

Writing a L^AT_EX Article – Embedding Figures

T. Stitz *

Abstract

Module 5 shows how to include external graphics in a L^AT_EX document. It discusses several packages that might be used to gain more control the floating behavior if needed and how the figures appear in the document.

1 Packages for Graphics

One of the following packages is needed for graphics: *graphics* or *graphicx*. The graphics driver selected will control how the graphic appears and is printed. For more information on graphics drivers, refer to [Kopka and Daly \(2004\)](#) and [Mittelbach, Goossens, Braams, Carlisle, and Rowley \(2004\)](#). We will use the *graphicx* package in this module.

2 File Format Compatible with L^AT_EX

When producing a DVI file, a postscript graphics file like encapsulated postscript (EPS) is used. When using the *pdflatex* command, several graphics formats can be used: PNG, JPG, and PDF; however, an EPS file cannot be used. Some distributions, such as T_EX Live in the Windows environment or MacT_EX, will automatically convert an EPS to a PDF file. If it does not automatically convert, an error message will appear.

Most books or other documentation suggest that you don't type the extension of the file when using the *includegraphics* command, which is used to display the image. Let's say that you try to use a file with the EPS format when you use the *pdflatex* command. If you do not use the extension in the filename, the error will be “! LaTeX Error: File '*filename*' not found.”

If you use the extension in the filename, the error will be “! LaTeX Error: Unknown graphics extension: .eps.” There are some packages, such as *epstopdf*, that can be used to convert EPS to PDF on the fly when the conversion is not automatic. Figures can be converted with several external graphics software packages as well.

3 Displaying Graphics

In module 4, the *table* environment and the *caption* package were discussed. Essentially, the *figure* environment has the same relationship with figures as the *table* environment has with tables. Figures are floated to optimize space, but you can use the same optional arguments for placement with or without the exclamation point: h, t, b, or p. The captions are automatically numbered and the caption package can be used to format the caption. For more information about the optional arguments for figure or the caption package, refer to module 4.

As stated in the previous section, the command to display the image is the *includegraphics* command. There are many optional arguments that you can use, but we will only use the width option in this module. Assuming the use of the *graphicx* package, a figure is included in a L^AT_EX document with a format similar to the following:

* Applied Sciences Librarian, Phone: 330-972-6192, Fax: 330-972-7033, E-mail: tstitz@uakron.edu

```

\begin{figure}[optional argument]
  \centering
  \includegraphics[optional arguments]{filename}
  \caption{caption text}
\end{figure}

```

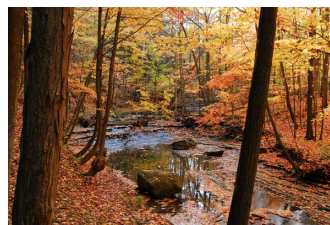


Figure 1: Bedford Reservation

There is special version of the *figure* environment, *figure**, that behaves exactly like *table**. For more information about floats, refer to Chapter 9 of [Kopka and Daly \(2004\)](#) and Chapter 6 of [Mittelbach et al. \(2004\)](#).

4 Commands and Packages used to Change Graphics Placement

\LaTeX abides by a set of rules to determine placement. Sometimes, the author would like figure placement elsewhere. There are some commands and packages that can be used to override typical behavior. Sometimes page formatting commands, such as *enlargethispage*, or space commands, such as *vspace*, can be used to change the locations of the graphic. [Additional formatting commands can be found in *Guide to \$\LaTeX\$*](#) . As a last resort, the code that displays the figure can be placed elsewhere in the text to achieve this goal.

There are commands that force floats to print, such as *\clearpage* when using the default option *oneside* or *\cleardoublepage* when using the *twopage* option for *article.cls*. Using these commands could force unwanted white space; therefore, most literature suggests using the *afterpage* package and the command, *\afterpage{clearpage}*. The *afterpage* package is not compatible with all tables and figure environments. For example, this command is not compatible with *longtable*.

