Getting Started with LaTeX

D. R. Wilkins

February 12, 1996

1 Introduction to LaTeX

1.1 What is LaTeX?

TeX is a computer program for typesetting documents. It takes a computer file, prepared according to the rules of TeX, and converts it to a form that may be printed on a high-quality printer, such as a laser writer, to produce a printed document of a quality comparable with good quality books and journals. Simple documents, which do not contain mathematical formulae or tables may be produced very easily: effectively all one has to do is to type the text straight in (though observing certain rules relating to quotation marks and punctuation dashes). Typesetting mathematics is somewhat more complicated, but even here TeX is comparatively straightforward to use when one considers the complexity of some of the formulae that it has to produce and the large number of mathematical symbols which it has to produce.

There are several 'dialects' of TeX, all based on the version of TeX created by D. E. Knuth which is known as Plain TeX. LaTeX (created by L. B. Lamport) is one of these 'dialects'. It is particularly suited to the production of long articles and books, since it has facilities for the automatic numbering of chapters, sections, theorems, equations etc., and also has facilities for cross-referencing. It is probably the most suitable version of TeX for beginners to use, and the manual "LaTeX—User's Guide and Reference Manual" is certainly easier to read than the corresponding manual (i.e., "The TeXbook") for Plain TeX.

1.2 A Typical LaTeX Input File

In order to produce a document using LaTeX, we must first create a suitable *input file* on the computer. We apply the TeX program to the input file and then use the printer to print out the so-called 'DVI' file produced by the TeX program (after first using another program to translate the 'DVI' file into a form that the printer can understand). Here is an example of a typical LaTeX input file:

```
\documentstyle[tcda,12pt]{article}
\begin{document}

The foundations of the rigorous study of {\it analysis}
were laid in the nineteenth century, notably by the
mathematicians Cauchy and Weierstrass. Central to the
study of this subject are the formal definitions of
{\it limits} and {\it continuity}.

Let $D$ be a subset of $\bf R$ and let
$f \colon D \to {\bf R}$ be a real-valued function on
$D$. The function $f$ is said to be {\it continuous} on
$D$ if, for all $\epsilon > 0$ and for all $x \in D$,
there exists some $\delta > 0$ (which may depend on $x$)
such that if $y \in D$ satisfies
\[ |y - x| < \delta \]
then
\[ |f(y) - f(x)| < \epsilon. \]

One may readily verify that if $f$ and $g$ are continuous
functions on $D$ then the functions $f+g$, $f-g$ and
$f.g$ are continuous. If in addition $g$ is everywhere
non-zero then $f/g$ is continuous.

\end{document}
```

When we apply TeX to these paragraphs we produce the text

The foundations of the rigorous study of *analysis* were laid in the nineteenth century, notably by the mathematicians Cauchy and Weierstrass. Central to the study of this subject are the formal definitions of *limits* and *continuity*.

Let D be a subset of $\mathbf{R}$ and let $f\colon D \to \mathbf{R}$ be a real-valued function on D. The function f is said to be *continuous* on D if, for all $\epsilon > 0$ and for all $x \in D$, there exists some $\delta > 0$ (which may depend on x) such that if $y \in D$ satisfies

$$|y - x| < \delta$$

then

$$|f(y) - f(x)| < \epsilon.$$

One may readily verify that if f and g are continuous functions on D then the functions $f + g$, $f - g$ and $f.g$ are continuous. If in addition g is everywhere non-zero then f/g is continuous.

This example illustrates various features of LaTeX. Note that the lines

```
\documentstyle[tcda,12pt]{article}
\begin{document}
```

are placed at the beginning of the input file. These are followed by the main body of the text, followed by the concluding line

```
\end{document}
```

Note also that, although most characters occurring in this file have their usual meaning, yet there are special characters such as `\`, `$`, `{` and `}` which have special meanings within TeX. Note in particular that there are sequences of characters which begin with a 'backslash' `\` which are used to produce mathematical symbols and Greek letters and to accomplish tasks such as changing fonts. These sequences of characters are known as *control sequences*.

1.3 Characters and Control Sequences

We now describe in more detail some of the features of TeX illustrated in the above example.

Most characters on the keyboard, such as letters and numbers, have their usual meaning. However the characters

```
\ { } $ ^ _ % ~ # &
```

are used for special purposes within TeX. Thus typing one of these characters will not produce the corresponding character in the final document. Of course these characters are very rarely used in ordinary text, and there are methods of producing them when they are required in the final document.

In order to typeset a mathematical document it is necessary to produce a considerable number of special mathematical symbols. One also needs to be able to change fonts. Also mathematical documents often contain arrays of numbers or symbols (matrices) and other complicated expressions. These are produced in TeX using *control sequences*. Most control sequences consist of a backslash `\` followed by a string of (upper or lower case) letters. For example, `\alpha`, `\it`, `\sum` and `\TeX` are control sequences.

In the example above we used the control sequences `\it` and `\bf` to change the font to *italic* and **boldface** respectively. Also we used the control sequences `\to`, `\in`, `\delta` and `\epsilon` to produce the mathematical symbols $\to$ and $\in$ and the Greek letters δ and ϵ.

> There is another variety of control sequence which consists of a backslash followed by a *single* character that is not a letter. Examples of control sequences of this sort are `\{`, `\"` and `\$`.

The special characters `{` and `}` are used for *grouping* purposes. Everything enclosed within matching pair of such brackets is treated as a single unit. We have applied these brackets in the example above whenever we changed fonts. We shall see other instances where one needs to use `{` and `}` in TeX to group words and symbols together (e.g., when we need to produce superscripts and subscripts which contain more than one symbol).

The special character `$` is used when one is changing from ordinary text to a mathematical expression and when one is changing back to ordinary text. Thus we used

```
for all $\epsilon > 0$ and for all $x \in D$,
```

to produce the phrase

> for all $\epsilon > 0$ and for all $x \in D$,

in the example given above. Note also that we used `\[` and `\]` in the example above to mark the beginning and end respectively of a mathematical formula that is displayed on a separate line.

The remaining special characters

```
^ _ % ~ # &
```

have special purposes within TeX that we shall discuss later.

2 Producing Simple Documents using LaTeX

2.1 Producing a LaTeX Input File

We describe the structure of a typical LaTeX input file.

The first line of the input file should consist of a `\documentstyle` command. The recommended such `\documentstyle` command for mathematical articles and similar documents has the form

```
\documentstyle[tcda,12pt]{article}
```

(You do not have to worry about what this command means when first learning to use LaTeX: its effect is to ensure that the final document is correctly positioned on A4 size paper and that the text is of a size that is easy to read.) There are variants of this `\documentstyle` command which are appropriate for letters or for books.

The `documentstyle` command may be followed by certain other optional commands, such as the `\pagestyle` command. It is not necessary to find out about these commands when first learning to use LaTeX.

After the `\documentstyle` command and these other optional commands, we place the command

```
\begin{document}
```

This command is then followed by the main body of the text, in the format prescribed by the rules of LaTeX.

Finally, we end the input file with a line containing the command

```
\end{document}
```

2.2 Producing Ordinary Text using LaTeX

To produce a simple document using LaTeX one should create a LaTeX input file, beginning with a `\documentstyle` command and the `begin{document}` command, as described above. The input file should end with the `\end{document}` command, and the text of the document should be sandwiched between the `\begin{document}` and `\end{document}` commands in the manner described below.

If one merely wishes to type in ordinary text, without complicated mathematical formulae or special effects such as font changes, then one merely has to type it in as it is, leaving a completely blank line between successive paragraphs. You do not have to worry about paragraph indentation: TeX will automatically indent all paragraphs with the exception of the first paragraph of a new section (unless you take special action to override the conventions adopted by TeX)

For example, suppose that we wish to create a document containing the following paragraphs:

> If one merely wishes to type in ordinary text, without complicated mathematical formulae or special effects such as font changes, then one merely has to type it in as it is, leaving a completely blank line between successive paragraphs.
>
> You do not have to worry about paragraph indentation: all paragraphs will be indented with the exception of the first paragraph of a new section.
>
> One must take care to distinguish between the 'left quote' and the 'right quote' on the computer terminal. Also, one should use two 'single quote' characters in succession if one requires "double quotes". One should never use the (undirected) 'double quote' character on the computer terminal, since the computer is unable to tell whether it is a 'left quote' or a 'right quote'. One also has to take care with dashes: a single dash is used for hyphenation, whereas three dashes in succession are required to produce a dash of the sort used for punctuation—such as the one used in this sentence.

