pst-grad:Gradients

v. 1.06 – 2006/11/27

Herbert Voß[*]

September 25, 2008

Abstract

pst-grad is also one of the older and smaller packages. It provides only one fill style. A gradient could be created with the macros known from PSTricks, too, the use of pst-grad offers advantages though, since one does not need to take care of the calculation of the intermediate colour values.

This version of pst-grad integrates the function of the pst-ghsb package, which supports the HSB color model.

Contents

[*] Thanks to Lars Kotthoff and Angelo Rossi for translating this documentation!

1 Introduction

All parameters are only available when **gradient** is used as fill style. There are further packages which support such fill styles, especially for circular gradients (`pst-slpe`).

2 Parameters

Table 1 shows a compilation of the special parameters valid for `pst-grad`.

Table 1: Summary of all parameters for `pst-grad` and `pst-ghsb`

name	values	default
`gradbegin`	`<colour>`	`gradbegin`
`gradend`	`<colour>`	`gradend`
`gradlines`	`<value>`	`500`
`gradmidpoint`	`<value>`	`0.9`
`gradangle`	`<angle>`	`0`
`gradientHSB`	`false|true`	`false`
`GradientCircle`	`false|true`	`false`
`GradientScale`	`<value>`	`1.0`
`GradientPos`	`<(x,y)>`	`(0,0)`

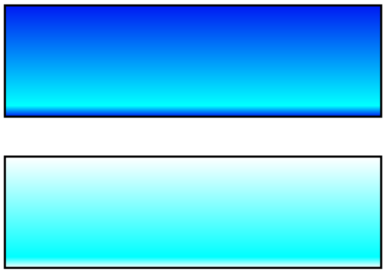
2.1 gradbegin

`gradbegin` denotes the parameter as well as the starting colour, which is a little bit confusing here.

```
\newrgbcolor{gradbegin}{0 .1 .95} % default
```

Consequently this starting colour can be changed by redefining the colour or by an assignment through the parameter.

```
\newrgbcolor{gradbegin}{0 0 1}
\definecolor{rgb}{gradbegin}{0 0 1} % requires color/xcolor package
\psset{gradbegin=blue}
```



```
\begin{pspicture}(5,3.5)
\psframe[fillstyle=gradient,gradbegin=white](5,1.5)
\newrgbcolor{gradbegin}{0 1 1}
\psframe[fillstyle=gradient](0,2)(5,3.5)
\end{pspicture}
```

- `gradbegin` should be defined as `RGB` colour, since a faultless function for `CMYK` or gray scales is not warranted in every case.

- ConTeXt users change the colour with `\definecolor{rgb}{gradbegin}{r=0,g=0,b=1}`

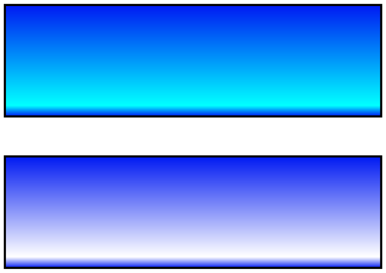
2.2 gradend

`gradend` is **not** the counterpart to `gradbegin`, for it is the colour which is reached at the relative point `gridmidpoint`. In every case it is ambiguous as `gradbegin` again.

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

Changes can be made differently again.

```
\newrgbcolor{gradend}{1 0 0}
\definecolor{rgb}{gradend}{1 0 0} % requires color/xcolor package
\psset{gradend=red}
```



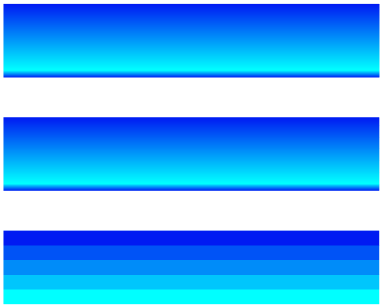
```
\begin{pspicture}(5,3.5)
\psframe[fillstyle=gradient,gradend=white](5,1.5)
\newrgbcolor{gradend}{1 0 0}
\psframe[fillstyle=gradient](0,2)(5,3.5)
\end{pspicture}
```

- `gradend` should be defined as `RGB` colour, since a faultless function for `CMYK` or gray scales is not warranted in every case.

