

PSTricks

Matthieu Felsinger

Vortragsreihe "L^AT_EX in der Mathematik"

16.01.2014

1 Einleitung

2 Erste Schritte

3 Das Paket `pst-func`

4 PSTricks und `pdflatex`

5 Zum Schluss...

1 Einleitung

2 Erste Schritte

3 Das Paket `pst-func`

4 PSTricks und `pdflatex`

5 Zum Schluss...

Grundlegendes

- Was ist PSTricks?

Eine Sammlung von $\text{T}_\text{E}\text{X}$ -Makros zur Grafikerzeugung auf der Basis der Programmiersprache PostScript. Verwendet Funktionen, die von `dvips` erledigt werden.

Grundlegendes

- Was ist PSTricks?

Eine Sammlung von $\text{T}_\text{E}\text{X}$ -Makros zur Grafikerzeugung auf der Basis der Programmiersprache PostScript. Verwendet Funktionen, die von `dvips` erledigt werden.

- Ist PSTricks aktuell?

Timothy Van Zandt initiierte 1991 PSTricks. (Weiter-)Entwicklung von Paketen dauert bis heute an. Hauptakteur mittlerweile: Herbert Voß

Grundlegendes

- Was ist PSTricks?

Eine Sammlung von $\text{T}_\text{E}\text{X}$ -Makros zur Grafikerzeugung auf der Basis der Programmiersprache PostScript. Verwendet Funktionen, die von `dvips` erledigt werden.

- Ist PSTricks aktuell?

Timothy Van Zandt initiierte 1991 PSTricks. (Weiter-)Entwicklung von Paketen dauert bis heute an. Hauptakteur mittlerweile: Herbert Voß

- Was kann PSTricks?

Hochwertige Vektorgrafiken mit überschaubarem Aufwand durch zahlreiche Pakete. Einfache Zusammenarbeit von `(pdf)latex` und PSTricks.

Grundlegendes

- Was ist PSTricks?

Eine Sammlung von T_EX-Makros zur Grafikerzeugung auf der Basis der Programmiersprache PostScript. Verwendet Funktionen, die von dvips erledigt werden.

- Ist PSTricks aktuell?

Timothy Van Zandt initiierte 1991 PSTricks. (Weiter-)Entwicklung von Paketen dauert bis heute an. Hauptakteur mittlerweile: Herbert Voß

- Was kann PSTricks?

Hochwertige Vektorgrafiken mit überschaubarem Aufwand durch zahlreiche Pakete. Einfache Zusammenarbeit von (pdf)latex und PSTricks.

- Was kann PSTricks nicht?

Hoch komplexe Grafikprogramme ersetzen (z.B. AutoCAD). Für *alle* Szenarien gut sein. Entscheidung für oder gegen PSTricks durch Anwender.

Vorbereitungen

- Mit (HV) gekennzeichnete Beispiele stammen aus dem Buch
 Herbert Voß: *PSTricks : Grafik mit PostScript für T_EX und L^AT_EX*,
Lehmanns, Berlin, 5. Auflage (2008)
Auch Grundlage dieses Vortrags.

Vorbereitungen

- Mit (HV) gekennzeichnete Beispiele stammen aus dem Buch
 Herbert Voß: *PSTricks : Grafik mit PostScript für T_EX und L_AT_EX*, Lehmanns, Berlin, 5. Auflage (2008)
Auch Grundlage dieses Vortrags.
- Alle Beispiele sind getestet mit der T_EX-Distribution T_EXLive 2013.

Warnung

Die T_EXLive Versionen in den Paketquellen von Ubuntu sind **nicht** aktuell. Die Installation einer aktuellen T_EX-Distribution ist empfehlenswert. Dies ist auch auf den Fakultätsrechnern möglich.

Vorbereitungen

- Mit (HV) gekennzeichnete Beispiele stammen aus dem Buch
 Herbert Voß: *PSTricks : Grafik mit PostScript für T_EX und L_AT_EX*, Lehmanns, Berlin, 5. Auflage (2008)
Auch Grundlage dieses Vortrags.
- Alle Beispiele sind getestet mit der T_EX-Distribution T_EXLive 2013.

Warnung

Die T_EXLive Versionen in den Paketquellen von Ubuntu sind **nicht** aktuell. Die Installation einer aktuellen T_EX-Distribution ist empfehlenswert. Dies ist auch auf den Fakultätsrechnern möglich.

- Vortrag bietet nur einen ersten Eindruck für Anwendungsmöglichkeiten. Ausführlichere Darstellungen finden sich u.a. im Buch von Herbert Voß, in den Dokumentationen der einzelnen Pakete, sowie unter <http://tug.org/PSTricks>

1 Einleitung

2 Erste Schritte

3 Das Paket `pst-func`

4 PSTricks und `pdflatex`

5 Zum Schluss...

2 Erste Schritte

Grundlagen

Achsenbeschriftung für trig. Funktionen

Erweiterungen

Die pspicture-Umgebung

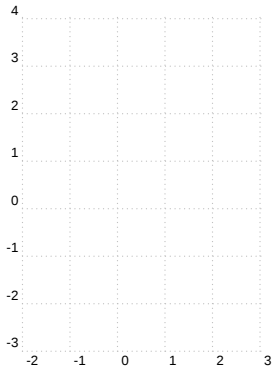
Reserviert den Platz für die Zeichnung. Syntax:

```
\begin{pspicture*}[Optionen](xMin,yMin)(xMax,yMax)  
...  
\end{pspicture*}
```

Ausmaße müssen geeignet gewählt werden. Die *-Variante sorgt dafür, dass alles außerhalb des Zeichenbereichs abgeschnitten wird.

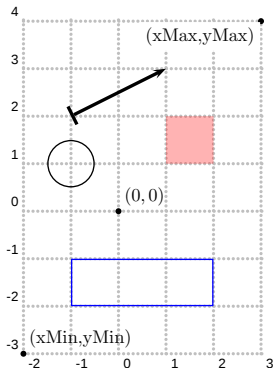
Zulässige Optionen: `shift=<dim>` für den vertikalen Offset (Voreinstellung: `0pt`) und `showgrid=true/false` (Voreinstellung: `false`).