To create this document using LaTeX we use the following input file:

```
\documentstyle[tcda,12pt]{article}
```

```
\begin{document}

If one merely wishes to type in ordinary text, without
complicated mathematical formulae or special effects such
as font changes, then one merely has to type it in as it
is, leaving a completely blank line between successive
paragraphs.

You do not have to worry about paragraph indentation:
all paragraphs will be indented with the exception of
the first paragraph of a new section.

One must take care to distinguish between the `left quote'
and the `right quote' on the computer terminal.  Also, one
should use two `single quote' characters in succession if
one requires ``double quotes''.  One should never use the
(undirected) `double quote' character on the computer
terminal, since the computer is unable to tell whether it
is a `left quote' or a `right quote'.  One also has to
take care with dashes: a single dash is used for
hyphenation, whereas three dashes in succession are required
to produce a dash of the sort used for punctuation---such as
the one used in this sentence.

\end{document}
```

Having created the input file, one then has to run it through the LaTeX program and then print it out the resulting output file (known as a 'DVI' file).

2.3 Blank Spaces and Carriage Returns in the Input File

TeX treats the carriage return at the end of a line as though it were a blank space. Similarly TeX treats tab characters as blank spaces. Moreover, TeX regards a sequence of blank spaces as though it were a single space, and similarly it will ignore blank spaces at the beginning or end of a line in the input file. Thus, for example, if we type

```
This is
    a
        silly
  example   of   a
file with many spaces.



                   This is the beginning
of a new paragraph.
```

then we obtain

> This is a silly example of a file with many spaces.
> This is the beginning of a new paragraph.

It follows immediately from this that one will obtain the same results whether one types one space or two spaces after a full stop: TEX does not distinguish between the two cases.

Any spaces which follow a control sequence will be ignored by TEX.

> If you really need a blank space in the final document following whatever is produced by the control sequence, then you must precede this blank by a *backslash* \. Thus in order to obtain the sentence
>
> TEX is a very powerful computer typesetting program.
>
> we must type
>
> ```
> \TeX\ is a very powerful computer typesetting program.
> ```
>
> (Here the control sequence `\TeX` is used to produce the TEX logo.)
>
> In general, preceding a blank space by a backslash forces TEX to include the blank space in the final document.

As a general rule, you should never put a blank space after a left parenthesis or before a right parenthesis. If you were to put a blank space in these places, then you run the risk that TEX might start a new line immediately after the left parenthesis or before the right parenthesis, leaving the parenthesis marooned at the beginning or end of a line.

2.4 Quotation Marks

Single left and right quotation marks are produced by `` ` `` and `'` respectively. Double left and right quotation marks are produced by ``` `` ``` and `''` respectively. Thus

> “What did you do yesterday?” he asked.

is produced by typing

```
``What did you do yesterday?'' he asked.
```

You should never use the character `"` to produce quotation marks. This is because TeX has no way of knowing whether you want a left quote or a right quote if you do this.

> You can use the control sequences `\lq` and `\rq` in place of `` ` `` and `'`. This is useful if your keyboard does not have a `` ` `` character.
>
> Sometimes you need two quotation marks following one another, as in
>
> > “I regard computer typesetting as being reasonably ‘straightforward’ ” he said.
>
> The way to do this is to use the control sequence `\,` between the quotation marks. Thus one would type
>
> ```
> ``I regard computer typesetting as being reasonably
> `straightforward'\,'' he said.
> ```
>
> However this problem arises very rarely.

2.5 Dashes

TeX allows you to produce dashes of various length. Typing `-` by itself produces a hyphen, as in ‘double-quote’. Typing `--` produces a dash suitable for denoting a range of numbers, as in the phrase ‘on pages 155–159’, produced by typing

```
on pages 155--159.
```

Finally, typing `---` produces a punctuation dash—this is a dash such as the one in this sentence.

2.6 Section Headings in LaTeX

Section headings of various sizes are produced (in the *article* document style) using the commands `\section`,`\subsection` and `\subsubsection` commands. LaTeX will number the sections and subsections automatically. The title of the section should be surrounded by curly brackets and placed immediately after the relevant command. Thus if we type

```
\section{Section Headings}
We explain in this section how to obtain headings
for the various sections and subsections of our
document.

\subsection{Headings in the ‘article’ Document Style}
In the ‘article’ style, the document may be divided up
into sections, subsections and subsubsections, and each
can be given a title, printed in a boldface font,
simply by issuing the appropriate command.
```

then the title of the section and that of the subsection will be printed in a large boldface font, and will be numbered accordingly.

Other document styles (such as the *book* and *letter* styles) have other ‘sectioning’ commands available (for example, the *book* style has a `\chapter` command for beginning a new chapter).

> Sometimes one wishes to suppress the automatic numbering provided by LaTeX. This can be done by placing an asterisk before the title of the section or subsection. Thus, for example, the section numbers in the above example could be suppressed by typing

```
\section*{Section Headings}
We explain in this section how to obtain headings
for the various sections and subsections of our
document.

\subsection*{Headings in the ‘article’ Document Style}
In the ‘article’ style, the document may be divided up
into sections, subsections and subsubsections, and each
can be given a title, printed in a boldface font,
simply by issuing the appropriate command.
```

2.7 Changing Fonts

Fonts are changed using the control sequences `\rm`, `\sl`, `\it`, `\tt` and `\bf`.

`\rm` changes to the normal "roman" font:	Roman
`\sl` changes to a slanted roman font:	*Slanted*
`\it` changes to an italic font:	*Italic*
`\tt` changes to an "typewriter" font:	`Typewriter`
`\bf` changes to a boldface font:	**Boldface**

It is best to use the special characters `{` and `}` when changing fonts. One encloses the text whose font is to be changed within these curly brackets and places the font-changing control sequence immediately after the opening bracket `{`. Thus the text

> In this sentence we have *italicized* a few words, set others in *slanting type* or **boldface type**, and typeset others using a `'typewriter' font in which all the letters have a fixed width`.

is produced by typing

```
In this sentence we have {\it italicized\/} a few words, set
others in {\sl slanting type\/} or {\bf boldface type}, and
typeset others using a {\tt 'typewriter' font in which all
the letters have a fixed width}.
```

The control sequence `\/` produces the so-called *italic correction*. The use of this is recommended when changing back from an *italic* or *slanted* font into a roman or **boldface** font, in order to produce extra space to compensate for the way in which some *italic* and *slanted* letters lean into the following blank space. However this italic correction should not be used before a comma or a full stop.

LaTeX provides the control sequence `\em` for *emphasizing* text. This will set the emphasized text in italic, unless the surrounding text is already in italic, in which case the text will be set in ordinary roman font. Thus

```
Here is some {\em emphasized text with {\em emphasized}
words embedded in the {\em emphasized} text} too.
```

produces

> Here is some *emphasized text with* emphasized *words embedded in the* emphasized *text* too.

2.8 Accents and other Symbols used in Text

There are a variety of control sequences for producing accents. For example, the control sequence `\'{o}` produces an acute accent on the letter `o`. Thus typing

```
Se\'{a}n \'{O} Cinn\'{e}ide.
```

produces

Seán Ó Cinnéide.

Similarly we use the control sequence `` \` `` to produce the grave accent in 'algèbre' and we use `\"` to produce the umlaut in 'Universität'. A list of the accents provided by TeX is given in Appendix A.

The control sequences `\i` and `\j` produce dotless $\imath$ and $\jmath$. These are required when placing an accent on the letter. Thus ī is produced by typing `\={\i}`. There are also control sequences for ligatures and other special symbols used within text. These are listed in Appendix A.

2.9 Special Characters

The characters

```
# $ % & \ ^ _ { } ~
```

have special purposes within TeX. Thus they cannot be produced in the final document simply by typing them directly. On the rare occasions when one needs to use the special characters

$ % & _ { }

in the final document, they can be produced by typing the control sequences

```
\# \$ \% \& \_ \{ \}
```

respectively. However, somewhat more ingenuity is required to produce `\`, `^` and `~`.

3 Mathematical Formulae using LaTeX

3.1 Mathematics Mode

In order to obtain a mathematical formula using TeX, one must enter *mathematics mode* before the formula and leave it afterwards. Mathematical formulae can occur either embedded in text or else displayed on a separate line. When a formula occurs within the text of a paragraph one should place a `$` sign before and after the formula, in order to enter and leave mathematics mode. Thus to obtain a sentence like

> Let f be the function defined by $f(x) = 3x + 7$, and let a be a positive real number.

one should type

```
Let $f$ be the function defined by $f(x) = 3x + 7$, and
let $a$ be a positive real number.
```

In particular, note that even mathematical expressions consisting of a single character, like f and a in the example above, are placed within `$` signs. This is to ensure that they are set in italic type, as is customary in mathematical typesetting.

