

# Ph3 LaTeX Week 3: Adding figures

Eric D. Black

August 19, 2021

## 1 Bringing graphics into L<sup>A</sup>T<sub>E</sub>X

First, to handle figures you will need to include the *graphicx* package, so open your “hello world” project, and add the following line to its preamble in the source code.

```
\usepackage{graphicx}
```

Next, upload your figure to overleaf using the icon at the top left of your browser window, the one that looks like this.



Figure 1: The overleaf upload icon.

Use the drawing you made last week in Inkscape. If you have not already done so, save a pdf version of it, and use that. If you have installed latex on your personal computer, put the figure in the same folder as your document’s source code.

The *graphicx* package can handle most file formats, including the vector formats .pdf, .eps, and .ps, and a host of bitmapped formats like .jpeg, .gif, .tiff, .png, etc.

Now, to insert a figure into your document you must first declare a *figure* environment, and that is done with the (you guessed it) `\begin{figure}` and `\end{figure}` commands. Inside this environment, the basic command for including a graphics object is `\includegraphics{filename.extension}`, where *filename* is the name of your graphics file, and *extension* is its extension, *i.e.* pdf, jpeg, etc.

Note that if you start typing `\begin{figure}` in overleaf, you will be presented with a suggestion for completing the environment. If you accept this suggestion

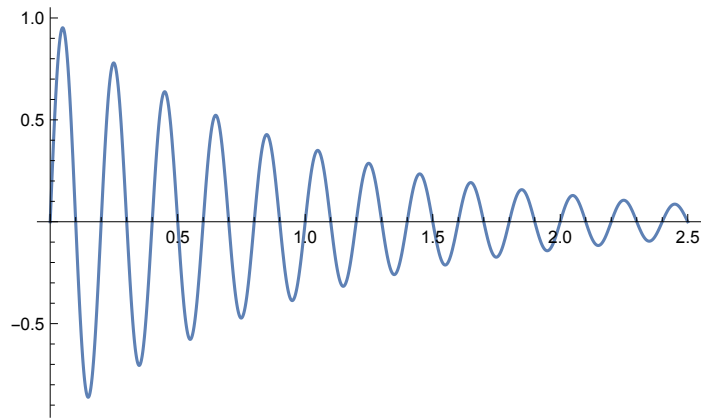


Figure 2: A figure included in a  $\text{\LaTeX}$  document.

(by hitting return before you finish typing the line) you will get a few extra commands.

```

\begin{figure}
  \centering
  \includegraphics{}
  \caption{Caption}
  \label{fig:my_label}
\end{figure}

```

The first, `\centering`, centers your figure within the *figure* environment, which you nearly always want to do. The next is `\includegraphics{}`, which we have already covered. After that comes `\caption{Caption}` which, as you can probably guess, lets you write a caption underneath your figure. The last one, `\label{fig:my_label}`, allows you to label your figure so that you can refer to it elsewhere in the text. It defines a way for you to say, for example, “see Figure 1” and automatically have that line refer to the correct figure even if you rearrange your graphics. We’ll cover this labeling feature next week.

Once you have populated the arguments to your commands your output should look something like Figure 2. Like Figure 2 it will probably be at the top or bottom of your page, not necessarily close to the text that refers to it. Latex does this by default, and this is usually ok. In fact, some journals may ask you to submit all of your figures at the end of your manuscript so that they can arrange them in the typesetting process. In this case you won’t get to choose exactly where your figure shows up in your paper, and for the most part that is ok too. If you have a professional typesetter helping you prepare your document be grateful. Let them do their job, and you focus on your content.

You can put anything in the figure environment. Text works, as shown in Figure 3.

A section of text inside the figure environment, and some math.

$$E = mc^2$$

Figure 3: A figure composed of nothing but text and some math.

## 2 Sizing and rotating figures

Very often you will want to scale your figures so that they will fit on the page. You do this by adding *options* to the `\includegraphics` command, and those options come before the argument and are enclosed in square brackets, like so.

```
\includegraphics[width=2.5in]{zplot.pdf}
```

The output looks like this.

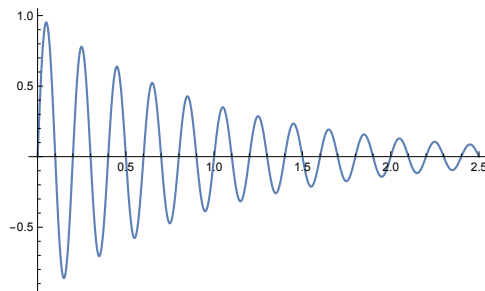


Figure 4: A scaled figure. You can scale the width or the height, and you can use other units besides inches too.

