An Introduction to TeX and LaTeX:

Lecture 1: The Basics

Tuesday 11 October 2016

- **Background**

- **Getting equipped**

- **Getting started in LaTeX**

- **Basics of TeX**

The background

T_EX is a typesetting system

Designed with mathematics particularly in mind

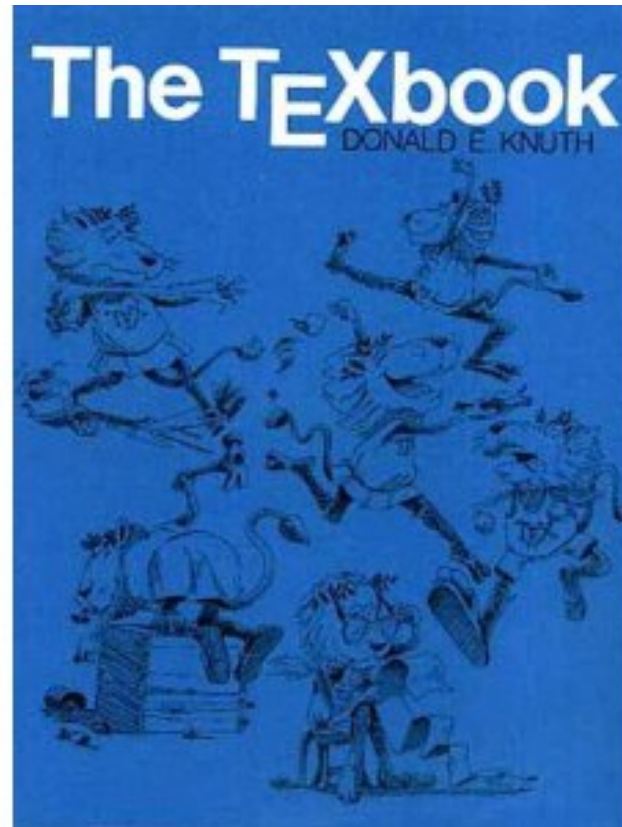Excellent for text, but also typesets formulae to the highest standards of mathematical typography

Created by the great Donald Knuth

Published in his beautiful The $T_EX$book (1984)

Knuth and his book

Donald Knuth of Stanford U.

The TeXbook (1984)

Dialects of TₑX

In use TₑX calls on style files, font packages, other packages, etc. to produce desired style

Main dialects: PlainTₑX, AMSTₑX, LᴬTₑX

LᴬTₑX created by Leslie Lamport (1985)

LᴬTₑX is perhaps more than just a dialect: it is an overlay of TₑX designed for organising documents as preprints, articles, books, letters, etc.

LᴬTₑX has become industry standard; most publishers require authors to submit LᴬTₑX source material; as does Oxford Mathematics Department

Type-setting mathematics: background

1439–50: Gutenberg invents printing from movable type

1885: Monotype machine casts hot lead in whole pages

Photosetting replaces hot metal from **1960s**

Computer-setting **+** laser-printing from **1980s**

Now: we do our own type-setting using T_EX and $\LaTeX$.

Learn their proper use: don't make a mess of it!

Getting equipped

You need an editor for plain-text input file

You need a compiler

You need a source of advice

All available in the Mathematical Institute:

https://www.maths.ox.ac.uk/members/it/faqs/latex

Personal T_EX equipment

Several T_EX systems available free of charge for download from the web—see cited web-page, especially CTAN

Myself, I use **TeXLive** as my T_EX system and **Texmaker** as a front end

Front end = editor into which to type my plain-text together with click-buttons to compile and view documents

