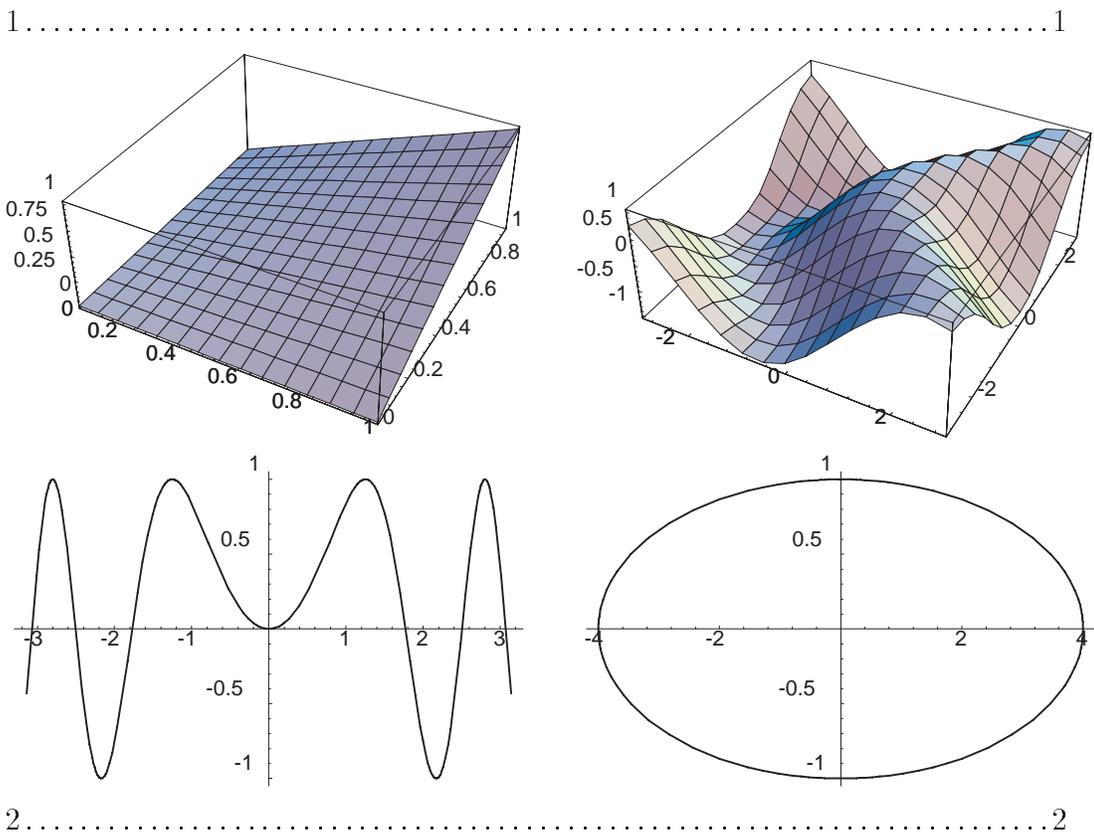
```
\epsfxsize70mm
\epsfbox[0 0 276.25 226.25]{tbs10.eps}
```

