An Introduction to LaTeX
Scientific writing made easy

Gustaf Hendeby

hendeby@isy.liu.se

This presentation is available from:
http://www.control.isy.liu.se/~hendeby/files/tex-course.pdf

September 10, 2014

Outline

Outline

What is LATEX?

- Based in TEX, a typesetting system by Donald E. Knuth
  - Released in 1978
  - Started by Knuth when he was writing "The Art of Computer Programming" series to 'simplify' things, this set him back 10 years
  - No longer extended, only bug fixed
- LATEX is an extension of TEX by Leslie B. Lamport dating back to 1984
  - Brings structure to TEX
- A *de facto* standard for scientific typesetting, often used in conferences and journals
- Very extensible, most anything you can think of has been done and is available on Internet

D. Knuth
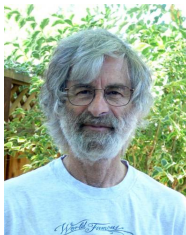
L. Lamport

Alternatives to LaTeX

LyX A free graphical document processor on top of LaTeX, advocates claims it produces LaTeX output; however, it is heavily preprocessed and augmented.

(http://www.lyx.org/)

Scientific WorkPlace Another graphical document processing system supposedly using LaTeX deep in its backend, but not usable interchangeably with LaTeX. It also integrates Maple to produce plots. Scientific WorkPlace is a commercial product for Windows only.

(http://www.mackichan.com/index.html?products/swp.html~mainFrame)

Ms Word Compared to LaTeX in more detail on the next slide.

(http://office.microsoft.com/en-us/word/)

A Comparison: LATEX *vs.* Ms Word

LATEX   **Ms Word**

A Comparison: LaTeX *vs.* Ms Word

LaTeX

- Focus is on the structure; similar to HTML

Ms Word

- Focus is on how the document looks, rather than structure

A Comparison: LaTeX *vs.* Ms Word

LaTeX

- Focus is on the structure; similar to HTML
- A text-file that needs a separate compilation step

Ms Word

- Focus is on how the document looks, rather than structure
- WYSIWYG (What You See Is What You Get) — in theory

A Comparison: LaTeX *vs.* Ms Word

LaTeX

- Focus is on the structure; similar to HTML
- A text-file that needs a separate compilation step
- Has a steep learning curve, but allows for full control

Ms Word

- Focus is on how the document looks, rather than structure
- WYSIWYG (What You See Is What You Get) — in theory
- Intuitive to get started with, difficult to control in detail

A Comparison: LaTeX *vs.* Ms Word

LaTeX

- Focus is on the structure; similar to HTML
- A text-file that needs a separate compilation step
- Has a steep learning curve, but allows for full control
- Works well with large documents, mathematics, cross-references, bibliographies

Ms Word

- Focus is on how the document looks, rather than structure
- WYSIWYG (What You See Is What You Get) — in theory
- Intuitive to get started with, difficult to control in detail
- Limited support for large documents, mathematics, cross-references, bibliographies

A Comparison: LATEX *vs.* Ms Word

LATEX

- Focus is on the structure; similar to HTML
- A text-file that needs a separate compilation step
- Has a steep learning curve, but allows for full control
- Works well with large documents, mathematics, cross-references, bibliographies
- Available for free on all major platforms

Ms Word

- Focus is on how the document looks, rather than structure
- WYSIWYG (What You See Is What You Get) — in theory
- Intuitive to get started with, difficult to control in detail
- Limited support for large documents, mathematics, cross-references, bibliographies
- Only available for Windows and MacOS, is expensive

A Comparison: LATEX *vs.* Ms Word

LATEX

- Focus is on the structure; similar to HTML
- A text-file that needs a separate compilation step
- Has a steep learning curve, but allows for full control
- Works well with large documents, mathematics, cross-references, bibliographies
- Available for free on all major platforms
- Can easily be versioned with standard tools, *e.g.*, git or SVN

Ms Word

- Focus is on how the document looks, rather than structure
- WYSIWYG (What You See Is What You Get) — in theory
- Intuitive to get started with, difficult to control in detail
- Limited support for large documents, mathematics, cross-references, bibliographies
- Only available for Windows and MacOS, is expensive
- Must rely on the build in versioning

LATEX More Than Writing Just Papers

PDF presentations using LATEX and Beamer

This presentation was created using the `beamer` package. As long as you do not require fancy animations, Beamer provides an excellent way to quickly produce beautiful presentations.

(`http://texdoc.net/texmf-dist/doc/latex/beamer/doc/beameruserguide.pdf`)

LATEX More Than Writing Just Papers

PDF presentations using LATEX and Beamer

This presentation was created using the `beamer` package. As long as you do not require fancy animations, Beamer provides an excellent way to quickly produce beautiful presentations.

(`http://texdoc.net/texmf-dist/doc/latex/beamer/doc/beameruserguide.pdf`)

LATEX when writing theses

We recommend LATEX for Master's theses and require it for Lic. and Ph. D. theses. Our thesis class `rtthesis` is freely available and highly recommended.

(`http://www.control.isy.liu.se/student/exjobb/liuthesis/liuthesis.html`)

LaTeX More Than Writing Just Papers

PDF presentations using LaTeX and Beamer

This presentation was created using the `beamer` package. As long as you do not require fancy animations, Beamer provides an excellent way to quickly produce beautiful presentations.

(`http://texdoc.net/texmf-dist/doc/latex/beamer/doc/beameruserguide.pdf`)

LaTeX when writing theses

We recommend LaTeX for Master's theses and require it for Lic. and Ph. D. theses. Our thesis class `rtthesis` is freely available and highly recommended.

(`http://www.control.isy.liu.se/student/exjobb/liuthesis/liuthesis.html`)

Other usages

CV's, letters, posters, lab instructions, exams, . . .

LATEX at Automatic Control (and ISY) ⋆

- Automatic Control strongly promotes LATEX
  - Almost all our target conferences and journal deals only with LATEX or have LATEX as an option
  - Everyone writes their thesis in LATEX, using our templates
- The ISY IT infrastructure supports this
  - The central system and all administrated computers come with LATEX installed
  - We have LATEX templates for most everything (reports, theses, posters, presentations, lab instructions, exams, . . .)
  - Our extensions are included in the local LATEX distribution
  - Non-administrated computer (laptops) install the extensions using SVN
- Our experience is that the initial cost to learn LATEX is heavily outweighed but later time savings

Obtaining and Installing LaTeX

- LaTeX is available for **free** for all operating systems (including sources)
- These are the most popular distributions of LaTeX:

  Windows MikTeX (http://miktex.org/)
  Comes with an easy to use installer, a good package handling system, and a simple LaTeX editor.

  *nix TeX Live (https://www.tug.org/texlive/)
  Available as a package for most Linux distributions, also available for Windows and Mac.

  MacOS MacTeX (https://tug.org/mactex/)
  Available packages for easy installation on Mac.

- A LaTeX installation contains two parts:
  - binaries: TeX, LaTeX, BibTeX, dvi-viewer, makeindex, . . .
  - packages/extensions: a LaTeX distribution is supposed to come with a minimal set of packages of extensions, often it offers more

LATEX Editors

- **Any** text editor will do
- Some common choices are:

AUCTEX Powerful emacs extension
(https://www.gnu.org/software/auctex/
(Mac: http://aquamacs.org/latex.shtml)

Kile (http://kile.sourceforge.net)

TEXniccenter Windows only
(http://www.texniccenter.org)

TEXworks Bundled with with TEX Live
(http://www.tug.org/texworks/)
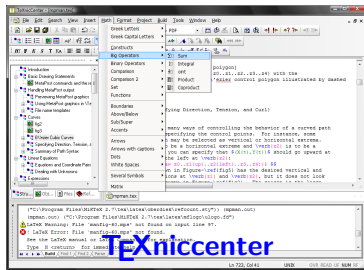
TEXStudio (http://texstudio.sourceforge.net/)

TEXShop Mac only
(http://texstudio.sourceforge.net/)

WinEdt Commercial, Windows only
(http://www.winedt.com/)

writeLATEX Online editing and collaboration
(https://www.writelatex.com)

LATEX Editors

- **Any** text editor will do
- Some common choices are:

AUCTEX Powerful emacs extension
(https://www.gnu.org/software/auctex/)
(Mac: http://aquamacs.org/latex.shtml)

Kile (http://kile.sourceforge.net)

TEXniccenter Windows only
(http://www.texniccenter.org)

TEXworks Bundled with with TEX Live
(http://www.tug.org/texworks/)

TEXStudio (http://texstudio.sourceforge.net/)

TEXShop Mac only
(http://texstudio.sourceforge.net/)

WinEdt Commercial, Windows only
(http://www.winedt.com/)

writeLATEX Online editing and collaboration
(https://www.writelatex.com)

Simple Example: input

LaTeX input: `simple.tex`

```latex
\documentclass[a4paper,10pt,english]{article}

\usepackage{lmodern}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}

\usepackage{babel}

\usepackage{graphicx}
\graphicspath{{./fig/}}

\usepackage{amsmath,amssymb}

\author{John Doe\\Springfield Elementary}
\date{Some date} % Comment out for today
\title{A Simple Example}

\begin{document}
\maketitle
\begin{abstract}
  Write an abstract here to wrap up your work.
\end{abstract}

\section{Introduction}
\label{sec:intro}

Write some nice stuff here\dots

\subsection{Notation}
\label{sec:notation}

See Figure~\ref{fig:notaion}

\subsection{Result}
\label{sec:result}

\begin{figure}
  \centering
  \includegraphics[height=.25\textheight]
                  {notation}
  \caption{This sets the notation.}
  \label{fig:notaion}
\end{figure}

We can prove the following \cite{euler:1200}:
\begin{equation}
  \label{eq:pyth}
  a^2 + b^2 = c^2
\end{equation}

\bibliographystyle{plain}
\bibliography{mybib}

\end{document}
```

Simple Example: input

Required Lines:

- \documentclass — provides the basic layout of the document
- \begin{document}
  ...
  \end{document}
  — marks where to put your text

LATEX input: `simple.tex`

```
\documentclass[a4paper,10pt,english]{article}

\usepackage{lmodern}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}

\usepackage{babel}

\usepackage{graphicx}
\graphicspath{{./fig/}}

\usepackage{amsmath,amssymb}

\author{John Doe\\Springfield Elementary}
\date{Some date} % Comment out for today
\title{A Simple Example}

\begin{document}
\maketitle
\begin{abstract}
  Write an abstract here to wrap up your work.
\end{abstract}

\section{Introduction}
\label{sec:intro}

Write some nice stuff here\dots

\subsection{Notation}
\label{sec:notation}

See Figure~\ref{fig:notation}

\subsection{Result}
\label{sec:result}

\begin{figure}
  \centering
  \includegraphics[height=.25\textheight]
                  {notation}
  \caption{This sets the notation.}
  \label{fig:notation}
\end{figure}

We can prove the following \cite{euler:1200}:
\begin{equation}
  \label{eq:pyth}
  a^2 + b^2 = c^2
\end{equation}

\bibliographystyle{plain}
\bibliography{mybib}

\end{document}
```

Simple Example: input

Required Lines:

- \documentclass — provides
  the basic layout of the document
- \begin{document}
  ...
  \end{document}
  — marks where to put your text

Principal components:

Preamble Everything before
\begin{document}

LATEX input: `simple.tex`

```
\documentclass[a4paper,10pt,english]{article}

\usepackage{lmodern}
\usepackage[T1]{fontenc}
\usepackage{utf8}{inputenc}

\usepackage{babel}

\usepackage{graphicx}
\graphicspath{{./fig/}}

\usepackage{amsmath,amssymb}

\author{John Doe\\Springfield Elementary}
\date{Some date} % Comment out for today
\title{A Simple Example}

\begin{document}
\maketitle
\begin{abstract}
  Write an abstract here to wrap up your work.
\end{abstract}

\section{Introduction}
\label{sec:intro}

Write some nice stuff here\dots

\subsection{Notation}
\label{sec:notation}

See Figure~\ref{fig:notation}

\subsection{Result}
\label{sec:result}

\begin{figure}
  \centering
  \includegraphics[height=.25\textheight]
    {notation}
  \caption{This sets the notation.}
  \label{fig:notation}
\end{figure}

We can prove the following \cite{euler:1200}:
\begin{equation}
  \label{eq:pyth}
  a^2 + b^2 = c^2
\end{equation}

\bibliographystyle{plain}
\bibliography{mybib}

\end{document}
```

Simple Example: input

Required Lines:

- \documentclass — provides the basic layout of the document
- \begin{document}

  ...

