

Kapitel 21

pst-grad und pst-slope: Farbverläufe und Schatten

21.1 pst-grad	341
21.2 pst-slope	344
21.3 pst-blur: Verschwommene Schatten	352

pst-grad ist ebenfalls eines der älteren und kleineren Pakete. Es stellt nur einen Füllstil zur Verfügung, der wie in Kapitel 7 auf Seite 93 beschrieben, anzuwenden ist. Grundsätzlich ließe sich ein Farbverlauf auch mit den bereits von PSTricks bekannten Makros erstellen, dennoch bietet die Anwendung von pst-grad Vorteile, da man sich nicht um die Berechnungen der Zwischenfarbwerte kümmern muss. Unterstützt werden die Farbmodelle HSB und RGB. Das Paket pst-slope füllt die Lücke, die durch pst-grad nicht ausgefüllt wird. Dazu gehören zum einen kreisförmige Farbverläufe und zum anderen mehrere, frei zu definierende Farben bei einem Verlauf zu nutzen. Zusätzlich unterstützt pst-slope auch lineare Farbverläufe, sodass man grundsätzlich kein weiteres Paket laden muss.

21.1 pst-grad

Die Farben für gradbegin und gradend sollten als RGB-Farbe definiert werden, da eine einwandfreie Funktion für CMYK oder Graustufen nicht unbedingt gewährleistet ist.  RGB-Farben
ConTeXt Anwender ändern die Farbe mit `\definecolor[gradbegin][r=0,g=0,b=1]`.

21.1.1 Parameter

Tabelle 21.1 zeigt eine Zusammenstellung der speziellen, für pst-grad geltenden Parameter. Alle Parameter stehen nur zur Verfügung, wenn als Füllstil gradient verwendet wird: `fillstyle=gradient`  Füllstil

Name	Werte	Vorgabe
gradbegin	Farbe	gradbegin
gradend	Farbe	gradend
gradlines	Wert	500
gradmidpoint	Wert	0.9
gradangle	Winkel	0
gradientHSB	Boolean	false
GradientCircle	Boolean	false
GradientScale	Wert	1.0
GradientPos	(x,y)	(0,0)

Tabelle 21.1: Zusammenfassung aller Parameter für pst-grad.

gradbegin

Bezeichnet sowohl den Parameter als auch den Namen der Anfangsfarbe, was hier etwas verwirrend ist. Die interne Definition folgt wieder der PSTricks-Syntax, sollte jedoch für reine L^AT_EX-Anwender nicht benutzt werden. Die color/xcolor-Syntax ist hier zu bevorzugen (siehe Abschnitt 2.1 auf Seite 10).

```
\newrgbcolor{gradbegin}{0 .1 .95} % interne Vorgabe
```

Somit kann man diese Anfangsfarbe zum einen über eine Neudefinition dieser Farbe oder durch Zuweisung über den Parameter geändert werden.

```
\newrgbcolor{gradbegin}{0 0 1} % veraltete TeX-Syntax
\definecolor{gradbegin}{rgb}{0, 0, 1} % benötigt color/xcolor Paket
\psset{gradbegin=blue}
```



```
\usepackage{pstricks,pst-grad}

\begin{pspicture}(5,2.25)
\psframe[fillstyle=gradient,gradbegin=white](5,1)
\definecolor{gradbegin}{rgb}{0, 1, 1}
\psframe[fillstyle=gradient](0,1.25)(5,2.25)
\end{pspicture}
```

21-01-1

gradend

Dieser Parameter ist **nicht** das Gegenstück zu gradbegin, denn es ist die Farbe, die am relativen Punkt gradmidpoint erreicht wird. In jedem Fall ist sie wieder wie gradbegin doppeldeutig.

```
\newrgbcolor{gradend}{0 1 1} % Vorgabe
```

Eine Änderung kann wieder unterschiedlich erfolgen.

```
\newrgbcolor{gradend}{1 0 0} % veraltete TeX-Syntax
\definecolor{gradend}{rgb}{1, 0, 0} % benötigt color/xcolor Paket
\psset{gradend=red}
```

21-01-2



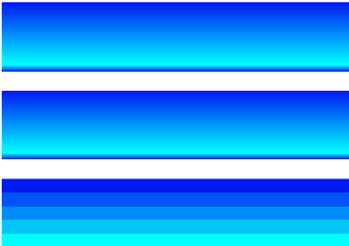
```
\usepackage{pstricks,pst-grad}

\begin{pspicture}(5,2.25)
\psframe[fillstyle=gradient,
gradend=white](5,1)
\definecolor{gradend}{rgb}{1, 0, 0}
\psframe[fillstyle=gradient](0,1.25)(5,2.25)
\end{pspicture}
```

gradlines

Ein Farbverlauf ist nichts anderes als ein Aneinanderreihen von farbigen Linien. Deren Dicke ist letztlich nur von der Auflösung des Bildschirms bzw. Druckers abhängig. Da dies jedoch sehr anwenderspezifisch ist, lässt `pst-grad` jede Anzahl an Linien zu, deren Zahl über `gradlines` geändert werden kann.

