Die TEXnische Komödie

DANTE
Deutschsprachige
Anwendervereinigung TEX e.V.

Herbert Voß: *Die mathematischen Funktionen von Postscript*, Die TEXnische Komödie 1/2002, S. 40–47.

Die TEXnische Komödie ist die Mitgliedszeitschrift von DANTE, Deutschsprachige Anwendervereinigung TEX e.V. Einzelne Hefte können von Mitgliedern bei der Geschäftsstelle von DANTE, Deutschsprachige Anwendervereinigung TEX e.V. erworben werden. Mitglieder erhalten Die TEXnische Komödie im Rahmen ihrer Mitgliedschaft.

Die mathematischen Funktionen von Postscript

Herbert Voß

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

Einführung

PostScript, sehr häufig nur als Druckertreiber bekannt oder im Zusammenhang damit als Seitenbeschreibungssprache, ist ähnlich wie TEX eine Programmiersprache, wobei PostScript eindeutig als sogenannte Hochsprache bezeichnet werden muss.[1] Während die Ursprünge der Entwicklung bis in die Anfänge der siebziger Jahre zurückgehen, datiert die erste veröffentlichte PostScript-Version von 1982, was nicht zufällig das Jahr der Gründung von *Adobe System Incorporated* ist. Der erste Laserdrucker mit vollständiger

PostScript-Implementierung war der *Apple Laserwriter*. Der einzige Computer, der PostScript auch für die Bildschirmausgabe (Display PostScript) verwendet, ist *Next*.

PostScript-Befehle

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

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

Anwendung mit pst-plot

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

Die grundsätzliche Struktur der hier behandelten Befehle ist:

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

Hierin bedeuten $[x_{min};\ x_{max}]$ und $[t_{min};\ t_{max}]$ das jeweilige Definitionsintervall (Start- und Endwert). Die möglichen Parameter können der Dokumentation zu `pstricks` entnommen werden, die zwar schon älteren Datums

ist, jedoch völlig ausreicht [2]. Hier ist im Zusammenhang mit den Funktionen lediglich der Parameter `plotpoints` interessant, welcher die Anzahl der Stützstellen angibt und standardmäßig auf 50 gesetzt ist. Sämtliche berechneten Punkte werden intern mit `\psline` verbunden, so dass sich eine zu geringe Anzahl an Stützstellen durch einen Polygonzug bemerkbar macht. Mit Werten um 200 liegt man in den meisten Fällen auf der richtigen Seite. In diesem Zusammenhang ist unbedingt zu beachten, dass geänderte Parameter so lange ihren Wert behalten, bis innerhalb eines LaTeX-Laufs eine neue Wertzuweisung erfolgt.

Der Variablenname für `psplot` ist per Definition x und für `parametricplot` t. Beide können nicht verändert werden, was jedoch bezüglich der Anwendung keinerlei Einschränkung darstellt. Die Variablen können beliebig oft innerhalb eines Ausdrucks verwendet werden, denn erst mit der schließenden Klammer für den Funktionsausdruck wird davon ausgegangen, dass die zweite Koordinate für einen Punkt des Graphen oben auf dem Stack liegt. Der einzige Unterschied zwischen diesen beiden Befehlen ist, dass bei `psplot` nur der oberste Stack-Wert (y) und bei `parametricplot` die obersten beiden Stack-Werte ($x; y$) als Argumente verwendet werden.

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

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

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

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

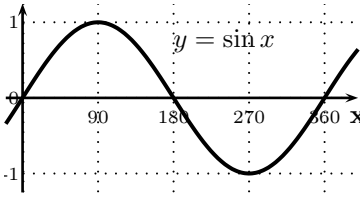
Beispiele für \psplot

Für alle Beispiele wird jeweils die komplette `pspicture`-Umgebung angegeben, sodass eine direkte Übernahme der Beispiele möglich ist. Eine Doku-

mentation des `multido`-Befehls findet man unter `CTAN:/macros/latex209/contrib/multido/multido.doc`, alle anderen in der Dokumenation zu `pstricks` [2].

