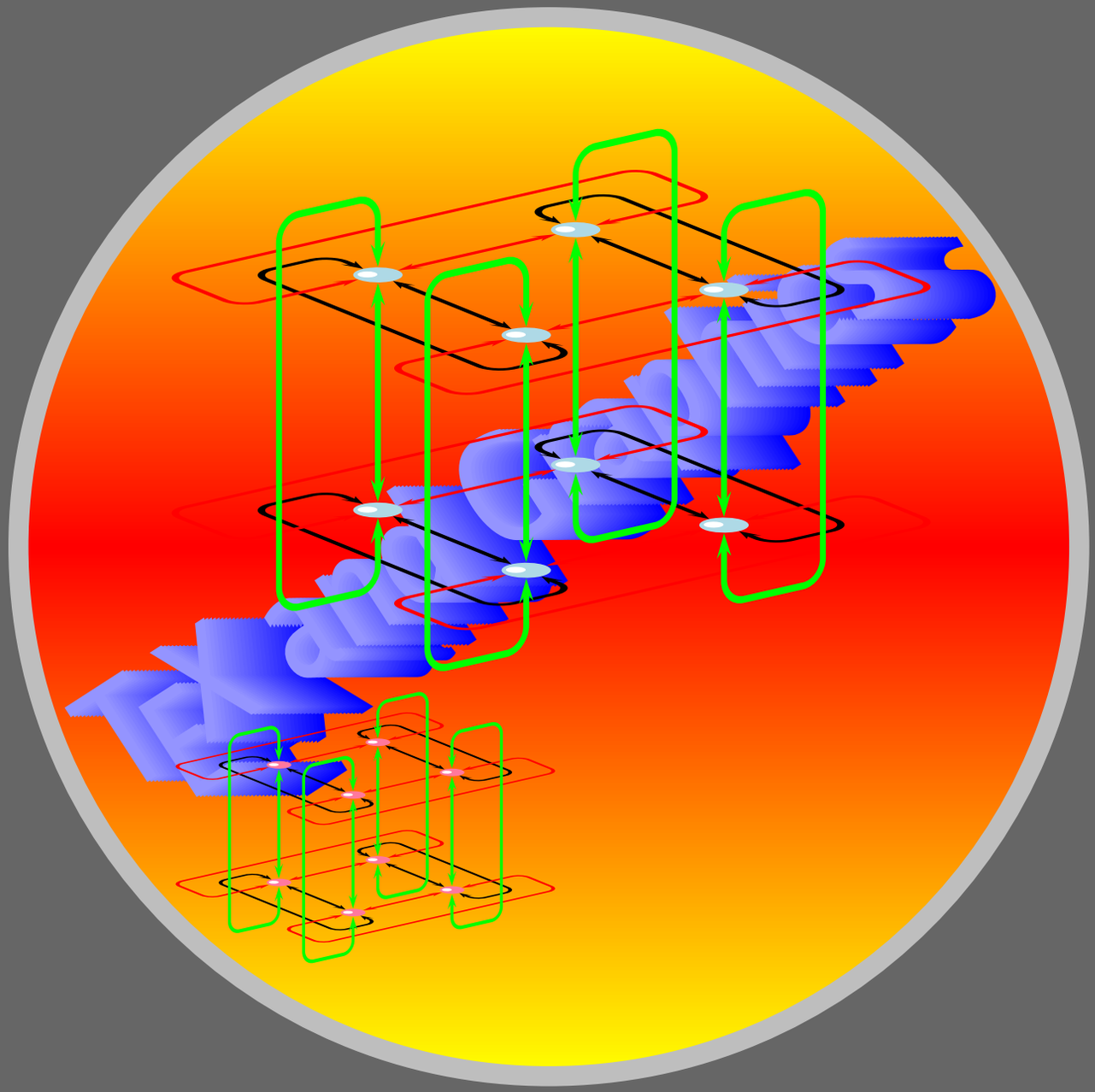


Colorful Tricks

Online LaTeX Tutorial

Part II – Graphics

PSTricks

The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

2. Colorful Tricks

Seeing the (ps)tricks so far, at least some of you may be wishing for a bit of color in the graphics. Here's good news for such people: you can have your wish! PSTricks comes with a set of macros that provide a basic set of colors and lets you define your own colors. However, it has some incompatibility with the LaTeX package color. However, David Carlisle has written a package pstcol which modifies the PSTricks color interface to work with LaTeX colors. All of our examples in this chapter assumes that this package is loaded, using the command `\usepackage{pstcol}` in the preamble. Note that this loads the pstricks package also, so that it need not be separately loaded.

