

LBT by Example

Gavin Sinclair

Document under active development from May 2025

Current version: 19 July 2025

Contents

Preface	v
Introduction	vi
Part I Core templates	1
1 Core commands	2
1.1 Document divisions	2
1.2 Ordinary text and whitespace	2
1.3 Margins and justification	4
1.4 Other stuff...	5
1.5 Passing content through to Latex	6
2 Lists and tables	8
2.1 Itemized and enumerated lists	8
2.2 Description lists	11
2.3 Automatic multi-level lists	11
2.4 Tables	12
3 Mathematical text (lbt.Math) – various macros	16
3.1 The <code>simplemath</code> macro	16
3.2 The <code>integral</code> macro	18
3.3 The <code>vector</code> macro	20
3.4 List, sequence, summation and product macros	21
3.5 Miscellaneous macros	24
3.6 Code to set up all macros	24
4 Mathematical text (lbt.Math) – the <code>MATH</code> command	26
4.1 Opening remarks	26
4.2 Simple equations with <code>MATH (.o equation)</code>	29
4.3 Long equations with <code>MATH .o multiline</code>	29
4.4 <code>gather</code>	30
4.5 Several-part equations with <code>eqsplit</code>	32

4.6	<code>align</code> and <code>alignat</code> and <code>flalign</code>	33
4.7	Other environments	42
4.8	Additional environments	42
4.9	Combinations	45
4.10	Summary of the <code>MATH</code> command	45
 Part II Non-core built-in templates		46
5	Worksheet or exam questions with <code>lbt.Questions</code>	47
 Part III Creating a new template		49
 Part IV Extra features		50

List of Examples

1.1	Chapters, sections and plain text paragraphs	3
1.2	<code>TEXT</code> (single and multiple paragraphs) and <code>VSPACE</code>	3
1.3	Using <code>VSTRETCH</code> to spread out paragraphs on a page	4
1.4	Using <code>INDENT</code> to adjust left and (optionally) right margins . .	4
1.5	Left, center and right justification	5
1.6	Passing Latex code through with <code>LATEX</code>	7
2.1	An itemized list	9
2.2	An enumerated list	9
2.3	A compact enumerated list with custom label	10
2.4	A custom shopping list	10
2.5	A mutli-level list	11
2.6	Simple table example	13
2.7	Simple table example with more formatting	13
2.8	Table with spacing, color, math mode and lines	14
2.9	Table with data loaded from a file	15
3.1	<code>simplemath</code> usage, both inline and display form	17
3.2	A collection of <code>simplemath</code> examples	17
3.3	<code>simplemath</code> 's handling of parentheses, brackets and <code>text</code> . . .	18
3.4	The <code>integral</code> macro	18
3.5	A collection of <code>integral</code> examples	19
3.6	A collection of <code>vector</code> and <code>vectorijk</code> examples	20
3.7	Vectors in bold or tilde or arrow format, document-wide . . .	21
3.8	Vectors in bold or tilde or arrow format, one-off	21
3.9	List/sequence macros such as <code>mathsum</code> and <code>mathseqproduct</code> .	23
3.10	Miscellaneous macros such as <code>primefactorisation</code>	24
3.11	Preamble code to set up all <code>lbt.Math</code> macros	25
4.1	<code>MATH .o equation</code> to format a simple equation	29
4.2	A long equation with <code>multiline</code>	30
4.3	Displaying several equations with <code>gather</code>	30
4.4	<code>gather</code> with selective numbering	31
4.5	<code>eggathered</code> : a single equation number for a gathered block .	31
4.6	The left/right justified variants of <code>eggathered</code>	32
4.7	Using <code>MATH .o eqsplit</code> for a multi-step equation	33
4.8	A <code>MATH .o eqsplit</code> equation that is referenced	33

4.9	Aligning a group of equations with <code>align</code>	34
4.10	Alignment without numbering	34
4.11	Alignment with selective numbering	35
4.12	Alignment in multiple columns	35
4.13	Full-length in multiple columns	36
4.14	Manual control of inter-column spacing	36
4.15	Using <code>alignat</code> and <code>MoveEqLeft</code> for a long equation	37
4.16	Using <code>MATH .o align</code> for a multi-step equation with commentary	38
4.17	Using <code>MATH .o alignat</code> to improve the previous example . .	39
4.18	Using <code>MATH .o eqalignedat</code> to align a single logical equation	40
4.19	Including a line of text amidst an aligned block	41
4.20	<code>continuation</code> supporting neat display of a long equation . . .	43
4.21	<code>commentary</code> supporting aligned text to the right of the multi-line equation	45
5.1	Question parts and subparts	47
5.2	Arranging question parts horizontally	48
5.3	Multiple choice answers	48
5.4	Question source and notes	48

Preface

The Preface will give some introductory remarks about LBT.

Introduction

The Introduction will give a brief but somewhat comprehensive example.

Part I

Core templates

1 Core commands

The `lbt.Basic` template implements:

- document divisions (part, chapter, section, subsection, subsubsection, paragraph, subparagraph)
- low-level typesetting facilities (vertical space, arbitrary commands and environments, `flushleft`, `flushright`, `center`)
- things you expect in normal Latex editing, whether built in or using a common plugin (`columns`, `verbatim`, various math environments)
- things that are generally useful (include PDF pages, three levels of headings, place two items side-by-side)

It also implements lists and tables, which are demonstrated in chapter 2 on page 8.

1.1 Document divisions

All the Latex commands are present. We present `CHAPTER` and `SECTION` in [Example 1.1](#) on the next page, but don't show the typeset results as we don't want to affect the chapter of this book!

1.2 Ordinary text and whitespace

[Example 1.2](#) shows the `TEXT` command, which outputs one or more paragraphs. `TEXT*` suppresses the (final) `\par`. You can input vertical space with `VSPACE`. Further, all commands accept optional arguments `pre` and `post` for including some surrounding whitespace.

Other low-level formatting commands include `CLEARPAGE` and `VFILL`, both of which are hard to demonstrate, but predictable. Finally, there is `VSTRETCH`, which helps spread items out vertically. For example, [Example 1.3](#) on page 4 spreads out three (very short) paragraphs to fill a page, and allocates more whitespace in the middle than the other two.

The `CLEARPAGE` is necessary to ensure that all vertical space on the page is used.

```
CHAPTER (label) ch:canopy :: Beneath the canopy

TEXT When we first visited the rainforest, \dots

SECTION (label) sec:life-leaves :: Life through the leaves

TEXT From the forest floor, you can't see any direct light at all \dots
```

```
\chapter{Beneath the canopy} \label{ch:canopy}
When we first visited the rainforest, \dots
\par
\section{Life through the leaves} \label{sec:life-leaves}
From the forest floor, you can't see any direct light at all \dots
\par
```

Example 1.1 Chapters, sections and plain text paragraphs

```
TEXT Can you guess which author wrote these sentences about chess?
VSPACE 1em
TEXT
:: The chessboard is the world; the pieces are the phenomena of the universe; the
↪ rules of the game are what we call the laws of Nature.
:: The beauty of a move lies not in its appearance but in the thought behind it.
:: In chess, as in life, a man is his own most dangerous opponent.
TEXT* .o pre = 2em :: It was\dots
TEXT an early world champion.
```

Can you guess which author wrote these sentences about chess?

The chessboard is the world; the pieces are the phenomena of the universe; the rules of the game are what we call the laws of Nature.

The beauty of a move lies not in its appearance but in the thought behind it.

In chess, as in life, a man is his own most dangerous opponent.

It was...an early world champion.

Example 1.2 TEXT (single and multiple paragraphs) and VSPACE

```

TEXT Top paragraph.
VSTRETCH 1
TEXT Middle paragraph.
VSTRETCH 1.5
TEXT Bottom paragraph.
VSTRETCH 1
CLEARPAGE

```

Example 1.3 Using VSTRETCH to spread out paragraphs on a page

```

INDENT 4cm, 2cm :: It is a truth universally acknowledged, that a single man in
→ possession of a good fortune, must be in want of a wife.

```

```

INDENT 6cm :: However little known the feelings or views of such a man may be on his
→ first entering a neighbourhood, this truth is so well fixed in the minds of the
→ surrounding families, that he is considered as the rightful property of some one
→ or other of their daughters.

```

It is a truth universally acknowledged, that a single man in
possession of a good fortune, must be in want of a wife.

However little known the feelings or views of such a man
may be on his first entering a neighbourhood, this truth is
so well fixed in the minds of the surrounding families, that
he is considered as the rightful property of some one or
other of their daughters.

Example 1.4 Using INDENT to adjust left and (optionally) right margins

1.3 Margins and justification

Example 1.4 demonstrates `INDENT`, which adjusts the left and right margin using the `adjustwidth` environment from the `changepage` package. The first argument defines the inward margin for left and right margins. (Note that the right margin adjustment defaults to zero.) The second argument is the text.

Example 1.5 on the following page demonstrates `FLUSHLEFT`, `FLUSHRIGHT` and `CENTER`, which affect the justification of the contained paragraph(s). Incidentally, it also demonstrates `STO`, which saves (“stores”) some content for a limited amount of time. For more information about `STO`, see ?? on page ??.