> LaTeX also allows you to use `\(` and `\)` to mark the beginning and the end respectively of a mathematical formula embedded in text. Thus
>
> Let f be the function defined by $f(x) = 3x + 7$.
>
> may be produced by typing
>
> ```
> Let \( f \) be the function defined by \( f(x) = 3x + 7 \).
> ```
>
> However this use of `\(...\)` is only permitted in LaTeX, whereas `$...$` is more standard, and is used in other dialects of TeX, such as Plain TeX and $\mathcal{AMS}$-TeX.

In order to obtain an mathematical formula or equation which is displayed on a line by itself, one places `\[` before and `\]` after the formula. Thus to obtain

The product of two first degree polynomials is a quadratic polynomial. For example, if $f(x) = 3x + 7$ and $g(x) = x + 4$ then

$$f(x)g(x) = 3x^2 + 19x + 28.$$

The converse does not hold for polynomials over the field of real numbers. However if we consider polynomials over the complex field then every polynomial factorizes as a product of first degree polynomials, by the Fundamental Theorem of Algebra.

one would type

```
The product of two first degree polynomials is a quadratic
polynomial. For example, if $f(x) = 3x + 7$ and $g(x) = x + 4$
then
\[ f(x)g(x) = 3x^2 + 19x +28. \]
The converse does not hold for polynomials over the field of
real numbers. However if we consider polynomials over the
complex field then every polynomial factorizes as a product
of first degree polynomials, by the Fundamental Theorem of Algebra.
```

It is also possible to use `$$` in place of both `\[` and `\]` to mark the beginning and end of a displayed mathematical equation. Thus

$$f(x)g(x) = 3x^2 + 19x + 28.$$

can be produced by typing

```
$$f(x)g(x) = 3x^2 + 19x +28.$$
```

Indeed this the method of producing displayed equations in dialects of TeX other than LaTeX, such as Plain TeX and $\mathcal{AMS}$-TeX.

LaTeX provides facilities for the automatic numbering of displayed equations. If you want an numbered equation then you use `\begin{equation}` and `\end{equation}` instead of using `\[` and `\]` . Thus

```
The product of two first degree polynomials is a quadratic
polynomial. For example, if $f(x) = 3x + 7$ and $g(x) = x + 4$
then
\begin{equation}
f(x)g(x) = 3x^2 + 19x +28.
\end{equation}
```

produces

> The product of two first degree polynomials is a quadratic polynomial. For example, if $f(x) = 3x + 7$ and $g(x) = x + 4$ then
>
> $$f(x)g(x) = 3x^2 + 19x + 28. \qquad (1)$$

3.2 Characters in Mathematics Mode

All the characters on the keyboard have their standard meaning in mathematics mode, with the exception of the characters

```
# $ % & ~ _ ^ \ { } '
```

Letters are set in italic type. In mathematics mode the character `'` has a special meaning: typing `$f' + g''$` produces $f' + g''$. When in mathematics mode the spaces you type between letters and other symbols do not affect the spacing of the final result, since TEX determines the spacing of characters in formulae by its own internal rules. Thus `$x ( y + z )$` and `$x(y+z)$` both produce $x(y+z)$. You can also type carriage returns where necessary in your input file (e.g., if you are typing in a complicated formula with many Greek characters and funny symbols) and this will have no effect on the final result if you are in mathematics mode.

> To obtain the characters
>
> $$\# \quad \$ \quad \% \quad \& \quad \_ \quad \{ \quad \}$$
>
> in mathematics mode, one should type
>
> ```
> \# \$ \% \& _ \{ \} .
> ```
>
> To obtain $\backslash$ in mathematics mode, one may type `\backslash`.

3.3 Subscripts and Superscripts

Subscripts and superscripts are obtained using the special characters `_` and `^` respectively. Thus the expression $t^3 + x_1^2 - x_2$ is obtained by typing `$t^3 + x_1^2 - x_2$`. When the subscript or superscript consists of more than one character then the characters involved should be enclosed in curly brackets. Thus to obtain the expression $u_{i,j}^{12}$ one would type `$u_{i,j}^{12}$`.

It is immaterial whether one specifies the subscript before the superscript or vica versa. Thus `$u_1^2$` and `$u^2_1$` both produce u_1^2. However TeX does not like it if you type `$s_n_j$` since this could be interpreted either as s_{nj} or as s_{n_j}. The first of these alternatives is obtained by typing `$s_{n j}$`, the second by typing `$s_{n_j}$`. A similar remark applies to superscripts. Incidentally, the second alternative illustrates the fact that one can obtain subscripts (or superscripts) on subscripts (or superscripts). However one should not go beyond this to try to obtain triple subscripts.

It is sometimes necessary to obtain expressions such as $R_i{}^j{}_{kl}$ in which the exact positioning of the subscripts and superscripts is important (e.g., in papers on general relativity and tensor analysis). The way this is done is to include the 'empty group' `{}` at the appropriate places to enable the superscripts and subscripts to be aligned correctly. Thus to obtain $R_i{}^j{}_{kl}$ one would type `$R_i{}^j{}_{kl}$`.

3.4 Greek Letters

Greek letters are produced in mathematics mode by preceding the name of the letter by a backslash `\`. Thus the Greek letters alpha (α), pi (π) and chi (χ) are obtained by typing `\alpha`,`\pi` and `\chi` respectively. Thus the sentence

The area A of a circle of radius r is given by the formula $A = \pi r^2$.

is obtained by typing

```
The area~$A$ of a circle of radius~$r$ is given by the
formula $A = \pi r^2$.
```

Upper case Greek letters are obtained by making the first character of the name upper case. Thus Γ,Φ and Λ are obtained by typing `\Gamma`,`\Phi` and `\Lambda`.

There is no special command for omicron: just use `o`.

Some Greek letters occur in variant forms. The variant forms are obtained by preceding the name of the Greek letter by 'var'. The following table lists the usual form of these letters and the variant forms:-

ϵ	`\epsilon`	ε	`\varepsilon`
θ	`\theta`	ϑ	`\vartheta`
π	`\pi`	ϖ	`\varpi`
ρ	`\rho`	ϱ	`\varrho`
σ	`\sigma`	ς	`\varsigma`
ϕ	`\phi`	φ	`\varphi`

3.5 Mathematical Symbols

There are numerous mathematical symbols that can be used in mathematics mode. These are obtained by typing an appropriate control sequence. These are listed in Appendix B. For example `\neq`, `\leq` and `\geq` produce $\neq$, $\leq$ and $\geq$ respectively, `\infty` produces ∞, `\times` and `\div` produce $\times$ and $\div$, both `\to` and `\rightarrow` produce $\rightarrow$, `\in` produces $\in$, `\cup`, `\cap`, `\setminus` and `\subset` produce $\cup$,$\cap$, $\setminus$ and $\subset$ respectively. The list seems endless.

3.6 Changing Fonts in Mathematics Mode

The *math italic* font is automatically used in mathematics mode unless you explicitly change the font. The rules for changing the font in mathematics mode are rather different to those applying when typesetting ordinary text. Firstly, any change only applies to the single character or symbol that follows. Secondly, to change a character to the roman font, the control sequence `\mathrm` must be used (rather than `\rm`). Thirdly to change a character to the $\boldsymbol{boldface}$ font the control sequence `\boldsymbol` must be used, *and in addition the* `amsbsy` *style must be included in the* `\documentstyle` *heading.*

There is also a 'calligraphic' font available in mathematics mode. This is obtained using the control sequence `\cal`. *This font can only be used for uppercase letters.* These calligraphic letters have the form

$$\mathcal{ABCDEFGHIJKLMNOPQRSTUVWXYZ}.$$

The following example shows how fonts are changed in an example involving mathematics. To obtain

Let $\boldsymbol{u}$,$\boldsymbol{v}$ and $\boldsymbol{w}$ be three vectors in $\boldsymbol{R}^3$. The volume V of the parallelepiped with corners at the points $\boldsymbol{0}$,$\boldsymbol{u}$,$\boldsymbol{v}$, $\boldsymbol{w}$,$\boldsymbol{u}+\boldsymbol{v}$, $\boldsymbol{u}+\boldsymbol{w}$,$\boldsymbol{v}+\boldsymbol{w}$ and $\boldsymbol{u}+\boldsymbol{v}+\boldsymbol{w}$ is given by the formula

$$V = (\boldsymbol{u} \times \boldsymbol{v}) \cdot \boldsymbol{w}.$$

one would type

```
Let $\boldsymbol u$,$\boldsymbol v$ and $\boldsymbol w$ be three vectors
in ${\boldsymbol R}^3$. The volume~$V$ of the parallelepiped with corners
at the points $\boldsymbol{0}$,$\boldsymbol{u}$,$\boldsymbol{v}$,
$\boldsymbol{w}$,$\boldsymbol{u}+\boldsymbol{v}$,
$\boldsymbol{u}+\boldsymbol{w}$,$\boldsymbol{v}+\boldsymbol{w}$
and $\boldsymbol{u}+\boldsymbol{v}+\boldsymbol{w}$
is given by the formula
\[ V = (\boldsymbol{u} \times \boldsymbol{v}) . \boldsymbol{w}. \]
```

