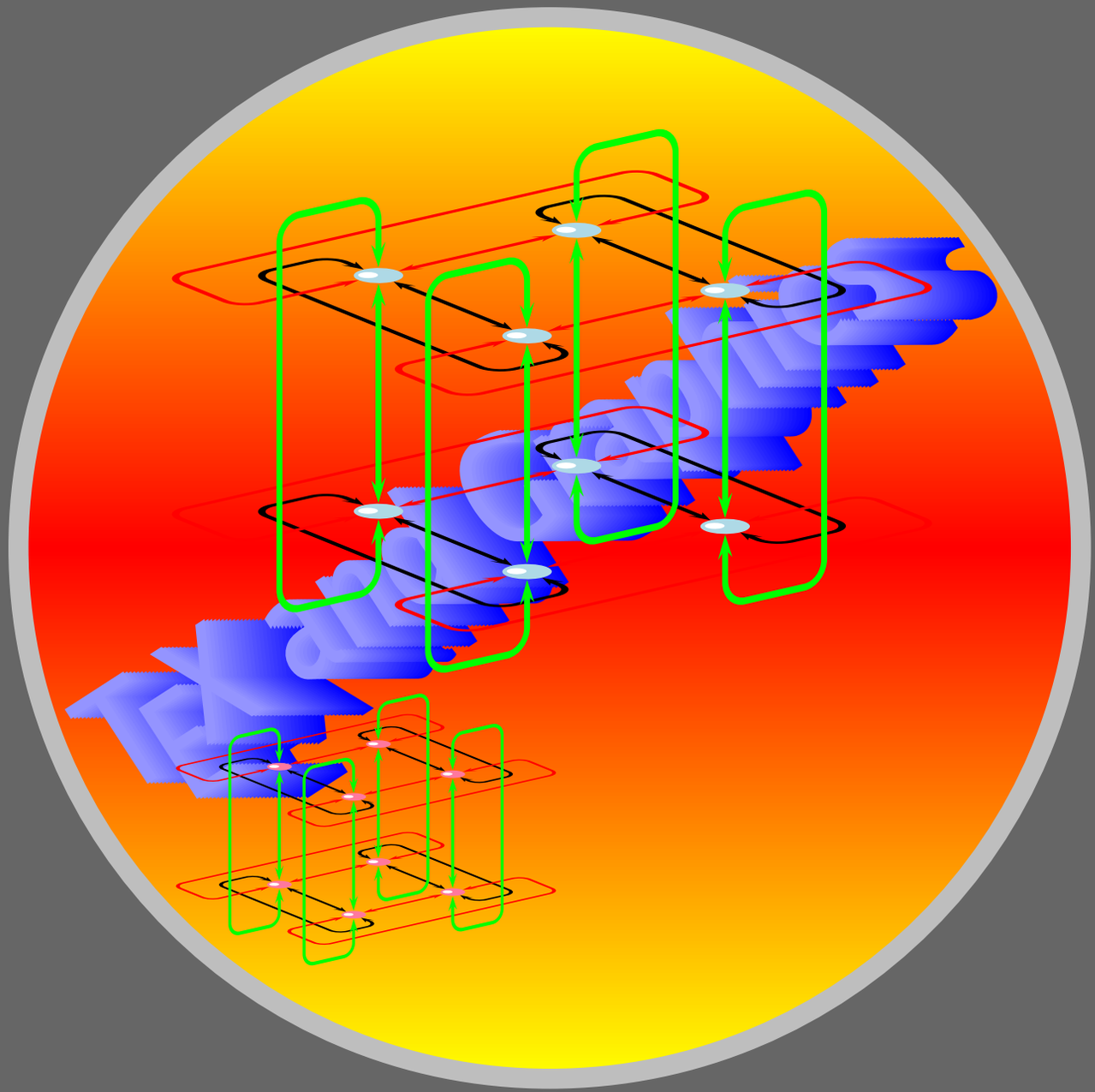


More on Coordinates

Online LaTeX Tutorial

Part II – Graphics

PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

This document is generated by PDFTeX with hyperref, pstricks, pdftricks and pdfscreen packages in an intel PC running GNU/LINUX and is released under LPPL

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

5. More on Coordinates

We have seen that in PSTricks, everything is done with coordinates. We now take a closer look at coordinates and see how we can track and manipulate them. It maybe a good idea to glance back at the first chapter, where we've discussed coordinates in some detail.

