

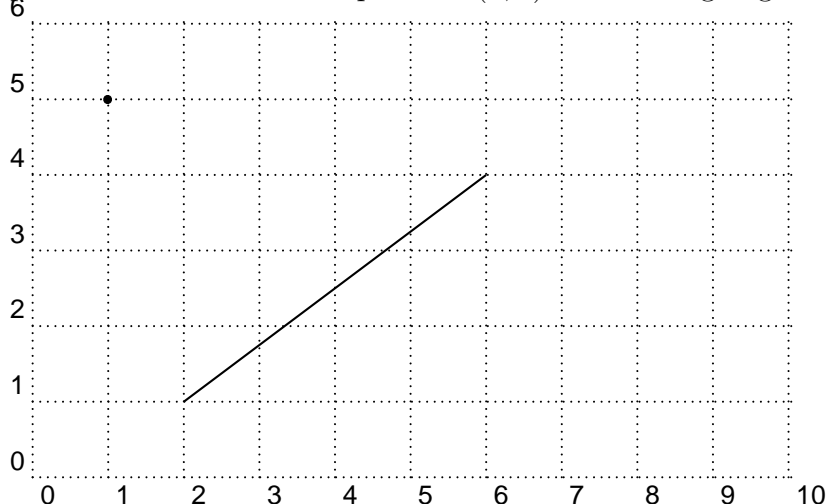
Getting started with PSTricks

written John Murray, 9/6/1998

updated Duncan Smith, 25/6/2003

Suppose you want to draw a simple picture for your \LaTeX document. Draw a grid in the space you want the plot with the gridlines 1cm apart. Place numbers on the grid lines. It is best to start the numbers from the bottom left of the grid. Each point has a reference on the grid in the form (a,b) , in the same way as places are represented on each page of a road map. The first number, or coordinate, represents its position from left to right, while the second coordinate represents its position from bottom to top.

The following picture contains a dot at position $(1,5)$ and a line going from $(2,1)$ to $(6,4)$.



This picture was produced with the following commands from PSTricks. (**Note:** the command `\psgrid ...` will produce the grid. It is not a bad idea when trying to draw objects to produce this grid by itself on paper, draw the object by hand on the grid, and then use the grid reference points and PSTricks commands to draw it properly. You can then comment out the grid command.) Your document will look like:

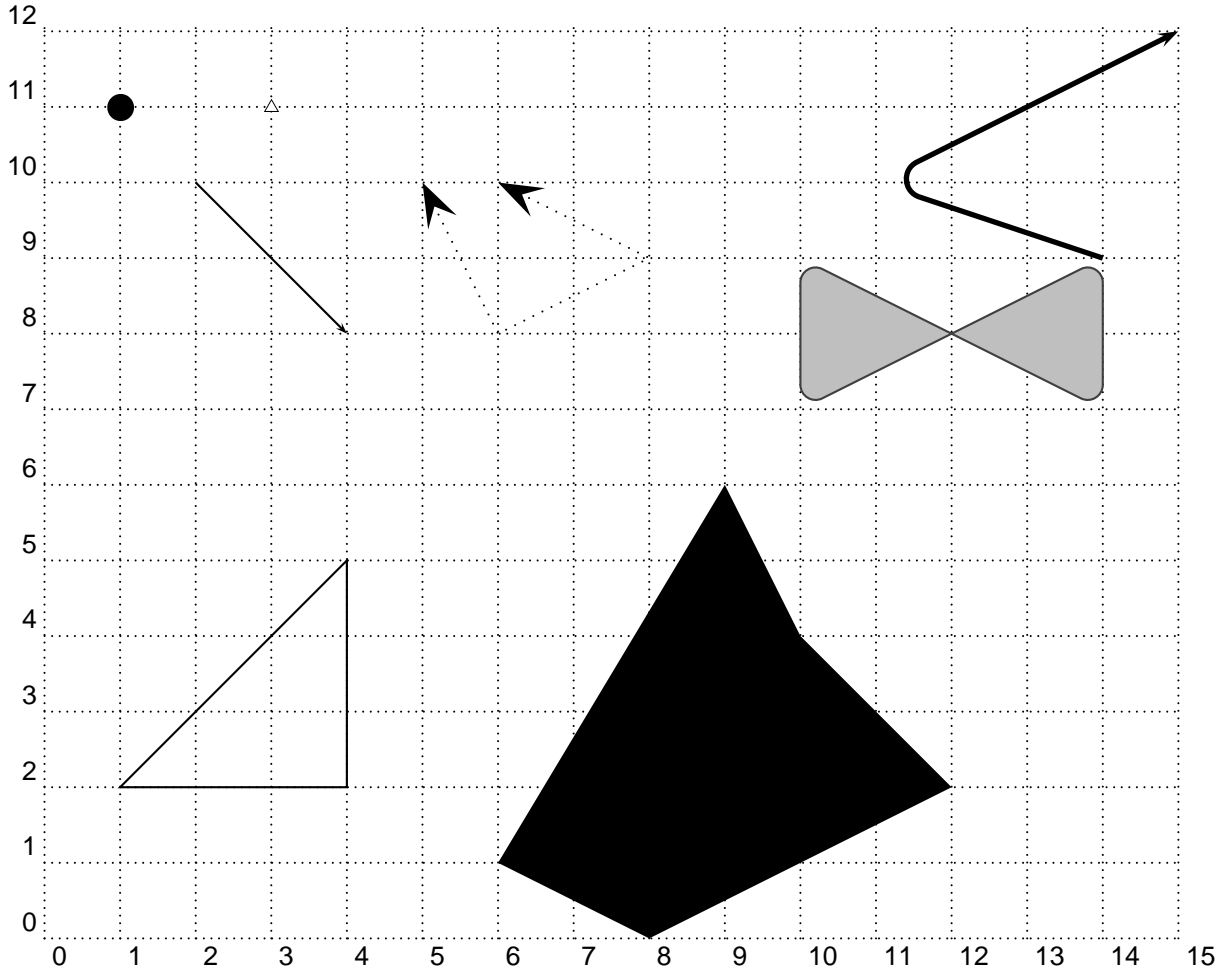
```
\documentclass[12pt]{article}
\usepackage{pstricks}
\begin{document}
. . .
\begin{center}
\begin{pspicture}(0,0)(10,6)
\psgrid[subgriddiv=1,griddots=10,gridlabels=10pt](0,0)(10,6)
\psdot*(1,5)
\psline(2,1)(6,4)
\end{pspicture}
\end{center}
```

Some details of the previous picture: the environment

```
\begin{pspicture}(0,0)(10,6) ... \end{pspicture}
```

creates an invisible grid of the type that has been displayed explicitly above. The first set of coordinates (0,0) gives the bottom left of the grid while the second set (10,6) gives the top right corner. The default dimension for PSTricks is centimetres so we have essentially placed a 10cm by 6cm box on the page. This creates the room for the drawing.

The dot was drawn with the command `\psdot*(1,5)`. The line was drawn with the command `\psline(2,1)(6,4)`. We can vary these a great deal if we want to change some of the defaults.



The commands for this are:

```
\begin{center}
\begin{pspicture}(0,0)(15,12)
\psgrid[subgriddiv=1,griddots=10,gridlabels=10pt](0,0)(15,12)
\psdot*[dotsize=10pt](1,11)
\psdot*[dotsize=5pt,dotstyle=triangle](3,11)
\psline{->}(2,10)(4,8)
\end{pspicture}
\end{center}
```