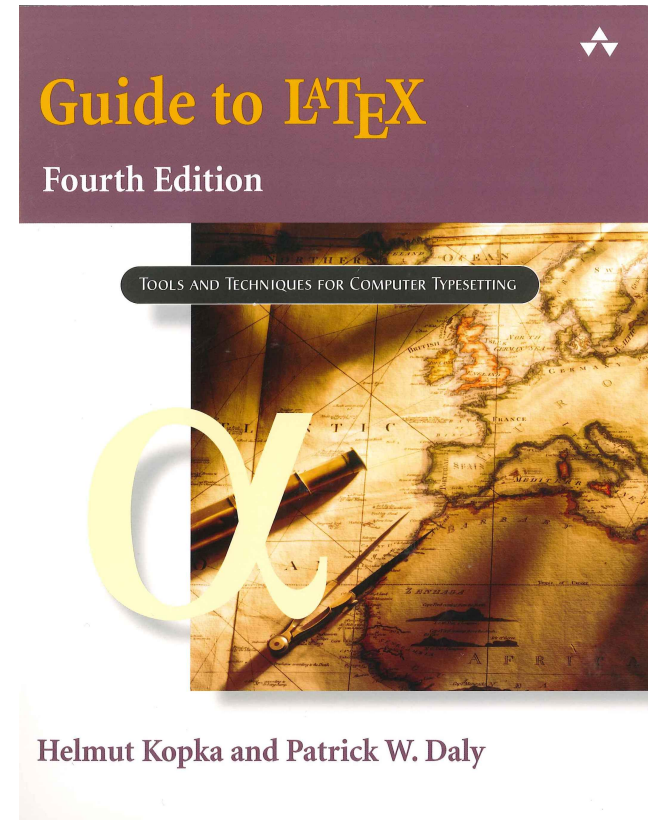All such systems are available for MS-Windows, Apple Mac, Linux etc.

A recommended manual

Helmut Kopka &
Patrick W. Daly

A Guide to LATEX

(4th ed.) Addison–Wesley 2003

Comes with TEX Live 2003 cd
(free TEX software)

Guide to LATEX

Fourth Edition

TOOLS AND TECHNIQUES FOR COMPUTER TYPESETTING

Helmut Kopka and Patrick W. Daly

Getting started in LaTeX

$\TeX$ and $\LaTeX$ are programming languages

Commands are introduced with a backslash \

Example: `\TeX` tells the compiler (type-setter) to produce $\TeX$

Example: `\noindent` at the start of a paragraph ensures that the paragraph is not indented

An important fact

Commands have form `\string` where '`string`' is a string of <u>letters</u>

Compiler seeks first non-letter key-stroke after backslash and stops there

Non-letter key-stroke might be a space, a numeral, a punctuation mark—anything non-literal

$\mathrm{T_EX}$ takes this literally: spaces after a command are taken simply as command terminator, so do not appear as a space

Example: `\LaTeX`   `is a wonderful system`

produces   LATEXis a wonderful system

LATEX input file

The basic structure of a LATEX input file consists of just three command lines with lines of other material between them:

```
\documentclass[X]{Y}
     [Preamble]

\begin{document}
     Your text

\end{document}
```

Here **X** is optional, **Y** is not; preamble material is optional

The document class

Options **X** available to \documentclass tell compiler about font size, paper size, etc.: see any LATEX manual

Class **Y** tells compiler what kind of document is to be produced;

it is a name for a **class** file such as **book, article, report**;

most publishers have their own class file **filename.cls** or **filename.sty** which they require their authors to use

Thus the first line of my lecture-notes file is

    \documentclass[a4paper,11pt]{article}

The preamble

Preamble follows the 'documentclass' line to give further information about preferences

Example: the two next lines in my file are

```
\usepackage{latexsym,amssymb}
\usepackage{graphicx}
```

First tells system to summon up LaTeX and AMS special symbol files and commands;   second requests one of the packages permitting me to insert pictures

The rest of my preamble contains my definitions of commands that I personally find useful

The basics of T$_E$X

T$_E$X program distinguishes text mode from math mode

Text is the natural mode—the compiler sets your typing into the specified type-fount, chooses line-breaks, chooses page-breaks, etc.

Important: When compiling, the system treats any positive number of inter-word spaces in your input file as just one space; it treats a single line break as an inter-word space

But when it meets one or more blank lines it takes this as an instruction to start a new paragraph

Therefore if you really want extra horizontal or vertical space you must use explicit commands (to be discussed later) to impose your will

Math mode, I

Mathmode comes in two forms, **textstyle** and **displaystyle**

Textstyle is for setting formulae such as $ax^2 + bx + c$ within text

Displaystyle is for complicated or lengthy formulae such as

$$f(a) = \int_{\Gamma_r(a)} \frac{f(z)}{z - a} \, \mathrm{d}z$$

or

$$f(a) = f(0) + f'(0)a + \frac{f''(0)}{2!}a^2 + \cdots + \frac{f^{(n-1)}(0)}{(n-1)!}a^{n-1} + R_n(f, a),$$

which can be too cramped or hard to read if embedded in text

Math mode, II: textstyle

Textstyle mathmode is created by enclosing the required formulae between dollar symbols.

The quadratic formula above came from `$ax^2 + bx + c$`.

