

Introduction to LaTeX

Eleni Akrida

What is Latex?

- A typesetting system used to produce professional looking documents.
- Particularly good at handling mathematically oriented manuscripts.
- Can be used to produce a variety of document types, including presentations, articles and reports.
- LaTeX is also good for management of a bibliography.

Installation

- The machines in the university already have various LaTeX editors and compilers installed on them.
- For installation on your own machine you should consult the following links:
 - MiKTeX Website: <http://miktex.org/>
 - TeXworks Website: <http://www.tug.org/texworks/>
 - TeXnicCenter Website: <http://www.texniccenter.org/>

Structure

```
\documentclass{...}  
\usepackage{...}
```

```
\begin{document}  
...  
\end{document}
```

- The area between `\documentclass{...}` and `\begin{document}` is called the *preamble*. It normally contains commands that affect the entire document.
- After the preamble, the text of your document is enclosed between two commands which identify the beginning and end of the actual document.

Document Classes


```
\documentclass[options]{class}
```



```
\documentclass  
[11pt,twoside,a4paper]{article}
```

- LaTeX needs to know the type of document the author wants to create.
- class specifies the type of document to be created.
- The options parameter customizes the behaviour of the document class

Document Classes

<code>article</code>	For articles in scientific journals, presentations, short reports, program documentation, invitations, ...
<code>IEEEtran</code>	For articles with the IEEE Transactions format.
<code>proc</code>	A class for proceedings based on the article class.
<code>minimal</code>	Is as small as it can get. It only sets a page size and a base font. It is mainly used for debugging purposes.
<code>report</code>	For longer reports containing several chapters, small books, thesis, ...
<code>book</code>	For real books.
<code>slides</code>	For slides. The class uses big sans serif letters.
<code>memoir</code>	For changing sensibly the output of the document. It is based on the <code>book</code> class, but you can create any kind of document with it [1] 
<code>letter</code>	For writing letters.
<code>beamer</code>	For writing presentations (see LaTeX/Presentations).

- LaTeX allows the user to specify a range of different document classes, these are some of the more common ones you will most likely use.
- The most common options for the standard document classes are listed on the next slide.

Document Classes

<code>10pt, 11pt, 12pt</code>	Sets the size of the main font in the document. If no option is specified, 10pt is assumed.
<code>a4paper, letterpaper, ...</code>	Defines the paper size. The default size is <code>letterpaper</code> ; However, many European distributions of TeX now come pre-set for A4, not Letter, and this is also true of all distributions of pdfLaTeX. Besides that, <code>a5paper</code> , <code>b5paper</code> , <code>executivepaper</code> , and <code>legalpaper</code> can be specified.
<code>fleqn</code>	Typesets displayed formulas left-aligned instead of centered.
<code>leqno</code>	Places the numbering of formulas on the left hand side instead of the right.
<code>titlepage, notitlepage</code>	Specifies whether a new page should be started after the document title or not. The article class does not start a new page by default, while report and book do.
<code>onecolumn, twocolumn</code>	Instructs LaTeX to typeset the document in one column or two columns.
<code>twoside, oneside</code>	Specifies whether double or single sided output should be generated. The classes <code>article</code> and <code>report</code> are single sided and the <code>book</code> class is double sided by default. Note that this option concerns the style of the document only. The option <code>twoside</code> does not tell the printer you use that it should actually make a two-sided printout.
<code>landscape</code>	Changes the layout of the document to print in landscape mode.
<code>openright, openany</code>	Makes chapters begin either only on right hand pages or on the next page available. This does not work with the <code>article</code> class, as it does not know about chapters. The <code>report</code> class by default starts chapters on the next page available and the <code>book</code> class starts them on right hand pages.
<code>draft</code>	makes LaTeX indicate hyphenation and justification problems with a small square in the right-hand margin of the problem line so they can be located quickly by a human. It also suppresses the inclusion of images and shows only a frame where they would normally occur.

Packages

```
\usepackage[options]{package}
```



```
\documentclass[11pt,a4paper,oneside]{report}
```

```
\usepackage{color}
```

```
\begin{document}
```

```
...
```

```
\end{document}
```

- You can enhance the capabilities of LaTeX using packages.
- These packages are defined in the preamble section of the document.
- Package is the name of the package.
- Options are a list of keywords that trigger special features in the package (options are not always necessary).

