Word processing
LaTeX
Table of contents

1 Word processing and typesetting (Outline)

- authoring and typesetting
- vector graphics versus bitmaps
- LaTeX (=LaTeX = latex)
  - ▷ compiling documents
  - ▷ structure
  - ▷ lists
  - ▷ references
  - ▷ equations
  - ▷ figures
  - ▷ floats
  - ▷ other information
  - ▷ pointers to more information.

We use this symbol → Section X in LaTeX Tutorial to point to section X in "The not so short introduction to LaTeX" which can be found on the course web site (pdf).

Creating documents

In an ideal world: → Section 1.3 in LaTeX Tutorial

- The author
  - ▷ writes the manuscript
  - ▷ divides it into chapters, sections, subsections
  - ▷ indicates in manuscript where a new section starts

  → concentrates on content and structure of document

- The typesetter
  - ▷ typesets the document, *i.e.* reads manuscript and layouts the text to emphasise sections, subsections,. . .
  - ▷ assigns numbers to sections

  → concentrates on (consistent) layout of document

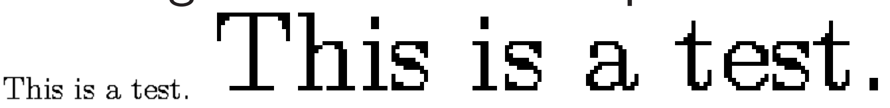
Type setting

- Past: separation of
  - ▷ writing manuscript and
  - ▷ typesetting it
- Present:
  - ▷ journals, books: separate typesetting
  - ▷ “small” documents: author and typesetter is the same person
- Future:
  - ▷ ?
- Software today mainly
  - ▷ MicroSoft Word
  - ▷ LaTeX (latex, LaTeX)
  - ▷ Corel WordPerfect
  - ▷ Open Source:
    - – Open Office
    - – Abiword (`http://www.abisource.com/`)
    - – . . .

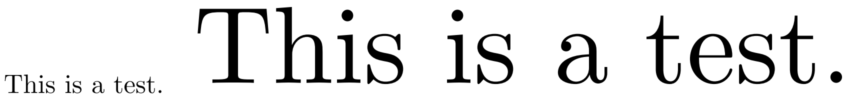
Bitmaps and vector-graphics

Different ways of storing/transmitting graphics (including text)

- Bitmaps
  - ▷ matrix of small points, for example 360 dots per inch (dpi) in each direction
  - ▷ store points individually
  - ▷ ideal for photographic images
  - ▷ formats: bmp, tif, gif, jpg, png, . . .
  - ▷ zooming reveals individual pixels:



- Vector graphics
  - ▷ store positions and shapes of graphical elements with mathematical formulae,
  - ▷ for example store a straight line as a vector from starting point to end point
  - ▷ ideal for text and schematic diagrams
  - ▷ formats: pdf (≈ compressed PostScript), ps, svg, cad-program data, . . .
  - ▷ zooming does not affect quality:



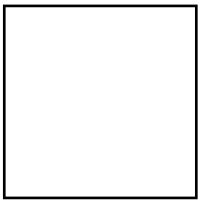
PostScript (ps)*

- “page description language”
- introduced in 1985
- describe pages in a device independent fashion (similar to vector graphics)
- Example: (`psbox.ps`)

```
%!
%% Draws a one square inch box and inch in from the bottom left

/inch 72 mul def      % Convert inches->points (1/72 inch)

newpath                 % Start a new path
1 inch 1 inch moveto    % an inch in from the lower left
2 inch 1 inch lineto    % bottom side
2 inch 2 inch lineto    % right side
1 inch 2 inch lineto    % top side
closepath               % Automatically add left side to close path
stroke                  % Draw the box on the paper
showpage                % We’re done... eject the page
```

produces: 

- Postscript viewers (“ghostscript” and “ghostview”) are freely available (http://www.cs.wisc.edu/~ghost/)

- PostScript is the dominant print-job description language on UNIX/Linux + Mac

- Better printers understand PostScript (for example most HP LaserJets)

- Can use software (ghostscript) to print PostScript files on other printers (on Windows, on UNIX/Linux and Mac, this is all working automatically)