Online LaTeX Tutorial

Part II – Graphics

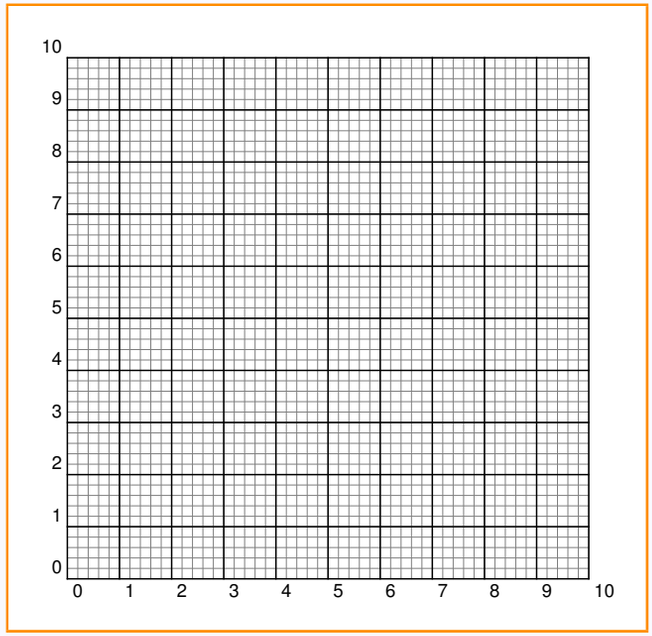
PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

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

5.1. Coordinate grids

To position objects where we want in a picture, we must specify the coordinates. Thus we must imagine an invisible "coordinate grid" (that is, a "graph paper") underlying our picture. But it'd be nice to have the coordinate grid visible, when we first draw a picture. The command `\psgrid` draws such a grid for us; by default, this command draws a 10×10 grid as shown below:



The dimensions of the grid and the positioning of the numbers denoting the intervals can be controlled by specifying coordinates: thus

$$\texttt{\textbackslash psgrid}(x_0, y_0)(x_1, y_1)(x_2, y_2)$$

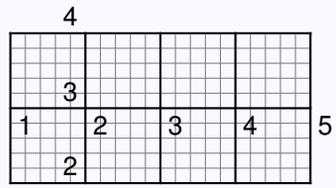
produces a grid with (x_1, y_1) and (x_2, y_2) as opposing corners, and the numbers denoting the x-coordinates running along the line with y-coordinate y_0 and the numbers denoting the y-coordinates running along the line with x-coordinate x_0. Maybe the idea is better understood by an

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

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

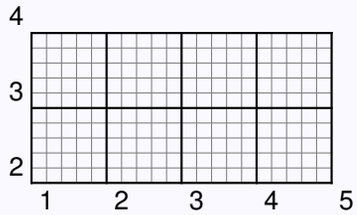
example:

```
\psgrid(2,3)(1,2)(5,4)
```



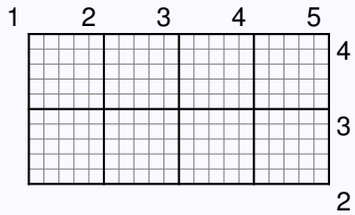
If we specify only *two* pairs of coordinates in a `\psgrid` command, then these are used for opposing corners of the grid and the first pair is used for positioning the numbers, as can be seen from the next example:

```
\psgrid(1,2)(5,4)
```



Note also that the position of the labels with respect to the reference lines (left/right, above/below) is determined by the *order* of specifying the corners. Compare the above example with the one below:

```
\psgrid(5,4)(1,2)
```

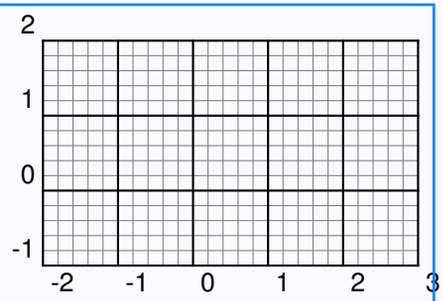


Within a `pspicture` environment, the command `\psgrid` without any coordinates specified, uses the coordinates of the `pspicture`, as shown below:

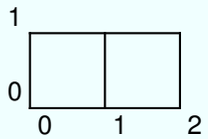
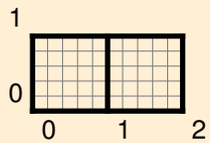
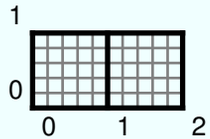
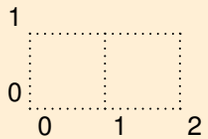
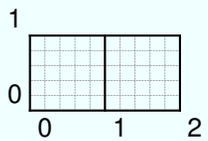
E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

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

```
\begin{pspicture}(-2,-1)(3,2)
  \psgrid
\end{pspicture}
```



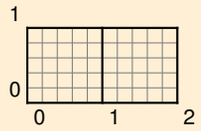
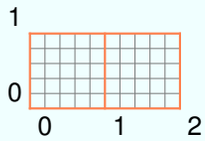
There are various parameters which control the look of the grid which can be tweaked to produce custom grids. See the table below:

