

An Informal Guide to L^AT_EX

Neil Carter

27th March 2009

Contents

1 Terminology	2
2 A Section	3
2.1 A subsection	4
3 Lists	5
4 Fonts	5
5 Text Symbols	6
6 Misc Symbols	7
7 Maths Symbols	8
8 Program Listings	9
9 Pictures	10
9.1 Displaying Graphical Files	11
9.2 Drawing Pictures with L ^A T _E X	12
10 Matrices	13
11 Tables	13
12 Networks	16
13 Miscellaneous Effects	18
14 Packages	19

List of Figures

1	Just a picture	11
2	This text appears in the Table of Figures	12
3	Figure 3's entry in table-of-figures	12
4	Sine wave drawn in the picture environment	12

List of Tables

1	Table 1's entry in table-of-figures	13
2	A long table that continues on multiple pages	16

Abstract

This is a template for the L^AT_EX article class. It provides examples of the most commonly-used features, with some explanations. The output (PDF) document should be read in parallel with the source (L^AT_EX) document since not all the commands used in the source are described in the output. To create this abstract, I used the `\abstract` command, but I could also have placed it between `\begin{abstract}` and `\end{abstract}` (an abstract *environment*).

Two very good sources of information on T_EX and L^AT_EX are <http://people.ee.ethz.ch/~oetiker/lshort/lshort.pdf> (The Not So Short Introduction to L^AT_EX), and <http://www.tex.ac.uk/cgi-bin/textfaq2html> (T_EX Frequently Asked Questions — the dot is not missing from the URL).

Comments and suggestions are welcome; please use the feedback form at <http://psy.swansea.ac.uk/staff/carter/>.

Thanks go to the various contributors to the Usenet newsgroup **comp.text.tex**, to the authors of the FAQ and The Not So Short Guide, and to Dr Ulrich Berger of Swansea University, for introducing me to L^AT_EX in the first place.

1 Terminology

Command Commands are a single special character such as a percent sign (%), or start with a backslash followed by a series of letters (possibly with a character, such as the at-sign (@), in between). Commands are sometimes followed by square brackets, which contain optional arguments, and sometimes by curly brackets, which contain required parameters, and sometimes both.

Declaration A declaration is a command that acts like a switch; it turns on some feature, which remains on until it is turned off by an `\end` command, or a right curly bracket (not necessarily the first one after the declaration) (not necessarily the first one after the declaration). It doesn't actually produce any text of its own, it just changes the appearance of subsequent text. For instance, `\em` is a declaration that has the same effect as the `\emph` command, which emphasises the following text.

Environment An environment is created by surrounding text with a begin-end pair. The begin and end commands are followed by curly braces which contain the name of the environment. For instance, this list is contained between a `\begin{description}` and an `\end{description}`.

Note that there are often equivalents between commands and declarations. For instance, the `\textbf{}` command is equivalent to the `\bfseries` declaration. In addition, all declarations have equivalent environments, where the name of the environment is the same as

the declaration with the backslash (e.g. `\bfseries` is equivalent to `\begin{bfseries}`...).

Package A package is a set of files that add some functionality to L^AT_EX. Standard distributions of L^AT_EX come with a variety of packages, and there are many more optional packages that are donated by users and that can be downloaded from www.ctan.org. To include a package's features in a document, use the `\usepackage` command.

2 A Section

If this document had the `report` class, we could also have a `\chapter`. (In the source file, note that the `\backslash` command has to be surrounded by dollar symbols.)

To change the spacing between lines, we use the `\doublespacing` command.

We could also use the `\renewcommand` command to change the values of `\baselinestretch` and `\linespread` to give the same effect, but they affect everything in the document, including tables and footnotes!

L^AT_EX starts a new paragraph where it finds a blank line in the source (or with the `\par` command, which is optional; I leave it out for simplicity). Normally, L^AT_EX indents the first line of each paragraph. However, the first paragraph after any break in the flow of paragraphs, say after a new section, or figure is not indented, but all the following paragraphs are until the next break.