- PostScript is used in book and journal printing industry

- pdf is a compressed PostScript (in 1st approximation) and therefore

- can easily convert PostScript to pdf (these pdf-lecture notes are converted from a PostScript file)

Microsoft Word (MS Word)

- commercial product (Office Standard Editition 2003 (Excel, Outlook, Word, Powerpoint) costs £369.99 (checked on 15/11/2006))
- meant to be “WYSIWYG” (What You See Is What You Get)
- writing and typesetting combined
- advantages
  - ▷ text and final layout immediately visible
  - ▷ widely used and available
- disadvantages
  - ▷ difficulty handling large files
  - ▷ layout changes depending on printer driver (*i.e.* no machine-independent format).
  - ▷ formulae not well laid out, difficult to enter
  - ▷ sometimes unstable (crashes)
  - ▷ need MS Word to open documents from others

LaTeX

- free product (can download from `www.latex-project.org`)
- program that compiles "text-source" into graphical document
- commonly used for books, journals, etc
- writing and typesetting are separated
- advantages
  - ▷ layout is printer-independent (device-independent)
  - ▷ final output is PostScript (or pdf) file: don't need LaTeX to read/print it
  - ▷ formulae are easy to enter and look good
  - ▷ very stable: no problems with huge documents
- disadvantages
  - ▷ Not "WYSIWYG": text and layout not immediately visible
  - ▷ Cannot use mouse to drag figures into place
- Good for
  - ▷ long documents
  - ▷ documents with formulae, software documentation
  - ▷ any scientific, mathematical, technical documents

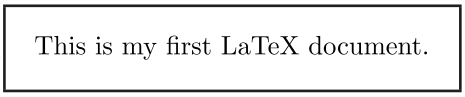
So what is this LaTeX?

- Pronounciation of LaTeX is Ley-tech with ch as in scottish “Loch” because last letter is χ (chi). Usually (in the UK) “Latec”.

- Our first LaTeX document with name `example1.tex`:

```
\documentclass[a4paper]{article}
\begin{document}
This is my first LaTeX document.
\end{document}
```

  Output: `example1.ps`



This is my first LaTeX document.

- How to get from `example1.tex` to output:
  - ▷ Create a dvi-file:

    `latex example1.tex`

    or on Windows machines: `latex.exe example1.tex` (or click on icon)
  - ▷ This creates `example.dvi` (dvi stands for DeVice Independent) which can be
    - – displayed on the computer screen (using a special dvi-viewer).
  - ▷ Then convert the dvi-file to PostScript file and
    - – print it or
    - – display it.

    Can also create pdf files straight away. See instructions for latex laboratory session.

