LaTeX: Personal reference guide*

Alexander Shorin
University of Auckland
PB 92019, Auckland
New Zealand

November 26, 2010

Abstract

This document contains a collection of solutions to non-trivial problems encountered since I started using LaTeX. The general rule for adding material to this document is *if it takes longer than 10 minutes to figure out a solution it must be added here.*

The source code and the project tree for this production can be found at http://www.cs.auckland.ac.nz/~al/latex/latexRef/.

To do list

Enable if required

*Document version 1.2

Contents

List of Activities

List of Tables

1 Good Practice Rules

1.1 All Projects

* Use two bibtex files (4.13)
* Keep project specific config files in a separate folder (4.11)
* Use type `Generic` for electronic and other obscure references (6.2)
* Maintain field `key` in all `bibtex` entries to allow sorting references for type `Generic` (6.2)
* Be careful with conference dates and conference publisher/location (6.4)
* Use verbatim text very carefully (8.6)
* Use a shortcut command and env for tabbing words (8.9)
* Trigger Acrobat bookmarks pane as required (12.2)
* Use T1 font in all docs (8.2)
* If using `hyperref` and `url` together configure `hyperref` properly (13.3)
* Do not make basic stylistic mistakes (6.4, 11.3)
* Do not use `\VerbatimFootnotes` from the `footmisc` package (7.4)
* Declare `footmisc` before `fancyhdr` and `setspace` (7.4)
* Declare `float` before `hyperref` (14.10)
* Declare `footmisc` before `hyperref` (14.13)
* Make sure `scrhack`, `footmisc`, and `hyperref` are imported in order (14.12), (14.13)
* Never put `\caption{}` after `\label{}` in figures and tables (9.6)
* Do not use text wrapping around pictures (9.15)
* Do not use `jpeg` for graphs and other fine illustrations (10.1), (10.10)
* Do not use anything other than INKSCAPE for drawings (10.10)
* Never use `eqnarray` (11.8)
* Do not put maths (etc) directly to section headings (8.15)
* Careful using line breaks (14.2)
* Do not put references to footnotes w/o configuring `hyperref` first (14.8)
* Use `ifpdf` if required while working with both `latex` and `pdflatex` (5.2)
* Do add option `hyperfootnotes=false` if using `hyperref`, (13.5)
* Do use `cite` package [6, p.693], to sort numbers in multi-citations
* Do export `\usepackage{array}` if creating tables with `tabular`
* Consider `mdwtab` if requiring more flexible tables
* Consider `mdwmath` if requiring more flexible math
* Consider `eqparbox` for creating automatically sized boxes of the same size
* Use `\hyphenation{op-tical net-works}` for global declarations. Or `op\-tical` for local
* Check if `\eqref` was used incorrectly for Sections etc
* Check if there are any unprotected references in captions (13.1)
* Import `scrhack` if using `koma` (14.11)
* Name `tex` files with only latin characters (4.5)
* Check if embedding of fonts is required (8.1) (to save space)
* If using ToC etc, strongly consider `hypdvips` (7.9)
* Make sure that references are titled correctly: Bibliography or References
* If using `bbm` fonts (package), check `dvi` profile to guarantee proper font rendering (8.19)
* Use `\usepackage[english]{babel}` if hyphenation does not work for some reason (14.14)

1.2 IEEE Productions

- Configure `natbib` accordingly if using it with non-natbib `IEEEtran` (6.6), (6.7)
- For conferences use `\documentclass[conference]{IEEEtran}`
- Use `cmex10` as `\usepackage[cmex10]{amsmath}` to ensure type I fonts for all sizes
- Use `algorithmic` for any algorithms (9.4)
- Amend `subfig` to `\usepackage[caption=false,font=footnotesize]{subfig}`
- Do not use `cuted`, `midfloat` or `pslatex` packages
- Do not use `\thanks{}`, use the last section for acknowledgements if required
- Do not put floats in the very first column AND page alike
- Do use only top floats
- Do not use appendix in conferences
- Do use `\IEEEtriggeratref{8}` to balance columns on the last page
- Do use `IEEEtran`
- Test `pdf` production flow: http:\www.michaelshell.org/tex/testflow/
- Check if embedding of all fonts is required (8.1)

1.3 LNCS Productions

- Customise the preamble to prevent a `\vec` warning (14.3)
- Configure `natbib` accordingly if using it with `lncs` (6.7)
- Use `cmex10` as `\usepackage[cmex10]{amsmath}` to ensure type I fonts for all sizes
- Amend `subfig` to `\usepackage[caption=false,font=footnotesize]{subfig}`
- Check if embedding of all fonts is required (8.1)

1.4 Environment Maintenance

- Use SHORTCUT keys for inserting repetitive items
- Export the `BibTex mine` style sheet from BIBLIOSCAPE periodically
- Test `pdf` production flow: http:\www.michaelshell.org/tex/testflow/

2 LaTeX Resources

2.1 Online resources

1. Home repo refs files for selection of title, section, chapter pages etc
2. The TeX Catalogue Online (packages)
   http://www.ctan.org/tex-archive/help/Catalogue/bytopic.html
3. Wikibooks (checked and excellent on figures and floats)
   http://en.wikibooks.org/wiki/LaTeX/Floats,_Figures_and_Captions
4. Official FAQ of LaTeX (first source of ref for harder problems as they have searching)
   http://www.tex.ac.uk/cgi-bin/texfaq2html

5. Andrew Roberts' webpage (tips, tricks and tutorials for the most common problems with tables, images, contents, pdfs, maths etc.)
   http://www.andy-roberts.net/misc/latex/index.html

6. UTexas LaTeXResources page (similar to above)
   http://www.utexas.edu/ogs/etd/LaTeX/

7. Fauskes.net (similar to above but limited to: numbering, margins, graphics, pdf, BibTex only.
   http://www.fauskes.net/nb/latextips/