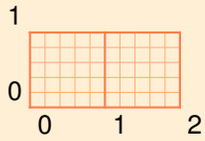
PARAMETER	MEANING	DEFAULT	EXAMPLE
`subgriddiv`	The number of subdivisions of the main grid	5	`\psgrid[subgriddiv=1]% (0,0)(2,1)` 
`gridwidth`	The width of lines in the main grid	0.8 pt	`\psgrid[gridwidth=2pt]% (0,0)(2,1)` 
`subgridwidth`	The width of lines in the subgrid	0.4 pt	`\psgrid[gridwidth=2pt,% subgridwidth=1pt]% (0,0)(2,1)` 
`griddots`	If this number is positive, then the main grid lines are dotted, with that many dots per division	0	`\psgrid[griddots=10,% subgriddiv=1]% (0,0)(2,1)` 
`subgriddots`	If this number is positive, then the subgrid lines are dotted, with that many dots per division	0	`\psgrid[subgriddots=10]% (0,0)(2,1)` 

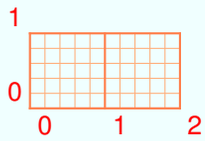
Online LaTeX Tutorial
Part II – Graphics
PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

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

PARAMETER	MEANING	DEFAULT	EXAMPLE
`gridlabels`	Font size of the numbers used to mark the grid	10 pt	`\psgrid[gridlabels=8pt]%` `(0,0)(2,1)` 
`gridcolor`	The color of the main grid lines	black	`\psgrid[gridcolor=Peach]%` `(0,0)(2,1)` 

`subgridcolor`	The color of the subgrid lines	black	`\psgrid[gridcolor=Peach,%`
			`subgridcolor=Apricot]%`
			`(0,0)(2,1)` 

`gridlabelcolor`	The color of the numbers used to mark the grid	black	`\psgrid[gridcolor=Peach,%`
			`subgridcolor=Apricot,%`
			`gridlabelcolor=Red]%`
			`(0,0)(2,1)` 

Another important parameter for `\psgrid` is `unit`. Since this parameter affects not only `psgrid`, but the entire picture, we'll consider it separately.

Online LaTeX Tutorial
Part II – Graphics
PSTricks

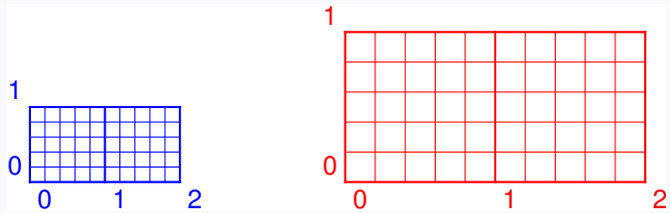
E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to `tutorialteam@tug.org.in`

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

5.2. Changing size

We've mentioned somewhere in the first chapter that the default unit in PSTricks is 1 cm, so that a point specified by $(2,3)$ is 2 centimeters away from the y-axis and 3 centimeters away from the x-axis. This can be changed by setting the `unit` parameter as in the example below:

```
\begin{pspicture}(0,0)(2,1)
  \psgrid[gridcolor=Blue,%
          subgridcolor=Blue,%
          gridlabelcolor=Blue]
\end{pspicture}
\hspace{2cm}
\begin{pspicture}(0,0)(2,1)
  \psgrid[unit=2cm,%
          gridcolor=Red,%
          subgridcolor=Red,%
          gridlabelcolor=Red]
\end{pspicture}
```



This can be used to "scale" a picture as illustrated in the next example:

```
\begin{pspicture}(0,0)(3,2)
  \pspolygon[linecolor=Blue]%
            (0,0)(2,0)(1,1)
\end{pspicture}
\hspace{2cm}
\begin{pspicture}(0,0)(3,2)
  \pspolygon[unit=1.5cm,%
            linecolor=Red]%
            (0,0)(2,0)(1,1)
\end{pspicture}
```

Online LaTeX Tutorial
Part II – Graphics
PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

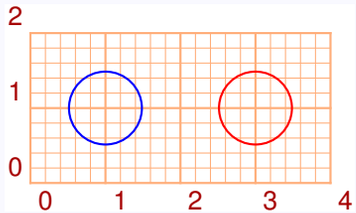
The Indian TEX Users Group
Floor III, SJP Buildings, Cotton Hills
Trivandrum 695014, INDIA
http://www.tug.org.in

Instead of scaling by the *same* amount horizontally and vertically, we can have unequal scaling by setting the `xunit` and `yunit` separately, as shown below:

```
\begin{pspicture}(0,0)(3,2)
  \pspolygon[linecolor=Blue]%
            (0,0)(2,0)(1,1)
\end{pspicture}
\hspace{2cm}
\begin{pspicture}(0,0)(3,2)
  \pspolygon[xunit=1.5cm,%
             yunit=0.5cm,%
             linecolor=Red]%
             (0,0)(2,0)(1,1)
\end{pspicture}
```