Packages

A comprehensive list of packages and documentation for them can be found: <http://www.ctan.org/>

Here are some of the more commonly used packages that you may find useful:

- `\usepackage{amsmath}` – most important for math environments
(`\align`)
- `\usepackage{graphicx}` – essential for inserting figures
(`\includegraphics`)
- `\usepackage[a4paper]{geometry}` – adjust margins of a certain page
- `\usepackage{hyperref}` – makes citations “clickable” in the .pdf document

Title Page

```
\title{Title Here}

\author{Eleni Akrida\\
Department of Computer
Science, \\
University of Liverpool\\}

\texttt{akridel@Liverpool.
ac.uk}

}

\date{\today}

\maketitle
```

- The double backslash (\\) is the command for a forced line break.
- The `\texttt` command formats the e-mail address using a mono-spaced font.
- The `\today` command is used to insert the current date.
- To omit the date simply leave the braces empty.

Abstract

```
\documentclass{article}  
\begin{document}  
  
\begin{abstract}  
Here is where you would  
write some nice abstract.  
  
\end{abstract}  
  
\end{document}
```

- Appears at the top of the main body of the document, but before the other sections.
- Available for the article and report document classes, not the book class.

Sections

```
\documentclass{article}
\begin{document}
\section{Introduction}
Some text here...
\section{PhD aims}
\subsection{The model we
study}
\subsubsection{Model
Variations}
\subsection{Preliminary
results}
\end{document}
```

- Sections can be used to help structure a document to make it more readable.
- Think of these like chapters or section headings.
- Text placed in the brackets will appear as headings on the document.
- Sections are also useful when creating a contents page.

TOC, LOF, LOT

Once you have your sections set up correctly you can then include a table of contents in your work. To do this simply insert the following command where you would like your contents to appear in the document.

```
\tableofcontents
```

You are also able to include a list of figures or tables that you have specified in the document in a similar way using the following commands.

```
\listoffigures
```

```
\listoftables
```

Lists

Bulleted List

```
\begin{itemize}  
  \item Item 1  
  \item Item 2  
\end{itemize}
```

Numbered List

```
\begin{enumerate}  
  \item Item 1  
  \item Item 2  
\end{enumerate}
```

Description List

```
\begin{description}  
  \item[First] Item 1  
  \item[Second] Item 2  
  \item[Second] Item 3  
\end{description}
```

Nested Lists

```
\begin{enumerate}  
  \item Item 1  
  \begin{enumerate}  
    \item Nested item 1  
    \item Nested item 2  
  \end{enumerate}  
  \item Item 2  
\end{enumerate}
```

Labels and Referencing

```
\section{Hello}  
\label{sec:hello}
```

```
Hello!
```

```
\section{Referencing}
```

```
Now we can refer to  
section~\ref{sec:hello}.
```

- You can label and reference your sections and pretty much any object in your document using the `\label` command.
- You can give a page reference in a similar way by using the `~\pageref{marker}` command.

Tables

```
\begin{tabular}[pos]{table spec}  
...table here...  
\end{tabular}
```

- The `table spec` argument tells LaTeX the alignment to be used in each column and the vertical lines to insert.
- The optional parameter `pos` can be used to specify the vertical position of the table relative to the baseline of the surrounding text.

Tables

The following symbols are available to describe the table columns (some of them require that the package `array` has been loaded):

<code>l</code>	left-justified column
<code>c</code>	centered column
<code>r</code>	right-justified column
<code>p</code> <code>{'width'}</code>	paragraph column with text vertically aligned at the top
<code>m</code> <code>{'width'}</code>	paragraph column with text vertically aligned in the middle (requires <code>array</code> package)
<code>b</code> <code>{'width'}</code>	paragraph column with text vertically aligned at the bottom (requires <code>array</code> package)
<code> </code>	vertical line
<code> </code>	double vertical line

Here is a list of basic commands that can be used in the `tabular` environment:

<code>&</code>	column separator
<code>\\</code>	start new row (additional space may be specified after <code>\\</code> using square brackets, such as <code>\\[6pt]</code>)
<code>\hline</code>	horizontal line
<code>\newline</code>	start a new line within a cell (in a paragraph column)
<code>\cline</code> <code>{i-j}</code>	partial horizontal line beginning in column <i>i</i> and ending in column <i>j</i>

You can use the following letters for the `pos` option:

<code>b</code>	bottom
<code>c</code>	center (default)
<code>t</code>	top

Tables

Basic table example 1:

```
\begin{tabular}{l c r }  
1 & 2 & 3 \\  
4 & 5 & 6 \\  
7 & 8 & 9 \\  
\end{tabular}
```

1	2	3
4	5	6
7	8	9

Tables

Basic table example 2:

```
\begin{tabular}{ l | c || r }  
 1 & 2 & 3 \\  
 4 & 5 & 6 \\  
 7 & 8 & 9 \\  
\end{tabular}
```