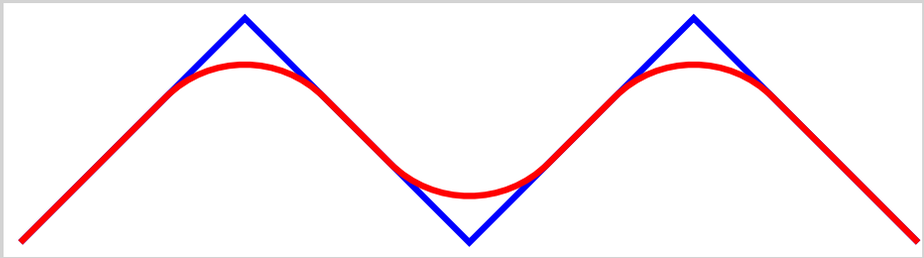
The Indian TeX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

2.1. Ordinary colors

The colors `red`, `green`, `blue`, `cyan`, `magenta`, `yellow`, `black`, `white` are predefined in pstcol and various parts of a picture can be colored with these by assigning these values to the various "color" parameters.

Lines are colored by setting the parameter `linecolor`. Thus we can colorfully distinguish the effect of `linearc` (do you remember this parameter?) as in the example below:

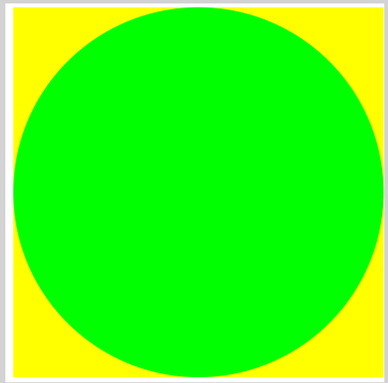
```
\begin{pspicture}(0,0)(5,2)
\psline[linecolor=blue](1,1)(2,2)(3,1)(4,2)(5,1)
\psline[linearc=0.5,linecolor=red](1,1)(2,2)(3,1)(4,2)(5,1)
\end{pspicture}
```



The same parameter `linecolor` can also be used to color "solid" objects made with "starred" commands as in the next example:

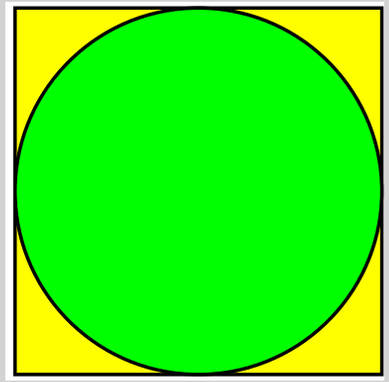
```
\begin{pspicture}(0,0)(3,3)
\psframe*[linecolor=yellow](0,0)(3,3)
\pscircle*[linecolor=green](1.5,1.5){1.5}
\end{pspicture}
```


The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in



Another way of coloring closed regions is to use the `fillstyle` and `fillcolor` parameters. For example

```
\begin{pspicture}(0,0)(3,3)
  \psframe[fillstyle=solid, fillcolor=yellow](0,0)(3,3)
  \pscircle[fillstyle=solid, fillcolor=green](1.5,1.5){1.5}
\end{pspicture}
```



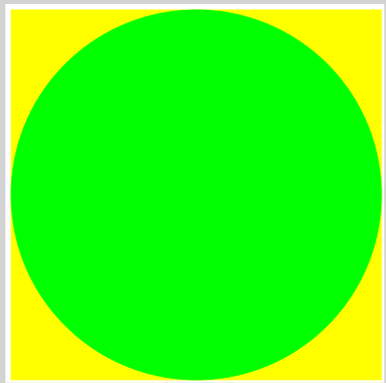
Do you see any difference? Yes, the black outlines. Note that with a "solid" object made with the starred commands and `linecolor`, you're sort of painting the *entire* object—and this includes the boundary—*line by line*,

The Indian TeX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

while in the case of a “closed” region and `fillcolor`, you’re painting only the region *enclosed* by the boundary *after* drawing the boundary in the default `linecolor`, which is black.

