1. Resources

(a) ImageMagick is a collection of (free) image manipulation tools. You can find out more by visiting http://www.imagemagick.com

(b) The \LaTeX{} Graphics Companion.

- Paperback: 976 pages
- Publisher: Addison-Wesley Pub Co; 2nd edition (August 12, 2007)
- ISBN: 978-0321508928

(c) The PSTricks web site.

http://www.pstricks.de

2. The Graphics Package

(a) Some preliminaries:

To use this package you must include one of the following lines in the preamble of the document

\begin{verbatim}
\usepackage{graphic}
\usepackage{graphicx}
\end{verbatim}

Today I will focus on the enhanced graphics package, graphicx.

Now the following commands allow users to define their own colors

\begin{verbatim}
\definecolor{ForestGreen}{rgb}{0.133,0.545,0.133}
\definecolor{Orange}{rgb}{1.,0.65,0.}
\definecolor{myGray}{rgb}{0.85,0.85,0.85}
\end{verbatim}

so that \texttt{\color{Orange} isn’t this nice and orange} yields

\begin{center}
\textit{isn’t this nice and orange}
\end{center}

One may also use predefined colors so that \texttt{\color{red} what color am I} yields

\begin{center}
\textit{what color am I}
\end{center}

Boxing environments are also available. Thus \texttt{\colorbox{myGray}{we’re in a shaded box}} becomes

\begin{center}
\textbf{we’re in a shaded box}
\end{center}

and \texttt{\fcolorbox{ForestGreen}{who picked this color} yields}

\begin{center}
\textbf{who picked this color}
\end{center}

(b) General image inclusion:

However, the enhanced graphics package allows the user to include imported images via the command

\begin{verbatim}
\includegraphics[options]{graphics}
\end{verbatim}

We will work with the following jpeg image

This above image was included in our document using the command

\begin{verbatim}
\includegraphics[bbllx=0,bbllty=0,bburrx=146,bburry=103]{cave.jpg}
\end{verbatim}

On my system, it is much easier to work with ps or eps files. Using the “convert” command line tool from the ImageMagick resource mentioned above, it is fairly straightforward to convert any image into a Postscript (or Encapsulated Postscript format). Now we can just use

\begin{verbatim}
\hspace{1in}\includegraphics{cave.eps}
\end{verbatim}
Image effects can now be achieved using optional arguments such as

\includegraphics[angle=30]{cave.ps}

or

\includegraphics[scale=1.5]{cave.ps}

or even

\includegraphics[bb=0 0 100 50,clip]{cave.ps}

Image effects can also be combined

3. PSTricks

To use this package you must include the following line in the preamble of the document

\usepackage{calc,pstricks,pstcol,pst-plot}
\usepackage{pst-all}

We should mention that there have been some incompatibilities between the pstcol package (used by PSTricks) and the graphics packages mentioned above.

Using colors with PSTricks is similar to what has already been discussed. The real power of the PSTricks package is the ability to create graphics using LaTeX-like syntax.
(a) Preliminaries

PSTricks provides users with the capability to draw using the familiar syntax of \LaTeX.

\begin{pspicture}(-3.2,2)(2,2)
\psline[linecolor=blue,linewidth=1.5pt](-3,1)(2,2)
\end{pspicture}

The previous example might be easier to understand if we include more detail in the sketch. Thus

\begin{pspicture}(-3,2)(2,2)
\psline[linecolor=blue,linewidth=1.5pt](-3,1)(2,2)
\newpsobject{showgrid}{psgrid}{gridlabels=0pt,griddots=0,gridwidth=0.5pt,gridcolor=gray,subgriddiv=0,subgridwidth=0.25pt,subgridcolor=red}
\showgrid
\end{pspicture}

(b) Basic Graphics Objects

Here’s a curve. Notice that the points used can be turned on (as shown) or off.

\begin{pspicture}(-6,-6)(6,6)
\pscurve[linecolor=red,linewidth=1.5pt,showpoints=true](-6,1)(0,2)(3,1)(6,6)
\end{pspicture}

where the values $\xmin, \ymin,$ etc. have been defined previously as

$\def\xmin{-6}\def\xmax(6)$
$\def\ymin(-6)\def\ymax(6)$
We begin by setting the default unit(s) in PSTricks using the command \texttt{\textbackslash psset{unit=1cm}}. This is actually the default value.

\begin{pspicture}(\xmin,\ymin)(\xmax,\ymax)
\showgrid
\pscurve[linecolor=red,linewidth=1.5pt,showpoints=true](\xmin,1)(0,2)(3,1)(\xmax,\ymax)
\psbezier[linecolor=green]{-}(-4,1)(-2,3)(1,-4)(5,5)
%\end{pspicture}

There are built-in shapes.

\begin{pspicture}(\xmin,\ymin)(\xmax,\ymax)
%\begin{pspicture}(\xmin,\ymin)(\xmax,\ymax)
\showgrid
\psellipse[linecolor=blue,linewidth=1.5pt](1,0)(1,1.5)
\psdots[linecolor=red,linewidth=1.25pt](0,0)
\SpecialCoor
\uput{6pt}[180](0,0){$(0,0)$}
\NormalCoor
%\end{pspicture}