 so kann man vier solchartige Bilder in einem Graphikfeld anordnen.  
 Drei mögliche Versionen von Boxbefehlen dazu sollen hier angeführt werden,  
 um auch auf die Vielfalt solcher Strukturen hinzuweisen.

```

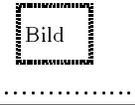
% --- Tableaus mit 4 Abbildungen Plot3D , Plot -----
% einfachste 1. Version
1\dotfill 1\\
\epsfxsize70mm\epsfbox[0 0 276.25 226.25]{tbs10.eps}\hspace{5mm}
\epsfxsize70mm\epsfbox[0 0 276.25 226.25]{sk02_5.eps}\\
\epsfxsize70mm\epsfbox[0 0 276.25 170.25]{sk02_1.eps}\hspace{5mm}
\epsfxsize70mm\epsfbox[0 0 276.25 170.25]{sk02_2.eps}\\
2\dotfill 2\\
% -----
% Version 2
\epsfxsize70mm\epsfbox[0 -170.25 276.25 226.25]{tbs10.eps}\\
% 3 weitere Bilder werden "hineingezogen"
% Reihenfolge : links unten, rechts oben, rechts unten
\epsfxsize70mm\epsfbox[ 0.00 -270.25 276.25 -100.00]{sk02_1.eps}
\epsfxsize70mm\epsfbox[-30.00 -441.50 246.25 -215.25]{sk02_5.eps}
\epsfxsize70mm\epsfbox[246.25 -265.25 522.50 -95.00]{sk02_2.eps}
% -----
% Version 3
\epsfxsize70mm\epsfbox[0 -170.25 276.25 226.25]{tbs10.eps}\\
% 3 weitere Bilder werden "hineingezogen"
% Reihenfolge : rechts oben, links unten, rechts unten
\epsfxsize70mm\epsfbox[-306.25 -441.50 -30.00 -215.25]{sk02_5.eps}
\epsfxsize70mm\epsfbox[ 276.25 -270.25 552.50 -100.00]{sk02_1.eps}
\epsfxsize70mm\epsfbox[ 246.25 -265.25 522.50 -95.00]{sk02_2.eps}

```



**Abb.6.** Tableau der Version 1 mit 4 Abbildungen  $s_{10}$ ,  $p_5$ ,  $p_1$ ,  $p_2$  als PS-File  $tbs_{10}.eps$ ,  $sk02_n.eps$ ,  $n=5,1,2$  (V2.2.3 Windows)

- Verkleinerung von Bildern in größerer Box  
Originalbildbreite= $w$ , -bildhöhe= $h$ ,  $w', w'', h', h'' > 0$

Boxbefehle	Verzerrung des Bildes	Bemerkung
<code>\epsfbox[0 0 w h]{*.eps}</code>		Original, $w \geq h$
<code>\epsfbox [-w' -h' w+w'' h+h'']{*.eps}</code>		Originalbild in größerer Box
<code>\epsfxsize <math>\tilde{w}</math> mm \epsfysize <math>\tilde{h}</math> mm \epsfbox [-w' -h' w+w'' h+h'']{*.eps}</code>		Bild in $x, y$ verkleinert gemeinsam mit Box auf Breite= $\tilde{w} < w+w'+w''$ Höhe = $\tilde{h} < h+h'+h''$

**Tab.9.** Bildverkleinerung mit Box

Die vertikalen Boxgrenzen unterliegen natürlich auch den Skalierungen, wurden aber nicht mit eingezeichnet.

Die Skalierung bezieht sich also auf die Ausdehnung der das Bild umgebenden Box. Mit ihrer Verkleinerung/Vergrößerung verändert sich auch proportional die darinliegende Graphik.

### 5.3.2 *Mathematica* V2.2.1 unter Windows (PS-File \*.ps)

Erste Anwendungen wurden schon im Kapitel 4 aufgezeigt. Das Originalbild in dieser Version

`\epsfbox{zi02_*.ps} \equiv \epsfbox[0 0 612 792]{zi02_*.ps}`

ist zu groß. Durch seine Ausdehnung in der Breite bis  $612\text{pt} = 215\text{mm}$  geht es i.a. über den rechten Papierrand hinaus. Seine Höhe bewegt sich je nach Graphiktyp im Intervall  $[377\dots 612\text{pt}] = [132\dots 215\text{mm}]$ . Die Graphik liegt als solche vertikal zentriert in der Box  $612 * 792\text{pt} = 215 * 279\text{mm}$ . Eine Unterschrift unter die Graphik „landet“ meistens auf der nächsten Seite.