Note that the `xunit` and `yunit` settings do not affect the *radius* of circles (but they do affect the center) as illustrated below:

```
\begin{pspicture}(0,0)(4,2)
  \psgrid{gridcolor=Apricot,%
          gridlabelcolor=Mahogany,%
          subgridcolor=Apricot}
  \pscircle[linecolor=Blue](1,1){0.5}
  \pscircle[xunit=1.5cm,%
            yunit=0.5cm,%
            linecolor=Red]%
            (2,2){0.5}
 \end{pspicture}
```



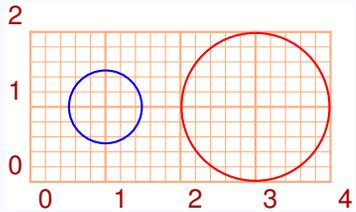
The radius can also be scaled by setting the “runit” parameter as in the next example:

Online LaTeX Tutorial
Part II – Graphics
PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

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

```
\begin{pspicture}(0,0)(4,2)
  \psgrid{gridcolor=Apricot,%
          gridlabelcolor=Mahogany,%
          subgridcolor=Apricot}
  \pscircle[linecolor=Blue](1,1){0.5}
  \pscircle[xunit=1.5cm,%
            yunit=0.5cm,%
            runit=2cm,%
            linecolor=Red]%
            (2,2){0.5}
 \end{pspicture}
```



Note that the parameter `unit` controls `xunit`, `yunit` *and* `runit`.

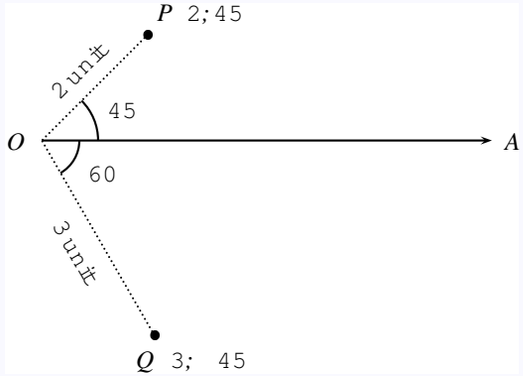
Online LaTeX Tutorial
Part II – Graphics
PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

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

5.3. Another type of coordinates

The Cartesian (or is it Fermatian?) method of using distances from two reference lines is not the only way of labeling points in a plane. Another device mathematicians use is to fix a a point O and a line OA through it and then label each point P by the distance OP and the angle AOP as shown below:



If the distance OP is equal to r and $\angle AOP$ is equal to θ, then r and θ are said to be the *polar coordinates* of P and P is labeled (r, θ). Thus in the picture above, P has polar coordinates $(2, 45)$ and Q has polar coordinates $(3, -60)$. Note that Q can also be represented as $(3, 300)$ (and P as $(2; 405)$, for that matter).

We can specify points using polar coordinates in PSTricks, by invoking the command

```
\SpecialCoor
```

Polar coordinates are specified as `(r;a)` where `r` is the distance and `a` is the angle. (Note that the separator is a *semicolon* and not a comma as in Cartesian coordinates.)

Polar coordinates are very convenient in certain contexts. Look at the example below:

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

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

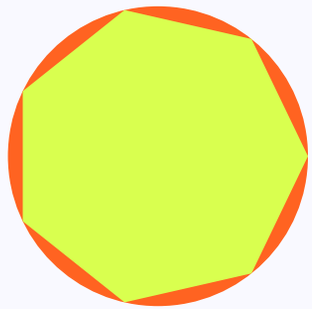
```
\begin{pspicture}(-2,-2)(2,2)
  \pscircle*[linecolor=Lavender](0,0){2}
  \SpecialCoor
  \pspolygon*[linecolor=CornflowerBlue]%
       (2;234)(2;90)(2;306)(2;162)(2;18)
\end{pspicture}
```

By default, angles in polar coordinates are to be specified in degrees; but this can be changed by the command

`\degrees[`*number*`]`

where *number* is the number of parts into which the circle is divided. Thus for example, a regular heptagon can be easily drawn (without calculating the actual angles), by specifying `\degrees[7]`, as in the example below:

```
\begin{pspicture}(-2,-2)(2,2)
  \pscircle*[linecolor=Orange](0,0){2}
  \SpecialCoor
  \degrees[7]
  \pspolygon*[linecolor=GreenYellow]%
   (2;1)(2;2)(2;3)(2;4)(2;5)(2;6)(2;7)
\end{pspicture}
```