Here is a circle centered at $(-2,-1)$ of radius 2.

\begin{pspicture}(\xmin,\ymin)(\xmax,\ymax)
\showgrid
\pscircle[linecolor=red,linewidth=1.5pt](-2,-1){2}
\end{pspicture}

Here is the same object filled-in and clipped.

\begin{pspicture}(\xmin,\ymin)(\xmax,\ymax)
%\begin{pspicture}(\xmin,\ymin)(\xmax,\ymax)
\showgrid
\psclip{\myframe(\xmin,\ymin)(\xmax,\ymax)}
\pscircle*[linecolor=red,linewidth=1.5pt](-2,-1){2}
\pswedge*[linecolor=white](-2,-1){1}{15}{105}
\endpsclip
%\end{pspicture}

\begin{pspicture}(\xmin,\ymin)(\xmax,\ymax)
%\begin{pspicture}(\xmin,\ymin)(\xmax,\ymax)
\showgrid
\psclip{\myframe(\xmin,\ymin)(\xmax,\ymax)}
\pscircle*[linecolor=red,linewidth=1.5pt]\myframe{(-2,-1)}{1}{15}{105}
\endpsclip
%\end{pspicture}
Finally, we plot some functions. To do this we’ll use some custom macros that give the user better control over the coordinate system. Thus

\begin{pspicture}(\xmin,\ymin)(\xmax,\ymax)
  \showgrid
  \myaxes{<->}(0,0)(\xmin,\ymin)(\xmax,\ymax)
  \xTickMarks{\xmin}{\xmax}{1}
  \yTickMarks{\ymin}{\ymax}{5}
  \psclip{\myframe(\xmin,\ymin)(\xmax,\ymax)}
    \psplot[style=myPlotStyle]{\xmin}{\xmax}{x dup mul} %% x^2
  \endpsclip
  \rput[lrt]{!}{\small $y=x^2$}
\end{pspicture}

And again, using custom grid controls.
Here's something useful for integration theory

Here is the code

```latex
\def\xmin{0}\def\xmax{6} \\
\def\ymin{0}\def\ymax{1} \\
\def\dommin{\xmin}\def\dommax{\xmax} \\
\VR{3in}{2.5in} \\
\newpsobject{newgrid}{psgrid}{% \\
  gridlabels=0pt% \\
  ,griddots=0pt% \\
  ,gridwidth=0.5pt% \\
  ,gridcolor=gray% \\
  ,subgriddiv=4% \\
  ,subgridwidth=0.25pt% \\
  ,subgridcolor=red% \\
} \\
\begin{pspicture}(\xmin,\ymin)(\xmax,\ymax) \\
\newgrid \\
%%%%%%%%%%%%%%%%%%%% \\
%% Axes and Ticks %% \\
%%%%%%%%%%%%%%%%%%%% \\
\SpecialCoor \\
%% Labels go here \\
\rput[lr]{\vector(1,1)}{y=\frac{1}{x}} \\
\NormalCoor \\
\myaxes{<->}(0,0)(\xmin,\ymin)(\xmax,\ymax) \\
\xTickMarks{\xmin}{\xmax}{1} \\
\yTickMarks{\ymin}{\ymax}{1} \\
\psclip{\psframe(\xmin,\ymin)(\xmax,\ymax)} \\
%% Graphing directives go here, e.g., \\
\psplot[style=myPlotStyle]{0.1}{\xmax}{1 x div} \\
\endpsclip \\
\SpecialCoor \\
\psline[fillstyle=crosshatch]{-}(3,0)(3.5,0)(!3.5 1 3 \div)(!3 1 3 \div)(3,0) \\
\psline[fillstyle=crosshatch]{-}(3.5,0)(4,0)(!4 1 3.5 \div)(!3.5 1 3.5 \div)(3.5,0) \\
\psline[fillstyle=crosshatch]{-}(4,0)(4.5,0)(!4.5 1 4 \div)(!4 1 4 \div)(4,0) \\
\psline[fillstyle=crosshatch]{-}(4.5,0)(5,0)(!5 1 4.5 \div)(!4.5 1 4.5 \div)(5,0) \\
\NormalCoor 
```

(c) Plotting Data from a File

Suppose that you wish to plot the following data.

\[
\begin{array}{cc}
0 & 0 \\
0.0628 & 0.06279 \\
0.1256 & 0.12533 \\
\vdots & \vdots \\
0.1256 & 0.12533 \\
\end{array}
\]

The following code does the trick.

\begin{verbatim}
\begin{pspicture}((xmin, ymin)(xmax, ymax)
  \showgrid
  \myaxes{<->}(0,0)(xmax,ymax)

  \psclip{\myframe(xmin, ymin)(xmax, ymax)}
  \fileplot{plotData.txt}
  \endpsclip

  \SpecialCoor

  \rput[lt](!xmax xmin sub 15 div ymax ymax sub 25 div sub){$y=\sin x$}

  \NormalCoor

\end{pspicture}
\end{verbatim}
4. Several examples from geometry.

5. A few exotic tricks.
   
   (a) A vector field.
(b) A three dimensional wedge.

(c) An ice-cream cone.
(d) A level surface.

\[ f(x, y, z) = c \]