Um den „Leerraum“ über und unter dem Bild zu eliminieren, modifizieren wir das Boxkommando zur Form (siehe Abb.7)

`\epsfbox[0 r 612 h+r]{zi02_*.ps}`

Will man einen kleinen Rand der Breite  $r'$  um das Bild lassen, so machen wir die Änderung

`\epsfbox[0 r-r' 612 h+r+r']{zi02_*.ps}`

A4-Seite (210\*297 mm)

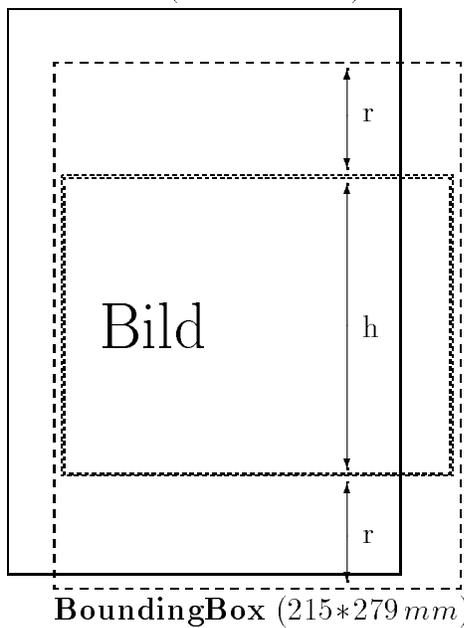


Abb.7

Lage der Box und Graphik zur A4-Seite,  
Parameterangaben  $h$ ,  $r$  gemäß Tab.10

Möchte man nun die Graphik vernünftig in den Text auf einer A4-Seite einordnen (Breite  $\approx 150\text{ mm}$ ), so muß man diese auf ca  $2/3$  ihrer Größe verkleinern. Wenden wir dazu die Skalierung der Breite an, so heißt dies

```
\epsfxsize438pt
\epsfbox[0 r-r' 612 h+r+r']{zi02_*.ps}
```

Bei Nutzung der Skalierung der Höhe ist es ein wenig komplizierter, da ja die Graphen verschieden hoch sind. Mit entsprechenden  $y$ -Maßen kann beispielsweise für  $r' = 20$  ebenfalls die Bildbreite 438  $pt$  erzielt werden.

Graphik	$h$ <i>pt</i>	$r$ <i>pt</i>	Bild mit Rand	Skalierung der Höhe
			$h + 2r'$	$y = 438 \frac{h+2r'}{w}$
			<i>pt</i>	<i>pt</i>
p1, p2	377	207.5	417	298
p3, p4	607	92.5	647	463
p5	501	145.5	541	387
p6	612	90.0	652	467
zi02_a6*.ps p	572	110.0	612	438

Tab.10. Bildhöhen  $h$  und Leerräume  $r$  vor und nach der Graphik,  
 $w = 612$ ,  $h + 2r = 792$ ,  $r' = 20$ ,  
Skalierungsmaßstab in `\epsfysize y pt`

Weitere Bildmanipulationen wie

- Verzerrung,
- Verschiebung,
- Vekleinerung mit Box

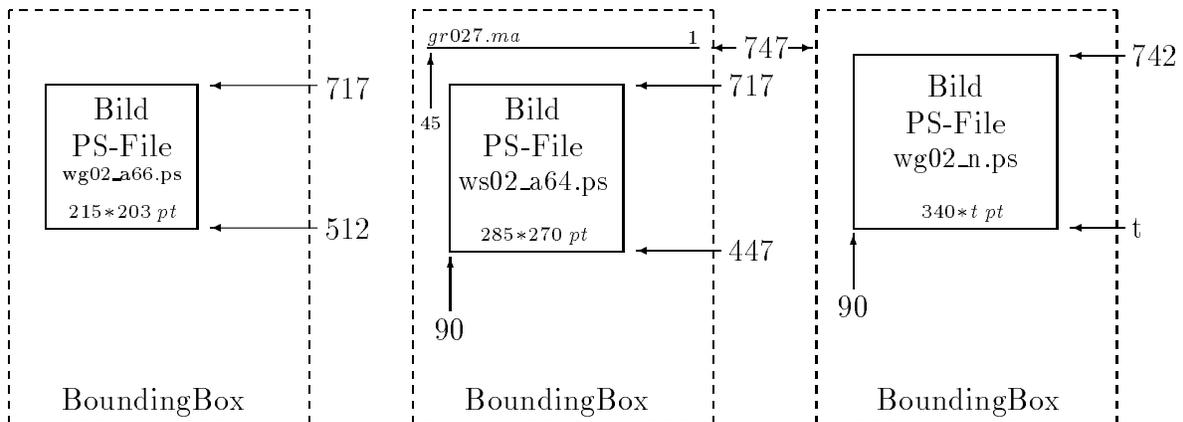
sowie deren Kombination funktionieren so wie in V2.2.3 mit PS-Files \*.eps.