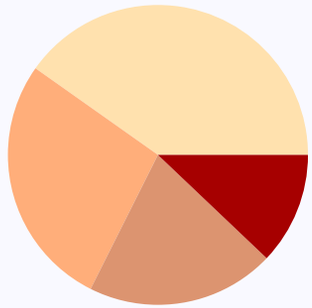
The command `\degrees` can be used even without invoking the `\SpecialCoor`. Thus `\degrees[100]` is a great help in drawing pie charts, where the data is given in percents, as in the example below:

Online LaTeX Tutorial
Part II – Graphics
PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

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

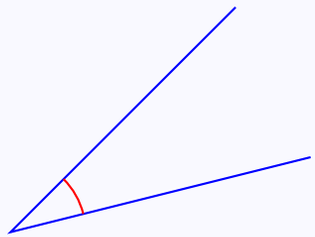
```
\definecolor{PaleApricot}{cmyk}
   {0,0.12,0.32,0}
\begin{pspicture}(-2,-2)(2,2)
  \degrees[100]
  \pswedge*[linecolor=PaleApricot]
    (0,0){2}{0}{40.2}
  \pswedge*[linecolor=Apricot]
    (0,0){2}{40.2}{67.6}
  \pswedge*[linecolor=Tan]
    (0,0){2}{67.6}{87.9}
  \pswedge*[linecolor=Mahogany]
    (0,0){2}{87.9}{100}
\end{pspicture}
```



Angles can be specified in radians by using the command `\radians`. It is equivalent to `\degrees[6.28319]`. (Remember that $\dot{A}$ radians = 180° and that $\dot{A}$ is approximately equal to 3.141592.)

Again in `\SpecialCoor`, angles can be specified in some other ways. We can specify a pair of coordinates indicating the *direction* of the angle as illustrated in the example below. (Note in particular the braces { } surrounding the coordinate pair.)

```
\begin{pspicture}(0,0)(4,3)
  \psline[linecolor=Blue](4,1)(0,0)(3,3)
  \SpecialCoor
  \psarc[linecolor=Red](0,0){1}{(4,1)}{(3,3)}
\end{pspicture}
```



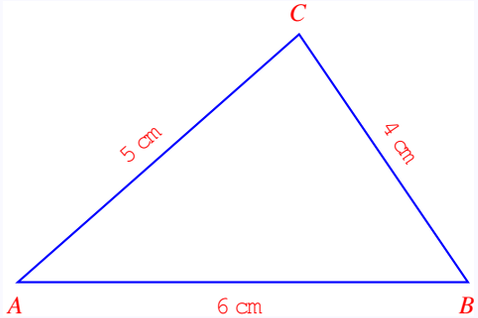
Another way of specifying an angle is to use raw PostScript code which evaluates a number. The code should be preceded by `!`. For example, sup-

Online LaTeX Tutorial
Part II – Graphics
PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

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

pose we want to draw a triangle with sides 2 cm, 3 cm and 4 cm as shown below:



We can specify A as $(0,0)$ and B as $(6,0)$, but what about C? If $\angle A = \theta$, then C has polar coordinates $(5,\theta)$. Now from elementary trigonometry, we have

$$A = 2\tan^{-1}\sqrt{\frac{(s-b)(s-c)}{s(s-a)}}$$

in any triangle ABC with $BC = a$, $CA = b$ and $AB = c$, where $s = \frac{1}{2}(a+b+c)$. For our triangle above, this works out to be

$$A = 2\tan^{-1}\sqrt{\frac{2.5\times 1.5}{7.5\times 3.5}}$$

This computation can be done by PostScript and in the syntax of this language, it is written

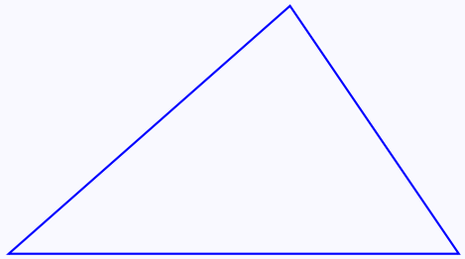
```
2.5 1.5 mul sqrt 7.5 3.5 mul sqrt atan 2 mul
```

(We will explain this a bit in the appendix to this chapter.) Now by the device of including PostScript code in an angle specification using `!`, we can produce the above triangle by

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

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

```
\begin{pspicture}(0,-0.5)(6,3.5)
  \SpecialCoor
  \pspolygon[linecolor=Blue]%
    (0,0)(6,0)%
    (5;!2.5 1.5 mul sqrt
     7.5 3.5 mul sqrt atan 2 mul)
\end{pspicture}
```



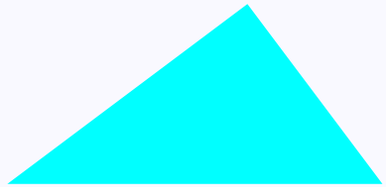
(What about the “labels” for the vertexes and the sides?, Well, that’s another story, better told in a separate chapter.)