In LaTeX one has alternative versions

`\(ax^2 + bx + c\)`

`\begin{math}ax^2 + bx + c\end{math}`

I do not know why

Math mode, III: displaystyle

Displaymath is created by enclosing the required formula-describing text between double dollar signs
(there are alternatives to be mentioned later)

Examples:

```
$$
 f(a) = \int_{\Gamma_r(a)} {f(z)\over z - a}\ \mathrm{d}z
$$


$$
 f(a) = f(0) + f'(0)a + {f''(0) \over 2!}a^2 + \cdots
 + {f^{(n-1}(0) \over (n-1)!} a^{n-1} + R_n(f,a),
$$
```

Important note: Modern LATEX has \[\] replacing $$ $$

Two further lectures

Lecture 2: The interior and the exterior of documents

The interior: how to get T_EX to give you what you want in text and mathematical formulae

The exterior: how to format a document in LAT_EX—including how to control your bibliography

Lecture 3: Sense and sensibility

Errors to avoid: how not to use LAT_EX

Style: some pointers about typography

An Introduction to TeX and LaTeX:
Lecture 2: Interior and exterior of documents

Tuesday 18 October 2016

- **The interior: text in TeX**

- **The interior: environments in LaTeX**

- **The interior: formulae in TeX**

- **The exterior: formatting a document in LaTeX**

Text in T_EX

Just type it in: by default T_EX prints it in roman type.

To change font use

- `{\it text}` or `\textit{text}` for *italics*
- `{\bf text}` or `\textbf{text}` for **bold face**
- `{\sc text}` or `\textsc{text}` for SMALL CAPS
- `{\tt text}` or `\texttt{text}` for `typewriter`
- `{\rm text}` or `\textrm{text}` for roman

A related command is `\emph{text}`. The effect is to turn 'text' into italic type if the ambient type is roman and *vice-versa*.

Adjusting type size

Just as you can change type font so you can change type size:

`{\large large}, {\Large larger}, {\LARGE very large},`

produce large, larger, very large.

Similarly

`{\small small}, {\footnotesize smaller}, {\tiny too small},`

produce small, smaller, too small.

Use of curly brackets (braces)

Braces { } delimit blocks. They create a unit.

Sometimes blocks limit the effect of a command as in
`{\it italics}` or `{\LARGE large}` to stop it from continuing in an
unwelcome fashion.

Sometimes the block becomes the argument to a command such
as `\textit` that requires input.

Sometimes, as in `{\TeX}`, the closing bracket provides the
non-letter character needed to terminate the command. Then it
must be matched with its paired opening bracket—in a civilised
world all brackets (of all types) always come in pairs.

Special symbols

A problem: how do we get T_EX to type-set special symbols such as \, {, }, $?

These are not the only so-called <u>command</u> characters. The full list is: \ $ # ~ _ ^ % { }.

All except backslash can be obtained by typing the character preceded by backslash. Thus \$ produces $.

Do the same for characters that are used in print but do not appear on your keyboard: \S produces §, \dag produces †, \ddag produces ‡, and (because it does not appear on other than British keyboards), \pounds produces £, etc.

Verbatim

How do we get backslash? Could use mathmode: `$\backslash$`.
Note that `$\backslash$` $\neq$ `$\setminus$`. The latter puts space
appropriate for a binary operator fore and aft as in $A \setminus B$.

Better to remain in text and go into 'verbatim' mode: type
`\verb=\=`. LaTeX compiler takes command `\verb=characterstring=`
and prints `characterstring` like that, exactly as you typed it.

Three rules: (1) the delimiters (here $=$) that bracket the
character string may be any symbol other than a letter or a space
and must be the same symbol fore and aft of the string;
(2) that symbol must not occur in the character string itself;
(3) the whole string `\verb=characterstring=` must be short enough
to fit into one line of typing.

The nine command characters

The nine command characters: what are they all for?

- Backslash \, $, braces { , }: already explained;
- # a place–holder used in commands that take one or more arguments;
- ˜ a protected space (a tie character);
- _, ˆ used for subscripts and superscripts in mathmode;
- % allows one to make comments in one's input file.

Any typing appearing on a line after % is ignored by the type-setting program. Applies only to individual lines.

Accented characters in text

- \' puts acute accent over letter that follows, as in é, ź;

- \` puts grave accent over next letter as in ù, x̀;

- \^ provides circumflex as in â, ŵ;

- \" provides umlaut as in ö, v̈;

- and there are many more.