  \end{document}

  — marks where to put your text

Principal components:

Preamble Everything before \begin{document}

Body Everything between \begin{document} and \end{document}

LATEX input: `simple.tex`

```
\documentclass[a4paper,10pt,english]{article}

\usepackage{lmodern}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}

\usepackage{babel}

\usepackage{graphicx}
\graphicspath{{./fig/}}

\usepackage{amsmath,amssymb}

\author{John Doe\\Springfield Elementary}
\date{Some date} % Comment out for today
\title{A Simple Example}

\begin{document}
\maketitle
\begin{abstract}
  Write an abstract here to wrap up your work.
\end{abstract}

\section{Introduction}
\label{sec:intro}

Write some nice stuff here\dots

\subsection{Notation}
\label{sec:notation}

See Figure~\ref{fig:notaion}

\subsection{Result}
\label{sec:result}

\begin{figure}
  \centering
  \includegraphics[height=.25\textheight]
                  {notation}
  \caption{This sets the notation.}
  \label{fig:notaion}
\end{figure}

We can prove the following \cite{euler:1200}:
\begin{equation}
  \label{eq:pyth}
  a^2 + b^2 = c^2
\end{equation}

\bibliographystyle{plain}
\bibliography{mybib}

\end{document}
```

Simple Example: input

Preamble

- Document type:

  article, book, report, ...
- Global document settings:

  a4paper, 10pt, english, ...
- Add packages to tune the look or add features
- Own macros

LaTeX input: `simple.tex`

```
\documentclass[a4paper,10pt,english]{article}

\usepackage{lmodern}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}

\usepackage{babel}

\usepackage{graphicx}
\graphicspath{{./fig/}}

\usepackage{amsmath,amssymb}

\author{John Doe\\Springfield Elementary}
\date{Some date}  % Comment out for today
\title{A Simple Example}

\begin{document}
\maketitle
\begin{abstract}
  Write an abstract here to wrap up your work.
\end{abstract}

\section{Introduction}
\label{sec:intro}

Write some nice stuff here\dots

\subsection{Notation}
\label{sec:notation}

See Figure~\ref{fig:notation}

\subsection{Result}
\label{sec:result}

\begin{figure}
  \centering
  \includegraphics[height=.25\textheight]
                  {notation}
  \caption{This sets the notation.}
  \label{fig:notation}
\end{figure}

We can prove the following \cite{euler:1200}:
\begin{equation}
  \label{eq:pyth}
  a^2 + b^2 = c^2
\end{equation}

\bibliographystyle{plain}
\bibliography{mybib}

\end{document}
```

Simple Example: input

Preamble

- Document type:

  article, book, report, …

- Global document settings:

  a4paper, 10pt, english, …

- Add packages to tune the look or add features

- Own macros

Body

- Structure commands: sections, subsections, …

- Text and equations

- Floats: figures, tables, …

- Bibliography, table of content, …

LATEX input: simple.tex

```
\documentclass[a4paper,10pt,english]{article}

\usepackage{lmodern}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}

\usepackage{babel}

\usepackage{graphicx}
\graphicspath{{./fig/}}

\usepackage{amsmath,amssymb}

\author{John Doe\\Springfield Elementary}
\date{Some date}  % Comment out for today
\title{A Simple Example}

\begin{document}
\maketitle
\begin{abstract}
  Write an abstract here to wrap up your work.
\end{abstract}

\section{Introduction}
\label{sec:intro}

Write some nice stuff here\dots

\subsection{Notation}
\label{sec:notation}

See Figure~\ref{fig:notaion}

\subsection{Result}
\label{sec:result}

\begin{figure}
  \centering
  \includegraphics[height=.25\textheight]
                  {notaion}
  \caption{This sets the notation.}
  \label{fig:notaion}
\end{figure}

We can prove the following \cite{euler:1200}:
\begin{equation}
  \label{eq:pyth}
  a^2 + b^2 = c^2
\end{equation}

\bibliographystyle{plain}
\bibliography{mybib}

\end{document}
```
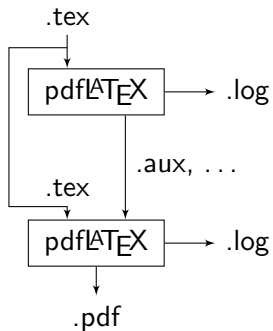
Simple Example: compile

1. Compile your document with LaTeX:

   `$ pdflatex simple.tex`

Simple Example: compile
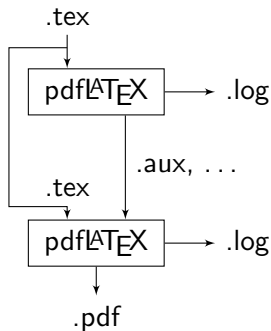
1. Compile your document with LATEX:
   $ pdflatex simple.tex
2. Repeat 1 until all necessary temporary files are in place

Simple Example: compile
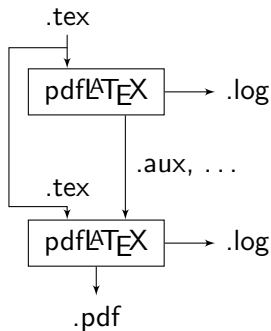
.tex

pdfLATEX → .log

.aux, ...

.tex

pdfLATEX → .log

.pdf

1. Compile your document with LATEX:
   $ pdflatex simple.tex
2. Repeat 1 until all necessary temporary files are in place

Tip

Most LATEX editors make a good job detecting if a rerun is needed, but they sometimes fail. Make it a habit to check the output/log file for warnings indicating a rerun is required.

Simple Example: result

Result: `simple.pdf`

LATEX input: `simple.tex`

```
\documentclass[a4paper,10pt,english]{article}

\usepackage{lmodern}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}

\usepackage{babel}

\usepackage{graphicx}
\graphicspath{{./fig/}}

\usepackage{amsmath,amssymb}

\author{John Doe\\Springfield Elementary}
\date{Some date}   % Comment out for today
\title{A Simple Example}

\begin{document}
\maketitle
\begin{abstract}
  Write an abstract here to wrap up your work.
\end{abstract}

\section{Introduction}
\label{sec:intro}

Write some nice stuff here\dots

\subsection{Notation}
\label{sec:notation}

See Figure~\ref{fig:notaion}

\subsection{Result}
\label{sec:result}

\begin{figure}
  \centering
  \includegraphics[height=.25\textheight]
                  {notation}
  \caption{This sets the notation.}
  \label{fig:notaion}
\end{figure}

We can prove the following \cite{euler:1200}:
\begin{equation}
  \label{eq:pyth}
  a^2 + b^2 = c^2
\end{equation}

\bibliographystyle{plain}
\bibliography{mybib}

\end{document}
```
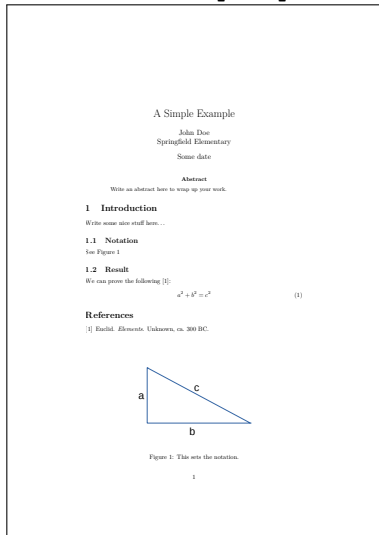
Outline

Plain Text

- All content goes into the *body* of the .tex file

- Text is written 'as is', LATEX will typeset the text and taking care of line and page breaks

- White-space are mostly unimportant, and can be used to make the source easier to read:
  - Words are separated by white-spaces
  - Paragraphs are separated by empty lines

- Macros start with an '\', *e.g.*, \LaTeX to typeset 'LATEX'

- Use % to add comments

Example

```
This is an example of how to write plain text in \LaTeX.
Simply just write
what
you          want
       to say, without caring about line breaks and extra
spaces. This is automatically taken care of.

Use an empty line to separate paragraphs. An appropriate
skip or indent will be automatically included according
to the class and style you are using.

This is one more paragraph. % Use '%' to add comments
```

Result

This is an example of how to write plain text in LATEX. Simply just write what you want to say, without caring about line breaks and extra spaces. This is automatically taken care of.

Use an empty line to separate paragraphs. An appropriate skip or indent will be automatically included according to the class and style you are using.

This is one more paragraph.

Adding Structure

- Add structure to the document using:
  - \part{Title}, \chapter{Title} (mostly for books and reports)
  - \section{Title}, \subsection{Title}, \subsubsection{Title}
  - \paragraph{Title}, \subparagraph{Title}
- Numbering is provided automatically
- Add a * to avoid numbering, *e.g.*, \section*{Title}
- A table of content can be added by adding \tableofcontents

Adding Structure

- Add structure to the document using:
  - `\part{Title}`, `\chapter{Title}` (mostly for books and reports)
  - `\section{Title}`, `\subsection{Title}`, `\subsubsection{Title}`
  - `\paragraph{Title}`, `\subparagraph{Title}`
- Numbering is provided automatically
- Add a * to avoid numbering, *e.g.*, `\section*{Title}`
- A table of content can be added by adding `\tableofcontents`

Tip: Run LATEX twice after changing the structure

Adding a table of content requires LATEX to be run twice to get the contents right! The temporary .toc file must be updated.