21-01-3



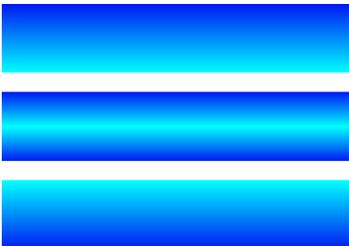
```
\usepackage{pstricks,pst-grad}

\begin{pspicture}(5,3.5)
\psset{fillstyle=gradient,linestyle=none}
\psframe[gradlines=5](5,1)
\psframe(0,1.25)(5,2.25)
\psframe[gradlines=1000](0,2.5)(5,3.5)
\end{pspicture}
```

gradmidpoint

Bezeichnet den relativen Punkt, bei dem die Farbe `gradend` erreicht wird. Danach geht es in umgekehrter Reihenfolge weiter.

21-01-4

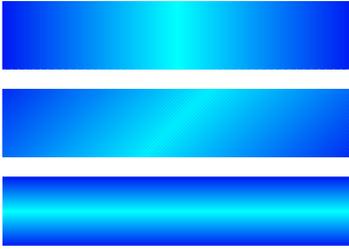


```
\usepackage{pstricks,pst-grad}

\begin{pspicture}(5,3.5)
\psset{fillstyle=gradient,linestyle=none}
\psframe[gradmidpoint=0](5,1)
\psframe[gradmidpoint=0.5](0,1.25)(5,2.25)
\psframe[gradmidpoint=1](0,2.5)(5,3.5)
\end{pspicture}
```

gradangle

`gradangle` legt den Steigungswinkel der Geraden fest.



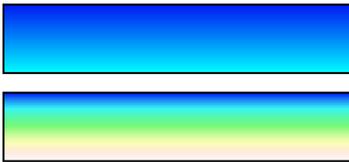
```
\usepackage{pstricks,pst-grad}

\begin{pspicture}(5,3.5)
\psset{fillstyle=gradient,linestyle=none,
gradmidpoint=0.5}
\psframe[gradangle=0](5,1)
\psframe[gradangle=45](0,1.25)(5,2.25)
\psframe[gradangle=90](0,2.5)(5,3.5)
\end{pspicture}
```

21-01-5

gradientHSB

gradientHSB veranlasst die Verwendung des HSB-Farbschemas, falls diese boolsche Variable auf true gesetzt wird.



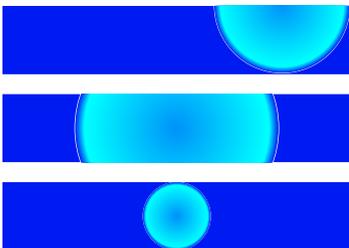
```
\usepackage{pstricks,pst-grad}

\begin{pspicture}(5,2.25)
\psset{gradientHSB=true,gradmidpoint=1}
\psframe[fillstyle=gradient,gradend=white](5,1)
\newrgbcolor{gradend}{1 0 0}
\psframe[fillstyle=gradient](0,1.25)(5,2.25)
\end{pspicture}
```

21-01-6

GradientCircle, GradientScale und GradientPos

Mit der Option GradientCircle lassen sich kreisförmige Farbverläufe erstellen, deren Radius mit GradientScale und Mittelpunkt mit GradientPos beeinflusst werden kann. Die Angabe der Koordinaten bezieht sich auf das zugrundeliegende Koordinatensystem, welches in der Regel durch die pspicture Umgebung gegeben ist.



```
\usepackage{pstricks,pst-grad}

\begin{pspicture}(5,3.5)
\psset{fillstyle=gradient,linestyle=none}
\psframe[GradientCircle=true](5,1)%
\psframe[GradientCircle=true,
GradientScale=3](0,1.25)(5,2.25)%
\psframe[GradientCircle=true,GradientScale=2,%
GradientPos={(4,3.5)}](0,2.5)(5,3.5)%
\end{pspicture}
```

21-01-7

21.2 pst-slope

21.2.1 Füllstile

Tabelle 21.2 zeigt eine Aufstellung aller verfügbaren neuen Füllstile, die grundsätzlich über den Parameter fillstyle (\Rightarrow Abschnitt 7.1.2 auf Seite 94) zuzuweisen sind, beispielsweise fillstyle=slope. Bei linearer Interpolation erfolgt die Farbänderung von

beispielsweise Rot $(1, 0, 0)$ nach Grün $(0, 1, 0)$ linear mit $(1 - dz, 0, 0)$ und $(0, dz, 0)$, wobei dz die Schrittweite ($0 < dz < 1$) bezeichnet. In vielen Fällen genügt dieser Übergang den Ansprüchen, doch erscheint es manchmal sinnvoller, nicht linear zu interpolieren, sondern über Zwischenfarben zu gehen. In diesem Beispiel über Gelb $(1, 1, 0)$ und Braun $(0.5, 0.5, 0)$. Diese Methode wird grundsätzlich dann angewendet, wenn die Pluralform des Füllstils gewählt wird (siehe Tabelle 21.2).

