

---

# Typesetting with T<sub>E</sub>X / L<sup>A</sup>T<sub>E</sub>X

---

## Part II: Formatting and Layout

F. C. Langbein

School of Computer Science  
Cardiff University





# Overview

- Part I: basic components and essential  $\text{\LaTeX}$
- **Part II:** formatting and layout
- Part III: figures and tables
- Part IV: basic mathematics and  $\text{AMS}\text{\LaTeX}$
- Part V:  $\text{PDF}\text{\LaTeX}$  and slides
- Part VI:  $\text{BIB}\text{\TeX}$  and  $\text{MakeIndex}$
- Part VII: useful things...

---

# Font Selection

---

# Font Selection Scheme

➤  $\text{\LaTeX}$  2<sub>ε</sub> font selection (**NFSS**):

- **Family**: gives the overall style: roman, typewriter, sans serif
- **Shape**: gives the form: upright, *italic*, *slanted*, SMALL CAPS
- **Series**: gives the weight: medium, **bold**

➤ Not all combinations exist as fonts

➤ Different realisation of fonts can be selected, e.g.:

- Helvetica or Computer Modern SS for sans serif
- Times or Computer Modern RM for roman
- A lot more depending on  $\text{\TeX}$  installation...

# Encodings

- There are **two types of encodings**:
  - **Input encoding** for input character to symbol
  - **Font encoding** for symbol to character in font
- **Select input encoding** (ascii, latin1 – latin9, ...):

```
\usepackage[latin1]{inputenc}
```
- **Select font encoding**:

```
\usepackage[OT1]{fontenc}
```

  - OT1 for Computer Modern fonts
  - T1 for new European CM fonts
  - OT1 supports type 1 PS fonts, T1 usually does not

# Font family, Shape, Series Selection

## ➤ Commands to **select family**:

`\textrm{roman}`

`{\rmfamily roman}`

`\texttt{typewriter}`

`{\ttfamily typewriter}`

`\textss{sans serif}`

`{\ssfamilly sans serif}`

## ➤ Commands to **select shape**:

`\textup{upright}`

`{\upshape upright}`

`\textit{italic}`

`{\itshape italic}`

`\textsl{slanted}`

`{\slshape slanted}`

`\textsc{small caps}`

`{\scshape small caps}`

## ➤ Commands to **select series**:

`\textmd{medium}`

`{\mdseries medium}`

`\textbf{bold}`

`{\bfseries bold}`

# Font Size

- Commands to **select font size** relative to standard size:

<code>\tiny</code> ( <small>tiny</small> )	<code>\scriptsize</code> ( <small>scriptsize</small> )
<code>\footnotesize</code> ( <small>footnotesize</small> )	<code>\small</code> ( <small>small</small> )
<code>\normalsize</code> ( <small>normalsize</small> )	<code>\large</code> ( <small>large</small> )
<code>\Large</code> ( <b>Large</b> )	<code>\LARGE</code> ( <b>LARGE</b> )
<code>\huge</code> ( <b>huge</b> )	<code>\Huge</code> ( <b>Huge</b> )
- Note, that here `\LARGE`, `\huge` and `\Huge` are the same size due to maximum font size available
- Commands switch font size, to limit effect use groups:  
`{\Large Large font}` normal font

# Emphasis, Underlining, etc.

➤ To **emphasise text** use the command `\emph{emphasised text}` or `{\em emphasis}`

- Emphasis is usually indicated by *italics*, sometimes it may be **boldface**

- Nested emphasis (default behaviour):

```
\emph{emphasised \emph{text}}
```

*emphasised text*

➤ To **underline** text:

```
\underline{underlined text}
```

- Underline is **obsolete** for *emphasis*

➤ To **switch to basefont**:

```
\textnormal{normal} or {\normalfont normal}
```