Add Highlighting and Formatting

To *Emphasize* words

Use: \emph{word}

Add Highlighting and Formatting

To *Emphasize* words

Use: `\emph{word}`

Direct changing the formatting (often discouraged)

- Change the font:
  bold: `\textbf{word}`,   *italic*: `\textit{word}`,   roman: `\textrm{word}`,   SMALL CAPS: `\textsc{word}`,
  `typewriter`: `\texttt{word}`,   <u>under line</u>: `\underline{word}`

- Change color: `\textcolor{color}{word}` — requires the `color` package

- Change size (use braces around the command and the word!):
  `\tiny`, `\scriptsize`, `\footnotesize`, `\small`, `\normalsize`, `\large`,
  `\Large`,   `\LARGE`,   `\huge`,   `\Huge`

Example: structure

LATEX input:

```
\tableofcontents

\section{Introduction}
Mei at error honestatis. Ex malis molestie vituperata
cum. Suscipit disputando est ea, ex pro stet purto
labitur. An pro populo probatus. Mel aperiam
concludaturque ne.

\subsection{Background}
\label{sec:background}
Ut sed suscipit democritum, ut ius velit atomorum, et
noluisse theophrastus nam. Ei est quis propriae
intellegebat.

Te solet aliquip labitur sit. Omnesque salutandi
maluisset eam ex. Bonorum consequat quo ut, nec
tollit iudicabit signiferumque an.

\subsection{Related Work}
Eu dicta molestie cum, nam ex quando laudem. Omnium
phaedrum et has, facete dolorum vivendo mel ea.

\section{Theory}
\label{sec:theory}
Est ridens noluisse eu, natum accusam in mel, sea no
scripta efficiendi.

\subsection{No Number}

Qui ex velit melius, an quo copiosae hendrerit evertitur,
autem nulla eu est. Eum tota electram te. Vix wisi elitr
eloquentiam et.

\subsection{Method 1}
Errem lobortis mei ad. Nec in justo audiam
definitionem. Errem elaboraret eum ei, per eu vidisse
voluptua intellegebat.

\subsubsection{Submethod 1a}
Has eu idque nullam essent. Modo civibus prodesset ut
cum. Elit wisi has an, lorem aeque ex vis.

\subsubsection{Submethod 1b}
In sale dicant scribentur duo, sale dissentias
necessitatibus has at. Nonumy fastidii scribentur qui
eu. Pro facilisi voluptatum vituperata ut, quod soleat
sadipscing no pro. An omnes iuvaret quo.

\section{Results}
Quo fastidii nominati laboramus id. Ferri delectus
forensibus at per, per partem intellegat ne. Vel labores
explicari corrumpit ea, nam ex feugiat salutatus. Ex sea
```

Result:

Contents

1 Introduction

Mei at error honestatis. Ex malis molestie vituperata cum. Suscipit disputando est ea, ex pro stet purto labitur. An pro populo probatus. Mel aperiam concludaturque ne.

1.1 Background

Ut sed suscipit democritum, ut ius velit atomorum, et noluisse theophrastus nam. Ei est quis propriae intellegebat.

Te solet aliquip labitur sit. Omnesque salutandi maluisset eam ex. Bonorum consequat quo ut, nec tollit iudicabit signiferumque an.

1.2 Related Work

Eu dicta molestie cum, nam ex quando laudem. Omnium phaedrum et has, facete dolorum vivendo mel ea.

2 Theory

Est ridens noluisse eu, natum accusam in mel, sea no scripta efficiendi.

2.1 No Number

Qui ex velit melius, an quo copiosae hendrerit evertitur, autem nulla eu est. Eum tota electram te. Vix wisi elitr eloquentiam et.

2.2 Method 1

Errem lobortis mei ad. Nec in justo audiam definitionem. Errem elaboraret eum ei, per eu vidisse voluptua intellegebat.

2.2.1 Submethod 1a

Has eu idque nullam essent. Modo civibus prodesset ut cum. Elit wisi has an, lorem aeque ex vis.

1

Environments

- Environments are used to group and format large blocks of text
- An environment starts with \begin{env} and ends with \end{env}
- The *body* is an example of an environment
- Examples, change env above to...:

center	To center the text
quotation	To enter a long quote
verbatim	Typewriter font, exactly as written writ line breaks
	⋮

- Many packages provide useful environments

Environments: lists

- Lists is a very useful type of environments
    - A bullet list is obtained using the environment `itemize`
    - An enumerated list is obtained using the environment `enumerate`
- `\item` is used to separate items in a list
- Use an optional argument to change the 'bullet' content, *e.g.*, `\item[My item content]`
- It is possible to have a list in a list
- Lists are highly configurable (roman or latin numbers, . . .), and many packages exist to do that for you, *e.g.*, `enumerate`

Environments: lists

- Lists is a very useful type of environments
  - A bullet list is obtained using the environment `itemize`
  - An enumerated list is obtained using the environment `enumerate`
- `\item` is used to separate items in a list
- Use an optional argument to change the 'bullet' content, *e.g.*,
  `\item[My item content]`
- It is possible to have a list in a list
- Lists are highly configurable (roman or latin numbers, . . .), and many packages exist to do that for you, *e.g.*, `enumerate`

---

enumerate list

```
\begin{enumerate}
\item Item 1.
\item Item 2 with subitems:
  \begin{enumerate}
  \item Subitem 1.
  \item Subitem 2.
  \end{enumerate}
\end{enumerate}
```

1. Item 1.
2. Item 2 with subitems:
   (a) Subitem 1.
   (b) Subitem 2.

Environments: lists

- Lists is a very useful type of environments
  - A bullet list is obtained using the environment itemize
  - An enumerated list is obtained using the environment enumerate
- \item is used to separate items in a list
- Use an optional argument to change the 'bullet' content, *e.g.*, \item[My item content]
- It is possible to have a list in a list
- Lists are highly configurable (roman or latin numbers, ...), and many packages exist to do that for you, *e.g.*, enumerate

---

itemize list

```
\begin{enumerate}
\item Item 1.
\item Item 2 with subitems:
  \begin{enumerate}
  \item Subitem 1.
  \item Subitem 2.
  \end{enumerate}
\end{enumerate}
```

- Item 1.
- Item 2 with subitems:
  - Subitem 1.
  - Subitem 2.

Typesetting Mathematics

- LATEX was designed to allow for easy typesetting of mathematical equations in a beautiful way. The ideas is that every Greek letter, symbol, operator, etc have their own 'text' representation, and that these can be combined using simple rules.

- Once mastered, typesetting mathematics using pure text is very powerful, unfortunately the learning curve is bit steep!
  Do not let this discourage you from using LATEX!

- Many of the suggested LATEX editor provide support via point-and-click interfaces similar to the equation editor in Ms Word. None-the-less, it is still important to understand what happens behind the scene. This way it is possible to express more advanced concepts and make small adjustments to make things look perfect.

Typesetting Mathematics

- LATEX was designed to allow for easy typesetting of mathematical equations in a beautiful way. The ideas is that every Greek letter, symbol, operator, etc have their own 'text' representation, and that these can be combined using simple rules.

- Once mastered, typesetting mathematics using pure text is very powerful, unfortunately the learning curve is bit steep!
  Do not let this discourage you from using LATEX!

- Many of the suggested LATEX editor provide support via point-and-click interfaces similar to the equation editor in Ms Word. None-the-less, it is still important to understand what happens behind the scene. This way it is possible to express more advanced concepts and make small adjustments to make things look perfect.

AMS mathematical packages

Make habit of including the `amsmath` and `amssymb` packages from the American Mathematical Society (AMS). Some of the examples presented here require them.

Mathematics: Greek letters and symbols

Greek letters (those differing from latin letters)

α: \alpha, β: \beta, γ: \gamma, Γ: \Gamma,
ω: \omega, Ω: \Omega, ε: \varepsilon, φ: \varphi

Mathematics: Greek letters and symbols

Greek letters (those differing from latin letters)

α: \alpha, β: \beta, γ: \gamma, Γ: \Gamma,

ω: \omega, Ω: \Omega, ε: \varepsilon, φ: \varphi

Operators

$+$: +, $-$: -, $/$: /, $\frac{a}{b}$: \frac{a}{b}, $\tfrac{a}{b}$: \tfrac{a}{b},

$\sum_{i=1}^{n}$: \sum_{i=1}^{n}, $\int_{0}^{\infty}$: \int_{0}^{\infty}

Mathematics: Greek letters and symbols

Greek letters (those differing from latin letters)

α: \alpha, β: \beta, γ: \gamma, Γ: \Gamma,
ω: \omega, Ω: \Omega, ε: \varepsilon, φ: \varphi

Operators

$+$: +, $-$: -, $/$: /, $\frac{a}{b}$: \frac{a}{b}, $\tfrac{a}{b}$: \tfrac{a}{b},
$\sum_{i=1}^{n}$: \sum_{i=1}^{n}, $\int_{0}^{\infty}$: \int_{0}^{\infty}

Relations

$<$: <, $\leq$: \leq, $\therefore$: \therefore, $\Rightarrow$: \Rigtharrow, $\forall$: \forall

Mathematics: Greek letters and symbols

Greek letters (those differing from latin letters)

α: `\alpha`, β: `\beta`, γ: `\gamma`, Γ: `\Gamma`,
ω: `\omega`, Ω: `\Omega`, ε: `\varepsilon`, φ: `\varphi`

Operators

$+$: `+`, $-$: `-`, $/$: `/`, $\frac{a}{b}$: `\frac{a}{b}`, $\tfrac{a}{b}$: `\tfrac{a}{b}`,
$\sum_{i=1}^{n}$: `\sum_{i=1}^{n}`, $\int_{0}^{\infty}$: `\int_{0}^{\infty}`

Relations

$<$: `<`, $\leq$: `\leq`, $\therefore$: `\therefore`, $\Rightarrow$: `\Rigtharrow`, $\forall$: `\forall`

Special fonts

$\mathcal{A}$: `\mathcal{A}`, $\mathbb{R}$: `\mathbb{R}`, $\phi\phi$: `\bm{\phi}\phi`,
txt: `\mathrm{txt}`

Mathematics: more fundamentals

- Many functions are defined as 'macros', simply prefix with '\'
  $\sin(x) \leftrightarrow sin(x)$: `\sin(x)\leftrightarrow sin(x)`

Mathematics: more fundamentals

- Many functions are defined as 'macros', simply prefix with '\'
  $\sin(x) \leftrightarrow sin(x)$: `\sin(x)\leftrightarrow sin(x)`
- Superscripts are obtained with ^: a^b: `a^b`
- Subscripts are obtained with _: a^b: `a_b`

Mathematics: more fundamentals

- Many functions are defined as 'macros', simply prefix with '\'
  $\sin(x) \leftrightarrow sin(x)$: `\sin(x)\leftrightarrow sin(x)`
- Superscripts are obtained with ^: a^b: `a^b`
- Subscripts are obtained with _: a_b: `a_b`
- Super and subscripts can be combined (max 1 of each):
  a_c^b (`a^b_c` or `a_c^b`)

Mathematics: more fundamentals

- Many functions are defined as 'macros', simply prefix with '\'
  $sin(x) \leftrightarrow sin(x)$: \sin(x)\leftrightarrow sin(x)
- Superscripts are obtained with ^: a^b: a^b
- Subscripts are obtained with _: a_b: a_b
- Super and subscripts can be combined (max 1 of each):
  a_c^b (a^b_c or a_c^b)
- To get a prime, use ': C': C'