Ein erstes Beispiel



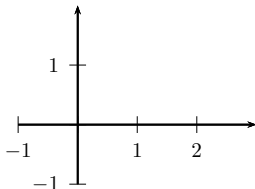
```
\begin{pspicture}[showgrid=true]%  
  (-2,-3)(3,4)  
\end{pspicture}
```

Ein erstes Beispiel



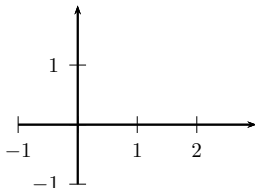
```
\psset{gridwidth=2pt}
\begin{pspicture}[showgrid=true]%
  (-2,-3)(3,4)
\psdot(0,0)\uput[45](0,0){$(0,0)$}
\psdot(-2,-3)\uput[45](-2,-3){(xMin,yMin)}
\psdot(3,4)\uput*[225](3,4){(xMax,yMax)}
\psline[linewidth=2pt]{| ->}(-1,2)(1,3)
\psframe[linecolor=blue](-1,-2)(2,-1)
\psframe*[linecolor=red,%
  opacity=0.3](1,1)(2,2)
\pscircle(-1,1){0.5}
\end{pspicture}
```

Koordinatensysteme

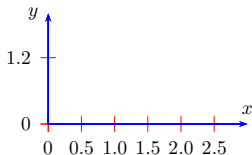


```
\usepackage{pstricks, pst-plot}  
\begin{pspicture}(-1.5,-1.5)(3.5,2.5)  
  \psaxes{>->}(0,0)(-1,-1)(3,2)  
\end{pspicture}
```

Koordinatensysteme

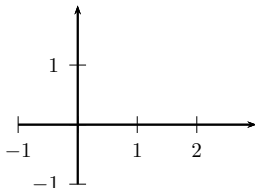


```
\usepackage{pstricks, pst-plot}
\begin{pspicture}(-1.5,-1.5)(3.5,2.5)
  \psaxes{->}(0,0)(-1,-1)(3,2)
\end{pspicture}
```

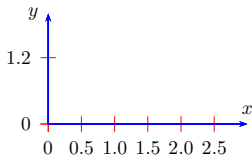


```
\usepackage{pstricks, pst-plot}
\psset{xunit=1.2}
\begin{pspicture}(-1,-1)(3.5,2.5)
  \psaxes[Dx=0.5,Dy=1.2,%
    linecolor=blue, tickcolor=red]{->}%
    (0,0)(3,2)[$x$,90][$y$,180]
\end{pspicture}
```

Koordinatensysteme



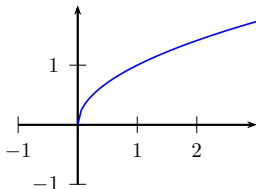
```
\usepackage{pstricks, pst-plot}
\begin{pspicture}(-1.5,-1.5)(3.5,2.5)
  \psaxes{->}(0,0)(-1,-1)(3,2)
\end{pspicture}
```



```
\usepackage{pstricks, pst-plot}
\psset{xunit=1.2}
\begin{pspicture}(-1,-1)(3.5,2.5)
  \psaxes[Dx=0.5,Dy=1.2,%
    linecolor=blue, tickcolor=red]{->}%
    (0,0)(3,2)[$x$,90][$y$,180]
\end{pspicture}
```

Es gibt zahlreiche weitere Optionen für `\psaxes`, siehe Dokumentation des Paketes `pst-plot`.

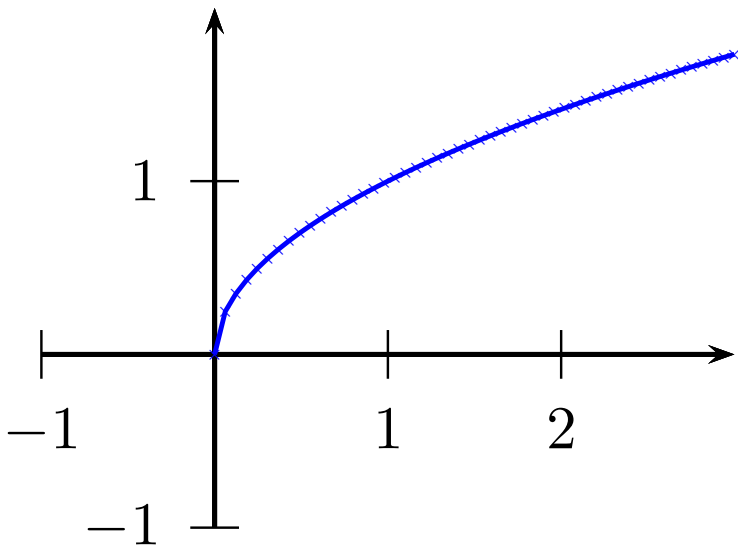
Eine erste Beispielfunktion



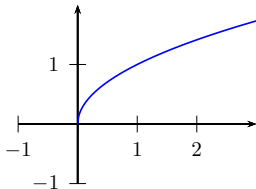
```
\usepackage{pstricks, pst-plot}
\begin{pspicture}(-1.5,-1.5)(3.5,2.5)
  \psaxes{>->}(0,0)(-1,-1)(3,2)
  \psplot[linecolor=blue]{0}{3}{x 0.5 exp}
\end{pspicture}
```

Eingabe der Funktion im PostScript-Modus (umgekehrte polnische Notation, UPN). PostScript ist stackorientiert.

Eine erste Beispielfunktion



Eine erste Beispielfunktion

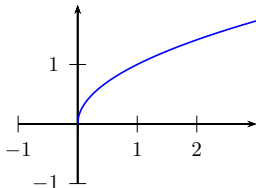


```
\usepackage{pstricks, pst-plot}
\begin{pspicture}(-1.5,-1.5)(3.5,2.5)
  \psaxes{>}(0,0)(-1,-1)(3,2)
  \psplot[linecolor=blue,algebraic,%
    plotpoints=200]{0}{3}{sqrt(x)}
\end{pspicture}
```

`plotpoints` gibt die Anzahl der zu berechnenden Stützstellen vor. “Glättender” Effekt.