Perhaps it is better to have a LaTeX macro to draw a triangle with specified sides. Here’s one:

```
\newcommand{\pstrilateral}[4][]{%
  \SpecialCoor
  \pspolygon[#1](0,0)(#4,0)%
   (#3;!#2 #3 add #4 sub #2 #3 sub #4 add mul sqrt
        #2 #3 add #4 add #3 #4 add #2 sub mul sqrt atan 2 mul)}
```

The command `\pstrilateral` can then be used to draw for example, a “solid” cyan colored triangle of sides 3 cm, 4 cm and 5 cm as shown below:

```
\begin{pspicture}(0,-0.5)(5,2.5)
      \pstrilateral[fillstyle=solid,%
                    fillcolor=Cyan,%
                    linestyle=none]%
                    {3}{4}{5}
    \end{pspicture}
```



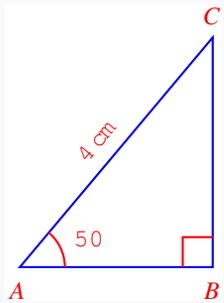
Online LaTeX Tutorial
Part II – Graphics
PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

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

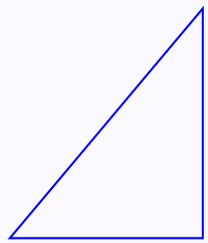
5.4. More special coordinates

Under `\SpecialCoor`, not only angles, but the entire pair of coordinates can be specified using raw PostScript code using the `!` signifier. For example, suppose we want to draw a right angled triangle of hypotenuse 4 cm and one angle equal to 50°, as shown below:



It can be easily seen that B has coordinates $(4\cos 50°, 0)$ and C has coordinates $(4\cos 50°, 4\sin 50°)$. The triangle (sans the labels) can be drawn by writing these coordinates in PostScript as below:

```
\begin{pspicture}(0,-0.5)(3,3.5)
  \SpecialCoor
  \pspolygon[linecolor=Blue]%
            (0,0)%
            (!50 cos 4 mul 0)%
            (!50 cos 4 mul 50 sin 4 mul)
  \end{pspicture}
```



Here, the top vertex (C in the first figure) can also be specified more simply in polar coordinates as `(4;50)`. There's a simpler way to specify B also. Note that the x-coordinate of B is the same as that of C and its y-coordinate is 0. Under `\SpecialCoor`, we can specify the coordinates of a point by referring to these coordinates (in any form) of two other points

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

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

such that the required point has x-coordinate equal to that of the first point and y-coordinate equal to the y-coordinate of the second point. The general syntax is

(*coordinates1*|*coordinates2*)

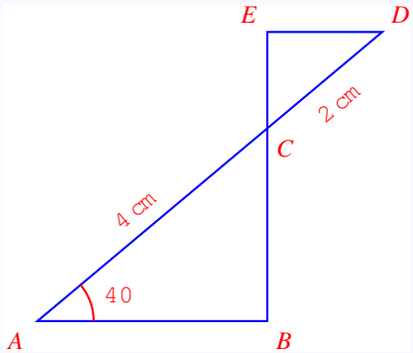
Thus in our example, the point B can be specified as

(4;50|0,0)

(Note that the coordinates of the two reference points are given without enclosing parentheses and a vertical bar | separates these coordinates.) Thus another way of drawing the above triangle is by

```
\begin{pspicture}(0,-1)(3,4)
  \SpecialCoor
  \pspolygon[linecolor=Blue](0,0)(4;50)(4;50|0,0)
\end{pspicture}
```

As an another illustration of this technique, consider the figure below:

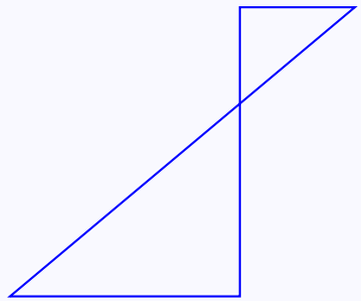


Taking A as $(0,0)$, we can specify C and D by polar coordinates as (4;40) and (6;40). Using the technique just described, B can be specified as (4;40|0,0) and E as (4;40|6;40). Thus this figure (without the labels, of course) can be produced as shown below:

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

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

```
\begin{pspicture}(0,-0.5)(5,4.5)
  \SpecialCoor
  \pspolygon[linecolor=Blue]%
            (0,0)(6;40)(4;40|6;40)(4;40|0,0)
\end{pspicture}
```



There are somewhere ways of specifying coordinates under \SpecialCoor, using “nodes” and these will be described in another chapter which deals with nodes and their connections using the pst-node package.

