

Workshop: (Slightly) Advanced L^AT_EX

Jenny Radcliffe, for Durham University Computing Society

3th February 2010

Contents

1	Introduction	2
1.1	Hints, Tips and Ways to Think	2
2	L^AT_EX – on the ITS PCs and on your home machine	3
2.1	On the ITS PCs	3
2.2	On Your Own Machine	3
2.3	Packages: What Are They and Where Do They Come From?	4
2.4	Your Own Pet Commands And Macros	4
3	BibTeX	5
3.1	Creating your Database of References	5
3.2	Inserting the BibTeX into your document	6
3.2.1	Using the natbib package	7
3.3	Compiling with BibTeX	7
4	Formatting your Documents	8
4.1	Why You Shouldn't Format Your Documents	8
4.2	The First Cludgy Hack: vspace	8
4.3	Indentation and Paragraph Spacing	9
4.4	Geometry	9
4.5	Hyperlinks	10
4.6	Fancyhdr	10
4.7	Multiple Columns	11
4.8	Fonts	11
5	Using the revtex4 package (as recommended by the Physics Department)	12
6	Presentations	12
6.1	Seminar	12
6.1.1	Example File Seminar	13
6.1.2	Example File Seminar2	15
6.2	Beamer	15
7	Posters	18
8	Typesetting Computer Code	18
9	Chemistry, Physics, Electronics – Special Diagrams	18
9.1	Electronics	19
9.2	Chemical Diagrams	20
9.2.1	Chemical Equations	20
10	Workshop Questionnaire	23

1 Introduction

This advanced workshop will cover a wide range of possible uses and functions in L^AT_EX, but in many respects it is about the right way to think about using L^AT_EX. Scattered through this workshop are hints and tips that apply not only to the sections they're in, but also to best practice in L^AT_EX generally. Where possible I have gathered them together in the next section, but it is likely that some have been missed out – it's hopefully worth your while reading all of the sections, even those not of immediate use to you, to see what's what.

It is highly unlikely that you'll have time to complete all of the sections in this worksheet during the workshop. I recommend you concentrate on the sections which are of most interest to you; all of the sections are independent and stand-alone. If you want to continue working on other sections after the workshop and run into problems, CompSoc members are often available to help on the IRC channel #compsoc (see [22]), or you can email me directly at jenny@durge.org with questions and problems.

1.1 Hints, Tips and Ways to Think

The key to becoming confident and competent with L^AT_EX is experimentation. If you wonder if something might work, try it – the worst that will happen is that you have to change it back to something you know works. If you fear making enormous changes, make a backup of your document to a file with another name in between making changes you think will cause problems – and then go wild experimenting. Version control software may make your life easier.

Google – or your search engine of choice, of course – is your friend. Most problems you have with L^AT_EX, and most problems you want to solve with it (not the same thing), have already been tackled by someone, and you will not need to reinvent the wheel.

Perhaps your first port of call for solutions to things you'd like to do in L^AT_EX should be the Comprehensive TeX Archive Network (see [3]; known as CTAN). This enormous repository has packages to make almost anything L^AT_EX related simple – most of them with excellent documentation. If you are wanting to try typesetting a musical score, or a Bridge game, or something else that seems quite unlikely, have a search on CTAN – there is very likely to be a package to help you. Most of these packages will already be installed with your L^AT_EX install, so in most cases you will not have to download and install anything, but the CTAN search will tell you what package to use, and will almost certainly have documentation for it, too. Other vital sources of information online are "The Not So Short Introduction to L^AT_EX" [16] and the Wikibook on L^AT_EX [5]. [16] may well be included with your L^AT_EX install (hiding in amongst the other documentation; run a search on "lshort" to find it), so you may not need to download it.

Compile early and often. Inevitably problems creep in – use of incorrect brackets is my bugbear – and the less you've added to a document between compilations, the less code you'll have to search through to find the source of the problem.

A statement very much of the obvious, and which you will probably ignore (but only at your peril): comment your code. Particularly if you've used a command for the first time, remember to annotate it so that you remember what it's done. The % symbol is a surprisingly powerful tool: it tells L^AT_EX to ignore everything after it up to the next new line, so that you can take notes for yourself. This has the additional advantage of making it easy for you to share code with other people, because what is perfectly obvious to the original author of a particularly pretty trick is not necessarily obvious to the next reader.

2 L^AT_EX – on the ITS PCs and on your home machine

2.1 On the ITS PCs

The NPCS machines provided by the ITS use TeXnicCenter as an Integrated Development Environment (IDE) for L^AT_EX, and this is almost certainly the best tool to use during this workshop if you are most accustomed to using Windows. You might instead want to connect to Vega to use a Linux command line.

TeXnicCenter is a free and open-source tool for Windows computers which, if you like, you can install on your own computers. It can be downloaded from [23].

To open TeXnicCenter on the NPCS machines, click Start -> Programs -> Miscellaneous -> MikTeX 2.7 -> TeXnicCenter. When opening TeXnicCenter on the Networked PC Service provided by Durham's ITS, it is necessary to go through a configuration wizard and tell it that you wish you use MiKTeX. This should not be necessary after the first use if you install TeXnicCenter on your own computer; the NPCS seems to be deleting the settings configured by the wizard each time you log off.

Two tool bars in TeXnicCenter are particularly important to bear in mind. The first (shown in figure 1) allows you to compile your L^AT_EX easily – use the menu labelled A to select your form of output (generally, LaTeX => PDF is recommended), use button C if you are compiling a single file, and use button B if you are using TeXnicCenter's "Project" functionality and have a project involving multiple files, particularly BibTeX.

The tool bar seen in figure 2 is particularly useful in determining errors in your L^AT_EX. If, after compiling, you have errors or warnings (and it is most unusual not to, particularly when using new code for the first time), this tool bar will help you to locate those errors and problems.



Figure 1: TeXnicCenter's tool bar for compiling L^AT_EX

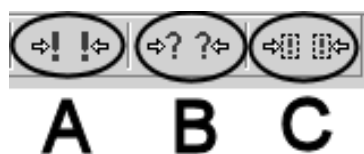


Figure 2: TeXnicCenter's tool bar for finding errors and warnings

2.2 On Your Own Machine

There are a range of IDEs (Integrated Development Environments) like TeXnicCenter for L^AT_EX on all operating systems. CompSoc members speak highly of TeXshop ([11]) for the Mac, Kile ([13]) and Winefish ([27]) under Linux, and TeXmaker ([1]) under Windows, Linux and MacOS.

Alternatively, you can use L^AT_EX more directly from the command line (using a standard terminal in Linux or OS X, or using Cygwin or andLinux in Windows). When working at the command line I also recommend use of the commands `dvips` (to convert the L^AT_EX DVI output to postscript) and `epstopdf` (to convert postscript to PDF). Sometimes, you may prefer to use `pdftex` or `pdflatex` which is a version of L^AT_EX which outputs PDF directly. The choice of L^AT_EX, `pdftex` or `pdflatex` is usually dependent on whether imported image files are in postscript, JPEG or other format.