# Textfont Settings

- **Change typeface** for family, shape, series selection:
  - Default is Computer Modern Fonts
  - Simple adjustments by loading packages:
    - `\usepackage{helvet}`: use helvetica for san-serif
    - `\usepackage{times}`: use times roman ps font
    - `\usepackage{pandora}`: use pandora fonts
    - `\usepackage{concrete}`: use Concrete Roman + Euler Maths font
- For more fonts, low-level redefinitions, etc. see literature
- Math fonts are separate from text fonts

---

# Centering, Justification, Space, Tabbing

---

# Centering Text

➤ To **centre text** use center environment:

```
\begin{center}  
  line 1\\ longer line 2\\  
  even longer line 3\\  
\end{center}
```

line 1  
longer line 2  
even longer line 3

➤ Obsolete commands:

- `\centering`: everything that follows is centred
- `\centerline{text}`: argument is centred

# Justification

➤ By default paragraphs are fully justified

➤ To **change justification**:

Declaration	<code>\flushleft</code>	<code>\flushright</code>	<code>\centering</code>
Environment	<code>flushleft</code>	<code>flushright</code>	<code>centering</code>

➤ Only whole paragraphs can be justified

```
\{\flushright This even  
longer text is right  
justified.\par\}
```

```
\begin{flushleft}  
This even longer text  
is right justified ...  
\end{flughleft}
```

This even longer text is  
right justified.

This even longer text is left  
justified. Some more text to  
get more lines.

# Paragraph Indentation

➤ Paragraph indentation is enabled by default

➤ Set **default paragraph indentation** in preamble:

```
\parindentlength
```

● E.g. to disable indentation use:

```
\parindent0cm
```

➤ `\indent` inserts horizontal space equal to paragraph indentation

➤ `\noindent` at **beginning of paragraph** suppresses indentation

# Horizontal Space

- Insert **horizontal space** with `\hspace{length}`:

```
Insert \hspace{5cm}space
```

Insert                      space

- Use `\hspace*{length}` for **space that will never be removed**

```
\par \hspace{5cm}Start
```

Start

```
\par \hspace*{5cm}Start
```

Start

- `\hfill` inserts **stretchable horizontal space**:

```
Stretched \hfill horizontal \hfill space
```

Stretched                      horizontal                      space

# Vertical Space

- Insert **vertical space** (between paragraphs/boxes) with the command `\vspace{length}`:

```
This is the first  
\vspace{3cm} paragraph.
```

```
And the second  
paragraph
```

This is the first paragraph.

And the second paragraph

- Use `\vspace*{length}` for **space that will never be removed**
- Use `\vfill` for **stretchable vertical space** (similar to `\hfill`)

# Quotations

➤ For **longer quotations** use `quote` or `quotation` environment:

H.P. Lovecraft:

```
\begin{quote}
```

```
He began to read  
into the odd angles  
a mathematical...
```

```
\end{quote}
```

H.P. Lovecraft:

He began to read into  
the odd angles a mathe-  
matical significance which  
seemed to offer vague  
clues regarding their pur-  
pose.

- `Quote` sets the text somewhat narrower than normal
- `Quotation` is like `quote`, but also indents paragraphs



# Tabbing Environments

- tabbing environments set **tabulator stops** similar to a type-write (or word processor)
  - Enclose everything in a tabbing environment
  - Tab stops are set with `\=`
  - There is no automatic line breaking, lines must be broken with the usual `\\`
  - After tab stops have been set, one moves to pre-set tab stops using `\>`
  - Tab stops can be set in a line that is not printed if the line is ended with the `\kill` command
  - Tab stops can be reset or added in every line, not just the first

# Tabbing Example

```
\begin{tabbing}
\hspace{6cm} \= \hspace{10cm} \= \kill
\textbf{Lecturers} \> Dr.\ J.\ Doe \> (0500) 1231
456 \\\
\> Homology\\
\> Dr.\ H.\ Foo \> (0500) 654 321\\
\> Cohomology
\end{tabbing}
```

<b>Lecturers</b>	Dr. J. Doe	(0500) 123 456
	Homology	
	Dr. H. Foo	(0500) 654 321
	Cohomology	

---

# Lists

---

# Itemize

- **Bullet lists** are created by an `itemize` environment:

```
\begin{itemize}
\item First item.
\item Second item.
\item Third item.
\end{itemize}
```

- First item.
- Second item.
- Third item.