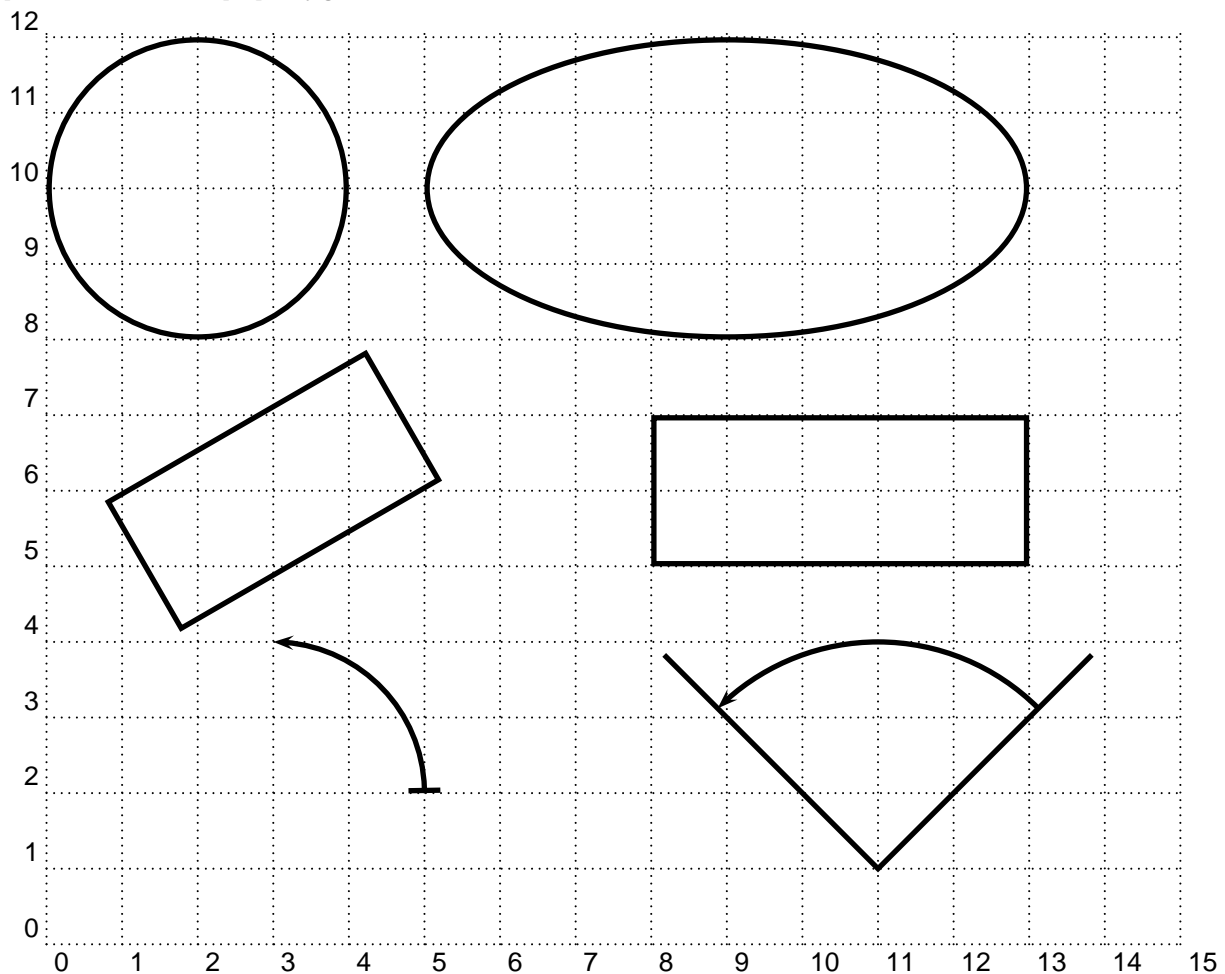
```

\psline[linestyle=dotted, arrowsize=12pt]{<->}(5,10)(6,8)(8,9)(6,10)
\psline[linewidth=2pt, linearc=.25]{<-}(15,12)(11,10)(14,9)
\pspolygon(1,2)(4,5)(4,2)
\pspolygon*(6,1)(9,6)(10,4)(12,2)(8,0)
\pspolygon[linecolor=darkgray,fillstyle=solid, fillcolor=lightgray,
  linearc=0.2](10,7)(10,9)(14,7)(14,9)
\end{pspicture}
\end{center}

```

Notice that the asterisk `*` fills in the object. It uses the same colour as the boundary line. The `\psline` command connects up the points you list, using straight line segments and connecting them in the order given. You can change the default width and style of the commands as above. The `\pspolygon` command does the same as the line command except it joins the first and last points to close up the shape.

There are plenty of other standard shapes and others that you can construct using the `\psline` and `\pspolygon` commands. Here are some standard ones.



