

pst-grad:Gradients

v. 1.05 – 2006/11/04

Herbert Voß*

November 4, 2006

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

1	Introduction	2
2	Parameters	2
2.1	gradbegin	2
2.2	gradend	3
2.3	gradlines	3
2.4	gradmidpoint	4
2.5	gradangle	4
2.6	GradientCircle, GradientScale and GradientPos	4
2.7	GradientHSB	6

*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
<code>gradbegin</code>	<code><colour></code>	<code>gradbegin</code>
<code>gradend</code>	<code><colour></code>	<code>gradend</code>
<code>gradlines</code>	<code><value></code>	500
<code>gradmidpoint</code>	<code><value></code>	0.9
<code>gradangle</code>	<code><angle></code>	0
<code>gradientHSB</code>	<code>false true</code>	false
<code>GradientCircle</code>	<code>false true</code>	false
<code>GradientScale</code>	<code><value></code>	1.0
<code>GradientPos</code>	<code><(x,y)></code>	(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}
```



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

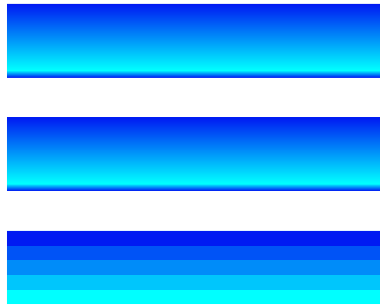


```
1 \begin{pspicture}(5,3.5)
2 \psframe[fillstyle=gradient,gradend=
  white](5,1.5)
3 \newrgbcolor{gradend}{1 0 0}
4 \psframe[fillstyle=gradient](0,2)
  (5,3.5)
5 \end{pspicture}
```

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



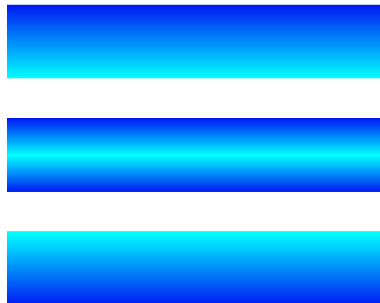
```

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

```

2.4 gradmidpoint

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



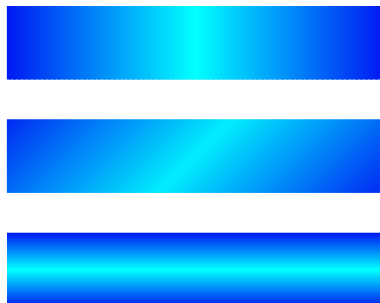
```

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

```

2.5 gradangle

gradangle determines the gradient angle of the straight line.



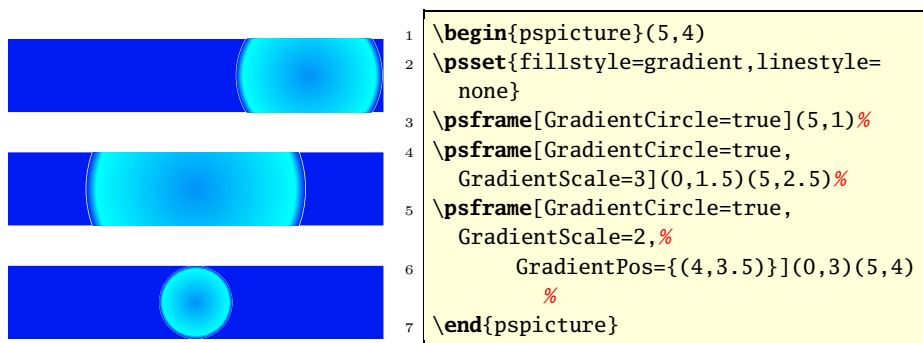
```

1 \begin{pspicture}(5,4)
2 \psset{fillstyle=gradient,linestyle=
  none,gradmidpoint=0.5}
3 \psframe[gradangle=0](5,1)
4 \psframe[gradangle=45](0,1.5)(5,2.5)
5 \psframe[gradangle=90](0,3)(5,4)
6 \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.