`algebraic` lässt die Eingabe der Funktion in Infix-Notation zu. Steht nur innerhalb des Plotbefehls zur Verfügung. Abhilfe: `\usepackage{infix-RPN}`

Eine erste Beispielfunktion



```
\usepackage{pstricks, pst-plot}
\begin{pspicture}(-1.5,-1.5)(3.5,2.5)
  \psaxes{->}(0,0)(-1,-1)(3,2)
  \psplot[linecolor=blue,algebraic,%
    plotpoints=200]{0}{3}{sqrt(x)}
\end{pspicture}
```

Neben +, -, *, / sowie $\wedge$ gibt es folgende Funktionen für die Anwendung in algebraischer Notation:

sin, cos, tan (Argument im Bogenmaß!)
ceiling, floor, truncate, round
sqrt, abs, fact

acos, asin
ln, log
Sum, IfTE

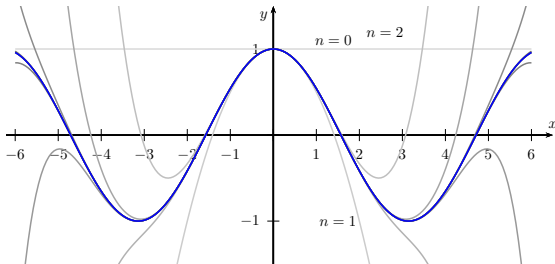
Sum(*Indexname, Start, Schrittweite, Ende, Funktion*)

```

\psset{algebraic,yunit=2,linewidth=1pt, plotpoints=200}
\begin{pspicture*}(-6.2,-1.5)(7,1.5)
  \psaxes{->}(0,0)(-7,-1.5)(6.5,1.5)[$x$,90][$y$,225]
  \psline[linecolor=black!15](-6,1)(6,1)
  \multido{\nA=1+1,\nB=20+5}{10}{%
    \psplot[linecolor=black!\nB]{-6}{6}{%
      Sum(i,0,1,\nA,(-1)^i*x^(2*i)/fact(2*i))}%
    \psplot[linecolor=blue]{-6}{6}{cos(x)}
  }
  \uput[180](2,1.1){$n=0$}\uput[0](0.9,-1){$n=1$}
  \uput[180](3.2,1.2){$n=2$}
\end{pspicture*} % (HV, Bsp. 15-03-11)

```

$$\cos(x) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{(2n)!}$$



2 Erste Schritte

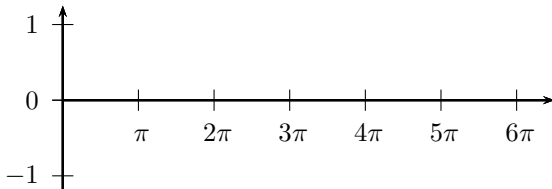
Grundlagen

Achsenbeschriftung für trig. Funktionen

Erweiterungen

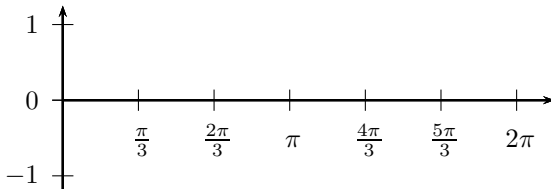
Beispiele mit trigLabels

```
\begin{pspicture}(-1,-1.5)(7,1.5)  
  \psaxes[trigLabels]{->}(0,0)(0,-1.2)(6.5,1.25)  
\end{pspicture}
```



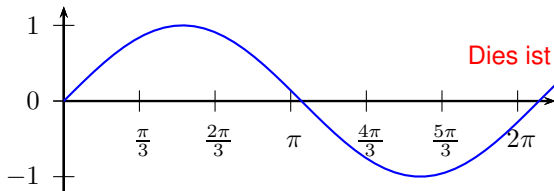
Beispiele mit trigLabels

```
\begin{pspicture}(-1,-1.5)(7,1.5)  
  \psaxes[trigLabels, trigLabelBase=3]{->}(0,0)(0,-1.2)(6.5,1.25)  
\end{pspicture}
```



Beispiele mit trigLabels

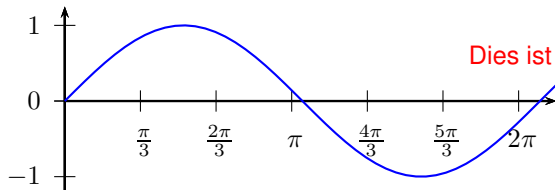
```
\begin{pspicture}(-1,-1.5)(7,1.5)
  \psaxes[trigLabels, trigLabelBase=3]{->}(0,0)(0,-1.2)(6.5,1.25)
  \psplot[algebraic, linecolor=blue]{0}{6.5}{sin(x)}
\end{pspicture}
```



Dies ist nicht der Graph von $\sin(x)$!!

Beispiele mit trigLabels

```
\begin{pspicture}(-1,-1.5)(7,1.5)
  \psaxes[trigLabels, trigLabelBase=3]{->}(0,0)(0,-1.2)(6.5,1.25)
  \psplot[algebraic, linecolor=blue]{0}{6.5}{sin(x)}
\end{pspicture}
```



Dies ist nicht der Graph von $\sin(x)$!!

Richtige Einstellung von `xunit` ist nötig. Dafür werden Konstanten bereitgestellt:

$$\backslash\text{psPiFour} = 4\pi$$

$$\backslash\text{psPi} = \pi$$

$$\backslash\text{pstRadUnit} = \pi/3 \text{ cm}$$

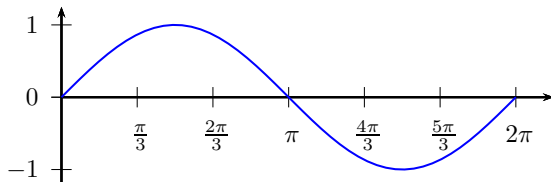
$$\backslash\text{psPiTwo} = 2\pi$$

$$\backslash\text{psPiH} = \pi/2$$

$$\backslash\text{pstRadUnitInv} = 3/\pi \text{ cm}$$

Beispiele mit trigLabels

```
\psset{xunit=\pstRadUnit}  
\begin{pspicture}(-1,-1.5)(7,1.5)  
  \psaxes[trigLabels, trigLabelBase=3]{->}(0,0)(0,-1.2)(6.5,1.25)  
  \psplot[xunit=1cm,algebraic,linecolor=blue]{0}{\psPiTwo}{sin(x)}  
\end{pspicture}
```