For a full list of accents see any TEX or LATEX manual.

The interior: environments in LaTeX

Environments provide a good way to organise blocks of special material in LaTeX. What you type is always of the form

```
\begin{environmentname}
 Your special material
\end{environmentname}
```

Thus if you type, for example,

```
\begin{theorem}
  All groups of order $1$ are purple.
\end{theorem}
```

you should get something like

Theorem 1 All groups of order 1 are purple.

Labels and cross referencing

One can label words and items. If I had typed

```
\begin{theorem}\label{Some purple groups}
 All groups of order $1$ are purple.
\end{theorem}
```

then I would have got exactly the same arrangement of my theorem, but would now be able to cross-reference it elsewhere in my document. Typing

```
Most groups are green but by Theorem~\ref{Some purple groups}
certainly not all.
```

yields

Most groups are green but by Theorem 1 certainly not all.

Labels and page references

One gets page references in a very similar way:

```
By the theorem on p.\,\pageref{Some purple groups}
not all groups are green.
```

yields

By the theorem on p. 8 not all groups are green.

Environments, II

Many environments are standard for all classes of document. You'll find

- theorem, lemma, proposition, definition ["you" define these];
- itemize, enumerate, description [for creating lists];
- large, Large, LARGE [for increasing type size];
- small, footnotesize, scriptsize, tiny [for decreasing type size];
- titlepage;
- quotation, quote, verse

and many, many more—as always,consult a proper manual.

The bibliography environment

A very important environment is `thebibliography` for dealing with
reference lists. I have used it for my lecture notes by typing

```
\begin{thebibliography}{\bf 10}

  \bibitem{Chaundy-etal-1954}
   {\sc T.~W.~Chaundy, P.~R.~Barrett, and Charles Batey}
   {\it The printing of mathematics},
   Oxford University Press, 1954.

  \bibitem{Knuth1984}
   {\sc Donald Knuth},
   {\it The {\TeX}book},
   Addison-Wesley, Reading MA, 1984.

\end{thebibliography}
```

Analysis of the bibliography environment, I

`{\bf 10}` is a sample label. It tells the system how much width to allow for item labels in the bibliography.

`\bibitem` indicates the start of the next item in the reference list;

its argument, here `Chaundy-etal-1954` or `Knuth1984`, is a key or label, which may be called with the command `\cite`. Thus if I type

```
This command is treated in Knuth's classic \cite[p.\,45]{Knuth1984}.
```

then I get

This command is treated in Knuth's classic [2, p. 45].

Analysis of the bibliography environment, II

By default bibliography labels are boldface numbers. But in fact `\bibitem` permits an optional label for the reference. If I had typed

```
\bibitem[Knuth (1984)]{Knuth1984}
```

in the list in the bibliography environment then my example above would have come out as

This command is treated in Knuth's classic [Knuth (1984), p. 45].

Note the protected spaces (tie characters) in

```
{\sc T.~W.~Chaundy, P.~R.~Barrett, and Charles Batey}
```

These protect against bad line-breaks and also turn the full stop into an abbreviation stop.

Alternatives to the bibliography environment

Many colleagues prefer to use programs such as BibTeX.

BibTeX allows you to build up a database from which it extracts bibliographic details.

If you want to use BibTeX (or other such systems) read the manuals or get advice from one of its fans.

The interior: formulae in T_EX

Return to mathmode. Recall that formulae are bracketed by two
$ symbols if they are to be incorporated into the text and by two
double dollar symbols $$ if they are to be displayed. Recall also
that there are variants of each of these.
In the displaymath case there is a fourth possibility, namely
        `\begin{equation} ... \end{equation}` which automatically
adds a sequential equation number.

Two more constructions:

        `\begin{eqnarray} ... \end{eqnarray}`

        `\begin{eqnarray*} ... \end{eqnarray*}`

These produce multiline formulae: example on next screen

Equation array: an example

Example:

```
\begin{eqnarray*}
 f(x) \kern-6pt&=&\kern-6pt (x - 1)(x - 2)(x - 3)(x - 4)(x - 5)\\
   \kern-6pt&=&\kern-6pt x^5 - 15 x^4 + 85x^3 - 225x^2 + 274x - 120.
 \end{eqnarray*}
```

yields

$$
\begin{aligned}
f(x) &= x(x-1)(x-2)(x-3)(x-4)(x-5) \\
     &= x^5 - 15x^4 + 85x^3 - 225x^2 + 274x - 120.
\end{aligned}
$$

The `eqnarray` environment adds equation numbers; the * form (used for this example) does not.

