

Including Graphics

Petra Harwin

November 25, 2012

Contents

1	Including graphics	1
1.1	Basic syntax	2
1.2	A simple example	2
1.3	Scaling and Rotating	3
1.3.1	height versus totalheight	4
1.4	Figures as floats	4
2	The subfigure Package	5
2.1	An example	5
3	Creating your own .eps files	6
3.1	The psfrag Package	6
3.1.1	Viewing your psfrag replacements	7

1 Including graphics

To include graphics we must use a package called `graphicx`. We do this by including the line

```
\usepackage[driver]{graphicx}
```

in the preamble.

I'm only going to teach you how to include encapsulated postscript (`.eps`) files so the optional argument `[driver]` may be left blank since `dvips` is the default. For details about how to include other types of image see section 6.2 in Kopka or Daly and section 10.2 in Mittlebach & Goossens.

1.1 Basic syntax

We can now include our `.eps` file using the following command

```
\includegraphics[key=value,...]{filename}.
```

The optional arguments `key=value` are of different types: those that take a numerical value and those that are flags with the values `true` or `false`. Giving the name of a flag without a value is equivalent to setting it to true. Here are some of the ones that you are most likely to use. (For a full list see page 170 of Kopka and Daly or pages 619–620 of Mittlebach & Goossens.)

`scale=` number; the amount by which the figure should be magnified relative to its natural size;

`width=` length; specifies the length to which the width of the figure should be scaled. If `height` is not given, it is scaled by the same factor as `width`;

`height=` same idea as `width`;

`keepaspectratio` true/false; this flag ensures that the original height width ratio remains unchanged even when both height and width are specified. The figure is scaled so that neither its height or width exceed the given dimensions.

`angle=` number; the angle by which the figure is to be rotated anticlockwise, in degrees.

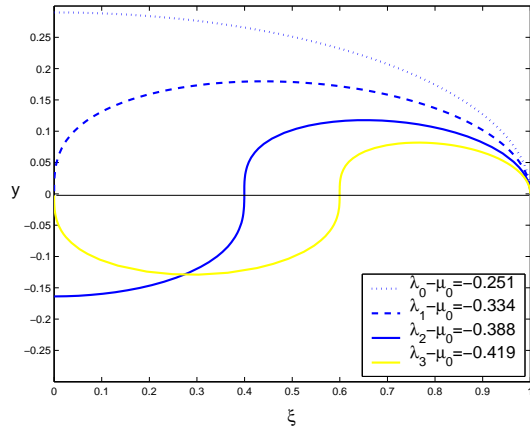
`totalheight=` length; like `height` but specifies the height (amount above the baseline) plus depth (amount below the baseline) of the figure. This should always be used in place of `height` if the figure has been rotated because this is the true height of the figure.

1.2 A simple example

Now we are in a position to include an `.eps` file. I have a file called `numerics.eps` in the same directory as my main `.tex` file. To include this file at this point in the document we type

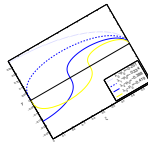
```
\includegraphics[scale=0.4]{numerics.eps}
```

This produces

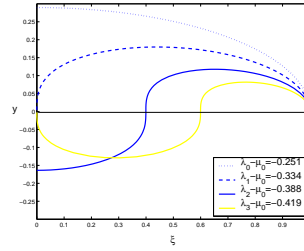


1.3 Scaling and Rotating

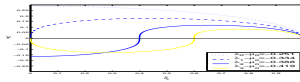
Let's use some keys.



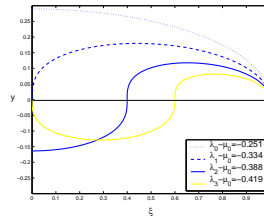
angle=30, totalheight=2cm,



width=4cm

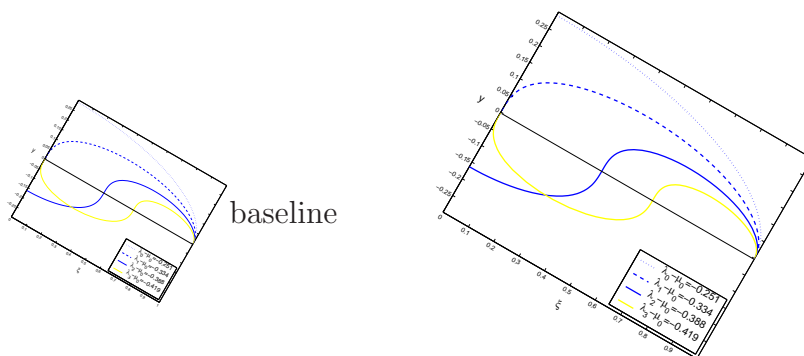


width=4cm, height=1cm



scale=0.2

1.3.1 height versus totalheight



`angle=-30, totalheight=3cm` `angle=-30, height=3cm`

Height of the figure is 3cm. Height above the baseline of the figure is 3cm.

Note that keys are read from left to right so `totalheight` should be set after the rotation.

1.4 Figures as floats

Just using `\includegraphics` is the same as using `tabular` outside a `table` environment. To make \LaTeX treat your figure as a float you need to include it in the `figure` environment. Its syntax is

```
\begin{figure}[pos]
\includegraphics[keys]{filename}
\caption{Caption text}
\label{figure label}
\end{figure}
```

`pos` is the positioning argument. It may take the following values:

h here: the float may appear at this point in the text.

t top: the float may appear at the top of the current page, provided there is enough room for both it and the previous text. If not then it is added at the top of the next page.

b bottom: the float may be placed at the bottom of the page; the subsequent text continues until the room left on the current page is just enough for the float. If there is already insufficient room then the float will be placed at the bottom of the next page.

p page: the float may be put on a special page containing only figures and/or tables.