Richtige Einstellung von `xunit` ist nötig. Dafür werden Konstanten bereitgestellt:

$$\backslash\text{psPiFour} = 4\pi$$

$$\backslash\text{psPi} = \pi$$

$$\backslash\text{pstRadUnit} = \pi/3 \text{ cm}$$

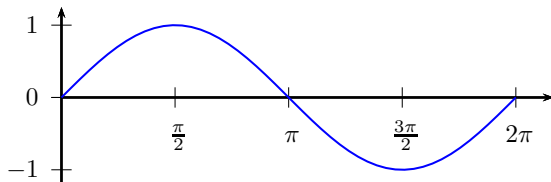
$$\backslash\text{psPiTwo} = 2\pi$$

$$\backslash\text{psPiH} = \pi/2$$

$$\backslash\text{pstRadUnitInv} = 3/\pi \text{ cm}$$

Beispiele mit trigLabels

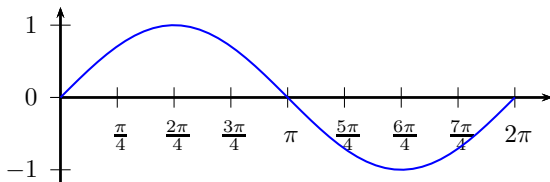
```
\psset{xunit=\pstRadUnit}  
\begin{pspicture}(-1,-1.5)(7,1.5)  
  \psaxes[trigLabels, trigLabelBase=2, dx=1.5]{->}%  
    (0,0)(0,-1.2)(6.5,1.25)  
  \psplot[xunit=1cm, algebraic, linecolor=blue]{0}{\psPiTwo}{sin(x)}  
\end{pspicture}
```



`dx` legt den physischen Abstand zweier aufeinander folgender Label fest.

Beispiele mit trigLabels

```
\psset{xunit=\pstRadUnit}  
\begin{pspicture}(-1,-1.5)(7,1.5)  
  \psaxes[trigLabels, trigLabelBase=4, dx=0.75]{->}%  
    (0,0)(0,-1.2)(6.5,1.25)  
  \psplot[xunit=1cm, algebraic, linecolor=blue]{0}{\psPiTwo}{sin(x)}  
\end{pspicture}
```



`dx` legt den physischen Abstand zweier aufeinander folgender Label fest.

2 Erste Schritte

Grundlagen

Achsenbeschriftung für trig. Funktionen

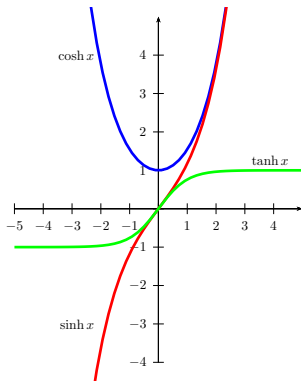
Erweiterungen

Das Paket pst-math

pst-math stellt weitere wichtige Funktionen für die Verwendung in UPN-Notation zur Verfügung, u.a.

SINH, COSH, TANH, ASINH, ACOSH, ATANH, EXP, GAUSS, GAMMA.

Ein Beispiel für die hyperbolischen Funktionen:



```
\usepackage{pst-math}
\psset{xunit=0.75}
\begin{pspicture*(-5.5,-4.5)(5,5.25)}
  \psaxes{->}(0,0)(-5,-4)(5,5)
  \psset{linewidth=2pt}
  \psplot[linecolor=blue]{-5}{5}{x COSH}
  \uput[180](-2,4){$\cosh x$}
  \psplot[linecolor=red]{-5}{5}{x SINH}
  \uput[180](-2,-3){$\sinh x$}
  \psplot[linecolor=green]{-5}{5}{x TANH}
  \uput[0](3,1.25){$\tanh x$}
\end{pspicture*} % (HV, Bsp. 28-01-01)
```

Das Paket pstricks-add

Kann als Bugfix-Paket angesehen werden für diverse Pakete, nach und nach werden Funktionalitäten in die Originalpakete aufgenommen. Sollte als letztes der PSTricks-Pakete geladen werden!

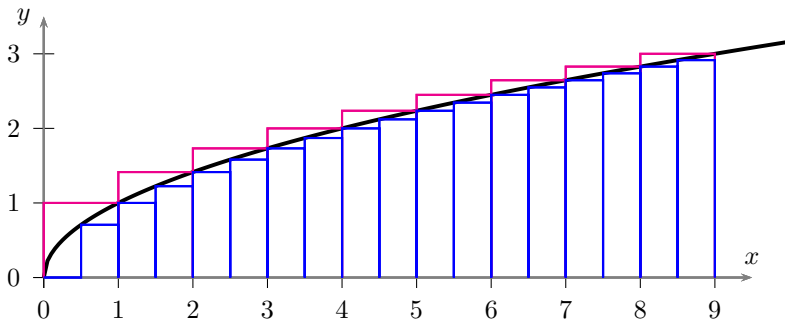
Stellt u.a. bereit:

`\begin{psgraph}... \end{psgraph}` (kombiniert `pspicture` und `\psaxes`)

<code>\psChart</code>	(für Tortendiagramme)
<code>\psHomothetie</code>	(für zentrische Streckungen)
<code>\psdice</code>	(für 2D-Ansichten von Würfelseiten)
<code>\psCancel</code>	(zum Durchstreichen von Objekten)
<code>\psStep</code>	(Ober- und Untersumme)
<code>\psplotTangent</code>	
<code>\psrotate</code>	

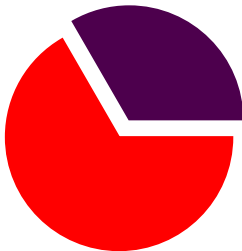
Beispiel zu psStep (mittlerweile in pst-plot aufgenommen)

```
\usepackage{pst-plot}
\begin{pspicture}(-0.5,-0.75)(10,4)
  \psaxes[linecolor=black!50]{->}(9.5,3.5)[$x$,90][$y$,180]
  \psplot[plotpoints=200,linewidth=1.5pt,algebraic]{0}{10}{sqrt(x)}
  \psStep[linecolor=magenta,StepType=upper](0,9){9}{x sqrt}
  \psStep[linecolor=blue,StepType=lower](0,9){18}{x sqrt}
\end{pspicture} % (HV), Bsp. 30-04-05
```



Beispiel zu psChart

```
\usepackage{pst-plot, pstricks-add}  
\begin{pspicture}(-3,-3)(3,3)  
  \psChart[chartColor=color]{45,90}{1}{2}  
\end{pspicture}
```



1 Einleitung

2 Erste Schritte

3 Das Paket `pst-func`

4 PSTricks und pdf $\LaTeX$

5 Zum Schluss...

③ Das Paket `pst-func`

Polynomfunktionen

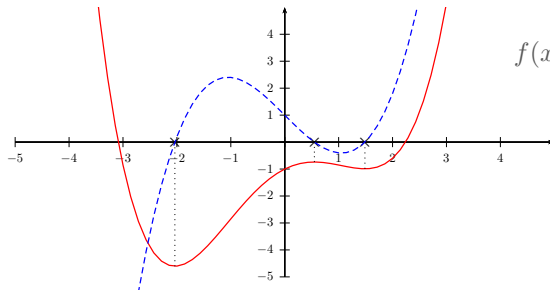
Integration mit `pst-func`

Verteilungsfunktionen

Darf's sonst noch was sein?