If you expect to use L^AT_EX extensively and for a wide range of projects, you should experiment with different IDEs and possibly also the command line, to work out what method of working suits you best. There is no one correct answer for how to use L^AT_EX – everyone has their own style, and their own way of working, and by experimenting you will find your own.

2.3 Packages: What Are They and Where Do They Come From?

As you will no doubt have discovered by now, L^AT_EX involves a large number of “packages”, which are loaded in the document preamble with the `\usepackage` command. You may have noticed that these packages have a range of authors/creators, and are also not necessarily consistent in the way they do things.

This is a product of the fact that L^AT_EX is Open Source software, and many people have contributed code to the project, mostly in the form of specific packages to perform specific tasks in which they are interested. This has the advantages, among others, that no matter how strange or esoteric a thing you wish to do with L^AT_EX, someone has probably already written a package to make it easier – and that there is a large community of people who are dedicated to helping you, and each other, with getting the best use of L^AT_EX.

It does have the disadvantage that it can, for example, be very hard to tell whether a command should be in a document’s preamble or whether it should appear after the `\begin{document}` command, because different programmers have had different ideas of what should go where.

This is much less of a problem than you might think. Providing you compile your code frequently, it is usually easy to catch where inconsistencies have appeared, and then to work out what is odd. It is important to remember that trial-and-error is the best way to learn: if you’ve added some commands to do with a package, and you’re now getting errors, remember to try putting them in the preamble and in the main body, and always remember to try commenting out commands systematically to work out where the exact problem lies. Remember, too, to check for “warnings” and not merely for “errors” – the warnings can be a big hint as to what you might be trying to fix.

Many major packages, including several covered in this worksheet, recommend not starting from scratch with an empty document, but using a template or example of code created by someone else and editing it to your satisfaction. This is often a very good idea, and as a consequence this worksheet is accompanied on the CompSoc website with a range of files giving places to start.

Most L^AT_EX installs have an enormous range of packages already installed with them, ready and available for you to use. Occasionally, however, you will need to install a new package, and exactly how you do this will depend slightly on which operating system you are using.

The cheating, cludgy hack method, which may be suitable if you expect to use a package only once, is to install the files in the same directory as the `.tex` file using the package. This should work under all circumstances, but becomes inconvenient if you use a package often. At this point, [5] and [16] can instruct you on installing your package centrally so that L^AT_EX regards it as part of your full installation.

2.4 Your Own Pet Commands And Macros

As time wears on, you will find that you are typing certain commands, sometimes lengthy ones, more often than you want to. Precisely which commands this applies to will depend on what you are using L^AT_EX for, of course.

At this point, it is probably time to define your own shorthand codes for your own convenience. For example, I got very tired of having to typeset “degrees centigrade” repeatedly: the conventional way of typing it is `$^\circ C`, but this seemed slightly counter-intuitive and definitely like a long way around.

Consequently, I started to include `\newcommand{\degC}{^\circ C}` in the preamble to all my L^AT_EX documents, so that I only needed to type `\degC` where I wanted a °C symbol.

Over time, I have developed a whole collection of these macros, and copied-and-pasted them from document to document every time I created a new file. Eventually, this began to seem impractical, and so I created a new file, `mymacros.sty`, which contained all of these new macros of my own. I can then load them into a new document with `\usepackage{mymacros}` just like any other package, which looked far less cluttered and made life much easier.

Should you decide to do this, it is important to put the macro file somewhere that L^AT_EX knows where to look for it. Precisely where this should be will depend on your L^AT_EX install; see section 2.3 on page 4 for more information.

3 BibTeX

BibTeX is a database program which makes storing your external references more flexible than the `thebibliography` option which is installed by default in L^AT_EX.

At its very simplest, it involves having an external file, with the file extension `.bib`, which contains all of the information you need about your references. Within your L^AT_EX file you then reference your BibTeX database, and when compiling the L^AT_EX you also compile using the BibTeX program.

Having an external database (because although it is simply a text file, the `.bib` file is effectively a database, and while it can be edited by hand it can also be edited and maintained with any of a wide range of database programs) is extremely convenient if you are producing many documents on one subject. For example, since I have written many documents about L^AT_EX I do not simply have a separate `.bib` file for each project; I have a single `.bib` file which contains most of the references I am likely to need, and which I have built up over several years.

3.1 Creating your Database of References

While you are getting started during this workshop it may be simplest for you to edit the `.bib` files by hand in order to understand what is going on. For further use, though, do look into the many reference managers which are available – for example listed at [7].

For the time being, open (in TeXnicCenter or your editor of choice if you're not using TeXnicCenter) a file with a `.bib` file extension.

To this, you will want to add your references. In general, most documents you wish to reference are likely to fall into one of three kinds: `book`, `article` and `misc`. There are additional options for booklets, conference proceedings, theses and many others – all of [12], [18] or [7] have more details.

Each entry type has a different selection of fields associated with it, some mandatory and some optional. For example, a `book` entry must have fields for author (or editor), title, publisher and year, while it may also have fields for volume (or number), series, address, edition, month or note. Thus to cite [12], I would at minimum require in my `.bib` file an entry reading

```
@Book{dailycupcake,  
      author = {Helmut Kopka and Patrick W. Daly},  
      title = {Guide to LaTeX},  
      year = {2004},  
      publisher = {Addison-Wesley}  
}
```

I could, however, have a much longer entry (or anything in between), reading

```
@Book{dailycupcake,  
  author = {Helmut Kopka and Patrick W. Daly},  
  title = {Guide to LaTeX},  
  year = {2004},  
  publisher = {Addison-Wesley}  
  volume = {1}  
  series = {Tools and Techniques for Computer Typesetting}  
  address = {Boston, MA}  
  edition = {Fourth Edition}  
  month = {unknown}  
  note = {Undoubtedly the most useful introduction and basic  
    reference to LaTeX that I have used}  
}
```

The article entry type should be used for papers from Journals and magazines. The required fields in this case are author, title, journal and year, with optional fields for volume, number, pages, month and note. An arbitrary example of a paper cited from a journal might be

```
@article{ahu61,  
  author={Arrow, Kenneth J. and Leonid Hurwicz and Hirofumi Uzawa},  
  title={Constraint qualifications in maximization problems},  
  journal={Naval Research Logistics Quarterly},  
  volume={8},  
  year = 1961,  
  pages = {175-191}  
}
```

In theory, the misc entry type should be used very rarely. In the case of this worksheet, however, where I am trying to provide you with references in the form of websites you can look at easily rather than books you have to go all the way to the library for, I have used it rather a lot. The misc entry type has no required fields at all, but can have any or all of author, title, howpublished, month, year and note. In the case of citing websites, I use the misc entry type, with the howpublished field containing the command `\url{http://web.location.thing}`. So for example, one of the items I have used in this worksheet appears in the .bib file as

```
@misc{wikibibtex,  
  author = {Wikipedia Editors},  
  title = {Reference Managers},  
  year={2010},  
  howpublished = {http://en.wikipedia.org/wiki/Bibtex},  
}
```

Many online resources provide ready-formatted BibTeX entries for their references, so if you are referring to a lot of papers and journals, do look out for this.