Note my use of `\kern-6pt` to correct the spacing.

Typing formulae for T_EX

Typing formulae for T_EX to compile is pretty intuitive. But——

Important: in mathmode the system ignores spaces and, unless instructed otherwise, prints all letters in the 'mathitalic' font. Thus

```
$|x| = x if x \geq 0 and |x| = -x if x < 0$.
```

produces $|x| = x if x \geqslant 0 and |x| = -x if x < 0$.

The sentence contains 4 separate formulae, so should be typed

```
$|x| = x$ if $x \geq 0$ and $|x| = -x$ if $x < 0$.
```

which yields: $|x| = x$ **if** $x \geqslant 0$ **and** $|x| = -x$ **if** $x < 0$.

18

Text within formulae

Sometimes one wishes (or even needs) to move into textmode within a formula. This is done with `\mbox{yourtext}`. And of course within an mbox one can move back into mathmode (textstyle mathmode, not displaymath). Thus

```
$$
  \mathbb{Q} := \{x \in \mathbb{R} \mid \mbox{$x$ is rational}\}.
$$
```

produces

$$\mathbb{Q} := \{x \in \mathbb{R} \mid x \text{ is rational}\}.$$

Commands for mathmode

Most commands are intuitive and easily learned. Examples:

- **Greek letters are** `\alpha, \Alpha, \beta, \Beta, ...;`

- `\int` **gives integral sign;**

- `\sum` **gives summation symbol (do not confuse with Gk** Σ**),**

- `\prod` **gives product symbol (do not confuse with Greek** Π**).**

- **For superscripts use** `^`**, for subscripts use** `_` **(underline key on your keyboard).**

Note that a multi-symbol argument for `^`, `_` must be put into a block picked out with braces `{ }`.
Thus we type `x^{-2}gx^2` to get $x^{-2}gx^2$.

Mathfonts for mathmode

Fonts available in mathmode (some just in $\mathcal{AMS}$ range):

- `\mathrm{something}` **will print** something **in roman type within your formula.** **Mathmode rules still apply, so** `$\mathrm{some thing}$` **also produces** something**.**

- `\mathbb` **(blackboard bold) yields** $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$ **etc.;**

- `\mathbf` **gives boldface type;**

- `\mathcal` **offers** $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, $\ldots$, $\mathcal{Z}$ **('caligraphic');**

- `\mathfrak` **yields** $\mathfrak{A}$, $\mathfrak{B}$, $\mathfrak{C}$, $\ldots$, $\mathfrak{Z}$ **('Fraktur'— sometimes called 'old German' in English and 'Altenglisch' auf Deutsch).**

For the full range see your LaTeX manual.

Brackets (delimiters)

Many kinds and sizes of brackets (also known as 'delimiters') are available in mathmode.

- the obvious ones: ( ), [ ], | | are typed as you would expect;
- to get braces { } you type \{ \}.
- to get angle brackets $\langle$ $\rangle$ type `\langle` `\rangle`.

All such brackets can be made in various larger-than-normal sizes if preceded with `\big`, `\Big`, `\bigg` or `\Bigg`. Thus

$$\texttt{\$\textbackslash Bigg(\textbackslash bigg(\textbackslash Big(\textbackslash big((x)\textbackslash big)\textbackslash Big)\textbackslash bigg)\textbackslash Bigg)\$}$$

produces $\Bigg(\bigg(\Big(\big((x)\big)\Big)\bigg)\Bigg)$. (I have never understood why `\big` produces no noticeable enlargement.)

Making brackets grow, I

We also have `\left` and `\right`. Placed in front of delimiters they produce growth to accommodate the bracketed formula. Thus

```
$\left(\left(\left(\left(\left((x^2)^2
   \right)^2\right)^2\right)^2\right)^2\right)^2$
```

produces $\left(\left(\left(\left(\left(\left((x^2)^2\right)^2\right)^2\right)^2\right)^2\right)^2\right)^2$.

Making brackets grow, II

Once in a while one wants an unpaired bracket that grows, but `\left` and `\right` must be used in pairs. Then `\left bracket` is paired with `\right.` to keep the balance.