8. Usenet Group for LaTeX (if everything else fails)
   http://groups.google.com/group/comp.text.tex/topics?hl=en

2.2 Textbooks and reference guides

1. The LaTeXCompanion: Tools and Techniques for Computer Typesetting [6]

2. Math mode - everything about maths [5]
   http://www.ctan.org/tex-archive/info/math/voss/mathmode/Mathmode.pdf

3. Math Guide by American Mathematical Society [4]
   ftp://ftp.ams.org/pub/tex/doc/amsmath/short-math-guide.pdf

4. Comprehensive symbol list [3]
   http://www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-a4.pdf

5. The LaTeXgraphics companion: illustrating documents with TeX and PostScript [2]

3 Applications/System Overview

3.1 Current Installation Breakdown

The current document is based on using a LaTeX2 installation:

Function	Product
compiler	MikTeX
editor	TeXnicCenter, Yap
biblio	Biblioscape, MS Word, ruby, `rtfToBib`
graphics interoperability	netpbm
graphics scientific	Matlab
graphics vector	Inkscape
graphics powerpoint	IguanaTex

3.2 Alternative Editors

TeXnicCenter	Contra: does not know Unicode
Texlipse	Pro: auto-completion, auto-correction, CVS, teams, and other Eclipse standard features
LEd	Pro: code on the left, resulting PDF on the right with fully working backreferencing
BaKoMa TeX	Contra: commercial, Win. Pro: true real-time preview.
Scientific Workplace	Contra: commercial, WYSIWYG, bad code generation
WinEdt	Contra: commercial. Similar to TXC
LyX	Contra: WYSIWYG, bad code and graphics generation
LyTeX	Same as LyX
TeXWorks	Pro: cross-platform, backreferencing, similar to TXC
Texmaker	Contra: editor font Courier. Similar to TXC

3.3 Spell-checking

If the build-in TeXnicCenter feature is not good enough, try `aspell`, a free spell-checker for `tex` files[1].

4 File System

4.1 System setup

The following directory structure to be supported in the local repository. Note that with a network drive I maintain two repos: local and network. All of the items listed in this subsection are stored on the network. But of course this can equally be done locally:

- `\repo\latex\utils` - location of all custom created utility applications, scripts, including:
  - All project batch files (see Section 4.12), such as located in individual subdirectories `rtfToBib`, `graphicsMerge`, or `backup`, but not exclusively limited to those (see Section 4.8).
  - Shortcuts for those batch files to be called inside individual projects in subdirectory `shortcuts`.
  - Package `netpbm` package for inter-format conversions of graphics files (see Section 10.5).
  - The file called `locations.txt` in `\repo\latex\utils`.
  - Runtime environment is required for `ruby` to run bibliography generation scripts described in Section 4.8.
- `\repo\latex-aux` to contain all files which are obsolete, backed up, or temporarily exluded from any other location in the system.
- `\repo\latex\biblioscape` for output styles, reference types, and individual ref type files for Biblioscape.

[1] http://ubuntuforums.org/showthread.php?t=1259557

File `locations.txt` (see above) must be copied to `C:\WINDOWS\system32` for all batch files being able to find the location the scripts. The `locations.txt` file contains all possible locations of `\repo\latex\utils` under different computers such that the scripts could run on multiple machines with the directory trees above `\repo` setup arbitrarily. The file contains the list of absolute paths ***not*** terminated with a backslash and must be saved in the UNIX format to be parsed correctly. No scripts will run if this requirement is violated.

4.2 TeXnicCenter setup

Customise dictionaries Add the dictionary file `\repo\texniccenter\dictionaries\`, make a copy before adding or overwriting it as it will be impossible to recover the file if anything goes wrong. Downlaod a NZ one if required.

Configure profiles correctly See Sections 4.11 and 7.14.

Configure `dvips` correctly See Section 8.19.

4.3 MikTeX setup

In addition to the latex files coming with the standard MikTex distribution, there is a user directory which contains all other configuration files not coming as standard with the distribution. The top directory of all these files is the same and will be called 'root' here. My convention to call the root `\latex` and store it in the repo section on my local space (rather than server) for speeding up purposes. Generally everything is already provided in the default MikTeX folder, however, sometimes I want to have specific versions of files available to my system which will not be overwritten or updated, or which I want to edit freely (such as `bst` or `sty` files. In this case, the following is advisable:

- The location of personalised root in `\repo\latex` has to be specified in MikTeX and being added *above* the default MikTeX root.
- Bibliography style formatting files go to `\repo\latex\bibtex\bst\`, read Section 4.13.
- Any other custom packages and custom fonts need to go to `\repo\latex\tex\latex\` and `\repo\latex\fonts\` respectively.
- In MikTex General tab press Refresh Now and Update now buttons.

4.4 Inkscape setup

The installation can be very painful since it relies on two other packages. The following steps should be sufficient. Download and install in `\repo\applications\graphicsmagick` the application called GraphicsMagick. An older version worked `1.2.5-Q16-windows-dll.exe` available from `ftp://ftp.graphicsmagick.org/pub/GraphicsMagick/1.2/windows/`. Install to the same location an older version `pstoeditsetup345.exe` of `pstoedit` from `http://sourceforge.net/projects/pstoedit/files/pstoedit/`. When installing select an option of configuring it to work with GraphicsMagick. Make sure that both paths are added to the windows `path` variable. Before proceeding with Inkscape check if both are installed correctly by calling `pstoedit.exe` from `cmd` and check if any errors are thrown.

You might also need to download `dll` files (msvcr70.dll and msvcp70.dll) and put them in the `pstoedit` folder from a safe location such as http://www.helga-glunz.homepage.t-online.de/pstoedit/msvc-runtimelibs.zip.

After these steps INKSCAPE can be installed. When running it click on `Effect->Render`, you should be able to see item `LaTeX formula`. If not, `pstoedit` is either not installed correctly or it has not been added to `path`. If the menu is there, click it and apply the default formula which it already comes with. If it gives any error about a `dll` file it means that previous steps of installing GRAPHICSMAGICK were not successful. If it does not give any errors and does not display anything after pressing `Apply` - again you will need to try reinstalling everything again.

Again the previous steps could work on later releases, but this did not work for me.

Once `Effect->Render->LaTeX formula` works, the final piece is to be installed `textext`, version 0.4.3 is fine and is available from http://pav.iki.fi/software/textext/. Install in the same directory as INKSCAPE, and then use with `Effect->Render->Tex Text`.

Applications mentioned are saved to `D:\applications\Inkscape-installation`. I have not run md5 check on them.