3.2 Inserting the BibTeX into your document

At the very basic level, having created your BibTeX database and referred to the documents within it using the `\cite{key}` command, you need only two commands in your L^AT_EX document. At the point at which you wish the bibliography to appear, you should define the style of your bibliography with `\bibliographystyle{style}`, and then insert the bibliography itself with the command

`\bibliography{filename}`. Note that, slightly counter-intuitively, you do *not* include the “.bib” part of the filename in the bibliography declaration. (So for this worksheet, my bibliography code is:

```
\bibliographystyle{plain}
\bibliography{latex}
```

L^AT_EX provides four basic formatting styles for your bibliographies. With the `plain` style the references are listed alphabetically in the bibliography, and indexed with a simple number in square brackets. The `unsrt` style orders documents in the bibliography in the order in which they are referenced in the text, and again labels them with a number in square brackets. The `alpha` style orders references in the bibliography alphabetically, but labels them with a combination of letters from the first author’s name and the year of publication. For example, a paper by Smith from 1987 would have the label [Smi87]. Finally, the `abbrv` style orders and labels references just as the `plain` style does, but lists references in the bibliography with abbreviations in the author names, months and journal names.

These four styles are sufficient for many uses, but are less informative and flexible than most people prefer. As a consequence, many people prefer to use the `natbib` package (by putting the command `\usepackage{natbib}` in the preamble) to increase their formatting options.

3.2.1 Using the natbib package

`Natbib` increases your options in a range of ways. One particularly useful one is that alongside `\cite`, you have two further ways of citing references. The `\citep` command may be used to insert references within parentheses, including notes, and the `\citet` makes textual references available. The usages are described in table 1

Sample command	Output
as reported by <code>\citet{Jon99}</code>	as reported by Jones et al., (1999)
as reported by <code>\citet[pg. ~22]{Jon99}</code>	as reported by Jones et al., (1999, pg. 22)
results show <code>\citep{Jon99}</code> that	results show (Jones et al., 1999)
results show <code>\citep[pg. ~22]{Jon99}</code> that	results show (Jones et al., 1999, pg. 22)
show <code>\citep[e.g.][] {Jon99}</code> that	show (e.g. Jones et al., 1999)
show <code>\citep[e.g.][pg. ~22]{Jon99}</code> that	show (e.g., Jones et al., 1999, pg. 22)
in <code>\citet{Jon99}</code> we see	in Jones, Baker and Toms (1999) we see
we see <code>\citep{Jon99}</code> that	we see (Jones, Baker and Toms, 1999) that

Table 1: A selection of the types of citation made possible with the `natbib` package.

The `natbib` package also provides a selection of different styles for formatting your bibliography, including those compatible with conventional Harvard notation and others. Examine [4] for more information.

3.3 Compiling with BibTeX

BibTeX is, strictly, a separate program to L^AT_EX. It therefore has to be run separately from the main L^AT_EX program.

In TeXnicCenter, this can be done automatically, using the in-built “project” functionality. With your main L^AT_EX document open, click “Project” and select “Create with active file as main file”. In the following dialogue box, select “Use BibTeX”. Now, if you use the tool bar shown in figure 1 on page 3, use button C to compile the project as a whole, instead of just compiling a single L^AT_EX file.

Other IDEs, of course, all have their own functionality for this sort of thing. If you are using the command line, it will be necessary for you to run the following commands on your hypothetical file `project.tex` (which includes reference to a file `references.bib`):

```
% latex project
% bibtex project
% latex project
% latex project
```

The first “pass” of L^AT_EX compiles the list of references in an auxiliary file, which BibTeX then compiles an actual bibliography from. The next L^AT_EX pass inserts the bibliography and the final one makes sure that the labels in the text actually fit with the ones in the bibliography. It’s a little tedious, but that’s how it works, unfortunately. Many of us create simple make files to run this sort of thing automatically; if you’re not familiar enough with the command line to know what that means, we’re going beyond the scope of this workshop, but CompSoc can help you out – just ask.

4 Formatting your Documents

4.1 Why You Shouldn’t Format Your Documents

L^AT_EX is designed to do your typesetting for you. This means that you should only be worrying about the content, and should let the program figure out what to put where, and how to format it.

This is particularly true if you have advanced to the position of submitting papers to academic journals, which will generally have their own style files they wish you to apply to your documents.

You may feel, particularly if you are used to outputting documents in Word, that formulae look the wrong sizes, or indentations are not quite right, or similar. From a typesetter’s point of view, and from the ease-of-reading perspective, you are almost certainly wrong. Avoid fiddling with the formatting L^AT_EX provides unless you’ve got a really, really good reason. And every time you change something from the default formatting, make sure you have a really good explanation for why.

Having said all of which, L^AT_EX does provide almost infinitely flexible opportunities for formatting your documents, and when you really do want to do something unusual, you can have a great deal of fun playing with it.

It is particularly strongly advised that you do not start playing with the formatting of your document until all of the content has been added. L^AT_EX will do interesting things with float placement and so on which can be unexpectedly impacted upon by later additions to the document. Worry about the content first; fiddle with formatting later.

4.2 The First Cludgy Hack: `vspace`

One of the things you are most likely to want to do when you start fiddling with L^AT_EX’s formatting is alter the amount of vertical space it is using. In particular, I often find that the vertical space it applies between an image and its captions is just a little bit too large.

Now ideally I should fix this for an entire document using some kind of universal setting. But often I find I only wish to edit it for one or two images. In this case, and in many others, the `\vspace` command becomes incredibly useful.

Bear in mind that almost invariably you *should not* be playing with this sort of formatting. It is very bad practice. It is also very useful.

For those times when you absolutely cannot avoid wanting to alter the vertical spacing of something, use the command `\vspace{distance}` with great care. *distance* can be in any unit of measurement that L^AT_EX recognises – this includes *cm*, *mm*, *in* (inches), *pt*, *em* and others – see [12] for more details.

It is particularly useful to remember that *distance* can be a negative number – thus the space between an image and its caption can be reduced by putting, for example, `\vspace{-0.5cm}` after the image, or at the start of the caption.

There are two further useful related commands, `\vspace*{distance}` and `\vfill`.

The `\vspace*{distance}` command will apply the vertical space (or negative space) even when a new page occurs or when the command appears at the top of a new page.

The `\vfill` command will insert a space such that top and bottom of the things the command appears between are at the very top and bottom of the page. For example, using the following code you will insert the file `image.eps` at the top of the page and the file `logo.jpg` at the bottom of the page, without having to calculate exactly how much vertical space to use.

```
\includegraphics*[width=3cm]{image.eps}
\vfill
\includegraphics*[width=6cm]{logo.jpg}
```

4.3 Indentation and Paragraph Spacing

One aspect of formatting which many people want to change about L^AT_EX is the paragraph indentation and spacing. The default for most L^AT_EX document classes is to have the first paragraph of any section unindented, and subsequent paragraphs indented by a small amount. There is then no additional spacing between the paragraphs. This default can be seen in this worksheet.