! used together with any of the other letters will suspend any extra spacing to try to position the float where you want it.

The commands `\label{figure label}` and `\caption{Caption text}` work as they did in the table environment.

2 The subfigure Package

To use this package you must include the line `\usepackage{subfigure}` in the preamble.

The `subfigure` package can be used to split one figure environment into many subfigures, which are labelled accordingly. It can only be used inside the figure environment. Its syntax is:

```
\begin{figure}[pos]
\subfigure[caption]{
\label{label1}
\includegraphics[keys]{filename}} spacing or newlines
\subfigure[caption]{
\label{label2}
\includegraphics[keys]{filename}} spacing or newlines
...
\caption{Main Caption}
\label{mainlabel}
\end{figure}
```

`pos`, `keys` These work in the same way as for the figure environment;

`mainlabel` This is a reference for the whole figure;

`label1`, `label2` These provide labels for the subfigures;

`caption` This provides a small caption for each subfigure and may be omitted;

`Main Caption` This provides a large caption underneath all the subfigures. This may also be omitted.

2.1 An example

Figure 1 was produced using the code

```
\begin{figure}[h]
\centering
\subfigure[Caption 1]{
\label{label1}
\includegraphics[scale=0.4]{numerics.eps}}
\subfigure[Caption 2]{
```

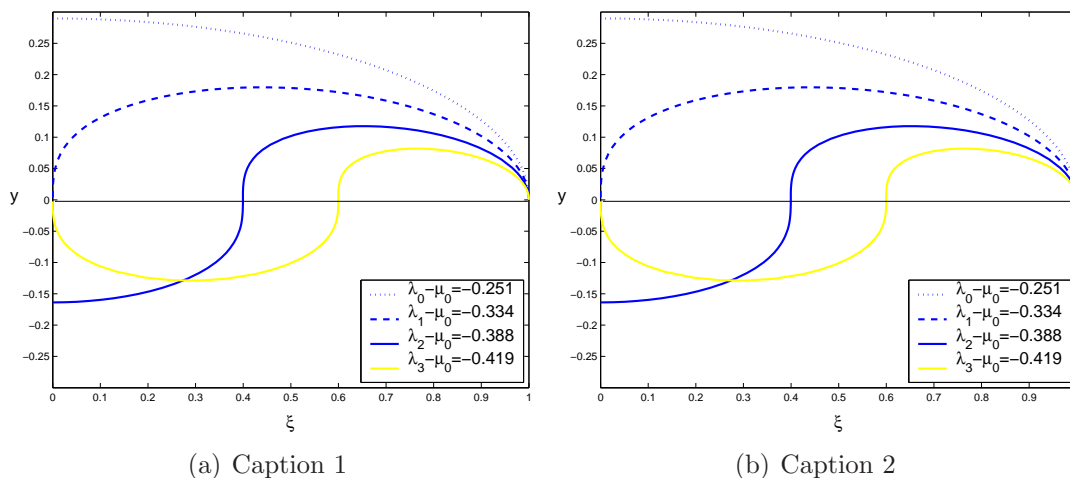


Figure 1: Main caption

```

\label{label2}
\includegraphics[scale=0.4]{numerics.eps}
\caption{Main caption}
\label{mainlabel}
\end{figure}

```

(The `\centering` command is a way of centering floats and does not generate the extra space above and below that the center environment does.)

Now we can refer to Figure 1(a) (Figure `\ref{label1}`) or Figure 1 (Figure `\ref{mainlabel}`) etc.

3 Creating your own .eps files

Suppose you want to draw a simple diagram rather than using software to create graphs. It is relatively easy to do this using a package called `xfig` that is available for UNIX and Linux. It can be started at the command line by simply typing `xfig`. It's fairly self-explanatory once you've got it running. You create a `.fig` file with `xfig` and then use the `export` command to export it as an `.eps` file.

3.1 The `psfrag` Package

To use this package you must include the line `\usepackage{psfrag}` in the preamble.

Often you want to include text or mathematics inside a figure. \LaTeX offers a nice package called `psfrag` that does this for you. It works by replacing a given string within an `.eps` file. To replace text in a figure the following command should be included just after `\begin{figure}` and its syntax is

```
\psfrag{orig}{repl}
```

where `orig` is what you wish to replace and `repl` is what you want to replace it with. You can have as many `\psfrag` commands as there are things you want to replace. The only thing to note is that `orig` must be in the default L^AT_EX font.

You may also use `psfrag` with the `subfigure` package. It should be used in the following way

```
\begin{figure}[pos]
\subfigure[caption]{
\label{label1}
\psfrag{orig1}{repl1}
\includegraphics[keys]{filename}} spacing or newlines
\subfigure[caption]{
\label{label2}
\psfrag{orig2}{repl2}
\includegraphics[keys]{filename}} spacing or newlines
...
\caption{Main Caption}
\label{mainlabel}
\end{figure}
```

3.1.1 Viewing your `psfrag` replacements

If you view the dvi file L^AT_EX creates you will only see a list of what the `psfrag` package has replaced, not the replacements themselves. This is because the replacements are made upon conversion to Postscript. Thus, in order to see the replacements you must convert your file to a Postscript (.ps) file.