- ConTeXt users change the colour with `\definecolor{rgb}{gradend}{r=1,g=1,b=0}`

2.3 gradlines

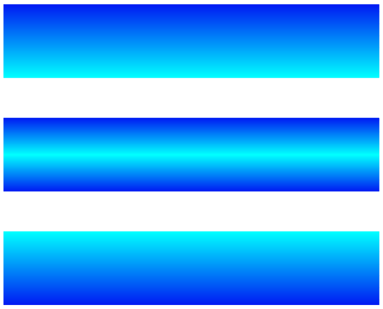
A gradient is nothing but a string of coloured lines. The width of those depends only on the resolution of the monitor resp. the printer in the end. But since this is very user-specific, `pst-grad` allows any number of lines, which can be changed through `gradlines`.



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

2.4 gradmidpoint

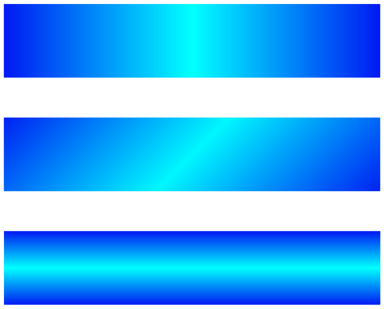
Denotes the relative point where the colour `gradend` is reached. Then it is proceeded in reverse order.



```
\begin{pspicture}(5,4)
\psset{fillstyle=gradient,linestyle=none}
\psframe[gradmidpoint=0](5,1)
\psframe[gradmidpoint=0.5](0,1.5)(5,2.5)
\psframe[gradmidpoint=1](0,3)(5,4)
\end{pspicture}
```

2.5 gradangle

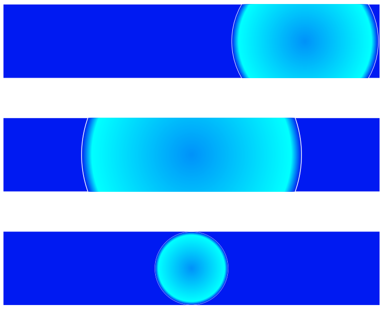
`gradangle` determines the gradient angle of the straight line.



```
\begin{pspicture}(5,4)
\psset{fillstyle=gradient,linestyle=none,
  gradmidpoint=0.5}
\psframe[gradangle=0](5,1)
\psframe[gradangle=45](0,1.5)(5,2.5)
\psframe[gradangle=90](0,3)(5,4)
\end{pspicture}
```

2.6 GradientCircle, GradientScale and GradientPos

With the option `GradientCircle` circular gradients can be created. The radius can be influenced through `GradientScale` and the centre with `GradientPos`. The specification of the coordinates refers to the based coordinate system, which is given by the `pspicture` environment as a rule.



```
\begin{pspicture}(5,4)
\psset{fillstyle=gradient,linestyle=none}
\psframe[GradientCircle=true](5,1)%
\psframe[GradientCircle=true,GradientScale=3](0,1.5)(5,2.5)%
\psframe[GradientCircle=true,GradientScale=2,%
      GradientPos={(4,3.5)}](0,3)(5,4)%
\end{pspicture}
```