Note

There is not currently an example for `STO`.

```
STO text :: 3 :: However little known the feelings or views of such a man may be on
→ his first entering a neighbourhood, this truth is so well fixed in the minds of
→ the surrounding families, that he is considered as the rightful property of some
→ one or other of their daughters.
```

```
FLUSHLEFT ◇text
```

```
CENTER .o pre = lex :: ◇text
```

```
FLUSHRIGHT .o pre = lex :: ◇text
```

However little known the feelings or views of such a man may be on his first entering a neighbourhood, this truth is so well fixed in the minds of the surrounding families, that he is considered as the rightful property of some one or other of their daughters.

However little known the feelings or views of such a man may be on his first entering a neighbourhood, this truth is so well fixed in the minds of the surrounding families, that he is considered as the rightful property of some one or other of their daughters.

However little known the feelings or views of such a man may be on his first entering a neighbourhood, this truth is so well fixed in the minds of the surrounding families, that he is considered as the rightful property of some one or other of their daughters.

Example 1.5 Left, center and right justification

Note

The examples show that `INDENT` and `CENTER`, et cetera, operate only on the paragraph(s) given to them. This raises a question: how do you center, say, a whole block of `LBT` code?

There are two answers. First, you could manually invoke the `center` environment with `BEGIN center`, then place your code, then `END center`. This is low-level and not in the spirit of `LBT`. The second option is to put all your content in a register with `STO .o lbt :: content :: .v << ... >>`, and then use `CENTER ◇content`.

TODO: For this documentation, readers should be directed to a section that demonstrates `STO` in detail.

1.4 Other stuff...

This is a placeholder.

1.5 Passing content through to Latex

The `LATEX` command is simple: it allows you to pass plain Latex code through to the compiler. This is already achievable with the `TEXT` or `TEXT*` command, but the name `LATEX` better represents the intention.

Recall¹ that `CMD` exists for single commands, so cases like `CMD bigskip` are already taken care of. Assuming, then, that you want to pass something more complicated through to Latex, the idiomatic way is to use the `.v << ... >>` verbatim block, which allows you to include newlines in your code.

Example 1.6 on the following page demonstrates typesetting some complex mathematics among some text. Note that `MATH` could handle the mathematics just fine, but the example serves to show the purpose of `LATEX`.

¹**NB:** This is not actually documented anywhere yet

TEXT We now compute the definite integral of $f(x)$ from -1 to 2π :

```
LATEX .v <<
\begin{align*}
\int_{-1}^{2\pi} f(x)\,dx
&= \int_{-1}^0 x^2\,dx
+ \int_0^{\pi} \sin x\,dx
+ \int_{\pi}^{2\pi} 1\,dx \\\
&= \left[\frac{x^3}{3}\right]_{-1}^0
+ \left[-\cos x\right]_0^{\pi}
+ \left[x\right]_{\pi}^{2\pi} \\\
&= \left(0 - \left(-\frac{1}{3}\right)\right)
+ \left(-\cos \pi + \cos 0\right)
+ \left(2\pi - \pi\right) \\\
&= \frac{1}{3} + (1 + 1) + \pi \\\
&= \frac{1}{3} + 2 + \pi
\end{align*}
>>
```

We now compute the definite integral of $f(x)$ from -1 to 2π :

$$\begin{aligned}
\int_{-1}^{2\pi} f(x) dx &= \int_{-1}^0 x^2 dx + \int_0^{\pi} \sin x dx + \int_{\pi}^{2\pi} 1 dx \\
&= \left[\frac{x^3}{3}\right]_{-1}^0 + [-\cos x]_0^{\pi} + [x]_{\pi}^{2\pi} \\
&= \left(0 - \left(-\frac{1}{3}\right)\right) + (-\cos \pi + \cos 0) + (2\pi - \pi) \\
&= \frac{1}{3} + (1 + 1) + \pi \\
&= \frac{1}{3} + 2 + \pi
\end{aligned}$$

Example 1.6 Passing Latex code through with LATEX

2 Lists and tables

The `lbt.Basic` template also implements commands related to lists and tables. The list commands make use of the `enumitem` package, and the table command uses `tabularray`. The commands are:

- `ITEMIZE` for bulleted lists;
- `ENUMERATE` for numbered lists;
- `LIST` for multi-level lists (bulleted and/or numbered);
- `TABLE` for tables

Tables can be inline (default) or floating, in which case you can specify a label, caption and position. Provision is made for loading the data in a table from a file.

2.1 Itemized and enumerated lists

The paragraphs in [Example 1.2](#) on page 3 would be better written as a list. [Example 2.1](#) on the following page shows the chess quotes in an itemized list. [Example 2.2](#) on the next page enumerates some principles of chess opening theory.

`ITEMIZE` and `ENUMERATE` use the `enumitem` package in the background. You can use the kwarg `spec` to pass options through to the underlying `itemize` or `enumerate` environment. Further, the oparg `compact` provides an easy way to tighten the list, as [Example 2.3](#) on page 10 demonstrates.

You can also use opargs `notop` and `sep` to control vertical space in a more specific but still convenient way. See the documentation.

Finally, using `enumitem`'s `newlist` and `setlist` commands, you can define your own list style with the formatting you require. Suppose you now have a `shoppinglist` environment. You can make use of that with the `env` oparg. This is demonstrated in [Example 2.4](#) on page 10

```
TEXT Emanuel Lasker wrote some punchy sentences in his \emph{Manual of Chess}
↳ (1925):
ITEMIZE
:: The chessboard is the world; the pieces are the phenomena of the universe; the
↳ rules of the game are what we call the laws of Nature.
:: The beauty of a move lies not in its appearance but in the thought behind it.
:: In chess, as in life, a man is his own most dangerous opponent.
```

Emanuel Lasker wrote some punchy sentences in his *Manual of Chess* (1925):

- The chessboard is the world; the pieces are the phenomena of the universe; the rules of the game are what we call the laws of Nature.
- The beauty of a move lies not in its appearance but in the thought behind it.
- In chess, as in life, a man is his own most dangerous opponent.

Example 2.1 An itemized list

```
TEXT An experienced player has three key objectives in the opening.
ENUMERATE
:: Focus pawns and/or pieces on the central four squares.
:: Activate the minor pieces.
:: Castle.
```

An experienced player has three key objectives in the opening.

1. Focus pawns and/or pieces on the central four squares.
2. Activate the minor pieces.
3. Castle.

Example 2.2 An enumerated list


```
TEXT An experienced player has three key objectives in the opening.
ENUMERATE .o compact :: (spec) (1)
:: Focus pawns and/or pieces on the central four squares.
:: Activate the minor pieces.
:: Castle.
```

An experienced player has three key objectives in the opening.

- (1) Focus pawns and/or pieces on the central four squares.
- (2) Activate the minor pieces.
- (3) Castle.

Example 2.3 A compact enumerated list with custom label

```
CMD newlist :: shoppinglist :: itemize :: 1
CMD setlist :: [shoppinglist]
:: label=\ding{111}, left=4pt, itemsep = 2pt, topsep = 2pt

TEXT Things to buy on the way home:
ITEMIZE .o env = shoppinglist
:: oat milk
:: tofu
:: vegan cheese
```

Things to buy on the way home:

- ☐ oat milk
- ☐ tofu
- ☐ vegan cheese

Example 2.4 A custom shopping list

```

LIST .o markers = circnum * 49
:: Fruits
:: * Citrus
:: * * Orange
:: * * Lemon
:: * * Lime
:: * Berries
:: * * Strawberry
:: * * Blueberry
:: Vegetables
:: * Leafy greens
:: * * Spinach
:: * * Kale
:: * Root vegetables
:: * * Carrot
:: * * Beetroot
:: * * Potato

```

① Fruits

- Citrus
 - ☞ Orange
 - ☞ Lemon
 - ☞ Lime
- Berries
 - ☞ Strawberry
 - ☞ Blueberry

② Vegetables

- Leafy greens
 - ☞ Spinach
 - ☞ Kale
- Root vegetables
 - ☞ Carrot
 - ☞ Beetroot
 - ☞ Potato

Example 2.5 A multi-level list

2.2 Description lists

Note

These are not implemented yet. Watch this space.

2.3 Automatic multi-level lists

If you want to typeset a multi-level list using standard LaTeX environments, you will end up with a lot of boilerplate. The LBT command `LIST` offers great convenience, as [Example 2.5](#) demonstrates.

The markers chosen are not a great fit for the list content, but they show some of the possibilities. The `49` refers to the dingbats provided by the `pifont` package. See the documentation for more details.

2.4 Tables

LBT makes an opinionated choice that tables are best set using `tabulararray`.¹ The `TABLE` command is a light layer over a `tblr` environment. You provide all the specifications (details of column alignment and cell formatting) in the mandatory `spec` kwarg.

Example 2.6 on the next page shows left, center and right column alignment, and bold top-row headings, and a horizontal line between the headings and the data. This is a very simple table.

Example 2.7 on the following page improves on the above by using the `booktabs` library to gain access to `\toprule`, `\midrule` and `\bottomrule`. It also introduces some padding to the third column, and increases the row separation.