3.7 Standard Functions and Embedded Text

The names of certain standard functions and abbreviations are obtained by typing a backlash \ before the name. The complete list in TEX is as follows:-

`\arccos`	`\cos`	`\csc`	`\exp`	`\ker`	`\limsup`	`\min`	`\sinh`
`\arcsin`	`\cosh`	`\deg`	`\gcd`	`\lg`	`\ln`	`\Pr`	`\sup`
`\arctan`	`\cot`	`\det`	`\hom`	`\lim`	`\log`	`\sec`	`\tan`
`\arg`	`\coth`	`\dim`	`\inf`	`\liminf`	`\max`	`\sin`	`\tanh`

Names of functions and other abbreviations not in this list can be obtained by converting to the roman font. Thus one obtains $\mathrm{Aut}(V)$ by typing `${\rm Aut}(V)$`.

Note that if one were to type simply `$Aut(V)$` one would obtain $Aut(V)$, because TEX has treated `Aut` as the product of three quantities A,u and t and typeset the formula accordingly.

The recommended way to obtain ordinary text in displayed mathematical formulae is to use `\mbox`. Thus one obtains

$$M^\perp = \{f \in V' : f(m) = 0 \mbox{ for all } m \in M\}.$$

by typing

```
\[ M^\bot = \{ f \in V' : f(m) = 0 \mbox{ for all } m \in M \}. \]
```

Note the blank spaces before and after the words 'for all' in the above example. Had we typed

```
\[ M^\bot = \{ f \in V' : f(m) = 0 \mbox{for all} m \in M \}. \]
```

we would have obtained

$$M^\bot = \{ f \in V' : f(m) = 0 \mbox{for all} m \in M \}.$$

One can use `\hbox` as an alternative to `\mbox` in mathematical formulae. Indeed `\mbox` is specific to LaTeX, whereas `\hbox` is used in Plain TeX and in other dialects of TeX. Also `\hbox` and `\mbox` only differ in their behaviour when used to begin a paragraph of ordinary text, and in particular behave in an identical manner when used in a mathematical formula.

3.8 Fractions,Roots and Ellipsis

Fractions of the form

$$\frac{numerator}{denominator}$$

are obtained in LaTeX using the construction

`\frac{`*numerator*`}{`*denominator*`}`.

For example, to obtain

> The function f is given by
>
> $$f(x) = 2x + \frac{x - 7}{x^2 + 4}$$
>
> for all real numbers x.

one would type

```
The function $f$ is given by
\[ f(x) = 2x + \frac{x - 7}{x^2 + 4} \]
for all real numbers $x$.
```

To obtain square roots one uses the control sequence `\sqrt`. For example, $\sqrt{x^2 + y^2}$ is produced by typing `$\sqrt{x^2 + y^2}$`. In LaTeX, an nth root is produced using

```
\sqrt[n]{expression}.
```

Thus $\sqrt[3]{x + 3y}$ is produced in LaTeX by typing `$\sqrt[3]{x + 3y}$`

Ellipsis (three dots) is produced in mathematics mode using the control sequences `\cdots` and `\ldots`. A low ellipsis, such as $(x_1, x_2, \ldots, x_n)$, is produced by typing

```
$(x_1,x_2,\ldots ,x_n)$.
```

A centred ellipsis, such as $x_1 + x_2 + \cdots + x_n$ is produced by typing

```
$x_1 + x_2 + \cdots + x_n$.
```

3.9 Accents in Mathematics Mode

The control sequences `\underline`, `\overline`, `\hat`, `\check`, `\tilde`, `\acute`, `\grave`, `\dot`, `\ddot`, `\breve`, `\bar` and `\vec` produce underlining, overlining, and various accents, *but only in mathematics mode.* For example, $\tilde{c}$ is produced by `$\tilde{c}$`. The effect of these accents on the letter a is shown in the table below:

`$\underline{a}$`	$\underline{a}$
`$\overline{a}$`	$\overline{a}$
`$\hat{a}$`	$\hat{a}$
`$\check{a}$`	$\check{a}$
`$\tilde{a}$`	$\tilde{a}$
`$\acute{a}$`	$\acute{a}$
`$\grave{a}$`	$\grave{a}$
`$\dot{a}$`	$\dot{a}$
`$\ddot{a}$`	$\ddot{a}$
`$\breve{a}$`	$\breve{a}$
`$\bar{a}$`	$\bar{a}$
`$\vec{a}$`	$\vec{a}$

You should bear in mind that when a character is underlined in a mathematical manuscript then it is normally typeset in bold face without any underlining. Underlining is used very rarely in print.

The control sequences such as `\'` and `\"`, used to produce accents in ordinary text, may not be used in mathematics mode.

3.10 Brackets and Norms

The frequently used left delimiters include (, [and {, which are obtained by typing `(`, `[` and `\{` respectively. The corresponding right delimiters are of course),] and }, obtained by typing `)`, `]` and `\}`. In addition $|$ and $\|$ are used as both left and right delimiters, and are obtained by typing `|` and `\|` respectively. For example, we obtain

> Let X be a Banach space and let $f \colon B \to \mathbf{R}$ be a bounded linear functional on X. The *norm* of f, denoted by $\|f\|$, is defined by
>
> $$\|f\| = \inf\{ K \in [0,+\infty) : |f(x)| \leq K \|x\| \mbox{ for all } x \in X \}.$$

by typing

```
Let $X$ be a Banach space and let $f \colon B \to {\bf R}$
be a bounded linear functional on $X$. The {\it norm} of
$f$, denoted by $\|f\|$, is defined by
\[ \|f\| = \inf \{ K \in [0,+\infty) :
          |f(x)| \leq K \|x\| \mbox{ for all } x \in X \}. \]
```

Larger delimiters are sometimes required which have the appropriate height to match the size of the subformula which they enclose. Consider, for instance, the problem of typesetting the following formula:

$$f(x,y,z) = 3y^2 z \left(3 + \frac{7x+5}{1 + y^2} \right).$$

The way to type the large parentheses is to type `\left(` for the left parenthesis and `\right)` for the right parenthesis, and let TEX do the rest of the work for you. Thus the above formula was obtained by typing

```
\[ f(x,y,z) = 3y^2 z \left( 3 + \frac{7x+5}{1 + y^2} \right). \]
```

If you type a delimiter which is preceded by `\left` then TEX will search for a corresponding delimiter preceded by `\right` and calculate the size of the delimiters required to enclose the intervening subformula. One is allowed to balance a `\left(` with a `\right]` (say) if one desires: there is no reason why the enclosing delimiters have to have the same shape. One may also nest pairs of delimiters within one another: by typing

```
\[ \left| 4 x^3 + \left( x + \frac{42}{1+x^4} \right) \right|. \]
```

we obtain

$$\left|4x^3 + \left(x + \frac{42}{1+x^4}\right)\right|.$$

By typing `\left.` and `\right.` one obtains *null delimiters* which are completely invisible. Consider, for example, the problem of typesetting

$$\left.\frac{du}{dx}\right|_{x=0}.$$

We wish to make the vertical bar big enough to match the derivative preceding it. To do this, we suppose that the derivative is enclosed by delimiters, where the left delimiter is invisible and the right delimiter is the vertical line. The invisible delimiter is produced using `\left.` and thus the whole formula is produced by typing

```
\[ \left. \frac{du}{dx} \right|_{x=0}. \]
```

3.11 Multiline Formulae in LaTeX

Consider the problem of typesetting the formula

$$\begin{aligned} \cos 2\theta &= \cos^2\theta - \sin^2\theta \\ &= 2\cos^2\theta - 1. \end{aligned}$$

It is necessary to ensure that the = signs are aligned with one another. In LaTeX, such a formula is typeset using the `eqnarray*` environment. The above example was obtained by typing the lines

```
\begin{eqnarray*}
\cos 2\theta & = & \cos^2 \theta - \sin^2 \theta \\
             & = & 2 \cos^2 \theta - 1.
\end{eqnarray*}
```

Note the use of the special character `&` as an it alignment tab. When the formula is typeset, the part of the second line of the formula beginning with an occurrence of `&` will be placed immediately beneath that part of the first line of the formula which begins with the corresponding occurrence of `&`. Also `\\` is used to separate the lines of the formula.