We can get rid of the “boundaries” in this example by setting the `linestyle` parameter to `none`. (Do you remember other possible values of this parameter?)

```
\begin{pspicture}(0,0)(3,3)
  \psframe[linestyle=none,fillstyle=solid,fillcolor=yellow](0,0)(3,3)
  \pscircle[linestyle=none,fillstyle=solid,fillcolor=green](1.5,1.5){1.5}
\end{pspicture}
```

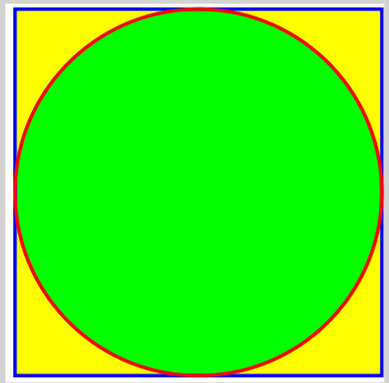


which is exactly the same output of the second example. (In fact what the starred versions of the commands do is to set `linewidth` to 0, `linestyle` to `none`, `fillcolor` to `linecolor` and `fillstyle` to `solid`.)

On the other hand, to put a boundary around a “solid” object colored with “linecolor”, just redraw the boundary, and you can do this with any color:

The Indian TeX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

```
\begin{pspicture}(0,0)(3,3)
   \psframe*[linecolor=yellow](0,0)(3,3)
   \pscircle*[linecolor=green](1.5,1.5){1.5}
   \psframe[linecolor=blue](0,0)(3,3)
   \pscircle[linecolor=red](1.5,1.5){1.5}
 \end{pspicture}
```



Online LaTeX Tutorial

Part II – Graphics

PSTricks

The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

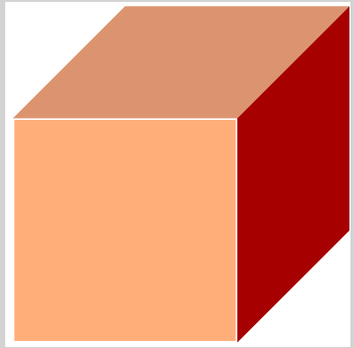
2.2. More colors

Some dvi drivers support a named color model, which means in practical terms that you can use the names of a certain set of predefined colors. For example, the dvips offers 64 colors as listed in the Figure 2.1. To use these colors, load the package pstcol with the option usenames as

```
\usepackage[usenames]{pstcol}
```

Then for example, with the code given below, you can produce the picture shown alongside:

```
\begin{pspicture}(0,0)(3,3)
 \psframe[linestyle=none, fillstyle=solid,fillcolor=Apricot](0,0)(2,2)
 \pspolygon[linestyle=none,fillstyle=solid,fillcolor=Tan](0,2)(2,2)(3,3)(1,3)
 \pspolygon[linestyle=none,fillstyle=solid,fillcolor=Mahogany]
                        (2,0)(3,1)(3,3)(2,2)
\end{pspicture}
```