Most tables created with Latex are for displaying information in articles or books. But tables can be used for other purposes, such as educational handouts. **Example 2.8** on page 14 demonstrates the use of space (setting column widths and row heights) and background colour. It also puts most cells in math mode, and shows all borders.

It takes a bit of digging in the `tabulararray` manual to find the right incantations to make all this happen, but ultimately it is easier to achieve the desired results with this package than with any other. And if you need multiple tables with the same format, it is easy to create your own table type with `\NewTblrEnviron`. If you created an `invoice` table, for instance, you could invoke it with `TABLE .o invoice`.

Note

No attempt has been made so far to create tables with spanning cells. Whether any specific support from LBT is required remains to be seen.

Note

The thing about `TABLE .o invoice` is a bald-faced lie. That needs to be implemented. (Easy, though.)

Loading table data from a file

The `TABLE` command makes it easy to load CSV or TSV data from a file and use it as the rows of the table. **Example 2.9** on page 15 demonstrates this with cumulative normal distribution values, which are clearly better located in a data file than in a Latex file.

¹If you really want LBT to support `tabularx` or something else, feel free to request it. Of course, you can implement it yourself, too.

```

TABLE .o center :: (spec) colspec = {l c r}, row{1}={font=\bfseries}
:: Author & Year of Birth & Published Works
:: \hline
:: William Shakespeare & 1564 & 39
:: Jane Austen          & 1775 & 7
:: Charles Dickens      & 1812 & 20
:: Leo Tolstoy          & 1828 & 48

```

Author	Year of Birth	Published Works
William Shakespeare	1564	39
Jane Austen	1775	7
Charles Dickens	1812	20
Leo Tolstoy	1828	48

Example 2.6 Simple table example

```

TABLE .o center :: (spec) colspec = {l c r}, row{1}={font=\bfseries},
↪ cell{2-Z}{3}={appto=\hspace*{3em}}, rowsep=3pt
:: \toprule
:: Author & Year of Birth & Published Works
:: \midrule
:: William Shakespeare & 1564 & 39
:: Jane Austen          & 1775 & 7
:: Charles Dickens      & 1812 & 20
:: Leo Tolstoy          & 1828 & 48
:: \bottomrule

```

Author	Year of Birth	Published Works
William Shakespeare	1564	39
Jane Austen	1775	7
Charles Dickens	1812	20
Leo Tolstoy	1828	48

Example 2.7 Simple table example with more formatting

```

TABLE .o center :: (spec) colspec = {cccc}, hlines, vlines, row{2-Z}={7mm,
↪ mode=math}, column{1-2}={2cm}, column{3-4}={3cm}, row{1}={bg=Turquoise!35,
↪ font=\bfseries}
:: Sum & Product
:: 11 & 18 & 9+2=11 & 9\times2=18
:: 21 & 110 & 10+11=21 & 10\times11=110
:: 12 & 36
:: 18 & 77
:: 16 & 48
:: 13 & 40

```

Sum	Product		
11	18	$9 + 2 = 11$	$9 \times 2 = 18$
21	110	$10 + 11 = 21$	$10 \times 11 = 110$
12	36		
18	77		
16	48		
13	40		

Example 2.8 Table with spacing, color, math mode and lines

```

TABLE .o centre
:: (spec) colspec={rrrrrrrrrrr}, hlines, vlines,    column{1}={bg=gray9,
→ font=\bfseries}, row{1} = {bg=gray9, font=\itshape},    cell{1}{1} = {mode=math}
:: (datafile) media/standard-normal-cumulative.tsv
:: @datarows 1
:: @datarows 22..27

```

<i>z</i>	<i>0.00</i>	<i>0.01</i>	<i>0.02</i>	<i>0.03</i>	<i>0.04</i>	<i>0.05</i>	<i>0.06</i>	<i>0.07</i>	<i>0.08</i>	<i>0.09</i>
2.0	0.9772	0.9778	0.9783	0.9788	0.9793	0.9798	0.9803	0.9808	0.9812	0.9817
2.1	0.9821	0.9826	0.9830	0.9834	0.9838	0.9842	0.9846	0.9850	0.9854	0.9857
2.2	0.9861	0.9864	0.9868	0.9871	0.9875	0.9878	0.9881	0.9884	0.9887	0.9890
2.3	0.9893	0.9896	0.9898	0.9901	0.9904	0.9906	0.9909	0.9911	0.9913	0.9916
2.4	0.9918	0.9920	0.9922	0.9925	0.9927	0.9929	0.9931	0.9932	0.9934	0.9936
2.5	0.9938	0.9940	0.9941	0.9943	0.9945	0.9946	0.9948	0.9949	0.9951	0.9952

Example 2.9 Table with data loaded from a file

3 Mathematical text (lbt.Math) – various macros

The `lbt.Math` template provides several affordances for typing mathematical text.

- The `simplemath` macro is a replacement for the inline and display math environments `$... $` and `$$.. $$` (or their Latex equivalents). It allows you to type mathematical text more succinctly and with fewer backslashes.
- The `integral` macro simplifies typing definite and indefinite (single) integrals.
- The `vector` and `vectorijk` macro greatly simplify typing vectors.
- A collection of macros like `mathlistand`, `mathsum`, `mathseq` and several others provide a convenient way to type mathematical text like “ a, b and c ” or “ $y_1, y_2, y_3, \dots, y_n$ ”.

Also found in `lbt.Math` is the `MATH` command, which aids in the typesetting of display equations. It appears in chapter 4 on page 26.

3.1 The `simplemath` macro

```
\lbtDefineMacros{sm = lbt.Math:simplemath}
```

The `simplemath` macro stands in for `$... $` or `$$.. $$` and allows you to type mathematical text succinctly. It recognises a lot of keywords and abbreviations, meaning far fewer backslashes are needed.

Example 3.1 on the following page demonstrates the use of `simplemath` in both math modes: inline and display. **Example 3.2** on the next page contains a large number of examples to show the variety of conveniences that `simplemath` offers.

Example 3.3 on page 18 shows that brackets, both round and square, are auto-sized (that is, `\left` and `\right` are applied intelligently) and that `text{...}` (or `\text{...}`) is recognised specially, and passed through to Latex without processing its contents. Note, however, that braces (`{}`) are *not* auto-sized.

TEXT Two interesting results from Euler are

» `\sm{ds arctan x = x - frac{x^3}{3} + frac{x^5}{5} - cdots}` and

» `\sm{ sum_{n=1}^{\infty} frac 1 {n^2} = frac {pi^2} 6. }`

Two interesting results from Euler are $\arctan x = x - \frac{x^3}{3} + \frac{x^5}{5} - \dots$ and

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}.$$

Example 3.1 `simplmath` usage, both inline and display form

<code>cos2 th + sin2 th equiv 1</code>	$\cos^2 \theta + \sin^2 \theta \equiv 1$
<code>log_2 n ge 5</code>	$\log_2 n \geq 5$
<code>ds (1+x)^n = sum_{r=0}^n binom n r x^r</code>	$(1+x)^n = \sum_{r=0}^n \binom{n}{r} x^r$
<code>A = B iff A subseq B vee B subseq A</code>	$A = B \iff A \subseteq B \vee B \subseteq A$
<code>ds al be + al ga + be ga = frac {-b} {2a}</code>	$\alpha\beta + \alpha\gamma + \beta\gamma = \frac{-b}{2a}$
<code>ds f'(x) = lim_{h to 0} frac {f(x+h)-f(x)} h</code>	$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$
<code>neg(P implies Q) equiv P vee neg Q</code>	$\neg(P \implies Q) \equiv P \vee \neg Q$
<code>P imp Q equiv neg P wedge Q</code>	$P \implies Q \equiv \neg P \wedge Q$
<code>12 = 2 cdot 2 cdot 3 text{ so } 5 nmid 12</code>	$12 = 2 \cdot 2 \cdot 3 \text{ so } 5 \nmid 12$
<code>ds prod_{i=1}^n x_i = x_1 x_2 cdots x_n</code>	$\prod_{i=1}^n x_i = x_1 x_2 \cdots x_n$
<code>sqrt 2 notin bbQ</code>	$\sqrt{2} \notin \mathbb{Q}$
<code>forall n in bbZ, n^2 in bbZ</code>	$\forall n \in \mathbb{Z}, n^2 \in \mathbb{Z}$
<code>OABC text{ is a parallelogram.}</code>	$OABC$ is a parallelogram.

Example 3.2 A collection of `simplmath` examples

TEXT Brackets will automatically assume the correct size:

» `\sm{ y = (1 + [\frac x 7])^2 quad text{([A]$ is the rounding function)} }`

TEXT Suppress this behaviour by setting the option `\texttt{simplemath.leftright} = false`.

Brackets will automatically assume the correct size:

$$y = \left(1 + \left[\frac{x}{7}\right]\right)^2 \quad ([A] \text{ is the rounding function})$$

Suppress this behaviour by setting the option `simplemath.leftright = false`.

Example 3.3 `simplemath`'s handling of parentheses, brackets and text

3.2 The `integral` macro

Typing integrals in regular Latex, or with `simplemath`, is not exactly a chore. But there is room for simplification, and the `integral` macro allows for definite or easy typing indefinite (single, simple) integrals. If the first argument is `ds` then a `\displaystyle` is inserted.