These were produced with

```

\begin{center}
\begin{pspicture}(0,0)(15,12)
\psset{linewidth=2pt}
\psgrid[subgriddiv=1,griddots=10,gridlabels=10pt](0,0)(15,12)
\pscircle(2,10){2}
\psellipse(9,10)(4,2)
\psframe(8,5)(13,7)
\rput{30}(3,6){\psframe(-2,-1)(2,1)}
\psarc{|->}(3,2){2}{0}{90}
\psarc{->}(11,1){3}{45}{135}
\pscustom{\translate(11,1)
\SpecialCoor
\psline(4;45)(0,0)(4;135)}
\end{pspicture}
\end{center}

```

Rather than changing the default linewidth to 2pt for each line, I changed the default globally using `\psset`. This can change other defaults including the standard length of 1cm. This is handy when rescaling a diagram. The shape commands have the following form:

`\pscircle(x,y){r}` is a circle, centre (x,y) radius r .

`\psellipse(x,y)(a,b)` is an ellipse, centre (x,y) , horizontal radius a and vertical radius b .

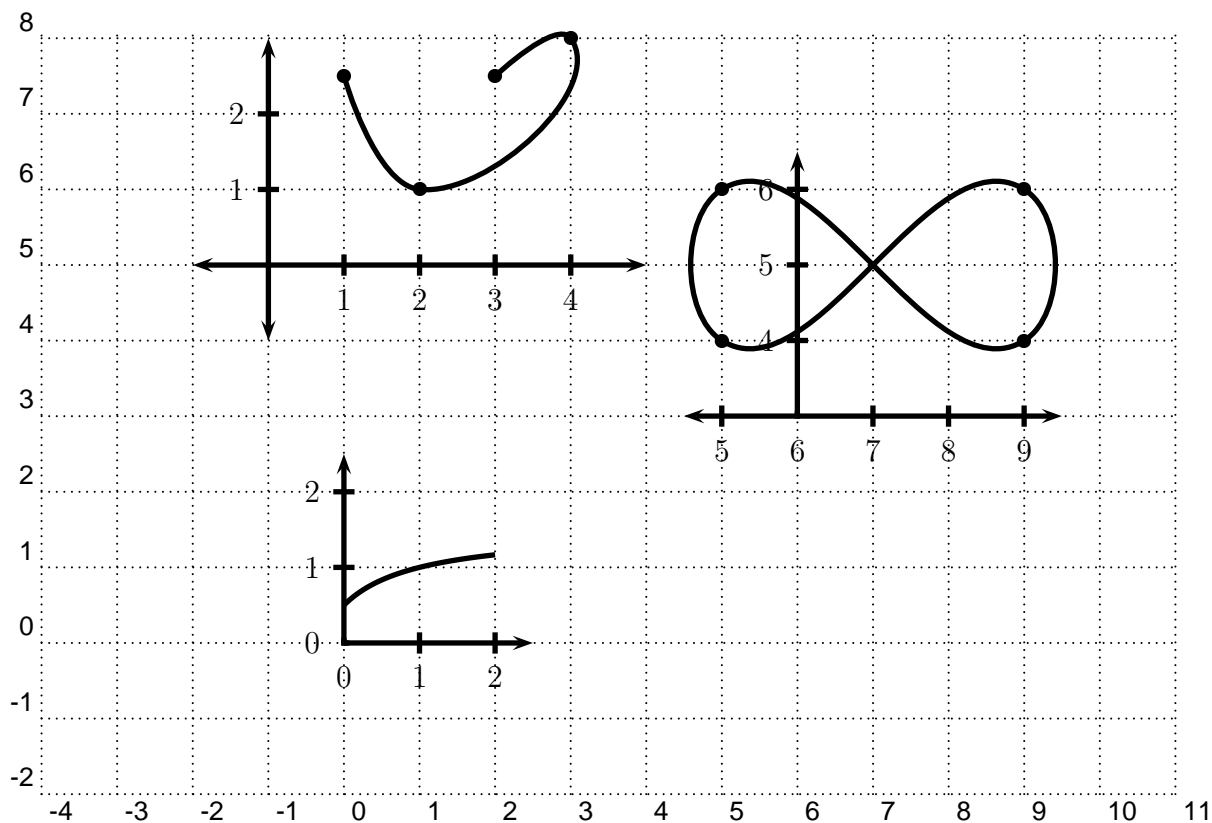
`\psframe(x,y)(a,b)` is a rectangle, bottom left corner at (x,y) , top right corner at (a,b) .