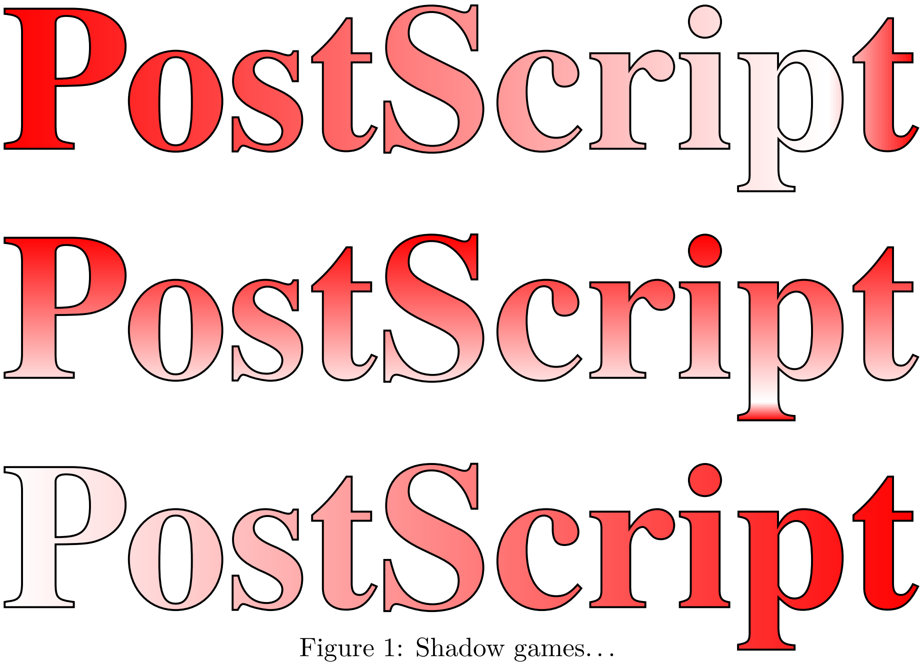
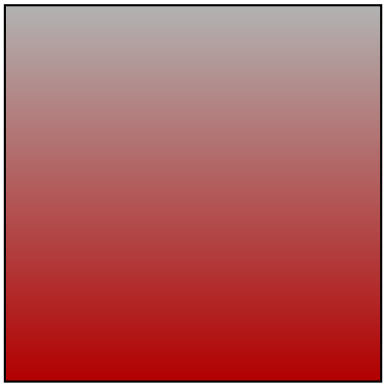
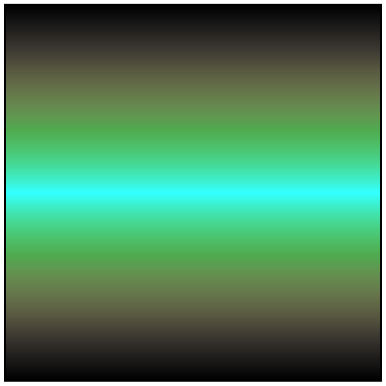


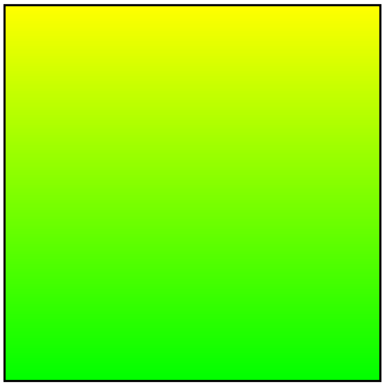
Figure 1: Shadow games...

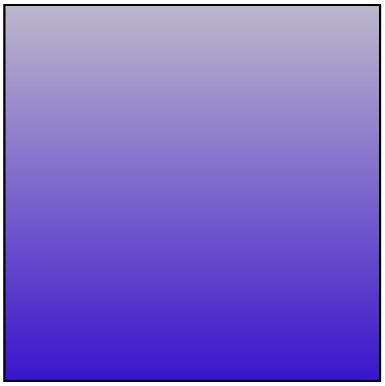
2.7 GradientHSB

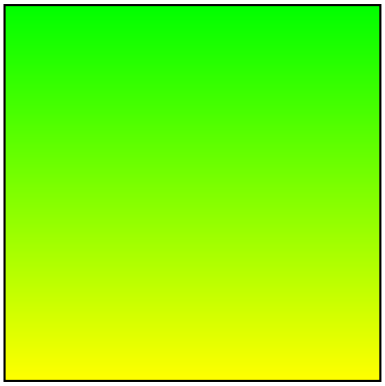
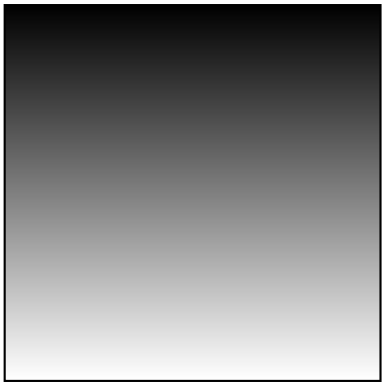