To return to single-spaced lines, we use the `\singlespacing` command. But note that we need to place a blank line between the command and its preceding paragraph, otherwise the single-spacing would be applied to its preceding paragraph.

And now we are back to the default single-spacing.

Because of the general structure of this document, the indentation applied to each paragraph makes it harder on the eye (in my opinion). We can prevent this indentation by reducing it to zero with

```
\setlength{\parindent}{0pt}
```

But if we do this, the text becomes cramped vertically, so we increase the gap between paragraphs with

```
\setlength{\parskip}{2ex plus 0.5ex minus 0.5ex}
```

The **plus** and **minus** values tell latex how much it can vary the gap to fit things on the page. Note that this also affects the space after headings. We can use a variety of measures for values like this, such as **mm** (millimetres), **in** (inches), **pt** (points), and **ex** (height of the letter x in the current font), and so on.

The problem with changing the parskip value is that it also affects the gaps around lists, headings, and tables-of-contents.

We can force a new line to be started in the middle of some text, say just after this line

by using two backslashes (`\`). Note that this does not start a new paragraph and that the space between the truncated line and the following one is not large like the gap between the real paragraphs in this document (well, since we increased the gap).

To show two backslashes in the actual document (and not have them treated as a new-line command, use the text `$\backslash\backslash$`

We use the `\footnote` command to create a footnote so just after this¹ there should be a little superscript marker referring you to the footnote, which should appear at the bottom of this page.

If we don't want the section heading to be numbered and form part of the contents, we append an asterisk to the section command, thus: `\section*{Section Heading}`

2.1 A subsection

This is a subsection, created using the `\subsection` command. And this is a reference (i.e. citation) [1] to the bibliography, created using the `\cite` command, in combination with the `\bibitem` command that is used in the bibliography.

And by using `\pageref`, I can tell you that you can find some pictures on page 12 without having to work out which page they are actually on, and by using a `\ref`, I can tell you automatically that they can be found in *section* 9. These links are referenced to the corresponding `\label` command.

These are references to Figures 2, and 3. Notice, in the source file, that there is a tilde character after the word "Figures", this is used to prevent a line-break after the word, and to prevent the space being widened by L^AT_EX's justification process.

If you keep getting messages along the lines of "Citation xxxx undefined" (or a broken-reference message), don't be fooled into thinking that it's actually

¹This is the footnote I refer to above

the `\cite`, `\ref`, or `\bibitem` command(s) that is/are broken. It's possible that there is an error elsewhere in your latex source that is preventing L^AT_EX from creating the document properly (e.g. the picture file referred to in an `\includegraphics` command is missing). If L^AT_EX can't create the document properly, the citations/references can't be completed, giving rise to this misleading error message!

To investigate the true source of these errors, check the full list of errors and warning returned by L^AT_EX, and perhaps comment-out the lines containing the `\cite`, `\pageref`, or `\ref` command(s) and run L^AT_EX again.

Again in the source file, notice that the previous `\LaTeX` command is followed by a space, yet the apostrophe is placed immediately after the L^AT_EX symbol without an intervening space. This is because L^AT_EX treats the space as merely the end of the command (and it allows me to place the apostrophe in its proper position). Where we really want a space after the L^AT_EX symbol, we end the L^AT_EX command with a backslash and a space, thus: "there will be a space after this `\LaTeX\` command: L^AT_EX but not this `\LaTeX` command: L^AT_EXas you can see".

3 Lists

- An itemised list
 - Second item
1. A numbered list
 2. Second item

Description A list of descriptions or definitions, like this one

Second item this is the description of the second item.

4 Fonts

`\emph` = *Emphasised text*

`\textit` = *Text in italics*

`\textsl` = *Slanted text*

`\texttt` = **Typewriter text (fixed-width, or monospace)**

`\textbf` = **Bold font**

`\textsc` = **SMALL CAPS**

To get raised (superscript) fonts, such as used for ordinal number (e.g. ‘st’, ‘nd’, and ‘th’ in dates), use the `\textsuperscript` command, thus:

25`th` **April 2006** = 25th April 2006

tiny

scriptsize

footnotesize

small

normalsize

large

Large

LARGE

huge

Huge

We have to turn `\normalsize` back on, or `\Huge` would remain in force for the rest of the document (or until we changed size again).