dvi, ps and pdf*

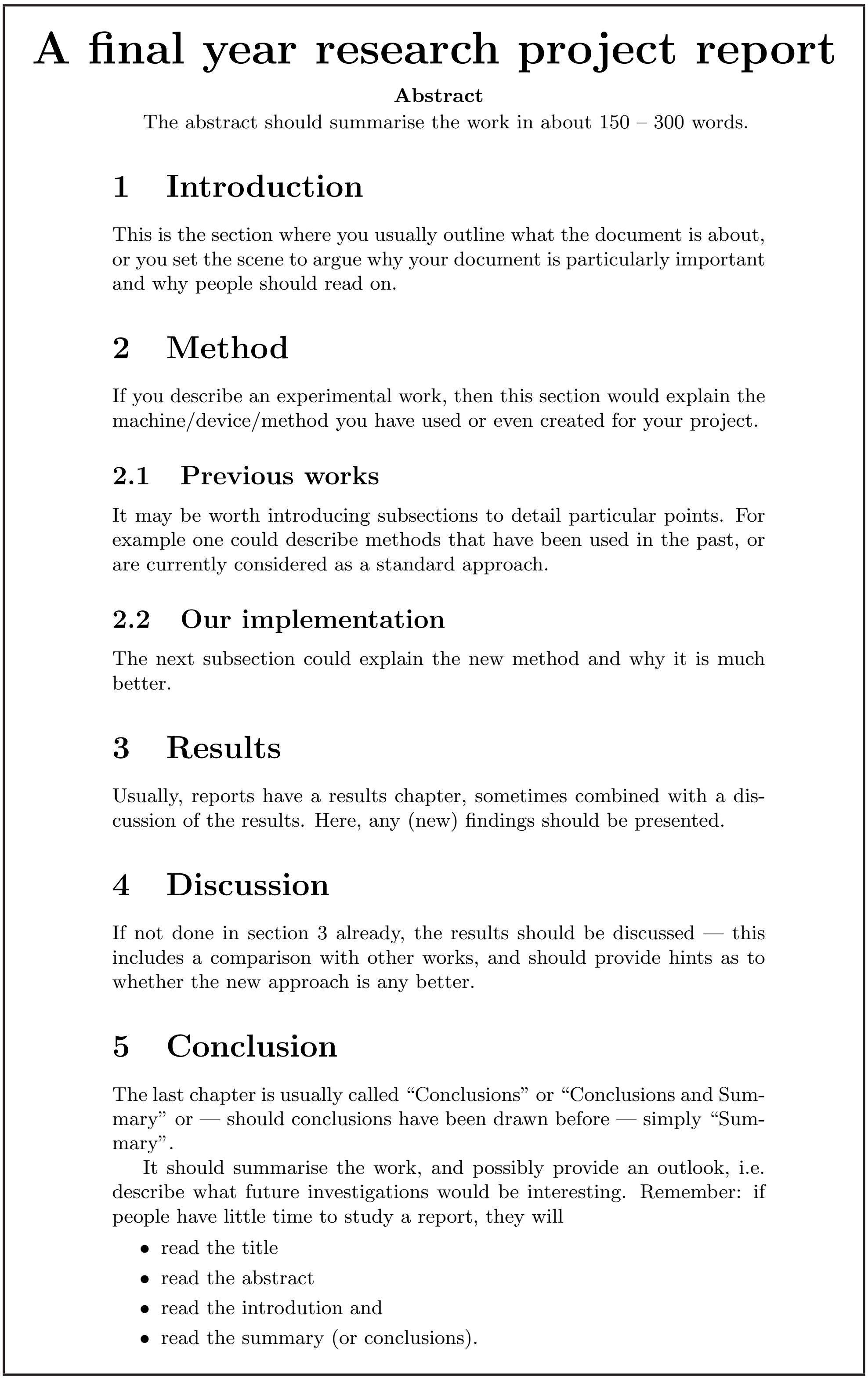
Why are there 3 device independent page description languages?

dvi: Invented as the device independent format for TEX(the predecessor of LATEX) in the early 1980s

ps: Invented after dvi. Everything dvi-files can, PostScript can do, too.

pdf: Even more recent, basically a compressed PostScript-format (plus new features). Everything PostScript can do, pdf files can do, too.

Structuring a LaTeX document: output



A final year research project report

Abstract

The abstract should summarise the work in about 150 – 300 words.

1 Introduction

This is the section where you usually outline what the document is about, or you set the scene to argue why your document is particularly important and why people should read on.

2 Method

If you describe an experimental work, then this section would explain the machine/device/method you have used or even created for your project.

2.1 Previous works

It may be worth introducing subsections to detail particular points. For example one could describe methods that have been used in the past, or are currently considered as a standard approach.

2.2 Our implementation

The next subsection could explain the new method and why it is much better.

3 Results

Usually, reports have a results chapter, sometimes combined with a discussion of the results. Here, any (new) findings should be presented.

4 Discussion

If not done in section 3 already, the results should be discussed — this includes a comparison with other works, and should provide hints as to whether the new approach is any better.

5 Conclusion

The last chapter is usually called “Conclusions” or “Conclusions and Summary” or — should conclusions have been drawn before — simply “Summary”.

It should summarise the work, and possibly provide an outlook, i.e. describe what future investigations would be interesting. Remember: if people have little time to study a report, they will

- read the title
- read the abstract
- read the introdution and
- read the summary (or conclusions).