The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

NAME	CMYK	COLOR	NAME	CMYK	COLOR
GreenYellow	0.15,0,0.69,0		RoyalPurple	0.75,0.90,0,0	
Yellow	0,0,1,0		BlueViolet	0.86,0.91,0,0.04	
Goldenrod	0,0.10,0.84,0		Periwinkle	0.57,0.55,0,0	
Dandelion	0,0.29,0.84,0		CadetBlue	0.62,0.57,0.23,0	
Apricot	0,0.32,0.52,0		CornflowerBlue	0.65,0.13,0,0	
Peach	0,0.50,0.70,0		MidnightBlue	0.98,0.13,0,0.43	
Melon	0,0.46,0.50		NavyBlue	0.94,0.54,0,0	
YellowOrange	0,0.42,1,0		RoyalBlue	1,0.50,0,0	
Orange	0,0.61,0.87,0		Blue	1,1,0,0	
BurntOrange	0,0.51,1,0		Cerulean	0.94,0.11,0,0	
Bittersweet	0,0.75,1,0.24		Cyan	1,0,0,0	
RedOrange	0,0.77,0.87,0		ProcessBlue	0.96,0,0,0	
Mahogany	0,0.85,0.87,0.35		SkyBlue	0.62,0,0.12,0	
Maroon	0,0.87,0.68,0.32		Turquoise	0.85,0,0.20,0	
BrickRed	0,0.89,0.94,0.28		TealBlue	0.86,0,0.34,0.02	
Red	0,1,1,0		Aquamarine	0.82,0,0.30,0	
OrangeRed	0,1,0.50,0		BlueGreen	0.85,0,0.33,0	
RubineRed	0,1,0.13,0		Emerald	1,0,0.50,0	
WildStrawberry	0,0.96,0.39,0		JungleGreen	0.99,0,0.52,0	
Salmon	0,0.53,0.38,0		SeaGreen	0.69,0,0.50,0	
CarnationPink	0,0.63,0,0		Green	1,0,1,0	
Magenta	0,1,0,0		ForestGreen	0.91,0,0.88,0.12	
VioletRed	0,0.81,0,0		PineGreen	0.92,0,0.59,0.25	
Rhodamine	0,0.82,0,0		LimeGreen	0.50,0,1,0	
Mulberry	0.34,0.90,0,0.02		YellowGreen	0.44,0,0.74,0	
RedViolet	0.07,0.90,0,0.34		SpringGreen	0.26,0,0.76,0	
Fuchsia	0.47,0.91,0,0.08		OliveGreen	0.64,0,0.95,0.40	
Lavender	0,0.48,0,0		RawSienna	0,0.72,1,0.45	
Thistle	0.12,0.59,0,0		Sepia	0,0.83,1,0.70	
Orchid	0.32,0.64,0,0		Brown	0,0.81,1,0.60	
DarkOrchid	0.40,0.80,0.20,0		Tan	0.14,0.42,0.56,0	
Purple	0.45,0.86,0,0		Gray	0,0,0,0.50	
Plum	0.50,1,0,0		Black	0,0,0,1	
Violet	0.79,0.88,0,0		White	0,0,0,0	

Figure 2.1: Named colors in dvips

Online LaTeX Tutorial

Part II – Graphics

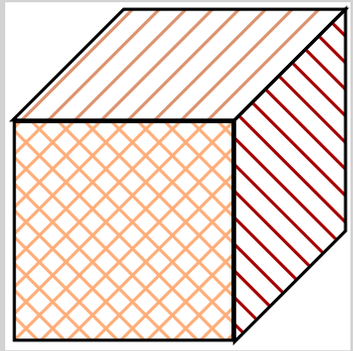
PSTricks

The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

2.3. Fill—in style

We've often used the setting `fillstyle=solid` in the examples above. There are various other ways of filling up closed regions, by assigning different values to the parameter `fillstyle`. The values `vlines`, `hlines` and `crosshatch` fill the region with vertical lines, horizontal lines and criss-cross lines, as shown in the example below:

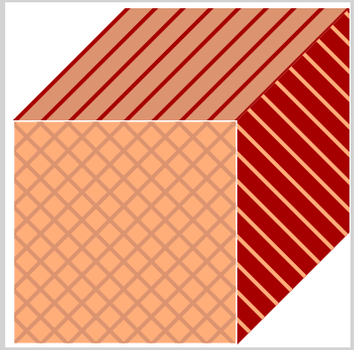
```
\begin{pspicture}(0,0)(3,3)
  \psframe[fillstyle=crosshatch,hatchcolor=Apricot](0,0)(2,2)
  \pspolygon[fillstyle=hlines,hatchcolor=Tan](0,2)(2,2)(3,3)(1,3)
  \pspolygon[fillstyle=vlines,hatchcolor=Mahogany](2,0)(3,1)(3,3)(2,2)
 \end{pspicture}
```



As can be seen from this example, the color of the lines making up the fill-pattern is set by the parameter `hatchcolor`. We can also set the background color using the parameter `fillcolor`, if we use the starred form of the *values* for the `fillstyle`. The example below illustrates this. Note also the use of the parameter `hatchwidth` which controls the width of the lines making up the pattern. Its default value is 0.8pt.