Der Befehl psPolynomial

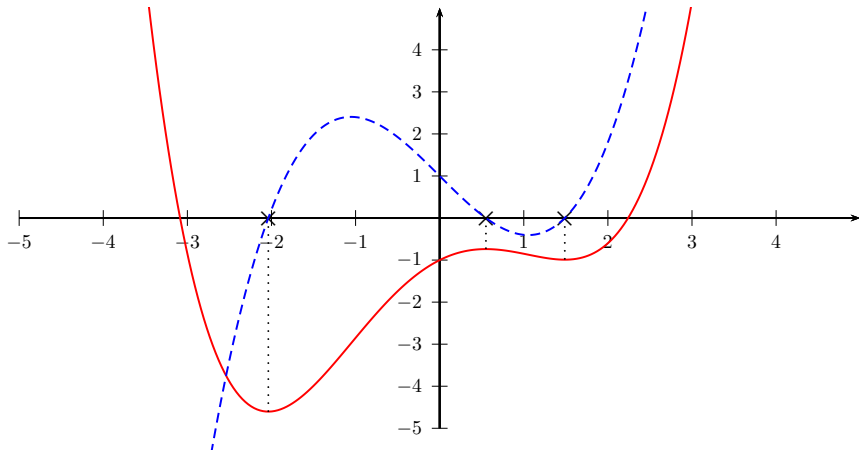
```
\usepackage{pst-func}
\def\coeff{-1 1 -1 0 0.15}
\psset{zeroLineStyle=dotted, linewidth=1pt, dotstyle=x, dotscale=2, %
  xunit=1.5cm, yunit=0.75cm, zeroLineWidth=1pt}
\begin{pspicture*}(-5.5,-5.5)(5,5)
  \psaxes{->}(0,0)(-5,-5)(5,5)%
  \psPolynomial[linecolor=red,coeff=\coeff][-4]{3}%
  \psPolynomial[markZeros,linecolor=blue,linestyle=dashed,%
    coeff=\coeff,Derivation=1,zeroLineTo=0][-4]{3}%
\end{pspicture*} %Dokumentation von pst-func
```



$$f(x) = -1 + x - x^2 + 0.15x^4$$

Beispiel zu psPolynomial

$$f(x) = -1 + x - x^2 + 0.15x^4.$$



③ Das Paket `pst-func`

Polynomfunktionen

Integration mit `pst-func`

Verteilungsfunktionen

Darf's sonst noch was sein?

pst-func stellt Plotfunktionen bereit für:

$$\backslash\text{psIntegral}(x) = \int_a^b f(x, t) dt$$

$$\backslash\text{psCumIntegral}(x) = \int_{x\text{Start}}^x f(t) dt$$

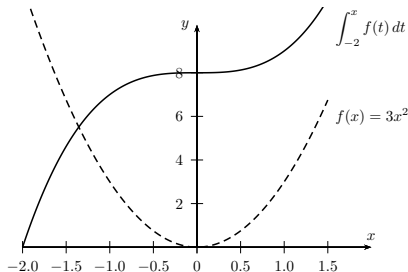
$$\backslash\text{psConv}(x) = \int_a^b f(t) g(x - t) dt.$$

Syntax:

```
\psIntegral[Options]{xStart}{xEnd}(a,b){function}  
\psCumIntegral[Options]{xStart}{xEnd}{function}  
\psConv[Options]{xStart}{xEnd}(a,b){function f}{function g}
```

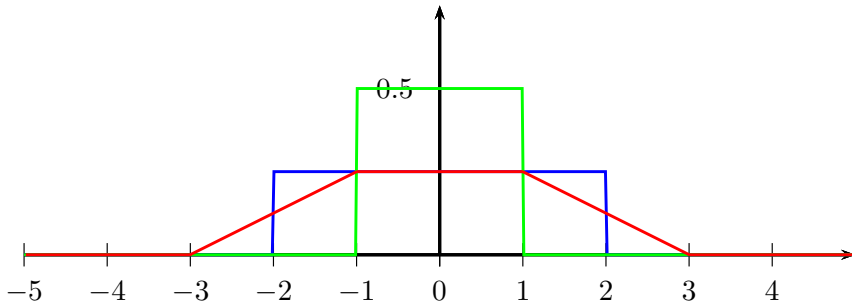
Beispiel zu psCumIntegral

```
\usepackage{pst-func}
\psset{xunit=2cm,yunit=0.5,plotpoints=200, Simpson=100}
\begin{pspicture*}[linewidth=1pt](-2.5,-1.2)(2.5,11)
  \psaxes[Dy=2,Dx=0.5]{->}(0,0)(-2,0)(2,10.2)[$x$,90][$y$,180]
  \psplot[linestyle=dashed]{-2}{1.5}{x dup mul 3 mul}
  \uput[0](1.5,6){$f(x)=3x^2$}
  \psCumIntegral{-2}{1.5}{dup mul 3 mul}
  \uput[0](1.5,10){$\displaystyle \int_{-2}^x f(t) dt$}
\end{pspicture*} %
```



Beispiel zu psConv

```
\psset{xunit=1cm,yunit=4cm, plotpoints=500}
\begin{pspicture}[linewidth=1pt](-5.5,-.2)(5,0.75)
  \psaxes[Dy=0.5]{->}(0,0)(-5,0)(5,0.75)
  \psplot[linecolor=blue]{-5}{5}{x abs 2 le {0.25}{0} ifelse}
  \psplot[linecolor=green]{-5}{5}{x abs 1 le {.5}{0} ifelse}
  \psConv[Simpson=1000,linecolor=red]{-5}{5}(-10,10)%
    {abs 2 le {0.25}{0} ifelse}{abs 1 le {.5}{0} ifelse}
\end{pspicture} %Dokumentation von pst-func
```



③ Das Paket `pst-func`

Polynomfunktionen

Integration mit `pst-func`

Verteilungsfunktionen

Darf's sonst noch was sein?