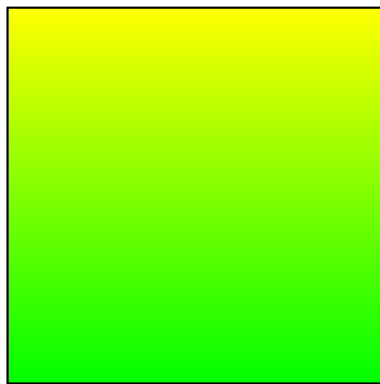
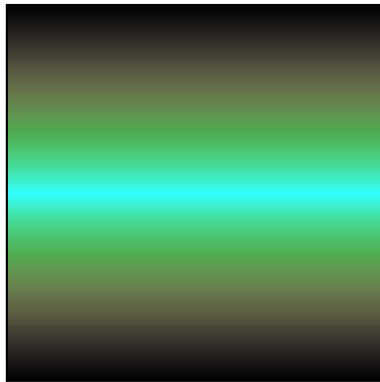
PostScript

PostScript

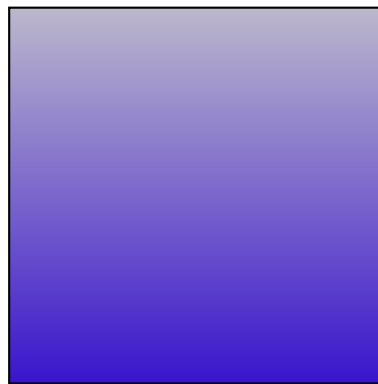
PostScript

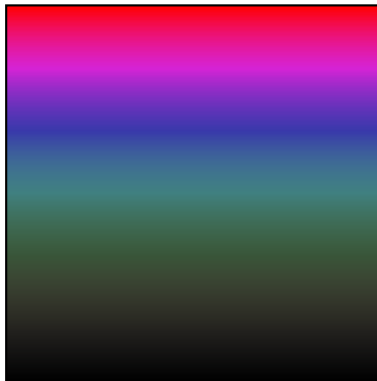
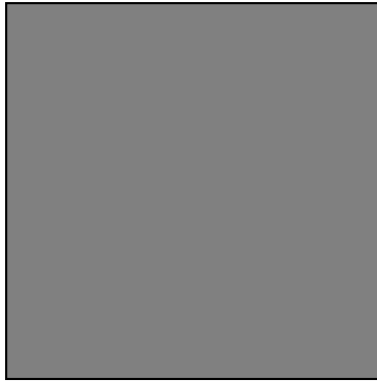
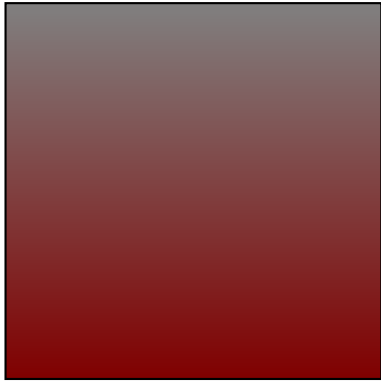
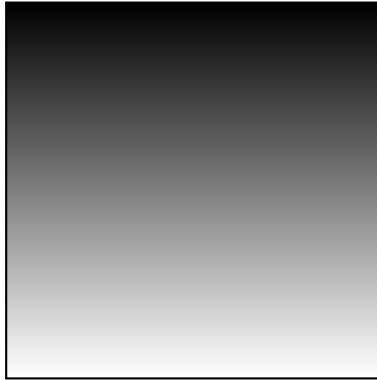
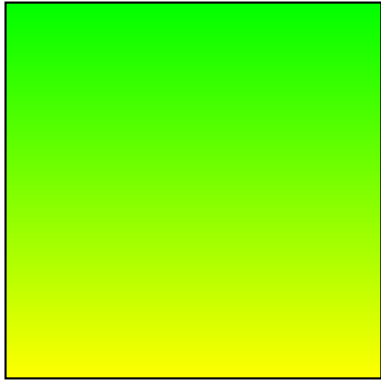
Figure 1: Shadow games...