The resulting Latex code is wrapped in `\ensuremath{\dots}`, so integrals do not need to be inside a math environment.

Example 3.4 shows a definite and indefinite integral in context. **Example 3.5** on the following page contains enough examples so that definite and indefinite are shown, as are normal style and display style.

TEXT The integral `\integral{ds,\frac{\sin x}{x},dx}` has practical importance but
→ can't be evaluated in closed form.

TEXT You can take advantage of `\code{simplemath}` as well:

» `\sm{ \integral{pi/3,pi,sqrt {1 + sin3 th},d th} }`

The integral $\int \frac{\sin x}{x} dx$ has practical importance but can't be evaluated in closed form.
You can take advantage of `simplemath` as well:

$$\int_{\pi/3}^{\pi} \sqrt{1 + \sin^3 \theta} d\theta$$

Example 3.4 The `integral` macro

<code>\integral{\sin x,dx}</code>	$\int \sin x dx$
<code>\integral{1,3,\sqrt{4z},dz}</code>	$\int_1^3 \sqrt{4z} dz$
<code>\integral{ds,\tan y,dy}</code>	$\int \tan y dy$
<code>\integral{ds,1,3,\sqrt{4z},dz}</code>	$\int_1^3 \sqrt{4z} dz$

Example 3.5 A collection of integral examples

3.3 The `vector` macro

With the `vector` macro you can easily typeset \mathbf{a} and \mathbf{b} and \vec{c} . Oh, and \overrightarrow{DE} and $\begin{pmatrix} 3 \\ -7 \end{pmatrix}$ and $(4, 0, 9)$. And finally, $2\mathbf{i} - 5\mathbf{j} + \mathbf{k}$ and $-3\mathbf{i} + 4\mathbf{k}$.

One might choose to set up this macro as `\V` as follows:

```
\lbtDefineMacros{V = \lbt.Math:vector}
```

A thorough set of examples is given in [Example 3.6](#).

<code>\V{3 1 -9} + \V{-2 5 4} = \V{1 6 -5}</code>	$\begin{pmatrix} 3 \\ 1 \\ -9 \end{pmatrix} + \begin{pmatrix} -2 \\ 5 \\ 4 \end{pmatrix} = \begin{pmatrix} 1 \\ 6 \\ -5 \end{pmatrix}$
<code>\V{row 2 7 -1 4 6}</code>	$(2, 7, -1, 4, 6)$
<code>\V{a} + \V{b} = \V{a_1 a_2} + \V{b_1 b_2}</code>	$\mathbf{a} + \mathbf{b} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$
<code>\V{r} = \V{r_1 \vdots r_n}</code>	$\mathbf{r} = \begin{pmatrix} r_1 \\ \vdots \\ r_n \end{pmatrix}$
<code>\V{3 1 -9} = \V{ijk}{3 1 -9}</code>	$\begin{pmatrix} 3 \\ 1 \\ -9 \end{pmatrix} = 3\mathbf{i} + \mathbf{j} - 9\mathbf{k}$
<code>\V{-1 0 4} = \V{ijk}{-1 0 4}</code>	$\begin{pmatrix} -1 \\ 0 \\ 4 \end{pmatrix} = -\mathbf{i} + 4\mathbf{k}$
<code>\V{-1 0} = \V{ijk}{-1 0}</code>	$\begin{pmatrix} -1 \\ 0 \end{pmatrix} = -\mathbf{i}$
<code>\V{ijk 3 4 5}</code>	$3\mathbf{i} + 4\mathbf{j} + 5\mathbf{k}$

Example 3.6 A collection of vector and vectorijk examples

Bold, tilde and arrow

Vectors are most commonly set in bold upright (\mathbf{a}), and that is the LBT default. You can, however, choose tilde (\tilde{a}) or arrow (\vec{a}) instead. [Example 3.7](#) on the

```
* Affect the whole Latex document
\lbtGlobalOpargs{vector.format = tilde}

* Affect one lbt expansion
\begin{lbt}
[@META]
TEMPLATE lbt.Basic
OPTIONS vector.format = arrow
...
\end{lbt}
```

Example 3.7 Vectors in bold or tilde or arrow format, document-wide

TEXT We can write `\vecbold{a}` or `\vectilde{b}` or `\vecarrow{c}`.

We can write **a** or \tilde{b} or \vec{c} .

Example 3.8 Vectors in bold or tilde or arrow format, one-off

following page shows how to make these choices for your LBT expansion or your whole Latex document. If you want just a one-off, you can define and use the following macros. [Example 3.8](#) demonstrates their use.

```
\lbtDefineMacros{vecbold = lbt.Math:vecbold}
\lbtDefineMacros{vectilde = lbt.Math:vectilde}
\lbtDefineMacros{vecarrow = lbt.Math:vecarrow}
```

3.4 List, sequence, summation and product macros

It is frequently necessary to type in a collection of numeric or algebraic terms expressed as a list or a sum or a product. Sometimes the terms are related, like a_1, a_2 etc. Typing such things into Latex can be a minor annoyance if each term is wrapped in dollar signs. There is repeated structure that can be abstracted by a macro, especially with a proper programming language available.

The table below shows some examples of each, with the Latex used to achieve them.

Desired output	Latex code
a, b, c	<code>\$a, b, c\$</code>
13, 14, 15 and 16	<code>\$13\$, \$14\$, \$15\$, and \$16\$</code>
X, Y or Z	<code>\$X\$, \$Y\$, or \$Z\$</code>
a, b, c, \dots, z	<code>\$a, b, c, \dots, z\$</code>
$1 + 2 + 3 + \dots + n$	<code>\$1 + 2 + 3 + \dots + n\$</code>
T_4, T_5, T_6	<code>\$T_4, T_5, T_6\$</code>
$T_4 + T_5 + T_6$	<code>\$T_4 + T_5 + T_6\$</code>
$a_1 a_2 \dots a_N$	<code>\$a_1 a_2 \dots a_N\$</code>
$a_1 \cdot a_2 \cdots a_N$	<code>\$a_1 \backslash cdot a_2 \backslash cdot \dots \backslash cdot a_N\$</code>

LBT offers some macros to make entering such things more convenient, which are demonstrated in [Example 3.9](#) on the following page. See [Section 3.6](#) on page 24 for code to include in your preamble to gain access to these macros.

<code>\mathlist{m,n,p,q,r}</code>	m, n, p, q, r
<code>\mathlistand{a,b,c,d}</code>	$a, b, c \text{ and } d$
<code>\mathlistor{X,Y,Z}</code>	$X, Y \text{ or } Z$
<code>\mathlist{3,4,5,\dots,9}</code>	$3, 4, 5, \dots, 9$
<code>\mathlist{2^0,2^1,2^2,\dots,2^{n-1},2^n}</code>	$2^0, 2^1, 2^2, \dots, 2^{n-1}, 2^n$
<code>\mathsum{a,b,c,d,e}</code>	$a + b + c + d + e$
<code>\mathsum{p_1,p_2,p_3,\dots,p_n}</code>	$p_1 + p_2 + p_3 + \dots + p_n$
<code>\mathproductcdot{p_1,p_2,p_3,\dots,p_n}</code>	$p_1 \cdot p_2 \cdot p_3 \cdots p_n$
<code>\mathseq{T,1,2,3,4,5}</code>	T_1, T_2, T_3, T_4, T_5
<code>\mathseqsum{T,1,2,3,4,5}</code>	$T_1 + T_2 + T_3 + T_4 + T_5$
<code>\mathseqproduct{T,1,2,3,4,5}</code>	$T_1 T_2 T_3 T_4 T_5$
<code>\mathseq{p,1,2,3,\dots,n}</code>	$p_1, p_2, p_3, \dots, p_n$
<code>\mathseqsum{p,22,23,24,\dots,29,30}</code>	$p_{22} + p_{23} + p_{24} + \dots + p_{29} + p_{30}$
<code>\mathseqproduct{p,1,2,\dots,5}</code>	$p_1 p_2 \dots p_5$
<code>\mathseqproductcdot{p,1,2,\dots,5}</code>	$p_1 \cdot p_2 \cdots p_5$

Example 3.9 List/sequence macros such as `mathsum` and `mathseqproduct`

3.5 Miscellaneous macros

LBT offers some other macros that bring convenience to certain tasks in mathematical typesetting.

Currently the only such macro is `primefactorisation`, which is demonstrated in [Example 3.10](#).