One situation where this is very useful is with arrays made with the array environment—see your LaTeX manual. An example:

```
$\displaystyle
  |x| := \left\{
  \begin{array}{rl}
  x &\mbox{if $x \geq 0$},\\
  -x &\mbox{if $x \leq 0$}.
  \end{array}
  \right.$
```

produces
$$|x| := \left\{ \begin{array}{rl} x & \text{if } x \geqslant 0, \\ -x & \text{if } x \leqslant 0. \end{array} \right.$$

Controlling space

Spacing in T$_E$X and L^{A}T$_E$X: learn about hboxes, hglue, vboxes, vglue. Read a manual.

Measures of length: $x$pt (points, 72 per inch), $x$mm, $x$cm or $x$in (inches). Here x is positive or negative; need not be integral.

Horizontal space commands: `\hglue15pt`, `\hspace{15pt}`, `\kern15pt`, and perhaps more. Not quite interchangeable: I need first for space at start of a line; I use second in most other contexts; I use third in mathmode (mostly). But my usage may be no more than habit. To learn their proper use you should read the manuals and you should experiment.

More on horizontal space

Two special spaces that can be useful, especially in mathmode:
`\quad` inserts a space as wide as the current type size; `\qquad` twice
as big.

There are `\hfil` and `\hfill` to fill out a line with space.
In particular, `\hfill\break` is a sort of 'return' that starts a new
line without trying to justify the current line.

Vertical space

For vertical space we have `\vglue15pt, \vspace{15pt}`

Also commands to give space before a new paragraph: `\smallskip,`
`\medskip` and `\bigskip`.
`\bigskip` gives a vspace that is as deep as a single line of type (the
vertical equivalent of the hspace `\quad`).
It is what usually appears before and after a theorem, a lemma, a
quotation, etc., when the appropriate environment is invoked.

We also have `\vfil` and `\vfill` (which differ a bit—see a manual
for details). These fill out a page and help control page-breaks.
In fact, however, we have `\newpage` which is equivalent to
`\vfil\pagebreak`.

Paragraph spacing

Spacing of paragraphs has its own commands.

Paragraph indent `\parindent` is set to 15pt by default in most document classes, I think.

For a paragraph that is not to be indented use command `\noindent`

Vertical space between paragraphs: default is none.
You can change that with `\setlength{\parskip}{x}`,
where x must be a length such as 3pt.

Printers' rules

A 'rule' in printers' language is a printed line.

The command `\rule[r pt]{w pt}{h pt}` produces a 'line' that is raised r pt above the baseline (below if r is negative), is w pt wide, and is h pt high.

`\rule[5pt]{10pt}{1pt}`, `\rule[-5pt]{1pt}{10pt}`, `\rule{10pt}{10pt}` give, respectively, $-$, $|$, and $\blacksquare$.

If w, the width, is set as 0 pt (or 0 cm, etc.) then the rule is invisible, but still has the height and the depth you have set.

Example: `$\left(1\rule[-5pt]{0pt}{20pt}\right)$` produces $\left(1\right)$.

The exterior: formatting a document in LaTeX

Return to the LaTeX document structure described in Lecture 1:

```
\documentclass[X]{Y}

   [Preamble]

\begin{document}

    Your text

\end{document}
```

to consider preamble and main text in a little more detail.

Already said: preamble might begin with `\usepackage` commands. At first you'll find a couple of font packages and a graphics package suffice. Once you have become a TeX expert you can be more adventurous.

More preamble

My own preamble then has lines that override the default for the 'article' class:

```
\setlength{\textwidth}{422pt}

\setlength{\textheight}{700pt}

\setlength{\topmargin}{-30pt}

\setlength{\oddsidemargin}{12pt}

\setlength{\parskip}{1pt plus 1pt}

\setlength{\mathsurround}{1pt}

\renewcommand{\baselinestretch}{1.05}
```

Explanations (where not self-explanatory) on next screen. Experiment to find the settings you like—they may be different for different documents to ensure good page-breaks.

Some explanations

Both `\oddsidemargin` and `\evensidemargin` exist in LaT_EX: some documents may need a different left margin for odd-numbered pages (rectos) and even-numbered pages (versos).

Command `\parskip`, paragraph skip, is the extra space (or none) between paragraphs; specification `{1pt plus 1pt}` says a 1pt space is norm, may expand up to 2 points.

Command `\mathsurround` puts a little extra space fore and aft of in-line formulae.

Command `\baselinestretch` expands (or contracts if the parameter < 1) line-spacing between lines. For example `\renewcommand{\baselinestretch}{1.09}` increases by 9%.