Die verfügbaren Verteilungsfunktionen in `pst-func`

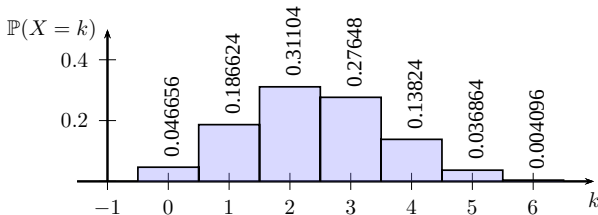
Das Paket stellt folgende Befehle für den Plot der entsprechenden Verteilung bereit:

<code>\psGauss</code>	<code>\psBinomial</code>	<code>\psPoisson</code>
<code>\psGammaDist</code>	<code>\psChiIIDist</code>	<code>\psTDist</code>
<code>\psFDist</code>	<code>\psBetaDist</code>	<code>\psCauchy</code>
<code>\psWeibull</code>	<code>\psVasicek</code>	

Beispiel: Binomialverteilung

`\psBinomial[Optionen]{N}{Wahrscheinlichkeit  $p$ }` zeichnet ein Diagramm der Binomialverteilung mit N Versuchen und Erfolgswahrscheinlichkeit $p \in (0, 1)$.

```
\psset[pst-func]{barwidth=1}
\psset{xunit=1cm,yunit=5cm}
\begin{pspicture}(-2.5,-0.15)(9,0.6)
  \psaxes[0x=-1,Dy=0.2,dy=0.2\psyunit]{->}(0,0)(-0.5,0)(8,0.5)%
    [$k$, -90] [$\mathbb{P}(X=k)$, 180]
  \rput{0}(1,0){%
    \psBinomial[markZeros, fillstyle=solid, fillcolor=blue!50,%
      opacity=0.3, printValue]{6}{0.4}}
\end{pspicture}
```

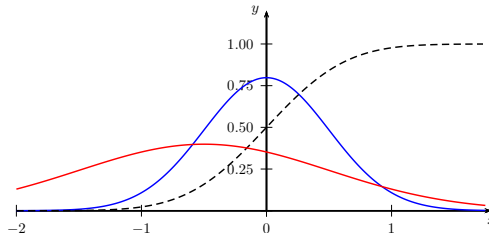


Beispiel: Normalverteilung

`\psGauss[Optionen]{ $x_0$ }{ $x_1$ }` zeichnet die Dichte der Normalverteilung im Intervall $[x_0, x_1]$. Optionen sind `sigma` (Vorgabe 0.5) und `mue` (Vorgabe 0).

`\psGaussI` zeichnet die Verteilungsfunktion, gleiche Syntax.

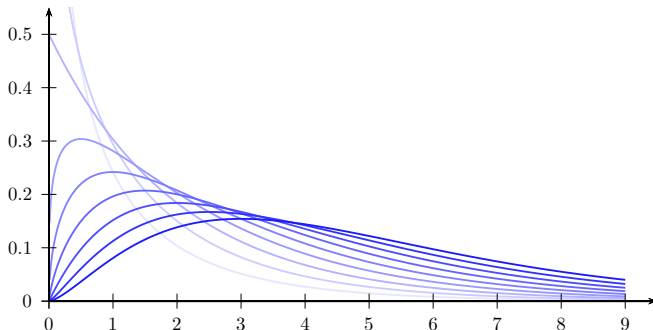
```
\psset{yunit=4cm,xunit=3, linewidth=1pt}
\begin{pspicture}(-2.2,-0.15)(2,1.25)
  \psaxes[Dy=0.25]{->}(0,0)(-2,0)(1.8,1.2)[$x$,270][$y$,180]
  \psGauss[linecolor=blue]{-2}{1.75}
  \psGaussI[linestyle=dashed]{-2}{1.75}
  \psGauss[linecolor=red, sigma=1, mue=-0.5]{-2}{1.75}
\end{pspicture} % Dokumentation von pst-func
```



Beispiel: χ^2 -Verteilung

`\psChiIIDist[Optionen]{ $x_0$ }{ $x_1$ }` zeichnet die Dichte der χ^2 -Verteilung im Intervall $[x_0, x_1]$. Option ist `nue=<real>`.

```
\psset{xunit=1.2cm,yunit=10cm,plotpoints=200,linewidth=1pt}
\begin{pspicture*}(-0.75,-0.1)(9.5,0.55)
\multido{\rnue=0.5+0.5,\ibblue=0+10}{10}{%
  \psChiIIDist[linecolor=blue!\ibblue,nue=\rnue]{0.01}{9}}
\psaxes[Dy=0.1]{->}(0,0)(9.5,.55)
\end{pspicture*} % Dokumentation von pst-func
```



3 Das Paket `pst-func`

Polynomfunktionen

Integration mit `pst-func`

Verteilungsfunktionen

Darf's sonst noch was sein?

Weitere Funktionen in `pst-func`

`\psBernstein`

`\psVolume`

`\psBessel`

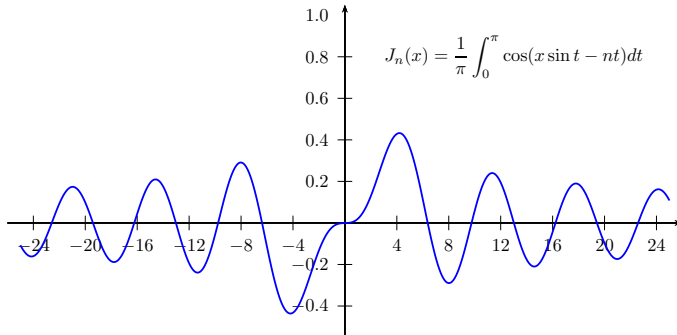
`\psSi`

`\psFourier`

`\psCi`

Beispiel: Besselfunktion

```
\psset{xunit=0.25,yunit=4}
\begin{pspicture}(-27,-.55)(27,1.1)
  \rput(13,0.8){$\displaystyle J_n(x)=\frac{1}{\pi}\%
    \int_0^\pi\cos(x\sin t-nt)dt$}
  \psaxes[Dy=0.2,Dx=4]{->}(0,0)(-26,-.6)(26,1.05)
  \psset{linewidth=1pt}
  \psBessel[linecolor=blue]{3}{-25}{25}
\end{pspicture} % (HV) Bsp. 28-02-14
```



- 1 Einleitung
 - 2 Erste Schritte
 - 3 Das Paket `pst-func`
 - 4 PSTricks und pdf $\LaTeX$**
 - 5 Zum Schluss...
-

Ziel

`dokument.tex` enthält eine Grafik, die mit PSTricks erstellt wurde.
Aus `dokument.tex` soll `dokument.pdf` erzeugt werden.