Simply loading the `parskip` package (using `\usepackage{parskip}` in the preamble) will default to having no indentation on any paragraphs, and about a line of space between paragraphs. Loading the package `indentfirst` will default to indenting all paragraphs, including the first paragraph in each section.

If you would like to have more control over the indentations and line spacings, it is time for you to investigate the `\setlength` command, which may be used in your document's preamble to define a wide range of commands.

In this instance, for example, you could add the following command to your preamble.

```
\setlength{\parskip}{1ex plus 0.5ex minus 0.2ex}
```

This command tells L^AT_EX to set the spacing between paragraphs to default at "1ex" ("ex" is a distance unit which depends on the size of the font being used; it is the height of the letter x in that font), but gives L^AT_EX the flexibility to stretch this to as much as 1.5ex or shrink it to as little as 0.8ex if this will make it fit on the page better. This is called a rubber length.

The other useful parameter you may wish to modify in this instance is `\parindent`, which affects the size of the indentation of the first line of a paragraph.

The `\setlength` command, and the concept of rubber lengths, are extremely useful for many things in L^AT_EX beyond paragraph spacing, line spacing and indentation. If you do much formatting, you will come across them again and again. You can find a multitude of examples in [16], [5], [12] and, no doubt, any other introduction to L^AT_EX.

4.4 Geometry

The `geometry` package is a very sophisticated package which will allow you to make many formatting changes to your page layout. It has excellent, if verbose, instructions at [24] and a simpler explanation is also given in [12].

In essence, when you load the `geometry` package and give it some parameters to work with about how you want the page formatted (for example by telling it you want margins of a particular size), it will then calculate all of the remaining layout parameters correctly for you. If you were to set the margins in L^AT_EX without the `geometry` package, L^AT_EX would not know how to calculate the correct page width and height, and could make a mess.

This worksheet has the following command in the preamble, for example.

```
\usepackage[lmargin=2cm,rmargin=2cm,tmargin=2cm,bmargin=3cm]{geometry}
```

4.5 Hyperlinks

The package `hyperref` allows you to insert clickable links into your document, providing both links to external websites and easy access with internal references.

Once you have loaded `hyperref` (which may require the option `pdftex` if you are using `pdftex` rather than L^AT_EX— experiment to see whether your setup handles this by default or not), any use of `\ref` or `\url` will result in a clickable link. By default, links are coloured red when they are internal links (say to a section or figure), green when a link to the bibliography, magenta if linking to a file, and blue when linking to the web; this can be altered if you prefer different colours. See [16] for more information.

If you want to have a reference which is not clickable, `hyperref` provides the commands `\ref*` and `\pageref*` which act like `\ref` and `\pageref` without the `hyperref` package loaded. A word of warning: I sometimes find that the first time I run L^AT_EX after loading the `hyperref` package, I get many L^AT_EX errors which I do not understand. I then run L^AT_EX again and find the errors disappear. Before you panic about errors, try this solution!

4.6 Fancyhdr

The `fancyhdr` package allows simple customisation of header and footer files in L^AT_EX. The full, excellent documentation at [25] is supplemented, again, with simple explanations in [12].

Using `\usepackage{fancyhdr}` in the preamble enables you to use a new pagestyle, supplementing the previous options of `plain`, `empty`, `headings` and `myheadings` (see [6] or [12] for explanations of these settings if you have not seen them before) with the option `fancy`, which is considerably more modifiable.

Having set `\pagestyle{fancy}` in your document, you may now configure your headings. There are eight parameters which you may wish to alter.

By default, there is a 0.4pt wide rule at the bottom of the header section, and a similar rule is available at the top of the footer, but this is by default set to a zero width. These can be modified by a command in the preamble of the form `\renewcommand{\footrulewidth}{0.4pt}`. Obviously, this can be `headrulewidth` if it is that rule which you wish to modify, and the width of the rule can be zero or any positive number which you choose. (Once again, you can use any units which L^AT_EX recognises, but you do need to specify a unit.)

The other six commands are `\lhead`, `\lfoot`, `\rhead`, `\rfoot`, `\chead` and `\cfoot`, for left, right and center headers and footers respectively.

These commands can contain almost anything you are interested in; image files, text, and commands such as `\thepage` to give current page numbers.

For example, the headers and footers for this document are produced using the following code.

```
\setlength{\headheight}{40pt} % Gives the header enough space
\pagestyle{fancy} % This uses the fancyhdr package to insert user information.
\lfoot{\thepage} % Puts the page number in the left bottom of the page
\lhead{\includegraphics[width=2.5cm]{logo.eps}} % Puts the CompSoc logo top left
\rfoot{\url{http://compsoc.dur.ac.uk/}} % Puts the CompSoc URL bottom right
\rhead{Advanced LATEX Workshop, Feb 2010} % LATEX Workshop & date top right
\cfoot{} % should make this empty
```

By default, `\cfoot` contains the page number, so if you want to leave it empty, it is necessary to modify it explicitly as shown above.

If your document is using the `book` class, it is possible to have `fancyhdr` alternate sides, so that, for instance, page numbers are always on the “outside” corner of the pages. [25] has full details of this functionality.

4.7 Multiple Columns

It is often desirable to have multiple columns of text in a document. L^AT_EX supports having two columns natively, but the `\onecolumn` and `\twocolumn` commands which are used to switch between the two types of text always start new pages, and leave columns of unequal length.

Consequently, tidier multicolumn documents may be produced with the use of the `multicol` package. As ever, excellent documentation is available from [14] and a short introduction is given in [12].

Having loaded the `multicol` package in the usual way, you are free to start using multiple columns at any time with the simple command `\begin{multicols}{number}`. The number may be anything between two and ten. Once you have finished the text you wish to have in multiple columns, end the `multicols` environment with `\end{multicols}` and the subsequent text will revert to a standard form.

At present it is not possible to include images and so on within a `multicol` environment; see [14] for more information on how to handle figures, floats and multiple columns in conjunction. Alternatively, the package `revtex4`, discussed in section 5 on page 12, allows the use of floats within a multicolumn environment.

4.8 Fonts

This section is a gross simplification of the descriptions in [12] and [15]. Font changes are not something I have great experience with, and by and large I think that they are best left well alone. However, here is a little information, in case you wish to proceed where I fear to tread.

To apply *emphasis* to text, use the `\emph{emphasised text}` command.

It is also possible to use the font commands in table 2 directly. However, I do not recommend doing this within text – it is very bad practice. If you wish to, for example, have all cases of a foreign language in your document italicised or otherwise emphasised, you should use a macro (see section 2.4 on page 4 for more details) to create a command using these font commands, so that the style is defined in one place only, making it easier to change later if you so wish. Separating content from design is always good practice.

Changing the font size of any given piece of text within L^AT_EX is simply accomplished by using one of the declarations demonstrated in table 3.

For example, to include a section of **large text within your** sentence, use the following code:

```
For example, to include a section of \LARGE large text within your
\normalsize sentence, use the following code:
```

The one font-related change which I do make without hesitation is to load an alternative package, to use the Palatino font in place of L^AT_EX's default Computer Modern Roman. This is achieved very simply with the command `\usepackage{palatino}` in the preamble. Other font packages are also available in a standard L^AT_EX install; investigate `times`, `newcent`, `bookman`, `avant` and `helvet`, for example, to see what you might find interesting and appealing.

