12 Mathematics

LaTeX is based on TeX, a typesetting package that was originally designed for use with mathematics. LaTeX inherits from TeX some very powerful mathematical typesetting facilities. The spacing rules in mathematics are entirely different from those in the text we have seen so far.

Displayed maths comes in two principal varieties, "in-line" and "displayed". In-line maths leads to expressions of the form:

> Einstein derived $\Delta E = \Delta m \, c^2$ from the more important expression $|p|^2 = m_0^2 \, c^2$.

whereas displayed maths appears as:

> Einstein derived
> $$\Delta E = \Delta m \, c^2$$
> from the more important expression
> $$|p|^2 = m_0^2 \, c^2.$$

There are three distinct ways to generate in-line maths:

- `\(\Delta E = \Delta m \, c^2\)`
- `$\Delta E = \Delta m \, c^2$`
- `\begin{math}`

  `\Delta E = \Delta m \, c^2`

  `\end{math}`

There are two ways to generate displayed maths:

- `\[\Delta E = \Delta m \, c^2\]`
- `\begin{displaymath}`

  `\Delta E = \Delta m \, c^2`

```
\end{displaymath}
```

The author's preferences are for the bracket forms, `\(...\)` and `\[...\]`.

If the environment `equation` is used in place of `displaymath` then, in addition to being displayed, the equation is numbered. Use of `\label` in this environment will set up references to the equation number. The evaluation of the equation numbers is done automatically, and the nature of the numbers assigned is determined by the style.

By default, displayed mathematical expressions are centred horizontally and equation numbers are placed at the right hand side of the page. There are two options that can be passed to the standard documentclasses, `leqno` and `fleqn`. `leqno` causes equation numbers to be on the left hand side of the page and `fleqn` causes displayed maths to be aligned so that it starts a fixed distance from the left margin.

12.1 Mathematical symbols

First we must note that the spacing around a symbol in maths mode depends on the character's mathematical status.

Simple variables (such as x) get minimal space around them,

Binary operations (such as the addition operator, $+$) get slightly more,

Relations (such as equality, $=$) get most.

It is sometimes necessary to help out LaTeX with the spacing and this will be covered later in section 12.5.

Table 9 gives the commands for generating the Greek alphabet. These are all simple variables. Note that some characters have multiple versions. For example `\epsilon` generates ϵ and `\varepsilon` generates ε. Also note that where a letter of the Greek alphabet is identical to a Latin letter, the Latin letter should be used. So lowercase omicron is given by `o` and uppercase beta by `B`.

Some characters are not available by default and require the `latexsym` package, loaded in by the `\usepackage{latexsym}` command in the preamble.

It isn't worth remembering which characters are in the default LaTeX and which are in the extra package, so the author recommends *always* loading the `latexsym` package in a mathematical document, even if it doesn't end up being used. For even more bizarre characters, the `amssymb` package may be available. The "LaTeX Companion" has details in chapter 8.

A limited set of accents are available in maths mode. These are given in table 10. Unlike text accents, these can be stacked fairly freely.

The binary operations are listed in table 11. It is impossible to create a complete set of mathematical binary operations; mathematicians take a perverse delight in coming up with new notations to annoy the most conscienscious of typesetters.

Binary relations are given in table 12. Note that it is possible to negate any binary relation by preceding it with a `\not`. So `\equiv` gives $\equiv$ and `\not\equiv` gives $\not\equiv$.

LaTeX's set of arrows is given in table 13. These have similar spacing to relations. Finally, the miscellanea are given in table 14.

12.2 Super- and subscripts

In maths mode the circumflex, "`^`", superscripts the next character or group of characters, and the underscore, "`_`", subscripts it. A fairly exhaustive set of examples is given in table 15.

It is important to remember that if more that one character is to be scripted then they must be grouped in braces.