Erste Variante: Falls `dokument.tex` mit `latex` kompiliert werden kann, dann:

```
latex dokument.tex    %<---- erzeugt dokument.dvi
dvips dokument.dvi    %<---- erzeugt dokument.ps
ps2pdf dokument.ps    %<---- erzeugt dokument.pdf
```

Dabei steht der PSTricks-Code direkt in `dokument.tex`

Skalieren der Grafik möglich mit dem Befehl `\resizebox` aus dem Paket `graphicx`

Beispieldokument für erste Variante

Ein Beispiel für Variante 1

16. Januar 2014

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Hier eine Abbildung in einer Gleitumgebung mit Beschriftung:

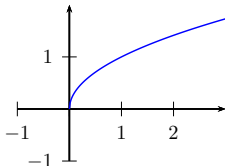


Abbildung 1: Mein erster Funktionsplot PSTricks

```
\documentclass{scrartcl}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[ngerman]{babel}% evtl. weitere Pakete...
\usepackage{lipsum, graphicx}
%
\usepackage{psticks, pst-plot}
%
\title{Ein Beispiel f\"ur Variante 1}
%
\parskip0.5em
\begin{document}
\maketitle

\lipsum[1]

Hier eine Abbildung in einer Gleitumgebung
mit Beschriftung:
\begin{figure}[h]\centering
\resizebox{0.7\linewidth}{!}{%
\begin{pspicture}(-1.5,-1.5)(3.5,2.5)
\psaxes{->}(0,0)(-1,-1)(3,2)
\psplot[linecolor=blue,algebraic,
plotpoints=200]{0}{3}{sqrt(x)}
\end{pspicture}
}%<-- von resizebox
\caption{Mein erster Funktionsplot PSTricks}
\end{figure} %%--> Leerzeile danach wichtig

\lipsum[1]
\end{document}
```

Falls dvips nicht zur Verfügung steht (z.B. wenn dokument.tex nur mit pdflatex kompilierbar ist), gibt es zwei weitere Varianten.

Zweite Variante: PSTricks-Code auslagern in grafik.tex und die entstehende Grafik in dokument.tex einbinden:

```
latex grafik.tex
dvips grafik.dvi
ps2pdf grafik.ps
pdfcrop grafik.pdf grafik-crop.pdf % <--- entfernt weisse Ränder
%                                   und schneidet pdf zu
pdflatex dokument.tex
```

Skalieren möglich über Optionen des `\includegraphics` Befehl.

Beispiel für zweite Variante

Inhalt von grafik.tex:

```
\documentclass{scrartcl}
\usepackage{pstricks, pst-plot}
\begin{document}
\pagestyle{empty}
\begin{pspicture}(-1.5,-1.5)(3.5,2.5)
  \psaxes{->}(0,0)(-1,-1)(3,2)
  \psplot[linecolor=blue,algebraic]{0}{3}{sqrt(x)}
\end{pspicture}
\end{document}
```

Beispiel für zweite Variante

Ein Beispiel für Variante 2

16. Januar 2014

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Hier eine Abbildung in einer Gleitumgebung mit Beschriftung:

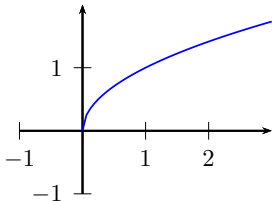


Abbildung 1: Mein erster Funktionsplot PSTricks

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut,

```
\documentclass{scrartcl}
\usepackage[T1]{fontenc} \usepackage{lipsum, graphicx}
\usepackage[ngerman]{babel}% evtl. weitere Pakete...
%
\title{Ein Beispiel f\"ur Variante 2}
\begin{document}
\maketitle
\parskip0.5em
\lipsum[1]
```

Hier eine Abbildung in einer Gleitumgebung mit Beschriftung:

```
\begin{figure}[h]\centering
\includegraphics[width=0.7\linewidth]{bsp2-grafik-crop}
\caption{Mein erster Funktionsplot PSTricks}
\end{figure} %%--> Leerzeile danach wichtig

\lipsum[1]
\end{document}
```

Dritte Variante: Das Paket `pst-pdf`. Sorgt dafür, dass im `latex`-Lauf nur PostScript Code berücksichtigt wird. Erzeugt Grafiken, die dann automatisch im `pdflatex`-Lauf eingebunden werden.

Vorgehen:

```
latex dokument.tex
dvips -o dokument-pics.ps document.dvi
ps2pdf dokument-pics.ps

pdflatex dokument.tex
```

Der Suffix `-pics` ist die Standardeinstellung von `pst-pdf`.
Ändert man nichts mehr an den Grafiken, dann reicht
`pdflatex dokument.tex`

Es gibt auch einen Befehl, der alles erledigt:

```
ps4pdf dokument.tex
```

Beispiel für Variante 3

Ein Beispiel für Variante 3

16. Januar 2014

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Hier eine Abbildung in einer Gleitumgebung mit Beschriftung:

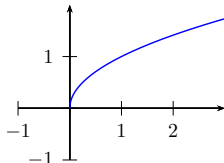


Abbildung 1: Mein erster Funktionsplot PSTricks

```
\documentclass{scrartcl}
\usepackage[T1]{fontenc} \usepackage{lipsum, graphicx}
\usepackage[ngerman]{babel}% evtl. weitere Pakete...
\usepackage{pstricks, pst-plot, pst-pdf}
%
\title{Ein Beispiel f\"ur Variante 3}
\begin{document}
\maketitle
\parskip0.5em
\lipsum[1]
```

Hier eine Abbildung in einer Gleitumgebung mit Beschriftung:

```
\begin{figure}[h]\centering
\resizebox{0.7\linewidth}{!}{%
  \begin{pspicture}(-1.5,-1.5)(3.5,2.5)
    \psaxes{->}(0,0)(-1,-1)(3,2)
    \psplot[linecolor=blue,algebraic,
      plotpoints=200]{0}{3}{sqrt(x)}
  \end{pspicture}
} %<-- von resizebox
\caption{Mein erster Funktionsplot PSTricks}
\end{figure} %%<-- Leerzeile danach wichtig

\lipsum[1]
\end{document}
```

- 1 Einleitung
 - 2 Erste Schritte
 - 3 Das Paket `pst-func`
 - 4 PSTricks und `pdflatex`
 - 5 Zum Schluss...**
-

⑤ Zum Schluss...