### 5.3.3 *Mathematica* V2.2.1 unter Unix (PS-File \*.ps)

An dieser Stelle gehen wir zunächst auf die im Kap. 5.1 erwähnten Unix-PS-Files `wg02_a66.ps`, `ws02_a64.ps` ein und erinnern an die Auswirkungen des Anbringens von Überschriften an die Graphen.

Der Vermerk über die BoundingBox  $612 * 792 pt$  im PS-File ist eher nichtssagend für das eigentliche Bild. Die Graphik befindet sich standardmäßig direkt unterhalb der Kopfzeile etwas horizontal nach links verschoben. Das Löschen der Kopfzeile durch entsprechende Maßnahmen heißt nicht, daß die Graphik automatisch um diesen Platz höher rutscht. Prinzipiell geht es wieder darum, das Bild - ev. mit Kopfzeile - zu erfassen und dann mit den inzwischen bekannten Kommandos zu modifizieren. Dazu kommt noch die Anordnung der Box auf der A4-Seite.

Veranschaulichen wir uns die Situation an einigen Bildern.



**Abb.8.** Graphen zu PS-Files und ihre Lage in der Box  $612 * 792 pt$ , Parameter  $t$  siehe Tab.11

Läßt man die Kopfzeile und will linksbündig mit dem Text auf der Seite sein, muß man diese zusammen mit dem Graph um ca  $45 pt$  nach links schieben. Deshalb

```
\epsfbox[45 447 657 792]{ws02_a64.ps}
```

**Ohne** Kopfzeile kann man den Graph um weitere  $45 pt$  nach links verschieben. Wichtiger hier ist jedoch, daß das Bild mit seinen Begrenzungen nunmehr alleine erfaßt und manipuliert werden kann.

Graphik wg02_n.ps n=1,...,6	Bildgröße  <i>Pixel</i>	Bildgröße 340 * t  <i>pt</i>	Box und Verschiebung $\backslash\text{epsfbox}[90\ 742-t\ 430\ 742]$  <i>pt</i>
p1, p2	300 * 185	340 * 210	[90 532 430 742]
p3, p4	300 * 298	340 * 338	[90 404 430 742]
p5	300 * 246	340 * 279	[90 463 430 742]
p6	300 * 300	340 * 340	[90 402 430 742]
p			
wg02_a64.ps	252 * 238	285 * 270	[90 447 375 717]
wg02_a66.ps	≈ 190 * 180	≈ 215 * 205	≈ [90 512 305 717]

**Tab.11.** Bildgrößen von PS-Files ohne Header wg02\_\*.ps

**Bemerkungen:**

- Um auf Textbreite der Seite zu kommen, sind die Bilder  $p1, \dots, p6$  und wg02\_a64.ps ungefähr auf das Anderthalbfache zu vergrößern, während wg02\_a66.ps zu verdoppeln ist.
- Die Boxgrenzen für die Bilder  $p1, \dots, p6$  kann man noch etwas verkleinern, womit die Umgebung näher heranrückt. Das macht bis zu 40pt in horizontaler Richtung aus, sowie bis 30pt vertikal. Zum Beispiel für  $p1, p6$

```
\epsfbox[90 532 390 732]{wg02_1.ps}
\epsfbox[90 417 390 727]{wg02_6.ps}
```