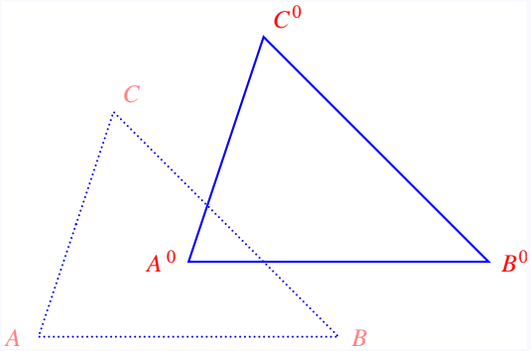
Online LaTeX Tutorial
Part II – Graphics
PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

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

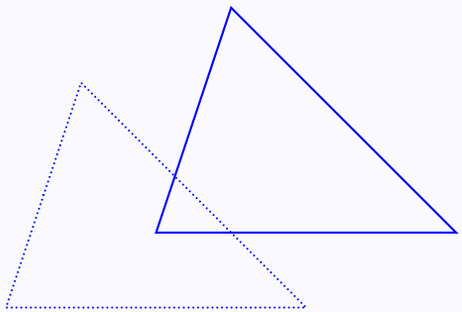
5.5. Changing the system

In drawing pictures, it is sometimes convenient to make some changes to the system of coordinates in the middle. For example, consider the picture below:



The bold triangle $A'B'C'$ is an exact replica of the dotted triangle ABC, only shifted to the right a little. Having drawn ABC, if we can shift the coordinate system to have the origin at A', then the same code could be used to draw $A'B'C'$ also. This can be done by (re)setting the parameter `origin` in the code for drawing $A'B'C'$. Thus the above picture (without the labels, as usual) can be drawn as shown below:

```
\begin{pspicture}(0,0)(6,4)
  \pspolygon[linestyle=dotted,%
             dotsep=1pt,%
             linecolor=Blue]%
            (0,0)(4,0)(1,3)
  \pspolygon[origin={-2,-1},%
             linecolor=Blue]%
            (0,0)(4,0)(1,3)
\end{pspicture}
```

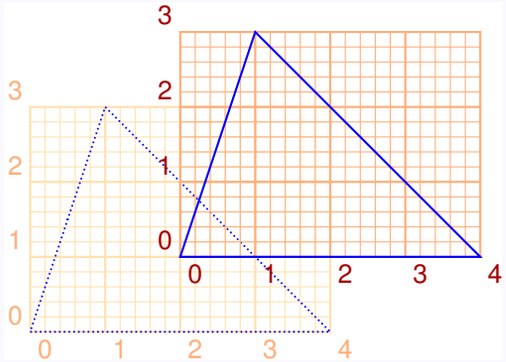


In the second `\pspolygon`, the setting `origin={-2,-1}` translates the co-

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

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

ordinate axes to a new position such that the original origin has coordinates $(-2,-1)$ with respect to this new system ¿This is better illustrated in the picture below, which shows the triangles together with the two coordinate systems.



Note that, the origin of the original system (shown by the pale grid) is $(-2,-1)$ with respect to the new system (shown by the darker grid). In general, the setting

```
origin={x,y}
```

translates the coordinate axes such that the origin of the original system is (x,y) with respect to the new system. In practical terms, this means, if we want the new origin to be at (x,y), set `origin={-x,-y}`. Note also the use of the curly braces `{ }` to enclose the coordinates, instead of the customary parentheses `( )` in this setting.

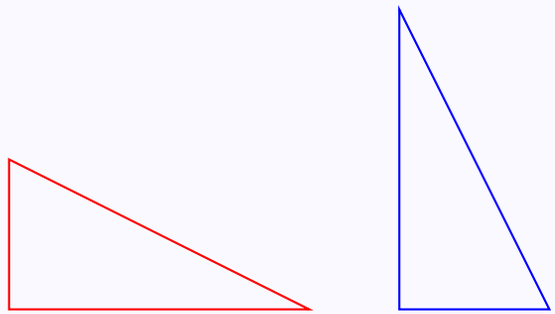
We can also interchange the x and y axes by setting the parameter `\swapaxes` to `true`. (Its default value is `false`.) This is helpful in changing the orientation of a picture. Look at the example below:

Online LaTeX Tutorial
Part II – Graphics
PSTricks

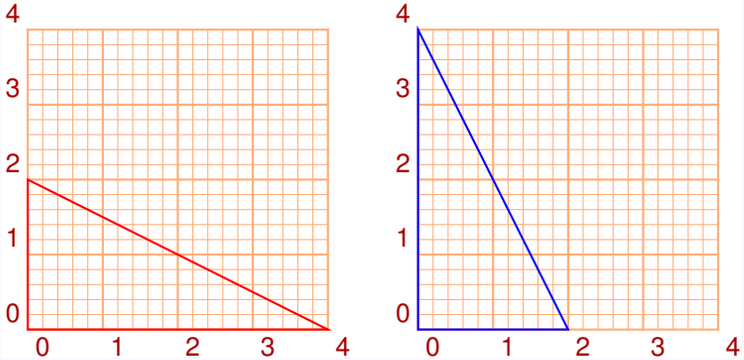
E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

