

L^AT_EX, A Short Course

Floating Figures Tables, and Numbered Environments

Mark G. Eramian

The University of Saskatchewan

February 4, 2017

- Tables are typeset with in the tabular environment.
- The syntax of the tabular environment is as follows:

```
\begin{tabular}[pos]{cols}  
rows  
\end{tabular}
```

- The parameters are:
 - pos* Controls table position (optional)
 - cols* Defines number and format of columns.
 - rows* Table rows (entries in the table).

- Options for *cols* argument are numerous.
- For basic operation *cols* is a string of characters defining table columns:
 - l Column contents are left justified.
 - r Column contents are right justified.
 - c Column contents are centered.
- p{*wth*} Text is set into lines of width *wth* aligned with the top of other columns (needed when table cell contents are to be on multiple lines).
- Each row consists of the contents of each column separated by &'s and ended with a double backslash (\\).

```

\begin{tabular}{lcp{2in}}
Subject& Course Number & Description\\
CMPT& 115& Principles of Computer Science.  Introduces more of the
basic concepts of computer science.\\
CMPT& 461& Intractable problems and models of computation.\\
CMPT& 859& Topics in Computer Vision.  Advanced topics in computer
vision are covered in depth.\\
\end{tabular}

```

Subject	Course Number	Description
CMPT	115	Principles of Computer Science. Introduces more of the basic concepts of computer science.
CMPT	461	Intractable problems and models of computation.
CMPT	859	Topics in Computer Vision. Advanced topics in computer vision are covered in depth.

- Tables can also be drawn with horizontal and vertical lines between rows and columns as desired.
- To insert a vertical line between a pair of columns, put a `|` character between the column entries in the `cols` argument.
- To insert a horizontal line between rows, put a `\hline` at the end of a table row (after the double backslash).
- `\hline` is also acceptable immediately following the `cols` parameter (draws a horizontal line before the first table row).
- A double pipe `||` in the `cols` argument inserts a double vertical line.
- Double horizontal lines are achieved by repeating `\hline`.

```

\begin{tabular}{|l|c|p{2in}|}\hline
Subject& Course Number & Description\\\hline\hline
CMPT& 115& Principles of Computer Science.  Introduces more of the
basic concepts of computer science.\\\hline
CMPT& 461& Intractable problems and models of computation.
CMPT& 859& Topics in Computer Vision.  Advanced topics in computer
vision are covered in depth.\\\hline
\end{tabular}

```

Subject	Course Number	Description
CMPT	115	Principles of Computer Science. Introduces more of the basic concepts of computer science.
CMPT	461	Intractable problems and models of computation.
CMPT	859	Topics in Computer Vision. Advanced topics in computer vision are covered in depth.

- Edit the file `workfiles/table.tex`.
- Typeset the information into a table so it looks like this:

Team	W	L	T	Points For	Points Against	Win Pct
Saskatchewan	8	1	0	235	134	.889
Simon Fraser	7	4	0	319	305	.636
Alberta	5	5	0	290	196	.500
Calgary	4	4	0	187	237	.500
Regina	4	5	0	294	299	.444
Manitoba	3	5	0	172	249	.375
British Columbia	0	8	0	132	260	.000

- Note how table column widths are automatically sized to the widest entry.

- The `array` environment works exactly the same as the `tabular` environment, but may only appear in math mode.
- The advantage is that each table entry is automatically typeset in math mode without the need to enclose each column entry with `$`'s.
- Although you don't need to enclose each table entry with `$`'s, you do need to enclose the entire `array` environment in math mode.
- The `array` environment is very useful for typesetting things like matrices by using the desired auto-sized brackets.

```
\mathbf{F}=\left(\begin{array}{ccc}a_{11}&a_{12}&a_{13} \\a_{21}&a_{22}&a_{23} \\a_{31}&a_{32}&a_{33}\end{array}\right)
```

Produces the following:

$$\mathbf{F} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

- It is possible to instruct \LaTeX to place tables and figures where they will look the “best”, rather than typesetting it at its precise location in the text source.
- This is achieved through the use of the `table` and `figure` environments.
- These environments also automatically number tables/figures and let you define references to them using `\label`.
- Another advantage to these environments is that a list of tables/list of figures can be automatically generated.
- We'll look at tables first...

- In the most basic form, you just place your entire tabular environment within the table environment. To get auto-numbering and captioning, use `\caption`:

```
\begin{table}
\begin{center}
\begin{tabular}{lcp{2in}}
Subject& Course Number & Description\\
CMPT& 115& Principles of Computer Science.
Introduces more of the basic concepts of
computer science.\\
CMPT& 461& Intractable problems and models of
computation.\\
\end{tabular}
\end{center}
\caption{Some courses to be taught in 2004-2005W}
\label{classtable}
\end{table}
```

- The source on the previous page typesets:

Subject	Course Number	Description
CMPT	145	Principles of Computer Science. Introduces more of the basic concepts of computer science.
CMPT	487	Image Processing and Computer Vision.

Table 1: Some courses to be taught in 2004-2005W

- Note: It is very important that the `\label` command appear after the `\caption` command.

- Take the table you made in `workfiles/table.tex` and enclose it in a `table` environment. Also center it if you wish.
- Add a caption and a label. Remember the label must appear after the caption and the caption must be outside the `tabular` environment, but inside the `table` environment.
- References can be forward! Place a sentence *before* the table in the source that uses `\ref` to refer to the table number before it is “declared”.
- When adding a forward reference, one must run \LaTeX twice before the reference will be correctly displayed (a warning will be issued during the first run).

- The `figure` environment works just like the `table` environment except it uses a different numbering counter.
- There is actually nothing stopping you from putting figures in table environments and tables in figure environments. The only difference is in the captions.
- We will see now how to include images. When using `pdflatex`, you can include images in JPEG, PNG, or PDF formats.

- To import images, we need to load the `graphicx` package.
- A *package* is just an “extension” to \LaTeX .
- To load the `graphicx` package, we place the following line in the preamble:

```
\usepackage{graphicx}
```

- Inclusion of this package defines a command called `\includegraphics`.
- The basic usage of this command is

```
\includegraphics{filename}
```

- Let's open up `workfiles/figs.tex` to examine more closely how this works.

- Sometimes \LaTeX will make a really bad decision about where to put floating figures. To force \LaTeX to place a figure/table exactly where it is defined in the source, add the optional `[h]` argument:

```
\begin{table}[h] ... \end{table}
```

- If you really mean it use

```
\begin{table}[h!] ... \end{table}
```

- \LaTeX will not span tabular environments across a page break. If you need this feature, you need to use the `longtable` package.

- The `amsmath` package can be used to define automatically numbered environments for definitions, examples, theorems, etc.
- `\label` and `\ref` can be used with these, just as with figures.

- Define a numbered environment:

```
\usepackage{amsmath}  
  
...  
  
\newtheorem{environment name}{label text}
```

- First argument is the name of the environment; label text gets typeset before the environment's assigned number.
- Some document classes pre-define such environments for definitions, examples, or theorems.

- Try adding a numbered definition to your `figs.tex`:

```
\newtheorem{definition}{Definition}

\begin{definition}
This is the best definition.
\end{definition}
```

- Don't forget to load the `amsmath` package in the preamble.
- Try putting a `\label` command within the definition environment; then add a `\ref` command somewhere to reference it.