Other details on INKSCAPE can be found in Sections 10.11, 4.12, and 4.8.

4.5 YAP setup

If things go wrong with YAP and its inverse search function does not get configured properly automatically, it can be done manually:

- Open Yap via Start/All Programs/MikTeX/DVI Viewer
- Once Yap starts, select the "View/Options" menu
- Click on the "Inverse Search" tab and type into the "Command line" field
  `C:\Program Files\TeXnicCenter\TEXCNTR /ddecmd "[goto('%f', '%l')]"`

Also, bear in mind that YAP can fail to back-forward link if certain characters in filenames of tex files are used (such as, e.g., hyphens). So it is best to name project source files with plain latin characters only.

4.6 MATLAB Setup

The following files to be copied to the searchable root: `laprint` and `uploadEpsToImgSrc`. Read Section 10.8 if generating graphics with matlab and `laprint`.

4.7 BIBLIOSCAPE Setup

Enable Importing Google Scholar Citations to Biblioscape Make sure that your Windows file association opens `enw` files with BIBLIOSCAPE, then just click on the Google Scholar link "Import to Endnote", select open and the ref should be automatically loaded to a currently open Biblioscape folder. Biblioscape may crash sometimes though. If there's a problem with reading the file, try to playing around with Import filters or styles, the appropriate one must be selected (i.e. EndNote-Refer). Alternatively, if everything else fails with automatic importing try File/Import/Tagged/Import-Filter and selecting 'EndNote-Refer'.

To setup Google properly in Google Scholar Preferences: select 'Endnote' in Bibliography Manager/Show links to import citations into.

Customising workplace To enable an automatic selection of the right style when exporting citations (rather then going through a long list every time), find on a toolbar a drop-down menu list called `Output style` and change to one required, e.g. `Bibtex mine` or `IEEE journals`.

Also you can add a particular output style to the list of favourites via Tools -> Styles -> Output Styles and then double clicking on styles.

Import customised output styles from repo Currently there are two `Bibtex mine` and `IEEE journals` which are discussed in Sections 6.2 and 6.3. Use Tools -> Styles -> Output Styles and select files from `\repo\latex\biblioscape\`. This step also requires copying over individual ref files (about a dozen) to overwrite ref files in windows' `Program Files` which were installed when installing Biblioscape.

User define fields in BIBLIOSCAPE entry form It can be changed by going to Tools -> Options -> Reference Types. If installing first off it's a good idea to import an already configured the mapping from `\repo\latex\biblioscape\`. Otherwise the form can be edited directly in the screen. Do not forget to export it for future references.

4.8 Scripts

Also a number of helper scripts may be required in any given LaTeXproject and their location is outlined in Section 4.1. A copy of all scripts used can be found in http://www.cs.auckland.ac.nz/~al/latex_ref/.

- `bckp_*.bat` scripts run a `projectBackup.bat` script with one of the four possible arguments. It basically zips the project files into a jar file in a project's `\bckp` directory.
  - `bckp_All.bat` - all files excluding the files in the `\bckp` folder
  - `bckp_Dirs_Listed.bat` - only the root + folders listed in the `dirsToBckp.txt` file in the root of the project. Note that `dirsToBckp.txt` must be saved in the UNIX format for it to run properly
  - `bckp_Source.bat` - only the root, `\src` and `\bib` folders
  - `bckp_Repo.bat` - the entire repo subtree of latex packages and configuration files
- `rtfToBIB.bat` converts the `bib.rtf` file exported by BIBLIOSCAPE to the `bib.bib` format required by LaTeX. Requires MS WORD and the `Ruby` runtime environment. See `batchmacro.rb` and `ToTxt.bas` below.
- `batchmacro.rb` is a ruby script called by `rtfToBib.bat` to bootstrap `ToTxt.bas` without requiring the user to explicitly open MS Word.
- `ToTxt.bas` is a MS Word VB macors which converts the rtf bibliography file exported from BIBLIOSCAPE to its plain text `.bib` clone. MS WORD must have macros enabled and also have this macros saved in the `normal.dot`. It is called by `batchmacro.rb`.
- `graphicMerge.bat` converts all graphics files as described in Sections 4.9 and 10.1. The script requires `netpbm`, a package which contains utilities for conversion between various graphic files and `eps`, see Section 4.1 for its configuration details. More on

this is in Section 10.1. The script also requires `epstopdf.exe`, a utility for converting `eps` files to the `pdf` format, which is supplied with a default installation of MikTeX. More on this is in Section 10.1. The script requires one of the two parameters: `true` or `false`. `true` - all files in `img-src` will be deleted and all images will be converted from the scratch from `eps`, `bitmap` and `3rdparty` directories. Slow for large projects but guarantees data consistency.

- `false` - only files not in `img-src` will be converted. Much faster, but some of the changed files, for example, might not be updated. Also files from `3rdparty` will not be copied at all. Careful with this.
- `uploadMatlabLaprint.bat` is envoked by the MATLAB script `uploadEpsToImgSrc` (Section 4.9). Kept together with `graphicsMerge`.
- `buildGlossaries.bat` see Section 7.14.
- `_SvgToEps.bat` see Section 10.11.

4.9 Graphics setup

Add `\graphicspath{../graphics/img-src/}` to preamble. Store all graphics as follows. All raster/bitmap graphics files (`jpeg`, `png`, and `tif`) should go to `./graphics/bitmaps` with any number of subfolders. All postscript `eps` files go to `./graphics/eps`. Before compiling `tex` files make sure that a script is run, `mergeGraphics.bat`, which puts all graphics graphics from `bitmaps` and `eps` into `./graphics/img-src/`.

A similar setup applies to graphics not added manually. Any graphics which are programmatically produced (e.g. with Matlab as described in Section 10.8) must be setup to be automatically moved to folder `./graphics/3rdparty`. The reasoning is that those graphics files are not "original" input but "derivative" (it is rather Matlab or Java etc code generating them constitutes original input), hence they can be overwritten by corresponding applications at any moment. In other words they do not belong to folders described just above for original graphics. This also includes certain `.tex` files automatically generated by, e.g., MATLAB and which are required for the correct displaying of graphics annotations. Just as with actual images they are saved as "derivative" `.tex` files programmatically in `.\graphics\3dparty`. So make sure that all third-party application output is automatically saved to `3rdparty`. Calling `mergeGraphics.bat` with `overwrite` flag will automatically get all files from `3rdparty` to `img-src` in the way consistent with the above.