- **Change labels** on a case-by-case basis by using an optional argument:

```
\begin{itemize}
\item[$\clubsuit$] clubs
\item[$\heartsuit$] hearts
\end{itemize}
```

♣ clubs  
♥ hearts

# Nested Itemize Environments

➤ Up to four `itemize` environments may be **nested**:

```
\begin{itemize}
\item Item 1
\begin{itemize}
\item Item 1.1
\item Item 1.2
\end{itemize}
\item Item 2
\begin{itemize}
\item Item 2.1
\end{itemize}
\end{itemize}
```

- Item 1
  - Item 1.1
  - Item 1.2
- Item 2
  - Item 2.1

# Enumerate

➤ **Numbered lists** are generated like itemized lists in an `enumerate` environment:

```
\begin{enumerate}  
\item Item 1  
\begin{enumerate}  
\item Item 1.1  
\item Item 1.2  
\end{enumerate}  
\item Item 2  
\begin{enumerate}  
\item Item 2.1  
\end{enumerate}  
\end{enumerate}
```

1. Item 1
  - (a) Item 1.1
  - (b) Item 1.2
2. Item 2
  - (a) Item 2.1

# Enumerate and Itemize Lists

- Enumerate and itemize lists may be nested (level is handled separately)

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

- First:
  1. Enum 1
  2. Enum 2

- Optional `\item` argument to set label can also be used for `enumerate`
- Check out the `enumerate` package for more numbered list styles

# Change Labels Globally

- The default label can be redefined for the whole level:

```
\begin{itemize}
\renewcommand{\labelitemi}{$\star$}
\item Item 1
\item Item 2
\end{itemize}
```

★ Item 1  
★ Item 2

- This can be done at all four levels in enumerate and itemize environments (see literature)
- Similarly spacing between items and other dimensions can be adjusted (see literature)



# Description

➤ For glossaries, etc. use description environment:

```
\begin{description}  
\item[Cabbage] A large  
round green vegetable  
\item[Brussel sprout] A  
small round green  
vegetable  
\end{description}
```

**Cabbage** A large round  
green vegetable

**Brussel sprout** A small  
round green vegetable

# General Lists

- General lists may be created using the `list` environment:

```
\begin{list}{label}{declarations}  
\item first item ...  
\end{list}
```

- **label**: is the label for any items not specifying their own
- **declarations**: list definition (lengths,...) commands

- Example for a list in a CV:

<b>EDUCATION</b>	Ph.D., Engineering Mechanics, 1993 <i>University of Wisconsin–Madison</i>
	M.S., Engineering Science, 1987 <i>Harvard University</i>
<b>EXPERIENCE</b>	Some jobs here and some jobs there.

# General Lists Example Code

```
\begin{list}{}{\setlength{\leftmargin}{8cm}
\setlength{\labelwidth}{8cm}
\setlength{\labelsep}{0cm}
\setlength{\parsep}{8pt plus 1 pt minus 0pt}
\setlength{\itemsep}{15pt plus 1 pt minus 0pt}
\setlength{\topsep}{10pt plus 1pt minus 0pt}}
\item[\textsc{\textbf{Education}}\hfill]
    Ph.D., Engineering Mechanics, 1993\\
    \emph{University of Wisconsin--Madison}

    M.S., Engineering Science, 1987\\\emph{Harvard University}
\item[\textsc{\textbf{Experience}}\hfill]
    Some jobs here and some jobs there.
\end{list}
```

---

# Boxes

---

# What are Boxes?

- A box is an **object** that  $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  treats as a unit
  - Each character is a box
  - A sequence of characters can be combined to a box
  - Paragraphs, columns, footer, headers, ... are boxes
  - $\text{T}_{\text{E}}\text{X}$  compiler **assembles boxes** in 2D
- A box can be moved around, but it can't be broken up
- $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  provides special boxes for layout/formatting