`\psarc(x,y){r}{a}{b}` is an arc of a circle whose centre is (x,y) , radius r , and the arc starts at a degrees and finishes at b degrees.

Another useful feature is to be able to use polar coordinates through `\SpecialCoor`. Then the references $(4;45)$ and $(4;135)$, (notice the ; not a ,) are in polar coordinates. The command `\psline(4;45)(0,0)(4;135)` drew a line between those points. I wanted the wedge at $(11,1)$ so to get it there I translated the axes from $(0,0)$ to $(11,1)$. This had to go in `\pscustom`. You can also rotate objects using the command `\rotate{angle}`. This must also appear in `\pscustom`. Another way of rotating and translating objects is to use the `\rput` command.

`\rput*[refpoint]{angle}(x,y){stuff}` rotates stuff by angle and translates its reference point (by default the centre of stuff) to (x,y) .

Now for axes and curves. For the axes you can either use `\psline{<->}` or standard ones. For the latter you will need to load the package `pst-plot.tex` or `pst-plot.sty`.



These were produced with

```
\begin{center}
\begin{pspicture}(-4,-2)(11,8)
\psset{linewidth=2pt}
\psgrid[subgriddiv=1,griddots=10,gridlabels=10pt](-4,-2)(11,8)
\psaxes{<->}(-1,5)(-2,4)(4,8)
\pscurve[showpoints=true](0,7.5)(1,6)(3,8)(2,7.5)
\psaxes[0x=6,0y=3]{<->}(6,3)(4.5,3)(9.5,6.5)
\psccurve[showpoints=true](5,4)(5,6)(9,4)(9,6)
\psaxes{<->}(0,0)(2.5,2.5)
\psplot{0}{2}{1.5 1 x 1 add div sub}
\end{pspicture}
\end{center}
```

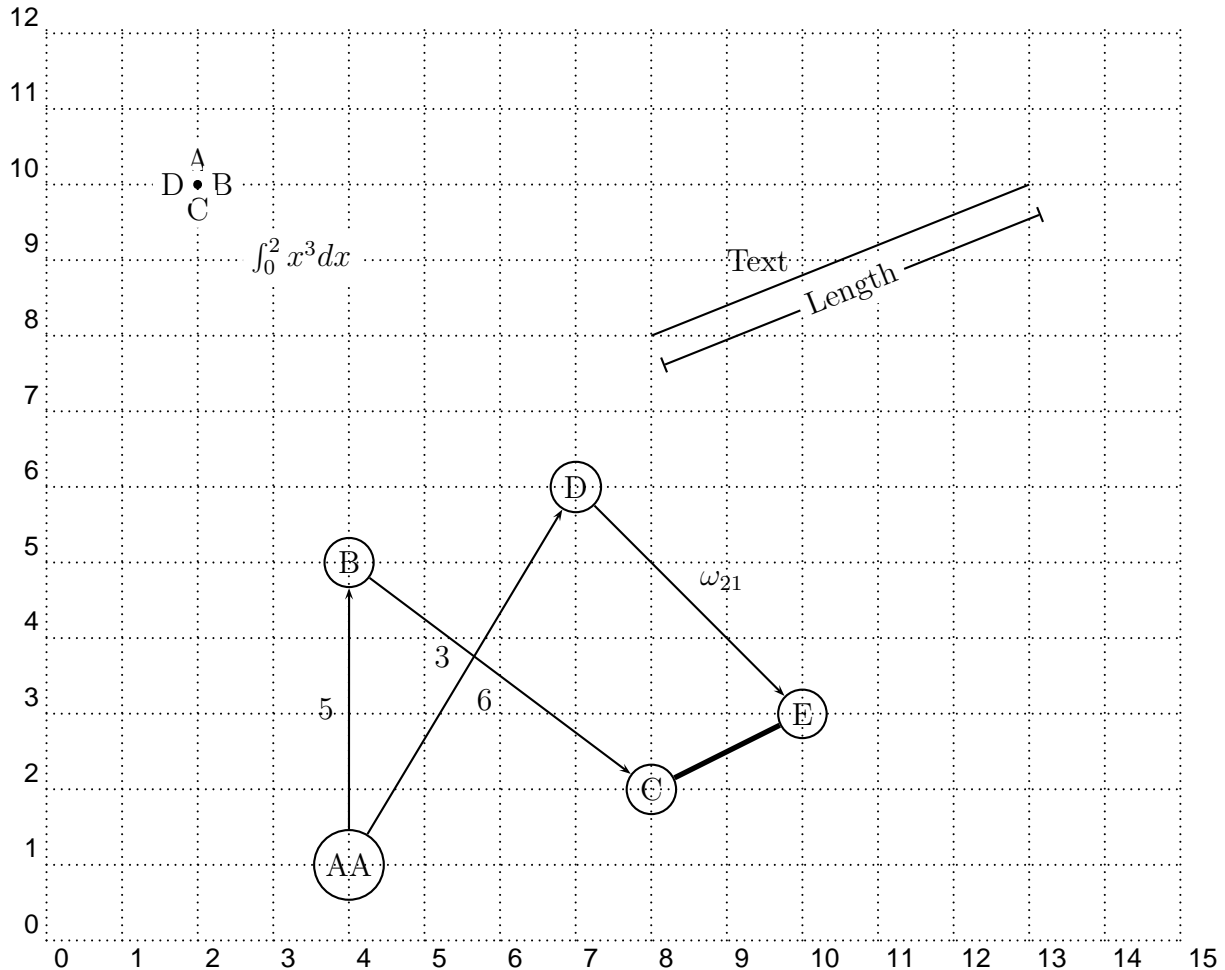
The command `\pscurve` fits a curve through a set of points while `\psccurve` fits a closed curve through them. The axes command

```
\psaxes{<->}(-1,5)(-2,4)(4,8)
```