LaTeX has two levels of scripting, "script" and "script script". In an expression like A^{BC} (`\(A^{BC}\)`), the B and C are both superscripts of A. However, in an expression like A^{B^C} (`\(A^{B^C}\)`) the C is a superscript of B which is, in turn, a superscript of A; C is a supersuperscript. The expressions "`A^B^C`" and "`A_B_C`" are ambiguous and will cause an error. Also note the difference between A^B_C (`\(A^B_C\)` or `\(A_C^B\)`) and ${A^B}_C$ (`\({A^B}_C\)`).

\alpha	α	\beta	β	\gamma	γ	\delta	δ
\epsilon	ϵ	\varepsilon	ε	\zeta	ζ	\eta	η
\theta	θ	\iota	ι	\kappa	κ	\lambda	λ
\mu	μ	\nu	ν	\xi	ξ	o	o
\pi	π	\varpi	ϖ	\rho	ρ	\varrho	ϱ
\sigma	σ	\varsigma	ς	\tau	τ	\upsilon	υ
\phi	ϕ	\varphi	φ	\chi	χ	\psi	ψ
\omega	ω						
A	A	B	B	\Gamma	Γ	\Delta	Δ
E	E	Z	Z	\Theta	Θ	H	H
I	I	K	K	\Lambda	Λ	M	M
N	N	\Xi	Ξ	O	O	\Pi	Π
P	P	\Sigma	Σ	\Upsilon	Υ	\Phi	Φ
X	X	\Psi	Ψ	\Omega	Ω		

Table 9: Letters of the Greek alphabet

\hat{o}	$\hat{o}$	\tilde{o}	$\tilde{o}$	\bar{o}	$\bar{o}$
\dot{o}	$\dot{o}$	\ddot{o}	$\ddot{o}$	\vec{o}	$\vec{o}$
\acute{o}	$\acute{o}$	\grave{o}	$\grave{o}$		
\check{o}	$\check{o}$	\breve{o}	$\breve{o}$		

Table 10: The set of maths accents

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

Table 11: Binary operations

`>`	$>$	`\succ`	$\succ$	`\supset`	$\supset$	`\sqsupset`	$\sqsupset$
`\geq`	$\geq$	`\succeq`	$\succeq$	`\supseteq`	$\supseteq$	`\sqsupseteq`	$\sqsupseteq$
`\in`	$\in$	`\vdash`	$\vdash$	`\smile`	$\smile$		
`<`	$<$	`\prec`	$\prec$	`\subset`	$\subset$	`\sqsubset`	$\sqsubset$
`\leq`	$\leq$	`\preceq`	$\preceq$	`\subseteq`	$\subseteq$	`\sqsubseteq`	$\sqsubseteq$
`\ni`	$\ni$	`\dashv`	$\dashv$	`\frown`	$\frown$		
`\sim`	$\sim$	`\approx`	$\approx$	`\neq`	$\neq$	`\mid`	$\mid$
`\simeq`	$\simeq$	`\cong`	$\cong$	`\doteq`	$\doteq$	`\parallel`	$\parallel$
`\equiv`	$\equiv$	`\asymp`	$\asymp$	`\Join`	$\Join$	`\bowtie`	$\bowtie$
`\models`	$\models$	`\perp`	$\perp$	`\propto`	$\propto$	`=`	$=$

Table 12: Binary relations

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

Table 13: Arrows

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

Table 14: Mathematical miscellany

A^B	A^B	A_B	A_B
A^BC	A^BC	A_BC	A_BC
A^{BC}	A^{BC}	A_{BC}	A_{BC}
A^{B^C}	A^{B^C}	A_{B_C}	A_{B_C}
{A^B}^C	${A^B}^C$	{A_B}_C	${A_B}_C$
A^B^C	*error*	A_B_C	*error*
A^B_C	A^B_C	A_B^C	A_B^C
A^{B_C}	A^{B_C}	A_{B^C}	A_{B^C}
{A^B}_C	${A^B}_C$	{A_B}^C	${A_B}^C$

Table 15: Super- and subscripts

12.3 Other common constructions

Some other commonly used mathematical constructions are illustrated here by example:

12.3.1 Surds

```
\( 2 = \sqrt{4} = \sqrt[4]{16} \)
```

$2 = \sqrt{4} = \sqrt[4]{16}$

12.3.2 Fractions

```
\( \frac{a+bi}{c+di} \)
```

$\frac{a+bi}{c+di}$

NB: The numerator and denominator are cast into script size.