# Simple Boxes

➤ **Simplest box** type for text: `\mbox{contents}`

- Prevents text inside it from being broken across a line
- Provides normal text inside maths environment
- Dimensions of box automatically computed to fit contents

➤ An mbox where width and justification can be specified:

```
\makebox[width][hor. alignment]{contents}
```

- **width**: length of box
- **hor. alignment**: l, c, r, s for left/centre/right/stretch alignment inside box

# Framed Boxes

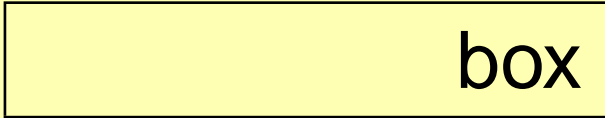
- A simple **framed box**:

```
\fbox{some text}
```

A simple rectangular box with a black border containing the text "some text".

- A **framed box with width and alignment**:

```
\framebox[8cm][r]{box}
```

A rectangular box with a black border, right-aligned, containing the text "box".

- **Shadow box** from shadow package

```
\shabox{Shadow}
```

A rectangular box with a black border and a shadow effect, containing the text "Shadow".

- More boxes are defined in fancybox package

# Paragraph Boxes

➤ A box to typeset **custom paragraphs**:

```
\parbox[pos][height][innerpos]{width}{text}
```

- **pos**: t, b, c for centre, top, bottom vertical alignment
- **height**: height of paragraph box
- **innerpos**: t, b, c for vertical alignment of contents
- **width**: width of paragraph
- **text**: contents of paragraph

➤ May be used to position paragraphs horizontally and vertically on page with specified dimensions





# Paragraph Box Example

➤ Example:

## Heading

This is a simple paragraph box with a heading.

This is another paragraph box

➤ Code:

```
\parbox[t]{16cm}{ \textbf{Heading}\\
  This is a simple paragraph box with a heading.}
\hspace{1cm}
\parbox[t][5cm][b]{8cm}{
  This is another paragraph box}
```

# Minipage

- A minipage environment creates a box which behaves like a **small page** inside the page:

```
\begin{minipage}[ver. pos]{width}  
Contents...  
\end{minipage}
```

- **ver. pos**: relative vertical position of minipage (t, b, c)
  - **width**: width of minipage (height determined by contents)
- Minipage is more general as a parbox
  - Minipage may contain anything a standard page contains



# Minipage Example

## ➤ Code:

```
\begin{minipage}[b]{8cm}
This is the first paragraph of the minipage.

Second paragraph follows here.
\end{minipage}
\begin{minipage}[c]{8cm}
And another minipage with one paragraph.
\end{minipage}
```

## ➤ Example:

This is the first paragraph of the minipage.

Second paragraph follows here.

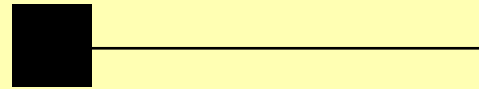
And another minipage with one paragraph.

# Rules and Struts

- A rule box allows to draw **thin lines** or **rules**:

```
\rule[lift]{width}{height}
```

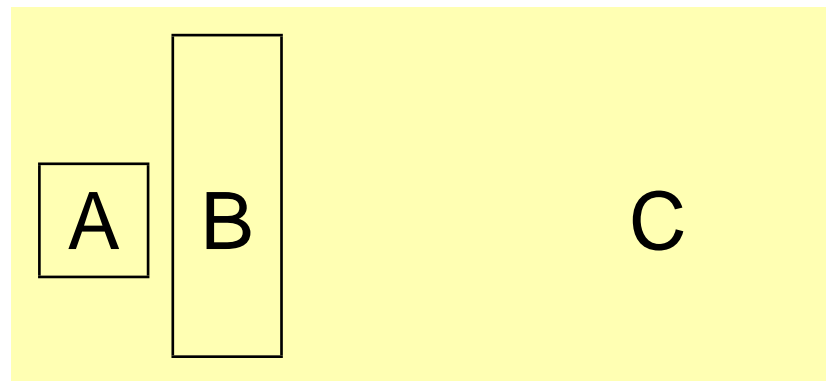
```
\rule[-5mm]{1em}{2ex}%  
\rule{5em}{1pt}
```



- What would happen without % ?

