9 The Figure Environment

Figures are really problematical to present in a document because they never split between pages. These leads to bad page breaks which leave blank space at the bottom of pages. For the fine-tuning of that document, typesetter has to adjust the page breaks manually.

But LaTeX provides floating figures which automatically move to suitable locations. So the positioning of figures is the duty of LaTeX.

9.1 Creating Floating Figures

Floating figures are created by putting commands in a `figure` environment. The contents of the figure environment always remains in one chunk, floating to produce good page breaks. The following commands put the graphic from `figure.eps` inside a floating figure

```
\begin{figure}
\centering
\includegraphics{figure.eps}
\caption{This is an inserted EPS graphic}
\label{fig1}
\end{figure}
```

9.1.1 Features

- The optional `\label` command, can be used with the `\ref`, and `\pageref` commands to reference the caption. The `\label` command must be placed immediately *after* the `\caption`
- If the figure environment contains no `\caption` commands, it produces an unnumbered floating figure.
- If the figure environment contains multiple `\caption` commands, it produces multiple figures which float together. This is useful in constructing side-by-side graphics or complex arrangements.

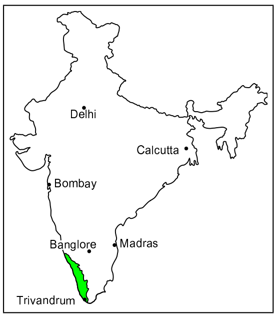


Figure 1. This is an inserted EPS graphic

- A list of figures is generated by the `\listoffigures` command.
- By default, the caption text is used as the caption and also in the list of figures. The caption has an optional argument which specifies the list-of-figure entry. For example,

  ```
  \caption[List Text]{Caption Text}
  ```

  causes “Caption Text” to appear in the caption, but “List Text” to appear in the list of figures. This is useful when using long, descriptive captions.
- The figure environment can only be used in *outer paragraph mode*, preventing it from being used inside any box (such as parbox or minipage).
- Figure environments inside the paragraphs are not processed until the end of the paragraph. For example:

```
............ text text text text text text
\begin{figure}
.........
\end{figure}
............ text text text text text text
```

9.2 Figure Placement

The `figure` environment has an optional argument which allows users to specify possible figure locations. The optional argument can contain any combination of the letters: `h`, `t`, `b`, `p`.

h Place the figure in the text where the figure command is located. This option cannot be executed if there is not enough room remaining on the page.

t Place the figure at the top of the page.

b Place the figure at the bottom of a page.

p Place the figure on a page containing only floats.

If no optional arguments are given, the placement options default to `[tbp]`.

When we input a float, LaTeX will read that float and hold it until it can place that at a better location. Unprocessed floats are those which are read by LaTeX but not yet placed on the page. Though the float-placing is done by LaTeX, sometimes user has to do something to process unprocessed floats. Following commands will do that job:

`\clearpage` This command places unprocessed floats and starts a new page.

`\FloatBarrier` This command causes all unprocessed floats to be processed. This is provided by the `placeins` package. It does not start a new page, unlike `\clearpage`.

Since it is often desirable to keep floats in the section in which they were issued, the `section` option

```
\usepackage[section]{placeins}
```

redefines the `\section` command, inserting a `\FloatBarrier` command before each section. Note that this option is very strict. This option does not allow a float from the old section to appear at the bottom of the page, since that is after the start of a new section.

The `below` option

```
\usepackage[below]{placeins}
```

is a less-restrictive version of the `section` option. It allows floats to be placed after the beginning of a new section, provided that some of the old section appears on the page.

`\afterpage/\clearpage`	The `afterpage` package provides the `\afterpage` command which executes a command at the next naturally-ocurring page break.

Therefore, using `\afterpage{\clearpage}` causes all unprocessed floats to be cleared at the next page break. `\afterpage{\clearpage}` is especially useful when producing small floatpage figures.

9.3 Customizing Float Placement

The following style parameters are used by LaTeX to prevent awkward-looking pages which contain too many floats or badly-placed floats.

9.3.1 Float Placement Counters

`\topnumber`	The maximum number of floats allowed at the top of a text page (the default is 2)
`\bottomnumber`	The maximum number of floats allowed at the bottom of a text page (the default is 1)
`\totalnumber`	The maximum number of floats allowed on any one text page (the default is 3)

These counters prevent LaTeX from placing too many floats on a text page. These counters do not affect float pages. Specifying a `!` in the float placement options causes LaTeX to ignore these parameters. The values of these counters are set with the `\setcounter` command. For example,

```
\setcounter{totalnumber}{2}
```

prevents more than two floats from being placed on any text page.

