

Environments, Equations, and Tables

David Freund

October 13, 2016

1 Environments

- In \LaTeX , environments are sections of code delineated by

`\begin{STUFF}` and `\end{STUFF}`.

- Most text formatting is done in environments (e.g., list structures, equations, tables, graphics).
- The `math` environment is special and is instead delineated by `$`. For example, the code for $e^{i\pi}$ is `$e^{i\pi}$`.
- **Important:** Remember to **end** any environment that you **begin**. Without both pieces, your compiler will be sad.

1.1 List Structures

As you would hope, there are a large variety of list structures in L^AT_EX. Moreover, there is a lot of customizability! (A good series of examples can be found at https://en.wikibooks.org/wiki/LaTeX/List_Structures.)

1. The basic list types are: `itemize` (bullets), `enumerate` (numbered), and `description`. These are all *vertical* lists and follow the same general structure:

```
\begin{LIST}
\item First item
\item Second item
\item Etc.
\end{LIST}
```

2. For the `description` environment, the labels are specified by

```
\item[First Thing]
```

rather having a “set” structure.

3. It is possible to nest any of these list environments.
 - (a) The `enumerate` and `itemize` environments will switch to a different label type.
 - (b) The lists do not have to be of the same type.
 - (c) Be sure to end each list!
4. The easiest way to customize labels (systemically) for `enumerate` is by including `\usepackage[shortlabels]{enumitem}` in the preamble. Then you can specify the labeling you want by an optional argument. For example,

```
\begin{enumerate}[(i)] or \begin{enumerate}[i.]
```

5. There is also a horizontal list structure `tasks` but this requires a non-standard package (included by writing `\usepackage{tasks}` in the preamble).

1.2 Equations

The `equation` environment is useful for having numbered equations throughout your document (for ease of reference). If you simply want to include an equation without numbering, the short-hand code is `\[x^n+y^n = z^n \]` (note that it is automatically put in the math environment!) and will return

$$x^n + y^n = z^n.$$

On the other hand, `\begin{equation} x^n + y^n = z^n\end{equation}` gives us

$$x^n + y^n = z^n. \tag{1}$$

While we can do a great deal with this, oftentimes equations won't fit on a single line¹:

$$(3x+1)(x^2+2)^2 = (3x+1)(x^4+2x^2+4) = 3x(x^4+2x^2+4)+1(x^4+2x^2+4) = (3x^5+6x^3+12x)+x^4+2x^2$$

Thankfully, the `align*` (doesn't include equation numbers) or `eqnarray` environments handle this beautifully:

$$\begin{aligned} (3x + 1)(x^2 + 2)^2 &= (3x + 1)(x^4 + 2x^2 + 4) \\ &= 3x(x^4 + 2x^2 + 4) + 1(x^4 + 2x^2 + 4) \\ &= (3x^5 + 6x^3 + 12x) + x^4 + 2x^2 + 4 \\ &= 3x^5 + x^4 + 6x^3 + 2x^2 + 12x + 4 \end{aligned}$$

To use this environment, we have to specify where the math should be *aligned* (using a `&`) and tell the environment where to *break* to a new line (using `\\`). Hence the above equations can be coded by

```
\begin{align*}
(3x+1)(x^2+2)^2 &= (3x+1)(x^4+2x^2+4)\\
&= 3x(x^4+2x^2+4) + 1(x^4+2x^2+4)\\
&= (3x^5+6x^3+12x) + x^4+ 2x^2 + 4\\
&= 3x^5+x^4+6x^3+2x^2+12x+4
\end{align*}
```

¹As a surprise, this code generated *zero* errors. Sometimes, L^AT_EX doesn't notice these "small" issues.

