

A SHORT GUIDE TO FLOATS AND FIGURES IN L^AT_EX

DARYL DEFORD, DAVID FREUND, AND KATIE HARDING

CONTENTS

1. Floating Environments	1
1.1. Example Table	2
2. Figures	2
2.1. Example Figures	3

This is an example document exploring some of the properties of floats and figures in L^AT_EX. The .tex file is included on the webpage so that you can use the examples as templates as you start creating your own documents.

1. FLOATING ENVIRONMENTS

L^AT_EX recognizes automatically two kinds of floating environments: tables and figures. Although it is not necessary to put tables or figures in a floating environment it is usually beneficial to do so. Most importantly, putting an object in a floating environment tells L^AT_EX not to try to split it across pages and allows the software to try to place it in an appropriate position in the document. Additionally, by creating a floating environment for our object, L^AT_EX will allow us to automatically number, caption, and label our object and update these labels as we continue to work.

To start a floating environment we use the familiar `\begin{...}\end{...}` syntax. Both of the floating environments takes an optional argument allowing you to “suggest” a placement for the float. The L^AT_EX processor considers the options in the following order, regardless of the order that you place them in the brackets¹

- (1) **h** for “inline”
- (2) **t** for top of page
- (3) **b** for bottom of page
- (4) **p** for new page figures

Adding an exclamation mark before the symbol (`[!h]`) tells L^AT_EX to ignore some of the standard restrictions on float placement. Even with this in mind floats can end up in surprising places due to size restrictions and the amount of free space available. The float package introduces a new placement argument (`[H]`) that forces floats to remain in place. However, this is usually a mistake – L^AT_EX almost always has a good reason for placing floats where it does.

Within a floating environment we can add captions and labels. The syntax for captioning is simple: `\caption{Description of figure goes here.}`. This will produce:

Figure #. Description of figure goes here.

¹so `[tb]` and `[bt]` will always produce the same output.

where # is the number of the figure in the document. L^AT_EX will place the caption either above or below the floating object depending on where the `\caption{}` is placed in the floating environment. The floating environments also support the `\label{*}` `\ref{*}` commands for automatic labeling and referencing.

1.1. **Example Table.** We learned how to construct a generic table last week. Now we can put that table in a floating environment:

TABLE 1. A great looking table ☺.

⊗	Colors		Gerunds		
1	Red	Blue	Studying	Constructing	Applying
2	Orange	Yellow	Building	Finding	Analyzing
3	Green	Pink	Investigating	Creating	Triangulating

Play around with inserting text above and below the table with the lipsum package (`\lipsum[1-4]`) and adjusting the optional position command (currently `[!h]`). How does the table move around the document? Putting a `\label` in the float means that we can reference the table later using `\ref{Tab:1}`. For example, we could write: the information in Table 1 is critically important. Even better it gets automatically linked for us using the hyperref package and L^AT_EX will take care of renumbering it for us if we decide to reorder our paper later.

2. FIGURES

The standard package for including figures in L^AT_EX is the `graphicx` package. You can see the we have included it in the preamble with `\usepackage{graphicx}`. The `subfig` package (`\usepackage{subfig}`) is also very useful for including and labeling several related images in the same figure.

The basic syntax for adding an image to your document is:

```
\includegraphics[options]{image_name}
```

you might have noticed that the example above² does not include a file extension for the figure. In general, this is a L^AT_EX best practice – as the compiler will determine the appropriate filetype if there are multiple figures with the same name – and as long as you are building to PDF^LA_TE_X you shouldn't have to worry too much.

There are three main types of optional arguments that go before the filename: sizing, rotating, and cropping. The size of the figure can either be controlled by a multiplicative scaling parameter `scaling=2` or a fixed height or width `height=2in`. Rotations are set by an angle measured in degrees, so `angle=180` will turn your figure upside down. Trimming the figure requires the argument `clips=true` as well as a list of four lengths describing how much to crop off each side counterclockwise from the left, so `trim=1em 2em 0em 5em` removes 1em off the left, 2em off the bottom, nothing off the right, and 5em off the top.

Occasionally, you may wish to include multiple figures in the same floating environment. In order to generate separate labels and captions for each image you can use the `subfloat` package. Inside the figure environment you can create subfigures with:

```
\subfloat[sublabel 1]{\includegraphics[options]{image1_name}}
\subfloat[sublabel 2]{\includegraphics[options]{image1_name}}
```

You can use the standard horizontal and vertical spacing commands between the subfloats to arrange them within the figure environment.

²and the examples below

2.1. **Example Figures.** Here are some example figures that you can use as templates. You can download the sample images for these figures here.

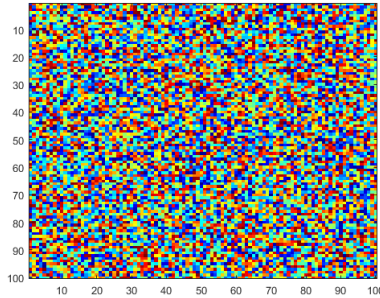


FIGURE 1. Basic Figure Example

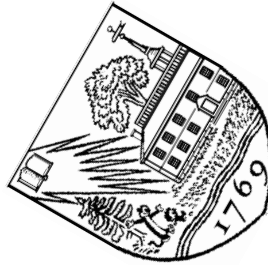
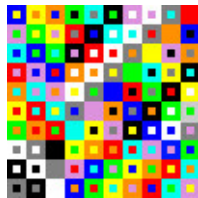


FIGURE 2. Scaled and Rotated Figure Example



(A)
Math
Logo

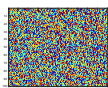


(B)
Noise

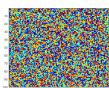


(C)
Seal

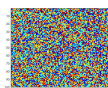
FIGURE 3. Subfig Example



(A)
.eps



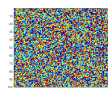
(B)
.png



(C)
.jpg



(D)
.pdf



(E)
none

FIGURE 4. Different File Extension Noise