The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

```
\begin{pspicture}(0,0)(4,4)
  \psframe[linestyle=none,fillstyle=crosshatch*,hatchcolor=Tan,%
           hatchwidth=1pt,fillcolor=Apricot](0,0)(2,2)
  \pspolygon[linestyle=none,fillstyle=hlines*,hatchcolor=Mahogany,%
             hatchwidth=1pt,fillcolor=Tan](0,2)(2,2)(3,3)(1,3)
 \pspolygon[linestyle=none,fillstyle=vlines*,hatchcolor=Apricot,%
            hatchwidth=1pt,fillcolor=Mahogany](2,0)(3,1)(3,3)(2,2)
\end{pspicture}
```

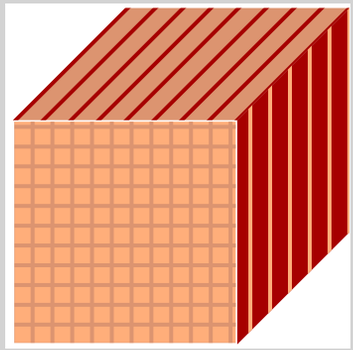


The slant of the lines in the pattern is controlled by the `hatchangle` parameter and its default value is 45 (degrees). The next example shows the effect of changing it.

```
\begin{pspicture}(0,0)(3,3)
  \psframe[linestyle=none,%
           fillstyle=crosshatch*,%
           hatchcolor=Tan,%
           hatchwidth=1pt,%
           hatchangle=90,%
           fillcolor=Apricot]%
          (0,0)(2,2)
  \pspolygon[linestyle=none,%
             fillstyle=hlines*,%
```


The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

```
              hatchcolor=Mahogany,%
              hatchwidth=1pt,%
              fillcolor=Tan]%
             (0,2)(2,2)(3,3)(1,3)
  \pspolygon[linestyle=none,%
              fillstyle=vlines*,%
              hatchcolor=Apricot,%
              hatchwidth=1pt,%
              hatchangle=180,%
              fillcolor=Mahogany]%
             (2,0)(3,1)(3,3)(2,2)
\end{pspicture}
```



Online LaTeX Tutorial
Part II – Graphics
PSTricks

The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

2.4. Custom colors

If you are not satisfied with any of the sixty four named colors, you can define your own colors using the `\definecolor` command. The syntax for this command is

```
\definecolor{name}{model}{spec}
```

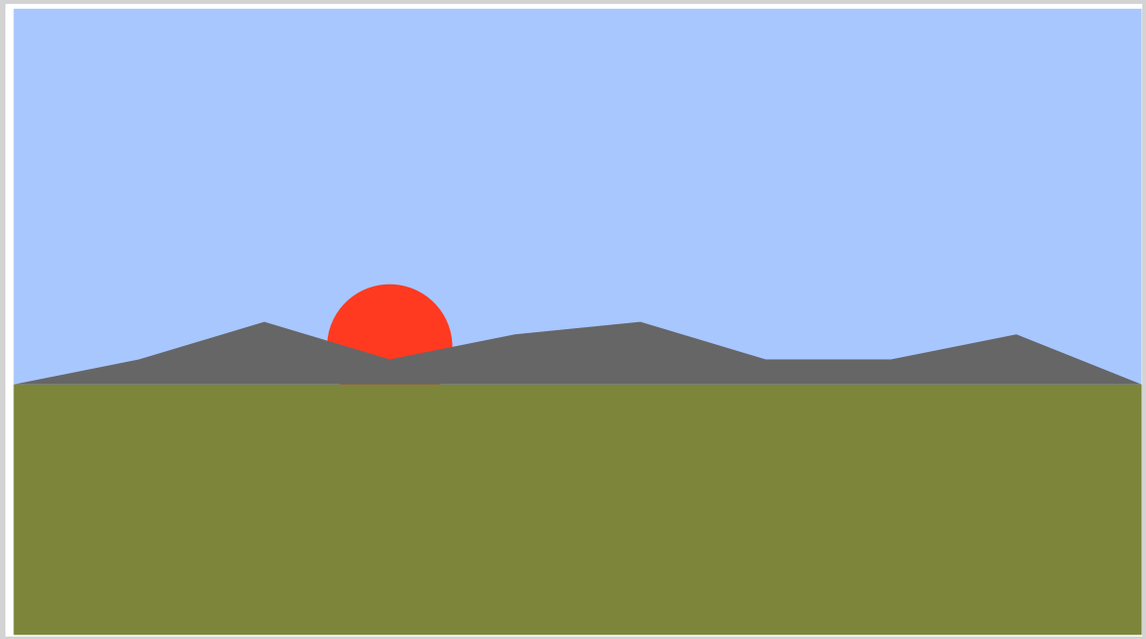
where *name* is the name of the color you want to create, *model* is the scheme of specifying the color such as `rgb`, `cmyk`, `gray` or `named`. For example, see how the colors `myblue`, `mygreen` and `mygray` are used in the code below.