Sinusfunktion

Funktion	PostScript
$y(x) = \sin x$	`x sin`

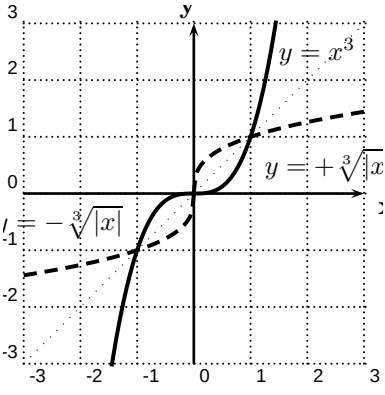



```
\psset{xunit=0.0111cm,yunit=1cm}
\begin{pspicture}(-20,-1.25)(400,1.25)
\psline[linewidth=1pt]{->}(-20,0)(400,0)
\psline[linewidth=1pt]{->}(0,-1.25)(0,1.25)
\multido{\n=+90}{5}{\psline[linestyle=dotted]%
  (\n,-1.25)(\n,1.25)\rput(\n,-0.25)%
  {\scriptsize \n}}
\multido{\n=-1+1}{3}{\psline[linestyle=dotted]%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

Potenzfunktion

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

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



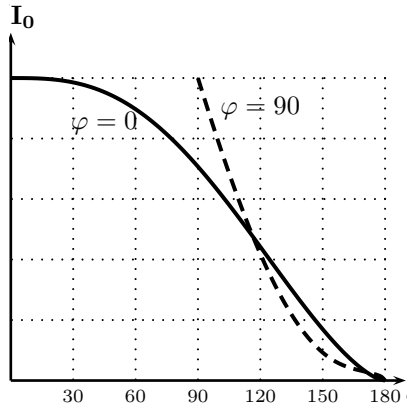

```
\begin{pspicture}(-3,-3)(3,3)
\psgrid[subgriddiv=1,griddots=10,%
   gridlabels=7pt](-3,-3)(3,3)%
\psline[linewidth=1pt]{->}(-3,0)(3,0)%
\psline[linewidth=1pt]{->}(0,-3)(0,3)%
\psline[linewidth=0.5pt,linestyle=dotted]%
   (-3,-3)(3,3)%
\psplot[plotstyle=curve,linewidth=1.5pt]%
   {-1.4}{1.4}{x 3 exp}%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

Beispiel aus der Leistungselektronik

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

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

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



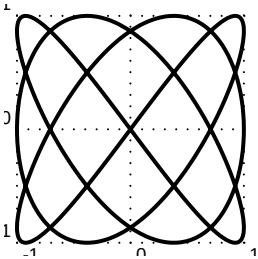

```
\psset{xunit=0.0333cm,yunit=3cm}
\begin{pspicture}(0,-0.25)(190,1.25)
\psline[linewidth=1pt]{->}(0,0)(190,0)
\psline[linewidth=1pt]{->}(0,0)(0,1.15)
\multido{\n=30+30}{6}{\psline[%
   linestyle=dotted](\n,0)(\n,1)%
   \rput(\n,-0.05){\scriptsize \n}}
\multido{\n=0.2+0.2}{5}{\psline[%
   linestyle=dotted](0,\n)(180,\n)%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

Beispiele für `\parametricplot`

Lissajous-Figur

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

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



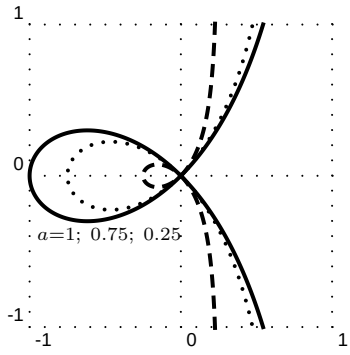
```
\psset{xunit=1.5cm,yunit=1.5cm}
\begin{pspicture}(-1,-1)(1,1)
\psgrid[subgriddiv=0,griddots=10,gridlabels=7pt]%
   (-1,-1)(1,1)
\parametricplot[plotstyle=curve,linewidth=1.5pt,%
   plotpoints=200]{-360}{360}{%
   t 1.5 mul sin t 2 mul 60 add sin}
\end{pspicture}
```

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

Strophoide

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

Funktion	PostScript
$x(t) = \frac{a\left(t^2-1\right)}{t^2+1}$	`t t mul 1 sub` a `mul t t mul 1 add div`
$y(t) = \frac{at\left(t^2-1\right)}{t^2+1}$	`t t mul 1 sub t mul` a `mul t t mul 1 add div`