<code>\primefactorisation{2,2,2,5,7,7,19}</code>	$2^3 \cdot 5 \cdot 7^2 \cdot 19$
<code>\primefactorisation{explicit 2,2,2,5,7,7,19}</code>	$2^3 \cdot 5^1 \cdot 7^2 \cdot 19^1$

Example 3.10 Miscellaneous macros such as `primefactorisation`

3.6 Code to set up all macros

As a convenience, [Example 3.11](#) on the next page contains the code that you can paste into your preamble to obtain access to all macros described in this chapter. They are grouped for readability.

```
\lbtDefineLatexMacros{
sm                = \bt.Math:simplemath,
integral          = \bt.Math:integral,
}
\lbtDefineLatexMacros{
V                = \bt.Math:vector,
vecbold          = \bt.Math:vecbold,
vecarrow         = \bt.Math:vecarrow,
vectilde         = \bt.Math:vectilde,
}
\lbtDefineLatexMacros{
mathlist         = \bt.Math:mathlist,
mathlistand      = \bt.Math:mathlistand,
mathlistor       = \bt.Math:mathlistor,
mathsum          = \bt.Math:mathsum,
mathproductcdot  = \bt.Math:mathproductcdot,
mathseq          = \bt.Math:mathseq,
mathseqsum       = \bt.Math:mathseqsum,
mathseqproduct   = \bt.Math:mathseqproduct,
mathseqproductcdot = \bt.Math:mathseqproductcdot,
}
\lbtDefineLatexMacros{
primefactorisation = \bt.Math:primefactorisation,
}
```

Example 3.11 Preamble code to set up all `\bt.Math` macros

4 Mathematical text (lbt.Math) – the **MATH** command

The **MATH** command gets its own chapter so that its various features can be displayed one section at a time.

MATH provides for a variety of display equations. It is a portal to various **amsmath** and **mathtools** environments like `split`, `gather`, `align`, and so on. The examples here give a good primer on their use, but readers should consult the relevant documentation to develop greater awareness of the details.

4.1 Opening remarks

Setting a display equation with `\[... \]`¹ is enough for a great many cases. If you want your equation to be numbered, you upgrade to the `equation` environment. If the math content to be displayed is more complicated than that, the author should decide which of the following applies:

- there is one logical equation with several parts (separated by `=` or `>` or `...`) that should appear on separate lines (`split`);
- there is one logical equation that is too long to fit on one line (`multline`);
- there are several logical equations to be displayed together (`gather`);
- there are several logical equations to be displayed reasonably with alignment (`align`);
- there are more complicated alignment requirements, perhaps involving comments to the side (also `align`).

Based on that, the author can choose an **amsmath** environment, which are demonstrated in [Table 4.1](#) on page 28. The table does not show *all* available environments, but it gives a good overview for readers who are not already familiar.

The sections of this chapter give more detailed information on these environments and more.

¹Or the Tex command `$$... $$`, which is lower-level and may produce different vertical spacing from `\[... \]`.

Note

In normal Latex code, equations are numbered by default. If you use the `align` environment then all lines are numbered. If you use `align*` then none of them are.

LBT is similar: use `MATH` to get numbered equations and `MATH*` to suppress numbering. If you want unnumbered equations by default, set the option `MATH.eqnum = false`.

The LBT examples that follow will demonstrate fully numbered, partially numbered, and unnumbered equations, as appropriate to the environment being demonstrated.

Table 4.1 Some environments provided by amsmath

Environment	Example	
<code>equation</code>	$a^2 + b^2 = c^2$	(4.1)
<code>gather</code>	$a^2 + b^2 = c^2$	(4.1)
	$E = mc^2$	(4.2)
	$F = k \frac{q_1 q_2}{r^2}$	(4.3)
<code>align (1)</code>	$a^2 + b^2 = c^2$	(4.1)
	$E = mc^2$	(4.2)
	$F = k \frac{q_1 q_2}{r^2}$	(4.3)
<code>align (2)</code>	$a^2 + b^2 = c^2$	$E = mc^2$ (4.1)
	$F = k \frac{q_1 q_2}{r^2}$	$F = ma$ (4.2)
<code>align (3)</code>	$2^{n+1} = 2 \cdot 2^n$	(4.1)
	$> 2 \cdot n^2$	by assumption (4.2)
	$= n^2 + \frac{1}{2}n^2 + \frac{1}{2}n^2$	(4.3)
	$> n^2 + 2n + 1$	reader to confirm (4.4)
	$= (n + 1)^2$	(4.5)
<code>split (inside equation)</code>	$ \begin{aligned} f'(x) &= \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} \\ &= \lim_{h \rightarrow 0} \frac{(x+h)^2 - x^2}{h} \\ &= \lim_{h \rightarrow 0} \frac{(x^2 + 2xh + h^2) - x^2}{h} \\ &= \lim_{h \rightarrow 0} \frac{2xh + h^2}{h} \\ &= \lim_{h \rightarrow 0} 2x + h \\ &= 2x \end{aligned} $	(4.1)
<code>multline</code>	$ \begin{aligned} (1+x)^n &= \sum_{r=0}^n \binom{n}{r} x^r = 1 + \binom{n}{1}x + \binom{n}{2}x^2 \\ &\quad + \cdots + \binom{n}{r}x^r + \cdots + \binom{n}{n}x^n \end{aligned} $	(4.1)

4.2 Simple equations with MATH (.o equation)

The `equation` environment provides for a simple numbered equation. **Example 4.1** demonstrates this in LBT. `equation` is in fact the default environment, so you can just write `MATH F = ma`, as the example shows.

TEXT Newton's second law is known to
 ↳ many.
 MATH $F = ma$

Newton's second law is known to many.

$$F = ma \quad (4.1)$$

TEXT You can suppress numbering in two
 ↳ ways.
 MATH* $F = ma$
 MATH .o noeqnum :: $F = ma$

You can suppress numbering in two ways.

$$F = ma$$

$$F = ma$$

TEXT `\code{equation}` is the
 ↳ `\emph{default}` environment for
 ↳ `\code{MATH}`, but you can be
 ↳ explicit if you wish.
 MATH .o equation :: $F = ma$

`equation` is the *default* environment for `MATH`, but you can be explicit if you wish.

$$F = ma \quad (4.2)$$

Example 4.1 `MATH .o equation` to format a simple equation

4.3 Long equations with MATH .o multiline

If an equation is too long for one line, you can insert linebreaks and the `amsmath` environment `multiline` will handle formatting with a mixture of left, center and right justification. **Example 4.2** on the following page demonstrates.

Both `MATH .o multiline` and `MATH .o multline` work, and do the same thing.

Note that the example includes the `oparg sm = false` to disable `simplemath`. This is necessary to prevent `be` from being rendered as β .

TEXT The display environment below has its margins adjusted so that the effect of
 \rightarrow `\code{multiline}`

MATH .o multiline, sm = false, adjustwidth = 2cm 2cm

$$\begin{aligned} &:: (a+b+c+d+e)^2 = \\ &:: a^2 + 2ab + 2ac + 2ad + 2ae + b^2 + 2bc + 2bd + 2be \\ &:: + c^2 + 2cd + 2ce + d^2 + 2de + e^2 \end{aligned}$$

The display environment below has its margins adjusted so that the effect of `multiline`

$$\begin{aligned} (a + b + c + d + e)^2 = \\ a^2 + 2ab + 2ac + 2ad + 2ae + b^2 + 2bc + 2bd + 2be \\ + c^2 + 2cd + 2ce + d^2 + 2de + e^2 \quad (4.3) \end{aligned}$$

Example 4.2 A long equation with `multiline`

4.4 gather

If you have a few equations that you want to display at once, and aren't too fussed about how they are presented, then you `gather` them and they will each be centered and numbered. Two examples suffice. [Example 4.3](#) demonstrates the default numbering. [Example 4.4](#) on the following page demonstrates selective numbering. If you want no numbering, just use `MATH*`.

Note that the first example ([Example 4.3](#)) demonstrates the option `linespace`, which provides some extra space between the lines.

TEXT Some interesting series results.
MATH .o gather, linespace = 1ex

$$\begin{aligned} &:: \sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6} \\ &:: \sum_{n=0}^{\infty} \frac{x^n}{n!} = e^x \end{aligned}$$

Some interesting series results.

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6} \quad (4.4)$$

$$\sum_{n=0}^{\infty} \frac{x^n}{n!} = e^x \quad (4.5)$$

Example 4.3 Displaying several equations with `gather`

```

TEXT Area formulas you must know:
MATH .o gather, eqnum = 2 4
:: A = pi r ^2
:: A = pi a b          \label{eq:a1}
:: A = \tfrac{1}{2} b h
:: A = \tfrac{1}{2} x y   \label{eq:a2}
TEXT We'll discuss \eqref{eq:a1} and \eqref{eq:a2} in more detail\dots

```

Area formulas you must know:

$$A = \pi r^2$$

$$A = \pi ab \tag{4.6}$$

$$A = \frac{1}{2}bh$$

$$A = \frac{1}{2}xy \tag{4.7}$$

We'll discuss (4.6) and (4.7) in more detail...

Example 4.4 gather with selective numbering

Building-block variants of gather

Using `gathered` instead of `gather`, you can gather some equations into one logical object, and place that inside of `equation` (or another display environment) to assign an equation number. In addition to `amsmath`'s `gathered`, `mathtools` provides `lgathered` and `rgathered`, which justify the equations left or right within the centered block.