12.3.3 Ellipsis

```
\( a,b,\ldots,y,z,
   a+b+\cdots+y+z \)
```

$a, b, \ldots, y, z, a + b + \cdots + y + z$

Vertical and diagonal dots are available in addition to low and centred horizontal dots. `\ldots` gives low dots, `\cdots` gives centred dots, `\vdots` gives vertical dots and `\ddots` gives diagonal dots, with the diagonal running from top left to bottom right.

12.3.4 Arrays

```
\(
\begin{array}{cccc}
A_{11} & A_{12} & \cdots & A_{1n} \\
A_{21} & A_{22} & \cdots & A_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
A_{m1} & A_{m2} & \cdots & A_{mn}
\end{array}
\)
```

$\begin{array}{cccc}
A_{11} & A_{12} & \cdots & A_{1n} \\
A_{21} & A_{22} & \cdots & A_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
A_{m1} & A_{m2} & \cdots & A_{mn}
\end{array}$

Note that, as in `tabular` mode, the final line does not have a trailing `\\`.

12.4 Symbols that vary according to context

LaTeX maths mode has some facilities for varying the display of characters based on context.

Some symbols change their size *and behaviour* according to whether they are in displayed maths or in-line maths. Consider `\sum_{i=0}^\infty x_i^2` in these two contexts:

- In-line:
  - `\(\sum_{i=0}^\infty x_i^2\)`
  - $\sum_{i=0}^\infty x_i^2$
- Displayed:
  - `\[\sum_{i=0}^\infty x_i^2\]`
  - $$\sum_{i=0}^\infty x_i^2$$

- Different size of sigma
- Different location of limits
- Note that `\sum` is *not* the same as `\Sigma`!

A set of these resizing symbols is given in table 16.

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

Table 16: Symbols that resize according to in-line or display mode

The other set of symbols that need to change their sizes according to context are the delimiters. The most common of these symbols are the parenteses (round brackets). These need to be of a size suitable for whatever it is that they delimit. For example, simple parentheses around an array (perhaps in an attempt to create a matrix) give daft results:

- Source:

```
\[ (
\begin{array}{cc}
A & B \\
C & D
\end{array}
) \]
```

- Output:

$$(\begin{array}{cc} A & B \\ C & D \end{array})$$

- *Wrong!*

Clearly we need a mechanism for causing the delimiters to resize. This is done with the `\left` and `\right` commands:

- Source:

```
\[ \left(
\begin{array}{cc}
A & B \\
C & D
\end{array}
\right) \]
```

- Output:

$$\left(\begin{array}{cc} A & B \\ C & D \end{array}\right)$$

- *Correct!*

The set of delimiters is given in table 17. The delimiters are resized to fit around the maths between the `\left` and the `\right`. The two delimiters used need not match. The uses of `\left` and `\right` must balance. There is an extra "invisible delimiter" generated by "`\left.`" and "`\right.`". This generates no output but can be used to match a delimiter being resized with the other of the `\left`, `\right` pair. Note that it is perfectly acceptable to use `\left)` or `\right(`.

In the author's opinion, the `\left` and `\right` commands should be used with *every* pair of delimiters in a maths text.