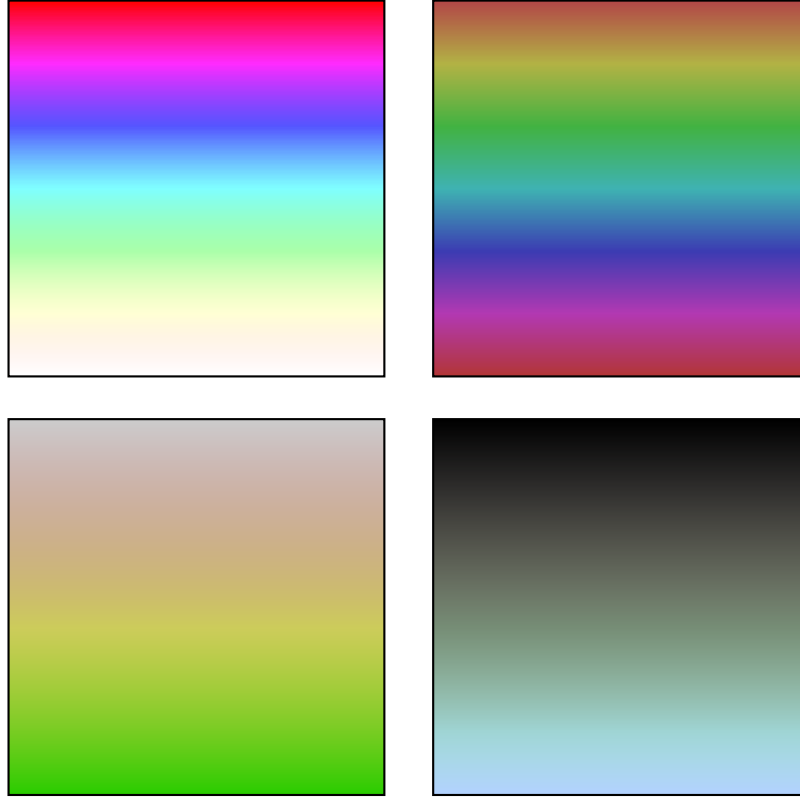
2.7 GradientHSB



```
1 \newcommand{\Fig}[1][[]]{%
2 \begin{pspicture}(5.5,5.5)
3   \psframe[#1](5,5)
4 \end{pspicture}}
5 \newhsbcolor{ColorA}{0 0 0.7}
6 \newhsbcolor{ColorB}{0 1 0.7}
7 \newhsbcolor{ColorC}{.5 0.8 0}
8 \newhsbcolor{ColorD}{.5 0.8 1}
9 \psset{fillstyle=gradient,
10        gradientHSB=true}
11 \Fig[gradmidpoint=1,gradbegin=ColorA,
12      ,gradend=ColorB]
13 \Fig[gradmidpoint=0.5,gradbegin=
14      ColorC,gradend=ColorD]
```







```

1 \definecolor{ColorA}{hsb}{0.7, 0.1, 0.8}
2 \definecolor{ColorB}{hsb}{0.7, 0.9, 0.8}
3 \definecolor{ColorC}{hsb}{0, 0, 0}
4 \definecolor{ColorD}{hsb}{0, 0, 1}
5 \definecolor{ColorE}{hsb}{0, 0, 0.5}
6 \definecolor{ColorF}{hsb}{0, 1, 0.5}
7 \definecolor{ColorG}{hsb}{0, 0, 0.5}
8 \definecolor{ColorH}{hsb}{0.99999, 0, 0.5} % As it's cyclic 1=0 !
9 \definecolor{ColorI}{hsb}{1, 1, 1}
10 \definecolor{ColorJ}{hsb}{1, 0, 0}
11 \definecolor{ColorK}{hsb}{0.99999, 1, 1} % As it's cyclic 1=0 !
12 \definecolor{ColorL}{hsb}{0, 1, 0}
13 \definecolor{ColorM}{hsb}{0.99999, 1, 1} % As it's cyclic 1=0 !
14 \definecolor{ColorN}{hsb}{0, 0, 1}
15 \definecolor{ColorO}{hsb}{0, 0.6, 0.7}
16 \definecolor{ColorP}{hsb}{0.99999, 0.7, 0.7} % As it's cyclic 1=0 !
17 \definecolor{ColorQ}{hsb}{0.3, 0, 0.8}
18 \definecolor{ColorR}{hsb}{0.3, 1, 0.8}
19 \definecolor{ColorS}{hsb}{0.6, 0.3, 0}
20 \definecolor{ColorT}{hsb}{0.6, 0.3, 1}