Weitere interessante Features

Aus der Galerie...

- Das Paket `pst-node`. Unterstützung für das Setzen, Speichern und Verbinden von Knoten.

- Das Paket `pst-node`. Unterstützung für das Setzen, Speichern und Verbinden von Knoten.
- Das Paket `pst-tree`. Basierend auf `pst-node` bietet es Makros für die Erstellung von Bäumen unterschiedlicher Art.

- Das Paket `pst-node`. Unterstützung für das Setzen, Speichern und Verbinden von Knoten.
- Das Paket `pst-tree`. Basierend auf `pst-node` bietet es Makros für die Erstellung von Bäumen unterschiedlicher Art.
- Das Paket `pst-3dplot`. Bietet 3D-Parallelprojektion von Funktionen.

- Das Paket `pst-node`. Unterstützung für das Setzen, Speichern und Verbinden von Knoten.
- Das Paket `pst-tree`. Basierend auf `pst-node` bietet es Makros für die Erstellung von Bäumen unterschiedlicher Art.
- Das Paket `pst-3dplot`. Bietet 3D-Parallelprojektion von Funktionen.
- Das Paket `pst-eucl` für euklidische Geometrie.

- Das Paket `pst-node`. Unterstützung für das Setzen, Speichern und Verbinden von Knoten.
- Das Paket `pst-tree`. Basierend auf `pst-node` bietet es Makros für die Erstellung von Bäumen unterschiedlicher Art.
- Das Paket `pst-3dplot`. Bietet 3D-Parallelprojektion von Funktionen.
- Das Paket `pst-eucl` für euklidische Geometrie.
- Das Paket `pst-fractal` zum Zeichnen von Fraktalen.

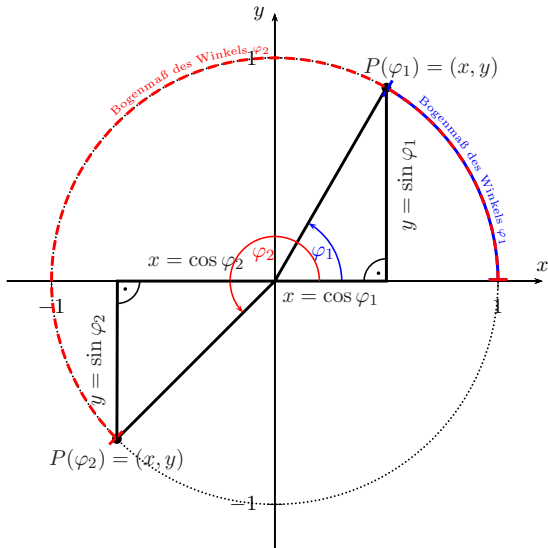
- Das Paket `pst-node`. Unterstützung für das Setzen, Speichern und Verbinden von Knoten.
- Das Paket `pst-tree`. Basierend auf `pst-node` bietet es Makros für die Erstellung von Bäumen unterschiedlicher Art.
- Das Paket `pst-3dplot`. Bietet 3D-Parallelprojektion von Funktionen.
- Das Paket `pst-eucl` für euklidische Geometrie.
- Das Paket `pst-fractal` zum Zeichnen von Fraktalen.
- Aktivierung von `\SpecialCoor`. Zusätzliche Möglichkeiten zur Eingabe von Koordinaten.

5 Zum Schluss...

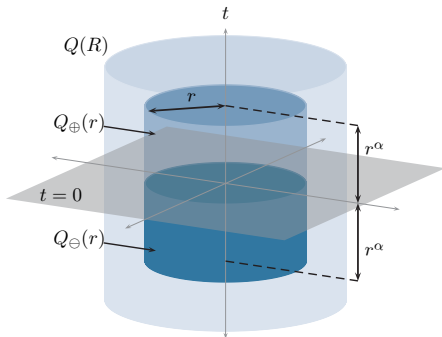
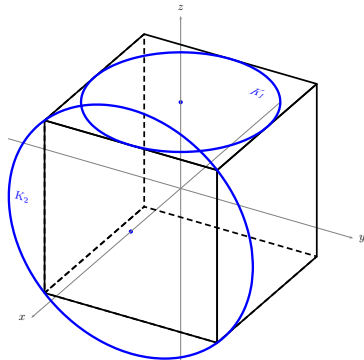
Weitere interessante Features

Aus der Galerie...

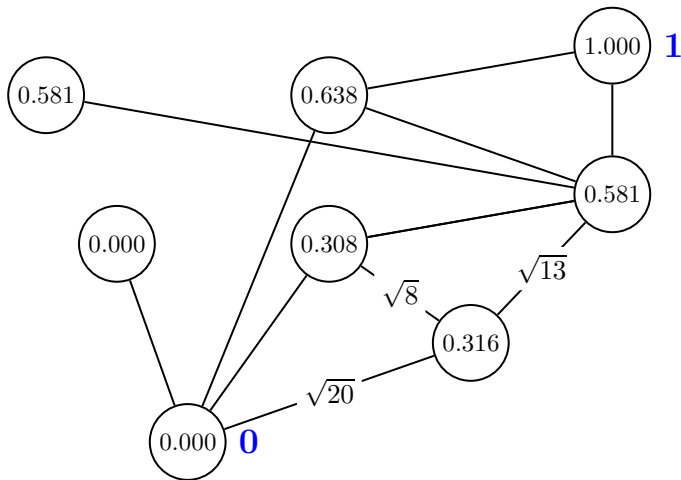
Mit pst-eucl und pst-text



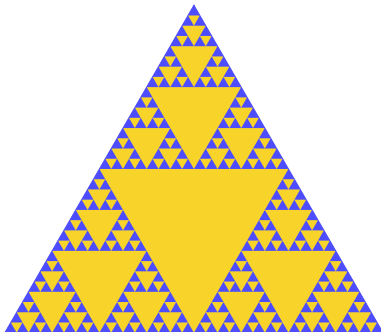
Parallelprojektionen mit pst-3dplot



Mit pst-node



Aus der Dokumentation von pst-fractal



```
\usepackage{pst-fractal}  
\begin{pspicture}(5,5)  
\psSier[linecolor=blue!70,%  
        fillcolor=fak_gelb](0,0){5cm}{5}  
\end{pspicture}
```

Ende des Vortrags

Vielen Dank für Ihre
Aufmerksamkeit!

Ende des Vortrags



**Vielen Dank für Ihre
Aufmerksamkeit!**