Note especially the definition of `mygray`: different shades of gray from white to black can be created by using the `gray` model and specifying a number between 0 and 1; the larger the number, the lighter the shade with 0 giving black and 1, white.

```
\definecolor{myblue}{rgb}{0.66,0.78,1.00}
\definecolor{mygreen}{rgb}{0.49,0.52,0.23}
\definecolor{mygray}{gray}{0.4}
\begin{pspicture}(0,0)(9,5)
  \psframe[fillstyle=solid,%
           fillcolor=myblue]%
          (0,2)(9,5)
  \pscircle[fillstyle=solid,%
            fillcolor=RedOrange]%
           (3,2.3){0.5}
  \pspolygon[fillstyle=solid,%
             fillcolor=mygray]%
            (0,2)(1,2.2)(2,2.5)%
            (3,2.2)(4,2.4)(5,2.5)%
            (6,2.2)(7,2.2)(8,2.4)(9,2)
```


The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

```
    \psframe[fillstyle=solid,%
             fillcolor=mygreen]%
            (0,0)(9,2)
  \end{pspicture}
```



Online LaTeX Tutorial
Part II – Graphics
PSTricks

The Indian TeX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

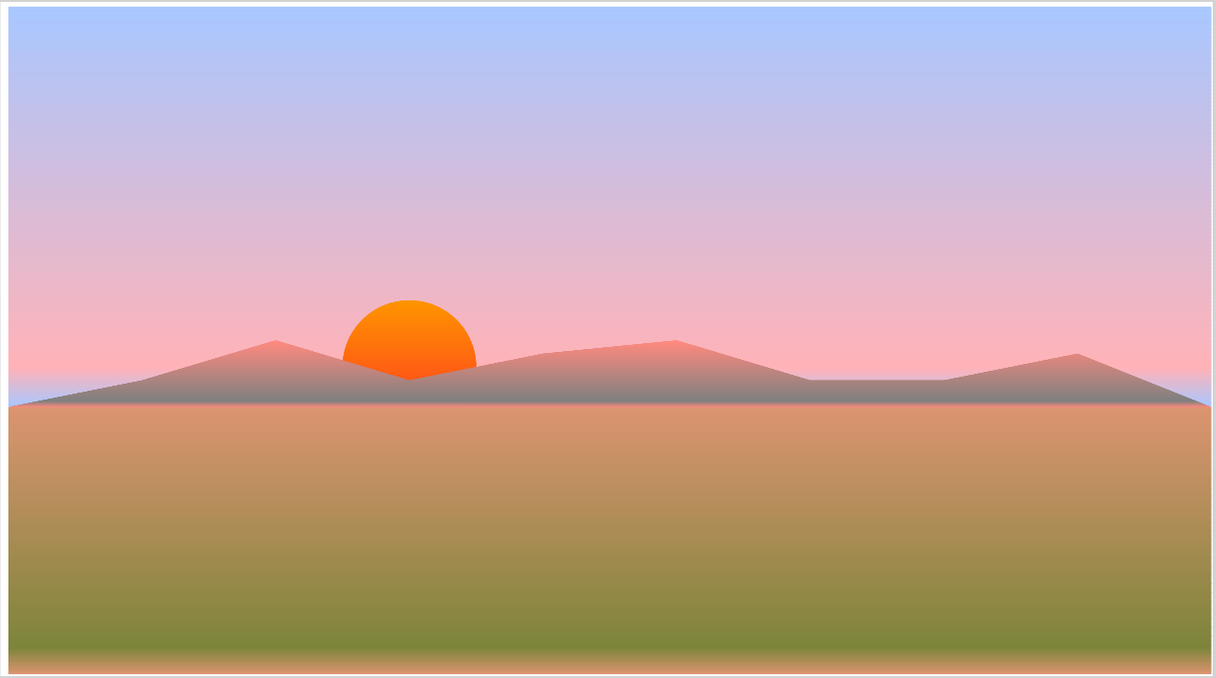
2.5. From one color to another

There's yet another `fillstyle` which is available, if we use the package `pst-grad`. This style is called `gradient` and it allows us to fill a closed region using *two* colors, the color gradually shifting from one to the other. We do this by setting color names to the parameters `gradbegin` and `gradend`. The example below shows how we can add more "effects" to the landscape we'd drawn earlier:

```
\definecolor{myblue}{rgb}{0.66,0.78,1.00}
\definecolor{mypink}{rgb}{1.00,0.70,0.72}
\definecolor{mygreen}{rgb}{0.49,0.52,0.23}
\begin{pspicture}(0,0)(9,5)
  \psframe[linestyle=none,%
           linewidth=0pt,%
           fillstyle=gradient,%
           gradbegin=myblue,%
           gradend=mypink]%
          (0,2)(9,5)
 \pscircle[linestyle=none,%
           linewidth=0pt,%
           fillstyle=gradient,%
           gradbegin=YellowOrange,%
           gradend=RedOrange]%
          (3,2.3){0.5}
\pspolygon[linestyle=none,%
           linewidth=0pt,%
           fillstyle=gradient,%
           gradbegin=Melon,%
           gradend=Gray]%
