

Introduction to Latex

John Hollerbach

January 21, 2016

This is a brief introduction to Latex. Please look at the links on the class web page for further information.

- The preamble should just be copied verbatim. Your text follows after `\begin{document}`. At the very end of your file, place `\end{document}`. The `document` environment is a typical environment, delineated by `\begin{environment}` and `\end{environment}` statements for some environment..
- The `\begin{center}` environment centers the text between beginning and end. Font size is specified by `\large` and `\Large` in this example. **Bold fonts** are specified by `\bf`. The scope of font control is delimited by curly braces. *Italic fonts* are specified by `\it`.

This `itemize` environment is an example of a list environment. Each list element is begun with an `\item` statement. The sentences in a list environment are indented, and there is a blank line between list items.

1. The `itemize` environment contains bulleted lists.
2. The `enumerate` environment creates numbered lists, where the numbers are generated automatically by each `\item` command.

1 Sentences, Paragraphs, and Sections

Just type the text that you want. Paragraphs are delineated by a blank line.

Here is the second paragraph. Indentation is automatic. The first paragraph after a section is not indented by Latex.

1.1 Subsections

Numbering of sections is automatic. Section titles are enclosed in curly braces. The next level of nesting uses `\subsection{}`.

1.1.1 Subsubsections

The final level of nesting uses `\subsubsection{}`.

Omitting Section Numbers

If you don't want section numbers, add an asterick to `section*`, etc.

2 Equations

A general-purpose equation environment is the `eqnarray` environment, which automatically generates equation numbers.

$$y = ax + b \tag{1}$$

If you don't want this equation numbered, use the `eqnarray*` environment.

$$y = ax + b$$

where x is the *input variable* and y is the *output variable*. In paragraph mode, set math symbols by enclosing in dollar signs, e.g., $y = ax + b$. To prevent indentation after the `eqnarray` environment, a `\noindent` statement is added. Please watch your indentations to eliminate spurious ones.

2.1 Equation Referencing

An equation number can be remembered for later referencing by adding a `\label{ }` command.

$$y = ax + b \tag{2}$$

Now we can refer to equation (2) using the `\ref{linear}` command. You can refer to an equation before the `\label` statement defines it, but then you have to run Latex twice over the file (more later on running Latex).

2.2 Aligned Equations

To align equations, delimit the point of alignment using ampersands `&`.

$$y^1 = ax^1 + b \tag{3}$$

$$y^2 = ax^2 + b \tag{4}$$

Alignment is done based on the equal signs. Each line of the set of equations is ended by double slashes `\\`. Equation numbers are generated for both lines. If you don't want equation numbers, use the `eqnarray*` environment. If you only want one of the equations to have a number, add a `\nonumber` command to each equation you don't want to have a number.

$$\begin{aligned} y^1 &= ax^1 + b \\ y^2 &= ax^2 + b \end{aligned} \tag{5}$$

In this example, only the second line gets a number. Also, extra space has been placed between equations by specifying `\[.lin]`. This can be any amount of space you like.

You don't have to align at equal signs. One example is a very long equation that you want to break up into two lines.

$$y = a_n x^n + a_{n-1} x^{n-1} + \dots$$

$$a_2 x^2 + a_1 x + a_0$$
(6)

The equal sign is aligned with a blank in the second line. Note how to create subscripts like `a_2` and superscripts like `x^2`. If more than one character, the subscripts and superscripts have to be enclosed in curly braces, such as `a_{n-1}`.

2.3 Vectors and Matrices

To create a vector or matrix, you can just use the `\matrix` command included in the header to this file.

$$\mathbf{A} = \begin{bmatrix} \cos(\theta_1) & \cos(\theta_1 + \theta_2) \\ \sin(\theta_1) & \sin(\theta_1 + \theta_2) \end{bmatrix}$$
(7)

Each row of matrix **A** is terminated by `\`, with possibly extra space added as in this example. Each column is delineated by an ampersand. You can simply add as many rows and columns as you like including just one column which makes it a vector **a**.

$$\mathbf{a} = \begin{bmatrix} \cos(\theta_1) \\ \sin(\theta_1) \end{bmatrix}$$
(8)

Please use the convention in the class, where

- **a** is a vector when lower-case bold font,
- **A** is a matrix when upper-case bold font, and
- *a* is a scalar when lower-case italic font.

