Including Postscript Images in your LaTeX Document

Paul A. Fishwick
University of Florida
email: fishwick@cis.ufl.edu

July 1, 1992

Abstract

The goal of this brief document is to demonstrate how to incorporate postscript files into your LaTeX file. It is quite straightforward if you know (or find!) the rules - therein lies the problem. I learned all of these methods through some trial and error. Perhaps this document will help you from having to do the same thing. Good luck and happy TeXing!

1 Introduction

To obtain postscript figures within your document, you need to do two things: 1) run `latex mydoc`, and 2) run `dvips mydoc`. The `dvips` program has many options and provides you with a clean interface. Look at the man page for *dvips* to see a great deal of information on how to incorporate Postscript images in LaTeX. A general note on LaTeX: you can find many style files and contributed files at any site that caters to LaTeX users. Hopefully, you will not have to ftp any files, but if you do you could check out: `ymir.claremont.edu [134.173.4.23]` for instance. There are many other sites as well.

2 Some Important LaTeX Lines

The following is header information that you must include in your document. First, make sure to use 'epsf' to reference the Encapsulated Postscript Style File:

```
\documentstyle[12pt,epsf]{article}
%macro for Postscript figures the easy way
\newcommand{\postscript}[2]
 {\setlength{\epsfxsize}{#2\hsize}
  \centerline{\epsfbox{#1}}}
```

3 UNIX Script Files

There are two UNIX script files which are used to do some necessary post-processing on the postscript output produced by some programs. They are specified below:

XFIGFIX

```
#!/bin/sh
#Normalize a postscript file to a file accessible to dvips.
while [ $# -gt 0 ]; do
  ps=`echo $1 | sed "s/\.fig$/.ps/"`
  #echo postscript output is $ps
  echo "%begin(plot)" > $ps
  /local/X11/f2ps -c -N -P $1 | \
          sed -e "s#^%begin(plot)#%%%begin(plot)#g" \
              -e "s#^%end(plot)#%%%end(plot)#g"     \
              -e "s#^showpage#%showpage#g" >> $ps
  echo "%end(plot)" >> $ps
  shift
done
```

IMAGEFIX

```
#!/bin/sh
#Normalize a postscript file to a file accessible to dvips.
while [ $# -gt 0 ]; do
  ps="temp9999.ps"
  echo "%begin(plot)" > $ps
  cat $1 | \
         sed -e "s#^%begin(plot)#%%%begin(plot)#g" \
             -e "s#^%end(plot)#%%%end(plot)#g"     \
             -e "s#^showpage#%showpage#g" >> $ps
  echo "%end(plot)" >> $ps
  shift
done
cat $ps
rm -rf $ps
```

XVGR2EPS

```
#! /usr/bin/csh -f
# convert the XVGR postscript file into encapsulated postscript
# usage
#       xvgr2eps xvgr-file.ps
mv $1 tmp$$.ps
unset noclobber
```

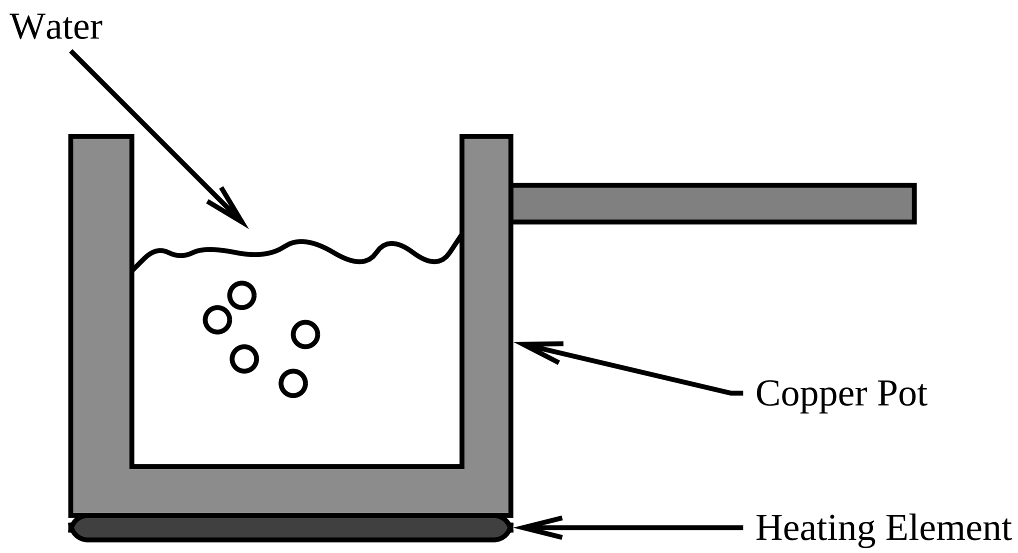


Figure 1: A big pot of boiling water.

```
echo '%\!PS-Adobe-3.0 EPSF-3.0' >! $1:r.ps
echo '%%BoundingBox: 40 50 550 700' >> $1:r.ps
sed '/%%Creator/,/%%Trailer/\!d' tmp$$.ps >> $1:r.ps
/bin/rm -f tmp$$.ps
```