Although we have placed corresponding occurrences of `&` beneath one another in the above example, it is not necessary to do this in the input file. It was done in the above example merely to improve the appearance (and readability) of the input file.

The more complicated example

> If $h \leq \frac{1}{2}|\zeta - z|$ then
> $$|\zeta - z - h| \geq \frac{1}{2}|\zeta - z|$$
> and hence
> $$\begin{aligned} \left| \frac{1}{\zeta - z - h} - \frac{1}{\zeta - z} \right| &= \left| \frac{(\zeta - z) - (\zeta - z - h)}{(\zeta - z - h)(\zeta - z)} \right| \\ &= \left| \frac{h}{(\zeta - z - h)(\zeta - z)} \right| \\ &\leq \frac{2|h|}{|\zeta - z|^2}. \end{aligned}$$

was obtained by typing

```
If $h \leq \frac{1}{2} |\zeta - z|$ then
\[ |\zeta - z - h| \geq \frac{1}{2} |\zeta - z| \]
and hence
\begin{eqnarray*}
\left| \frac{1}{\zeta - z - h} - \frac{1}{\zeta - z} \right|
& = & \left|
\frac{(\zeta - z) - (\zeta - z - h)}{(\zeta - z - h)(\zeta - z)}
\right| \\  & = &
\left| \frac{h}{(\zeta - z - h)(\zeta - z)} \right| \\
  & \leq & \frac{2 |h|}{|\zeta - z|^2}.
\end{eqnarray*}
```

The asterisk in `eqnarray*` is put there to suppress the automatic equation numbering produced by LaTeX. If you wish for an automatically numbered multiline formula, you should use `\begin{eqnarray}` and `\end{eqnarray}`.

3.12 Matrices and other arrays in LaTeX

Matrices and other arrays are produced in LaTeX using the **array** environment. For example, suppose that we wish to typeset the following passage:

The *characteristic polynomial* $\chi(\lambda)$ of the 3×3 matrix

$$\left(\begin{array}{ccc} a & b & c \\ d & e & f \\ g & h & i \end{array} \right)$$

is given by the formula

$$\chi(\lambda) = \left| \begin{array}{ccc} \lambda - a & -b & -c \\ -d & \lambda - e & -f \\ -g & -h & \lambda - i \end{array} \right| .$$

This passage is produced by the following input:

```
The {\em characteristic polynomial} $\chi(\lambda)$ of the
$3 \times 3$~matrix
\[ \left( \begin{array}{ccc}
a & b & c \\
d & e & f \\
g & h & i \end{array} \right) \]
is given by the formula
\[ \chi(\lambda) = \left| \begin{array}{ccc}
\lambda - a & -b & -c \\
-d & \lambda - e & -f \\
-g & -h & \lambda - i \end{array} \right|. \]
```

First of all, note the use of `\left` and `\right` to produce the large delimiters around the arrays. As we have already seen, if we use

`\left(` ... `\right)`

then the size of the parentheses is chosen to match the subformula that they enclose. Next note the use of the alignment tab character `&` to separate the entries of the matrix and the use of `\\` to separate the rows of the matrix, exactly as in the construction of multiline formulae described above. We begin the array with `\begin{array}` and end it with `\end{array}`. The only thing left to explain, therefore, is the mysterious `{ccc}` which occurs immediately after `\begin{array}`. Now each of the `c`'s in `{ccc}` represents a column of the matrix and indicates that the entries of the column should be *centred*. If the `c` were replaced by `l` then the corresponding column would be typeset with all the entries flush *left*, and `r` would produce a column with all entries flush *right*. Thus

```
\[ \begin{array}{lcr}
\mbox{First number} & x & 8 \\
\mbox{Second number} & y & 15 \\
\mbox{Sum} & x + y & 23 \\
\mbox{Difference} & x - y & -7 \\
\mbox{Product} & xy & 120 \end{array} \]
```

produces

$$\begin{array}{lcr}
\mbox{First number} & x & 8 \\
\mbox{Second number} & y & 15 \\
\mbox{Sum} & x + y & 23 \\
\mbox{Difference} & x - y & -7 \\
\mbox{Product} & xy & 120 \end{array}$$

We can use the array environment to produce formulae such as

$$|x| = \left\{ \begin{array}{ll} x & \mbox{if $x \geq 0$};\\ -x & \mbox{if $x < 0$}.\end{array} \right.$$

Note that both columns of this array are set flush left. Thus we use `{ll}` immediately after `\begin{array}`. The large curly bracket is produced using `\left\{`. However this requires a corresponding `\right` delimiter to match it. We therefore use the *null delimiter* `\right.` discussed earlier. This delimiter is invisible. We can therefore obtain the above formula by typing

```
\[ |x| = \left\{ \begin{array}{ll}
         x & \mbox{if $x \geq 0$};\\
        -x & \mbox{if $x < 0$}.\end{array} \right.  \]
```

3.13 Derivatives, Limits, Sums and Integrals

The expressions

$$\frac{du}{dt} \mbox{ and } \frac{d^2 u}{dx^2}$$

are obtained in LaTeX by typing `\frac{du}{dt}` and `\frac{d^2 u dx^2}` respectively. The mathematical symbol ∂ is produced using `\partial`. Thus partial derivatives such as $\frac{\partial u}{\partial t}$ and $\frac{\partial^2 u}{\partial x^2}$ are obtained in LaTeX by typing

`\frac{\partial u}{\partial t}` and `\frac{\partial^2 u}{\partial x^2}`.

To obtain mathematical expressions such as

$$\lim_{x \to +\infty}, \inf_{x > s} \text{ and } \sup_K$$

in displayed equations we type `\lim_{x \to +\infty}`, `\inf_{x > s}` and `\sup_K` respectively. Thus to obtain

$$\lim_{x \to 0} \frac{3x^2 + 7}{x^2 + 1} = 3.$$

(in LaTeX) we type

```
\[ \lim_{x \to 0} \frac{3x^2 +7x^3}{x^2 +5x^4} = 3. \]
```

To obtain a summation sign such as

$$\sum_{i=1}^{2n}$$

we type `\sum_{i=1}^{2n}`. Thus

$$\sum_{k=1}^n k^2 = \frac{1}{2} n(n+1).$$

is obtained by typing

```
\[ \sum_{k=1}^n k^2 = \frac{1}{2} n (n+1). \]
```

We now discuss how to obtain *integrals* in mathematical documents. A typical integral is the following:

$$\int_a^b f(x)\,dx.$$

This is typeset using

```
\[ \int_a^b f(x)\,dx. \]
```

The integral sign $\int$ is typeset using the control sequence `\int`, and the *limits of integration* (in this case a and b) are treated as a subscript and a superscript on the integral sign. It remains to describe the purpose of the `\,` occurring immediately before the `dx`. This is the means of telling TEX to put extra space before the d. This is necessary to produce the correct appearance.

Most integrals occurring in mathematical documents begin with an integral sign and contain one or more instances of `d` followed by another (Latin or Greek) letter, as in dx, dt, and $d\theta$. To obtain the correct appearance one should put extra space before the d, using `\,`. Thus

$$\int_0^{+\infty} x^n e^{-x} \,dx = n!.$$

$$\int \cos \theta \,d\theta = \sin \theta.$$

$$\int_{x^2 + y^2 \leq R^2} f(x,y)\,dx\,dy = \int_{\theta=0}^{2\pi} \int_{r=0}^R f(r\cos\theta,r\sin\theta) r\,dr\,d\theta.$$

and

$$\int_0^R \frac{2x\,dx}{1+x^2} = \log(1+R^2).$$

are obtained by typing

```
\[ \int_0^{+\infty} x^n e^{-x} \,dx = n!. \]

\[ \int \cos \theta \,d\theta = \sin \theta. \]

\[ \int_{x^2 + y^2 \leq R^2} f(x,y)\,dx\,dy
   = \int_{\theta=0}^{2\pi} \int_{r=0}^R
      f(r\cos\theta,r\sin\theta) r\,dr\,d\theta. \]
```

and

```
\[ \int_0^R \frac{2x\,dx}{1+x^2} = \log(1+R^2). \]
```

respectively.