1.3 Cases

Very similar to the `align*` environment is the `cases` environment. This is useful for piecewise functions such as $f(x) = \begin{cases} x+2 & x < 0 \\ x^2 & x \geq 0 \end{cases}$ which is generated by

```
$f(x) = \begin{cases}
x+2 & x < 0 \\
x^2 & x \geq 0
\end{cases}$
```

The two “sides” are separated by `&` while each case is separated by `\\`. The entire `cases` environment assumes math mode, so you will have to be careful if you want text. For example,

```
$f(x) = \begin{cases}
x+2 & \text{\text{if } } x < 0 \\
x^2 & \text{\text{if } } x \geq 0
\end{cases}$
```

will return $f(x) = \begin{cases} x+2 & \text{if } x < 0 \\ x^2 & \text{if } x \geq 0 \end{cases}$. However, without the text formatting, we obtain $f(x) = \begin{cases} x+2 & \text{if } x < 0 \\ x^2 & \text{if } x \geq 0 \end{cases}$.

1.4 Matrices

At some point, you may feel the need to include a matrix in your writing. These have their own environments where you have to specify what kind of border you want. The two standard options are `bmatrix` (square braces) or `pmatrix` (round braces). For example, we have

$$\begin{bmatrix} 1 & 3 & 6 \\ 1 & 2 & 4 \\ 1 & 5 & 10 \end{bmatrix} \text{ vs. } \begin{pmatrix} x & \cos x & e^x \\ 3 & 3 & 0 \end{pmatrix}$$

The latter example can be input as follows:

```
\begin{pmatrix}
x & \cos x & e^x \\
3 & 3 & 0
\end{pmatrix}
```

When creating a matrix, it's important to remember:

- You must specify each row, separating entries by a `&` and a row by `\\`.
- Each row must have the same number of entries.
- The math environment is automatically enabled for each entry.

Another issue that comes up is making arbitrary “big” matrices. When handwriting these matrices, it's customary to use various series of dots to represent a continuing pattern:

<code>... \ldots</code>	<code>... \cdots</code>	<code>∴ \vdots</code>	<code>∴ \ddots</code>
-------------------------	-------------------------	-----------------------	-----------------------

Using these, a generic $n \times n$ matrix $A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$ can be input

via

```
$A = \begin{bmatrix}
a_{11} & a_{12} & \cdots & a_{1n} \\
a_{21} & a_{22} & \cdots & a_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
a_{n1} & a_{n2} & \cdots & a_{nn}
\end{bmatrix}$
```

1.5 Tables

Tables are the most complicated objects we're going to discuss today. A table is created using the `tabular` environment and you must specify how *many* columns you want and the *text alignment* (left, center, right) for each column.

For example, the table

Right	Center	Left	Center
$x + 1$	x	$x - 1$	$\tan x$
bold	plain		Red

can be input into \LaTeX via

```
\begin{tabular}{rclr}
Right & Center & Left & Right\\
 $x+1$  &  $x$  &  $x-1$  &  $\tan x$ \\
\textbf{bold} & plain & & {\color{red}{Red}}\\
\end{tabular}
```

This example highlights some key features of `tabular`:

- You must specify the number of columns when beginning `tabular`.
- You must specify each row, separating entries by a `&` and a row by `\\`.
- Each row must have the same number of entries.
- The math environment must be enabled for each entry.

Also notice that our table has no borders. In \LaTeX , these too must be specified. The vertical lines are specified at the beginning but the horizontal lines are added individually. For instance, combining a lot of options simultaneously,

```
\begin{tabular}{|r||cl|}
Right & Center & Left & Right\\ \hline
 $x+1$  &  $x$  &  $x-1$  &  $\tan x$ \\ \hline
\textbf{bold} & plain & & {\color{red}{Red}}\\
\end{tabular}
```

returns

Right	Center	Left	Center
$x + 1$	x	$x - 1$	$\tan x$
bold	plain		Red

From this “starting” place, there are a lot more options to consider for tables:

- Customizing the height of a row or width of a column.
- Customizing the space between rows or columns.
- Changing the color of a cell, row, or column. (Requires the `xcolor` package.)
- Rotating text (e.g., for labeling a row). (Requires `array` and `graphicx` packages.)
- Combining rows and columns. (Requires the `multirow` and/or the `multicolumn` package.)
- Customizing vertical and horizontal lines (e.g., so that they span only a few cells). (Use `\cline{i-j}` for a horizontal line between columns i and j for that row; the `multicolumn` package is required for partial vertical lines.)

A good reference is <https://en.wikibooks.org/wiki/LaTeX/Tables>.