Structuring a LaTeX document: input

```
\documentclass[a4paper]{article}
\begin{document}

\centerline{
\textbf{
\huge A final year research project report
} %end textbf
} %end centerline

\abstract{The abstract should summarise the work in about 150 -- 300 words.}

\section{Introduction}
This is the section where you usually outline what the document is about, or you set the

\section{Method}
If you describe an experimental work, then this section would explain the machine/device

\subsection{Previous works}
It may be worth introducing subsections to detail particular points. For example one cou

\subsection{Our implementation}
The next subsection could explain the new method and why it is much better.

\section{Results}\label{sec:results}
Usually, reports have a results chapter, sometimes combined with a discussion of the res

\section{Discussion}
If not done in section \ref{sec:results} already, the results should be discussed --- th

\section{Conclusion}
The last chapter is usually called ‘‘Conclusions’’ or ‘‘Conclusions and Summary’’ or ---

It should summarise the work, and possibly provide an outlook, i.e. describe what future
\begin{itemize}
\item read the title
\item read the abstract
\item read the introduction and
\item read the summary (or conclusions).
\end{itemize}
\end{document}
```

The syntactical outline of the document

→ Section 1.6 in LaTeX Tutorial

- Each LaTeX2ε file (this is the current version) should start with

  ```
  \documentclass[OPTIONS]{CLASS}
  ```

- useful `OPTIONS` are

  ▷ `a4paper` use A4 rather than U.S. letter
  ▷ `10pt` set default font size to 10 points
  ▷ `11pt` set default font size to 11 points
  ▷ `12pt` set default font size to 12 points
  ▷ `twocolumn` layout text with 2 columns

  Separate `OPTIONS` with a comma, for example:

  ```
  \documentclass[a4paper,12pt]{article}
  ```

- useful `CLASS`es are

  ▷ `article`, with sections, subsections, subsubsections
  ▷ `report` as article plus chapters
  ▷ `book` as report plus parts

  There are a few more differences between books, reports and articles (for example books are double-sided).

- The beginning of the document has to be indicated by

  ```
  \begin{document}
  ```

- Any commands between `\documentclass[...]{article}` and `\begin{document}` are said to be in the *preamble* of the document.

  The preamble can be used to define

  - ▷ textheight and width on page
  - ▷ header and footer lines
  - ▷ style of page numbering
  - ▷ . . .

- The main text is located between `\begin{document}`and `\end{document}` and is referred to as the body of the document. Further commands (starting with a backslash) can be used in the body to detail formatting instructions.

- Each document has to end with

  ```
  \end{document}
  ```

- A minimal LaTeX document looks like this:

```
\documentclass[a4paper]{article}
\begin{document}
\end{document}
```

This will just produce a blank A4 page.

Chapters, sections, subsections, . . .

→ Section 2.7 in LaTeX Tutorial

- `\section{TITLE}` starts a new section with title `TITLE`.

  LaTeX automatically changes the font size accordingly, and inserts a section number.

  Headings without numbers are created with `\section*{TITLE}`

- `\subsection{TITLE}` and `\subsubsection{TITLE}` start new subsections and subsubsections, respectively.

- In books and reports, the command `\chapter{TITLE}` starts a new chapter with title `TITLE`.

- Paragraphs are defined by empty lines in the source file.

- A new line can be enforced using `\\` at the point where the line-break should occur.

- Everything after a `%` character is a comment (and ignored by LaTeX). We can use this to document the LaTeX code, and to place comments into the document which we do not want to be seen by the reader.

→ Section 2.11.1 in LaTeX Tutorial

Lists

- Itemized lists

  - read the title
  - read the abstract
  - read the introduction and
  - read the summary (or conclusions).

  can be created using

```
\begin{itemize}
\item read the title
\item read the abstract
\item read the introduction and
\item read the summary (or conclusions).
\end{itemize}
```

- Enumerated lists such as

  1. Understand how LaTeX works
  2. Write your own documents
  3. Do something else

  are created using:

```
  \begin{enumerate}
  \item Understand how LaTeX works
  \item Write your own documents
  \item Do something else
  \end{enumerate}
```

- There is no limit with respect to the number of items that a list can take.

- Lists can be nested.