draws a set of axes that cross at $(-1, 5)$, the bottom left hand endpoint of the invisible box that contains the axes is at $(-2, 4)$, and the top right endpoint of the box is at $(4, 8)$.

The command `\psplot` uses PostScript to draw a curve and incorporates it into the document. You will need to know the mathematical commands from PostScript to be able to use this.

Now let's look at labelling our graphs. We will use some commands for node connections and labelling so need to load `pst-node.tex` or `pst-node.sty`.



These were produced with:

```
\begin{center}
\begin{pspicture}(0,0)(15,12)
\psgrid[subgriddiv=1,griddots=10,gridlabels=10pt](0,0)(15,12)
\psdot*(2,10)
\uput[u](2,10){A}
\uput[r](2,10){B}
\uput[d](2,10){C}
\uput[l](2,10){D}
\uput{1cm}[-45](2,10){$\int_0^2 x^3 dx$}
\psline(8,8)(13,10)
```

```

\uput[1](10,9){Text}
\pcline[offset=-12pt]{|-|}(8,8)(13,10)
\lput*{:U}{Length}
\cnodeput(4,1){A}{AA}
\cnodeput(4,5){B}{B}
\cnodeput(8,2){C}{C}
\cnodeput(7,6){D}{D}
\cnodeput(10,3){E}{E}
\ncline{->}{A}{B}
\Aput{5}
\ncline{->}{B}{C}
\Bput{6}
\ncline{->}{A}{D}
\Aput{3}
\ncline{->}{D}{E}
\Aput{$\omega_{21}$}
\ncline[linewidth=2pt]{C}{E}
\end{pspicture}
\end{center}

```

The command `\uput[u](x,y){stuff}` places the label “stuff” at position (x,y) above the point, *u* stands for up. I will let you guess what *r*, *d*, and *l* stand for. You can place the label at different angles by specifying the degrees of the direction. The `\lput` command places a label half way along the line specified by `\pcline`. The `{:U}` command says that the label should be parallel to the line and the `*` tells it to overwrite the line.

There are plenty of other things that can be done with PSTricks. Some additional information and examples are available from the school’s L^AT_EX web page and in the book “The L^AT_EX graphics companion” which all secretaries in the school should have. The original documentation and other information on PSTricks can be found at

<http://tug.org/applications/PSTricks>

You can now view PSTricks plots approximately correctly with `xdvi`. So to look at them type:

```

latex fred
xdvi fred

```

If you use `xdvi` on this document you get everything OK except some of the labels on the last picture. When that complication arises you should use `ghostview` as follows:

```

latex fred
dvips fred
ghostview fred.ps

```