LBT by Example

Gavin Sinclair

Document under active development from May 2025
Current version: 23 June 2025

Contents

List of Examples

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 **??**.

1.1 Document divisions

All the Latex commands are present. We present `CHAPTER` and `SECTION` in Example 1.1, 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 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 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 **??**.

```
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.

Recall[1] 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 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 $2\pi$:
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 would be better written as a list. Example 2.1 shows the chess quotes in an itemized list. Example 2.2 enumerates some principles of chess opening theory.

`ITEMIZE` and `ENUMERATE` use the enumitem package in the backgound. 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 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

2.2 Description lists

Note

These are not implemented yet. Watch this space.

```
TEXT Emanual 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.
```

Emanual 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 mutli-level list

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 tabularray.[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.

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

Example 2.6 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 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 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 tabularray 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 demonstrates this with cumulative normal distribution values, which are clearly better located in a data file than in a Latex file.

```
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	$9+2=11$	$9\times2=18$
21	110	$10+11=21$	$10\times11=110$
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
```

z	*0.00*	*0.01*	*0.02*	*0.03*	*0.04*	*0.05*	*0.06*	*0.07*	*0.08*	*0.09*
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 `integal` 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 "$y_1, y_2, y_3, \ldots, y_n$".

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

3.1 The `simplemath` macro

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

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

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

Example 3.3 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 $\arctan x = x - \frac{x^3}{3} + \frac{x^5}{5} - \cdots$ and

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

Example 3.1 simplemath usage, both inline and display form

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

Example 3.2 A collection of simplemath 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{...}`, 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 contains enough examples so that definite and indefinte 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

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

Example 3.5 A collection of `integral` examples

3.3 The vector macro

With the vector macro you can easily typeset $\mathbf{a}$ and $\underset{\sim}{b}$ 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.

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

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 ($\underset{\sim}{a}$) or arrow ($\vec{a}$) instead. Example 3.7 shows

```
* 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 $\mathbf{a}$ or $\underset{\sim}{b}$ or $\vec{c}$.

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

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	`$a, b, c$`
13, 14, 15 and 16	`$13$, $14$, $15$, and $16$`
X, Y or Z	`$X$, $Y$, or $Z$`
$a, b, c, \dots, z$	`$a, b, c, \dots , z$`
$1 + 2 + 3 + \cdots + n$	`$1 + 2 + 3 + \dots + n$`
T_4, T_5, T_6	`$T_4, T_5, T_6$`
$T_4 + T_5 + T_6$	`$T_4 + T_5 + T_6$`
$a_1 a_2 \dots a_N$	`$a_1 a_2 \dots a_N$`
$a_1 \cdot a_2 \cdot \cdots \cdot a_N$	`$a_1 \cdot a_2 \cdot \dots \cdot a_N$`

LBT offers some macros to make entering such things more convenient, which are demonstrated in Example 3.9. See Section 3.6 for code to include in your preamble to gain access to these macros.

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

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.

`\primefactorisation{2,2,2,5,7,7,19}`	$2^3 \cdot 5 \cdot 7^2 \cdot 19$
`\primefactorisation{explicit 2,2,2,5,7,7,19}`	$2^3 \cdot 5^1 \cdot 7^2 \cdot 19^1$