```



```

21 \psset{fillstyle=gradient,gradmidpoint=1}
22 \Fig[gradbegin=yellow,gradend=green]
23 \Fig[gradientHSB=true,gradbegin=ColorA,gradend=ColorB]
24
25 \Fig[gradbegin=green,gradend=yellow]
26 \psset{gradientHSB=true}
27 \Fig[gradbegin=ColorC,gradend=ColorD]
28
29 \Fig[gradbegin=ColorE,gradend=ColorF]
30 \Fig[gradbegin=ColorG,gradend=ColorH]
31
32 \Fig[gradbegin=ColorI,gradend=ColorJ]
33 \Fig[gradbegin=ColorK,gradend=ColorL]
34
35 \Fig[gradbegin=ColorM,gradend=ColorN]
36 \Fig[gradbegin=ColorO,gradend=ColorP]
37
38 \Fig[gradbegin=ColorQ,gradend=ColorR]
39 \Fig[gradbegin=ColorS,gradend=ColorT]

```

References

- [1] Hendri Adriaens. *The xkeyval - package*.
CTAN:/macros/latex/contrib/xkeyval/, 2004.
- [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 Goossens, Frank Mittelbach, and Alexander Samarin. *The L^AT_EX Graphics Companion*. Addison-Wesley Publishing Company, Reading, Mass., 1997.
- [5] Laura E. Jackson and Herbert Voß. Die plot-funktionen von pst-plot. *Die T_EXnische 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 T_EX, Documented Code*. [CTAN:/graphics/pstricks/obsolete/doc/src/pst-code.tex](http://ctan.org/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. *The `pstricks-add` - package*. [CTAN:/graphics/pstricks/contrib/pstricks-add/](http://ctan.org/graphics/pstricks/contrib/pstricks-add/), 2004.
- [13] Herbert Voß. *PSTricks – Grafik für T_EX und L^AT_EX*. DANTE – Lehmanns, Heidelberg/Hamburg, third edition, 2006.
- [14] Herbert Voß and Jana Voß. The plot functions of `pst-plot`. *TUGboat*, 22-4:314–318, December 2001.
- [15] Michael Wiedmann. *References for T_EX and Friends*. <http://www.miwie.org/tex-refs/>, 2004.
- [16] Timothy van Zandt. *PSTricks - PostScript macros for generic T_EX*. <http://www.tug.org/application/PSTricks>, 1993.
- [17] Timothy van Zandt. *multido.tex - a loop macro, that supports fixed-point addition*. [CTAN:/graphics/pstricks/generic/multido.tex](http://ctan.org/graphics/pstricks/generic/multido.tex), 1997.
- [18] Timothy van Zandt. *pst-eps: Exporting eps images*. [CTAN:/graphics/pstricks/generic/](http://ctan.org/graphics/pstricks/generic/), 1999.
- [19] Timothy Van Zandt. *The `pst-plot` - package*. [CTAN:/graphics/pstricks/generic/](http://ctan.org/graphics/pstricks/generic/), 2004.
- [20] Timothy van Zandt and Denis Girou. Inside PSTricks. *TUGboat*, 15:239–246, September 1994.

Index

gradangle, 2, 4
gradbegin, 2
gradend, 2, 3
GradientCircle, 2, 4
gradientHSB, 2
GradientPos, 2, 4
GradientScale, 2, 4
gradlines, 2, 3
gradmidpoint, 2, 4

Paket

pst-grad, 1
pst-slpe, 2

Parameter

GradientCircle, 2
GradientPos, 2
GradientScale, 2
gradangle, 2
gradbegin, 2
gradend, 2
gradientHSB, 2
gradlines, 2
gradmidpoint, 2
pst-grad, 1

RGB, 3