- A zero width/height rule is called a strut for invisible place holders

```
\fbox{A}  
\fbox{\rule[-30pt]{0pt}{100pt}B}  
\rule{4cm}{0pt} C
```



---

# Page Layout

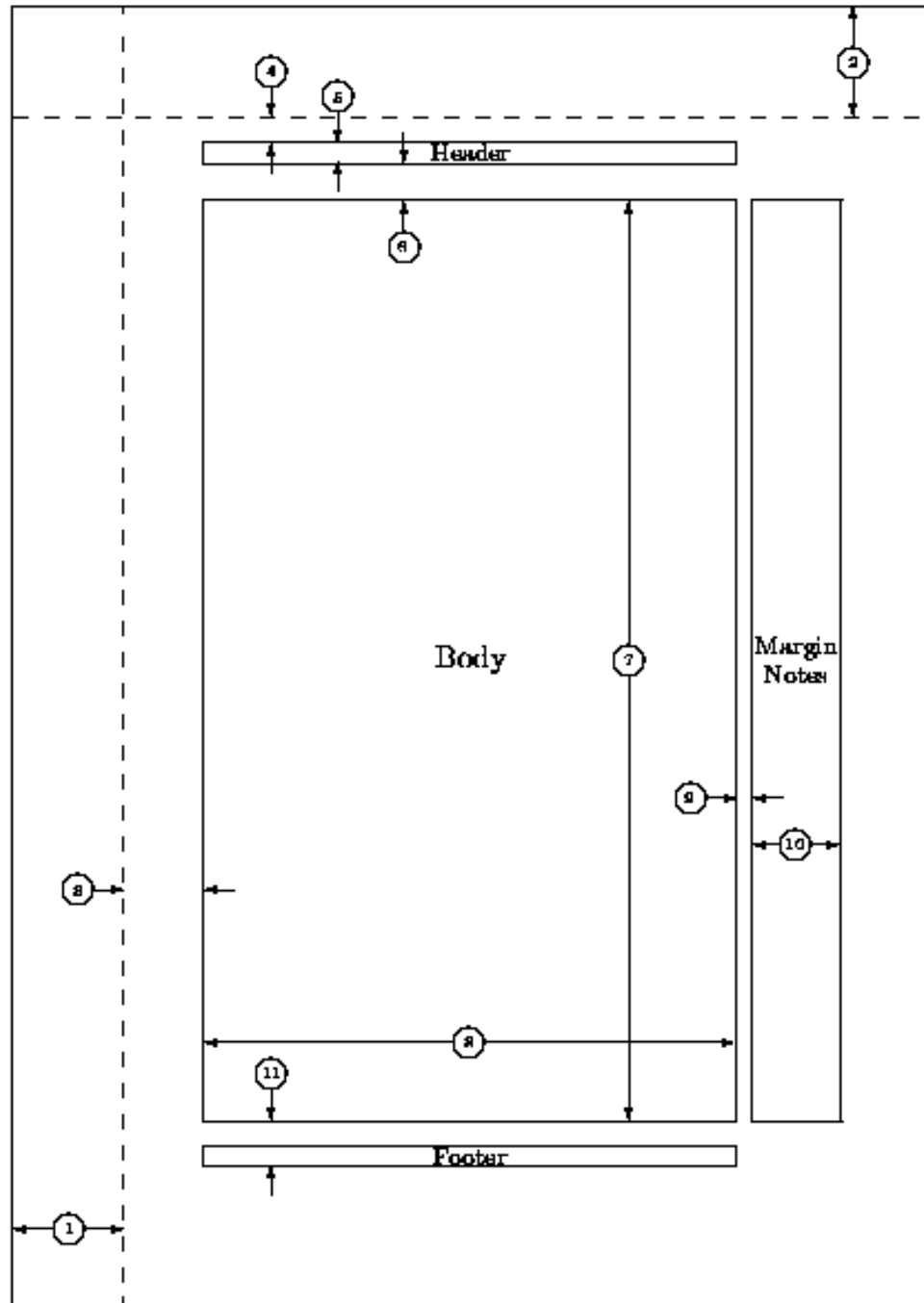
---



# Basic Page Layout

- Page layout can be adjusted by
  - Document class and package options
  - Page layout / size packages and special commands
  - Adjusting low-level page dimensions
- Each page consists of a head, a body and a foot (boxes)
  - There is an additional box for margin notes
- Distances are used to determine the location of these elements
- Note,  $\text{\LaTeX}$  measures all distances from the edge of a box which is one inch from the top of the paper and one inch from the left edge

# Page Layout



- 1 `1in + \hoffset`
- 2 `1in + \voffset`
- 3 `\oddsidemargin`
- 4 `\topmargin`
- 5 `\headheight`
- 6 `\headsep`
- 7 `\textheight`
- 8 `\textwidth`
- 9 `\marginparsep`
- 10 `\marginparwidth`
- 11 `\footskip`
- `\marginparpush`
- `\paperwidth`
- `\paperheight`



# Page Dimensions

- Page dimensions may be adjusted with `\setlength`:

```
\setlength{\textwidth}{6.5in}  
\setlength{\oddsidemargin}{0.0in}
```

- Note that `\oddsidemargin` is left margin in oneside mode, in twoside mode there is also `\evensidemargin`
- `geometry` package contains many useful macros for setting page sizes
- Use `vmargin` package for macros for setting page margins



# More on Lengths

➤ There are **rigid** and **rubber** lengths:

- Rigid lengths are fixed lengths: `4in`
- Rubber lengths are elastic lengths: `2in plus 0.1in minus 0.2in`
- A rubber length tells  $\text{\LaTeX}$  a preferred length and the amount of deviation that is acceptable

➤ Length units:

<code>pt (1/72.72in)</code>	<code>bp (1/72in)</code>	<code>mm (2.845pt)</code>