The *placeins* package can be used to ensure that a float does not deviate past some boundary in your document. To force all floats to be printed in the section that they are declared, there is an optional argument, *section*, that can be used with *placeins*. If there is another barrier, it is more complicated. Let's say that we are interested in forcing all floats to print in the subsection they are declared. We don't want to change any other formatting...only add a float barrier. Since we are using *article.cls*, we find the definition for subsection in *article.cls* and copy the code. In the preamble of our document, first, we use the *makeatletter* command. Next, we use the code that we copied from *article.cls* with *\FloatBarrier* added to the last required argument. Lastly, we use the *makeatother* command. The code excerpt is

```

\makeatletter
\renewcommand\subsection{\@startsection{subsection}{2}{\z@}%
  {-3.25ex\@plus -1ex \@minus -.2ex}%
  {1.5ex \@plus .2ex}%
  {\normalfont\large\bfseries\FloatBarrier}}
\makeatother

```

The *FloatBarrier* command can be used within the text of the document to force floats to print as well.

5 Creating your Own Floating Environment

With \LaTeX , the floating items available are figures and tables. Using the *float* package, allows you to configure other floating environments. We are going to create a floating environment for algorithms. There are several packages that can be used to display computer programs in their entirety or display particular sections of the code. The *listings* package gives you much control of the format of your code excerpts. In this module, we will use verbatim. The verbatim environment will print the text exactly as it is typed. There are three commands that

are used to format and declare your floating environment. For it to be true for the entire document, they would appear in the preamble.

```
\floatstyle{plaintop}
\floatname{algorithm}{\bfseries Algorithm}
\newfloat{algorithm}{htb}{alg}
```

In this example, we used the `plaintop` format style. Essentially, it places the label and caption above the floating environment and it does not display any lines. It just prints the label and caption. For other formatting styles, [refer to the documentation for the `float` package on ctan.org](#). The `floatname` sets the label for the floating environment. The `newfloat` command is the declaration of your environment. The first argument is the name of the environment. The second argument lists placements, so it can be here (h), top (t), or bottom (b) in our example. The last argument will only be used when creating a table of contents. For our example, the extension of the file containing information needed to create a “list of algorithms” is `.alg`. To use this new environment, the structure is very similar to tables and figures. The code is

```
\begin{algorithm}
\captionsetup{singlelinecheck=false}
\caption{Simple for loop}
\begin{verbatim}
for(k=0;k<1000;k++)
{
  y(k)=2*k+1;
}
\end{verbatim}
\end{algorithm}
```

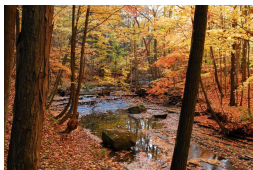
The output is

Algorithm 1: Simple for loop

```
for(k=0;k<1000;k++)
{
  y(k)=2*k+1;
}
```

Other features of the `float` package can be found in the package file documentation.

6 Wrapping Text around Figures



When using the `figure` environment, the figure resides on its own line vertically. It will print without any text next to it. If you want to place a picture with regular text that flows around it, you could use packages like `wrapfig` or `parpic`.

When using `wrapfig` in conjunction with the `float` package, `float` must be loaded first. As in the case of `figure*`, wrapped figures could be out of sequence with regular floats. Sometimes, there is too much whitespace when using the `wrapfig` command. The `vspace` command can be used to eliminate this space. The example shown in this document was generated as follows:

```
\begin{wrapfigure}{l}{0.2\textwidth}
\vspace{-12pt}
\includegraphics[width=0.2\textwidth]{bedford}
\vspace{-20pt}
\end{wrapfigure}
```

where the placement of the image is left of the text (l) and it is 20% of the text width. The image takes the entire space reserved for the figure; thus, the optional argument for *includegraphics* command is the same as the size given in the wrapfigure environment. The *vspace* command trimmed the whitespace around the image.

When using *parpic*, pictures can be framed in addition to text wrapping. The following example would produce the same image at the beginning of the paragraph with a shadowbox:

```
\shadowthickness{3pt}
\parpic[s]{\includegraphics[width=0.2\textwidth]{bedford}}
```

7 Using Subfigures

Sometimes figures are related and it is desirable to group them together. The *subfig* package can be used for this purpose. It is very similar to the code in this module thus far. The main difference is use of the *subfloat* command. There is an option to include subcaption text. By default, the label for the captions will be (a), (b), etc. Figure 2 was generated using the following:

```
\begin{figure}
  \centering
  \subfloat[subcaption text]{\includegraphics[optional arguments]{filename}}
  \subfloat[subcaption text]{\includegraphics[optional arguments]{filename}}
  \caption{caption text for the entire figure}
\end{figure}
```