Other useful commands

→ Section 6.2 in LaTeX Tutorial

- `\centerline{TEXT}`

  prints `TEXT` centered. If `TEXT` is longer than one line, then one should use the center-environment:

  ```
  \begin{center}
  LONG TEXT
  \end{center}
  ```

- `\textbf{TEXT} \textit{TEXT}`

  prints `TEXT` in bold face and italics, respectively

test	`\textrm{test}`	RoMan (normal)
test	`\textbf{test}`	BoldFace
test	`\textit{test}`	ITalic
`test`	`\texttt{test}`	TypewriTer
test	`\emph{test}`	EMPHasize

- `\emph{TEXT}`

  can be used to *emph*asise `TEXT`. This usually (but not necessarily) means to print it in italics. This is the recommended way to emphasise words.

- `\huge \LARGE ...`

test	`\huge`
test	`\LARGE`
test	`\Large`
test	`\large`
test	`\normalsize`
test	`\small`
test	`\footnotesize`
test	`\tiny`

  change the font-size

  ▷ Either enclose the font-size changing command in curly braces to limit its effect:

  ```
  This is normal text. {\huge This sentence is
  huge.} Back to normal.
  ```

  This is the better method (because you go back to whatever font size was used before you switched to “huge”).

  ▷ Or switch back to normal size at a later point:

  ```
  This is normal text. \huge This sentence is
  huge. \normalsize Back to normal.
  ```

- `\footnote{TEXT}`

  creates a footnote containing TEXT.

- `\LaTeX`

  inserts the LaTeX-logo into the document.
  If you want a space after “LaTeX”, you should use

  `\LaTeX{}`

References to sections

At the time of writing we don't know which section number will be assigned to which section (because LaTeX does this at compile time). Therefore, we should never insert numbers of sections into the document when referring to other sections. Instead we use labels.

- Here is a bad example (schematic):

```
\section{Results}
Usually, reports have a results chapter, ...
\section{Discussion}
If not done in section 3 already, the results ...
```

- Here is a good example (schematic):

```
\section{Results}\label{sec:results}
Usually, reports have a results chapter, ...
\section{Discussion}
If not done in section \ref{sec:results} already, the results ...
```

- The command `\label{LABELNAME}` defines a label with name `LABELNAME`

- The command `\ref{LABELNAME}` refers to the label `LABELNAME`. When LaTeX compiles this document, it will insert the correct section number for `\ref{LABELNAME}`, in this case "3".

Other references

- References to sections, subsections, subsubsections, and chapters are done exactly as for sections, *i.e.* use a label and then refer to it.

- References to figures and tables use the same system. Note that the label for figures and tables should be in the caption (see slide 29).

- References to equations use the same system. Note that the label for equations should be placed in the equation environment.

- References to page numbers use `\pageref{LABELNAME}` to insert the page number on which the label `LABELNAME` is defined.

Typesetting equations (general guidelines)

→ Section 3 in LaTeX Tutorial

General guidelines for typesetting equations

- variables should appear in italics, *i.e.*

  $$x = y + 10$$

  not

  $$\mathrm{x} = \mathrm{y} + 10$$

- names that clarify the role of a variable should be upright, *i.e.*

  $$x_{\mathrm{max}} = 100$$

  not

  $$x_{max} = 100$$

  because max looks like m multiplied by a multiplied by x.

- Similarly, use $f(x) = \sin(x)$, not $f(x) = sin(x)$.

- Be aware that not all scientists and engineers stick to these conventions.

Equations in LaTeX

We distinguish

- “in-line equations” such as

  Our computations show that $\sqrt{4} = 2$.

  - ▷ switch on and off using $
  - ▷ Example: `Our computations show that $\sqrt{4}=2$.`

- “displayed equations” without numbering

  Our computations show that
  $$\sqrt{4} = 2.$$

  - ▷ switch on and off using $$
  - ▷ Example: `Our computations show that $$\sqrt{4}=2.$$`

- “displayed numbered equations”

  Our computations show that
  $$\sqrt{4} = 2. \qquad (1)$$

  - ▷ switch on and off using `\begin{equation}` and `\end{equation}`
  - ▷ Example:

```
Our computations show that
\begin{equation}
  \sqrt{4}=2.
\end{equation}
```