If writing matlab code to produce graphics with `laprint` and save them to `3rdparty`, add to it matlab script `uploadEpsToImgSrc(fileName)` which will take an `eps` generated by `laprint` from `3rdparty` and copy it and its `pdf` clone to `img-src`. So `mergeGraphics` will not have to be called manually for each run of matlab script. Function `uploadEpsToImgSrc` is to be called after `laprint` is called in your matlab code. Examples and more information is given in Section 10.8.

The similar workflow applies to graphics produced by INKSCAPE and documented in Section 10.11. The `3rdparty` folder will contain an extra subdirectory `inkscape`.

WARNING! Never put any files in `./graphics/img-src/` manually as it is a synthetic folder with all files being deleted by the script at some stage. All files stored there will be lost. The detailed discussion about the strategy given here and graphics formats is given in Section 10.1.

4.10 Compiler choice: `latex` vs. `pdflatex`

In certain cases, one can be forced or strongly encouraged to use only one of the compilers available with MikTeX distribution (such as force by a choice of `laprint`, see Section 10.8).

Choosing `latex`	Choosing `pdflatex`
• Can import figures from MATLAB with `laprint` and `psfrag` (see Section 10.8) • If the `dvi` file is further compiled to `pdf` with `dvips` and `ps2df`, you may put a bit more reliance that the output will appear exactly the same way as in the dvi viewer. • Urls may not be broken correctly or not broken at all. A workaround for this though is to use package `\breakurl` if compiling to dvips. Perhaps it is best to import it with `ifpdf` support of Section 5.2. • smaller sizes if compiling `latex` with certain graphics or fonts. `pdflatex` embeds all the fonts while `latex` not. The size varies with either `latex` or `pdflatex` case if using `\usepackage[T1]{fontenc}` too.	• Package `microtype` uses only `pdflatex`. This is one neat package allowing beautiful typesetting in narrow columns and narrow containers such as labels (also see Section (11.7) for more on how it can be used). The following features are possible with `microtype`: character protrusion, font expansion, additional kerning, spacing, tracking/letterspaciing. • Full support for correct url breaking with `hyperref`. In contrast, urls with `hyperref` might not be wrapped under certain conditions with `latex`. This still can be fixed if `breakurl` imported. • Full support of `hyperref`. One small unresolved issue with `latex` is that links are visible but not clickable in the DVI viewer. Another suspicion is that `hyperref`'s behavior is different in the two profiles and has to be tweaked such as using various drivers in the declaration line (read answers to my posts).

Somebody in the groups suggested that `latex` and `pdflatex` profiles could still be used with `psfrag` and `laprint` if employing `auto-pst-pdf` or `pst-pdf` . `auto-pst-pdf` provides `\matlabfig{}` to insert `laprint`-generated graphics. I tried `auto-pst-pdf` but could not get rid of error messages (tried `\matlabfig{ml_output.tex}` and `\matlabfig{ml_output}` to no avail).

(Another suggestion was to switch from `laprint` to `matlabfrag` then `pstool` could be used instead and thus both compilers)

4.11 Infrequent and project specific config files: `sty`, `cls`, etc

Configuration files which are reused commonly are stored in the LaTeX repo and are available since the MikTex compiler knows where they are as described in Section 4.1. For project specific configuration files which are rare and for which it is unwarranted to put them in the repo, or which are best to be distributed with the project e.g. for collaboration purposes, the following convention is adopted. They are to be placed in a directory separate from `src`.

Specifically, all `sty`, `ind`, `cls` and other files necessary to the project should be stored in the directory `config`.

For all those config files to be included in the build the following changes must be made to the `latex` and `pdflatex` commands[2]. Just add the following parameter to it: `--include-directory="..\config"`. Repeat for DVI and PDF profiles and other profiles too if required.

An alternative treatment for config files is to prepend their import statements with the full path. E.g. `\documentclass[12pt,onecolumn]{../config/IEEEtran}` for document class or `\usepackage{../config/selectcp}` for an ordinary package. However, upon a quick test, some packages refused to be imported correctly under this approach. So, perhaps better stick to the first approach of supplying a command line variable.

4.12 Project Setup

Make sure that the project name does not contain any spaces. Each project is to contain the following tree structure:

`.` - root to contain 4 shortcuts to scripts for backups, 2 shortcuts to scripts for graphics processing, 1 shortcut to a bibliography processing script, and the textfile with the list of directories for one of selective backups (Section 4.8)

`.\bckp` - for backup files generated by scripts

`.\bib` - bib.rtf file outputs from Biblioscape, its plain text copy bib.bib which was created by one of the scripts, plus their raw.rtf and raw.bib cousins (Section 4.13), and a shortcut to the biblioscape catalogue

`.\config` - see Section 4.11

`.\cvs` - for older version copies (unpacked), keep as few folders as required but perhaps bib, config, graphics (without img-src), matlab-laprint, and src are the minimum to recover the old version if required completely.

`.\graphics` - see Section 4.9

`.\inkscape` - see Section 10.11

`.\matlab-laprint` - see Section 10.8

`.\otherInput` - anything else which does not fall in any other categories

`.\src`

4.13 Project Setup: Bibtex Files

It's my convention to store the bibliography file called `bib.bib` with all document references in `../bib/` folder relative to the folder where the source latex files are saved. Add `\bibliography{ ../bib/bibliography_file}` where `bibliography_file` is the name of the the file, e.g. in this case `bib.bib` file.

[2]For example in TeXnicCenter they can be found in `Define Output Profiles` menu under the `Command line arguments to pass to the compiler` box

Remove all `bst` files from all MikTex roots and its `Program Files` installation folder. Instead copy them all under `\repo\latex\bibtex` folder, and add the folder to the MikTex roots. This way you will know exactly which `bst` files are being used. The further tree convention is to separate between the standard MikTex distribution's `bst` files and any personal additions made on top of it. All unused files need to be renamed by, e.g., prefixing their names with a tilde.