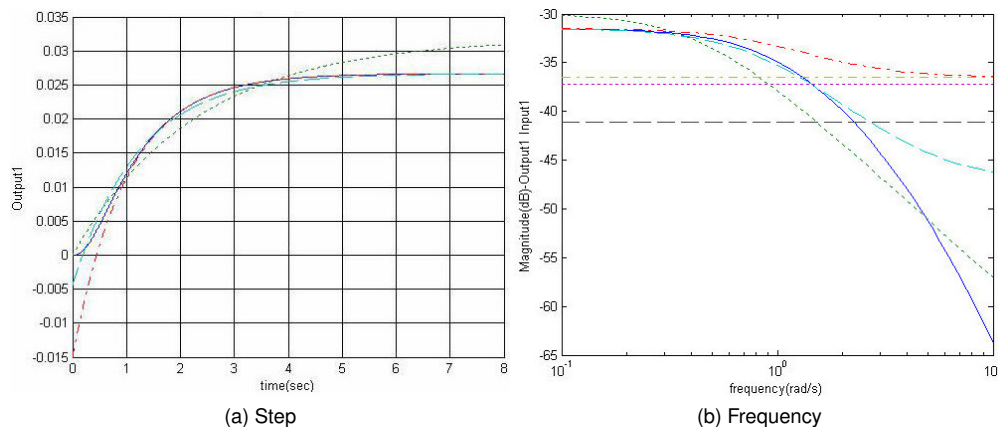


Figure 2: System Response

7.1 Changing the Captions of Subfigures

The labels for the subfigures can be changed using the *caption* package too. Just use the optional argument, *subfigure*. If we wanted a different numbering scheme than a, b, c, etc., we can renew *thesubfigure* command as we did for ordered lists in module 3. A code excerpt to change the numbering of the subfigures to 1. and 2. is

```
\captionsetup[subfigure]{labelformat=simple,justification=raggedright}
\renewcommand{\thesubfigure}{\arabic{subfigure}.}
```

8 Placing Graphics beside Tables

Sometimes you want placement that is different than regular text wrapping around an image or an image on a line by itself. For example, you can place a graphic beside a table by using the *minipage* environment. The trick is adding a point of reference to align the image and the table.

In our example, we want to align an image and a table on their top edge. We place each in a *minipage* and we use the optional argument *t* to top align; however, the alignment is not what we expect. We need to add a dummy command `\vspace{0pt}` as the first argument in each *minipage* on which to align. Each *minipage* is half of the text width. Notice the width of the graphic is `\textwidth`. From the perspective of the *includegraphics* command, the text width is the width of the *minipage*. The code excerpt is

```
\begin{figure}
\begin{minipage}[t]{.5\textwidth}
  \vspace{0pt} %This dummy command is used for proper alignment
  \includegraphics[width=\textwidth]{bedford}
\end{minipage}
\begin{minipage}[t]{.5\textwidth}
  \vspace{0pt} %This dummy command is used for proper alignment
  \begin{tabular}{lp{.8\textwidth}}
    Name: & Bridal Veil Falls \\
    Park: & Bedford Reservation \\
    Location: & Bedford, Ohio \\
    Date: & October 2008 \\
    Notes: & This picture was taken when hiking the trail by Bridal Veil Falls at Bedford Reservation.
  \end{tabular}
\end{minipage}
\caption{Bridal Veil Falls}
\end{figure}
```

The output is



Name: Bridal Veil Falls
Park: Bedford Reservation
Location: Bedford, Ohio
Date: October 2008
Notes: This picture was taken when hiking the trail by Bridal Veil Falls at Bedford Reservation.

Figure 3: Bridal Veil Falls

References

- Kopka, H., & Daly, P. W. (2004). *Guide to L^AT_EX* (4th ed.). Boston: Addison-Wesley. Retrieved from <http://proquest.safaribooksonline.com/9780321617736>
- Mittelbach, F., Goossens, M., Braams, J., Carlisle, D., & Rowley, C. (2004). *The L^AT_EX companion* (2nd ed.). Boston: Addison-Wesley. Retrieved from <http://proquest.safaribooksonline.com/0201362996>