→ Section 3.1-3.10 in LaTeX Tutorial

output	latex	comment
$z = x + y$	`z = x + y`	
$a_0 = 0^0 = 1$	`a_0 = 0^0 = 1`	subscript (_) and superscript (^)
$c^2 = a^2 + b^2$	`c^2 = a^2 + b^2`	
$c = \sqrt{a^2+b^2}$	`c = \sqrt{a^2+b^2}`	curly braces group
$\gamma = \pm\sqrt{\Gamma}$	`\gamma = \pm\sqrt{\Gamma}`	pm = PlusMinus
$y = f(x)$	`y = f(x)`	
$1/2 = \frac{1}{2}$	`1/2 = \frac{1}{2}`	fraction
$x_1 = 0, \quad x_2 = 1$	`x_1 = 0, \quad x_2 = 1`	quad inserts space
p_{max}	`p_{max}`	bad: looks like m times a times x
p_max	`p_\mathrm{max}`	good: “max” is name
$f(x) = \sin(x)$	`f(x) = \sin(x)`	
$g(t) = \frac{1}{2}\cos(\omega t)$	`g(t) = \frac{1}{2}\cos(\omega t)`	
$x = \frac{a+b}{a-b}$	`x = \frac{a+b}{a-b}`	
$\int \sin(x) dx = \cos(x)$	`\int \sin(x) d x = \cos(x)`	not ideal: d looks like variable

$\int \sin(x) \mathrm{d} x = \cos(x)$	`\int \sin(x)`
	`\mathrm{d} x = \cos(x)`
		better: d looks like operator

$\int_0^{2\pi} \sin(x) \mathrm{d} x = 0$	`\int_0^{2\pi} \sin(x) \mathrm{d} x = 0`	
$\int\limits_0^{2\pi} \sin(x) \mathrm{d} x = 0$	`\int\limits_0^{2\pi}\sin(x) \mathrm{d} x = 0`	
$A_\Delta = h \sum_{i=0}^n f(x_i)$	`A_\Delta = h \sum_{i=0}^n f(x_i)`	