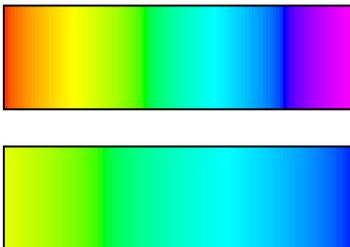
Name	Erklärung
slope	lineare Interpolation
slopes	zusätzlich mit Zwischenfarben
ccslope	kreisförmig mit linearer Interpolation
ccslopes	zusätzlich mit Zwischenfarben
radslope	radial mit linearer Interpolation
radslopes	zusätzlich mit Zwischenfarben

Tabelle 21.2: Zusammenfassung aller durch `pst-slope` definierten neuen Füllstile.

slope und slopes

Der Füllstil `slope` entspricht vollständig dem Stil `gradient` des Pakets `pst-grad` (\Rightarrow Abschnitt 21.1 auf Seite 341), während `slopes` nur von `pst-slope` unterstützt wird.

21-02-1

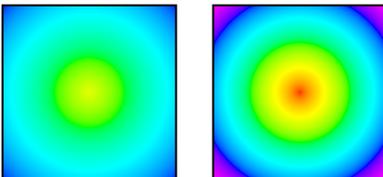


```
\usepackage{pstricks,pst-slope}
\begin{pspicture}(5,3.5)
  \psframe[fillstyle=slope](5,1.5)
  \psframe[fillstyle=slopes](0,2)(5,3.5)
\end{pspicture}
```

ccslope und ccslopes

Die Füllstile ermöglichen kreisförmige Farbverläufe.

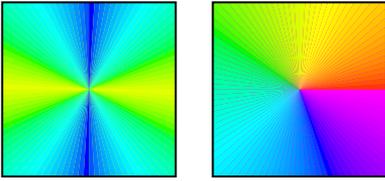
21-02-2



```
\usepackage{pstricks,pst-slope}
\begin{pspicture}(5.5,2.5)
  \psframe[fillstyle=ccslope](2.5,2.5)
  \psframe[fillstyle=ccslopes](3,0)(5.5,2.5)
\end{pspicture}
```

radslope und radslopes

Die Füllstile ermöglichen radiale Farbverläufe.



```
\usepackage{pstricks,pst-slope}

\begin{pspicture}(5.5,2.5)
  \psframe[fillstyle=radslope](2.5,2.5)
  \psframe[fillstyle=radslopes](3,0)(5.5,2.5)
\end{pspicture}
```

21-02-3

21.2.2 Parameter

Tabelle 21.3 zeigt eine Aufstellung aller verfügbaren Parameter. Zu beachten ist, dass die im folgenden angegebenen Farbsetzungen der PSTricks-Syntax folgen, wie sie im Paket selbst festgelegt sind. Für L^AT_EX-Anwender sollte unbedingt die color/xcolor-eigene Syntax angewendet werden (siehe Abschnitt 2.1 auf Seite 10).

Tabelle 21.3: Zusammenfassung aller Parameter für pst-slope.

<i>Name</i>	<i>Werte</i>	<i>Vorgabe</i>
slopebegin	<i>Farbe</i>	slopebegin
slopeend	<i>Farbe</i>	slopeend
slopecolors	<i>Farbliste</i>	0.0 1 0 0 0.4 0 1 0 0.8 0 0 1 1.0 0 1 0
slopesteps	<i>Wert</i>	100
slopecenter	<i>x y</i>	0.5 0.5
slopeangle	<i>Winkel</i>	0
sloperadius	<i>Wert Einheit</i>	0
fading	<i>Boolean</i>	false
startfading	<i>Wert</i>	0
endfading	<i>Wert</i>	1

slopebegin

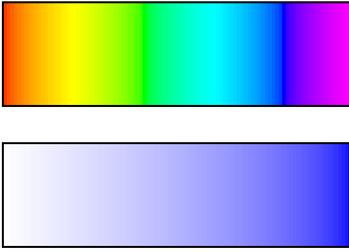
slopebegin bezeichnet sowohl den Parameter als auch den Namen der Anfangsfarbe, was hier etwas verwirrend ist.

```
\newrgbcolor{slopebegin}{0.9 1 0} % interne Vorgabe
```

Somit kann man diese Anfangsfarbe über eine Neudefinition dieser Farbe oder durch Zuweisung über den Parameter ändern.

```
\newrgbcolor{slopebegin}{0 0 1} % veraltete TEX-Syntax
\definecolor{slopebegin}{rgb}{0, 0, 1} % benötigt color/xcolor-Paket
\psset{slopebegin=blue}
```

21-02-4



```
\usepackage{pstricks,pst-slpe}

\begin{pspicture}(5,3.5)
  \psframe[fillstyle=slope,
    slopebegin=white](5,1.5)
  \definecolor{slopebegin}{rgb}{0,1,1}
  \psframe[fillstyle=slopes](0,2)(5,3.5)
\end{pspicture}
```

slopeend

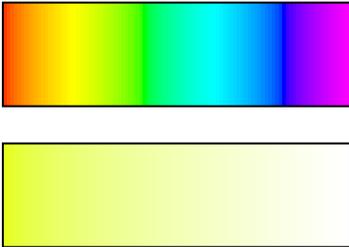
slopeend ist **nicht** das Gegenstück zu slopebegin, denn es ist die Farbe, die am Ende erreicht wird. In jedem Fall ist sie wieder wie slopebegin doppeldeutig.

```
\newrgbcolor{slopeend}{1 0 0} % Vorgabe
```

Eine Änderung kann wieder unterschiedlich erfolgen.

```
\newrgbcolor{slopeend}{1 0 0} % veraltete TEX-Syntax
\definecolor{slopeend}{rgb}{1,0,0} % benötigt color/xcolor Paket
\psset{slopebegin=red}
```