<code>\texttt{text}</code>	Typewriter
<code>\textit{text}</code>	<i>Italic</i>
<code>\textsl{text}</code>	<i>Slanted</i>
<code>\textsc{text}</code>	SMALL CAPS
<code>\textbf{text}</code>	Bold face

Table 2: The attribute commands which can be applied to any text.

<code>\tiny</code>	<code>\large</code>
<code>\scriptsize</code>	<code>\Large</code>
<code>\footnotesize</code>	<code>\LARGE</code>
<code>\small</code>	<code>\huge</code>
<code>\normalsize</code>	<code>\Huge</code>

Table 3: Demonstration of font sizes and the commands used to produce them

5 Using the `revtex4` package (as recommended by the Physics Department)

`revtex4` can be downloaded from CTAN at <http://tug.ctan.org/tex-archive/macros/latex/contrib/revtex/> and installed for your individual L^AT_EX install as recommended in section 2.3 on page 4.

Having installed the package, you will now be able to use the documentclass `revtex4-1` in place of `article`, `report` or similar. You may wish when loading the package to specify things like paper size and orientation, as `normal`, and also, if required, columns. For example, you might open with `\documentclass[a4paper,twocolumn]{revtex4-1}`.

Most commonly-used packages, including the `geometry` package as described in section 4.4 on page 9, work in conjunction with `revtex4`, and it is quite likely that you will want to use packages like this as `normal`.

A sample file using `revtex4` is in the Advanced Workshop guide in the Tutorials and Guides section of the CompSoc website, along with the file it outputs. It demonstrates use of `revtex4`, and provides sample code which should make using it easy.

6 Presentations

L^AT_EX provides a number of ways of producing a presentation. The two most commonly used are the Seminar and Beamer packages, and a brief overview of each is discussed here.

6.1 Seminar

The Seminar class is installed by default in most versions of L^AT_EX and can be accessed by using `\documentclass{seminar}` at the start of your L^AT_EX document. There is some information about it in [12] and the full documentation is available online [28].

The preamble of a Seminar presentation can be used to define in great detail the formatting of your presentation. A few examples of this will be shown below.

Once the preamble has been completed, you move on to creating the slides in a `slide` environment. You may wish to allow L^AT_EX to choose where to break between slides, in which case you can have all of your content between one `\begin{slide}` and `\end{slide}` pair of commands. Alternatively, and more conventionally, you may prefer to define for yourself the contents of the slides, in which case each slide needs to be encompassed in a `\begin{slide}` and `\end{slide}` pair of commands.

If you are using TeXnicCenter to produce your Seminar files, it is important, first, to be using the LaTeX => PS => PDF option (see figure 1, and to select Build -> Define Output Profiles -> Postprocessor and add the code `-t landscape` in front of the `-P pdf` `%Bm.dvi`` section in the "Arguments" box. See figure 3 for clarification.

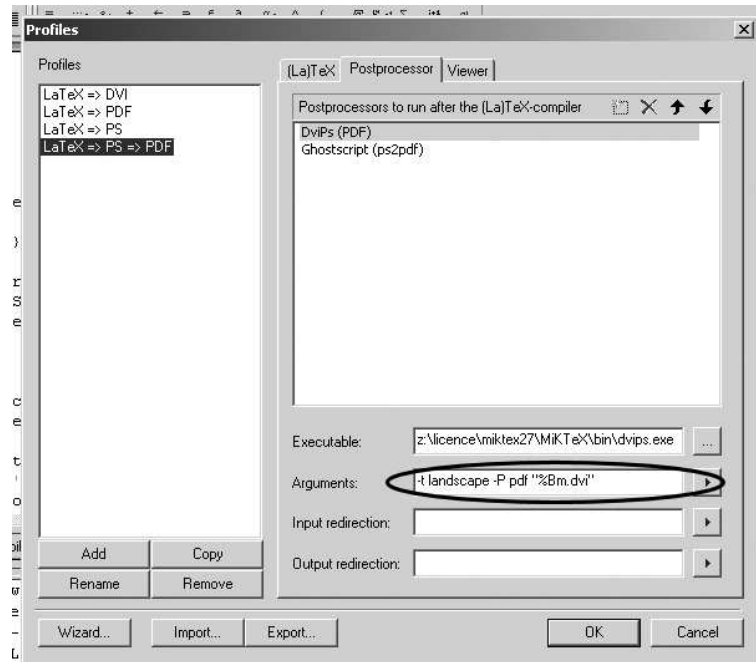


Figure 3: TeXnicCenter’s Output Profile options (accessed from the “Define Output Profiles ...” option on the “Build” menu) which must sometimes be edited if landscape output is required.

If you will be compiling the L^AT_EX from the command line, I recommend the use the following sequence of commands:

```
% latex file.tex
% dvips -t landscape file
% epstopdf file.ps
```

Remember, though, that if you are using BibTeX, you should ensure that this is compiled in full before outputting to postscript and PDF.

If you are using other IDEs, you will probably need to look for similar options in their programming, to be sure that the compiler knows that it is trying to output landscape pages. (In the unlikely event that you are trying to use portrait-aligned presentations, of course, this need not apply. In this case, I would recommend using [12] and [8] to guide you, but also feel free to contact me for help.)

There are two example files which we will now look at, along with their output, in the L^AT_EX Workshop section of the CompSoc website.

6.1.1 Example File Seminar

The commented code which outputs the file Seminar.pdf can be seen in figure 4 on page 14. This is a very simple file, and perhaps the only feature worth looking at is the use of the `\slideframe` command, which defines the format of the slide. The code, and resultant .pdf, are available in the Advanced Workshop section of the Guides and Tutorials section of the CompSoc website.

```
\slideframe[\psset{linecolor=black,fillcolor=Background,fillstyle=solid}]
{scplain}
```

This command can be broken down: `\slideframe[options]{style}`

The options are applied using the command `\psset` command, which is part of the `pstricks` package. You may use a whole range of commands, some alternatives to which you can see in the

```
\documentclass[a4,landscape]{seminar} % In theory, landscape should happen by
% default. In practice, it is as well to include it. Without the a4 option,
% the default slide size will be Letter (11x8.5inches).

%\documentclass[a4,landscape,slidesonly] % This line will, when not commented
% out, output only the slides, without any notes.

%\documentclass[a4,landscape,notesonly] % This line will, when not commented
% out, output only the notes, without any slides.

\usepackage{semcolor} % Enables Seminar Colours as used below

\usepackage{palatino} % This package uses the Palatino font family instead of the
% usual LaTeX font of Computer Modern Roman. I find this considerably more
% appealing, and note that it looks less like a scary exam-paper.

\usepackage{fancybox}

\pagestyle{empty} % Means there are no page numbers or other headers or footers

\definecolor{Background}{rgb}{1,1,0.9} % Defines the colour I use in the slides;
% could be done within the formatting commands that follow, but left in a
% separate command so I can find it to alter it more easily.

\slideframe[\psset{linecolor=black,fillcolor=Background,fillstyle=solid}]{scplain}
% Defines the format of the slide. See the main worksheet text for
% a breakdown of the commands.

\newcommand{\slidetitle}[1]{\Large{#1}} % This command defines a new
% command called ``slidetitle'' which will format the slides' titles in Large
% font. If I later wish to edit the formatting of the slide titles, I need only
% change this one command, rather than editing each slide.

\begin{document}

\begin{slide} % Opens the slide environment
\begin{center}\Large This is a slide \end{center}
\end{slide} % Closes the slide environment

This is a note % Since this is outside the slide environments, this will appear on
% a separate notes page, except when the slidesonly option is added to the
% document class, when it will be suppressed altogether.

\begin{slide} \slidetitle{Another slide}

Here is a slide with a list

\begin{itemize}
\item list item the first
\item a second list item
\item yet another list item
\end{itemize}

\end{slide}

\end{document}
```