Mathematics: more fundamentals

- Many functions are defined as 'macros', simply prefix with '\'
  $\sin(x) \leftrightarrow sin(x)$: `\sin(x)\leftrightarrow sin(x)`
- Superscripts are obtained with ^: a^b: `a^b`
- Subscripts are obtained with _: a^b: `a_b`
- Super and subscripts can be combined (max 1 of each):
  a_c^b (`a^b_c` or `a_c^b`)
- To get a prime, use ': C': `C'`
- Grouping is obtained with {...}:
  - $a^{bc} \Leftrightarrow a^bc$: `a^{bc}\Leftrightarrow a^bc`
  - `a^b^c`: Error! A symbol cannot have several superscripts
  - a^{b^c}: `{a^b}^c` Here a^b is treated as a symbol

Including Mathematics

Two types of mathematics:

Text style math

Mathematics inlined in text, added by surrounding the math with $...$

Display style math

Mathematics displayed on separate (numbered) line(s); in an environment, *e.g.*, equation

Example

```
Inline mathematics is suitable for short
mathematical expression's that fit well
into the remaining text.   Assume
$z = f(x, y)$, where $x\in\mathbb{R}$ and
$y,z\in\mathbb{C}$.  However it is difficult
to refer to inline math, as it lacks numbering.

Display style math, such as,
\begin{equation}
  \label{eq:pythagoras}
  z^2
  = x^2 + y^2 ,
\end{equation}
yields only math on a line, and a number
to refer to.
```

Result

Inline mathematics is suitable for short mathematical expression's that fit well into the remaining text. Assume $z = f(x, y)$, where $x \in \mathbb{R}$ and $y, z \in \mathbb{C}$. However it is difficult to refer to inline math, as it lacks numbering.

Display style math, such as,

$$z^2 = x^2 + y^2,\qquad (1)$$

yields only math on a line, and a number to refer to.

Display Mathematics

- Display math environments comes in many variations, some important ones are:

  equation A single plain equation

  gather Several equations, not aligned

  align Several equations, aligned with &, *e.g.*, on =-signs

  multline One equation spread over several lines.

- Use \\ to separate lines (equations) in multi-line environments
- Suffix the environment with a * to drop the numbering
- \tag/\notag can be used to force a number/no number for an equation
- Use \intertext{this yields} for a short text between equations in a multiequation environment
- Putting several equations in a subequations environment yields sub-equation numbering: (1a), (1b),...

Example: mathematics

Example

```
This is what an align-environment looks like
\begin{align}
    \label{eq:a:dyn}
    x_{t+1} &= f(x_t, u_t)\\
    \label{eq:a:meas}
    y_t     &= h(x_t, u_t) .
\end{align}
Whereas gather equations are not aligned
\begin{subequations}
    \begin{gather}
        \label{eq:g:dyn}
        x_{t+1} = f(x_t, u_t)\\
        \label{eq:g:meas}
        y_t = h(x_t, u_t) .
    \end{gather}
\end{subequations}
And this is a really long equation
\begin{multline}
    \label{eq:long}
    f(x, y, z) = \int_x^y \sin(\tau)^x\,d\tau \\
    \sin(x)\cos(y)\|z\|\\
    \dot f(x, y, z)
\end{multline}

Finally, an example without numbering
\begin{align*}
    x_{t+1} &= f(x_t, u_t)\\
    y_t     &= h(x_t, u_t) .
\end{align*}
```

Result

This is what an align-environment looks like

$$x_{t+1} = f(x_t, u_t) \tag{2}$$
$$y_t = h(x_t, u_t). \tag{3}$$

Whereas gather equations are not aligned

$$x_{t+1} = f(x_t, u_t) \tag{4a}$$
$$y_t = h(x_t, u_t). \tag{4b}$$

And this is a really long equation

$$f(x, y, z) = \int_x^y \sin(\tau)^x\,d\tau$$
$$\sin(x)\cos(y)\|z\|$$
$$\dot f(x, y, z) \tag{5}$$

Finally, an example without numbering

$$x_{t+1} = f(x_t, u_t)$$
$$y_t = h(x_t, u_t).$$

More Mathematics

- Newton-notation derivatives: $\dot{x}$ (\dot{x}), $\ddot{x}$ (\ddot{x}), $\dddot{x}$ (\dddot{x})
- Other 'hat'-like annotations: $\hat{x}$: \hat{x}, $\tilde{x}$: \tilde{x}, $\bar{x}$: \bar{x}

More Mathematics

- Newton-notation derivatives: $\dot{x}$ (\dot{x}), $\ddot{x}$ (\ddot{x}), $\dddot{x}$ (\dddot{x})
- Other 'hat'-like annotations: $\hat{x}$: \hat{x}, $\tilde{x}$: \tilde{x}, $\bar{x}$: \bar{x}
- For matrices use the pmatrix or bmatrix environment, separate elements with & (columns) and \\ (rows)

$$\text{pmatrix:} \begin{pmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{pmatrix} \qquad \text{bmatrix:} \begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix}$$

More Mathematics

- Newton-notation derivatives: $\dot{x}$ (\dot{x}), $\ddot{x}$ (\ddot{x}), $\dddot{x}$ (\dddot{x})
- Other 'hat'-like annotations: $\hat{x}$: \hat{x}, $\tilde{x}$: \tilde{x}, $\bar{x}$: \bar{x}
- For matrices use the `pmatrix` or `bmatrix` environment, separate elements with & (columns) and \\ (rows)

$$\text{pmatrix: } \begin{pmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{pmatrix} \qquad \text{bmatrix: } \begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix}$$

- Parenthesis/delimiter sizing
  - Prefer using \big, \bigg, \Big, \Bigg and prefix with l, m, or r if necessary to indicate side.
    \Biggl\{\bigg|\Bigm\|\big(\Biggr]
  - Using a \left \right pair is an option, but usually yields too big symbols (use '.' as nothing)

$$\left\{ \left| \, \right\| \left(\, \right] \right.$$

Cross-References

Cross-references are easy in LATEX:

1. Mark potential targets for cross-references with label
   - Use descriptive and unique labels, *e.g.*, \label{sec:intro}
   - Using \label stores the last generated 'number' and its location
   - Tip: A common error is to put the label command before the numbering is generated and get the wrong numbering
2. Use one of the following to make a reference in the text:
   - \ref{sec:intro}: refer to a item number
   - \pageref{sec:intro}: refer to the page with the item on it
   - \eqref{eq:pyt}: refer to an equation (the result is enclosed in parenthesis)
   - Additional packages can generate fancier references

Example: cross-references

LATEX input:

```
\section{Results}
Quo fastidii nominati laboramus id. Ferri delectus
forensibus at per, per partem intellegat ne. Vel labores
explicari corrumpit ea, nam ex feugiat salutatus. Ex sea
vivendo periculis, eam iusto nobis in. Quot duis dolores
mea id.

As indicated in Sec.~\ref{sec:background} on page
\pageref{sec:background} and further described in
Sec.~\ref{sec:theory}, the following holds for certain
triangles
\begin{equation}
  \label{eq:pythagoras}
  x^2 + y^2 = z^2 .
\end{equation}

Referring back to \eqref{eq:pythagoras}, we can conclude
that $\lvert x \rvert < \lvert x \rvert$.
```

Result:

2.2.2 Submethod 1b

In sale dicant scribuntur duo, sale dissentias necessitatibus has at. Nonumy fastidii scribentur qui eu. Pro facilisi voluptatum vituperata ut, quod soleat sadipscing no pro. An omnes iuvaret quo.

3   Results

Quo fastidii nominati laboramus id. Ferri delectus forensibus at per, per partem intellegat ne. Vel labores explicari corrumpit ea, nam ex feugiat salutatus. Ex sea vivendo periculis, eam iusto nobis in. Quot duis dolores mea id.

As indicated in Sec. 1.1 on page 1 and further described in Sec. 2, the following holds for certain triangles

$$x^2 + y^2 = z^2. \qquad (1)$$

Referring back to (1), we can conclude that $|x| < |x|$.

Example: cross-references

LATEX input:

```
\section{Results}
Quo fastidii nominati laboramus id. Ferri delectus
forensibus at per, per partem intellegat ne. Vel labores
explicari corrumpit ea, nam ex feugiat salutatus. Ex sea
vivendo periculis, eam iusto nobis in. Quot duis dolores
mea id.

As indicated in Sec.~\ref{sec:background} on page
\pageref{sec:background} and further described in
Sec.~\ref{sec:theory}, the following holds for certain
triangles
\begin{equation}
  \label{eq:pythagoras}
  x^2 + y^2 = z^2.
\end{equation}

Referring back to \eqref{eq:pythagoras}, we can conclude
that $\lvert x \rvert < \lvert x \rvert$.
```

Result:

2.2.2 Submethod 1b

In sale dicant scribentur duo, sale dissentias necessitatibus has at. Nonumy fastidii scribentur qui eu. Pro facilisi voluptatum vituperata ut, quod soleat sadipscing no pro. An omnes iuvaret quo.

3 Results

Quo fastidii nominati laboramus id. Ferri delectus forensibus at per, per partem intellegat ne. Vel labores explicari corrumpit ea, nam ex feugiat salutatus. Ex sea vivendo periculis, eam iusto nobis in. Quot duis dolores mea id.

As indicated in Sec. 1.1 on page 1 and further described in Sec. 2, the following holds for certain triangles

$$x^2 + y^2 = z^2. \tag{1}$$

Referring back to (1), we can conclude that $|x| < |x|$.

Warning: Make sure to rerun LATEX enough times

Cross-references the table of content and similar cannot be obtained in only reading the text once; hence, it is important to rerun LATEX enough times to get the references right.

The log file indicates when it is necessary to rerun LATEX.