4 Xfig Postscript Files

The first three figures are were created in *xfig* in the X-Window system. The *xfig* data file is saved as "somefile.fig" where *somefile* can be any name you choose. Then you need to do one of two things depending on what programs you have available):

1. `fig2dev -L ps somefile.fig > somefile.ps`

2. `xfigfix somefile.fig`

The use of 'fig2dev' is preferred (if you have that program) since it has many options including a choice of the target device language. This will produce a postscript file called "somefile.ps" which you insert into your LaTeX file as demonstrated in this document. Note how you can change the size of the postscript file using the second argument to `\postscript`. Figure 1 is a figure representing a pot of boiling water at 75% of full size using:

```
\begin{figure}
\postscript{water.ps}{0.75}
\caption{A big pot of boiling water.}
\label{water1}
\end{figure}
```

Figure 2 is a figure representing a pot of boiling water scaled down 50% of the orginal size using:

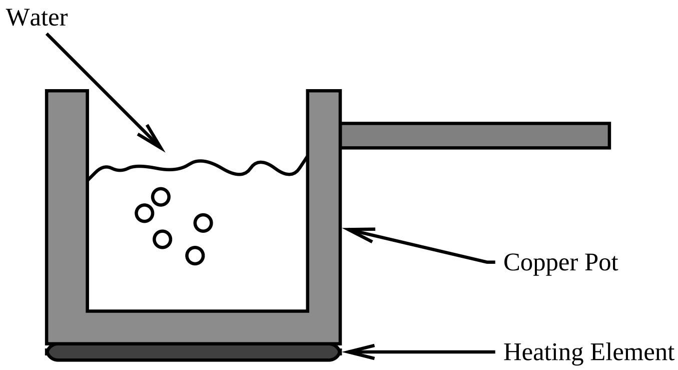


Figure 2: A medium pot of boiling water.

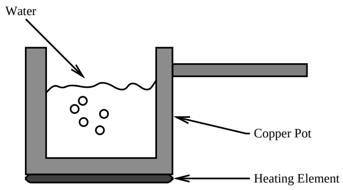


Figure 3: A tiny pot of boiling water.

```
\begin{figure}
\postscript{water.ps}{0.50}
\caption{A medium pot of boiling water.}
\label{water2}
\end{figure}
```

Figure 3 is a figure representing a pot of boiling water at 25% of full size using:

```
\begin{figure}
\postscript{water.ps}{0.25}
\caption{A tiny pot of boiling water.}
\label{water3}
\end{figure}
```

5 GnuPlot Postscript Files

Figure 4 is a graph of boomerang trajectories at 75% of full size. It was created using *gnuplot* using the option of outputting a postscript file. The postscript file is postprocessed by doing "imagefix graph.ps > gnuplot.ps" where "graph.ps" is the postscript file obtained from *gnuplot* and "gnuplot.ps" is the new file to be referenced in your document. Here is the figure environment:

```
\begin{figure}
\postscript{gnuplot.ps}{0.75}
\caption{Four boomerang trajectories.}
\label{graph}
```