Figure 4: The commented code producing a simple set of slides using Seminar.tex

next example. In this case, I have dictated that the border should be black and filled with a single solid colour which I have defined earlier.

The styles available depend partly on what other packages you're using. By default, the options `plain` (which gives a plain square box with no fill-colour) and `none` (which gives no border at all, irrespective of what options you have applied in the square brackets).

If you have loaded the package `semcolor`, you may also use `scplain` (which provides the rounded box we see here, with the colours as given in the options), `sdbl` (which gives a rounded box like the one given here, but with two thinner lines instead of a single line) or `scshadow` (which provides another rounded box, but with a shadow).

If you have loaded the package `fancybox`, you get a choice of four styles of border, none of which will include the colour options: `shadow` and `double` (which should be self-explanatory), `oval` and `Oval` (which give a thin oval border and a thick oval border respectively).

6.1.2 Example File Seminar2

The commented code which outputs the file `Seminar2.pdf` can be seen in figure 5 on page 16. The code, and resultant `.pdf`, are available in the Advanced Workshop section of the Guides and Tutorials section of the CompSoc website.

This somewhat more complex file demonstrates more of the options available in `seminar`. In particular, notice the use of conventional L^AT_EX features like the customised headers and footers, and the use of maths. By and large, you will find that most features you are used to using in L^AT_EX will work in `seminar` class, too.

The other potentially interesting thing to notice is the more advanced use of `\psset` to create a graded background to the slides.

```
\slideframe[\psset{fillstyle=gradient,gradmidpoint=0.5,
                    gradbegin=Dark,gradend=Light}]{scplain}
```

Notice as before that the colours used have been defined earlier; there are colour models available which will give you a wide range of colours pre-defined, and you could probably also insert the definition directly inside the `\psset` command. However, I find that creating my own definitions gives me the most ease of control over the appearance of my slides.

Note that the midpoint of the gradient can be halfway down a slide, indicated by 0.5, or at any other point between 0 and 1 (representing the top and bottom of the slides respectively).

It seems likely that you can use `\psset` to create yet more weird and wonderful slide formatting. It is not a command I have examined in detail, because I have so far been satisfied with these relatively simple commands. Go forth and explore!

6.2 Beamer

The other popular option for creating slides in L^AT_EX is the `beamer` class, which, like most packages, has online documentation [20]. Among the interesting features of `beamer` is that it includes, by default, navigation buttons at the bottom of each slide, as you can see in the example given in the workshop files.

A short example of `beamer`'s code is shown in figure 6 on page 17. The code, and resultant `.pdf`, are available in the Advanced Workshop section of the Guides and Tutorials section of the CompSoc website. With `beamer`, the formatting is almost entirely controlled by Themes, which are available in great profusion from CTAN [21]. It is to be assumed that these can be modified to produce your own chosen formatting and colouring choices, but it is advised that you start by experimenting to see what's available before you leap into editing.

Among the useful `beamer` features is the appearance of sections and subsections included in slides by the theme you have chosen. Not all themes will do this, so it is possible to have very simple

```

\documentclass{seminar}

\usepackage{semcolor}
\usepackage{slidesec} % Appears to define the slidetitle command
\input{seminar.bug} % a necessary to semcolor on some occasions
\usepackage{palatino}
\usepackage{pst-grad} % enables the gradient used in the background
\usepackage{fancyhdr} % enables personalisable headers and footers outside borders
\pagestyle{fancy} % turns on customised headers and footers

\lhead{Jenny Radcliffe \hspace*{5cm} Everything to do with \LaTeX} % creates
% headers. Note the use of \hspace*{} rather than \hfill because \hfill is
% mysteriously incompatible with something

\renewcommand\headrulewidth{0pt} % removes the standard rule from headers
\rfoot{} \cfoot{} % defines footers as empty

\definecolor{Dark}{rgb}{1,0.84,0} % defines a dark colour
\definecolor{Light}{rgb}{1,0.92,0.6} % defines a light colour
\slideframe[\psset{fillstyle=gradient,gradmidpoint=0.5,
                  gradbegin=Dark,gradend=Light}]{scplain} % defines formats as
% seen in the main text

\raggedslides[0mm] % justifies paragraph text within slides

\begin{document}

\begin{slide}[9in,6.3in] % Change the size of slides if you should be so inclined
\begin{center}\Large Everything To Do With \LaTeX

\large Jenny Radcliffe

\small \texttt{jenny@durge.org}\end{center}

\normalsize Slide Here!

\end{slide}

\begin{slide}[9in,6.3in] \slideheading{Equations}


$$\frac{\partial f}{\partial t} = \left[ \sum_{i=1}^N \sum_{j=1}^N \frac{\partial^2}{\partial x_i \partial x_j} D_{ij}^2(x_1, \dots, x_N) - \sum_{i=1}^N \frac{\partial}{\partial x_i} D_i^1(x_1, \dots, x_N) \right] f$$


\footnotesize For those wondering, this is the general form of the
Fokker-Planck equation describing the time evolution of the
probability density function of position and velocity of a particle.

\end{slide}

\end{document}

```

Figure 5: The commented code producing a simple set of slides using Seminar2.tex

slides, or more complex ones which give your audience a hint on every slide as to how far through the talk you are. All I can recommend is that you experiment!

```
\documentclass[beamer]{beamer}
\usepackage{graphics}

\usetheme{Malmoe} % Beamer has many themes available, as described in the main text

\title{An Unconvincing Presentation}
\author{Iain, but mutilated by Jenny}
\date{November 25, 2008} % This data is inserted by Beamer into places on the
    % slides dictated by the choice of theme

\begin{document}
\begin{frame} % exactly comparable to the 'slide' command in Seminar
\titlepage % This is a standard LaTeX command, useful in many types of document
\end{frame}

\section{A Section} % Sectioning and subsectioning data, again, is displayed by
    % Beamer in places appropriate to the theme
\subsection{A Subsection}
\begin{frame}{A slide Title}

Some things in a slide

\end{frame}

\subsection{Another Subsection}

\begin{frame}{Another slide}

Slide contents

\end{frame}

\section{A second section}
\subsection{A further subsection}
\begin{frame}{Yet another slide}

Some information

\end{frame}

\section{Questions}
\begin{frame}{Questions}

Do you have any, or can we go to the pub?

\end{frame}
\end{document}
```

Figure 6: The commented code producing a simple set of slides using Beamer.tex

7 Posters

A very useful L^AT_EX package, `a0poster`, makes the creation of large posters including all manner of data extremely straightforward. The online documentation, [10], is as usual really quite helpful. A sample of code producing an A0 poster is in the Advanced Workshop section in the Tutorials and Guides section of the CompSoc website, which also contains the resulting .pdf.