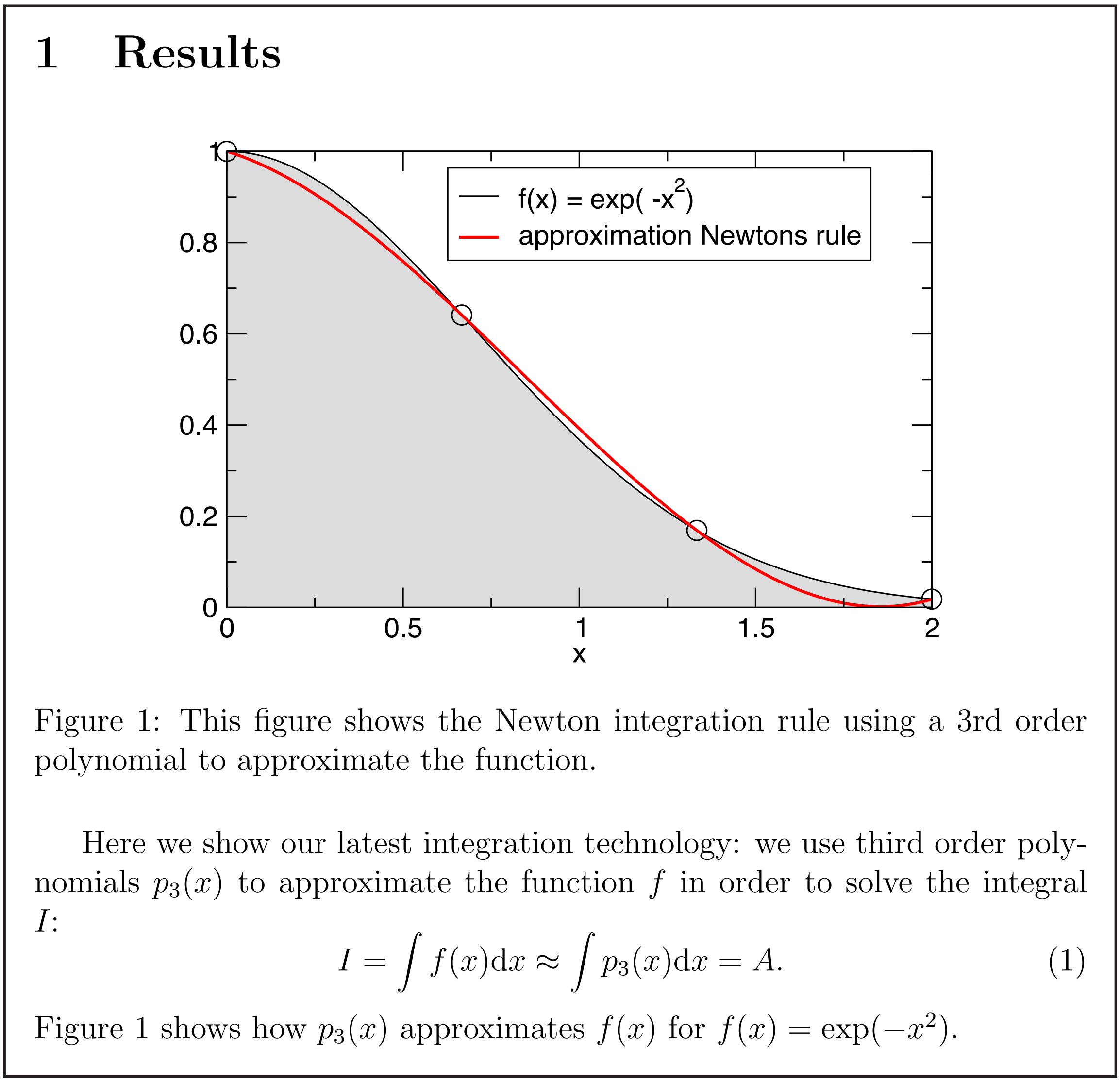
→ Section 4.1 in LaTeX Tutorial

Including figures

- LaTeX can include figures in "encapsulated PostScript" (eps) format
  - ▷ eps is very similar to PostScript (eps-files have bounding box and don't fill complete page)
  - ▷ can convert any bitmap (such as png) into eps (using for example CorelDraw)
  - ▷ create eps files straight away (using `pylab`). This is the preferred way because it generates genuine vector graphics that scale as required.

- (pdflatex can also include pdf and png files. See latex laboratory session.)

Example



1 Results

1
0.8
0.6
0.4
0.2
0
0
0.5
1
1.5
2
x
f(x) = exp(-x^2)
approximation Newtons rule

Figure 1: This figure shows the Newton integration rule using a 3rd order polynomial to approximate the function.

Here we show our latest integration technology: we use third order polynomials $p_3(x)$ to approximate the function f in order to solve the integral I:

$$I = \int f(x)\mathrm{d}x \approx \int p_3(x)\mathrm{d}x = A. \tag{1}$$

Figure 1 shows how $p_3(x)$ approximates $f(x)$ for $f(x) = \exp(-x^2)$.

Example latex source

Here is the LaTeX source for the document from slide 28

```
\documentclass[12pt,a4paper]{article}
\usepackage{graphicx}      %need this to include pictures
\begin{document}

\section{Results}

\begin{figure}[tbph]
  \centerline{
    \includegraphics[clip,width=10cm]{fig.eps}
    }% end centerline
  \caption{This figure shows the Newton integration rule using a
    3rd order polynomial to approximate the function.
    \label{fig:Newton}
    }% end caption
\end{figure}

Here we show our latest integration technology: we use third
order polynomials $p_3(x)$ to approximate the function $f$ in
order to solve the integral $I$:
\begin{equation}
  I = \int f(x) \mathrm{d}x \approx \int p_3(x) \mathrm{d}x = A.
\end{equation}
Figure \ref{fig:Newton} shows how $p_3(x)$ approximates $f(x)$ for
$f(x)=\exp(-x^2)$.
\end{document}
```

Including figures

- Use an extension to standard LaTeX by adding

  ```
  \usepackage{graphicx}
  ```

  to the preamble.