## 5.4 Anwendung des Style psfig.sty

In der Datei **PSFIG.STY** findet der Nutzer Hinweise zur Einbindung, und wir notieren daraus die für uns zutreffenden wesentlichen Anteile.

```
To use with LaTeX, use \documentstyle[psfig,...]{...}
```

Command synopsis:

```
\psfig
usage: \psfig{file=, figure=, height=, width=,
            bllx=, billy=, bburx=, bbury=,
            rheight=, rwidth=, clip=, angle=, silent=}
```

"file" is the filename.  
If no path name is specified and the file is not found in the current directory, it will be looked for in directory \psfigurepath.  
"figure" is a synonym for "file".

By default, the width and height of the figure are taken from the BoundingBox of the figure.  
If "width" is specified, the figure is scaled so that it has the specified width. Its height changes proportionately.

If "height" is specified, the figure is scaled so that it has the specified height. Its width changes proportionately. If both "width" and "height" are specified, the figure is scaled anamorphically.

"bblx", "bblly", "bburx" and "bbury" control the PostScript BoundingBox. If these four values are specified \*before\* the "file" option, the PSFIG will not try to open the PostScript file.

"rheight" and "rwidth" are the reserved height and width of the figure, i.e., how big TeX actually thinks the figure is. They default to "width" and "height".

The "clip" option ensures that no portion of the figure will appear outside its BoundingBox. "clip=" is a switch and takes no value, but the '=' must be present.

The "angle" option specifies the angle of rotation (degrees, ccw). The "silent" option makes \psfig work silently.

Die Skalierung eines Bildes in Breite und Höhe bezieht sich also immer auf die Angaben zur BoundingBox, unabhängig davon wo und wie dort die eigentliche Graphik platziert ist. Deshalb ist der Umgang mit den Postscript-Files \*.eps von *Mathematica* V2.2.3 am anschaulichsten, weil dort Boxgröße gleich Graphikausdehnung ist. Die beiden anderen Versionen haben entweder in einer übergroßen Box eine breite, aber vertikal zentrierte Graphik (V2.2.1 Windows) oder eine kleinere Graphik unterhalb des oberen Boxrandes, ev. noch mit einer Kopfzeile dazwischen (V2.2.1 Unix). Aus der Vielfalt der Möglichkeiten möchten wir an dieser Stelle nur auf einige gebräuchliche Varianten verweisen. Die horizontalen Begrenzungslinien 1 bzw. 2 stehen für das Textende vor der Abbildung und den Textanfang danach.

- Ausgabe des PS-Files `tbs10.eps` in Originalgröße und seine horizontal zentrierte Platzierung als Gleitobjekt (siehe auch Kap. 5.2).

```
1\dotfill 1\\
% Positionierung an der Stelle im Text, wo diese Umgebung auftritt
\begin{figure}[h]
\centerline{\psfig{figure=TBS10.EPS}}
\centerline{\bf Figure 1}
\end{figure}\\
2\dotfill 2\\
```

- Ausgabe des PS-Files `tbs10.eps` linksbündig in Textbreite 438 pt.

```
1\dotfill 1\\
\psfig{figure=TBS10.EPS,width=438pt}
2\dotfill 2\\
{\bf Figure 2}
```

- Ausgabe des PS-Files `zi02_1.ps` in Textbreite 438 pt.  
Das Originalbild 612 \* 377 pt in BoundingBox 612 \* 792 pt wird reduziert auf ein Bild in Textbreite 438 \* 290 pt in einer Box 438 \* 569 pt. Damit ergibt sich ein Heranrücken des Textes oben und unten von ca 140 pt.

```

1\dotfill 1\[-140pt]
\psfig{figure=zi02_1.ps,width=438pt}\[-140pt]
2\dotfill 2\
{\bf Figure 3}