1	2	3
4	5	6
7	8	9

Tables

Basic table example 3:

```
\begin{tabular}{ l | c || r }  
 \hline  
 1 & 2 & 3 \\  
 4 & 5 & 6 \\  
 7 & 8 & 9 \\  
 \hline  
 \end{tabular}
```

1	2	3
4	5	6
7	8	9

Tables

Basic table example 4:

```
\begin{center}
\begin{tabular}{l|c|r}
\hline
1 & 2 & 3 \\ \hline
4 & 5 & 6 \\ \hline
7 & 8 & 9 \\ \hline
\end{tabular}
\end{center}
```

1	2	3
4	5	6
7	8	9

Theorems

```
\usepackage{amsthm}

\newtheorem{name}{printed output}

...title, begin main body...
\begin{name}
...some text...
\end{name}
...end main body
```

- Make sure to enable the `amsthm` package.
- Define the theorem in the preamble.
- The first argument is what you will use to reference your theorem.
- The second argument is the output LaTeX will print whenever you use it.
- For more information on Theorems in LaTeX visit: <http://en.wikibooks.org/wiki/LaTeX/Theorems>

Algorithms

There are a number of different methods used to define pseudo code in LaTeX, below is an example using the `algorithm2e` package but more can be found at <http://en.wikibooks.org/wiki/LaTeX/Algorithms>

To use the `algorithm2e` package you must declare it in the preamble with the following command:

```
\usepackage{options}{algorithm2e}
```



```
\begin{algorithm} [H]
\SetAlgoLined
\KwData{this text}
\KwResult{how to write algorithm with \LaTeX2e }
initialization\;
\While{not at end of this document}{
  read current\;
  \eIf{understand}{
    go to next section\;
    current section becomes this one\;
  }{
    go back to the beginning of current section\;
  }
}
\caption{How to write algorithms}
\end{algorithm}
```

```
Data: this text
Result: how to write algorithm with  $\LaTeX 2\epsilon$ 
initialization;
while not at end of this document do
  read current;
  if understand then
    go to next section;
    current section becomes this one;
  else
    go back to the beginning of current section;
  end
end
```

Algorithm 1: How to write algorithms

Maths

For more complicated formulas you should use the following package:

```
\usepackage{amsmath}
```

Inline example:

```
$a + b = c$
```

Displayed example:

```
$$a + b = c$$
```

There are two basic types of formulas that can be used in LaTeX.

The first is inline formulas. These types of formulas are the simplest to include and are defined and displayed in the text you type.

The second are displayed formulas. These are separate from the main text.

For greater depth on either of these visit <http://en.wikibooks.org/wiki/LaTeX/Mathematics>

Maths

```
\alpha,  
\Alpha,  
\beta,  
\Beta,  
\gamma,  
\Gamma,  
\pi, \Pi,  
\phi, \varphi,  
\Phi
```

$\alpha, A, \beta, B, \gamma, \Gamma, \pi, \Pi, \phi, \varphi, \Phi$

```
\frac{n!}{k!(n-k)!} = \binom{n}{k}
```

$$\frac{n!}{k!(n-k)!} = \binom{n}{k}$$

```
\frac{\frac{1}{x} + \frac{1}{y}}{y-z}
```

$$\frac{\frac{1}{x} + \frac{1}{y}}{y-z}$$

Basic symbols:

+ - = ! / () [] < > | ' :

Beyond these special commands must be used in the form of `\command` where you want to display the symbols or functions.

There are many different commands, a comprehensive list can be found at http://en.wikibooks.org/wiki/LaTeX/Mathematics#List_of_Mathematical_Symbols

Advanced Maths

Using the double \$ symbol to create displayed style equations sometimes produces issues when compiling. Therefore it is best to use the `\equation` environment.

This environment will automatically number your equation so that it is easier to refer to it in your main body of text where `someequation` is the name you would give your equation and then refer to it that way in the text as shown below.

```
\begin{equation} \label{eq:someequation}
5^2 - 5 = 20
\end{equation}

this references the equation \ref{eq:someequation}.
```

$5^2 - 5 = 20$ (1)
this references the equation 1.