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)
  \pspolygon[linecolor=Red]%
  (0,0)(4,0)(0,2)
\end{pspicture}
\hspace{1cm}
\begin{pspicture}(0,0)(4,4)
  \pspolygon[linecolor=Blue,%
             swapaxes=true]%
            (0,0)(4,0)(0,2)
\end{pspicture}
```



Note that with `\swapaxes=true` in effect, a point with coordinates specified as (a, b) is plotted with x-coordinate b and y-coordinate a. The figures above with the coordinate grids used to draw them makes this clear.



Online LaTeX Tutorial
Part II – Graphics
PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

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

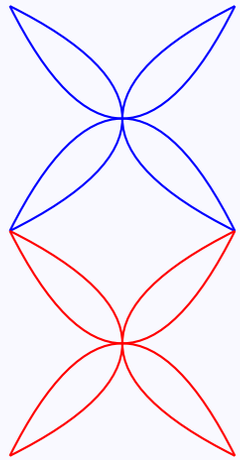
5.6. Setting parameters

Instead of setting the parameters `origin` and `\swapaxes` locally for each object for which we need such effects, we can set them globally with the `\psset` command. This is true for the other graphics parameters such as `linewidth`, `linecolor`, `linestyle` and so on, which we have discussed earlier. The general syntax is

`\psset{`*parameter1=value1*`,`*parameter2=value2*`,...}`

The example below illustrates this:

```
\psset{linecolor=Blue,unit=1.5}
\begin{pspicture}(-1,0)(1,4)
  \parabola(1,1)(0,0)
  \parabola(-1,-1)(0,0)
  \psset{swapaxes=true}
  \parabola(1,1)(0,0)
  \parabola(-1,-1)(0,0)
  \psset{origin={0,2},linecolor=Red}
  \parabola(1,1)(0,0)
  \parabola(-1,-1)(0,0)
  \psset{swapaxes=false}
  \parabola(1,1)(0,0)
  \parabola(-1,-1)(0,0)
\end{pspicture}
```



Online LaTeX Tutorial
Part II – Graphics
PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to tutorialteam@tug.org.in

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

Appendix—Math in PostScript

We've given a few examples of specifying coordinates using raw PostScript code in the section on "Special Coordinates". Here we give a list of mathematical operators available in this language and their syntax.

OPERATOR	MEANING	syntax	example	
			PostScript CODE	VALUE
`add`	sum of two numbers	*number1* *number2* `add`	`7 2 add`	9
`sub`	difference of two numbers	*number1* *number2* `sub`	`7 2 sub`	5
`mul`	product of two numbers	*number1* *number2* `mul`	`7 2 mul`	14
`div`	quotient of two numbers	*number1* *number2* `sub`	`7 2 div`	3.5
`exp`	power of a number	*number1* *number2* `exp`	`7 2 exp`	49
`idiv`	integral part of the quotient of two integers	*number1* *number2* `idiv`	`7 2 idiv`	3
`mod`	reminder obtained on dividing an integer by an integer	*number1* *number2* `mod`	`7 2 mod`	1
`sqrt`	square root of a number	*number* `sqrt`	`16 sqrt`	4
`neg`	negative of a number	*number* `neg`	`7 neg`	−7
`abs`	absolute value of a number	*number* `abs`	`-7 abs`	7
`ceiling`	smallest integer greater than or equal to a number	*number* `ceiling`	`7.6 ceiling`	8

Online LaTeX Tutorial
Part II – Graphics
PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team. Comments and suggestions may be mailed to `tutorialteam@tug.org.in`

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

OPERATOR	MEANING	syntax		example	
				PostScript CODE	VALUE
floor	largest integer less than or equal to a number	*number*	floor	7.6 floor	7
round	round a number to the nearest integer	*number*	round	7.6 round 7.2 round	8 7
sin	sine of number in degrees	*number*	sin	30 sin	0.5
cos	cosine of number in degrees	*number*	cos	60 cos	0.5
atan	inverse tangent of number in degrees	*number*	atan	1 atan	45
ln	natural logarithm (base e) of number	*number*	ln	2.71828182 ln	1
log	logarithm of number to base 10	*number*	log	100 log	2

Online LaTeX Tutorial
Part II – Graphics
PSTricks

E Krishnan, CV Radhakrishnan and AJ Alex constitute the graphics tutorial team.
Comments and suggestions may be mailed to
tutorialteam@tug.org.in

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