`(`	$($	`)`	$)$
`[`	$[$	`]`	$]$
`\{`	$\{$	`\}`	$\}$
`|`	$\vert$	`\\|`	$\Vert$
`\lfloor`	$\lfloor$	`\rfloor`	$\rfloor$
`\lceil`	$\lceil$	`\rceil`	$\rceil$
`\langle`	$\langle$	`\rangle`	$\rangle$
`/`	$/$	`\backslash`	$\backslash$
`\uparrow`	$\uparrow$	`\downarrow`	$\downarrow$
`\Uparrow`	$\Uparrow$	`\Downarrow`	$\Downarrow$
`\updownarrow`	$\updownarrow$	`\Updownarrow`	$\Updownarrow$

Table 17: The set of resizable delimiters in LaTeX

12.5 Spacing in mathematical formulas

While text has relatively straightforward spacing rules, mathematical expressions are vastly more complicated to get right automatically. So LaTeX doesn't try too hard; it has some simple rules and allows the user to make modifications to taste. To this end, the spacing commands in table 18 are provided.

thin *negative* space	`\!`
thin space	`\,`
medium space	`\:`
thick space	`\;`
interword space	`\␣`

Table 18: Spacing commands for maths mode

An example of where thin space would be used is in the expression $\Delta E = \Delta m\, c^2$. LaTeX does not know that `\Delta` is supposed to bind closely to `E` and `m`. It cannot distinguish it from an equation involving the product of a variable called Δ and a variable called E (or m). So there are three principal ways to resolve this:

1. Ignore the problem:
   - `\(\Delta E = \Delta m c^2\)`
   - $\Delta E = \Delta m c^2$
   - This relies on context; a mathematician reading this will not make the mistake of treating Δ as a variable.
2. Pull in the Δ to its arguments:
   - `\(\Delta\! E = \Delta\! m c^2\)`
   - $\Delta\! E = \Delta\! m c^2$
   - In the author's *subjective* opinion this looks a bit cramped.
3. Push the Δm construct away from the c^2 construct:
   - `\(\Delta E = \Delta m\, c^2\)`
   - $\Delta E = \Delta m\, c^2$
   - In the author's *subjective* opinion this is the best of the three.

Note that we have been forced to make typesetting decisions. We are now pushing LaTeX hard enough that it is starting to fail in its design aim of removing the need for typesetting skills from the user.

Another example of the need to use explicit spacing in LaTeX is in expressions involving double integrals. Compare the following:

- `\int \int z dx dy`
- $\int \int z dx dy$
- `\int \!\! \int z \, dx \, dy`
- $\int \!\! \int z \, dx \, dy$

12.6 Fonts

Just as text mode has commands for changing font (section 10.2), maths mode has something similar. Table 19 illustrates the commands available. Note that only the Latin characters and the upper case Greek characters have been altered.

In addition to the standard fonts there is also a "caligraphic" font available for capital (Latin alphabet) letters only. The caligraphic alphabet looks like this: $\mathcal{ABCDEFGHIJKLMNOPQRSTUVWXYZ}$. Note that applying the corresponding font command, `\mathcal`, to anything other than these 26 characters will produce unpredictable results.

`\mathrm{\Psi=\psi+X^2+y^2}`	$\mathrm{\Psi=\psi+X^2+y^2}$
`\mathsf{\Psi=\psi+X^2+y^2}`	$\mathsf{\Psi=\psi+X^2+y^2}$
`\mathtt{\Psi=\psi+X^2+y^2}`	$\mathtt{\Psi=\psi+X^2+y^2}$
`\mathit{\Psi=\psi+X^2+y^2}`	$\mathit{\Psi=\psi+X^2+y^2}$
`\mathbf{\Psi=\psi+X^2+y^2}`	$\mathbf{\Psi=\psi+X^2+y^2}$
`\mathcal{\Psi=\psi+X^2+y^2}`	$\ominus = \psi + \mathcal{X}^{\in} + \dagger^{\in}$

Table 19: Maths mode font modifying commands

Once again, you should not use explicit font changes in the body of your text but should create new commands based on the reason for the change of font. If you want to write all your vectors in a bold face then you should create a new command:

```
\newcommand{\vect}[1]{\mathbf{#1}}
```

Then you can write

```
\( \vect{a} = \vect{b} \times \vect{c} \)
```

and not worry about how the final style is planning to typeset vectors. If the final editor wants to use the "$\vec{a}$" notation then simply changing the line in the preamble to "`\newcommand{\vect}[1]{\vec{#1}}`" will do.