LBT provides `eqgathered` and `eqlgathered` and `eqrgathered` as composite environments which wrap the building-block inside `equation`. These are likely to be of only niche interest. **Example 4.5** and **Example 4.6** on the next page demonstrate these environments. Note the way they assign only one equation number.

```

MATH .o eqgathered
:: sin x = opp/hyp / text{hyp} quad text{and} quad cos x = text{adj} / text{hyp}
:: sin^2 x + cos^2 x equiv 1

```

$$\sin x = \text{opp}/\text{hyp} \quad \text{and} \quad \cos x = \text{adj}/\text{hyp}$$

$$\sin^2 x + \cos^2 x \equiv 1 \tag{4.8}$$

Example 4.5 eqgathered: a single equation number for a gathered block

```

MATH .o eqlgathered
:: sin x = text{opp} / text{hyp} quad text{and} quad cos x = text{adj} / text{hyp}
:: sin2x + cos2x equiv 1
MATH .o eqrgathered
:: sin x = text{opp} / text{hyp} quad text{and} quad cos x = text{adj} / text{hyp}
:: sin2x + cos2x equiv 1

```

$$\sin x = \text{opp}/\text{hyp} \quad \text{and} \quad \cos x = \text{adj}/\text{hyp} \quad (4.9)$$

$$\sin^2 x + \cos^2 x \equiv 1$$

$$\sin x = \text{opp}/\text{hyp} \quad \text{and} \quad \cos x = \text{adj}/\text{hyp} \quad (4.10)$$

$$\sin^2 x + \cos^2 x \equiv 1$$

Example 4.6 The left/right justified variants of `eqgathered`

4.5 Several-part equations with `eqsplit`

The `amsmath` environment `split` is designed for a single logical equation that is broken into two or more lines, like the example below.

$$\begin{aligned} (a+b)^2 &= (a+b)(a+b) \\ &= a^2 + ab + ab + b^2 \\ &= a^2 + 2ab + b^2 \end{aligned} \quad (4.11)$$

However, `split` is a sub-environment that can only occur within a display environment like `equation` or `gather` or `align`. The most common use would be for a `split` appear alone in an `equation` environment, so LBT provides `MATH .o eqsplit` for that purpose.

An `eqsplit` equation gets one number overall, not one number per line, because it is one logical equation. It is the `equation` environment that provides the number, not the `split` environment contained within.

Example 4.7 on the following page demonstrates an unnumbered split equation that is aligned on `=` and `>`. **Example 4.8** on the next page shows a split equation that is numbered and referenced.

Note that it could be desirable to add aligned comments to the right of each line in **Example 4.7** on the following page. Unfortunately that is not possible with `eqsplit` (or the underlying `split`). Later, in **Section 4.6** on the next page, this example will be revisited using the more flexible `align`. However, `align` emits one equation number per line, because it sees each line as a separate equation. So it is not a perfect match for providing side-by-side commentary, but it can be adapted to the task.

```

TEXT Part of a proof by induction.
STO half :: 1 :: $\tfrac 1 2$
MATH* .o eqsplit
:: 2^{n+1} &= 2 \cdot 2^n
::      &> 2 \cdot n^2
::      &= n^2 + \langle half n^2 + \langle half n^2
::      &> n^2 + 2n + 1
::      &= (n+1)^2

```

Part of a proof by induction.

$$\begin{aligned}
 2^{n+1} &= 2 \cdot 2^n \\
 &> 2 \cdot n^2 \\
 &= n^2 + \frac{1}{2}n^2 + \frac{1}{2}n^2 \\
 &> n^2 + 2n + 1 \\
 &= (n+1)^2
 \end{aligned}$$

Example 4.7 Using MATH .o eqsplit for a multi-step equation

```

MATH .o eqsplit, label = eq1
:: (a+b)^2 &= (a+b)(a+b)
::      &= a^2 + ab + ab + b^2
::      &= a^2 + 2ab + b^2

TEXT As shown in \eqref{eq1}, $(a+b)^2$
  ↪ does not equal $a^2 + b^2$!

```

$$\begin{aligned}
 (a+b)^2 &= (a+b)(a+b) \\
 &= a^2 + ab + ab + b^2 \\
 &= a^2 + 2ab + b^2
 \end{aligned} \tag{4.12}$$

As shown in (4.12), $(a+b)^2$ does not equal $a^2 + b^2$!

Example 4.8 A MATH .o eqsplit equation that is referenced

4.6 align and alignat and flalign

The `amsmath` environments `align` and its variants are supported directly by `MATH .o align` and friends to produce mathematical output using specified alignment points. Note that `split` (Section 4.5 on the preceding page) does this as well, but it only allows one alignment point per line. `align` and `alignat` allow as many alignment points as you wish.

The difference between `align` and the others is as follows:

- `align` determines the horizontal spacing between alignment columns itself, while centering the material overall;
- `flalign` (“full-length align”) spreads the columns out as far as possible, using the full text width of the page;
- `alignat` requires the author to specify the number of columns and to control the spacing between them.

There are different logical uses for the alignment environments, as the subsections below demonstrate.

Presenting a group of equations with simple alignment

The mathematical content in [Example 4.9](#), [Example 4.10](#) and [Example 4.11](#) on the next page is the same, but different numbering choices are demonstrated.

These examples use only one alignment point, so the material could technically be typeset by `eqsplit`. This would be the wrong logical choice, however, because these are three equations, not one. Accordingly, `align` produces (up to) three equation numbers whereas `eqsplit` would only produce one.

TEXT All equations numbered (the default).

MATH `.o align`

`:: (a+b)^3 &= a^3 + 3a^2b + 3ab^2 + b^3`

`:: (a-b)(a+b) &= a^2 - b^2`

`:: c^2 &= a^2 + b^2`

All equations numbered (the default).

$$(a + b)^3 = a^3 + 3a^2b + 3ab^2 + b^3 \quad (4.13)$$

$$(a - b)(a + b) = a^2 - b^2 \quad (4.14)$$

$$c^2 = a^2 + b^2 \quad (4.15)$$

Example 4.9 Aligning a group of equations with `align`

TEXT Numbering suppressed.

MATH* `.o align`

`:: (a+b)^3 &= a^3 + 3a^2b + 3ab^2 + b^3`

`:: (a-b)(a+b) &= a^2 - b^2`

`:: c^2 &= a^2 + b^2`

Numbering suppressed.

$$(a + b)^3 = a^3 + 3a^2b + 3ab^2 + b^3$$

$$(a - b)(a + b) = a^2 - b^2$$

$$c^2 = a^2 + b^2$$

Example 4.10 Alignment without numbering

[Example 4.11](#) on the following page shows a special feature of `MATH`: selective numbering. In ordinary `Latex`, you use `\notag` on any line you do not want numbered. (You can do that in `MATH` too if you wish.) The `MATH` oparg `eqnum`

gives you convenient control over which lines are numbered, without editing the lines themselves.

```
TEXT Selective numbering.
MATH .o align, eqnum = 1 3
:: (a+b)^3 &= a^3 + 3a^2b + 3ab^2 + b^3
:: (a-b)(a+b) &= a^2 - b^2
:: c^2 &= a^2 + b^2
```

Selective numbering.

$$(a+b)^3 = a^3 + 3a^2b + 3ab^2 + b^3 \quad (4.16)$$

$$(a-b)(a+b) = a^2 - b^2$$

$$c^2 = a^2 + b^2 \quad (4.17)$$

Example 4.11 Alignment with selective numbering

Aligning equations in multiple columns

Suppose you wanted to demonstrate three kinds of derivative: polynomial, trigonometric and exponential. And you wanted to do so in minimal vertical space. Then you might typeset something like **Example 4.12**.

```
MATH* .o align
:: f(x) &= x^3 - 7x^2 + 4x + 1 & g(x) &= sin(2x) - tan x & h(x) &= 3^x
:: f'(x) &= 6x^2 - 14x + 4 & g'(x) &= 2cos(2x) - sec2x & h'(x) &= 3^x \ln 3
```

$$\begin{array}{lll} f(x) = x^3 - 7x^2 + 4x + 1 & g(x) = \sin(2x) - \tan x & h(x) = 3^x \\ f'(x) = 6x^2 - 14x + 4 & g'(x) = 2 \cos(2x) - \sec^2 x & h'(x) = 3^x \ln 3 \end{array}$$

Example 4.12 Alignment in multiple columns

The space between the “columns” is determined by the `amsmath` package – see the relevant documentation for details. If you want to really spread things out, you can use `flalign`, which uses the “full length” of the page, as shown in **Example 4.13** on the following page.