The command `\v` included in this file's header is a convenient way to create bold characters.

Sines, Cosines, Etc.

There is a long list of predefined math functions in Latex.

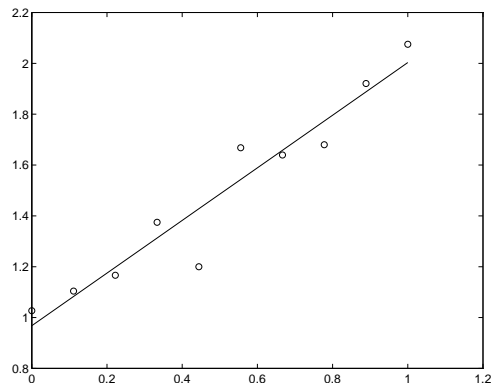


Figure 1: Fitting a straight line in the plane.

$$\begin{aligned} & \sin(\theta) \\ & \cos(\theta) \\ & \tan(\theta) \\ & \arctan(x^2) \\ & \log(x^2 + 1) \end{aligned}$$

More examples can be found in the on-line Latex documentation.

Summations and their limits are created by adding subscripts and superscripts to `\sum`.

$$y = a_0 + \sum_{i=1}^n a_i x^i \tag{9}$$

Integrals and their limits are created similarly:

$$y = \int_0^1 e^x dx \tag{10}$$

2.4 Symbols

Greek letters are created by adding a backslash, such as θ and ϕ . Just capitalize to get upper-case Greek letters like Δ . To get bold font Greek letters such as ϕ , create a definition like that at the beginning of this file:

```
\def\vphi{\hbox{\boldmath$\phi$}}
```

To display this command, the `verbatim` environment (see the source to this file) has been used. It is like `\verb`, but rather than being in-line it is made its own paragraph.

There are quite a few other math symbols, which you can learn about on-line.

2.5 Fractions

Use `\over` to create fractions. The numerator appears to the left, the denominator to the right. Everything is delimited by curly braces.

$$\min_{a,b} S = \sum_{i=1}^P \frac{(y^i - ax^i - b)^2}{1 + a^2} \quad (11)$$

Within text, the math gets crunched in size, such as $\mathbf{C} = \frac{\partial \mathbf{f}}{\partial \phi}$.

3 Figures and Tables

3.1 Single Figures

The `figure` environment is used to create figures. For example, Figure 1 was generated using the commands:

```
\begin{figure}
\centerline{\includegraphics[height=2in]{Figures/Figure2-4.ps}}
\caption{Fitting a straight line in the plane.}
\label{2-4}
\end{figure}
```

Figures have to be created in postscript (.ps) or encapsulated postscript (.eps). You can use any drawing or CAD program for this. The class notes were prepared using `xfig`, available on all Linux platforms.

- The command `\includegraphics[]{ }` is required to display the figure.
- Following the keyword `figure=` is the path, in this case the subdirectory `Figures/` followed by the file name `Figure2-4.ps`.
- The sizing command `height=2in` follows; the figure dimensions are scaled proportionally.
- The `\centerline{ }` command centers the figure. Otherwise, it would be justified to the left by default.
- The `\caption{ }` command creates a caption with a figure number.
- The figure number is determined automatically by Latex, and can be remembered for later referencing using `\label{ }`.

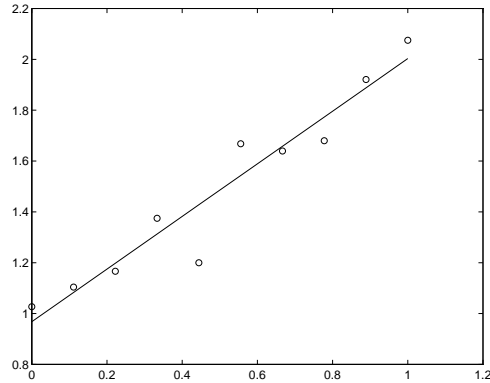


Figure 2: An attempt to fix this figure's position.