- An eps-file can be included using

  ```
  \includegraphics[OPTIONS]{FILENAME}
  ```

  Useful `OPTIONS` are

  ▷ `width=10cm` to scale the plot to be 10 cm wide.
  ▷ `height=6cm` to scale the plot to be 6 cm high.
  ▷ `clip` sometimes necessary if eps-file overlays text

  Example:

  ```
  \includegraphics[clip,width=10cm]{fig.eps}
  ```

- “clever” choice of width: use `\textwidth` to extract the current text width. To make the figure fill the complete width, use

  ```
  \includegraphics[clip,width=1\textwidth]{fig.eps}
  ```

  To make it half as wide as the text width, use

  ```
  \includegraphics[clip,width=0.5\textwidth]{fig.eps}
  ```

- Placement of figure
  - ▷ the `\includegraphics` command inserts the eps at the place where it is written.
  - ▷ Better: allow LaTeX to place the plot by using “floats”. We do this by surrounding the `includegraphics` command with a `figure`-environment (as already used on slide 29). The next slides explain this in more detail.

Float placement

→ Section 2.12 in LaTeX Tutorial

- Figures (and tables) should
  - ▷ be placed as close as to where they are mentioned first in the text
  - ▷ have a caption explaining what is shown
  - ▷ have a unique number
- The best way to achieve this in LaTeX is to use the figure environment:

```
\begin{figure}[tbph]         Begin figure environment
                             See slide for 33 [tbph]
  \centerline{               Center plot of eps-file
    \includegraphics[clip,width=10cm]{fig.eps}
                             This reads the file fig.eps
    }% end centerline        Close centerline brace
  \caption{This figure ...   Add the caption
      ... the function.      . . . still caption
    \label{fig:Newton}       Add LABEL into caption
    }% end caption           End caption
\end{figure}                 End figure environment
```

- Where LaTeX places floats
  - ▷ the letters t,b,p and h in `\begin{figure}[tbph]` determine where we allow LaTeX to place the float.
    - Top(t): allow figure at top of page
    - Bottom(b): allow figure at bottom of page
    - Here(h): allow figure to be place where the figure-environment is in the text
    - Page(p): allow figure to be place on a separate page (which contains only figures)

    The letters t, b, h and p can be combined as preferred.

- LaTeX will try to choose the best position for the figure

- This usually works well for large documents

- Fine-tuning of figure placements is possible
  (explained for example in book by Kopka and Daly, see slide 35)

Other random bits

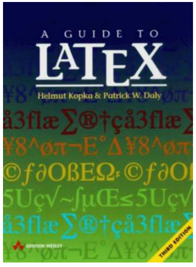
- Special symbols: → Section 3.10 in LaTeX Tutorial

$\approx$	`\approx`	need math-mode
∞	`\infty`	need math-mode
$\in$	`\in`	need math-mode
$\vec{x}$	`\vec{x}`	need math-mode
$\mathbf{x}$	`\mathbf{x}`	need math-mode
$\hat{x}$	`\hat{x}`	need math-mode
$\dot{x}$	`\dot{x}`	need math-mode
$\ddot{x}$	`\ddot{x}`	need math-mode
$\to$	`\to`	need math-mode
$\Rightarrow$	`\Rightarrow`	need math-mode
$\longrightarrow$	`\longrightarrow`	need math-mode
$\Longrightarrow$	`\Longrightarrow`	need math-mode
. . .	`\ldots`	
£	`{\pounds}`	

- functions
  - ▷ `\today` inserts today's date
  - ▷ `\vspace{1cm}` inserts 1 cm vertical space
  - ▷ `\vspace{-2cm}` inserts -2 cm vertical space, *i.e.* moves up

Further reading

- Book: "A Guide to LATEX" by Kopka and Daly. Around £30. Best book available, covers everything the average user needs.



- LaTeX Tutorial: "The not so short introduction to LaTeX" (pdf) is on the course-web-page http://www.soton.ac.uk/~sesg1009 under "online documents".

  We point to section X of this document (version 4.11) using → Section X in LaTeX Tutorial.
  Use this as a free reference with examples.

- There is more information on the web. Note:

  ▷ The current version of latex is 2ϵ, often written as LaTeX 2_εor LaTeX2e.
  ▷ The older version is LaTeX 2.09.

  Make sure that you use literature referring to LaTeX 2_ε.