And if you want to determine your own spacing, you can: the `alignat` environment gives the author that control. **Example 4.14** on the next page demonstrates the use of `\qqquad` to separate the two columns. When you use

```
MATH* .o flalign
:: f(x)  &= x^3 - 7x^2 + 4x + 1  & g(x)  &= sin(2x) - tan x  & h(x)  &= 3^x
:: f'(x) &= 6x^2 - 14x + 4      & g'(x) &= 2cos(2x) - sec2x & h'(x) &= 3^x\ln 3
```

$$\begin{array}{lll} f(x) = x^3 - 7x^2 + 4x + 1 & g(x) = \sin(2x) - \tan x & h(x) = 3^x \\ f'(x) = 6x^2 - 14x + 4 & g'(x) = 2 \cos(2x) - \sec^2 x & h'(x) = 3^x \ln 3 \end{array}$$

Example 4.13 Full-length in multiple columns

`alignat`, you need to provide the oparg `ncols` to specify how many columns there are.²

```
MATH* .o alignat, ncols = 2
:: f(x)  &= x^3 - 7x^2 + 4x + 1  &\hspace{4em} g(x)  &= sin(2x) - tan x
:: f'(x) &= 6x^2 - 14x + 4      & g'(x) &= 2cos(2x) - sec2x
```

$$\begin{array}{ll} f(x) = x^3 - 7x^2 + 4x + 1 & g(x) = \sin(2x) - \tan x \\ f'(x) = 6x^2 - 14x + 4 & g'(x) = 2 \cos(2x) - \sec^2 x \end{array}$$

Example 4.14 Manual control of inter-column spacing

Generally speaking, the default spacing should be sufficient. A more useful purpose for `alignat` is shown in *Multiple alignment points among equations* on page 42, where polynomials have their like terms lined up regardless of the width of coefficients.

Improving the display of a single long equation

Earlier, we saw how `multiline` can be used to manually break up an equation that doesn't fit on one line. The mixed-justification formatting that `multiline` applies may suit some equations but not others. If you prefer a left-justified equation as shown in [Example 4.15](#) on the following page, this can be achieved with `alignat` and a single column. Note the use of `\MoveEqLeft` from the `math-tools` package to place the first line correctly. Also note the use of `` to align the continuation lines nicely.

²Note that in this example, there are two columns and $2(2) - 1 = 3$ ampersands per line. It is helpful to keep this relationship in mind.

```

STO ph :: 1 :: \phantom{=}
MATH .o alignat, ncols = 1, eqnum = 5
:: \MoveEqLeft (x - r_1)(x - r_2)(x - r_3)(x - r_4)
:: quad &= x^4 - (r_1 + r_2 + r_3 + r_4)x^3
::      &\phi + (r_1r_2 + r_1r_3 + r_1r_4 + r_2r_3 + r_2r_4 + r_3r_4)x^2
::      &\phi - (r_1r_2r_3 + r_1r_2r_4 + r_1r_3r_4 + r_2r_3r_4)x
::      &\phi + r_1r_2r_3r_4

```

$$\begin{aligned}
 &(x - r_1)(x - r_2)(x - r_3)(x - r_4) \\
 &= x^4 - (r_1 + r_2 + r_3 + r_4)x^3 \\
 &\quad + (r_1r_2 + r_1r_3 + r_1r_4 + r_2r_3 + r_2r_4 + r_3r_4)x^2 \\
 &\quad - (r_1r_2r_3 + r_1r_2r_4 + r_1r_3r_4 + r_2r_3r_4)x \\
 &\quad + r_1r_2r_3r_4
 \end{aligned} \tag{4.18}$$

Example 4.15 Using alignat and MoveEqLeft to improve the formatting of a long equation

Providing comments to the right

It is common that an author wants to write some brief commentary to the right of a line of working in a multi-step equation. We can achieve this using `align` with two columns, as [Example 4.16](#) on the next page demonstrates.

The inter-column spacing in [Example 4.16](#) on the following page is too large, so we assert manual control using `alignat`, as shown in [Example 4.17](#) on page 39. Note that we insert a `\qqquad` in the *longest* line of working.

In these examples we have been typesetting a single multi-step equation (for which `split` is designed) using `align`, which is designed for multiple logical equations. It is perhaps a shame that the `amsmath` package does not support this use-case—providing commentary on a split equation—more directly.

Having said that, there is the option to use `alignedat`, instead of `alignat`, and place that inside `equation`. `alignedat` does the logical layout without doing any numbering, and `equation` displays the result and assigns a number. LBT supports this combination with `eqalignedat`, as shown in [Example 4.18](#) on page 40.

```

TEXT Part of a proof by induction.
STO half :: 2 :: $\tfrac 1 2$
MATH* .o align
:: 2^{n+1} &= 2 \cdot 2^n
::          &> 2 \cdot n^2                && \text {(by assumption)}
::          &= n^2 + \langle half n^2 + \langle half n^2
::          &> n^2 + 2n + 1                && \text {(reader to confirm)} \tag{*}
::          &= (n+1)^2

TEXT The reader who is interested in tackling (*) might like to consider how we know
  ↳ that $\langle half n^2 > 2n$ and $\langle half n^2 > 1$.

```

Part of a proof by induction.

$$\begin{aligned}
 2^{n+1} &= 2 \cdot 2^n \\
 &> 2 \cdot n^2 && \text{(by assumption)} \\
 &= n^2 + \tfrac{1}{2}n^2 + \tfrac{1}{2}n^2 \\
 &> n^2 + 2n + 1 && \text{(reader to confirm)} && (*) \\
 &= (n+1)^2
 \end{aligned}$$

The reader who is interested in tackling (*) might like to consider how we know that $\frac{1}{2}n^2 > 2n$ and $\frac{1}{2}n^2 > 1$.

Example 4.16 Using MATH .o align for a multi-step equation with commentary

```

TEXT Part of a proof by induction.
STO half :: 2 :: $\tfrac 1 2$
MATH* .o alignat, ncols = 2
:: 2^{n+1} &= 2 \cdot 2^n
::          &> 2 \cdot n^2                && \text {(by assumption)}
::          &= n^2 + \langle half n^2 + \langle half n^2 \quad \&\& \text {}
::          &> n^2 + 2n + 1                && \text {(reader to confirm)} \quad \tag{*}
::          &= (n+1)^2

TEXT The reader who\dots

```

Part of a proof by induction.

$$\begin{aligned}
 2^{n+1} &= 2 \cdot 2^n \\
 &> 2 \cdot n^2 && \text{(by assumption)} \\
 &= n^2 + \tfrac{1}{2}n^2 + \tfrac{1}{2}n^2 \\
 &> n^2 + 2n + 1 && \text{(reader to confirm)} && (*) \\
 &= (n+1)^2
 \end{aligned}$$

The reader who...

Example 4.17 Using MATH .o alignat to improve the previous example

TEXT Part of a proof by induction.

STO half :: 1 :: $\frac{1}{2}$

MATH .o eqalignedat, ncols = 2, label = eq:induc

:: $2^{n+1} = 2 \cdot 2^n$

:: $> 2 \cdot n^2$ && text {(by assumption)}

:: $= n^2 + \frac{1}{2}n^2 + \frac{1}{2}n^2$ \quad && text {}

:: $> n^2 + 2n + 1$ && text {(reader to confirm)}

:: $= (n+1)^2$

TEXT The techniques in \eqref{eq:induc} should be mastered by all students.

Part of a proof by induction.

$$\begin{aligned}
 2^{n+1} &= 2 \cdot 2^n \\
 &> 2 \cdot n^2 && \text{(by assumption)} \\
 &= n^2 + \frac{1}{2}n^2 + \frac{1}{2}n^2 && (4.19) \\
 &> n^2 + 2n + 1 && \text{(reader to confirm)} \\
 &= (n+1)^2
 \end{aligned}$$

The techniques in (4.19) should be mastered by all students.

Example 4.18 Using MATH .o eqalignedat to align a single logical equation

Incorporating lines of text

Sometimes you want to interrupt a sequence of aligned equations to provide a line or even paragraph of commentary. This is achieved with the `amsmath` command `\intertext` or (if tighter vertical spacing is desired) the `mathtools` command `\shortintertext`. In LBT you can use these as you would in plain LaTeX, or you can use the more LBT-ish versions `TEXT` and `SHORTTEXT`. [Example 4.19](#) demonstrates.

```
MATH .o align, eqnum = 3 5
::      S_n &= a + ar + ar^2 + dots + ar^{n-1}
:: TEXT Multiply both sides by $r$, $:
::      r S_n &= ar + ar^2 + ar^3 + dots + ar^n
:: TEXT Now subtract the second equation from the first:
:: S_n - r S_n &= (a + ar + ar^2 + dots + ar^{n-1}) - (ar + ar^2 + dots + ar^n)
:: TEXT All terms cancel except the first and the last:
:: (1 - r) S_n &= a - ar^n
:: TEXT Finally, divide both sides by $1-r$ (assuming $r \neq 1$):
::      S_n &= \frac{a(1 - r^n)}{1 - r}
```

$$S_n = a + ar + ar^2 + \dots + ar^{n-1}$$

Multiply both sides by r :

$$rS_n = ar + ar^2 + ar^3 + \dots + ar^n$$

Now subtract the second equation from the first:

$$S_n - rS_n = (a + ar + ar^2 + \dots + ar^{n-1}) - (ar + ar^2 + \dots + ar^n) \quad (4.20)$$

All terms cancel except the first and the last:

$$(1 - r)S_n = a - ar^n$$

Finally, divide both sides by $1 - r$ (assuming $r \neq 1$):

$$S_n = \frac{a(1 - r^n)}{1 - r} \quad (4.21)$$

Example 4.19 Including a line of text amidst an aligned block

Note

It would be good to demonstrate `SHORTTEXT` as well.