5 Text Symbols

A short dash (hyphen) -

A medium (en) dash –

A long (em) dash —

Note that dashes and hyphens are different things and should not be confused. There are rules governing the correct use of each, which can be found on the web (Wikipedia has some informative articles on each of the symbols).

Quotes must be written as two apostrophes (in each direction), e.g. “left, right”.

6 Misc Symbols

If you want to print a symbol that L^AT_EX would normally treat as part of a command, etc., you must tell L^AT_EX that it is merely to be printed literally.

For example, to print a curly brace, it must be prefixed with a backslash (`\{`). To print a backslash, the word “backslash” must be prefixed with a backslash symbol, and both must be surrounded by dollar symbols (`\backslash`).

A left curly brace: `{`

The backslash symbol: `\`

The percent symbol: `%`

The hash symbol: `#`

A dollar symbol: `$`

The underscore symbol: `_`

Tilde *accents*: `~` and `~`

A tilde *character*: `~`

A circumflex or caret, but not as an accent: `^`

The symbols for temperature °C and ° can be found in the `textcomp` package, or you can use the mathematical constructions: `^\circ` or the shorter `^\circ`

When writing instructions for a command line interface, it’s useful to be able to indicate that the return key is pressed. The conventional symbol for this can be generated with a custom L^AT_EX command. The command shown in the source code doesn’t work in the **verbatim** environment, however.

The return command is treated as just plain text. `\return`

To get around this, use the `\Verbatim` environment (from the **fancyvrb** package) instead.

The return command is treated as just plain text. `\return`

7 Maths Symbols

To produce special mathematical symbols, we have to place dollar symbols on either side. For example:

$\$-\$$ = A minus sign $-$

The less-than sign $<$

The greather-than sign $>$

For more complicated equations, we create an environment with `\begin{equation}` and `\end{equation}`. L^AT_EX automatically numbers equations.

We can use the maths dollar symbols to refer to variables such as T and I . To get the little tick (prime) mark, we use the `\prime` command thus D' . An example of an equation is given below.

$$T \times D = F \times D' \tag{1}$$

Where

$T \in \mathbb{N}$ = Number of trials

$D' \notin \mathbb{N}$ = Number of distractors

Here's another example of an equation

$$T = \frac{F \times D'}{D} \tag{2}$$

One can also build symbols using `backslashbuildrel` and `backslashover`. For example, to indicate a fast-fourier transform operation, one might use the following:

\xrightarrow{FFT}

Multi-line equations can be handled by the `\eqnarray`; append an asterisk to suppress the printing of line numbers.

$$\begin{aligned} \text{Sin}(\alpha) &= O/H \\ \text{Cos}(\alpha) &= A/H \\ \text{Tan}(\alpha) &= O/A \end{aligned}$$

8 Program Listings

To display source code text, use the `listings` package. This knows about various computer languages and can typeset the syntax appropriately.

We can specify how we want the text to appear, either in the document preamble (for global options) or with each environment's `\begin` command. Here's the result for some Pascal code (see the preamble in the source code for the display options):

```

1 procedure calcvectors(
2   s_toelmnts : integer;
3   s_wavedist, s_elmntalpha_r, s_theta_r : real;
4   var eh, ev : real );
5 var
6   element : integer;
7   dist, alpha_r, phi_r : real;
8 begin {calcvectors}
9   dist := 0;
10  alpha_r := 0;
11  Eh := 0;
12  Ev := 0;
13  {Calculate the power emitted by each element}
14  for element := 1 to s_toelmnts - 1 do
15  begin
16    {distance from the 'measuring point' to the
17    element we're considering is cumulative -
18    we start at the first element}
19    dist := dist + s_wavedist;
20
21    {phase angle of signal from this element is
22    the phase angle of the signal feeding it
23    this value is cumulative from the first
24    element}
25    alpha_r := alpha_r + s_elmntalpha_r;
26
27    {Take the invariant-result calculations
28    outside the element loop if optimisation
29    becomes necessary}
30    phi_r := 2 * pi * COS(s_theta_r) * dist;
31    Eh := Eh + COS( phi_r + alpha_r );
32    Ev := Ev + SIN( phi_r + alpha_r );
33  end;
34 end; {calcvectors}