21-02-5



```
\usepackage{pstricks,pst-slpe}

\begin{pspicture}(5,3.5)
  \psframe[fillstyle=slope,
    slopeend=white](5,1.5)
  \definecolor{slopeend}{rgb}{1,0,0}
  \psframe[fillstyle=slopes](0,2)(5,3.5)
\end{pspicture}
```

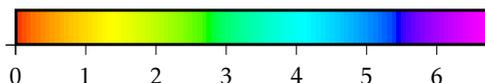
slopecolors

slopecolors definiert die Farbliste bei nicht-linearer Verteilung mittels slopebegin und slopeend. slopecolors wird immer dann angewendet, wenn die Plural-Fülloption aus Tabelle 21.2 gewählt wurde. Die Farbliste setzt sich zusammen aus einer eindimensionalen Koordinate und der zugehörigen Farbe. Die Liste wird abgeschlossen mit der Anzahl der Stützpunkte (Farben).

```
\usepackage{pstricks,pst-plot,pst-slpe}

\begin{pspicture}(0,-0.25)(\linewidth,0.75cm)
  \psaxes[axesstyle=frame,tickstyle=bottom](\linewidth,0.5cm)
  \psframe[fillstyle=slopes](\linewidth,0.5cm)
\end{pspicture}
```

21-02-6



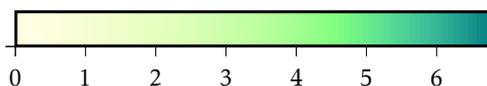
Obiges Beispiel zeigt den vorgegebenen Verlauf, der einen Regenbogen darstellt.

```
0.0 1 0 0 % Startfarbe Rot
0.4 0 1 0 % Zwischenfarbe bei 40% gleich Grün
0.8 0 0 1 % Zwischenfarbe bei 80% gleich Blau
1.0 1 0 1 % Endfarbe Violett
4      % Anzahl der Stufen
```

Die vier Stützpunkte werden auf die komplette x -Breite des zu füllenden Bereiches bezogen und gegebenenfalls angepasst. Für obiges Beispiel kann man direkt die x -Achse als Maßstab für die Farbpunkte wählen.

```
\usepackage{pstricks,pst-plot,pst-slope}

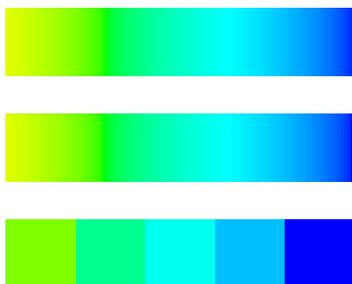
\begin{pspicture}(0,-0.25)(\linewidth,0.75cm)
\psset{slopecolors= 0.0  1.0 1.0 0.9
                  8.5  0.5 1.0 0.5
                  12.5 0.0 0.5 0.5
                  3}
\psaxes[axesstyle=frame,tickstyle=bottom](\linewidth,0.5cm)
\psframe[fillstyle=slopes](\linewidth,0.5cm)
\end{pspicture}
```



21-02-7

slopesteps

Ein Farbverlauf ist nichts anderes als ein Aneinanderreihen von farbigen Linien. Deren Dicke ist letztlich nur von der Auflösung des Bildschirms bzw. Druckers abhängig. Da dies jedoch sehr anwenderspezifisch ist, lässt `pst-slope` jede Anzahl von Schritten zu, die über `slopesteps` geändert werden können.



```
\usepackage{pstricks,pst-slope}

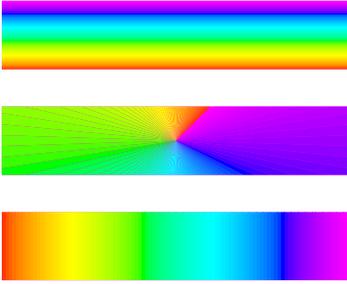
\begin{pspicture}(5,4)
\psset{fillstyle=slope,linestyle=none}
\psframe[slopesteps=5](5,1)
\psframe(0,1.5)(5,2.5)
\psframe[slopesteps=1000](0,3)(5,4)
\end{pspicture}
```

21-02-8

slopeangle

`slopeangle` legt den Steigungswinkel der Geraden fest. Bei dem Füllstil `radslope[s]` wird entsprechend rotiert und bei `ccslope[s]` ist dieser Parameter wirkungslos.

21-02-9



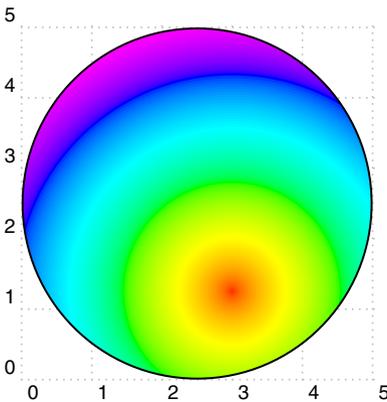
```
\usepackage{pstricks,pst-slpe}

\begin{pspicture}(5,4)
  \psset{fillstyle=slopes,linestyle=none}
  \psframe[slopeangle=0](5,1)
  \psframe[fillstyle=radslopes,
    slopeangle=45](0,1.5)(5,2.5)
  \psframe[slopeangle=90](0,3)(5,4)
\end{pspicture}
```

slopecenter

`slopecenter` legt das Zentrum für die Füllstile `radslope[s]` und `ccslope[s]` fest. Die Koordinatenangaben beziehen sich relativ auf die aktuelle Bounding-Box. Um sicherzustellen, dass diese eindeutig definiert ist, empfiehlt es sich, `\pscustom` (siehe Abschnitt 11) zu benutzen. In beiden folgenden Beispielen wird das Zentrum auf 60 % der Breite und 25 % der Höhe gesetzt.