In some multiple integrals (i.e., integrals containing more than one integral sign) one finds that TEX puts too much space between the integral signs. The way to improve the appearance of of the integral is to use the control sequence \! to remove a thin strip of unwanted space. Thus, for example, the multiple integral

$$\int_0^1 \! \int_0^1 x^2 y^2\,dx\,dy.$$

is obtained by typing

```
\[ \int_0^1 \! \int_0^1 x^2 y^2\,dx\,dy. \]
```

Had we typed

```
\[ \int_0^1 \int_0^1 x^2 y^2\,dx\,dy. \]
```

we would have obtained

$$\int_0^1 \int_0^1 x^2 y^2\,dx\,dy.$$

A particularly noteworthy example comes when we are typesetting a multiple integral such as

$$\int \!\!\! \int_D f(x,y)\,dx\,dy.$$

Here we use \! three times to obtain suitable spacing between the integral signs. We typeset this integral using

```
\[ \int \!\!\! \int_D f(x,y)\,dx\,dy. \]
```

Had we typed

```
\[ \int \int_D f(x,y)\,dx\,dy. \]
```

we would have obtained

$$\int \int_D f(x,y)\,dx\,dy.$$

The following (reasonably complicated) passage exhibits a number of the features which we have been discussing:

> In non-relativistic wave mechanics, the wave function $\psi(\mathbf{r}, t)$ of a particle satisfies the *Schrödinger Wave Equation*
>
> $$i\hbar\frac{\partial\psi}{\partial t} = \frac{-\hbar^2}{2m} \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \right) \psi + V\psi.$$

It is customary to normalize the wave equation by demanding that

$$\int\!\!\!\int\!\!\!\int_{\mathbf{R}^3} \left| \psi(\mathbf{r},0) \right|^2\,dx\,dy\,dz = 1.$$

A simple calculation using the Schrödinger wave equation shows that

$$\frac{d}{dt} \int\!\!\!\int\!\!\!\int_{\mathbf{R}^3} \left| \psi(\mathbf{r},t) \right|^2\,dx\,dy\,dz = 0,$$

and hence

$$\int\!\!\!\int\!\!\!\int_{\mathbf{R}^3} \left| \psi(\mathbf{r},t) \right|^2\,dx\,dy\,dz = 1$$

for all times t. If we normalize the wave function in this way then, for any (measurable) subset V of $\mathbf{R}^3$ and time t,

$$\int\!\!\!\int\!\!\!\int_{V} \left| \psi(\mathbf{r},t) \right|^2\,dx\,dy\,dz$$

represents the probability that the particle is to be found within the region V at time t.

One would typeset this in LaTeX by typing

```
   In non-relativistic wave mechanics, the wave function
$\psi({\bf r},t)$ of a particle satisfies the
{\it Schr\"{o}dinger Wave Equation}
\[ i\hbar\frac{\partial \psi}{\partial t}
  = \frac{-\hbar^2}{2m} \left(
    \frac{\partial^2}{\partial x^2}
    + \frac{\partial^2}{\partial y^2}
    + \frac{\partial^2}{\partial z^2}
  \right) \psi + V \psi. \]
It is customary to normalize the wave equation by
demanding that
\[ \int \!\!\! \int \!\!\! \int_{{\bf R}^3}
      \left| \psi({\bf r},0) \right|^2\,dx\,dy\,dz = 1. \]
A simple calculation using the Schr\"{o}dinger wave
equation shows that
\[ \frac{d}{dt} \int \!\!\! \int \!\!\! \int_{{\bf R}^3}
      \left| \psi({\bf r},t) \right|^2\,dx\,dy\,dz = 0, \]
```

```
and hence
\[ \int \!\!\! \int \!\!\! \int_{{\bf R}^3}
      \left| \psi({\bf r},t) \right|^2\,dx\,dy\,dz = 1 \]
for all times~$t$. If we normalize the wave function in this
way then, for any (measurable) subset~$V$ of ${\bf R}^3$ and
time~$t$,
\[ \int \!\!\! \int \!\!\! \int_V
      \left| \psi({\bf r},t) \right|^2\,dx\,dy\,dz \]
represents the probability that the particle is to be found
within the region~$V$ at time~$t$.
```

4 Further Features of LaTeX

4.1 Producing Blank Space in LaTeX

To produce (horizontal) blank space within a paragraph, use `\hspace` and `\hspace*`, followed by the length of the blank space enclosed within curly brackets. The length of the skip should be expressed in a unit recognized by TeX. These recognized units are given in the following table:

`pt`	point	(1 in = 72.27 pt)
`pc`	pica	(1 pc = 12 pt)
`in`	inch	(1 in = 25.4 mm)
`bp`	big point	(1 in = 72 bp)
`cm`	centimetre	(1 cm = 10 mm)
`mm`	millimetre	
`dd`	didot point	(1157 dd = 1238 pt)
`cc`	cicero	(1 cc = 12 dd)
`sp`	scaled point	(65536 sp = 1 pt)

Thus to produce a horizontal blank space of 20 mm in the middle of a paragraph one would type `\hspace{20 mm}` (or `\hspace*{20 mm}`.

The difference between `\hspace` and `\hspace*` is that if TeX decides to break between lines at the point where an `\hspace` is specified, then the `\hspace` is ignored. Using `\hspace*` forces TeX to produce a horizontal space, whether of not TeX breaks between lines.

To produce (vertical) blank space between paragraphs, use `\vspace` and `\vspace*`, followed by the length of the blank space enclosed within curly brackets. A \vspace will be ignored if it comes at a break between pages, whereas blank space will always be produced by \vspace*, whether or not there is a page break. Thus to obtain

> This is the first paragraph of some text. It is separated from the second paragraph by a vertical skip of 10 millimetres.
>
> This is the second paragraph.

one should type

```
This is the first paragraph of some text.  It is
separated from the second paragraph by a vertical skip of
10 millimetres.

\vspace{10 mm}

This is the second paragraph.
```

4.2 Blank Spaces: Fine Tuning

We describe certain features of TeX relating to blank spaces and paragraph indentation which will improve the appearance of the final document. Experienced users of TeX will improve the appearance of their documents if they bear these remarks in mind.

First note that, as a general rule, you should never put a blank space after a left parenthesis or before a right parenthesis. If you were to put a blank space in these places, then you run the risk that TeX might start a new line immediately after the left parenthesis or before the right parenthesis, leaving the parenthesis marooned at the beginning or end of a line.

TeX has its own rules for deciding the lengths of blank spaces. For instance, TeX will put an extra amount of space after a full stop if it considers that the full stop marks the end of a sentence.

> The rule adopted by TeX is to regard a period (full stop) as the end of a sentence if it is preceded by a lowercase letter. If the period is preceded by an uppercase letter then TeX assumes that it is not a full stop but follows the initials of somebody's name.

This works very well in most cases. However TeX occasionally gets things wrong. This happens with a number of common abbreviations (as in 'Mr. Smith' or in 'etc.'), and, in particular, in the names of journals given in abbreviated form (e.g., 'Proc. Amer. Math. Soc.'). The way to overcome this problem is to put a backslash before the blank space in question. Thus we should type

```
Mr.\ Smith
etc.\ and
Proc.\ Amer.\ Math.\ Soc.
```

TeX determines itself how to break up a paragraph into lines, and will occasionally hyphenate long words where this is desirable. However it is sometimes necessary to tell TeX not to break at a particular blank space. The special character used for this purpose is `~`. It represents a blank space at which TeX is not allowed to break between lines. It is often desirable to use `~` in names where the forenames are represented by initials. Thus to obtain 'W. R. Hamilton' it is best to type `W.~R.~Hamilton`. It is also desirable in phrases like 'Example 7' and 'the length l of the rod', obtained by typing `Example~7` and `the length~$l$ of the rod`. This feature of TeX may be safely ignored by beginners, though more experienced TeXnical typists should gradually accustom themselves to using it occasionally where appropriate.

TeX will automatically indent paragraphs (with the exception of the first paragraph of a new section). One can prevent TeX from indenting a paragraph though by beginning the paragraph with the control sequence `\noindent`. Thus one obtains

> This is the beginning of a paragraph which is not indented in the usual way. This has been achieved by placing an appropriate control sequence at the beginning of the paragraph.

by typing

```
\noindent
This is the beginning of a paragraph which is not
indented in the usual way.  This has been achieved
by placing an appropriate control sequence at the
```

```
beginning of the paragraph.
```

Conversely, the control sequence `\indent` forces TEX to indent the paragraph.

4.3 The Preamble of the LATEX Input file

We describe the options available in LATEX for specifying the overall style of a document.