The most useful options for `includegraphics` are `width` and `height`, but you can also control the scale and angle of your figure with those options.

```
\includegraphics[scale=0.35,angle=45]{zplot.pdf}
```

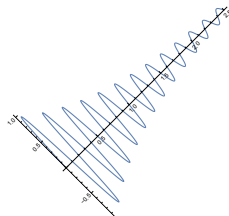


Figure 5: The original figure scaled to 35% of its original size and rotated by a 45 degree angle.

There are many more, but I just want to focus on a few of the most useful ones in this introductory class.

### 3 How figure placement works: Floats

You can split paragraphs across pages easily enough. If your typesetting program reaches the end of a page before a particular paragraph is finished, all it needs to do is just pick up where it left off, starting the next line at the top of the next page. You can even split words across lines by hyphenating them, like the word “pro-gram” split between the end of the first line and the beginning of the second of this paragraph. Latex generally tries to avoid doing this by stretching or compressing the spaces between words just a little bit. It’s not generally enough to notice if you aren’t looking for it, but I’ll bet you can go back and see it in this document, now that you know to look for it.

That’s text. Figures (and tables) are different. They can’t really be split up, so latex has to do what it can to place them appropriately and then move the text around them. This process is called *floating*, and the object (figure or table) being floated is called a *float*. As I mentioned above, the usual default position for a figure is at the top or bottom of the page, or sometimes on a separate page all by itself. Sometimes, however, you want to override this default and place your figure at a specific place in the text, like I did with Figures 1 and 4. The way you do this is by supplying *location arguments* to the environment command. Like options in ordinary commands, location arguments are enclosed in square brackets, but unlike regular options, location arguments come after the type of environment, like so

```
\begin{figure}[h]
```

Location arguments are single, lower-case letters that stand for the words *top*, *bottom*, *here* and *page*. *Top* means place the figure at the top of the page. *Bottom*, as you can guess, means place it at the bottom of the page. *Here* means place the figure right *here* in the text, and *page* means put the figure and its caption on a separate page by itself, without any other text around it. You can have more than one, so if the first location doesn’t work out latex will know what to try next. For example, if you want your figure *here* but would settle for the *bottom* of the page if that’s not possible, you would use

```
\begin{figure}[hb]
```

If you don’t specify any location arguments, the default is “tbp”.

An exclamation point (!) in the location argument tells latex you really, really mean what you say and to override any other rules (within reason) that would interfere with placing the figure right *here!*

```
\begin{figure}[h!]
```

## 4 Wrapping text around figures



Sometimes you want to wrap text around a figure, as shown to the left of this paragraph. To do this you will need to use the package *wrapfig*, which makes the *wrapfigure* environment available. Do this by adding the following line to the preamble of your document.

```
\usepackage{wrapfig}
```

Then, in the text where you want your figure, establish a *wrapfigure* environment with the commands

```
\begin{wrapfigure}{position}{size}
  \includegraphics[size]{your_figure.ext}
\end{wrapfigure}
```

The argument *position* defines whether the figure goes to the *right* of the text or the *left*, and it is just an upper- or lowercase letter R (r) or L (l). The uppercase version allows the figure to float. The lowercase version means *exactly here* [2].

## 5 Exercise

*Exercise 1:* Add the drawing you made in Inkscape last week to your Hello World program. Write the following two lines in your document, “The figure will appear immediately below this line.” and “The figure will appear immediately above this line.” then place your figure between them, like so.

The figure will appear immediately below this line.

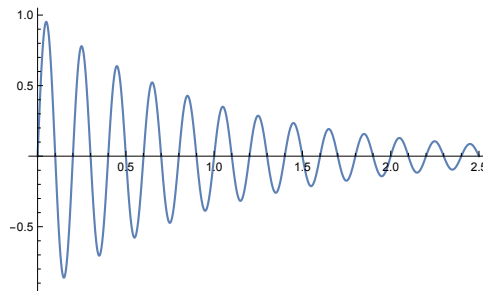


Figure 6: The figure

The figure will appear immediately above this line.

You may have to adjust your document to get the figure to appear between the lines. (In fact, I have arranged things such that you will, if you’ve been following along with the previous exercises.) Once you have done so, add a paragraph explaining what happened and why, and how you got your figure and text to arrange correctly.

*Hint:* There are many ways to solve this problem, and we've covered all the tools you need to do so. One that we haven't covered is the command `\newpage`, which forces a page break. If you choose to use this one, be sure and explain in your paragraph what is going on and how this command resolves the issue.

## References

- [1] Older code sometimes uses the *graphics* package. The *graphicx* package is an newer, *extended* version which supersedes *graphics* and should be your default choice. The *graphicx* package is backward compatible with *graphics* and supports all of the old commands, so there is no reason to use the older, more limited package.
- [2] [https://www.overleaf.com/learn/latex/wrapping\\_text\\_around\\_figures](https://www.overleaf.com/learn/latex/wrapping_text_around_figures). There are two other position arguments possible, O (o) and I (i), for *outside* and *inside* edges of the page, but I cannot recall ever using them.