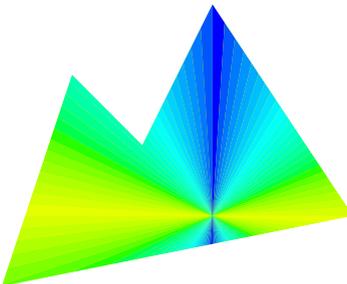
21-02-10



```
\usepackage{pstricks,pst-slpe}

\begin{pspicture}[showgrid](5,5)
  \pscustom[fillstyle=ccslopes,
    slopecenter={0.6 0.25}]{%
    \pscircle(2.5,2.5){2.5}}
\end{pspicture}
```

21-02-11



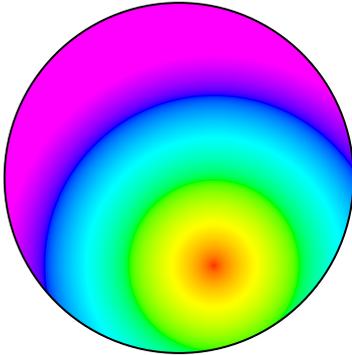
```
\usepackage{pstricks,pst-slpe}

\begin{pspicture}(5,4)
  \pscustom[fillstyle=radslope,
    linestyle=none,slopecenter=0.6 0.25]{%
    \psline(0,0)(1,3)(2,2)(3,4)(5,1)(0,0)}
\end{pspicture}
```

sloperadius

Grundsätzlich wird der Farbverlauf bei kreisförmigem bzw. radialem Verlauf ausgehend vom Zentrum erstellt, beginnend mit der ersten Farbe bis zum Rand der Bounding-Box und endend mit der letzten Farbe, sodass der gesamte Bereich gefüllt wird. Dies kann mit

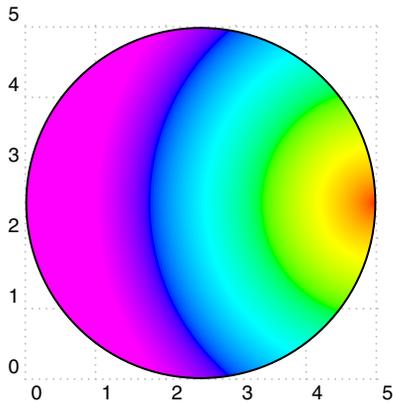
dem Parameter `sloperadius` beeinflussen. Dieser gibt den relativen maximalen Radius vor.



```
\usepackage{pstricks,pst-slpe}
```

21-02-12

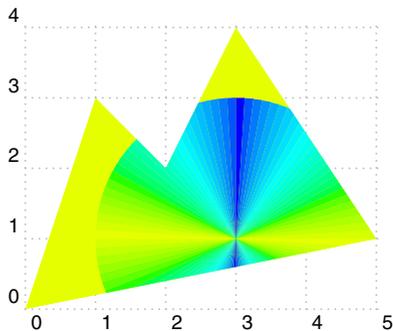
```
\begin{pspicture}(5,5)
  \pscustom[fillstyle=ccslopes,
    slopecenter={0.6 0.25},
    sloperadius=3cm]{%
    \pscircle(2.5,2.5){2.5}}
\end{pspicture}
```



```
\usepackage{pstricks,pst-slpe}
```

21-02-13

```
\begin{pspicture}[showgrid](5,5)
  \pscustom[fillstyle=ccslopes,
    slopecenter={1 0.5},
    sloperadius=4cm]{%
    \pscircle(2.5,2.5){2.5}}
\end{pspicture}
```



```
\usepackage{pstricks,pst-slpe}
```

21-02-14

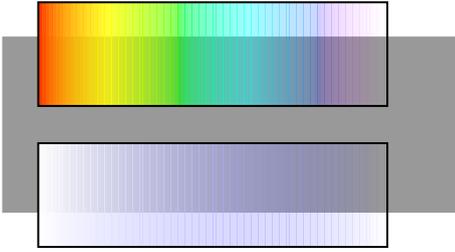
```
\begin{pspicture}[showgrid](5,4)
  \pscustom[fillstyle=radslope,
    linestyle=none,slopecenter=0.6 0.25,
    sloperadius=2cm]{%
    \psline(0,0)(1,3)(2,2)(3,4)(5,1)(0,0)}
\end{pspicture}
```

fading, startfading und endfading

Mit dem so genannten »fade out« kann eine Transparenz variabel gestaltet werden. Dies kann hier sowohl linear als auch radial geschehen. Aktiviert wird dieser Effekt durch `fading=true`, wobei der Start- und Endwert für die Transparenz durch `startfading` und `endfading` festzulegen sind. Die Standardeinstellung ist 0 und 1, was in der Ausgabe keiner und totaler Transparenz entspricht.