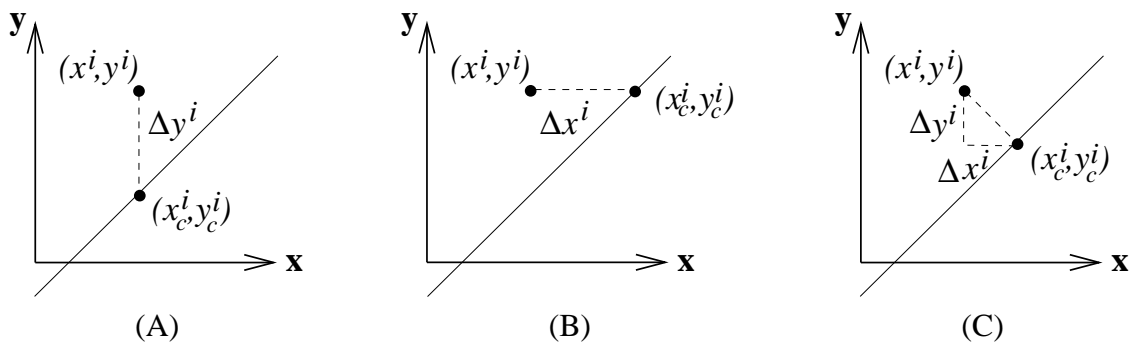


Figure 3: Fitting a straight line in case of (A) output noise only, (B) input noise only, or (C) both input and output noise.

Figures float, that is to say, Latex puts them in places it thinks are convenient. Getting figures on the page you want usually involves moving the location of the figure command around in the text. Sometimes it helps to add [h] to `\begin{figure}[h]`, which may or may not work. This is one of the more frustrating aspects of Latex.

In this case, it didn't work because there wasn't enough space left on the page. It also made this paragraph have the wrong indentation, because I was expecting that the figure would be squeezed between here and the paragraph above.

3.2 Subfigures

To tile multiple images within the same figure, the `tabular` environment should be used. For example, Figure 3 from Chapter 2 was created with:

```
\begin{figure}
\begin{center}\begin{tabular}{c@{\hspace{0.5in}}c@{\hspace{0.5in}}c}
\includegraphics[height=1.5in]{Figures/Figure2-1.eps} &
\includegraphics[height=1.5in]{Figures/Figure2-2.eps} &
\includegraphics[height=1.5in]{Figures/Figure2-3.eps} \\
(A) & (B) & (C)
\end{tabular}\end{center}\end{figure}
```

i	a_i	d_i	α_i	θ_i
1	a_1	d_1	$-\frac{\pi}{2}$	*
2	0	d_2	$\frac{\pi}{2}$	*

Table 1: A spatial 2-DOF rotary manipulator.

```

\end{tabular}\end{center}
\caption{Fitting a straight line in case of (A) output noise only,
(B) input noise only, or (C) both input and output noise.}
\label{2-1}
\end{figure}

```

The `tabular` environment has a preamble `{c@{\hspace{0.5in}}c@{\hspace{0.5in}}c}` which controls the number of columns, the alignment, and any extra space between columns.

- Each column must have one of `l`, `r`, `c` to indicate flush left, flush right, or centered.
- Extra space between columns is created with `@{\hspace{0.5in}}`. If you don't want the extra space, then leave this out. The preamble would then simply be `{ccc}`.

Three figures are placed in a single row. The columns are delineated by an ampersand, and the end of the row is indicated by double backslashes. The next row gives the figures the labels (A), (B), and (C). You can add as many rows as you like, until the end statement is reached. The `tabular` environment is placed within the `center` environment in order to center the figure horizontally.

In displaying these Latex commands, the `verbatim` environment produces rather large text, so you will note in the source file that `\footnotesize` was used to scale it down as compared to the previous example.