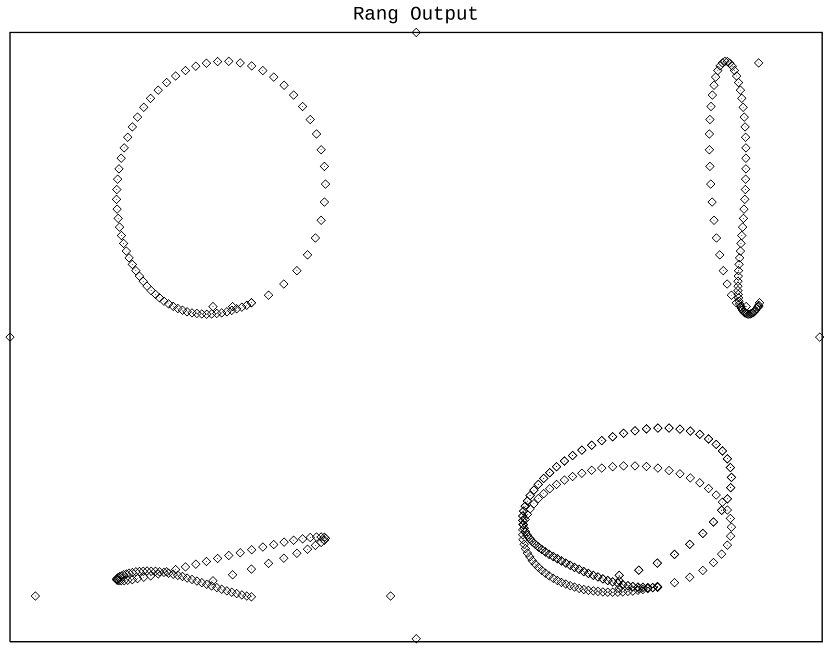


Figure 4: Four boomerang trajectories.

```
\end{figure}
```

6 xvgr Postscript files

Figure 5 is a graph at 40% of full size. It was created using Paul Turner's *xvgr* plotting package using the option of outputting a postscript file. The postscript file is postprocessed by doing "xvgr2eps graph.ps" where "graph.ps" is the postscript file obtained from *xvgr* Here is the figure environment:

```
\begin{figure}
\postscript{graph.ps}{0.40}
\caption{Sample graph from xvgr.}
\label{graph2}
\end{figure}
```

7 Utah Raster Postscript Files

The picture below (fig. 6) is a postscript file obtained by running the Utah Raster Toolkit tool called *rletops* which takes a run length encoded image file (rle-file) and converts it to Postscript. Note that if you use *rletops*, you must use the "-s" (scribe) option so that a "showpage" is not generated. Also, you may want to use the "-h" option to scale the image. If the rle-file is "rlefile"