ab Version 1.3

21-02-15



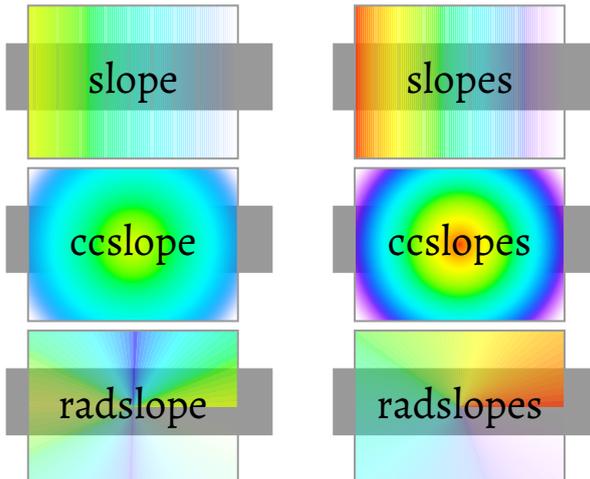
```
\usepackage{pstricks,pst-slpe}

\begin{pspicture}(5,3.5)
\psframe*[linecolor=black!40](-.5,.5)(6,3)
\psframe[fillstyle=slope,fading,
slopebegin=white](5,1.5)
\definecolor{slopebegin}{rgb}{0,1,1}
\psframe[fading,
fillstyle=slopes](0,2)(5,3.5)
\end{pspicture}
```

```
\usepackage{pstricks,pst-slpe}
```

```
\LARGE\psset{fading,endfading=0.75,linecolor=black!40}
\begin{tabular}{cc}
\psframe*(-0.3,-0.25)(3.5,20pt)\psframebox[fillstyle=slope]{\st{slope}} & \quad
\psframe*(-0.3,-0.25)(3.5,20pt)\psframebox[fillstyle=slopes]{\st{slopes}} & \quad \\
\psframe*(-0.3,-0.25)(3.5,20pt)\psframebox[fillstyle=ccslope]{\st{ccslope}} & \quad
\psframe*(-0.3,-0.25)(3.5,20pt)\psframebox[fillstyle=ccslopes]{\st{ccslopes}} & \quad \\
\psframe*(-0.3,-0.25)(3.5,20pt)\psframebox[fillstyle=radslope]{\st{radslope}} & \quad
\psframe*(-0.3,-0.25)(3.5,20pt)\psframebox[fillstyle=radslopes]{\st{radslopes}} & \quad \\
\end{tabular}
```

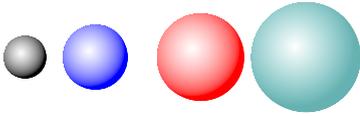
21-02-16



21.2.3 Makros

Das Paket stellt vorrangig für die `itemize`-Umgebung in Präsentationen das Makro `\psBall` bereit, welches farbige Kugeln mit einem 3D-Effekt ermöglicht.

```
\psBall [Optionen] {Farbe}{Radius}
\psBall [Optionen] (x,y){Farbe}{Radius}
```



```
\usepackage{pst-slope}
```

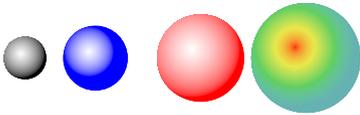
```
\begin{pspicture}(-0.5,-0.5)(5,0.5)
\psBall{black}{2ex}
\psBall(1,0){blue}{3ex}
\psBall(2.5,0){red}{4ex}
\psBall(4,0){green!50!blue!60}{5ex}
\end{pspicture}
```

21-02-17

Grundsätzlich muss das Makro nicht innerhalb einer `pspicture`-Umgebung angewendet werden. Es ist allerdings zu beachten, dass es selbst in einer Box der Größe `0pt` gesetzt wird und bei fehlenden Koordinatenangaben symmetrisch zum aktuellen Punkt ausgegeben wird. Die vordefinierten Parameter

```
sloperadius=0.075Radius
fillstyle=ccslope
slopebegin=white
slopeend=Farbe
slopecenter=0.4 0.6
linestyle=none
```

können in der üblichen Weise überschrieben werden, um andere Effekte zu erreichen.



```
\usepackage{pst-slope}
```

```
\begin{pspicture}(-0.5,-0.5)(5,0.5)
\psBall{black}{2ex}
\psBall[sloperadius=10pt](1,0){blue}{3ex}
\psBall(2.5,0){red}{4ex}
\psBall[slopebegin=red](4,0){green!50!blue!60}{5ex}
\end{pspicture}
```

21-02-18

21.3 pst-blur: Verschwommene Schatten

In Abschnitt 4.1.16 auf Seite 57 und Kapitel 22.1 auf Seite 358 wurden bereits verschiedene Möglichkeiten der Erstellung eines Schattens angegeben. In allen Fällen wurde dabei ausschließlich eine einzige Schattenfarbe verwendet. Das Paket `pst-blur` von Martin Giese unterstützt dagegen die Erstellung von verschwommenen Schatten, was einen erheblich besseren Eindruck macht. [20] Voraussetzung ist eine geschlossene Fläche, also zum Beispiel keine offenen Linienzüge. Wie derartige Schatten von `pst-blur` gebildet werden, wird in Abschnitt 11.3.15 auf Seite 148 behandelt.