Split a Document in Several Files ⋆

- Writing large documents in a single file is inconvenient, splitting in logical parts simplifies handling
- Use \input{filename} to input filename as is into your file
  - Can be nested
- Use \include{filename} to include filename
  - Will produce a page break before the content (suitable for chapters)
  - Cannot be nested
  - Use \includeonly to include just some parts, the numbering will work as if the excluded parts were included
- Use \include for chapters in large reports/theses, and \input for tablas/scripted figures or sections in lengthy papers
- Good LATEX editors allow for easy handling of multi-file documents

Useful Tricks: breaks ⋆

- To enforce breaks use:

  \linebreak break the current line, the text will remain justified

  \\ break the current line, the content is flushed to the side; use \\[1ex] to add extra spacing

Useful Tricks: breaks ★

- To enforce breaks use:

  \linebreak break the current line, the text will remain justified

  \\ break the current line, the content is flushed to the side; use \\[1ex] to add extra spacing

  \pagebreak Start a new page/column at the end of the current line, do not start a new paragraph

  \newpage Start a new page/column right here, the page is flushed to the top

  \clearpage Like \newpage but always produces a new page and processes any leftover floats before continuing

  \cleardoublepage Like \clearpage but adds an extra blank page to make sure to start on a right hand side (odd numbered) page in twopage mode

Useful Tricks: breaks, spaces, and special characters ⋆

- Use \␣ to enforce a space

Useful Tricks: breaks, spaces, and special characters ⋆

- Use `\␣` to enforce a space
- To avoid line break:

~	Non-breaking space, avoids a line break between words
`\mbox{...}`	Everything in `\mbox` is treated as an inseparable unit

Useful Tricks: breaks, spaces, and special characters ⋆

- Use `\␣` to enforce a space
- To avoid line break:

  ~ Non-breaking space, avoids a line break between words
  `\mbox{...}` Everything in `\mbox` is treated as an inseparable unit

- Escape special characters with special meaning to LATEX:
  $: \$,   &: \&,   %: \%,   #: \#,   {: \{,   }: \},   _: _

Useful Tricks: breaks, spaces, and special characters ⋆

- Use `\␣` to enforce a space
- To avoid line break:

  ˜ Non-breaking space, avoids a line break between words
  `\mbox{...}` Everything in `\mbox` is treated as an inseparable unit
- Escape special characters with special meaning to LATEX:
  $: `\$`, &: `\&`, %: `\%`, #: `\#`, {: `\{`, }: `\}`, _: `\_`
- When using non-ASCII/Swedish characters (å, ä, ö, ...):
  - Set the input encoding to UTF8/Latin1 accordingly:
    `\usepackage[utf8]{inputenc}`
  - Set the font encoding to allow for searching the PDF:
    `\usepackage[T1]{fontenc}`
  - Choose a font with proper characters, *e.g.*, `lmodern` or `times`
  - Watch out for conference/journal requirements!
    They might disallow some of these steps!

Miscellaneous ⋆

- To produce a footnote[1], use the
  `\footnote{Footnote goes here.}` command.
- The `babel` package provides internationalization for words such as
  "Figure", "Table of Contents", etc. It also sets the right hyphenation
  pattern. (MikTEX users should make sure that Swedish language
  support is installed before attempting to use it.)
- It is sometimes useful to be able to typeset content in a container with
  a given width. The `minipage` environment provides this ability, one
  usage is to combine figure material to be put side by side on a page.

---

[1]This is a footnote!

Outline

Floats

- LATEX typeset material to appear out of the normal linear text flow (figures, tables, etc), in *floats*
- Floats are produced by specific environments
  - A float is, to LATEX, a box that is placed automatically, based on you preferences
  - Floats are placed to achieve a visually appealing look according to strict rules (which can be overridden)
  - The placement of a float cannot be fully controlled (without extra packages)
- Float environments typically come with the option to add a caption

Figures

- Create a figure float with \begin{figure}...\end{figure}
- You can put **any** LaTeX content in a figure, not just images
- To add a caption, use the \caption{My caption} command; it can be used several times if you want
- For a list of all figures in a document, use \listoffigures which works similar to the table of content

Figure: `graphicx` package

Including images

To include images use the `graphicx` package:

- Syntax: `\includegraphics[opts]{image}`
- `opts` modify the result, *e.g.*,
  `width=1cm`, `width=.95\columnwidth`, `height=.9\textheight`,
  `page=2`, `trim=l b r t`, `keepaspectratio`, ...
- The flowing image formats are accepted:
  LaTeX .eps (and .mps)
  pdfLaTeX .pdf, .jpg, and .png

Figure: `graphicx` package

Including images

To include images use the `graphicx` package:

- Syntax: `\includegraphics[opts]{image}`
- `opts` modify the result, *e.g.*,
  `width=1cm`, `width=.95\columnwidth`, `height=.9\textheight`,
  `page=2`, `trim=l b r t`, `keepaspectratio`, …
- The flowing image formats are accepted:

  LATEX .eps (and .mps)
  pdfLATEX .pdf, .jpg, and .png

Tip: including images

- Drop file extensions, LATEX will itself pick the best file available
- Use `\graphicspath{{./fig/}{./images/}}` to specify where to look for images

Creating Graphical Content

- Use **vector graphics** whenever possible (.eps, .pdf);
  it looks best in print and is often a conference/journal requirement
- Some programs that can export vector graphics:
  - Adobe illustrator (http://www.adobe.com/products/illustrator.html)
  - Incscape (http://www.inkscape.org/en/)
  - LibreOffice Draw (http://www.libreoffice.org/discover/draw/)
  - MATLAB (http://www.mathworks.com)
  - yEd (http://www.yworks.com/en/products_yed_about.html)
- Figures can be created in LATEX specific code using, *e.g.*,
  PGF/TikZ, XyPic, MetaPost
  - Most suited for block diagrams and simple line-art
  - All but the most simple things has a steep learning curve

Exporting Figures From MATLAB to LATEX

- Print to an EPS file and convert the result to PDF
    - `>> print(fig_nr, '-depsc2', '-painters', 'myfile.eps')`
    - Use the package epstopdf to automatically convert from EPS to PDF figures
- Use a MATLAB package to produce a PDF directly, from MATLAB, *e.g.*, *E.g.*, export_fig

    (http://www.mathworks.com/matlabcentral/fileexchange/23629-export-fig)

- Use the MATLAB package Matfig2PGF to generate pgf code to do the figures in LATEX

    (http://www.mathworks.com/matlabcentral/fileexchange/12962-matfig2pgf)

Example: figures

Result
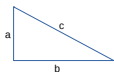
LaTeX input

```
\usepackage{graphicx}
\graphicspath{{./fig/}}

\begin{document}

This is how you insert a pdf image in a document
\begin{figure}[t]
  \centering
  \includegraphics[height=.25\textheight]
                  {notation}
  \caption{This sets the notation.}
  \label{fig:notaion}
\end{figure}

This is a \textsc{Matlab}-plot going here\dots

\begin{figure}[h]
  \centering
  \includegraphics[width=.45\textwidth]
                  {trigs}
  \caption{Trigonometric functions.}
  \label{fig:trigs}
\end{figure}

And this is how you insert a png.
\begin{figure}[b!]
  \centering
  \includegraphics[width=.25\linewidth]
                  {knuth}
  \caption{D. Knuth once again.}
  \label{fig:notaion}
\end{figure}
```

Tables

- Create a standard table float with \begin{table}...\end{table}
- table and figure work exactly the same except for the caption text
- Add a caption (above the table) using \caption{My caption.}
- List tables in a document using \listoftables

Tables: create the table material

- \begin{tabular}{colspec}
  colspec defines the content in each row in a table:

l	Left-aligned column
c	Centered column
r	Right-aligned column
p{width}	Multi-line column
|	Vertical line
@{decl}	Drop inter-column space and use decl instead

- Each row in the table is then build up from elements separated by &
- Lines are separated by \\
- Use \hline for a horizontal line
- \end{tabular}

Tables: advanced usage ⋆

Tabular extensions

- Use the `array` package for more column options
- For cells spanning several columns, use the `\multicolumn` command.
- For cells spanning several rows, use the `\multirow` command (requires the `multrow` package)
- Use the `dcolumn` package to align columns on the "decimal point"

Extensive (long) tables

- Several packages to automatically split extensive tables over several pages, *e.g.*,
  - `supertabular`
  - `longtable`
- Note: Multi-page tables are nontrivial, several reruns of LATEX might be needed, and still there might be unexpected results!

Example: table

Example

```
\begin{table}[t]
  \centering
  \caption{Trigonometric functions}
  \label{tab:trig}

  \begin{tabular}{>{$}c<{$}|>{$}c<{$}>{$}c<{$}>{$}c<{$}}
    \hline
    \alpha & \sin(\alpha) & \cos(\alpha) & \tan(\alpha)\\
    \hline
    0 & 0 & 0 & 0\\
    \frac{\pi}{6} & \frac{1}{2} &
    \frac{\sqrt{3}}{2} & \frac{1}{\sqrt{3}}\\
    \frac{\pi}{4} & \frac{1}{\sqrt{2}} &
    \frac{1}{\sqrt{2}} & 1\\
    \frac{\pi}{3} & \frac{\sqrt{3}}{2} &
    \frac{\sqrt{3}}{2} & \sqrt{3}\\
    \frac{\pi}{2} & 1 & 0 & \text{---}\\
    \pi & 0 & -1 & 0\\
    \frac{3\pi}{2} & -1 & 0 & \text{---}\\
    2\pi & 0 & 1 & 0\\
    \hline
  \end{tabular}
\end{table}
```

Result

Table 1: Trigonometric functions

α	$\sin(\alpha)$	$\cos(\alpha)$	$\tan(\alpha)$
0	0	0	0
$\frac{\pi}{6}$	$\frac{1}{2}$	$\frac{\sqrt{3}}{2}$	$\frac{1}{\sqrt{3}}$
$\frac{\pi}{4}$	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$	1
$\frac{\pi}{3}$	$\frac{\sqrt{3}}{2}$	$\frac{\sqrt{3}}{2}$	$\sqrt{3}$
$\frac{\pi}{2}$	1	0	—
π	0	-1	0
$\frac{3\pi}{2}$	-1	0	—
2π	0	1	0

Float Placement

- Use \begin{figure}[pos], where pos is a hint to LaTeX where to put the float.
- pos is a hint about where to put the float, it should be a combination of h, t, b, p, and !.
- The placement algorithm is roughly:
    1. h Try put the figure right here
    2. t Try at the top of this page
    3. b Try at the bottom of this page
    4. p Create a float-only page
  If this fails 2–3 are rerun on the next page
- Add ! to make LaTeX less picky about the placement

Float Placement

- Use `\begin{figure}[pos]`, where pos is a hint to LaTeX where to put the float.
- pos is a hint about where to put the float, it should be a combination of h, t, b, p, and !.
- The placement algorithm is roughly:
    1. h Try put the figure right here
    2. t Try at the top of this page
    3. b Try at the bottom of this page
    4. p Create a float-only page

    If this fails 2–3 are rerun on the next page
- Add ! to make LaTeX less picky about the placement

H and the `float` package

The `float` package adds the placement option H that makes the float go *Here* and stop float.
Use with care as the result is usually ugly with poor page utilization!

Miscellaneous Useful Packages ⋆

subfig The subfig packages provides the
\subfloat[Sub-caption]{content} command to provide
sub-division of figures, tables, and some other floats.

Miscellaneous Useful Packages ⋆

subfig The subfig packages provides the
\subfloat[Sub-caption]{content} command to provide
sub-division of figures, tables, and some other floats.

endfloats Some journals requires floats to be put very last in the
manuscript, with just a placement marker in the document.
This package helps you do with just an extra line, while still
editing the document with the floats inline.

Miscellaneous Useful Packages ⋆

subfig The subfig packages provides the
\subfloat[Sub-caption]{content} command to provide
sub-division of figures, tables, and some other floats.

endfloats Some journals requires floats to be put very last in the
manuscript, with just a placement marker in the document.
This package helps you do with just an extra line, while still
editing the document with the floats inline.

algorithms The algorithms packages provides environments which can
be used to specify algorithms.