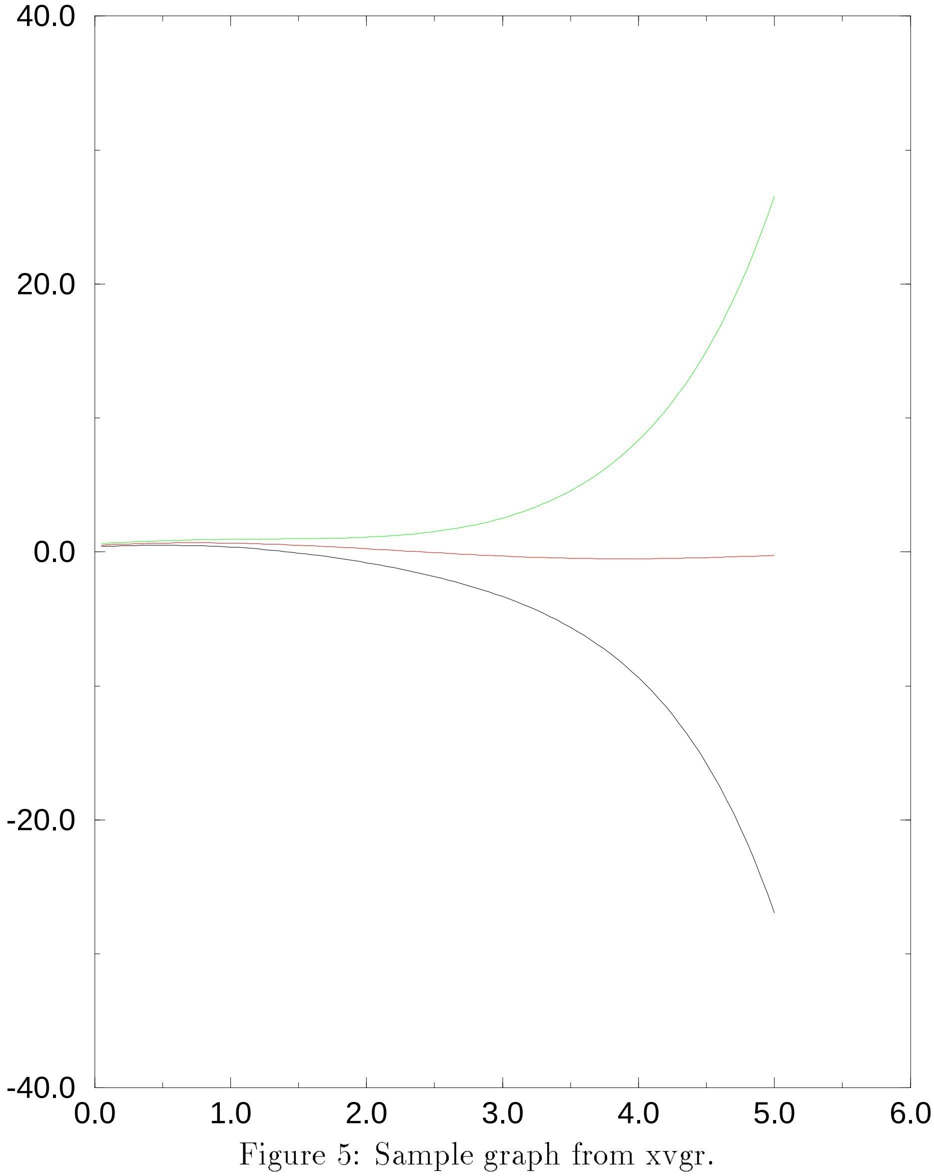


Figure 5: Sample graph from xvgr.

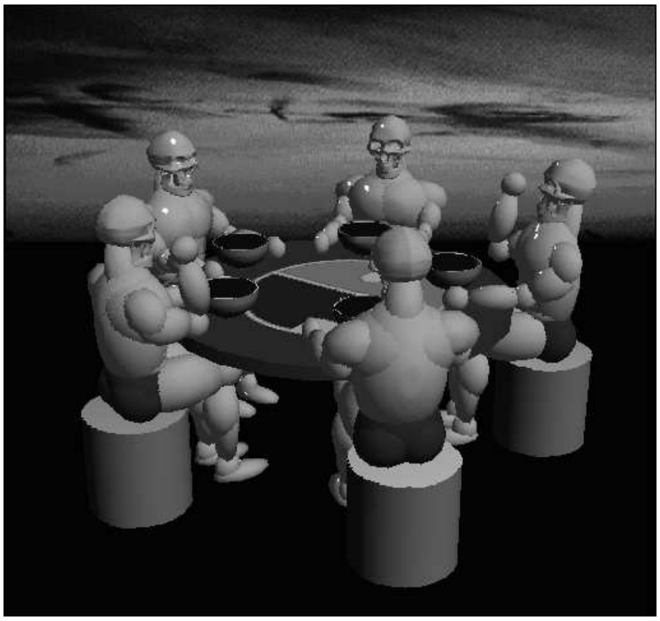


Figure 6: Five dining terminators.

then we do the following: "rletops -s rlefile > file.ps". Also, you will have to estimage the amount of space and use the vspace latex command. Here is the figure environment:

```
\begin{figure}
\vspace{3in}
\special{psfile=file.ps}
\caption{Five dining terminators.}
\label{teapot}
\end{figure}
```

The problem with the tool 'rletops' is that while it produces Postscript, it does not produce encapsulated Postscript which is what you would ideally like for LaTeXdocuments. Another problem is that you have to be careful when doing two column document types such as 'proc' (proceedings) - you must use the center option on *rletops* to manually place the figure the right number of inches from the left margin. The default is 3.25 inches which is perfect for single column documents.

A way around this problem is to convert the file that you get from *rletops -s* into encapsulated postscript. At the beginning of the *rletops* produced file, you will have something such as the following:

```
%!
%BoundingBox: 119 0 348 216
%EndComments
```

This, at least, gives you the bounding box information. Take the bounding box coordinates and replace these 3 lines with:

```
%begin(plot)
%!PS-Adobe-2.0 EPSF-1.2
%%BoundingBox: 119 0 348 216
%%EndComments
```

Also, append the following to the end of your .ps file:

```
%end(plot)
```

Then, inside your document, you just use the standard method for embedding encapsulated postscript files:

```
\begin{figure}
\postscript{tea.ps}{0.8}
\caption{This is a small tea pot.}
\label{teapot}
\end{figure}
```

8 Xgrabsc Postscript Files

When in X-Windows you can "grab" any part of the screen using the utility *xgrabsc*. Make sure to use the "-E" encapsulated postscript option as follows: `xgrabsc -E -o output.ps`. Figure 7 provides an example screen capture in X. After grabbing the file, use "imagefix" as shown previously. The figure environment is shown below.

```
\begin{figure}
\postscript{calvin.ps}{0.50}
\caption{Calvin \& Hobbs.}
\label{calvin}
\end{figure}
```