```

Note that the default typewriter/fixed-width font in L^AT_EX does not have a bold series. Therefore we need to use a font that does, if we

want to highlight keywords in bold. **luximono** is a package of fixed-width (monospace) fonts that has an emboldened version. However, this font is available only in a T1 encoding, so we need also to load the **fontenc** package with the **[T1]** option.

We have to consider the ‘**left and right**’ quote symbols which in `texttt` and `verbatim` formats, curl in the appropriate direction. However, programming languages typically use the right quote only, which points straight down and, thus, looks wrong in L^AT_EX, for instance:

```
myString = 'hello'
```

The **upquote** package solves this problem by modifying `verbatim` mode (but not `texttt`) to show the programming-style quote symbol: for instance:

```
myString = 'hello'
```

Web addresses (URLs) can contain characters that confuse L^AT_EX. To avoid these problems, and display the URL in the conventional fixed-font, use the **url** package and the `\url` command. Have a look at the source code for this document for numerous examples.

The standard (maths mode) backslash character, which is used in this guide for displaying commands in the running text, might look ‘wrong’ in `texttt` format:

```
c:\windows\system32\
```

but here’s an alternative:

```
c:\windows\system32\
```

and another for ordinary text, using `\textbackslash`:

```
c:\windows\system32\
```

9 Pictures

Be careful when using ‘floats’, which are things like figures and tables, and are declared with commands like `\begin{figure}` and `\begin{table}`. These can cause real headaches when L^AT_EX places them where you’re not expecting. It does this when it can’t find enough space on the current page for them — it places them at the next available space. Also, between the current position and the eventual position of the

float, L^AT_EX will place the text that comes *after* the float in your L^AT_EX source file, which can be confusing for you the writer, and the reader.

To tell L^AT_EX to *try* to place the float where it appears in the source code, use the “h” (for ‘Here’) argument, as in: `\begin{figure}[h]` and if that doesn’t work, try forcing L^AT_EX to obey by using “h!” (follow the “h” with an exclamation mark).

It’s difficult to explain how the float system works, but you can find more information about it on the web (there’s a newsgroup called `comp.text.tex`, where people discuss the use of L^AT_EX), and in the online documentation. However, it is usually possible to solve the float problem by experimenting with the document’s layout (try using the `\newpage` command (which starts a new page at the exact position of the command), or moving the float’s entry in the source file.

9.1 Displaying Graphical Files

L^AT_EX knows how to show digital photographs and images from many different graphical file formats. When giving the picture’s filename, do not include the extension. The reason for this is that pdfLatex and L^AT_EX have different default graphics file formats. latex expects .eps files, whilst pdflatex expects, .pdf, .jpg, or .png files. Thus, you can use the same source with either program, and have them select the appropriate file automatically. You can convert jpeg files to eps format using the program `jpeg2ps`. Note: these days, L^AT_EX and pdflatex are one and the same (at least on my MikTeX 2.7 system); the latex command runs pdflatex in DVI output mode.

Here is a picture (but it may not appear immediately after this text, if L^AT_EX cannot fit it onto the page at this point). We can tell L^AT_EX to display the picture with a width of 90% of the available space (the `\textwidth` variable), whilst retaining the picture’s proportions, very simply:

Figure 1: A picture on its own (Small-eared mouse)

The text between the square brackets after the `\caption` command is placed in the table of figures at the start of the article, whilst the text between the curly brackets is placed next to the figure itself.

Following are two pictures side by side. Again, L^AT_EX may not be able to place the pictures directly under this text if they do not fit on the page.