Regarding the convention for keeping bibliographies while writing a paper, it was decided to have 2 different databases and hence 2 different bibtex files. One is for fully formatted bibliographies, and the other one is draft. The idea is to minimize manual work when it is unclear if a particular reference is gonna be used in the paper. We still need both bibtex files to create the proper bibliography citations in the document while writing it, however, it is only at the very end of the writing process there is going to be only one fully formatted bibtex file, the file used to generate the final paper draft. According to the convention adopted the following command `\bibliography{../bib/bib,../bib/raw}` should be used. So now citations from either file [3, 6] could be inserted.

5 Configuring LaTeX Environment

5.1 Conditional Statements and Global Boolean Variables

Global variables can be declared and used from package `\usepackage{ifthen}`. Declaration `\newboolean{sunday}` while evaluation `\setboolean{sunday}{true}`. Once they are initialised you can use `if` statements with

```
\ifthenelse{\boolean{sunday}}
    {This is sunday}
    {his is no sunday}
```

Another approach with conditionals is described in Section (5.2).

Yet another approach is described in Section (5.5). Bear in mind thought that that section uses plain TeX commands and thus might not be suitable for certain situations. For example, if declaring a global boolean variable `cond` like this

```
\renewcommand{\flag}{\newif \ifcond\condtrue}
```

and if using it with a conditional TeX statement like `\ifcond\else\if`, you *cannot* change the value of that variable inside the conditional, i.e. you *cannot* write `\condfalse` inside the `ifcond` statement. That's why sometimes the only way of handling the situation is to use `ifthen`. On the other hand sometimes the `\renewcommand` way of variable declaration can be the only option, particularly if it is required before `documentclass`. The use of both approaches is necessitated and demonstrated in Section (5.5).

5.2 Conditional Compilation: `latex` and `pdflatex`

Sometimes different package options have to be loaded with different compilers. One example of this is given in Section 13.3. Use `ifpdf` package for this in the preamble. This is an example which worked for that section:

```
\usepackage{ifpdf}
\ifpdf
  \usepackage[hyperfootnotes=false]{hyperref} %add version for pdflatex
\else
  \usepackage[dvips,hyperfootnotes=false]{hyperref} %add version for dvi
  \usepackage{breakurl}
\fi
```

Note however, that the use of option `dvips` above is discouraged as discussed in Section 7.9. If `dvips` is replaced by `hypdvips` the two lines above become identical and can be moved above the `ifpdf` routine.

Other examples of implementing the conditionals in LaTeX are given in Section 5.1.

5.3 Output only selected pages

Use the `\outputonly{2-3}`^{selectp} command in the preamble, where 2-3 is the range of pages to print. I think it supports a list of pages as well if required.

Syntax: use only with space separation! E.g. `{2-3 4 5 7-9}`.

5.4 Commands and Environments

My current limited knowledge is as follows. In case of the following command:

`\newcommand{\resitem}[1]{\item #1 \vspace{-2pt}}`

we have:

`{\resitem}` - command name declaration

`[1]` - how many arguments (i.e. items in `{}` the command would have

`#1` - refers to the 1st argument (i.e. the content of the first set of `{}`)

and the usage is simply `\resitem`.

An example of the environment declaration is as follows:

```
\newenvironment{Roles}
        {\begin{minipage}{\textwidth}}
        {\end{minipage}\par}
```

and its usage is with standard `\begin{Roles}` `\end{Roles}` markers.

5.5 Working with large documents

The following file organisation is arguably the best for handling very large multi-document projects. If the following describes the file system

```
.\src\boot.tex
.\src\header.tex
.\src\footer.tex
.\src\chapters\ch1.tex
.\src\chapters\ch2.tex
.\src\chapters\footer.tex
```

Two solutions exist, only the first one was tried at this point. The second one seems to be superior.

- **A naïve aproach** Split the entire projects into chapters and then use `\include` and `\includeonly` commands from the bootstrap file `boot.tex`[3]. However, the main disadvantage is that the individual files will not contain any preambles and cannot be compiled individually. As a result working in TeXnicCenter requires editing in `ch1` while compiling from `boot`, i.e. constant inconvenient swapping between tabs. A possible quick workaround: using command line for compiling and viewing. Still less convenient as extra redundant keyboard strokes have to be issued.

- **Best approach** Use a smart typesetting approach as outlined below.

We want to be able to compile the project from either `boot` or individual `chXX` files plus to rely on `\includeonly` as the way to work on standalone sections of the entire project from either starting point. The simplest approach is described here. Specifically, let `boot.tex` be:

```
1  \providecommand\projectroot{}
2  \input{\projectroot header}
3  \renewcommand{\setflag}{\newif \ifskipheader \skipheadertrue}
4  \setboolean{mainfile}{false}
5  \include{chapters/ch1}
6  \include{chapters/ch2}
7  \input{footer}
```

while chapter file `ch1.tex`:

```
1  \providecommand\projectroot{../}
2  \input{\projectroot header}
3  \renewcommand{\setflag}{\newif \ifskipheader \skipheadertrue}
4  Text
5  \ifthenelse{\boolean{mainfile}}{
6    \setboolean{mainfile}{false}
7    \include{\projectroot chapters/ch2}
8    \input{footer}}{}
```

Either file loads the preamble of the project up to `\begin{document}` with line `2`. After this was done once we want any further calls to `header.tex` to be skipped by using line `3`. If we run the project from `ch1.tex` we add the content of the chapter at this point as on line `4`, and after that enter the `if-then` loop on lines `5-6`. From this point line `6-7` of `ch1.tex` and lines `4-6` of `boot.tex` are identical in terms of marking any further recursive calls to individual chapter `tex` files as being calls to non-*mainfiles* by setting variable `mainfile=false`. This way any recursive calls to all other chapters will skip any `if-then` loop equivalents of lines `5-8` in `ch1.tex`. Note this also guarantees only one input of the footer per project `footer.tex`:

```
1  %\bibliographystyle{IEEEtranSN}
2  %\printglossaries
3  \end{document}
```

[3]Command `\input` can be used instead. `Include` starts a new page and allows you to use `includeonly`, and hence could be preferred for this particular case.