9.3.2 Figure Fractions

The commands in the below table control what fraction of a page can be covered by floats (where “fraction” refers to the height of the floats divided by `\textheight`). The first three commands pertain only to text pages, while the last command pertains only to float pages. Specifying a `!` in the float placement options causes LaTeX to ignore the first three parameters, but `\floatpagefraction` is always used. The value of these fractions are set by `\renewcommand`. For example,

```
\renewcommand{\textfraction}{0.3}
```

`\textfraction`	The minimum fraction of a text page which must be occupied by text. The default is 0.2, which prevents floats from covering more than 80% of a text page.

`\topfraction`	The maximum fraction of a text page which can be occupied by floats at the top of the page. The default is 0.7, which prevents any float whose height is greater than 70% of `\textheight` from being placed at the top of a page.
`\bottomfraction`	The maximum fraction of a text page which can be occupied by floats at the bottom of the page. The default is 0.3, which prevents any float whose height is greater than 40% of `\textheight` from being placed at the bottom of a text page.
`\floatpagefraction`	The minimum fraction of a float page that must be occupied by floats. Thus the fraction of blank space on a float page cannot be more than `1-\floatpagefraction`. The default is 0.5.

9.4 Using Graphics in LaTeX

This section shows the methods to use graphics in LaTeX documents. While LaTeX can import virtually any graphics format, Encapsulated PostScript (EPS) is the easiest graphics format to import into LaTeX. The ‘eps’ files are inserted into the file using command `\includegraphics`{*file.eps*}

9.4.1 The \includegraphics Command

`\includegraphics`[*options*]{*filename*}

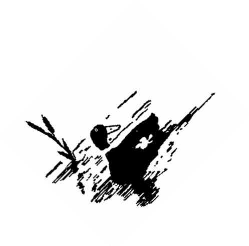
Following are the options available in `\includegraphics` command:

width	The width of the graphics (in any of the accepted TeX units).
height	The height of the graphics (in any of the accepted TeX units).
totalheight	The totalheight of the graphics (in any of the accepted TeX units).
scale	Scale factor for the graphic. Specifying scale=2 makes the graphic twice as large as its natural size.
angle	Specifies the angle of rotation, in degrees, with a counter-clockwise (anti-clockwise) rotation being positive.

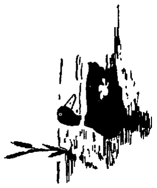
```
\includegraphics[width=.5\textwidth]{filename}
\includegraphics[height=2in]{filename}
\includegraphics[totalheight=2in]{filename}
\includegraphics[scale=2]{filename}
```

\includegraphics[width=1in]{duck}

\includegraphics[height=1.5in]{duck}



\includegraphics[scale=.25,angle=45]{duck}



\includegraphics[scale=.25,angle=90]{duck}

9.4.2 Graphics Search Path

By default, LaTeX looks for graphics files in any directory on the TeX search path. In addition to these directories, LaTeX also looks in any directories specified in the `\graphicspath` command. For example,

```
\graphicspath{{dir1/}{dir2/}}
```

tells LaTeX to also look for graphics files in `dir1/` and `dir2/`. For Macintosh, this becomes

```
\graphicspath{{dir1:}{dir2:}}
```

9.4.3 Graphics Extensions

The `\DeclareGraphicsExtensions` command tells LaTeX which extensions to try if a file with no extension is specified in the `\includegraphics` command. For convenience, a default set of extensions is pre-defined depending on which graphics driver is selected. For example if `dvips` is used, the following graphic extensions (defined in `dvips.def`) are used by default

```
\DeclareGraphicsExtensions{.eps,.ps,.eps.gz,.ps.gz,.eps.Z}
```

With the above graphics extensions specified, `\includegraphics{`*file*`}` first looks for `file.eps`, then `file.ps`, then file `file.eps.gz`, etc. until a file is found. This allows the graphics to be specified with

```
\includegraphics{file}
```

instead of

```
\includegraphics{file.eps}
```

9.5 Rotating and Scaling Objects

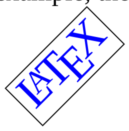
In addition to the `\includegraphics` command, the `graphicx` package includes 4 other commands which rotates and scale any LATEX object: text, EPS graphic, etc.

```
\scalebox{2}{\includegraphics{file.eps}}
\resizebox{4in}{!}{\includegraphics{file.eps}}
\rotatebox{45}{\includegraphics{file.eps}}
```

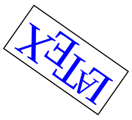
produces the same three graphics as

```
\includegraphics[scale=2]{file.eps}
\includegraphics[width=4in]{file.eps}
\includegraphics[angle=45]{file.eps}
```

For example, the following is produced with



```
\rotatebox{45}{\fbox{\Large
  \textcolor{blue}{\LaTeX}}}
```



```
\rotatebox{180}{\fbox{\Large
  \textcolor{blue}{\LaTeX}}}
```

However, the `\includegraphics` is preferred because it is faster and produces more efficient PostScript.