Figure 2: This text appears next to the figure (Bee chafer)

Figure 3: Figure 3's Caption (Robin's pincushion)

9.2 Drawing Pictures with L^AT_EX

One can also *draw* pictures using L^AT_EX:

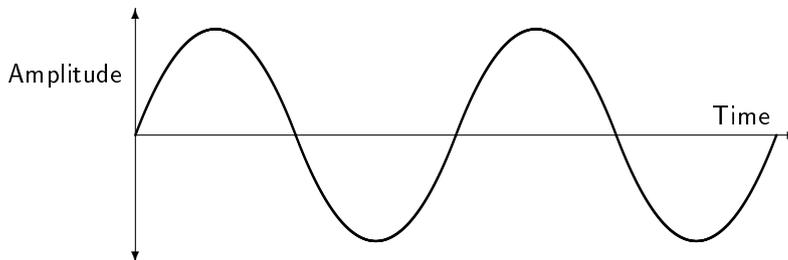


Figure 4: Sine wave drawn in the picture environment

Unfortunately, L^AT_EX's built-in drawing feature is not that powerful; the **epic** package adds some commands (such as `\drawline`) that make life easier:

$$\begin{array}{c} \alpha \backslash \\ \quad \text{H} \quad \quad \text{O} \\ \quad \quad \quad \quad \quad \quad \blacktriangledown \\ \quad \quad \quad \quad \quad \quad \text{A} \end{array}$$

Note that **epic** on its own does not always produce smooth diagonals, but also loading the **eeepic** package solves that problem. Note, however, that **eeepic** is not supported by pdflatex when generating PDF output (parts of the picture will disappear). One solution to this is to generate a DVI file and then use dvips and ps2pdf to produce the PDF.

10 Matrices

We use the **displaymath** and **array** environments to display a matrix such as the one below. **displaymath** tells L^AT_EX that the text should be treated as a mathematical object (don't break across lines, make text stand out on the page, etc.), whilst **array** tells L^AT_EX that we want the text to be structured in an array (or matrix) format.

The `\left[` and `\right]` commands produce square brackets that are drawn at the correct size to encompass the text that goes between them.

Note the use of the `\ddots` command, which draws the diagonal line of dots. The array environment is very much like the table environment, the `&` symbol separates columns, and the `\\` ends a row. The **c** characters in the parameter list cause the text to be centred in the respective column.

$$\begin{bmatrix} 1,1 & 1,2 & 1,3 & \dots & 1,n \\ 2,1 & 2,2 & 2,3 & \dots & 2,n \\ 3,1 & 3,2 & 3,3 & \dots & 3,n \\ \vdots & & & \ddots & \\ m,1 & m,2 & m,3 & \dots & m,n \end{bmatrix}$$

11 Tables

You can create tables in L^AT_EX using **table** and **tabular** environments. **tabular** is used to structure the text into rows and columns, whilst **table** is used to tell L^AT_EX the text is a table that should be suitably arranged on the page (it's a float, like a **figure**). The **[h]** argument tells L^AT_EX to try to place the table where it appears in the text, rather than where it thinks best.

Here is a small table.

Column 1	Column 2	Column 3
Item 1	this and that	and whatever else
Item 2	more of same	and yet more comments

Table 1: Table 1's Caption (Small-table example)

The row spacing can be adjusted using the `\arraystretch` command; the additional space is spread above and below the row text. Note

that the effect of the `\renewcommand` command would apply to the remainder of the document if we didn't place it inside the `center` environment.

Column 1	Column 2	Column 3
Item 1	this and that	and whatever else
Item 2	more of same	and yet more comments

Here is the same table, but with a bigger row height (created with `\extrarowheight`, which is provided by the `array` package). Note that the extra space is added *above* the row text.

Column 1	Column 2	Column 3
Item 1	this and that	and whatever else
Item 2	more of same	and yet more comments

Note that the previous table won't work if it is longer (number of rows) than one page. To have a multi-page table, we need the `longtable` format.

First, we'll reset `\extrarowheight`, and make the font size a lot smaller. We'll put the `\footnotesize` command inside the `center` environment, so it applies only to the text in that environment; otherwise, all text thereafter would be similarly reduced in size (until another sizing command changed it).