9 Putting Figures Side by Side

To put figures side by side, you must use the `subfigure.sty` style file in your `documentstyle` command as follows:

```
\documentstyle[12pt,epsf,subfigure]{article}
```

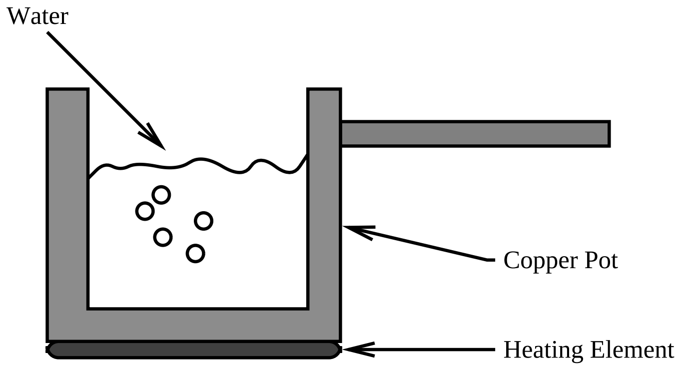
Now, we will use this convenient style to specify a figure containing 5 sub-figures each with its own caption. Here is the code to specify such a figure:

```
\begin{figure}[h]
\begin{center}
\begin{tabular}{c}
\setlength{\epsfxsize}{0.5\hsize}
```

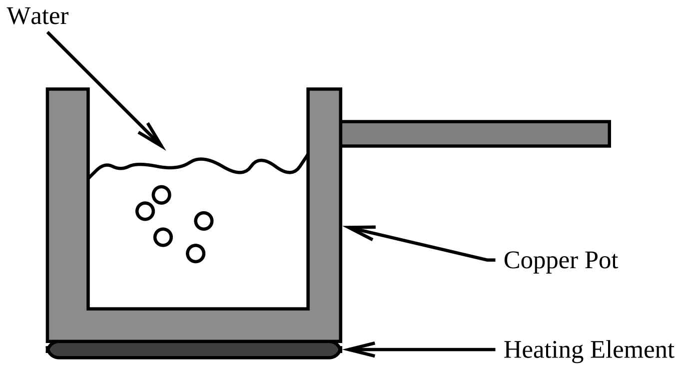
Figure 7: Calvin & Hobbs.

```
\subfigure[First Pot]{\epsfbox{water.ps}}
\hspace{0.5in}
\subfigure[Second Pot]{\epsfbox{water.ps}}\\
\setlength{\epsfxsize}{0.3\hsize}
\subfigure[Third Pot]{\epsfbox{water.ps}}
\subfigure[Fourth Pot]{\epsfbox{water.ps}}
\subfigure[Fifth Pot]{\epsfbox{water.ps}}
\end{tabular}
\end{center}
\caption{Five Subfigures.}
\end{figure}
```

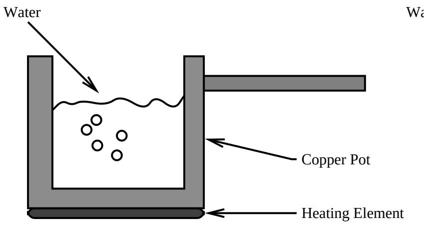
The figure with sub-figures is shown in figure 8. Note how we have optionally included an `hspace` command when we wanted to increase the horizontal spacing between the figures. Other optional vertical spacing commands can be seen at the head of the style file `subfigures.sty`. It is not necessary to include the `[FIGURECaption]` argument for `subfigure`; the default is just to have the small letters (a), (b), (c), etc.



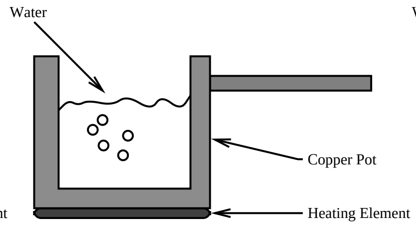
(a) First Pot



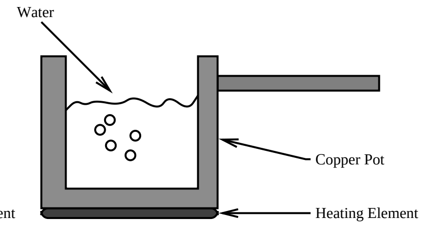
(b) Second Pot



(c) Third Pot



(d) Fourth Pot



(e) Fifth Pot

Figure 8: Five Subfigures.