Multiple alignment points among equations

Aligning equations near the left margin

4.7 Other environments

`split`

Revisit this text in light of it being in the “other” section

`MATH` provides the `split` option to access the `split` environment, but it is not likely to be all that useful, because of the need to enclose it in another environment. The example below shows the LBT code and resulting LaTeX code.

```
MATH .o split
:: (a+b)^2 &= (a+b)(a+b)
::      &= a^2 + ab + ab +
↳ b^2
::      &= a^2 + 2ab + b^2
```

```
\begin{split}
\ensuremath{\left(a+b\right)^2} &= \left(a+b\right)\left(a+b\right)
\ensuremath{\&= a^2 + ab + ab + b^2} \\
\ensuremath{\&= a^2 + 2ab + b^2}
\end{split}
\par
```

4.8 Additional environments

`MATH`’s main *raison d’être* is to provide convenient LBT-style access to the `amsmath` and `mathtools` environments. However, two use cases demonstrated in [Example 4.15](#) on page 37 (allowing an equation to “continue” over multiple lines) and [Example 4.18](#) on page 40 (providing neatly aligned “commentary” to the right) have some boilerplate code that could be abstracted away. Thus `MATH` provides the `continuation` and `commentary` pseudo-environments.

The two examples cited above are reproduced inline in the subsections below, for convenience, to provide an easy comparison between the `amsmath`-style approach and the LBT-style approach.

`continuation`

[Example 4.15](#) on page 37 demonstrated a long equation continuing over several lines with left justification and attention to alignment. The example is reproduced below for convenience.

```

STO ph :: 1 :: \phantom{=}
MATH .o alignat, ncol = 1, eqnum = 5
:: \MoveEqLeft (x - r_1)(x - r_2)(x - r_3)(x - r_4)
:: quad &= x^4 - (r_1 + r_2 + r_3 + r_4)x^3
::      &\phi + (r_1r_2 + r_1r_3 + r_1r_4 + r_2r_3 + r_2r_4 + r_3r_4)x^2
::      &\phi - (r_1r_2r_3 + r_1r_2r_4 + r_1r_3r_4 + r_2r_3r_4)x
::      &\phi + r_1r_2r_3r_4

```

$$\begin{aligned}
& (x - r_1)(x - r_2)(x - r_3)(x - r_4) \\
&= x^4 - (r_1 + r_2 + r_3 + r_4)x^3 \\
&\quad + (r_1r_2 + r_1r_3 + r_1r_4 + r_2r_3 + r_2r_4 + r_3r_4)x^2 \\
&\quad - (r_1r_2r_3 + r_1r_2r_4 + r_1r_3r_4 + r_2r_3r_4)x \\
&\quad + r_1r_2r_3r_4
\end{aligned} \tag{4.22}$$

`MATH` provides a `continuation` option that allows the same result to be achieved more easily, relieving authors of the need to be experts in `phantom` and `MoveEqLeft`. It is demonstrated in [Example 4.20](#).

```

MATH .o continuation
:: (x - r_1)(x - r_2)(x - r_3)(x - r_4)
:: = x^4 - (r_1 + r_2 + r_3 + r_4)x^3
::   + (r_1r_2 + r_1r_3 + r_1r_4 + r_2r_3 + r_2r_4 + r_3r_4)x^2
::   - (r_1r_2r_3 + r_1r_2r_4 + r_1r_3r_4 + r_2r_3r_4)x
::   + r_1r_2r_3r_4

```

$$\begin{aligned}
& (x - r_1)(x - r_2)(x - r_3)(x - r_4) \\
&= x^4 - (r_1 + r_2 + r_3 + r_4)x^3 \\
&\quad + (r_1r_2 + r_1r_3 + r_1r_4 + r_2r_3 + r_2r_4 + r_3r_4)x^2 \\
&\quad - (r_1r_2r_3 + r_1r_2r_4 + r_1r_3r_4 + r_2r_3r_4)x \\
&\quad + r_1r_2r_3r_4
\end{aligned} \tag{4.23}$$

Example 4.20 `continuation` supporting neat display of a long equation

The code produced behind the scenes is the same as the code demonstrated above, using `alignat` and `MoveEqLeft` and `phantom`.

Note that when using `MATH .o continuation`, an equation number is applied to the final line. Naturally, the equation number is suppressed if `MATH*` is used.

commentary

Example 4.16 on page 38 demonstrated a multi-line equation with some commentary to the right. It is reproduced below for convenience, with an equation number appended this time.

```
STO half :: 2 :: $\tfrac 1 2$
MATH .o alignat, ncols = 2, eqnum = 5
:: 2^{n+1} &= 2 \cdot 2^n
::      &> 2 \cdot n^2                && text {(by assumption)}
::      &= n^2 + \langle half n^2 + \langle half n^2 \quad && text {}
::      &> n^2 + 2n + 1                && text {(reader to confirm)} \tag{*}
::      &= (n+1)^2
```

$$\begin{aligned}
 2^{n+1} &= 2 \cdot 2^n \\
 &> 2 \cdot n^2 && \text{(by assumption)} \\
 &= n^2 + \frac{1}{2}n^2 + \frac{1}{2}n^2 \\
 &> n^2 + 2n + 1 && \text{(reader to confirm)} && (*) \\
 &= (n+1)^2 && (4.24)
 \end{aligned}$$

MATH provides a **commentary** option that allows the same result to be achieved more easily. It is demonstrated in **Example 4.21** on the next page. The author is relieved of the burden of recalling **alignat** syntax, and the comments will be aligned sympathetically with the longest expression without needing to control that manually.

To use **commentary**, the **|** symbol must be preceded by at least two spaces, so that it can be distinguished from a mathematical use of the same symbol. Naturally, a greater amount of space will produce more readable input text. A trailing **\tag{...}** command is detected and treated separately from the comment.

Note that alignment markers must be provided in **commentary**, whereas they are automatic in **continuation**.

Note

This inconsistency (automatic vs manual alignment markers) may be removed in time, once these environments see more real use.

Like with **continuation**, **MATH .o commentary** will apply an equation number to the final line, and **MATH*** will suppress it.

```
STO half :: 2 :: $\tfrac 1 2$
```

```
MATH .o commentary
```

```
:: 2^{n+1} &= 2 \cdot 2^n
```

```
::      &> 2 \cdot n^2 | (by assumption)
```

```
::      &= n^2 + \langle half n^2 + \langle half n^2
```

```
::      &> n^2 + 2n + 1 | (reader to confirm) \tag{*}
```

```
::      &= (n+1)^2
```

$$\begin{aligned}
 2^{n+1} &= 2 \cdot 2^n \\
 &> 2 \cdot n^2 && \text{(by assumption)} \\
 &= n^2 + \frac{1}{2}n^2 + \frac{1}{2}n^2 \\
 &> n^2 + 2n + 1 && \text{(reader to confirm)} && (*) \\
 &= (n+1)^2 && (4.25)
 \end{aligned}$$

Example 4.21 commentary supporting aligned text to the right of the multi-line equation

4.9 Combinations

4.10 Summary of the MATH command

Part II

Non-core built-in templates

5 Worksheet or exam questions with lbt.Questions

The `lbt.Questions` template offers useful commands for typesetting questions, subquestions, and multiple-choice options.

Use `Q` for a top-level question and `QQ` for a question part (see [Example 5.1](#)).

[Example 5.2](#) on the next page shows the use of `QQ*` to lay out questions parts horizontally.

Use `MC` or `MC*` to lay out **multiple-choice options** (see [Example 5.3](#) on the following page).

A question can have a **source** and/or a **note** preceding the text, and you can change the colour of the question marker (see [Example 5.4](#) on the next page).

If you want *all* questions in your document to have a purple marker, you can include the line `OPTIONS Q.color = purple` in the `[@META]` part of your LBT environment.

`Q` Name three different kinds of clouds.

`Q` Evaluate the following.

`QQ` $3 + 12 / 4$

`QQ` $$(3 + 12) / 4$$

Question 1 Name three different kinds of clouds.

Question 2 Evaluate the following.

(a) $3 + 12/4$

(b) $(3 + 12)/4$

Example 5.1 Question parts and subparts

Q How many vowels appear in each word?

QQ* [ncols=3]

:: appear :: Augustine :: crimson :: toast :: glyph :: transformer

Question 3 How many vowels appear in each word?

(a) appear

(b) Augustine

(c) crimson

(d) toast

(e) glyph

(f) transformer

Example 5.2 Arranging question parts horizontally

Q Which planet of the solar system has the most moons?

MC Earth :: Mars :: Jupiter :: Saturn

Q Which planet of the solar system has the fewest moons?

MC* [ncols=4] :: Mercury :: Venus :: Uranus :: Neptune

Question 4 Which planet of the solar system has the most moons?

(A) Earth

(B) Mars

(C) Jupiter

(D) Saturn

Question 5 Which planet of the solar system has the fewest moons?

(A) Mercury

(B) Venus

(C) Uranus

(D) Neptune

Example 5.3 Multiple choice answers

Q .o color=purple :: (source) HSC 2005 :: (note) sigma notation

:: Evaluate $\sum_{n=3}^5 (2n+1)$.

Question 6 [HSC 2005] (*sigma notation*) Evaluate $\sum_{n=3}^5 (2n+1)$.

Example 5.4 Question source and notes

Part III

Creating a new template

Part IV

Extra features