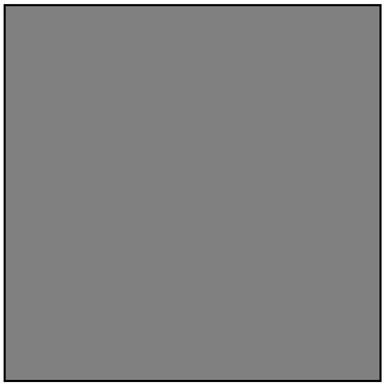


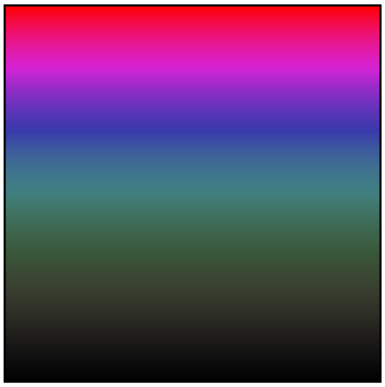
```
\newcommand{\Fig}[1][]{%
\begin{pspicture}(5.5,5.5)
 \psframe[#1](5,5)
\end{pspicture}}
\newhsbcolor{ColorA}{0 0 0.7}
\newhsbcolor{ColorB}{0 1 0.7}
\newhsbcolor{ColorC}{.5 0.8 0}
\newhsbcolor{ColorD}{.5 0.8 1}
\psset{fillstyle=gradient,gradientHSB=true}
\Fig[gradmidpoint=1,gradbegin=ColorA,gradend=ColorB]
\Fig[gradmidpoint=0.5,gradbegin=ColorC,gradend=ColorD]
```

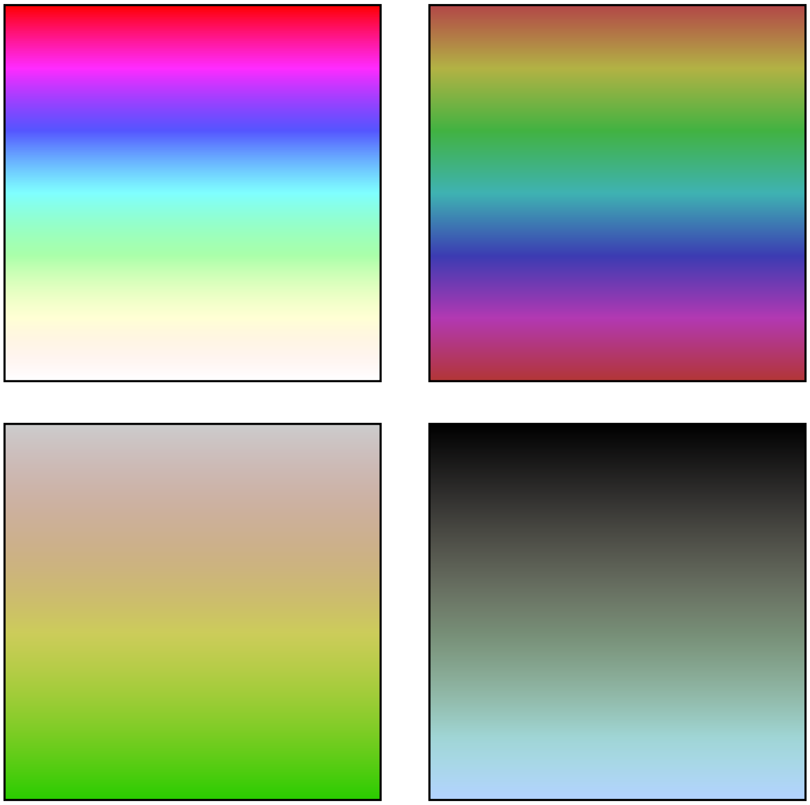










```
\definecolor{ColorA}{hsb}{0.7, 0.1, 0.8}
\definecolor{ColorB}{hsb}{0.7, 0.9, 0.8}
\definecolor{ColorC}{hsb}{0, 0, 0}
\definecolor{ColorD}{hsb}{0, 0, 1}
\definecolor{ColorE}{hsb}{0, 0, 0.5}
\definecolor{ColorF}{hsb}{0, 1, 0.5}
\definecolor{ColorG}{hsb}{0, 0, 0.5}
\definecolor{ColorH}{hsb}{0.99999, 0, 0.5}   % As it's cycliç 1=0 !
\definecolor{ColorI}{hsb}{1, 1, 1}
\definecolor{ColorJ}{hsb}{1, 0, 0}
\definecolor{ColorK}{hsb}{0.99999, 1, 1}     % As it's cyclic 1=0 !
\definecolor{ColorL}{hsb}{0, 1, 0}
\definecolor{ColorM}{hsb}{0.99999, 1, 1}     % As it's cyclic 1=0 !
\definecolor{ColorN}{hsb}{0, 0, 1}
\definecolor{ColorO}{hsb}{0, 0.6, 0.7}
\definecolor{ColorP}{hsb}{0.99999, 0.7, 0.7} % As it's cyclic 1=0 !
\definecolor{ColorQ}{hsb}{0.3, 0, 0.8}
\definecolor{ColorR}{hsb}{0.3, 1, 0.8}
\definecolor{ColorS}{hsb}{0.6, 0.3, 0}
\definecolor{ColorT}{hsb}{0.6, 0.3, 1}
\psset{fillstyle=gradient,gradmidpoint=1}
\Fig[gradbegin=yellow,gradend=green]
\Fig[gradientHSB=true,gradbegin=ColorA,gradend=ColorB]