Latin name	Common name
<i>Acer pseudoplatanus</i>	Sycamore
<i>Achillea millefolium</i>	Yarrow
<i>Alliaria petiolata</i>	Garlic Mustard
<i>Alnus glutinosa</i>	Alder
<i>Artemisia vulgaris</i>	Mugwort
<i>Barbarea vulgaris</i>	Winter Cress
<i>Bellis perennis</i>	Daisy
<i>Betula pendula</i>	Silver Birch
<i>Brassica rapa</i>	Wild Turnip
<i>Bryum argenteum</i>	Moss
<i>Buddleja davidii</i>	Buddleja or Butterfly Bush
<i>Calliargon cuspidatum</i>	Moss
<i>Calystegia sepium</i>	Hedge Bindweed
<i>Cardamine pratensis</i>	Cuckoo Flower or Lady's Smock

continued on next page

Table 2: *continued*

Latin name	Common name
<i>Carex pendula</i>	Pendulous Sedge
<i>Centaurea nigra</i>	Black/Common Knapweed
<i>Centaurea scabiosa</i>	Greater Knapweed
<i>Cerastium fontanum</i>	Common Mouse-ear
<i>Chamerion angustifolium</i>	Rosebay Willowherb
<i>Cirsium arvense</i>	Creeping Thistle
<i>Cirsium palustre</i>	Marsh Thistle
<i>Convolvulus arvensis</i>	Field Bindweed
<i>Crataegus monogyna</i>	Hawthorn
<i>Crepis vesicaria</i>	Beaked Hawksbeard
<i>Cynosurus cristatus</i>	Crested Dog's-tail
<i>Cytisus scoparius</i>	Broom
<i>Dactylis glomerata</i>	Cocksfoot
<i>Daucus carota</i>	Wild Carrot
<i>Digitalis purpurea</i>	Foxglove
<i>Dipsacus fullonum</i>	Teasel
<i>Epilobium hirsutum</i>	Great Willowherb
<i>Equisetum arvense</i>	Common Horsetail
<i>Eupatorium cannabinum</i>	Hemp Agrimony
<i>Eurynchium praelongum</i>	Moss
<i>Fallopia japonica</i>	Japanese Knotweed
<i>Festuca ovina</i>	Sheep's Fescue
<i>Filipendula ulmaria</i>	Meadow-sweet
<i>Fraxinus excelsior</i>	Ash
<i>Galium aparine</i>	Common Cleavers
<i>Geranium dissectum</i>	Cut-leaved Cranesbill
<i>Geranium pratense</i>	Meadow Cranesbill
<i>Geranium pyrenaicum</i>	Hedgerow/Mountain Cranesbill
<i>Geranium robertianum</i>	Herb Robert
<i>Glechoma hederacea</i>	Gound Ivy
<i>Heracleum mantegazzianum</i>	Giant Hogweed
<i>Hirschfeldia incana</i>	Hoary Mustard
<i>Holcus lanatus</i>	Yorkshire Fog
<i>Hypericum maculatum</i>	Imperforate St John's Wort
<i>Impatiens glandulifera</i>	Himalayan balsam
<i>Impatiens parviflora</i>	Small Balsam
<i>Juncus conglomeratus</i>	Common Rush
<i>Juncus inflexus</i>	Hard Rush
<i>Lathyrus pratensis</i>	Meadow Vetchling
<i>Leucanthemum vulgare</i>	Oxeye Daisy
<i>Lotus corniculatus</i>	Common Birds-foot-trefoil
<i>Lychnis flos-cuculi</i>	Ragged Robin
<i>Lythrum salicaria</i>	Purple Loosestrife
<i>Malus domestica</i> agg.	Cultivated Apple
<i>Matricaria matricarioides</i>	Pineapple Weed
<i>Medicago lupulina</i>	Black Medic
<i>Melilotus altissima</i>	Tall Melilot
<i>Mentha spicata</i>	Spearmint
<i>Myosotis sylvatica</i>	Field Forget-me-not

continued on next page

Table 2: *continued*

Latin name	Common name
<i>Oeanthe crocata</i>	Hemlock Water Dropwort

Table 2: A long table that continues on multiple pages

We can create a coloured background for individual cells, using the `\cellcolor` command from the `colortbl` package. Note the syntax difference between `\cellcolor{red}` and `\cellcolor[gray]{0.7}`.