          (0,2)(1,2.2)(2,2.5)(3,2.2)(4,2.4)%
          (5,2.5)(6,2.2)(7,2.2)(8,2.4)(9,2)
\psframe[linestyle=none,
         linewidth=0pt,
         fillstyle=gradient,
```


The Indian TeX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

```
            gradbegin=Tan,
            gradend=mygreen]
           (0,0)(9,2)
\end{pspicture}
```



By default, this style of filling starts with the `gradbegin` color from the top, gets to the `gradend` color *near* the bottom and again starts with the `gradbegin` color. (If you look at the picture above closely, you can see that the sky goes from blue to pink and there's a small strip of blue again after the pink. The same thing can be seen in the grass also.) Just where the `gradend` color appears is controlled by the `gradmidpoint` parameter, which can take a number between 0 and 1 as its value. The default value is 0.9. See the effect of setting this to 1 in the picture above:

The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

```
\begin{center}
  \definecolor{myblue}{rgb}{0.66,0.78,1.00}
  \definecolor{mypink}{rgb}{1.00,0.70,0.72}
  \definecolor{mygreen}{rgb}{0.49,0.52,0.23}
  \begin{pspicture}(0,0)(9,5)
      \psframe[linestyle=none,linewidth=0pt,%
                fillstyle=gradient,gradbegin=myblue,%
              gradend=mypink,gradmidpoint=1]%
             (0,2)(9,5)
    \pscircle[linestyle=none,%
              linewidth=0pt,%
              fillstyle=gradient,%
              gradangle=0,%
              gradbegin=YellowOrange,%
              gradend=RedOrange]%
             (3,2.3){0.5}
    \pspolygon[linestyle=none,%
               linewidth=0pt,%
               fillstyle=gradient,%
               gradbegin=Melon,%
               gradend=Gray,%
               gradmidpoint=1]%
              (0,2)(1,2.2)(2,2.5)(3,2.2)(4,2.4)%
              (5,2.5)(6,2.2)(7,2.2)(8,2.4)(9,2)
    \psframe[linestyle=none,%
             linewidth=0pt,%
             fillstyle=gradient,%
             gradbegin=Tan,%
             gradend=mygreen,%
             gradmidpoint=1]%
            (0,0)(9,2)
    \psline[linecolor=Tan](0,2)(9,2)
  \end{pspicture}
\end{center}
```