```

```
\Fig[gradbegin=green,gradend=yellow]
\psset{gradientHSB=true}
\Fig[gradbegin=ColorC,gradend=ColorD]

\Fig[gradbegin=ColorE,gradend=ColorF]
\Fig[gradbegin=ColorG,gradend=ColorH]

\Fig[gradbegin=ColorI,gradend=ColorJ]
\Fig[gradbegin=ColorK,gradend=ColorL]

\Fig[gradbegin=ColorM,gradend=ColorN]
\Fig[gradbegin=ColorO,gradend=ColorP]

\Fig[gradbegin=ColorQ,gradend=ColorR]
\Fig[gradbegin=ColorS,gradend=ColorT]
```

References

[1] Hendri Adriaens. *The `xkeyval` - package.* CTAN:/macros/latex/contrib/xkeyval/, 2006.

[2] D. P. Carlisle and S. P. Q. Rahtz. *The `keyval` - package.* CTAN:/macros/latex/required/graphics/keyval.dtx, 2001.

[3] Denis Girou. Présentation de PSTricks. *Cahier GUTenberg*, 16:21–70, April 1994.

[4] Michel Goosens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Graphics Companion.* Addison-Wesley Publishing Company, Reading, Mass., 1997.

[5] Laura E. Jackson and Herbert Voß. Die plot-funktionen von `pst-plot`. *Die TeXnische Komödie*, 2/02:27–34, June 2002.

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

[7] Rolf Niepraschk and Herbert Voß. *`PSTricks` - mehr als nur ein alter Hut.* DANTE 2004 in Darmstadt, http://PSTricks.de/docs/Darmstadt2004.pdf, 2004.

[8] Sebastian Rahtz. An introduction to PSTricks, part I. *Baskerville*, 6(1):22–34, February 1996.

[9] Sebastian Rahtz. An introduction to PSTricks, part II. *Baskerville*, 6(2):23–33, April 1996.

[10] Timothy Van Zandt. *PSTricks - PostScript macros for Generic TeX, Documented Code.* CTAN:/graphics/pstricks/obsolete/doc/src/pst-code.tex, 1997.

[11] Herbert Voß. Three dimensional plots with `pst-3dplot`. *TUGboat*, 22-4:319, December 2001.

[12] Herbert Voß. ***PSTricks** – Grafik für TEX und LATEX*. DANTE – Lehmanns, Heidelberg/Hamburg, forth edition, 2007.

[13] Herbert Voß. *The **pstricks-add** - package*. CTAN:/graphics/pstricks/contrib/pstricks-add/, 2007.

[14] Herbert Voß and Jana Voß. The plot functions of `pst-plot`. *TUGboat*, 22-4:314–318, December 2001.

[15] Michael Wiedmann. *References for TEX and Friends*. http://www.miwie.org/tex-refs/, 2004.

[16] Timothy van Zandt. ***pst-eps**: Exporting eps images*. CTAN:graphics/pstricks/generic/, 2003.

[17] Timothy van Zandt. ***multido.tex** - a loop macro, that supports fixed-point addition*. CTAN:/graphics/pstricks/generic/multido.tex, 2004.

[18] Timothy Van Zandt. *The **pst-plot** - package*. CTAN:/graphics/pstricks/generic/, 2004.

[19] Timothy van Zandt. *PSTricks - PostScript macros for generic TEX*. http://www.tug.org/application/PSTricks, 2006.

[20] Timothy van Zandt and Denis Girou. Inside PSTricks. *TUGboat*, 15:239–246, September 1994.

Index