Some confusion may arise from this sample file as to which `background.sty` package you are looking for: in this case, you are looking for the one that is part of the `ppower4` package on [3], specifically at <http://tug.ctan.org/cgi-bin/ctanPackageInformation.py?id=ppower4>.

8 Typesetting Computer Code

There are three main ways of typesetting a string of computer code in L^AT_EX. To include a small snippet in a main paragraph, use the command `+\verb+x+`, where the `x` is the code you want to input and the `+` characters may be replaced with any character which is not included in the code you are displaying. For example, the clause giving Jenny's email address in the introduction has been typeset with the following code:

```
you can email Jenny directly at \verb+jenny@durge.org+ with questions
```

To typeset a longer piece of code, as a separate section, you should use the `verbatim` environment by using the `\begin{verbatim}` and `\end{verbatim}` commands. For example, the code given below has been simply surrounded by `\begin{verbatim}` and `\end{verbatim}` tags.

```
\includegraphics*[width=3cm]{image.eps}  
\vfill  
\includegraphics*[width=6cm]{logo.jpg}
```

Sometimes, it may be simpler to input an entire file rather than copying and pasting it into your document. For this, you can use the `fancyvrb` package. This package also allows you to do an extraordinary amount of customisation of your `verbatim` text, though this is far beyond the scope of this workshop. The documentation is excellent, and if you are interested in this sort of thing, I strongly recommend looking at [26].

To simply input a file full of code, add `\usepackage{fancyvrb}` to the preamble of your document and then put `\VerbatimInput{filename}` at the point where you wish to have your code output in your document.

Remember that these `verbatim` environments are a time when L^AT_EX will pay attention to the spaces and newlines in your code: text is, quite literally, produced `verbatim`.

9 Chemistry, Physics, Electronics – Special Diagrams

L^AT_EX has packages and extensions which make it possible to produce all manner of diagrams: molecular, electronic, Feynman

That doesn't mean you should. In fact, almost certainly, you shouldn't. No, *really*. It's not the best tool for the job. You can crack a pistachio nut with a sledgehammer, too, but it's not the best way to do it.

Having said which, if you really want to – and I applaud your courage and determination if you do – you should start with [9] and dive right on in.

It should be noted that many of the more sensible programs which you can use to create scientific diagrams can output these diagrams in `eepic` format. This is an image format that makes use of L^AT_EX's inbuilt line drawing macros. Producing such diagrams by hand is impractical, but a drawing

program can of course output these macros easily. This output can then be input into L^AT_EX using commands like `\input` or even copy and paste. The advantage of this is that some minimal editing can then be done after the event, rather than having to generate a whole new image, and that L^AT_EX commands can be included in, for example, axis labels, to improve formatting.

Among the examples of programs which are often used to produce this sort of output are the drawing program `xfig` and the graphing program `gnuplot`.

Just to prove that such things are an option, if you really really want to, let's have a quick look at some of the packages which are available to you.

9.1 Electronics

CompSoc in general recommends the use of `Xfig` for electronic, optical and circuit diagrams, though there are many other programs out there which can be used.

L^AT_EX does, however, have the `circ` package for those who prefer to do their diagrams this way. There is good documentation available online [19] and in [9].

An example, taken from the online documentation, is shown in figure 7 on page 20, deriving from the following code.

```
\begin{circuit}{0}
  \nl\Laser1 {} r           % laser
  \oa 2 r                   % optical axis
  \Polar1 {} r             % polarizer
  \oa 2 r
  \nl\OM1 P1 {} {} {} {} h % optical modulator
  \atpin OM1P3
  \oa 4 r
  \cc\BLens1 {} r          % 1st lens
  \oa 2 r
  \nl\Pinhole1 {} r       % pinhole
  \oa 7 r
  \cc\SLens2 {} r         % 2nd lens
  \oa 1 r
  \Polar2 {} r            % analyzer
  \oa 3 r
  \oa 3 r
  \nl\BSplit1 {} + d      % beam splitter
  \atpin BSplit1+        % continue drawing right
  \oa 2 r
  \nl\Cam1 {camera} . r   % camera
  \atpin BLens1: \shift 0 2 \P3 % mark middle of 1st lens
  \atpin Pinhole1: \shift 0 2 \P4 % mark middle of pinhole
  \atpin SLens2: \shift 0 2 \P5 % mark middle of 2nd lens
  \Dtext{\small $f_1$} from P3 to P4 % f_1
  \Dtext{\small $f_2$} from P4 to P5 % f_2
  \atpin BSplit1.        % second part of beam down
  \oa 5 d
  \Mirror1 {} * R         % mirror
  \oa 2 r
  \nl\ScrL1 {screen} h    % white screen
\end{circuit}
```

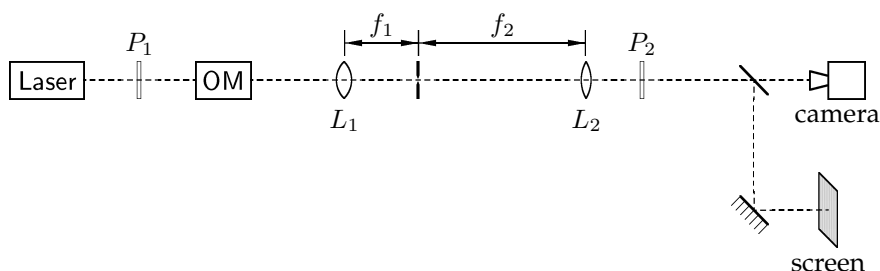
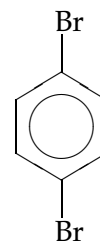
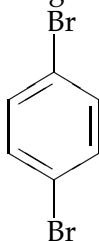
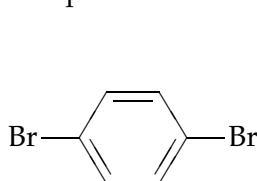


Figure 7: An Optical diagram created with the circ package

9.2 Chemical Diagrams

Using the `carom` package, I have been able to produce the following diagrams using the code you see next to them. I have no idea what the codes mean; I've simply copied it all from [9]. But you can see that quite decent diagrams can be produced using very concise text commands. However, I believe CompSoc would recommend installing and using the program ChemDraw [2] instead.



```
\bzdrh{1==Br;4==Br} \bzdrv{1==Br;4==Br} \bzrdv[c]{1==Br;4==Br}
```

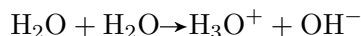
9.2.1 Chemical Equations

As an aside, those of you interested in Chemistry may have noticed that the equations L^AT_EX output are not perfect from your point of view. L^AT_EX defaults to italicised letters in equations, whereas chemical equations are conventionally set with an upright font. In addition, the default arrows supplied by L^AT_EX are often felt to be not quite right for “proper” chemical equations. Have no fear, solutions exist! The de-italicisation of an equation can be brought about by simply encasing the code in a `\mathrm{}` command within the equation or maths environment. Thus for example the code `\mathrm{Fe_2^{2+}Cr_2}_4` produces the output $\text{Fe}_2^{2+}\text{Cr}_2\text{O}_4$.