A few more details are revealed if we look at `header.tex`:

```
1  \providecommand{\setflag}{\newif \ifskipheader \skipheaderfalse}
2  \setflag
3  \ifskipheader\else
4  \documentclass[a4paper]{book}
5  %\usepackage{glossaries}\makeglossaries
6  \usepackage{ifthen}\newboolean{mainfile}\setboolean{mainfile}{true}
7  %\includeonly{\projectroot chapters/ch2}
8  \input{\projectroot backmatter/glossary.tex}
9  %\begin{document}
0  \fi
```

Lines `1-2` conditionally reset variable `skipheader` which signals when to load the header or when to skip it. `\providecommand` is used to make it conditional, i.e. being able to initialise its value to `false` only on the first `input` of `header.tex`. Line `1` will be ignored at any consecutive calls to `header.tex`, thus providing a mechanism to set this value to `true` elsewhere in the source code and thus skip headers on consecutive runs. Specifically we set `skipheader` to `true` either on line `3` of either `boot.tex` or `ch1.tex`; note that here we use the unconditional `\renewcommand` instead to overwrite the `false` value which was initially set when `header.tex` was called immediately before then. Clearly, as a result the code inside the `if-else` statement on lines `3-10` of `header.tex` is only executed once.

On the first run, `header.tex` loads all preamble statements up to `\begin{document}` including one extra boolean variable `mainfile` which signals if the call to the current `tex` file is a *mainfile* or *not*, i.e. if it is a recursive call from `boot.tex` or another chapter. We also set `mainfile = true` in the preamble to be able to enter the loop on lines `5-8` of `ch1.tex`.

Lines `1` in either `boot.tex` or `ch1.tex` define the current location of the main bootstrap with variable-command `projectroot` thus providing a location cue to commands `input`, `include` and `includeonly`. This way we can store `boot.tex` in the source root while keeping chapters wherever we want them to be (i.e. in `chapters` folder). Note a bit awkward spelling of the argument which is `<>` in `boot.tex` and `<../>` in `ch1.tex`. This is because `includeonly` command requires the relative path to start either from `../` or the name of the subfolder e.g. `chapters`, in other words it will reject either `/chapters` or `./chapters`. Otherwise we would have defined `projectroot` with more customary `<.>` and `<..>` in respective files.

This setup allows to load only a single chapter thus saving a lot of compiling time required for larger projects. Note the use of `includeonly` in `header.tex`. As with a standalone usage of `includeonly`, for references to work correctly, the *entire* project has to be compiled at least once beforehand to generate `aux` files for individual chapters. Once it's done you do not have to do it again. Do not forget to indent out chapters either in lines `5-6` if working from `boot.tex` or line `7` if compiling from `ch1.tex`.

In terms of chapter orders, if desired, code similar to lines `5-8` of `ch1.tex` can be added to immediately below line `3` of `ch1.tex` to allow for correct numbering of chapters. Do not forget to remove a call to the footer in that case.

A final remark for this setup is that footers will be taken from the directory local to the bootstrap `tex` file. If this is not required prepend the path to a unique `footer.tex` file to the `input` statements in both `boot.tex` and `ch1.tex`.

More information about why two types of conditionals were used in this approach is available in Section (5.1).

6 Citations

6.1 Exporting citations from Biblioscape to bibtex (*.bib) format

Note that the steps below after the `rtf` file was created have been automated by the scripts described in Section 4.1.

- In BIBLIOSCAPE: Make sure that `Output style` drop down menu on the toolbar is set to `BibTex mine`[4].
- Select required entries, if nothing is selected the entire currend folder will be exported
- Select `BibTex mine` in File/Export/RTF/Formatting file for journal...
- Save file as *.rtf
- In Word: open *.rtf file[5]
- Save it as a plain *.txt
- Rename it to a *.bib file
- It is now ready for use by LaTeX[6]

6.2 Generating bibtex bibliography from BIBLIOSCAPE

A very detailed discussion about various bibtex types, styles etc used in IEEE publications and examples of referencing a variety of sources used in Computer Science is available on the Internet [1].

To address all the problems given in this section a special BIBLIOSCAPE stylesheet `BibTex mine` was created. A copy of the file is kept in `\repo\latex\biblioscape\`. All editing to it is done inside BIBLIOSCAPE and it can be changed at will if required. The following describes the conventions adopted for this new stylesheet.

1. Only the following types are defined. Also, more "manual" clones could be created as required.

bibtex	Biblioscape
article	Journal article, Journal article (manual)
book	Book, Book (manual)
incollection	Book Edited
inbook	Book Section
inproceeding	Conference Proceedings, Conference (manual), LNCS
electronic	Generic

[4] Note that the `BibTex mine` is the style which I created myself. More details on how the style file was created and the decisions about this are in Section 6.2

[5] All steps from now on are automated as described in Section 4.1

[6] **ATTENTION**: *Do not make any changes to either* `bib` *or* `txt` *files directly if there is anything wrong with the exported bibliography. The master copy of the bibliography is supposed to be stored in the Biblioscape database. All formatting with respect to fields must be done in Biblioscape. This also includes changing them if a duplicate bibtexkey is generated by Biblioscape. Never change bibtexkeys or any other fields outside Biblioscape.*

patent	Patent
techreport	Report
phdthesis	Thesis

2. Manual clones of standard types are exactly the same as their respective originals but in one respect: they use the content of the `File_As` field for generating bibtex keys.

3. Each type has at most the fields listed in the IEEE reference document [1].

4. Symbols for separating fields in bibtex entries has been changed from the default BIBLIOSCAPE double quotes to braces.

5. The word `Proc.` was hard-coded into the two `Conference` types. See below.

6. The type `Generic` was rewritten completely to allow only the `Title` and `File_As` fields as described below in this section. Note that the mapped field for `Generic` is actually `electronic` not `misc`. This is because `IEEEtran` inserts quotes around titles in `misc`. An example of using `Generic` with references containing urls is given below.

Unwanted fields in the document Biblioscape will export some fields which the user might not want to include in the bibliography page. The easiest way to deal with this situation is to edit the style sheet `BibTex mine` is BIBLIOSCAPE by deleting the fields I do not want to export. Just go to `Tools` / `Styles` / `OutputStyles` and select `BibTex mine`.

Accented characters If accented characters need to be displayed in bibliography any of the three methods mentioned in Section 8.17 can be used, unless the accented chars occur in the first author field.