12.7 Aligning equations

So far we have only considered the display of single equations. A common requirement is for multiple equations to be displayed, normally with their equals signs (or other relations) vertically aligned. LaTeX provides a pair of environments called `eqnarray` and `eqnarray*` for this. `eqnarray` will number each equation unless instructed not to for a specific equation. `eqnarray*` will not number its equations. Consider this example of the use of `eqnarray`:

```
\begin{eqnarray}
x & = & y+5-4 \\ \label{x-y}
  & = & y+1 \nonumber \\
  & > & y \label{x-y-inequality}
\end{eqnarray}
```

$$\begin{array}{rcl} x & = & y+5-4 \qquad (1) \\ & = & y+1 \\ & > & y \qquad (2) \end{array}$$

Note that the equations are treated as a three column array with `[rcl]` alignment and that, like other arrays the final line does not end with a `\\`.

12.8 Function names

When a function, such as "sin", is being processed, LaTeX cannot distinguish the sine function from the product of the variables s, i and n. To resolve this, special LaTeX commands are provided for representing the common function names in roman style, rather than the ambient maths italic. For example, the sine function has the LaTeX representation `\sin`.

- Wrong
  - `\( sin^2(\theta) + cos^2(\theta) \equiv 1 \)`
  - $sin^2(\theta) + cos^2(\theta) \equiv 1$
- Right
  - `\( \sin^2(\theta) + \cos^2(\theta) \equiv 1 \)`
  - $\sin^2(\theta) + \cos^2(\theta) \equiv 1$
  - (Actually, we should use `\left(` and `\right)`.)

The complete set of function names defined in LaTeX is given in table 20.

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

Table 20: The set of maths functions names in LaTeX

12.9 Theorems etc.

Mathematics has a huge number of names for statements which are presented in a fairly common manner: theorems, conjectures, postulates, definitions, hypotheses, propositions, lemmas, axioms and many, many more. Furthermore, people's numbering schemes vary widely. Some people have a single numbering scheme so that theorem 1 is followed by definition 2 which is followed by lemma 3 and so on. Other people might group all the statements that need proofs but have separate number tracks for axioms, definitions and hypotheses, so definition 1 would be followed by theorem 1, lemma 2, lemma 3, definition 2, axiom 1 and so on.

Because of this wide variety, the default LaTeX styles define *no* "theorem-style" environments at all, but instead provide a command, `\newtheorem` to create them at the user's behest.

Suppose I want to create an environment for laws (Newton's, Thermodynamics', Murphy's,...) with each law started with the label "Law". I run the command

```
\newtheorem{law}{Law}
```

The first argument to `\newtheorem` is the name of the environment to be created, and the second argument is the label to put at the start of the generated output. If I use the command

```
\begin{law}
The harder you kick it, the faster it accelerates.
\end{law}
```

I get the output:

> **Law 1** *The harder you kick it, the faster it accelerates.*

and if I follow it with another "law",

```
\begin{law}
You can't push a rope.
\end{law}
```

I get the output

> **Law 2** *You can't push a rope.*

so a counter is being incremented automatically. If I want to name my laws then I can do so with an optional argument to the `\begin{law}` command:

```
\begin{law}[Murphy]
If it can go wrong, it will go wrong.
\end{law}
```

Law 3 (Murphy) *If it can go wrong, it will go wrong.*

Now suppose I want to define another environment of this style, sharing the same numbering track as the laws are using. It is to be called "`para`" and label its text as "Paradox". To do this, I insert an optional argument in the `\newtheorem` command in the preamble:

```
\newtheorem{para}[law]{Paradox}
```

and then the command

```
\begin{para}
If Murphy's law can go wrong, it will go wrong.
\end{para}
```

Paradox 4 *If Murphy's law can go wrong, it will go wrong.*