Miscellaneous Useful Packages ⋆

subfig The subfig packages provides the
        \subfloat[Sub-caption]{content} command to provide
        sub-division of figures, tables, and some other floats.

endfloats Some journals requires floats to be put very last in the
           manuscript, with just a placement marker in the document.
           This package helps you do with just an extra line, while still
           editing the document with the floats inline.

algorithms The algorithms packages provides environments which can
            be used to specify algorithms.

listings This package provides an easy way to include source code in
          a document. It automatically handles special characters and
          provides syntax highlighting for most major languages.

Outline

Managing Citations

- Citation and bibliography handling is tightly integrated in LaTeX, but is handled in a separate step by BibTeX
- References are stored in a separate database (.bib file)
  - The .bib file is a pure text file, like all other LaTeX files, and is human readable
  - Each bib-entry contain information about a paper/article/... but no formatting information
  - The type of the reference is decides what information is needed
- LaTeX+BibTeX handles all formatting, using different bib-styles (harvard, author-year, numbers, ...)

Adding Citations: basics

- Indicate which .bib files to use:
    ```
    \bibliography{myrefs}
    ```
  This also indicates where the bibliography is inserted

Adding Citations: basics

- Indicate which .bib files to use:
    `\bibliography{myrefs}`
  This also indicates where the bibliography is inserted
- Add a line to specify what the citation style to use:
    `\bibliographystyle{plain}`
  Replace plain with the style you want/need, *e.g.*, most journals have
  their own style

Standard bibliography styles

plain	Produce [1] style citations, the bibliography is sorted alphabetically
usrt	Similar to plain but sorted in order of appearance
alpha	Produce [Hen02] style citations, otherwise like plain
abbrv	Similar to plain, but everything possible is abbreviated

Adding Citations: basics

- Indicate which .bib files to use:
  `\bibliography{myrefs}`
  This also indicates where the bibliography is inserted
- Add a line to specify what the citation style to use:
  `\bibliographystyle{plain}`
  Replace plain with the style you want/need, *e.g.*, most journals have their own style
- Start making citations in your text, using `\cite{tag}`

Standard bibliography styles

plain	Produce [1] style citations, the bibliography is sorted alphabetically
usrt	Similar to plain but sorted in order of appearance
alpha	Produce [Hen02] style citations, otherwise like plain
abbrv	Similar to plain, but everything possible is abbreviated

Adding Citations: basics

- Indicate which .bib files to use:
    `\bibliography{myrefs}`
  This also indicates where the bibliography is inserted
- Add a line to specify what the citation style to use:
    `\bibliographystyle{plain}`
  Replace plain with the style you want/need, *e.g.*, most journals have their own style
- Start making citations in your text, using `\cite{tag}`
- Run and (re)run BibTeX when asked to do so

Standard bibliography styles

plain	Produce [1] style citations, the bibliography is sorted alphabetically
usrt	Similar to plain but sorted in order of appearance
alpha	Produce [Hen02] style citations, otherwise like plain
abbrv	Similar to plain, but everything possible is abbreviated

Adding Citations: author-year citations with `natbib`

- Plain BibTeX lack ability to author-year (harvard) style citations, external packages are needed
- The `natbib` packages does this and much more
- `natbib` requires more detailed styles, *e.g.*, `plainnat`, `usrtnat`, or `abbrvnat`
- Citation style is determined by package options, *e.g.*, `numbers`, `authoryear`, `super`, `sort&compress`
- The set of citation commands is extended, *e.g.*,

  `\citet[Ch. 2]{jon90}` Textual citation:

  Jones et al. (1990, Ch. 2)

  `\citep[see][Ch. 2]{jon90}` Parenthetical citation:

  (see Jones et al., 1990, Ch. 2)

  `\citeauthor{jon90}` Authors: Jones et al.

  `\citeyear{jon90}` Year: 1990

The BibTeX File

- The bibliography database consist of bib-entries, one per reference
- Important bib-entry types and their required fields:
  - `article` (journal paper): `author`, `title`, `journal`, `year`
  - `inproceedings` (conference paper): `author`, `title`, `booktitle`, `year`
  - `book` (whole book) `author/editor`, `title`, `publisher`, `year`
- Enclose everything put pure numbers in { . . . }
- Separate authors with and!
- Look up the details or use an editor with BibTeX support to get auto-completion

A bib-entry template

```
@type{authorname:YYYY,
  author = {lname1, fname1 and lname2, Jr, fname2},
  title = {Title of Paper},
  year = 2012,
  ...
}
```

Example with Bibliography: input

LATEX input (numbers)

```
\usepackage[numbers, square, sort&compress]{natbib}

\begin{document}

This text is just to make sure to cite all references:
The book by \citet{kailathSH:2000}, the article
\citep*{hendebyKG:2010a}, and the paper
\citep{hendebyKG:2007}. If not cited, they will not
appear in the bibliography.

\bibliographystyle{plainnat}
```

LATEX input (author-year)

```
\usepackage[authoryear, round, sort&compress]{natbib}

\begin{document}

This text is just to make sure to cite all references:
The book by \citet{kailathSH:2000}, the article
\citep*{hendebyKG:2010a}, and the paper
\citep{hendebyKG:2007}. If not cited, they will not
appear in the bibliography.

\bibliographystyle{plainnat}
```

BibTEX input

```
@String{J_EURASIP_ASP = {{EURASIP} Journal on Applied
                          Signal Processing}}
@String{P_PH = {Prentice-Hall, Inc}}

@Book{kailathSH:2000,
    author =    {Kailath, Thomas and Sayed, Ali H.
                and Hassibi, Babak},
    title =     {Linear Estimation},
    publisher = P_PH,
    year =      2000,
    ISBN =      {0-13-022464-2}
}

@Article{hendebyKG:2010a,
    author =    {Hendeby, Gustaf and Karlsson, Rickard
                and Gustafsson, Fredrik},
    title =     {Particle Filtering: {T}he Need for Speed},
    journal =   J_EURASIP_ASP,
    year =      2010,
    pages =     {Article ID 181403}
}

@InProceedings{hendebyKG:2007,
    author =    {Hendeby, Gustaf and Karlsson, Rickard
                and Gustafsson, Fredrik},
    title =     {A New Formulation of the
                {R}ao-{B}lackwellized Particle Filter},
    booktitle = {{IEEE} Workshop on Statistical
                Signal Processing},
    year =      2007,
    month =     aug,
    address =   {Madison, WA},
}
```

Example with Bibliography: input

LATEX input (numbers)

```
\usepackage[numbers, square, sort&compress]{natbib}

\begin{document}

This text is just to make sure to cite all references:
The book by \citet{kailathSH:2000}, the article
\citep*{hendebyKG:2010a}, and the paper
\citep{hendebyKG:2007}. If not cited, they will not
appear in the bibliography.

\bibliographystyle{plainnat}
```

LATEX input (author-year)

```
\usepackage[authoryear, round, sort&compress]{natbib}

\begin{document}

This text is just to make sure to cite all references:
The book by \citet{kailathSH:2000}, the article
\citep*{hendebyKG:2010a}, and the paper
\citep{hendebyKG:2007}. If not cited, they will not
appear in the bibliography.

\bibliographystyle{plainnat}
```

BibTEX input

```
@String{J_EURASIP_ASP = {{EURASIP} Journal on Applied
                         Signal Processing}}
@String{P_PH = {Prentice-Hall, Inc}}

@Book{kailathSH:2000,
    author =    {Kailath, Thomas and Sayed, Ali H.
                 and Hassibi, Babak},
    title =     {Linear Estimation},
    publisher = P_PH,
    year =      2000,
    ISBN =      {0-13-022464-2}
}

@Article{hendebyKG:2010a,
    author =    {Hendeby, Gustaf and Karlsson, Rickard
                 and Gustafsson, Fredrik},
    title =     {Particle Filtering: {T}he Need for Speed},
    journal =   J_EURASIP_ASP,
    year =      2010,
    pages =     {Article ID 181403}
}

@InProceedings{hendebyKG:2007,
    author =    {Hendeby, Gustaf and Karlsson, Rickard
                 and Gustafsson, Fredrik},
    title =     {A New Formulation of the
                 {R}ao-{B}lackwellized Particle Filter},
    booktitle = {{IEEE} Workshop on Statistical
                 Signal Processing},
    year =      2007,
    month =     aug,
    address =   {Madison, WA},
}
```
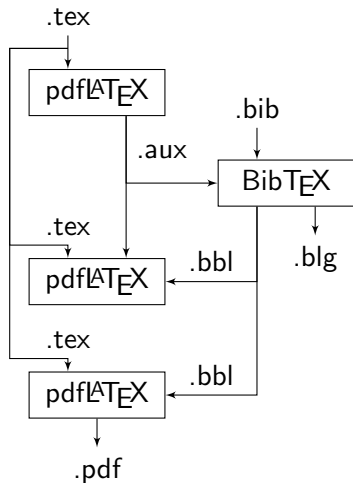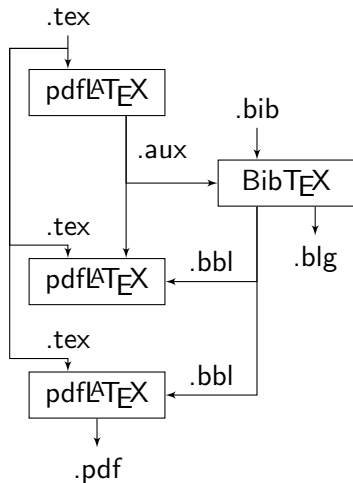
- **Note:** Only a single line differs between the numbered and the author-year example!

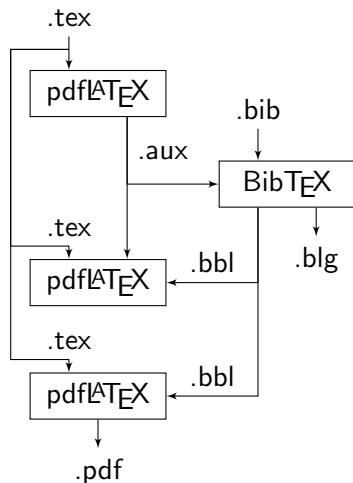# Example with Bibliography: compile

.tex

pdfLATEX

.bib

.aux

BibTEX

.tex

pdfLATEX   .bbl   .blg

.tex

pdfLATEX   .bbl

.pdf

1. Compile your document with LATEX:
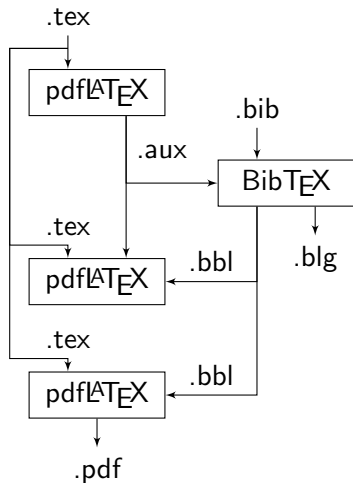 $ pdflatex simple.tex

Example with Bibliography: compile

1. Compile your document with LaTeX:
   $ pdflatex simple.tex
2. Compile your document with BibTeX:
   $ bibtex simple.aux

Example with Bibliography: compile

.tex

pdfLaTeX

.aux

.bib

BibTeX

.tex

pdfLaTeX   .bbl   .blg

.tex

pdfLaTeX   .bbl

.pdf

1. Compile your document with LaTeX:
   $ pdflatex simple.tex
2. Compile your document with BibTeX:
   $ bibtex simple.aux
3. Compile your document with LaTeX:
   $ pdflatex simple.tex

Example with Bibliography: compile

1. Compile your document with LaTeX:
   $ pdflatex simple.tex
2. Compile your document with BibTeX:
   $ bibtex simple.aux
3. Compile your document with LaTeX:
   $ pdflatex simple.tex
4. Repeat 3 until all necessary temporary are in place

A Simple Example: result

Result (numbers)

This text is just to make sure to cite all references: The book by Kailath et al. [3], the article [2], and the paper [1]. If not cited, they will not appear in the bibliography.