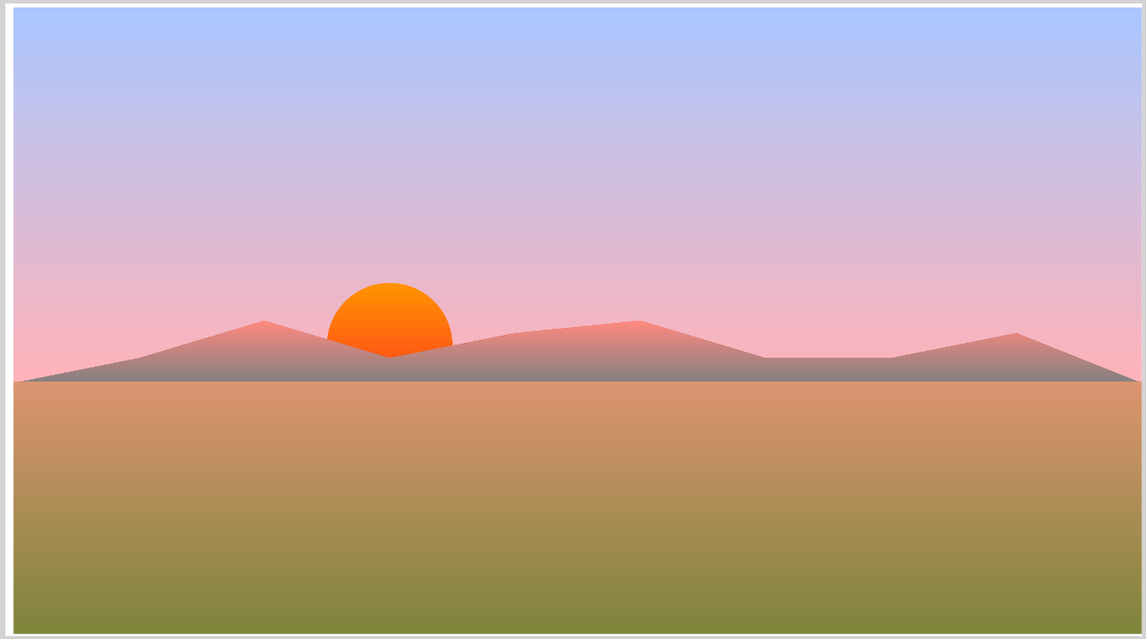
Colorful Tricks

Online LaTeX Tutorial

Part II – Graphics

PSTricks

The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in



The angle of color transition is set by the parameter `gradengle` with default value 0. The example below shows our landscape with different values for this parameter:

```
\begin{center}
  \definecolor{myblue}{rgb}{0.66,0.78,1.00}
  \definecolor{mypink}{rgb}{1.00,0.70,0.72}
  \definecolor{mygreen}{rgb}{0.49,0.52,0.23}
  \begin{pspicture}(0,0)(9,5)
    \psframe[linestyle=none,%
             linewidth=0pt,%
             fillstyle=gradient,%
             gradangle=350,%
             gradbegin=myblue,%
             gradend=mypink,%
             gradmidpoint=1]%
           (0,2)(9,5)
```


The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

```
  \pscircle[linestyle=none,%
            linewidth=0pt,%
            fillstyle=gradient,%
            gradangle=0,%
            gradbegin=YellowOrange,%
            gradend=RedOrange]%
            (3,2.3){0.5}
  \pspolygon[linestyle=none,%
             linewidth=0pt,%
             fillstyle=gradient,%
             gradangle=90,%
             gradbegin=Melon,%
             gradend=Gray,%
             gradmidpoint=1]%
             (0,2)(1,2.2)(2,2.5)(3,2.2)(4,2.4)%
             (5,2.5)(6,2.2)(7,2.2)(8,2.4)(9,2)
   \psframe[linestyle=none,%
            linewidth=0pt,%
            fillstyle=gradient,%
            gradangle=10,%
            gradbegin=Tan,%
            gradend=mygreen,%
            gradmidpoint=1]%
            (0,0)(9,2)
  \psline[linecolor=Tan](0,2)(9,2)
\end{pspicture}
\end{center}
```


Online LaTeX Tutorial

Part II – Graphics

PSTricks

The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

With this, we close our discussion on colors. But the general discussion on PSTricks is far from over.

Online LaTeX Tutorial
Part II – Graphics
PSTricks

The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in