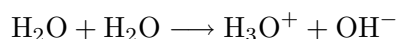
For more suitable arrows, you will need to load the package `chemarrow`. Details and documentation on this, as ever, are available online [17].

An example of code using the most basic arrow that `chemarrow` makes available is

```
$$\mathrm{H_2O + H_2O \chemarrow H_3O^+ + OH^-}$$
```



Contrast this with the same equation using the standard L^AT_EX `\longrightarrow` to decide whether the `chemarrow` package is going to be useful to you. Note also, though, that the arrows in the `chemarrow` package are configurable as to length, etc.



References

- [1] Pascal Brachet. Texmaker downloads. <http://www.xmlmath.net/texmaker/>, 2010.

- [2] Cambridgesoft. Chemdraw. <http://www.cambridgesoft.com/software/ChemDraw/>, 2009.
- [3] a Plethora of Contributors. The comprehensive tex archive network. <http://tug.ctan.org>, 2009.
- [4] Patrick W. Daly. Natural sciences citations and references. <http://tug.ctan.org/tex-archive/macros/latex/contrib/natbib/natbib.pdf>, 2009.
- [5] Wikibooks Editors. Latex. <http://en.wikibooks.org/wiki/LaTeX>, 2009.
- [6] Wikibooks Editors. Page layout. http://en.wikibooks.org/wiki/LaTeX/Page_Layout, 2009.
- [7] Wikipedia Editors. Reference managers. <http://en.wikipedia.org/wiki/Bibtex>, 2010.
- [8] Denis Girou. Seminar faq. <http://www.tug.org/applications/Seminar/Seminar-FAQ.html>, 2010.
- [9] Michel Goossens, Sebastian Rahtz, and Frank Mittelbach. *The LaTeX Graphics Companion*. Addison Wesley, 1997.
- [10] Gerlinde Kettl and Matthias Weiser. a0poster. http://tug.ctan.org/tex-archive/macros/latex/contrib/a0poster/a0_eng.pdf, 2010.
- [11] Richard Koch, Max Horn, and Dirk Olmes. Texshop downloads. <http://www.uoregon.edu/~koch/texshop/>, 2010.
- [12] Helmut Kopka and Patrick W. Daly. *Guide to LaTeX*. Addison Wesley, 2004.
- [13] Michel Ludwig. Kile downloads. <http://kile.sourceforge.net/>, 2010.
- [14] Frank Mittelbach. An environment for multicolumn output. <http://tug.ctan.org/tex-archive/macros/latex/required/tools/multicol.pdf>, 2006.
- [15] Frank Mittelbach and Michel Goossens. *The LaTeX Companion*. Addison Wesley, 2004.
- [16] Tobias Oetiker, Hubert Partl, Irene Hyna, and Elisabeth Schlegl. The not so short introduction to L^AT_EX. <http://tobi.oetiker.ch/lshort/lshort.pdf>, 2006.
- [17] Thomas Schroeder. Chemarrow package. <http://tug.ctan.org/tex-archive/macros/latex/contrib/chemarrow/>, 2007.
- [18] Michael Shell and David Hoadley. Bibtex tips and faq. <http://tug.ctan.org/tex-archive/biblio/bibtex/contrib/doc/btxFAQ.pdf>, 2007.
- [19] Sebastian Tannert and Andreas Tille. Package circ. <http://tug.ctan.org/tex-archive/macros/generic/diagrams/circ/circ.pdf>, 2003.
- [20] Till Tantau. The beamer class. <http://tug.ctan.org/tex-archive/macros/latex/contrib/beamer/doc/beameruserguide.pdf>, 2010.
- [21] Till Tantau and Others. A selection of beamer themes. <http://tug.ctan.org/tex-archive/macros/latex/contrib/beamer/>, 2010.
- [22] Sandy Taylor. Connecting to irc. <http://compsoc.dur.ac.uk/content/tutorials/connecting-irc>, 2008.
- [23] TeXnicCenter.org. Texniccenter downloads. <http://www.texniccenter.org/>, 2010.

- [24] Hideo Umeki. The geometry package. <http://tug.ctan.org/tex-archive/macros/latex/contrib/geometry/geometry.pdf>, 2008.
- [25] Piet van Oostrum. Page layout in latex. <http://tug.ctan.org/tex-archive/macros/latex/contrib/fancyhdr/fancyhdr.pdf>, 2004.
- [26] Timothy van Zandt. The fancyvrb package: Fancy verbatims in latex. <http://tug.ctan.org/tex-archive/macros/latex/contrib/fancyvrb/fancyvrb.pdf>, 2008.
- [27] Winefish. Winefish downloads. <http://winefish.berlios.de/>, 2010.
- [28] Timothy Van Zandt. seminar.sty: A latex style for slides and notes - a user's guide. <http://tug.ctan.org/tex-archive/macros/latex/contrib/seminar/doc/sem-user.pdf>, 1993.

Acknowledgements

This worksheet would have been considerably more dull and less educated without not only the collective wisdom of CompSoc's IRC channel, but in particular the assistance of Andrew Thurman, Sandy Taylor, Dan Kolb, Edwin Brady, Iain Georgeson, Simon Armstrong and Jon Kennedy. In particular the section on `revtex` would be empty without sample code from Sandy Taylor. The workshop also owes a great deal to Wikipedia. Thank you all.

10 Workshop Questionnaire

We would very much appreciate it if you would complete the following questionnaire about this workshop. Please feel free to do so anonymously, but if we would love to be able to contact you further if you have interesting suggestions. If you don't mind us doing so, please put your email address in the "any further comments" section.

1. How did you hear about this workshop, and from whom? (If from an academic department, which one?)

2. Why did you decide to come to this workshop? (If it is because someone has recommended L^AT_EX to you, please tell us who.)

3. How useful has this workshop been to you?

- Extremely useful
- Somewhat useful
- Not really useful
- Utterly hopeless

4. Was there anything else you would have liked this workshop to cover?

5. Would you be interested in attending workshops run by CompSoc on other subjects? If so, what subjects? (Please include your email address if you'd like us to notify you of specific workshops.)

- Yes
- Maybe
- No

6. This event has cost CompSoc some money in printing and was provided on a voluntary basis by a professional, and so we have asked for a small donation to cover printing costs. In the event that we formalised this by charging an actual fee for future workshops – rather than asking for voluntary donations – would you be inclined to recommend them to your friends or departments, and how much, if any, would you yourself have been willing to pay?

- Yes
- Maybe
- No

Comments?

7. Do you have any more comments, queries, suggestions, criticisms?