```
\psset{xunit=2cm,yunit=2cm}
\begin{pspicture}(-1.25,-1.25)(1.25,1.25)
\psgrid[subgriddiv=1,griddots=10,%
   gridlabels=7pt](-1,-1)(1,1)
\parametricplot[plotstyle=curve,plotpoints=200,%
   linewidth=1.5pt]{-1.85}{1.85}%
   {t t mul 1 sub t t mul 1 add div % a=1
   t t mul 1 sub t mul t t mul 1 add div}
\parametricplot[plotstyle=curve,plotpoints=200,%
   linewidth=1.5pt, linestyle=dashed]{-5}{5}%
   {t t mul 1 sub 0.25 mul t t mul 1 add div %
   t t mul 1 sub t mul 0.25 mul t t mul 1 add div}
\parametricplot[plotstyle=curve,plotpoints=200,%
   linewidth=1.5pt, linestyle=dashed]{-1}{1}%
   {t t mul 1 sub 0.75 mul t t mul 1 add div %
   t t mul 1 sub t mul 0.75 mul t t mul 1 add div}
\rput[l](-0.95,-0.4){$\scriptstyle %
    a=1;\ 0.75;\ 0.25$}
\end{pspicture}
```

Zusammenstellung der mathematischen PostScript-Funktionen

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

	integer	*real*
kleinster Wert	-2^{32}	$\pm 10^{-38}$
größter Wert	$2^{32}-1$	$\pm 10^{38}$

Name	*Bedeutung*	*Anwendung*	*Beispiel*
abs	Absolutwert	<num> abs	−3 abs → 3
add	Addition[1]	<num1> <num2> add	5 7 add → 12
atan	Arcus Tangens[2]	<real1> <real2> atan	2 45 atan→ 2.54
cos	Cosinus[3]	<real> cos	60 cos→ 0.5
cvi	Real→Integer	<real> cvi	14.13 cvi→ 14
cvr	Integer→Real	<int> cvr	14 cvr→ 14.00
div	Division[4]	<real1> <real2> div	100 8 div→ 12.5
dup	Dupliziere oberstes Stack-Element	<any> dup	12 dup→ 12 12
exch	Exchange[5]	<any1> <any2> exch	12 13 exch→ 13 12
exp	Potenz	<real1> <real2> exp	3 4 exp→ 81.0
idiv	ganzzahlige Division	<int1> <int2> idiv	100 8 idiv→ 12
ln	natürlicher Logarithmus	<real> ln	12 ln→ 2.48491
log	Zehner-Logarithmus	<real> log	1000 log→ 3.00
mod	Modulo	<int1> <int2> mod	5 3 mod→ 2
mul	Multiplikation[1]	<num1> <num2> mul	5 3 mul→ 15
neg	Negiere Vorzeichen	<num> neg	5 neg→ −5
round	Runden	<real> round	5.7 round→ 6
sin	Sinus[3]	<real> sin	30 sin→ 0.5
sqrt	Quadratwurzel	<real> sqrt	16 sqrt→ 4.0
sub	Subtraktion[1]	<num1> <num2> sub	17 19 sub→ −2
truncate	Dezimalteil abtrennen[6]	<real> truncate	−33.33 → 33.00

Anmerkungen:

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

[2] Entspricht der Darstellung $\alpha = \arctan\frac{<real1>}{<real2>}$.

[3] Das Argument wird im Gradmaß erwartet.

[4] Das Ergebnis ist grundsätzlich vom Typ *real*.

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

[6] Das Ergebnis bleibt vom Typ *real*.

Literatur

[1] Nikolai G. Kollock: *PostScript richtig eingesetzt: vom Konzept zum praktischen Einsatz*; IWT; Vaterstetten; 1989.

[2] Timothy Van Zandt: *PSTricks - PostScript macros for Generic TeX*; 1993; http://www.tug.org/application/PSTricks.