TL	TM	TR
BL	BM	BR

We can also apply the background to an entire row using `\rowcolor`:

TL	TM	TR
BL	BM	BR

We can change the colour of the text in an entire column:

TL	TM	TR
BL	BM	BR

And there's `\columncolor` too, which changes the background colour and goes in the column argument of the `\tabular` command:

TL	TM	TR
BL	BM	BR

12 Networks

In order to display interconnections such as in flowcharts, system diagrams, neural networks, and in graph theory, we can use the `XY` package. Below is a diagram of a very simple neural network (modified to demonstrate L^AT_EX rather than neural networks).

In the source code for the diagram below, the whole diagram is created within the argument to the `\xymatrix` command. We then create a matrix of nodes (labelled with letters, for instance). As with the tabular environment, columns are separated by ampersands and rows are separated by the double backslash. We can then draw lines and arrows between the nodes.

Caution: it's intuitive to read left-to-right, so it's easy to make the mistake of defining arrows as if they are 'directionless' lines drawn in the same position as their command, instead of being 'directionfull' arrows drawn from an origin. Thus, the arrow commands must appear in the same 'cell' as their node of origin (and not between the text of the nodes).

`e_{\mathbf{a}}` produces the e with a subscript a. The open circular nodes are produced by the `\circ` command, whilst the solid disc nodes are produced by the `\bullet` commands.

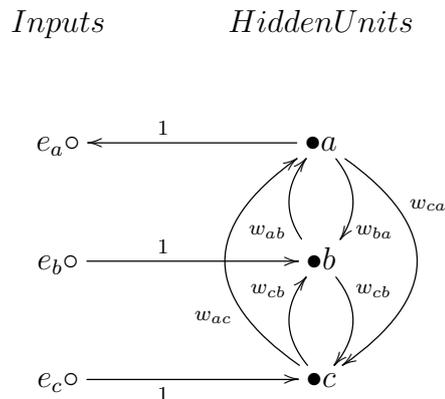
The horizontal arrows are displayed with the `\ar` command. Note that the `\ar` arrow command must appear *after* the entry for its origin (i.e. after `\mathbf{a}`), and *not* between the entries which it connects (between `e_{\mathbf{a}}` and `\mathbf{a}`).

The `\mathbf{l}`, `\mathbf{r}`, `\mathbf{u}`, and `\mathbf{d}` in square brackets refer to the direction of the arrow. Each letter moves the arrow one node (or cell) in the matrix, so `\mathbf{dr}` moves down one row and right one column, and `\mathbf{dd}` moves down two rows.

The arrow's labels (e.g. `\mathbf{1}` and `\mathbf{w}_{\mathbf{ba}}`) are given in the curly brackets at the end of the arrow command. The position of the labels is controlled with `\^` and `_` commands. Note that to place the 1 above the line going from right to left (from `a` to `e_a`) we use an underscore (`_`), but the other horizontal arrows use a caret (`\^`).

The number in round brackets between the position and the value of the arrows label refers to the label's position along the arrow. Note again that the position of the top arrow is different (0.6) to that of the other horizontal arrows (0.4). This is because the position is relative to the origin and the direction of the arrow. The number is a proportion of the arrow's length.

The at-sign after the `\ar` commands create curved arrows; the `/\^1pc/` parameter increases the curvature by one pica (12pt).



13 Miscellaneous Effects

The page headers and footers in this document are not the defaults. They were produced with the **fancyhdr** package. See the `\lhead`, `\chead`, `\rhead` commands (and the corresponding ‘foot’ commands) in the preamble at the start of this document’s source code.

The `c`, `l` and `r` refer to centre, left and right, and the `head` and `foot` are similarly obvious. The centre footer contains the page number by default, so to turn it off, use the `\cfoot` command with empty brackets.

The **LastPage** label used in our page numbering is provided by the `\lastpage` package, and indicates the number of pages of the document.