Example 3.10 Miscellaneous macros such as `primefactorisation`

3.6 Code to set up all macros

As a convenience, Example 3.11 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                 = lbt.Math:simplemath,
integral           = lbt.Math:integral,
}
\lbtDefineLatexMacros{
V                  = lbt.Math:vector,
vecbold            = lbt.Math:vecbold,
vecarrow           = lbt.Math:vecarrow,
vectilde           = lbt.Math:vectilde,
}
\lbtDefineLatexMacros{
mathlist           = lbt.Math:mathlist,
mathlistand        = lbt.Math:mathlistand,
mathlistor         = lbt.Math:mathlistor,
mathsum            = lbt.Math:mathsum,
mathproductcdot    = lbt.Math:mathproductcdot,
mathseq            = lbt.Math:mathseq,
mathseqsum         = lbt.Math:mathseqsum,
mathseqproduct     = lbt.Math:mathseqproduct,
mathseqproductcdot = lbt.Math:mathseqproductcdot,
}
\lbtDefineLatexMacros{
primefactorisation = lbt.Math:primefactorisation,
}
```

Example 3.11 Preamble code to set up all lbt.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. 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.

[1] Or 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 numbereding. If you want unnumbered equations by default, set the oparg `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
`equation`	$a^2 + b^2 = c^2 \quad (4.1)$
`gather`	$a^2 + b^2 = c^2 \quad (4.1)$ $E = mc^2 \quad (4.2)$ $F = k\frac{q_1 q_2}{r^2} \quad (4.3)$
`align` (1)	$a^2 + b^2 = c^2 \quad (4.1)$ $E = mc^2 \quad (4.2)$ $F = k\frac{q_1 q_2}{r^2} \quad (4.3)$
`align` (2)	$a^2 + b^2 = c^2 \quad E = mc^2 \quad (4.1)$ $F = k\frac{q_1 q_2}{r^2} \quad F = ma \quad (4.2)$
`align` (3)	$2^{n+1} = 2 \cdot 2^n \quad (4.1)$ $> 2 \cdot n^2 \quad$ by assumption $\quad (4.2)$ $= n^2 + \frac{1}{2}n^2 + \frac{1}{2}n^2 \quad (4.3)$ $> n^2 + 2n + 1 \quad$ reader to confirm $\quad (4.4)$ $= (n+1)^2 \quad (4.5)$
`split` (inside `equation`)	$$\begin{aligned} f'(x) &= \lim_{h\to 0} \frac{f(x+h) - f(x)}{h} \\ &= \lim_{h\to 0} \frac{(x+h)^2 - x^2}{h} \\ &= \lim_{h\to 0} \frac{(x^2 + 2xh + h^2) - x^2}{h} \\ &= \lim_{h\to 0} \frac{2xh + h^2}{h} \\ &= \lim_{h\to 0} 2x + h \\ &= 2x \end{aligned} \quad (4.1)$$
`multline`	$$(1+x)^n = \sum_{r=0}^{n} \binom{n}{r} x^r = 1 + \binom{n}{1}x + \binom{n}{2}x^2 + \cdots + \binom{n}{r}x^r + \cdots + \binom{n}{n}x^n \quad (4.1)$$

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

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

Newton's second law is known to many.

$$F = ma \tag{4.1}$$

You can suppress numbering in two ways.

$$F = ma$$

$$F = ma$$

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

$$F = ma \tag{4.2}$$

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 multline will handle formatting with a mixture of left, center and right justification. Example 4.2 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
↪  \code{multiline}

MATH .o multiline, sm = false, adjustwidth = 2cm 2cm
:: (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
```

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

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

Example 4.2 A long equation with `multiline`

4.4 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.

$$\begin{aligned}(a+b)^2 &= (a+b)(a+b) \\ &= a^2 + ab + ab + b^2 \\ &= a^2 + 2ab + b^2\end{aligned} \tag{4.4}$$

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 contained `split`.

Example 4.3 demonstrates an unnumbered split equation that is aligned on = and >. Example 4.4 shows a split equation that is numbered and referenced.

Note that it could be desirable to add aligned comments off to the right in Example 4.3. Unfortunately that is not possible with `eqsplit` (or the underlying `split`). Later, in Section 4.5, this example will be revisited. The problem there is that `align` gives one number per line, because it sees each line as a separate equation.

```
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 + ◊half n^2 + ◊half n^2
::         &> n^2 + 2n + 1
::         &= (n+1)^2
```

Part of a proof by induction.

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

Example 4.3 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 $a^2 + b^2$!
```

$$\begin{aligned} (a+b)^2 &= (a+b)(a+b) \\ &= a^2 + ab + ab + b^2 \\ &= a^2 + 2ab + b^2 \end{aligned} \tag{4.5}$$

As shown in (4.5), $(a+b)^2$ does not equal $a^2 + b^2$!

Example 4.4 A `MATH .o eqsplit` equation that is referenced

4.5 `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 4.4) 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 4.5, Example 4.6 and Example 4.7 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).

$$(a+b)^3 = a^3 + 3a^2b + 3ab^2 + b^3 \tag{4.6}$$
$$(a-b)(a+b) = a^2 - b^2 \tag{4.7}$$
$$c^2 = a^2 + b^2 \tag{4.8}$$

Example 4.5 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 4.6 Alignment without numbering

Example 4.7 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.

$$(a+b)^3 = a^3 + 3a^2b + 3ab^2 + b^3 \tag{4.9}$$
$$(a-b)(a+b) = a^2 - b^2$$
$$c^2 = a^2 + b^2 \tag{4.10}$$

Example 4.7 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 4.8.