```

- Ausgabe des PS-Files `ws02_a64.ps` mit Kopfzeile in Textbreite  $438\text{ pt}$ . Dabei werden folgende Überlegungen an Hand der Abb.3 gemacht. Um die Graphen im Tableau nicht zu sehr zu verkleinern, testen wir aus, welche Boxgröße mit Über-/Unterschrift gerade noch auf eine A4-Seite paßt. Bei proportionaler Stauchung der Seiten ist es die Boxdimension  $483 * 625\text{ pt}$ . Die ursprüngliche Kopfzeile der Länge  $\approx 500\text{ pt}$  wird dabei auf  $\approx 400\text{ pt}$  reduziert. Ihre mögliche Streckung auf Textbreite liefert die Angabe  $width = 530\text{ pt}$ . Damit werden auch die Graphiken etwas breiter. Anschließend wird die Box noch nach links gerückt und der darunterstehende Text „herangezogen“. Das Ergebnis ist vergleichbar mit Abb.4 bei Einhaltung der Textbreite.

```

1\dotfill 1\[-35pt]
\hspace*{-45pt}
\psfig{figure=ws02_a64.ps,width=530pt,height=625pt}\[-350pt]
2\dotfill 2\
{\bf Figure 4}

```

- Ausgabe des PS-Files `wg02_a66.ps` ohne Kopfzeile in Textbreite  $438\text{ pt}$ . Wir gehen hier schrittweise vor.
  1. Wie oben, aber ohne Kopfzeile. Dadurch kann die Graphik noch etwas mehr nach oben und links versetzt werden.

```

1\dotfill 1\[-60pt]
\hspace*{-90pt}
\psfig{figure=wg02_a66.ps,width=530pt,height=625pt}\[-385pt]
2\dotfill 2\
{\bf Figure 5}

```

2. Beibehaltung der Originalgröße.

Hier wurde ein kleiner Trick mit einer Parbox gemacht, um das Stauchen der Graphik zu vermeiden.

```

1\dotfill 1\[-200pt]
\parbox[t]{150mm}{
\vspace{120pt}
\hspace*{-90pt}\psfig{figure=wg02_a66.ps}\[-500pt]
2\dotfill 2\
}\
{\bf Abb.9.} \verb| wg02_a66.ps | Originalgraph ohne Kopfzeile

```

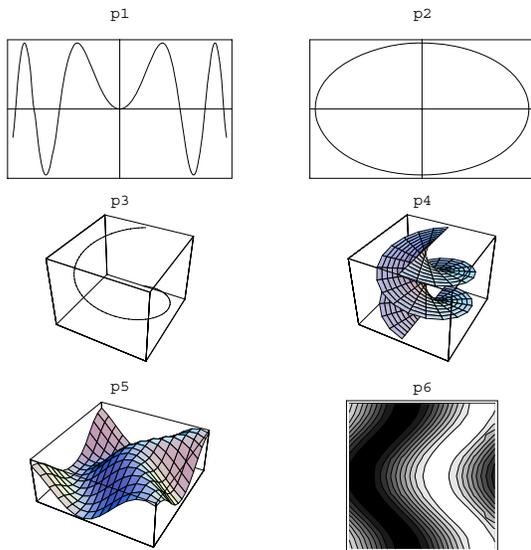


Abb.9. wg02\_a66.ps Originalgraph ohne Kopfzeile

3. Eine weitere Vergrößerung der Graphik auf Textbreite bedarf neuer Klimmzüge mit einer Parbox. Mit Geschick (Vergrößern bzw. Verschiebungsstrategien) und Austesten erhält man die entsprechende Kommandofolge.

```

\newpage
1\dotfill 1\[-750pt]
\parbox[t]{150mm}{
\vspace{580pt}
\hspace*{-210pt}
\psfig{figure=wg02_a66.ps,width=1230pt}\[-1000pt]
2\dotfill 2\
}\
{\bf Figure 6}\[1em]

```

Das Ergebnis zwischen den Begrenzungslinien ist sehr verwandt zur Abb.1.