More preamble lines, I

Next my preamble has the following three lines:

```
\newtheorem{theorem}{\indent\sc Theorem}[section]

\newtheorem{lemma}{\indent\sc Lemma}[section]

\newcommand{\thmstop}{\hglue-6pt.\kern6pt}
```

The first defines the theorem environment for me and customises it just a little. First input parameter `theorem` or `lemma` names the environment; then it is called with the command

```
\begin{theorem} . . . \end{theorem}

\begin{lemma} . . . \end{lemma}
```

respectively.

More preamble lines, II

`{\indent\sc Theorem}`, `{\indent\sc Lemma}` **are the structure names, here Theorem or Lemma, printed with a number to form the name of the theorem, the lemma, etc. Default is non-indented and boldface, which I dislike, so I have customised it.**

Option `[section]` **says to print the number as** $a.b$**: here** a **is current section-number,** b **is number of theorem in section.**

I hold that there should be full stop after theorem number. I use last of the three lines shown above as in the example:

```
\begin{theorem}\thmstop
  All groups of order $27$ are green.
\end{theorem}
```

which yields

Theorem 2. All groups of order 27 **are green.**

Yet more preamble

After those commands my preamble has a few definitions of notation. Examples:

```
\newcommand{\Half}{{1\over2}}

\newcommand{\Nats}{\mathbb N}

\newcommand{\Rats}{\mathbb Q}

\newcommand{\Reals}{\mathbb R}

\newcommand{\Sym}{\mathrm{Sym}\kern1pt}
```

Instead of putting such definitions into the preamble one could put them into a separate file called, say, `macros.tex`.
Then call this file (in preamble or at start of main text) with the command `\include{macros}`.

Note: The modern way to get $\frac{1}{2}$ is `\frac{1}{2}`

The main text

Your main text comes between `\begin{document}` **and** `\end{document}`.

**For a short article it is not unreasonable to type it in directly.
A longer article, a dissertation, or a book, however, is probably
best split into sections or chapters, each typed into its own file
summoned by the command** `\include{filename}`.

Use file-names like `SectIntro.tex` **or** `ChIntegration.tex` **(summoned
by the command** `\include{SectIntro}` **or** `\include{ChIntegration}`**).
Best not to use** `Sect1.tex` **or** `Ch5.tex`**, because section and chapter
numbers tend to change as a long document grows.**

Note the LaTeX commands `\chapter, \section, \subsection`.
They produce automatic numbering.

The bibliography

Finally, of course, comes the bibliography, of great importance for any work of scholarship.

A convenient way of producing it was discussed earlier.

Next week: Style in T_EX and $L\!AT_EX$

An Introduction to TeX and LaTeX:

Lecture 3: Sense and sensibility

Tuesday 25 October 2016

- A remark that is nothing to do with $T_{\!E}\!X$

- Hyphens and dashes

- Stops

- Correct use of symbols

- Some suggestions about style

Nothing to do with T_EX

Good practice: ensure that all your work has

- title,

- author (yourself),

- date.

For a long work, such as a dissertation, a version number can also help you to know where you have reached with your revisions.

Hyphen and dashes

Do not confuse or misuse hyphen -, en-dash –, em-dash ——.

Hyphens are used if a word has to be broken at line end;
also used in compound adjectival phrases made from nouns as in
'a white-knuckle ride', 'a red-nosed reindeer';
also in double-barrelled names such as Heath-Brown and
Swinnerton-Dyer.

En-dash is used for ranges such as 1984–2009 or pp. 123–254.
Type two adjacent hyphens -- in $\TeX$.

Em-dash is a form of punctuation used——much as parentheses
are——to mark off certain kinds of subordinate clauses.
Type three adjacent hyphens --- in $\TeX$.

More on en-dash

Mathematicians have acquired a habit of using en-dash to create eponymous names for theorems, conjectures, and the like: the Schröder–Bernstein Theorem; the Cantor–Schröder–Bernstein Theorem.

[To me this seems daft: to use what looks like a minus sign instead of & or $+$.]

In recent years I have seen authors go a step further and use a construction that makes no sense at all: 'the theorem of Cantor–Bernstein' or 'the conjecture of Birch–Swinnerton-Dyer' and even 'Birch–Swinnerton-Dyer's conjecture'.
In this context the conjunction MUST be 'and' or '&'.

Stops

Full stop (period in American English) and abbreviation stop are typographically different.