References

[1] Gustaf Hendeby, Rickard Karlsson, and Fredrik Gustafsson. A new formulation of the Rao-Blackwellized particle filter. In *IEEE Workshop on Statistical Signal Processing*, Madison, WA, August 2007.

[2] Gustaf Hendeby, Rickard Karlsson, and Fredrik Gustafsson. Particle filtering: The need for speed. *EURASIP Journal on Applied Signal Processing*, page Article ID 181403, 2010.

[3] Thomas Kailath, Ali H. Sayed, and Babak Hassibi. *Linear Estimation*. Prentice-Hall, Inc, 2000. ISBN 0-13-022464-2.

Result (author-year)

This text is just to make sure to cite all references: The book by Kailath et al. (2000), the article (Hendeby, Karlsson, and Gustafsson, 2010), and the paper (Hendeby et al., 2007). If not cited, they will not appear in the bibliography.

References

Gustaf Hendeby, Rickard Karlsson, and Fredrik Gustafsson. A new formulation of the Rao-Blackwellized particle filter. In *IEEE Workshop on Statistical Signal Processing*, Madison, WA, August 2007.

Gustaf Hendeby, Rickard Karlsson, and Fredrik Gustafsson. Particle filtering: The need for speed. *EURASIP Journal on Applied Signal Processing*, page Article ID 181403, 2010.

Thomas Kailath, Ali H. Sayed, and Babak Hassibi. *Linear Estimation*. Prentice-Hall, Inc, 2000. ISBN 0-13-022464-2.

A Simple Example: result

Result (numbers)

This text is just to make sure to cite all references: The book by Kailath et al. [3], the article [2], and the paper [1]. If not cited, they will not appear in the bibliography.

References

[1] Gustaf Hendeby, Rickard Karlsson, and Fredrik Gustafsson. A new formulation of the Rao-Blackwellized particle filter. In *IEEE Workshop on Statistical Signal Processing*, Madison, WA, August 2007.

[2] Gustaf Hendeby, Rickard Karlsson, and Fredrik Gustafsson. Particle filtering: The need for speed. *EURASIP Journal on Applied Signal Processing*, page Article ID 181403, 2010.

[3] Thomas Kailath, Ali H. Sayed, and Babak Hassibi. *Linear Estimation*. Prentice-Hall, Inc, 2000. ISBN 0-13-022464-2.

Result (author-year)

This text is just to make sure to cite all references: The book by Kailath et al. (2000), the article (Hendeby, Karlsson, and Gustafsson, 2010), and the paper (Hendeby et al., 2007). If not cited, they will not appear in the bibliography.

References

Gustaf Hendeby, Rickard Karlsson, and Fredrik Gustafsson. A new formulation of the Rao-Blackwellized particle filter. In *IEEE Workshop on Statistical Signal Processing*, Madison, WA, August 2007.

Gustaf Hendeby, Rickard Karlsson, and Fredrik Gustafsson. Particle filtering: The need for speed. *EURASIP Journal on Applied Signal Processing*, page Article ID 181403, 2010.

Thomas Kailath, Ali H. Sayed, and Babak Hassibi. *Linear Estimation*. Prentice-Hall, Inc, 2000. ISBN 0-13-022464-2.

Note on BibTeX usage

- The citations style is changed by how the natbib package is imported, without **any** changes to the bibliography database
- The same BibTeX database will do for all that you write, saving tons of time and helps to provide correct and consistent references

Advanced BibTeX Usage

- Titles are automatically capitalized according to style guidelines, avoid unwanted lower-casing by enclosing upper case letters in { ... }
- Make use of predefined macros:
  - All standard styles define months with their 3-letter abbreviation
  - IEEE defines all their journals and magazines in `IEEEfull.bib` and `IEEEabrv.bib`; *e.g.*, `IEEE_J_SP` (IEEE Trans. Signal Process.)
- Separate bibliographies per chapter or units can be realized using, *e.g.*, `natbib`, `chapterbib`, or `bibunits`
- Insert bibliography entries inline using `bibentry`

Advanced BibTeX Usage

- Titles are automatically capitalized according to style guidelines, avoid unwanted lower-casing by enclosing upper case letters in { . . . }
- Make use of predefined macros:
  - All standard styles define months with their 3-letter abbreviation
  - IEEE defines all their journals and magazines in `IEEEfull.bib` and `IEEEabrv.bib`; *e.g.*, `IEEE_J_SP` (IEEE Trans. Signal Process.)
- Separate bibliographies per chapter or units can be realized using, *e.g.*, `natbib`, `chapterbib`, or `bibunits`
- Insert bibliography entries inline using `bibentry`

The `biblatex` package and Biber (the future?) ⋆

An up and coming alternative, that solves most of the issues above, is the `biblatex` package in combination with Biber (replacing BibTeX). Unfortunately `biblatex` seems incompatible with many conference and journal templates around, and requires other changes to your LaTeX file.

Organizing Your References

JabRef

JabRef is a program that helps organize your references and associated PDFs. Selected entries can then be exported in, *e.g.*, BibTEX format.
(`http://jabref.sourceforge.net/`)

Organizing Your References

JabRef

JabRef is a program that helps organize your references and associated PDFs. Selected entries can then be exported in, *e.g.*, BibTeX format.

(`http://jabref.sourceforge.net/`)

Mendeley

Mendeley is an interesting option for online handling of PDF files and references, much like JabRef but online. The Mendeley content can be synchronized in between different computers.

(`http://www.mendeley.com/`)

Organizing Your References

JabRef

JabRef is a program that helps organize your references and associated PDFs. Selected entries can then be exported in, *e.g.*, BibTeX format.

(`http://jabref.sourceforge.net/`)

Mendeley

Mendeley is an interesting option for online handling of PDF files and references, much like JabRef but online. The Mendeley content can be synchronized in between different computers.

(`http://www.mendeley.com/`)

EndNote

EndNote is a reference handling system recommend by the library at LiU. A pro is that is well integrated with the online full-text databases provided at LiU; a con is that it only seems possible export information to a BibTeX friendly format with **with "some" work**.
(`http://www.bibl.liu.se/citera-och-referera/endnote?l=en`)

Outline

Why Custom Macros and Environments?

Main advantages:

- Reduce the need for excessive typing by reducing long expressions to a short macros or environments
- Hide unnecessary complexity from less experience users
- Help obtain a consistent notation throughout a document or several (using style files) by providing macros for common notions
- Make it easy and error free to change the notation or look throughout a whole document

Macros: basics

- LaTeX can be extended with your own macros and environments
- The general syntax is:
  1. `\newcommand{\cmd}{tex-code}`
  2. `\newcommand{\cmd}[n]{tex-code-using-#1-#2...}`
  3. `\newcommand{\cmd}[n][default]{tex-code-using-#1-#2...}`

Macros: basics

- LATEX can be extended with your own macros and environments
- The general syntax is:
  1. `\newcommand{\cmd}{tex-code}`
  2. `\newcommand{\cmd}[n]{tex-code-using-#1-#2...}`
  3. `\newcommand{\cmd}[n][default]{tex-code-using-#1-#2...}`
- Usage (assuming n is 3 where applicable):
  1. `\cmd` (Will usually consume the following spaces!)
  2. `\cmd{foo}{bar}{yada}` where #1 is foo, #2 is bar, . . .
  3. `\cmd[foo]{bar}{yada}` where #1 is foo, #2 is bar, . . .
     or using the default value
     `\cmd{bar}{yada}` where #1 is default, #2 is bar, . . .

Macros: basics

- LaTeX can be extended with your own macros and environments
- The general syntax is:
  1. `\newcommand{\cmd}{tex-code}`
  2. `\newcommand{\cmd}[n]{tex-code-using-#1-#2...}`
  3. `\newcommand{\cmd}[n][default]{tex-code-using-#1-#2...}`
- Usage (assuming `n` is 3 where applicable):
  1. `\cmd` (Will usually consume the following spaces!)
  2. `\cmd{foo}{bar}{yada}` where #1 is `foo`, #2 is `bar`, ...
  3. `\cmd[foo]{bar}{yada}` where #1 is `foo`, #2 is `bar`, ...
     or using the default value
     `\cmd{bar}{yada}` where #1 is `default`, #2 is `bar`, ...
- Use `\newcommand*` to allow only a single paragraph in each argument (when appropriate), this simplifies debugging

Macros: basics

- LaTeX can be extended with your own macros and environments
- The general syntax is:
    1. `\newcommand{\cmd}{tex-code}`
    2. `\newcommand{\cmd}[n]{tex-code-using-#1-#2...}`
    3. `\newcommand{\cmd}[n][default]{tex-code-using-#1-#2...}`
- Usage (assuming n is 3 where applicable):
    1. `\cmd` (Will usually consume the following spaces!)
    2. `\cmd{foo}{bar}{yada}` where #1 is foo, #2 is bar, ...
    3. `\cmd[foo]{bar}{yada}` where #1 is foo, #2 is bar, ...
       or using the default value
       `\cmd{bar}{yada}` where #1 is default, #2 is bar, ...
- Use `\newcommand*` to allow only a single paragraph in each argument (when appropriate), this simplifies debugging
- An already defined macro can be redefined using `\renewcommand`

Macros: hints ⋆

Ensuring math mode

When writing macros with mathematical features for use in both text and math mode, use the \ensuremath macro, *e.g.*,

    \newcommand*{\allX}{\ensuremath{\mathbb{X}}}

\allX ($\mathbb{X}$) can now be used in both and text and math mode

Macros: hints ⋆

Ensuring math mode

When writing macros with mathematical features for use in both text and math mode, use the \ensuremath macro, *e.g.*,

    \newcommand*{\allX}{\ensuremath{\mathbb{X}}}

\allX (𝕏) can now be used in both and text and math mode

Avoid consuming trailing spaces after macros

The xspace package offers the \xspace macro that can be used last a macro to avoid that it consumes the following white space characters.

Macros: hints ⋆

Ensuring math mode

When writing macros with mathematical features for use in both text and math mode, use the \ensuremath macro, *e.g.*,

   `\newcommand*{\allX}{\ensuremath{\mathbb{X}}}`

\allX ($\mathbb{X}$) can now be used in both and text and math mode

Avoid consuming trailing spaces after macros

The xspace package offers the \xspace macro that can be used last a macro to avoid that it consumes the following white space characters.