Every LATEX document should begin with a `\documentstyle` command followed by the `\begin{document}` command, and must end with the `\end{document}` command. A typical such document is the following:

```
\documentstyle[a4,12pt]{article}
\begin{document}

This is the first paragraph of a typical document. It is
produced in a `12~point' size. A {\it point} is a unit of
length used by printers. One point is approximately
$1/72$~inch. In a `12~point' font the height of the
parentheses is 12~points (i.e. about $1/6$~inch) and the
letter~`m' is about 12 points long.

This is the second paragraph of the document. There are
also `10 point' and `11 point' styles available in \LaTeX.
The required size is specified in the `documentstyle'
command. If no such size is specified then the 10~point
size is assumed.

\end{document}
```

The syntax of the `\documentstyle` command is as follows. The command begins with `\documentstyle` and ends with the names of one of the available styles, enclosed in curly brackets. The available styles are '`article`', '`report`', '`book`' and '`letter`'. Between the "`\documentstyle`" and the name of the document style, one may place a list of *options*. These options are separated by commas and the list of options is enclosed in square brackets (as in the above example). The options available (which are usually the names of certain 'style files') include the following:

11pt Specifies a size of type known as *eleven-point*, which is ten percent larger than the ten-point type normally used.

12pt Specifies a twelve-point type size, which is twenty percent larger than ten-point.

twocolumn Produces two-column output.

a4 A 'style file' which ensures that the page is appropriately positioned on A4 size paper. It is recommended that most LaTeX documents produced on the LaserWriter in the School of Mathematics use this option.

> Typing simply `\documentstyle{article}` will produce a document in ten-point type size. However the printed output will not be nicely positioned on A4 paper, since the default size is intended for a different (American) paper size.

Pages will be automatically numbered at the bottom of the page, unless you specify otherwise. This can be done using the `\pagestyle` command. This command should come after the `\documentstyle` command and before the `\begin{document}` command. This command has the syntax `\pagestyle{`*option*`}`, where the *option* is one of the following:

plain The page number is at the foot of the page. This is the default page style for the `article` and `report` document styles.

empty No page number is printed.

headings The page number (and any other information determined by the document style) is put at the top of the page.

myheadings Similar to the **headings** pagestyle, except that the material to go at the top of the page is determined by `\markboth` and `\markright` commands (see the LaTeX manual).

For example, the input file

```
\documentstyle[a4]{article}
\pagestyle{empty}
\begin{document}

The main body of the document is placed here.

\end{document}
```

produces a document without page numbers, using the standard ten-point type size.

4.4 Defining your own Control Sequences in LaTeX

Suppose that we are producing a paper that makes frequent use of some mathematical expression. For example, suppose that integrals like

$$\int_{-\infty}^{+\infty} f(x)\,dx.$$

occur frequently throughout the text. This formula is obtained by typing

```
\[ \int_{-\infty}^{+\infty} f(x)\,dx. \]
```

It would be nice if we could type `\inftyint` (say) to obtain the integral sign at the beginning. This can be done using `\newcommand`. What we do is to place a line with the command

```
\newcommand{\inftyint}{\int_{-\infty}^{+\infty}}
```

near the beginning of the input file (the best place being after the `\documentstyle` command but before the `\begin{document}` command). Then we only have to type

```
\[ \inftyint f(x)\,dx. \]
```

to obtain the above formula.

We can modify this procedure slightly. Suppose that we we defined a new control sequence `\intwrtx` by putting the line

```
\newcommand{\intwrtx}[1]{\int_{-\infty}^{+\infty} #1 \,dx}
```

at the beginning of the input file. If we then type the line

```
\[ \intwrtx{f(x)}. \]
```

then we obtain

$$\int_{-\infty}^{+\infty} f(x)\,dx.$$

What has happened is that the expression in curly brackets after `\intwrtx` has been substituted in the expression defining `\intwrtx`, replacing the `#1` in that expression.

The number 1 inside square brackets in the `\newcommand` line defining `\intwrtx` indicates to LaTeX that it is to expect one expression (in curly brackets) after `\intwrtx` to substitute for `#1` in the definition of `\intwrtx`. If we defined a control sequence `\intwrt` by

```
\newcommand{\intwrt}[2]{\int_{-\infty}^{+\infty} #2 \,d #1}
```

then it would expect two expressions to substitute in for `#1` and `#2` in the definition of `\intwrt`. Thus if we then type

```
\[ \intwrt{y}{f(y)}. \]
```

we obtain

$$\int_{-\infty}^{+\infty} f(y)\,dy.$$

A Control Sequences used in Text (LaTeX)

Control Sequences for Changing Fonts in Text

`\rm` changes to the normal "roman" font:	Roman
`\sl` changes to a slanted roman font:	*Slanted*
`\it` changes to an italic font:	*Italic*
`\tt` changes to an "typewriter" font:	`Typewriter`
`\bf` changes to a boldface font:	**Boldface**

In LaTeX the control sequence `\em` *emphasizes* text, switching from non-italic to *italic* and *from italic to* roman.

Control Sequences for obtaining Accents in Text

`\'{e}`	é	e.g., `math\'{e}matique` yields 'mathématique'
`` \`{e} ``	è	e.g., `` alg\`{e}bre `` yields 'algèbre'
`\^{e}`	ê	e.g., `h\^{o}te` yields 'hôte'
`\"{o}`	ö	e.g., `H\"{o}lder` yields 'Hölder'
`\~{n}`	ñ	e.g., `ma\~{n}ana` yields 'mañana'
`\={o}`	ō	
`\.{o}`	ȯ	
`\u{o}`	ŏ	
`\v{c}`	č	e.g., `\v{C}ech` yields 'Čech'
`\H{o}`	ő	
`\t{oo}`	o͡o	
`\c{c}`	ç	e.g., `gar\c{c}on` yields 'garçon'
`\d{o}`	ọ	
`\b{o}`	o̲	

These accents are for use in ordinary text. They cannot be used within mathematical formulae, since different control sequences are used to produce accents within mathematics.

Special Symbols used in Text

`\oe, \OE`	œ, Œ
`\ae, \AE`	æ, Æ
`\aa, \AA`	å, Å
`\o, \O`	ø, Ø
`\l, \L`	ł, Ł
`\ss`	ß
`` ?` ``	¿
`` !` ``	¡
`\dag`	†
`\ddag`	‡
`\S`	§
`\P`	¶
`\copyright`	©
`\pounds`	£
`\i`	ı
`\j`	ȷ

B Control Sequences used in Mathematics (LaTeX)

B.1 Font Changes, Accents and Standard Functions

Changing Fonts in Mathematical Expressions

Fonts are changed using suitable control sequences.

`\mit` changes to the 'math italic' font:	$MathItalic$
`\rm` changes to the roman font:	Roman
`\sl` changes to a slanted roman font:	*Slanted*
`\it` changes to an italic font:	*Italic*
`\tt` changes to an "typewriter" font:	`Typewriter`
`\bf` changes to a boldface font:	**Boldface**
`\cal` changes to a calligraphic font:	$\mathcal{CALLIGRAPHIC}$

The default font for mathematics is $MathItalic$. The $\mathcal{CALLIGRAPHIC}$ font is only available for uppercase letters. Any change of font made within a group enclosed within curly brackets `{` and `}` will only apply to text within that group. On leaving the group, the current font is restored to what it was before entering the group.

Accents in Mathematics Mode

Accents in mathematics mode are produced using appropriate control sequences. The effect of these on the letter a is exhibited in the following table.

`$\underline{a}$`	$\underline{a}$
`$\overline{a}$`	$\overline{a}$
`$\hat{a}$`	$\hat{a}$
`$\check{a}$`	$\check{a}$
`$\tilde{a}$`	$\tilde{a}$
`$\acute{a}$`	$\acute{a}$
`$\grave{a}$`	$\grave{a}$
`$\dot{a}$`	$\dot{a}$
`$\ddot{a}$`	$\ddot{a}$
`$\breve{a}$`	$\breve{a}$
`$\bar{a}$`	$\bar{a}$
`$\vec{a}$`	$\vec{a}$

These control sequences should only be used for mathematics, not for ordinary text.

You should bear in mind that when a character is underlined in a mathematical manuscript then it is normally typeset in bold face without any underlining. Underlining is used very rarely in print.