Advanced Maths

Sometimes you may need to use the `cases` environment if you want to write piecewise functions as shown below.

```
\[
u(x) =
\begin{cases}
\exp{x} & \text{if } x \geq 0 \\
1 & \text{if } x < 0
\end{cases}
\]
```

$$u(x) = \begin{cases} \exp x & \text{if } x \geq 0 \\ 1 & \text{if } x < 0 \end{cases}$$

If you want to add sub- or superscript in summation style symbols with both prime and a limit to a symbol, one have to use the `\sideset` command as shown here.

```
\begin{equation}
\sideset{}{\prime}\sum_{n=1}C_n
\end{equation}
```

$$\sum'_{n=1} C_n$$

There is virtually an unlimited number of combinations that you can put together. More on this can be found here:

http://en.wikibooks.org/wiki/LaTeX/Advanced_Mathematics

Graphics

```
\usepackage{graphicx}
```

```
\includegraphics{image.png}
```

```
\includegraphics[scale=0.5]{image.png}
```

```
\includegraphics[width=2.5cm]{image.png}
```

- Must use the `graphicx` package to import images.
- Use `\includegraphics` command in your text to import the image.
- Many customisation options available.
- For more customisation visit http://en.wikibooks.org/wiki/LaTeX/Importing_Graphics

Graphics

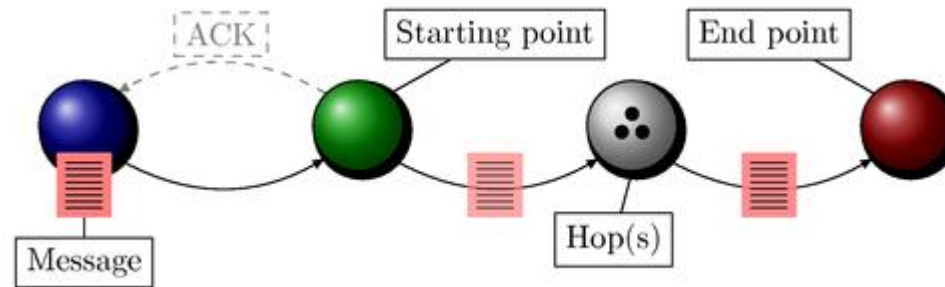
If you want to draw your own diagrams directly in LaTeX you will need to use a specialist drawing package like TikZ.

More on this can be found here

<http://en.wikibooks.org/wiki/LaTeX/PGF/TikZ>

Or <http://paws.wcu.edu/tsfoguel/tikzpgfmanual.pdf>

Or <http://www.texample.net/tikz/examples/>



Figures

```
\begin{figure}[placement specifier]  
... figure contents ...  
\end{figure}
```

Specifier	Permission
h	Place the float <i>here</i> , i.e., <i>approximately</i> at the same point it occurs in the source text (however, not <i>exactly</i> at the spot)
t	Position at the <i>top</i> of the page.
b	Position at the <i>bottom</i> of the page.
p	Put on a special <i>page</i> for floats only.
!	Override internal parameters LaTeX uses for determining "good" float positions.
H	Places the float at precisely the location in the LaTeX code. Requires the <code>float</code> package, ^[1] e.g., <code>\usepackage{float}</code> . This is somewhat equivalent to <code>h!</code> .

- To create a figure that floats, use the figure environment.
- The `placement specifier` parameter exists as a compromise, and its purpose is to give the author a greater degree of control over where certain floats are placed.

Figures

```
\begin{figure}[h!]  
... figure here ...  
  \caption{caption here}  
\end{figure}
```

- You can create captions for your figures using the `\caption` command.
- For more on figures visit http://en.wikibooks.org/wiki/LaTeX/Floats,_Figures_and_Captions

Bibliography

- There are two main ways to include a bibliography in your work.
- You can store them in an external BibTeX file and then link to them via a command to your current document. This way you can create a file of your references that you can simply link to them when you need. For more information on this method visit www.bibtex.org
- However for single documents it is much simpler to embed the references in the file itself.

Bibliography

```
\begin{thebibliography}
{99}

\bibitem{Akrida16}
Eleni Akrida.
Introduction to LaTeX.
\emph{Doctoral Training
Seminar Series}, 2016.
\end{thebibliography}
```

- Place all of your references in separate `\bibitems`.
- Then place all of these inside of `thebibliography` environment.
- Usually placed at the bottom of the main body of your document.
- Refer to the item in your text by typing `~\cite{Gorry13}`

Formats

When writing a document for publication either in conference proceedings or a journal it is important to make sure your document is formatted according to the publishers standards.

Each publisher has their own set of rules and templates that you must use, a few of the more common ones can be found here:

- http://www.ieee.org/conferences_events/conferences/publishing/templates.html
- https://www.acm.org/publications/submissions/latex_style
- <http://www.acm.org/sigs/publications/proceedings-templates>

Tips

- Learning LaTeX is best achieved through doing.
- Google is very useful when using LaTeX
- Compile your PDF frequently as this will help debug any errors that may occur.
- If you can, let LaTeX position things like tables and figures automatically.
- Error messages are easy to understand, you just need to take the time to read them.
- Templates: <http://www.latextemplates.com/>