Avoid \def

The pure TEX equivalent of \newcommand is \def, and it is suggested by many online resources. Do not use it! It sidesteps LATEXsafety mechanisms, and can lead to difficult to find errors. Use \(re)newcommand instead!

Macros: mathematical context

- Define new functions using `\DeclareMathOperator` or `\DeclareMathOperator*`
- The star form puts sub- and superscripts below and above the operator
- Function example, normal sub- and superscripts:
  `\DeclareMathOperator{\sinc}{sinc}`
  Now `\sinc^{-1}(x)` yields

$$\text{sinc}^{-1}(x)$$

- Operator example, sub- and superscripts below and above
  `\DeclareMathOperator*{\argmin}{arg\,min}`
  Now `\argmin_x f(x)` yields

$$\underset{x}{\arg\min} f(x)$$

Environments ⋆

- Theorems, definitions, and other theorem-like environments are best obtained via the `amsthm` or the `ntheorem` packages
  - Predefined theorem-like environments
  - Ability to create new theorems, at a rather high conceptual level
  - Predefined environments can be redefined
- If necessary, you can define your own environments using `\newenvironment`
  - Useful for examples, specialized lists, etc
  - The `list` and `trivlist` environments make good starting points for new environments

Style Files

- If you see that the same macros and settings are used over and over in many manuscripts, consider creating a style-file
- A style file (.sty) is a package that can be included with \usepackage
  - Style files starts with \ProvidesPackage{name} where name is the name of the file without the extension .sty
  - Then list put the LATEX code you want executed in the preamble (it may not produce text output)
- Put the file where LATEX looks for files, and reuse whenever you are writing something new
- For information about where LATEX looks for files, consult the documentation for your LATEX installation and

  http://tug.ctan.org/tds/tds.pdf

Example: style file

mydefs.sty

```
\ProvidesPackage{mydefs}

\usepackage{xspace}
\usepackage{amsmath, amssymb, textcomp}

%% Standard abbreviations (latin -> emph)
\newcommand*\eg{\emph{e.g.}\@\xspace}
\newcommand*\ie{\emph{i.e.}\@\xspace}
\newcommand*\viz{\emph{viz.}\@\xspace}

\newcommand*\matlab{\mbox{\textsc{Matlab}%
    {\scriptsize$^{\text{\textregistered}}$}}\xspace}

%% Technical abbreviations
\newcommand*\gpu{\textsc{gpu}\xspace}
\newcommand*\hmm{\textsc{hmm}\xspace}
\newcommand*\ieee{\textsc{ieee}\xspace}
\newcommand*\mmse{\textsc{mmse}\xspace}
\newcommand*\mpf{\textsc{mpf}\xspace}
\newcommand*\mmse{\textsc{mmse}\xspace}

%% Math operators
\DeclareMathOperator*\argmax{arg\,max}
\DeclareMathOperator\bigO{\mathcal{O}}
\DeclareMathOperator\diag{diag}
\newcommand*\KLinfo{\mathcal{I}^{\textsc{KL}}\!}
\newcommand*\pseudoinverse{\dagger}
\DeclareMathOperator\var{\mathsf{var}}

%% Useful stuff
\newsavebox{\foo@lastonrow}
\newcommand*\lastonrow[1]{%
    \hspace{\fill}\nolinebreak[1]^\hspace*{\fill}{#1}}%

\newcommand*\qedsym{\ensuremath{\blacksquare}}
\newcommand*\qefsym{\ensuremath{\square}}
\newcommand*\qed{\lastonrow{\qedsym}}
\newcommand*\qef{\lastonrow{\qefsym}}
```

Example: style file

mydefs.sty

```latex
\ProvidesPackage{mydefs}

\usepackage{xspace}
\usepackage{amsmath, amssymb, textcomp}

%% Standard abbreviations (latin -> emph)
\newcommand*\eg{\emph{e.g.}\@\xspace}
\newcommand*\ie{\emph{i.e.}\@\xspace}
\newcommand*\viz{\emph{viz.}\@\xspace}

\newcommand*\matlab{\mbox{\textsc{Matlab}%
    {\scriptsize$^{\text{\textregistered}}$}}\xspace}

%% Technical abbreviations
\newcommand*\gpu{\textsc{gpu}\xspace}
\newcommand*\hmm{\textsc{hmm}\xspace}
\newcommand*\ieee{\textsc{ieee}\xspace}
\newcommand*\mmse{\textsc{mmse}\xspace}
\newcommand*\mpf{\textsc{mpf}\xspace}
\newcommand*\mmse{\textsc{mmse}\xspace}

%% Math operators
\DeclareMathOperator*\argmax{arg\,max}
\DeclareMathOperator\bigO{\mathcal{O}}
\DeclareMathOperator\diag{diag}
\newcommand*\KLinfo{\mathcal{I}^{\textsc{KL}}\!}
\newcommand*\pseudoinverse{\dagger}
\DeclareMathOperator\var{\mathsf{var}}

%% Useful stuff
\newsavebox{\foo@lastonrow}
\newcommand*\lastonrow[1]{%
    \hspace{\fill}\nolinebreak[1]\hspace*{\fill}{#1}}%

\newcommand*\qedsym{\ensuremath{\blacksquare}}
\newcommand*\qefsym{\ensuremath{\square}}
\newcommand*\qed{\lastonrow{\qedsym}}
\newcommand*\qef{\lastonrow{\qefsym}}
```

mydefs.sty (robust version)

```latex
\ProvidesPackage{mydefs}

\RequirePackage{xspace}
\RequirePackage{amsmath, amssymb, textcomp}

%% Standard abbreviations (latin -> emph)
\DeclareRobustCommand*\eg{\emph{e.g.}\@\xspace}
\DeclareRobustCommand*\ie{\emph{i.e.}\@\xspace}
\DeclareRobustCommand*\viz{\emph{viz.}\@\xspace}

\DeclareRobustCommand*\matlab{\mbox{\textsc{Matlab}%
    {\scriptsize$^{\text{\textregistered}}$}}\xspace}

%% Technical abbreviations
\DeclareRobustCommand*\gpu{\textsc{gpu}\xspace}
\DeclareRobustCommand*\hmm{\textsc{hmm}\xspace}
\DeclareRobustCommand*\ieee{\textsc{ieee}\xspace}
\DeclareRobustCommand*\mmse{\textsc{mmse}\xspace}
\DeclareRobustCommand*\mpf{\textsc{mpf}\xspace}
\DeclareRobustCommand*\mmse{\textsc{mmse}\xspace}

%% Math operators
\DeclareMathOperator*\argmax{arg\,max}
\DeclareMathOperator\bigO{\mathcal{O}}
\DeclareMathOperator\diag{diag}
\DeclareRobustCommand*\KLinfo{\mathcal{I}^{\textsc{KL}}\!}
\DeclareRobustCommand*\pseudoinverse{\dagger}
\DeclareMathOperator\var{\mathsf{var}}

%% Useful stuff
\newsavebox{\foo@lastonrow}
\DeclareRobustCommand\lastonrow[1]{%
    \hspace{\fill}\nolinebreak[1]\hspace*{\fill}{#1}}%

\DeclareRobustCommand*\qedsym{\ensuremath{\blacksquare}}
\DeclareRobustCommand*\qefsym{\ensuremath{\square}}
\providecommand*\qed{\lastonrow{\qedsym}}
\providecommand*\qef{\lastonrow{\qefsym}}
```

Debugging: *very short* introduction

- Errors and Warnings
    - All errors start with a '!'
    - The number line indicates where the error occurred, *e.g.*, 1.47 indicates an error on line 47
    - Note: errors resulting from forgetting a } or \end{...} are usually reported too late
- Useful information in the log file
    - [n] indicates that page n has been written out, this is useful to find strange errors; however, note that there can be a delay of a 1–2 pages compared to which code is currently processed
    - LaTeX puts "(file_name" in the log when it starts processing a file, and ")" when it stops, this helps finding the offending file
- See Appendix B of "The LaTeX Companion" for quite complete handling of how to deal with errors and warnings

Common Errors

! Missing $ inserted

Something that can only exist in math mode was found in text mode.

Common Errors

! Missing $ inserted

Something that can only exist in math mode was found in text mode.

! Undefined control sequence.
l.6 \dtae
{December 2004}

\dtae is not an known macro. The problem is the last thing printed on the middle line.

Common Errors

```
!  Missing $ inserted
```

Something that can only exist in math mode was found in text mode.

```
!  Undefined control sequence.
l.6 \dtae
{December 2004}
```

\dtae is not an known macro. The problem is the last thing printed on the middle line.

```
Runaway argument?
{December 2004 \maketitle
!  Paragraph ended before \date was complete.
<to be read again>
\par
l.8
```

An argument for a macro was not appropriately closed. This was noticed when the current paragraph ended before the closing '}' was found.

Index Generation ⋆

- As with citations, LaTeX needs an external to provide an index
- Add \makeindex in the preamble
- Mark index terms in the document using the \index command
- Add \printindex to make the index appear
- Run LaTeX followed by makeindex (or the more modern xindy) on the generated .idx file, rerun LaTeX again to produce the list of index terms.
- For more information, see Ch. 11 of "The LaTeX Companion"

Various Useful Packages ⋆

hyperref With this package, references in your document become clickable in PDF files.

fancyhdr The `fancyhdr` package allows you to easily adjust the headers and footers of your document.

geometry This package can be used to change the page layout (size, margins, etc) in an easy and intuitive way.

ifthen, calc Provide useful functionality for more advanced macros and environments.

Outline

Learn More: books

- LaTeX: A Document Preparation System. 2 ed., Leslie Lamport. 1994.
  Basic introduction to LaTeX by the creator

- The LaTeX Companion. 2 ed. Michel Goossens, Sebastian Rahtz, Frank Mittelbach, 2006.
  The LaTeX reference

- The LaTeX Graphics companionm 2 ed. Michel Goossens, Frank Mittelbach, Sebastian Rahtz, Denis Roegel, Herbert Voß. 2008.
  A complete reference on LaTeX graphics
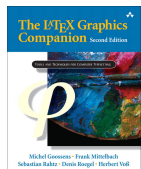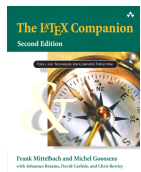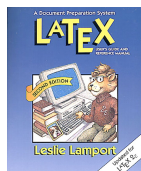
Learn More: internet resources

Documentation:

- The Not So Short Introduction to LaTeX 2_ε— Or LaTeX 2_ε in 157 minutes, Tobias Oetiker, et al. (http://ctan.uib.no/info/lshort/english/lshort.pdf). Great first introduction

- TeX Frequently Asked Questions on the Web, (http://www.tex.ac.uk/cgi-bin/texfaq2html?introduction=yes)

- LaTeX Wikibooks, (http://en.wikibooks.org/wiki/LaTeX)

- Google (but make sure to have parental mode activated :) Most likely what you want to do have already been done

- TeX Users Group web site (http://tug.org/)

Packages:

- TeXUsers Group web site (http://tug.org/ctan.html)