4. Eine andere Möglichkeit besteht darin, in das PS-File wg02\_a66.ps einzugreifen und dort die BoundingBox-Angaben zu modifizieren. Die Boxgrenzen für die Dateien wg02\_\*.ps kann man z.B. der Tabelle 11 entnehmen. Die Streckungsfaktoren beziehen sich dann auf diese Boxausdehnung, und die Situation ähnelt der bei \*.eps-Dateien.

## 6 Das Postscript-File vom $\text{\TeX}$ -Dokument

Sobald man PS-Files im  $\text{\LaTeX}$ -Dokument hat, ist dieses Dokument zwecks vollständiger Ansicht mit allen Graphiken zuvor in ein PS-File selber zu transformieren.

- **TeXShell V2.6**

*Zusätze*  $\Rightarrow$  *PS – File erzeugen*  
 $\Rightarrow$  *PS – File ansehen*  
 $\Rightarrow$  *PS – File drucken*

Zwecks Ausgabe auch

*Druck*  $\Rightarrow$  *Druckertreiber*  $\Rightarrow$  *DVI PostScript (HP LaserJet 4M)*

- **TeXShell V2.7.1 $\beta$**

**PS-File Erzeugen:**

*Druck*  $\Rightarrow$  *Druckertreiber*  $\Rightarrow$  *PS – File erzeugen*  
 $\Rightarrow$  *PS – File no memory* (große Datei)

**PS-File Ansehen:**

*Zusätze*  $\Rightarrow$  *PS – File ansehen*

**PS-File Drucken:**

(a) *Zusätze*  $\Rightarrow$  *GS Nadeldrucker*  
 $\Rightarrow$  *GS Laserjet*  
(b) *Druck*  $\Rightarrow$  *Druckertreiber*  $\Rightarrow$  *PS – Drucker*  
 $\Rightarrow$  *PS – Drucker no mem*

(Ausdrucken geht etwas schneller)

## 7 Nachwort

Die vorliegende Beschreibung kann nicht die Nutzung aller diesbezüglichen  $\text{\LaTeX}$ -Möglichkeiten vermitteln. Der Normalanwender sollte jedoch zunächst angebotene und getestete Standardstrukturen preferieren, bevor er individuelle Formatierungen im graphischen Design verwirklicht.

Die inzwischen vorliegende **Version 3** von *Mathematica* liefert möglicherweise neue Aspekte zur untersuchten Thematik.

Ich habe mich bemüht, die Hinweise aus Diskussionen mit Kollegen zu berücksichtigen. Für solche Anregungen möchte ich mich hier namentlich bei DM Brigitte Walther und Dr. Axel Wolf (beide TU Ilmenau) bedanken. Eventuelle Zuschriften und Anfragen versuche ich zu beantworten. Für weitere Hinweise bin ich dankbar.

# Literatur

- [1] WOLFRAM, S.: *Mathematica*. A System for Doing Mathematics by Computer. Addison-Wesley Publ. Company, Inc., Redwood City, California 1991.
- [2] LAMPORT, L.: *ΛT<sub>E</sub>X*. A Document Preparation System, User's Guide and Reference Manual. Addison-Wesley Publishing Company 1985. ISBN 0-201-15790-X.
- [3] WONNEBERGER, R.: *Kompaktführer ΛT<sub>E</sub>X*. Addison-Wesley Deutschland 1987. ISBN 3-925118-46-2.
- [4] KNUTH, DONALD E.: *The T<sub>E</sub>X Book*. Band A der Reihe *Computers and Typesetting*. Addison-Wesley Publishing Company 1984. ISBN 0-201-13448-9.
- [5] SCHWARZ, N.: *Einführung in T<sub>E</sub>X*. Addison-Wesley Deutschland 1987. ISBN 3-925118-25-X.
- [6] HOFMANN, T.: *ΛT<sub>E</sub>X Erweiterungen und Modifikationen*. CIBA-GEIGY AG Basel 1987.

## Anschrift:

Dr. Werner Neundorf  
Technische Universität Ilmenau Institut für Mathematik  
PF 10 0565  
D - 98684 Ilmenau  
e-mail : neundorf@mathematik.tu-ilmenau.de