21.3.1 Parameter

Tabelle 21.4 zeigt eine Zusammenstellung der speziellen, für `pst-blur` geltenden Parameter und Abbildung 21.1 die Zuordnung. Einige der von `pst-blur` verwendeten Parameter sind identisch zu den in Abschnitt 4.1.16 auf Seite 57 beschriebenen, sodass hier nicht weiter darauf eingegangen werden muss.

21-03-1

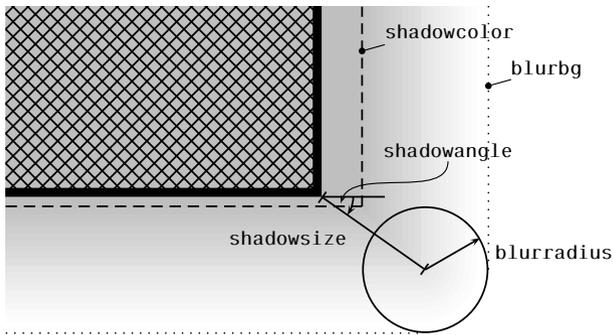


Abbildung 21.1: Parameter für verschwommene Schatten (Martin Giese)

Tabelle 21.4: Zusammenfassung aller Parameter für pst-blur

Name	Werte	Vorgabe
blur	Boolean	false
blurradius	Wert Einheit	1.5pt
blursteps	Wert	20
blurbg	Farbe	white

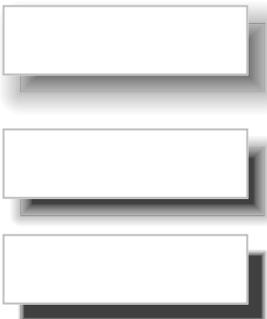
blur

`blur=true` ist Voraussetzung für die Aktivierung des Paketes. Anderenfalls erfolgt die normale Schattenbildung, wie in Abschnitt 4.1.16 ausführlich beschrieben. Voraussetzung ist jedoch, dass `shadow=true` gilt, andernfalls ist wiederum `blur` ohne jede Wirkung (siehe folgende Beispiele).

blurradius

Wie in Abbildung 21.1 dargestellt, bezeichnet `blurradius` den Radius der abgerundeten Ecken, bzw. den daraus folgenden Bereich des verschwimmenden Schattens. Wie man dem Beispiel entnehmen kann, hängen `shadowsize` (siehe Abschnitt 4.1.16) und `blurradius` zusammen, denn der optische Eindruck kehrt sich ins Negative, wenn `blurradius` größer wird als `shadowsize`.

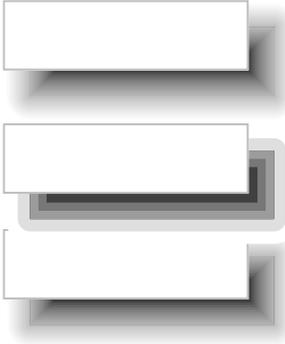
21-03-2



```
\usepackage{pst-blur}
\begin{pspicture}(4,4.25)
  \psset{shadow=true,blur=true,shadowsize=10pt,
    linecolor=lightgray}
  \psframe(3.5,1)
  \psframe[blurradius=5pt](0,1.5)(3.5,2.5)
  \psframe[blurradius=10pt](0,3.25)(3.5,4.25)
\end{pspicture}
```

blursteps

Dieser Parameter gibt an, in wie viel verschiedene Stufen die Schattenfarbe unterteilt wird, beginnend mit shadowcolor, endend bei blurbg. Je größer die Zahl der Zwischenschritte ist, desto länger die Rechenzeit.



```
\usepackage{pstricks,pst-blur}

\begin{pspicture}(4,4.25)
  \psset{shadow=true,blur=true,shadowsize=15pt,
    blurradius=10pt,linecolor=lightgray}
  \psframe(3.5,1)
  \psframe[blursteps=5](0,1.5)(3.5,2.5)
  \psframe[blursteps=50](0,3.25)(3.5,4.25)
\end{pspicture}
```

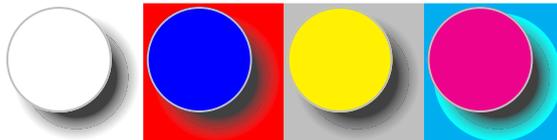
21-03-3

blurbg

blurbg bezeichnet die Hintergrundfarbe, womit die auslaufende Schattenfarbe gemeint ist, der von shadowcolor nach blurbg geht. Grundsätzlich kann man hierfür jede beliebige Farbe nehmen, nur bekommt man dann einen gänzlich anderen Schatteneffekt.

```
\usepackage{pst-blur}

\begin{pspicture}(8,2)
  \psset{shadow=true,blur=true,shadowsize=10pt,
    blurradius=5pt,linestyle=none,linecolor=lightgray}
  \psframe[shadow=false](2,2)\pscircle[linestyle=solid](0.8,1.2){0.75}
  \psframe*[shadow=false](4,0)(6,2)
  \pscircle[linestyle=solid,fillcolor=yellow,blurbg=lightgray](4.8,1.2){0.75}
  \psframe*[linecolor=red,shadow=false](2,0)(4,2)
  \pscircle[linestyle=solid,fillcolor=blue,blurbg=red](2.8,1.2){0.75}
  \psframe*[linecolor=cyan,shadow=false](6,0)(8,2)
  \pscircle[linestyle=solid,fillcolor=magenta,blurbg=cyan](6.8,1.2){0.75}
\end{pspicture}
```