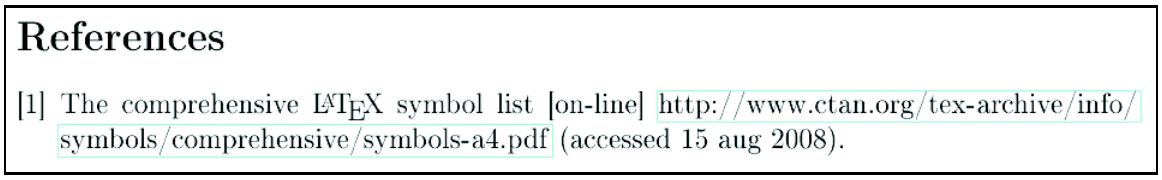
Missing author or year A particularly hard problem is how to reference a source without an author or year as the bibtex key is generated by BIBLIOSCAPE automatically in the {AUTHOR:YEAR} format. The solution is to redefine the bibtex key generation for `Generic` as just {File_As} which is implemented in `Bibtex mine`. So the BibTex key must be manually entered to the field `File_As` and then all information with required latex-syntaxed formatting field to `title`. Do not forget that special characters like braces, backslashes etc must be escaped with a backslash. However, this approach is not reasonable for references with too many fields such as, for example, references from conference proceedings. For this type of references defining a manual clone of an already existing style is more reasonable.

Sorting Bibliography with `natbib` and `Generic` The task is complicated here since sorting is performed with field `Author`, and this field does not exist in `Generic` type as discussed above. Hence, if using natbib with an option to sort bibliography entires, e.g. with `IEEEtranSN.bst`, you will receive warnings:

```
Warning--to sort, need author, organisation, or key in SOMEID:2009
```

Additionally, all `Generic` entries will be put on top of the list in the bibliography section of the produced manuscript. To resolve this issue the definition of `Generic` type needs to be redefined to include a redundant field `key` which will contain the same value as the entry ID, i.e. SOMEID:2009. This value will be used for the correct sorting of bibliographies.

References with urls It is another example where type `Generic` is possibly the best choice such as with referencing of resources on the web. To produce a bibliography entry similar to one given in [3]:



BibTeX will require the following entry in the bibtex file:

```
title={{\LaTeX} List [on-line] \url{http://www.ctan.org/a.pdf} (accessed 15 Aug 2008)}
```

and its corresponding entry in BIBLIOSCAPE should be:

```
\{\\LaTeX\} List [on-line] \\url\{http://www.ctan.org/a.pdf\} (accessed 15 Aug 2008)
```

A few remarks about those. First, note that all reference information is provided in a single field "title". Second, note the choice of `url` for formatting urls, not `texttt`. This allows the url be broken down nicely if it's too long as is shown in the figure. Third, special characters such as {,},\ etc must be escaped with the backslash in BIBLIOSCAPE. Finally, the url string can be typeset in a smaller font[7]. Although do it in conjunction with warnings of Section 13.3.

Discussion on how to treat special characters in urls if using them in bibtex files can be found in Section 6.4.

If compiling to `dvi` with `latex` urls in general and in references in particular may not break properly. A workaround is given in Section 4.10.

First author with special or accented chars or whitespaces This is much harder since a special character or a whitespace has to make it to the bib key identifier and thus it will throw an error no matter which of the three ways of displaying accented characters you would use. The solution when a source of this type occurs is to create a clone of an already existing BIBLIOSCAPE type, e.g. `Conference (manual)` and configure it identically to `Conference`, but with the BibTex key taken from the `File_As` field. Then a value for the BibTex key is directly typed in that field. Be careful with defining a manual clone for Conferences since in the automatic Conference Proceedings the word "Proc." is added programmatically.

Double quotation marks with Biblioscape Since by default the boundary symbol for field values in BIBLIOSCAPE is " (double quote), no double quotes can be used by default anywhere in field values. This is particularly hard if one wants to use the escape sequence for an accented character such as `\"{o}` for the German umlaut ö (see Section 8.17 for more details). The bibtex file will not be parsed at all. So the solution is to rewrite all definitions in BIBLIOSCAPE such that all quotes are replaced with braces. Again this is implemented in the style sheet `BibTex mine`.

[7] A command for it can be found http://www.kronto.org/thesis/tips/url-formatting.html

Preserving formatting of strings typed in Biblioscape It must be a bug somewhere through exporting from BIBLIOSCAPE that a title with custom typed headers like "M.I.T." will be automatically converted to "M.i.t". Generally the best way to deal with this is to get into the habit of surrounding all titles with the braces so the formating gets preserved exactly as it was typed in BIBLIOSCAPE. But since BIBLIOSCAPE removes all braces from all fields when exporting them, braces must be escaped with a backslash. In that case they make it to a bib file as braces (loosing escaping backslashes on the way).

Adding "Proc." to a conference name Also added to the `Bibtex mine` so it is added automatically. If defining the manual `Conference` clone as discussed above do not forget to add the word "Proc." to the programmatic definition of the field `Conference Name` in `Bibtex mine`.

6.3 Generating HTML bibliography from BIBLIOSCAPE

HTML code can be almost effortlessly generated from a BIBLIOSCAPE database. This describes how to do this with a couple of extra customisations due to the fact that the generated bibliography is likely to be my own portfolio, hence, it will have different requirements from publishable material such as having links to its pdf files urls and also providing extra information in the reference listings which otherwise will be left out. This approach requires a new BIBLIOSCAPE output stylesheet being maintained, `IEEE Journals`. A copy of the file with exported settings is in `\repo\latex\biblioscape\`.

1. Make sure `IEEE Journals` output style is selected in BIBLIOSCAPE. The current backup contains its entries for 5 reference types: edited book (UOA conference proceedings I chaired), conference proceedings (2), journal article, and LNCS (used to provide more info about LNCS conferences I published in, such as editors and LNCS volume information).

2. This output style uses formatting which I configured manually

3. This output style also has an extra field for creating an `href` tag in the output. The field used for this is called `Description` and it should contain the exact path (presumably only relative, but it might also work with absolute ones) to the url, e.g. SHORIN08.pdf for the file in the same location as the generated bibliography.

4. This url location needs to be entered in every reference in the database you are using if you want them to be linked in the final html document.