Standard Functions

The names of certain standard functions and abbreviations are obtained by typing a backlash \ before the name. The complete list in TeX is as follows:-

`\arccos`	`\cos`	`\csc`	`\exp`	`\ker`	`\limsup`	`\min`	`\sinh`
`\arcsin`	`\cosh`	`\deg`	`\gcd`	`\lg`	`\ln`	`\Pr`	`\sup`
`\arctan`	`\cot`	`\det`	`\hom`	`\lim`	`\log`	`\sec`	`\tan`
`\arg`	`\coth`	`\dim`	`\inf`	`\liminf`	`\max`	`\sin`	`\tanh`

B.2 Control Sequences for Mathematical Symbols

Lowercase Greek Letters

α	\alpha	ι	\iota	ϱ	\varrho
β	\beta	κ	\kappa	σ	\sigma
γ	\gamma	λ	\lambda	ς	\varsigma
δ	\delta	μ	\mu	τ	\tau
ϵ	\epsilon	ν	\nu	υ	\upsilon
ε	\varepsilon	ξ	\xi	ϕ	\phi
ζ	\zeta	o	o	φ	\varphi
η	\eta	π	\pi	χ	\chi
θ	\theta	ϖ	\varpi	ψ	\psi
ϑ	\vartheta	ρ	\rho	ω	\omega

Uppercase Greek Letters

Γ	\Gamma	Ξ	\Xi	Φ	\Phi
Δ	\Delta	Π	\Pi	Ψ	\Psi
Θ	\Theta	Σ	\Sigma	Ω	\Omega
Λ	\Lambda	Υ	\Upsilon		

Miscellaneous Symbols

$\aleph$	\aleph	$\prime$	\prime	$\forall$	\forall
$\hbar$	\hbar	$\emptyset$	\emptyset	$\exists$	\exists
$\imath$	\imath	∇	\nabla	$\neg$	\neg
$\jmath$	\jmath	$\surd$	\surd	$\flat$	\flat
ℓ	\ell	$\top$	\top	$\natural$	\natural
$\wp$	\wp	$\bot$	\bot	$\sharp$	\sharp
$\Re$	\Re	$\Vert$	\\|	$\clubsuit$	\clubsuit
$\Im$	\Im	$\angle$	\angle	$\diamondsuit$	\diamondsuit
∂	\partial	$\triangle$	\triangle	$\heartsuit$	\heartsuit
∞	\infty	$\backslash$	\backslash	$\spadesuit$	\spadesuit

"Large" Operators

$\sum$	\sum	$\bigcap$	\bigcap	$\bigodot$	\bigodot
$\prod$	\prod	$\bigcup$	\bigcup	$\bigotimes$	\bigotimes
$\coprod$	\coprod	$\bigsqcup$	\bigsqcup	$\bigoplus$	\bigoplus
$\int$	\int	$\bigvee$	\bigvee	$\biguplus$	\biguplus
$\oint$	\oint	$\bigwedge$	\bigwedge		

Binary Operations

$\pm$	\pm	$\cap$	\cap	$\vee$	\vee
$\mp$	\mp	$\cup$	\cup	$\wedge$	\wedge
$\setminus$	\setminus	$\uplus$	\uplus	$\oplus$	\oplus
$\cdot$	\cdot	$\sqcap$	\sqcap	$\ominus$	\ominus
$\times$	\times	$\sqcup$	\sqcup	$\otimes$	\otimes
$\ast$	\ast	$\triangleleft$	\triangleleft	$\oslash$	\oslash
$\star$	\star	$\triangleright$	\triangleright	$\odot$	\odot
$\diamond$	\diamond	$\wr$	\wr	$\dagger$	\dagger
$\circ$	\circ	$\bigcirc$	\bigcirc	$\ddagger$	\ddagger
$\bullet$	\bullet	$\bigtriangleup$	\bigtriangleup	$\amalg$	\amalg
$\div$	\div	$\bigtriangledown$	\bigtriangledown		

Relations

$\leq$	\leq	$\geq$	\geq	$\equiv$	\equiv
$\prec$	\prec	$\succ$	\succ	$\sim$	\sim
$\preceq$	\preceq	$\succeq$	\succeq	$\simeq$	\simeq
$\ll$	\ll	$\gg$	\gg	$\asymp$	\asymp
$\subset$	\subset	$\supset$	\supset	$\approx$	\approx
$\subseteq$	\subseteq	$\supseteq$	\supseteq	$\cong$	\cong
$\sqsubseteq$	\sqsubseteq	$\sqsupseteq$	\sqsupseteq	$\bowtie$	\bowtie
$\in$	\in	$\ni$	\ni	$\propto$	\propto
$\vdash$	\vdash	$\dashv$	\dashv	$\models$	\models
$\smile$	\smile	$\mid$	\mid	$\doteq$	\doteq
$\frown$	\frown	$\parallel$	\parallel	$\perp$	\perp

Negated Relations

$\not<$	`\not<`	$\not>$	`\not>`	$\not=$	`\not=`
$\not\leq$	`\not\leq`	$\not\geq$	`\not\geq`	$\not\equiv$	`\not\equiv`
$\not\prec$	`\not\prec`	$\not\succ$	`\not\succ`	$\not\sim$	`\not\sim`
$\not\preceq$	`\not\preceq`	$\not\succeq$	`\not\succeq`	$\not\simeq$	`\not\simeq`
$\not\subset$	`\not\subset`	$\not\supset$	`\not\supset`	$\not\approx$	`\not\approx`
$\not\subseteq$	`\not\subseteq`	$\not\supseteq$	`\not\supseteq`	$\not\cong$	`\not\cong`
$\not\sqsubseteq$	`\not\sqsubseteq`	$\not\sqsupseteq$	`\not\sqsupseteq`	$\not\asymp$	`\not\asymp`

Arrows

$\leftarrow$	`\leftarrow`	$\longleftarrow$	`\longleftarrow`	$\uparrow$	`\uparrow`
$\Leftarrow$	`\Leftarrow`	$\Longleftarrow$	`\Longleftarrow`	$\Uparrow$	`\Uparrow`
$\rightarrow$	`\rightarrow`	$\longrightarrow$	`\longrightarrow`	$\downarrow$	`\downarrow`
$\Rightarrow$	`\Rightarrow`	$\Longrightarrow$	`\Longrightarrow`	$\Downarrow$	`\Downarrow`
$\leftrightarrow$	`\leftrightarrow`	$\longleftrightarrow$	`\longleftrightarrow`	$\updownarrow$	`\updownarrow`
$\Leftrightarrow$	`\Leftrightarrow`	$\Longleftrightarrow$	`\Longleftrightarrow`	$\Updownarrow$	`\Updownarrow`
$\mapsto$	`\mapsto`	$\longmapsto$	`\longmapsto`	$\nearrow$	`\nearrow`
$\hookleftarrow$	`\hookleftarrow`	$\hookrightarrow$	`\hookrightarrow`	$\searrow$	`\searrow`
$\leftharpoonup$	`\leftharpoonup`	$\rightharpoonup$	`\rightharpoonup`	$\swarrow$	`\swarrow`
$\leftharpoondown$	`\leftharpoondown`	$\rightharpoondown$	`\rightharpoondown`	$\nwarrow$	`\nwarrow`
$\rightleftharpoons$	`\rightleftharpoons`				

Openings

$[$	`\lbrack`	$\lfloor$	`\lfloor`		`\lciel`
$\{$	`\lbrace`	$\langle$	`\langle`		

Closings

$]$	`\rbrack`	$\rfloor$	`\rfloor`		`\rciel`
$\}$	`\rbrace`	$\rangle$	`\rangle`		

Alternative Names

$\neq$	`\ne` or `\neq`	(same as `\not=`)
$\leq$	`\le`	(same as `\leq`)
$\geq$	`\ge`	(same as `\geq`)
$\{$	`\{`	(same as `\lbrace`)
$\}$	`\}`	(same as `\lbrace`)
$\to$	`\to`	(same as `\rightarrow`)
$\gets$	`\gets`	(same as `\leftarrow`)
$\ni$	`\owns`	(same as `\ni`)
$\land$	`\land`	(same as `\wedge`)
$\lor$	`\lor`	(same as `\vee`)
$\lnot$	`\lnot`	(same as `\neg`)
$\vert$	`\vert`	(same as `|`)
$\Vert$	`\Vert`	(same as `\\|`)
$\iff$	`\iff`	(same as `\Longleftrightarrow`, but with extra space at each end)
$\colon$	`\colon`	(same as :, but with less space around it and less likelihood of a line break after it)

B.3 Some frequently used Control Sequences of LaTeX

Control Sequences

We list some of the control sequences and environments of LaTeX that are frequently used when typesetting mathematical formulae. The list is by no means exhaustive. For information on how to apply these control sequences, consult the appropriate manual (e.g. 'LaTeX—User's Guide and Reference Manual') Here is the list of control sequences.

`\frac`	produces fractions
`\sqrt`	produces square roots and nth roots
`\left`	produces left delimiter of required size
`\right`	produces right delimiter of required size
`\,`	produces a thin space
`\!`	removes a thin space
`\mbox`	creates a box of text within mathematics

Environments

The following environments are often used in typesetting mathematics.

```
\begin{equation}  ...  \end{equation}
```

(produces a numbered displayed formula)

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

(produces a numbered displayed multiline formula)

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

(produces an unnumbered displayed multiline formula)

```
\begin{array}  ...  \end{array}
```

(produces an array or matrix)

Contents