Typographical convention expects extra space after a full stop and before the start of the following sentence, and T_EX inserts such space automatically.

For an abbreviation stop, as seen in the works of D. E. Knuth for example (compare D. E. Knuth), we type .\ or we use protected space (tie) by typing .\~ if we wish to inhibit line-breaking after the stop. Or use thinspace by typing \,

Correct use of symbols, I

In mathmode we have all the correct symbols for mathematics: do not use the wrong ones. Compare

$$| < a, b | a^2 = b^2 = (ab)^3 = 1 > | = 6 \quad \textbf{and} \quad e^{iy} = cosy + isiny.$$

with

$$|\langle a, b \mid a^2 = b^2 = (ab)^3 = 1 \rangle| = 6 \quad \textbf{and} \quad \mathrm{e}^{\mathrm{i}y} = \cos y + \mathrm{i}\sin y.$$

5

Correct use of symbols, II

- **Never use** $<\ >$ **as angle brackets. As binary relation symbols they come with spaces fore and aft. Use** `\langle` `\rangle` **which give the genuine brackets** $\langle\ \rangle$.

- **Use** `\mid` **for 'such that' in contexts such as** $\{x \in \mathbb{R} \mid x^2 < 2\}$: **it is spaced correctly, whereas vertical bar is not. (Or change your style and use colon** $\{x \in \mathbb{R} : x^2 < 2\}$**)**

Correct use of symbols, III

- **Functions such as log, exp, cos, sin, max, min, are always printed in roman type in formulae. Use** `\log`, `\exp`, `\cos`, `\sin`, `\max`, `\min`, **and other such commands that are built into T$_{\text{E}}$X.**

- **Functions like Hom, Aut, End, Tor, Sym, Alt should likewise be printed in roman type in formulae, as in**
$$\mathrm{Aut}\left(\mathrm{Alt}\left(n\right)\right) \cong \mathrm{Sym}\left(n\right).$$
For most of these T$_{\text{E}}$X does not have built in commands. Either use `\mathrm` **each time or make definitions such as**
`\newcommand{\Aut}{\mathrm{Aut}}`.

Style, I

A selection of the advice written in about 1980 by the London Mathematical Society editors of the time, much of it still to be found on the LMS publications website:

- Mathematics should be written in grammatically correct language and should be properly punctuated, even in sentences that include formulae or displayed material.

- Words such as 'assume', 'suppose', 'show' and 'imply' should usually be followed by 'that'.

- Where 'if' introduces a conditional clause, it should usually be followed by 'then', as in 'if $x = 3$ then $y = 4$'.
  This is essential where omitting 'then' would result in juxtaposition of formulae, as in 'if $x = 3$, $y = 4$'.

Style, II

Adjectives should not usually be used as nouns.

For example, 'a unitary' should not be used when what is meant is 'a unitary operator'.

It is particularly obnoxious to use proper names such as Frobenius, Sylow as ordinary nouns by taking their eponymous adjectival uses and dropping the noun as in
'(the) Frobenius' for 'the Frobenius automorphism' or
'a Sylow' for 'a Sylow subgroup'.

Even worse is to use a proper name as an adjective without its noun, as in 'we see that (s_n) is Cauchy and therefore converges'.

Style, III

Sentences should begin with words rather than symbols; it is particularly bad to start a sentence with a lower case symbol.

This holds especially when the preceding sentence ends with a formula or symbol, so that two formulae could apparently coalesce to make one which, at first reading, is nonsense.

Formulae should never be separated by punctuation marks only, except in lists.

Try to avoid having quotation marks, reference citations or footnote symbols adjacent to formulae.

Never use 'apostrophe s' for plurals—as in x_i's.
For the possessive with symbols use 'of'.

Style, IV

Use abbreviation sparingly. Mathematical writing is already very concentrated.

Do not use 'e.g.', 'i.e.', 's.t.' adjacent to formulae.

Never use symbols such as $\Rightarrow$, $\forall$, $\exists$ as lazy abbreviations in text.
The abbreviation 'iff' is best avoided in print.
The full form 'if and only if' is easier to read (and looks less like a misprint).

Avoid the use of symbols such as $=$, $<$ as abbreviations in text.
The scope of such binary relation symbols should be made clear: usages such as 'The number of prime divisors of $30 = 3$' are unnecessarily disturbing.

Enough already!

Enjoy making professional mathematical text.

May your essays, reports, dissertations, articles give pleasure to you and to your readers.