3.3 Tables

Use the `table` environment to create tables.

```

\begin{table}
\begin{center}
$\begin{array}{lccrc}
i & & & & & \\
1 & & & & & \\
2 & & & & & \\
\end{array}$
\end{center}
\caption{A spatial 2-DOF rotary manipulator.}
\label{2-DOF-table}
\end{table}

```

The `table` environment is very much like the `figure` environment, except that you get Table 1 rather than Figure 1, etc. Latex doesn't care what you put inside either environment: you can put tables in figures, and photos in tables. In this example, a true table was created using the `array` environment.

The array environment is just like the tabular environment, except that it exists in math mode. Therefore it is bracketed by dollar signs. Adding `\hline` after the double slashes places a horizontal line.

Tables can be put side-by-side just as figures can. Here is an example with two tables:

i	a_i	d_i	α_i	θ_i		i	a_i	d_i	α_i
1	a_1	d_1	$-\frac{\pi}{2}$	*		1	a_1	d_1	$-\frac{\pi}{2}$
2	0	d_2	$\frac{\pi}{2}$	*		2	0	d_2	$\frac{\pi}{2}$
(A)						(B)			

Table 2: (A) A spatial 2-DOF rotary manipulator. (B) The θ_i column is usually omitted for an all-rotary manipulator.

```

\begin{table}[h]
\begin{center}
\begin{tabular}[b]{c@{\hspace*{0.5in}}c}
\begin{array}{lccrc}
i & & & & & \\
1 & & & & & \\
2 & & & & & \\
\end{array} & & & & & \\
\end{array} & & & & & \\
\begin{array}{lccrc}
i & & & & & \\
1 & & & & & \\
2 & & & & & \\
\end{array} & & & & & \\
\end{tabular}
\caption{(A) A spatial 2-DOF rotary manipulator. (B) The  $\theta_i$ 
column is usually omitted for an all-rotary manipulator.}
\label{2-DOF-table-2}
\end{center}
\end{table}

```

A row of tables is created using the tabular environment, just like the multiple figure example.

4 Defining Your Own Latex Commands

You can define your own Latex commands, as shortcuts for longer sequences that are frequently used, or to customize the behavior of Latex. Two examples appear at the beginning of this file's source. The first is:

```
\def\v#1{\hbox{\bf #1}}
```

Typing `\v f` is shortcut for `\hbox{\bf f}`. The command `\def` creates a new command, whose name is `\v` in this case. The `#1` is a placeholder for the first argument, which in the call

`\v f` gets set to `f`. The value of `#1` is then substituted in the command definition. You can have as many arguments as you like, they are numbered consecutively. E.g., a three-argument definition for a quadratic root would be:

```
\def\quadratic#1#2#3{-#2 \pm \sqrt{#2^2 - 4#1#3} \over 2 #1}
```

and calling `\quadratic{a}{b}{c}` would yield:

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (12)$$

Parentheses have been placed around the arguments `a`, `b` and `c`. Otherwise, separate arguments need to be delineated by spaces. E.g., `\quadratic a b c`.

The definition of `quadratic` was placed in this file's source just before it is used. It is better to place all such definitions at the beginning of the file. Better yet would be to place all of your custom definitions into a separate file, and link to it. It is your own customized package, which can be used in all of your Latex work. Create a file named as you like, e.g., `mathop.tex`, and then place the command

```
\input mathop
```

before `\begin{document}`.

5 Compiling Your Latex File

All Latex source files should have the extension `.tex`. You invoke the Latex compiler on a Linux system by the command:

```
latex latexintro.tex
```

Since Latex is a single-pass compiler, you often have to invoke Latex twice in a row to get the references right. A `.dvi` file `latexintro.dvi` is created, which is converted to postscript by a subsequent command:

```
dvipdf latexintro
```

If your figures are in image form (e.g., `jpg`) or `pdf` form, then you have to use `pdflatex`. You cannot mix `eps` files in.

6 References

References are created using BibTex. We will cover BibTex a little later in the course, but you can read about it in the meantime online.