We’ve already seen centering at work, but there’s also `\flushleft`, which forces the text to be flushed straight against the left margin, which is, of course, standard behaviour in normal text.

In contrast, `\flushright` forces the text to be flushed straight against the right margin, which can look decidedly odd. The `flush*` environments seem to be most useful for showing tables and pictures side-by-side; you can push the tables/pictures together or force them apart with the appropriate use of the `flush*` commands.

`\fbox` makes it easy to surround things with a thin line.

To display text in a given colour, use the `\textcolor` command from the `color` package; **this text should appear in red**. And this text should appear in dark magenta, a colour that we defined ourselves with the `\definecolor` command (check out the preamble in this source code).

Sometimes, we want to display file names and URLs that are longer than the text width. L^AT_EX doesn't do a very good job of wrapping these, since they rarely contain spaces. The `\url` command from the `url` package provides a convenient means of formatting long strings like the location of the source code for this document: `http://psy.swan.ac.uk/staff/carter/unix/article-guide.tex`.

In order to indicate where text has changed since the previous edition, it is conventional to place a vertical line in the margin, as demonstrated here. The package `changebar` and the commands `\cbstart{}` and `\cbend}` allow us to do this.

In order to temporarily change the textwidth (i.e. the margins) without disrupting the rest of the document, we create a `changemargin` environment. The definition for our own environment is a `\newenvironment` command that is placed in the preamble. We then place the text that is to use the different margins between the usual `\begin` and `\end` pair. L^AT_EX then replaces the `begin` and `end` with the text that we provided in the `\newenvironment` command, inserting the text in between. The command takes two parameters; the amounts to adjust the left and right margins by. In this case, we subtract 30pt from the left margin and add 30pt to the right margin, thus shifting the text to the left.

14 Packages

The packages used in this document are thus:

`amsmath` gives us double-thickness mathematical-set symbols.

`array` gives us a way to control the height of the rows in tables.

`babel` tells L^AT_EX about the traditional typographical settings for the specified country (hyphenation and date format, for example). We placed the `british` setting in the option list of the `\documentclass` command so that it is set globally, in case any other packages need to know the country.

`calc` allows us to perform calculations in a parameter (such as for `\setlength`).

- changebar** provides a way of marking the margin to show where text has changed.
- color** gives us a way of displaying text in various colours.
- colortbl** provides commands to change the background colour in tables.
- datetime** provides commands for displaying time and date.
- epic, eepic** make drawing pictures easier, and make the pictures look better.
- fancybox** gives us commands for drawing the RETURN/ENTER key symbol.
- fancyhdr** gives us more control over the page header and footer.
- fancyvrb** allows us to include L^AT_EX commands in a verbatim environment.
- fontenc** gives us access to the T1 font encoding, which gives us a broader array of characters to chose from. Specifically, it was needed for the luximono fonts used in the listings example.
- graphicx** makes it easier to insert pictures into our document.
- inputenc** tells L^AT_EX how to interpret the individual bytes of the source file. Although we see the characters in our editor, all L^AT_EX can ‘see’ are the numeric values of the bytes. It doesn’t necessarily know what character each byte is supposed to mean. Some input encodings (e.g. Unicode) may contain more than one byte per character. Generally, as long as you use the document source on computers with the same regional settings, this isn’t a problem, but to make your document universal, you should explicitly state what input encoding you used.
- lastpage** gives us the number of pages in our document.
- listings** gives us a way to display computer language listings.
- luximono** gives us a bold typewriter font, which is not available with L^AT_EX’s default font set (Computer Modern).
- longtable** gives us the long (i.e. multi-page) table feature.
- luximono** provides a fixed-width font with a bold mode.
- resize** gives us a way to change the font size relative to the current size.
- setspace** gives us a way to adjust the spacing between lines.
- upquote** provides an left-quote/apostrophe that doesn’t curl, in keeping with its appearance in programming/text editors.
- url** gives us a way of displaying computer names properly.
- xy** (also called xypic) enables us to draw networks and graphs.

References

- [1] Carter, N.; *A typical Reference Title*; Publisher; 2003