21-03-4

21.3.2 \psblurbox

Das folgende Beispiel zeigt die unterschiedlichen Auswirkungen der verschiedenen Boxen mit einem Schatten.

```
\usepackage{pst-blur}
```

Wenn man in einem normalen Text die standardmäßige `\psframebox[shadow,blur]{%`
`\texttt{\textbackslash psframebox}}` mit der Option `\texttt{blur}` benutzt, so
gibt dies in der Regel ein schlechtes Ergebnis, da `\TeX` nicht berücksichtigt, dass
aufgrund des Schattens eigentlich eine einmalige Vergrößerung des Zeilenabstandes
notwendig wäre. Etwas anders sieht das Ergebnis mit der `\psshadowbox[blur]{%`
`\texttt{\textbackslash psshadowbox}}` aus, auch wenn es immer noch unbefriedigend ist;
bei Anwendung der Option `\texttt{blur}` wird der `\texttt{blurradius}` nicht
berücksichtigt. `\psblurbox{\texttt{\textbackslash psblurbox}}` sorgt hingegen dafür,
dass auch hier eine korrekte Zeilenschaltung erfolgt. Für dieses Makro kann der
Schattenwinkel allerdings nicht verändert werden, er ist konstant auf
`\texttt{shadowangle=45}` gesetzt.

21-03-5

Wenn man in einem normalen Text die standardmäßige `\psframebox` mit der Op-
tion `blur` benutzt, so gibt dies in der Regel ein schlechtes Ergebnis, da `TeX` nicht berück-
sichtigt, dass aufgrund des Schattens eigentlich eine einmalige Vergrößerung des Zeilen-
abstandes notwendig wäre. Etwas anders sieht das Ergebnis mit der `\psshadowbox`
aus, auch wenn es immer noch unbefriedigend ist; bei Anwendung der Option `blur`
wird der `blurradius` nicht berücksichtigt. `\psblurbox` sorgt hingegen dafür, dass
auch hier eine korrekte Zeilenschaltung erfolgt. Für dieses Makro kann der Schatten-
winkel allerdings nicht verändert werden, er ist konstant auf `shadowangle=45` gesetzt.

```
\psblurbox [Optionen] {Material}
```

Über Sinn oder Unsinn derartiger Boxen im laufenden Text soll hier nicht weiter disku-
tiert werden.

21-03-6

$$F(x) = \int f(x) dx$$

$$F(x) = \int f(x) dx$$

$$x_1 = x_0 - \frac{F(x_0)}{F'(x_0)} \geq x_0 \quad (1)$$

```
\usepackage{pstricks,pst-blur,amsmath}
\psset{linestyle=none}
\psblurbox{\parbox{0.9\linewidth}{%
  \[F(x)=\int f(x)dx\]} \}[10pt]
\psblurbox[blurbg=red]{%
  \parbox{0.9\linewidth}{%
    \[F(x)=\int f(x)dx\]} \}[12pt]
\psblurbox[blurbg=blue!30!red!80,
  blurradius=5pt]{%
  \parbox{0.9\linewidth}{%
    \begin{align}
      x_1 &= x_0 - \frac{F(x_0)}{F'(x_0)} \\
      & \{F^\prime(x_0)\} \geq x_0
    \end{align}}}
\end{align}}}
```

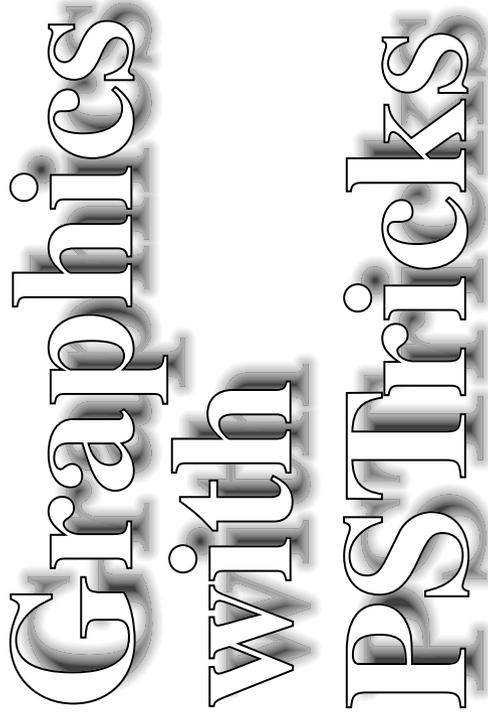
The image displays the text "Grapahics with PSTricks" in a decorative, outlined serif font. The text is rendered with a soft, grey-to-white gradient shadow that appears to float slightly above and to the right of the main text, creating a 3D effect. The word "Grapahics" is at the top, "with" is in the middle, and "PSTricks" is at the bottom. The shadow is most prominent on the letters 'h', 'i', 'c', 's', 't', 'r', 'i', 'c', 'k', 's'.

Abbildung 21.2: Schattenspiele...