```
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
```

$$f(x) = x^3 - 7x^2 + 4x + 1 \qquad g(x) = \sin(2x) - \tan x \qquad h(x) = 3^x \tag{4.11}$$
$$f'(x) = 6x^2 - 14x + 4 \qquad g'(x) = 2\cos(2x) - \sec^2 x \qquad h'(x) = 3^x \ln 3 \tag{4.12}$$

Example 4.8 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 4.9. Numbering is suppressed in that example to show the effect better.

And if you want to determine your own spacing, you can: the `alignat` environment gives the author that control. Example 4.10 demonstrates the use of `\qquad` to separate the two columns. When you use `alignat`, you need to provide the oparg `ncols` to specify how many columns there are.[2]

[2]Note 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.

```
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
```

$$f(x) = x^3 - 7x^2 + 4x + 1 \qquad g(x) = \sin(2x) - \tan x \qquad h(x) = 3^x$$
$$f'(x) = 6x^2 - 14x + 4 \qquad g'(x) = 2\cos(2x) - \sec^2 x \qquad h'(x) = 3^x \ln 3$$

Example 4.9 Full-length in multiple columns

```
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
```

$$f(x) = x^3 - 7x^2 + 4x + 1 \qquad g(x) = \sin(2x) - \tan x$$
$$f'(x) = 6x^2 - 14x + 4 \qquad g'(x) = 2\cos(2x) - \sec^2 x$$

Example 4.10 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 35, 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 4.11, this can be achieved with `alignat` and a single column. Note the use of `\MoveEqLeft` from the mathtools package to place the first line correctly. Also note the use of `\phantom{=\ }` to align the continuation lines nicely.

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 4.12 demonstrates.

```
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
::      &\◊ph + (r_1r_2 + r_1r_3 + r_1r_4 + r_2r_3 + r_2r_4 + r_3r_4)x^2
::      &\◊ph - (r_1r_2r_3 + r_1r_2r_4 + r_1r_3r_4 + r_2r_3r_4)x
::      &\◊ph + r_1r_2r_3r_4
```

$$\begin{aligned}(x-r_1)(x-r_2)(x-r_3)(x-r_4)&\\ &= x^4-(r_1+r_2+r_3+r_4)\,x^3\\ &\quad+(r_1r_2+r_1r_3+r_1r_4+r_2r_3+r_2r_4+r_3r_4)\,x^2\\ &\quad-(r_1r_2r_3+r_1r_2r_4+r_1r_3r_4+r_2r_3r_4)\,x\\ &\quad+r_1r_2r_3r_4 \end{aligned} \tag{4.13}$$

Example 4.11 Using `alignat` and `MoveEqLeft` to improve the formatting of a long equation

The inter-column spacing in Example 4.12 is too large, so we assert manual control using `alignat`, as shown in Example 4.13. Note that we insert a `\qquad` 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 4.14.

Incorporating lines of text

Multiple alignment points among equations

Aligning equations near the left margin

```
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 + ◊half n^2 + ◊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 $◊half n^2 > 2n$ and $◊half n^2 > 1$.
```

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 4.12 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 + ◊half n^2 + ◊half n^2   \qquad && text {}
::         &> n^2 + 2n + 1                  && text {(reader to confirm)} \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 4.13 Using `MATH .o alignat` to improve the previous example

```
TEXT Part of a proof by induction.
STO half :: 1 :: $\tfrac 1 2$
MATH .o eqalignedat, ncols = 2, label = eq:induc
:: 2^{n+1} &= 2 cdot 2^n
::         &> 2 cdot n^2                    && text {(by assumption)}
::         &= n^2 + ◊half n^2 + ◊half n^2   \qquad && 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.

$$\begin{aligned} 2^{n+1} &= 2 \cdot 2^n \\ &> 2 \cdot n^2 && \text{(by assumption)} \\ &= n^2 + \tfrac{1}{2}n^2 + \tfrac{1}{2}n^2 \\ &> n^2 + 2n + 1 && \text{(reader to confirm)} \\ &= (n+1)^2 \end{aligned} \tag{4.14}$$

The techniques in (4.14) should be mastered by all students.

Example 4.14 Using `MATH .o eqalignedat` to align a single logical equation

4.6 gather

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

4.8 Combinations

4.9 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 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).

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).

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

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

Example 5.4 Question source and notes

Part III

Creating a new template

Part IV

Extra features