<code>cm (28.45pt)</code>	<code>in (2.54cm)</code>	
<code>ex (height of current font x)</code>		
<code>em (width of current font em-dash)</code>		

# Even more on Lengths

➤ Add value to length:

```
\addtolength{cmd}{delta}
```

● Length values can be multiplied with constants, e.g.

```
\addtolength{\textwidth}{.5\textwidth}
```

➤ Additional commands to set lengths to dimensions of text:

```
\settowidth{length}{text}  
\settoheight{length}{text}  
\settodepth{length}{text}
```

➤ Create new length:

```
\newlength{length-cmd}
```

➤ Typeset value of length:

```
\the length
```



# Page Styles

- Style of page determines contents of the header and footer
  - Page numbers appear automatically
  - Page number omitted on title page

- Page style can be changed using:

```
\pagestyle{style}
```

- Common styles are: plain, empty, headings

- Change style of current page only:

```
\thispagestyle{style}
```

# Fancy Header

➤ fancyhdr package provides a simple way to adjust headers and footers

- Include fancyhdr package

- Select fancy page style: `\pagestyle{fancy}`

➤ Commands to define header/footer left/centre/right:

```
\lhead{text}, \chead{text}, \rhead{text},  
\lfoot{text}, \cfoot{text}, \rfoot{text}
```

- E.g.: `\cfoot{\thepage}`

➤ Set height of footer/header with

```
\renewcommand{\footrulewidth}{x}  
\renewcommand{\headrulewidth}{x}
```

# Fancy Header and Even/Odd Pages

➤ To clear all header/footer fields:

```
\fancyhf{}
```

➤ Set header contents with:

```
\fancyhead[SPEC1,SPEC2,...] {contents}
```

- Similarly for footer contents: `\fancyfoot`

- Comma separated list of **SPECx** identifies pages:

E: even page, O: odd page, L: left, C: center, R: right

➤ Examples:

```
\fancyhead[LE,R0]{\thepage}  
\fancyfoot[LE,R0]{\thepage}
```