5. In BIBLIOSCAPE's Output Styles mode I appended the field called `Description` to the end of the `Templates` list and then entered the following text in it:
   - text before the field: `</a> <a href=`
   - text after the field: `>link`

It simply means that we are adding an extra string to the generated code such that it terminates the previous <a> tag which is automatically generated by BIBLIOSCAPE, then we create a new <a> tag for our url link. Note that the link will be automatically entered from the field `Description` in our references. Finally we close the <a> environment and add the word `link` which will appear as the name of the url link on the html webpage. Note that we do not terminate our string with since this is done automatically by the code

generated. It's a bit hacky, but this is the way BIBLIOSCAPE generates its html code which requires me to create this bizzar hack.

6. Make sure all references contain the url for their respective sources in `Description`

7. Before exporting the database, if desired, select `IEEE Journals` from the toolbar drop down menu called `Output Style`. This will make sure that this style is automatically selected each time you use the exporting dialogue.

8. Click `File\Export`, make sure that `IEEE Journals` is selected as a journal and that `HTML` is selected as file type.

9. Open the generated html file and run the replacement to delete all `<!>` stings.

10. Done. The generated html code will have one reference per line and it can simply be copied over as required.

11. The generated bibliography will be numerically sorted and numbered. If a different numbering system is required, try playing around with `Citation Number` in the list of Templates in output styles. If more than one lists is required with restarted numbering it can be accomplished by simply arranging references in separate child folders in BIBLIOSCAPE.

6.4 Stylistic notes and common errors

Conferences: dates When adding dates to a conference reference put them in the same field as `month` with a double hyphen and the comma at the end. The problem is that if just the month is given without the range of dates, then the output would be, say, March 2005, while if we add days we want to see the comma just before the year as in, say, March 23--25, 2005.

Do not try to use the IEEE macro as suggested by IEEE on page 4 of their bibtex reference manual [1].

Conferences: publisher and location They make the comma disappear (in fact it is replaced by a period) in `IEEEtran` if used together, which is different from the presentation when all fields in the bibliography are separated by commas. If you do not like the look you can simply remove the publisher, otherwise leave it since this 'undesired' behavior is still consistent with the way books are referenced.

Urls in a typed font Add latex syntaxed formatting to the url field if required and remember that introduction of `\texttt{}` to a BIBLIOSCAPE field requires escaping both characters `\` and `{`. However, there is a beter way as described in Section 6.2.

Special characters with bibtex and BIBLIOSCAPE It is mentioned already that `{}`, `_` and `\` must be escaped with a backslash in BIBLIOSCAPE. In addition there are other special characters to be treated separately. The tilde (`~`) has to be replaced with the sequence `\~{ }` where there is a single whitespace between the braces.

More then four authors If there are more than four authors it is customary to say "AuthorOne et al". It does NOT have a period after "et" and DOES have one after "al".

Using citing numbers as words A common mistake, try to visualise it as the illegal "A similar strategy is discussed by [15]". However, this usage will be allowed: "AuthorOne discusses this point further in her dissertation [2002]."

6.5 Choosing your own bibliography style

Generally it is a pretty hard task if I to take GG's style into account. The closest available is `IEEEtran.bst` which still has a few unwanted differences such as the last names coming after initials, titles surrounded in quotes, editors of the `incollection` type not surrounded in paranthesis, pp. vol. no. used. But still it is the best alternative. See Figure 1 for detailed breakdown of formatting of 9 alternative styles for 5 types of references. The Uni `thesis` bst file is really bad and was not even included. Also if the "author-year" referencing is required try one of the variations of the `IEEE` style which comes with the support for the `natbib` package, `IEEEtranN` and `IEEEtranSN`. Bear in mind, though, that (a) those two styles are not standard for IEEE publications, (b) some citation formatting discussed in Section 6.6 may not work without those two. So it's a tradeoff.

6.6 Choosing your own citation style

Various citation styles using Author-Number or Author-Date are provided in the LaTeX cheat sheet. Most of them require `\usepackage[numbers]{natbib}` or `\usepackage{natbib}` respectively. Hence the restriction on which bibliography styles can be used with those. E.g. usual IEEEtran bibtex stylesheets will not work with those unless one imports `IEEEtranN` or `IEEEtranSN` as discussed in Section 6.5. Other alternatives to using `natbib` in these circumstances are given in Section 6.7.

6.7 Configuring `natbib` with LNCS and IEEE

Package `natbib` may be required if using the Author-based system of citation or with extended citation commands like `\citeauthor{}`. Unfortunately, `natbib` has a couple of problems such as starting a bibliography from a new page (which is unacceptable in papers) and calling it "Bibliography" rather than "References". As discussed in Section 6.6 the IEEE has separate stylesheets if `natbib` is required. However, LNCS has only a single stylesheet, `splncs.bst`. So to resolve the problem the `natbib` import must be customised to include the following:

```
\usepackage[numbers]{natbib}
\makeatletter
\renewcommand\bibsection%
{
  \section*{\refname
    \@mkboth{\MakeUppercase{\refname}}{\MakeUppercase{\refname}}}
}
\makeatother
```

features	abbrv	acm	amsplain	plain	siam	splncs	unstr	ieeetrans	GG
citations									
[1-2,5,4] or [1-5] ?						short	short		-
general									
author all initials			+	+			+		no
last name first		+				+			yes
author small caps		+			+				-
authors terminated with	.	.	,	.	,	:	.	,	.
all dates surrounded by parenthesis						+			no
type=electronic									
auto insert [online] and available								+	-
auto insert accessed	-	-	-	-	-	-	-	-	-
all italicised			+		+				-
fatal errors	no url	no url	no url	no url	no url	key	no url	-	-
type=incollection									
title formatting			italic		italic			quotes	
word in used for series	In	In	-	In	in	In	In	in	-
editors	editors,	Eds.	eds.	editors,	eds.	eds.:	editors,	Eds.	eds.
editors in parenthesis			+						yes
series formatting	italic	italic		italic			italic	italic	italic
pages	pages	pp.	pp.	pages	pp.	-	pages	pp.	-
order: (l)ocation, (y)ear, (p)ages	ply	lyp	lyp	ply	lyp	l(y)p	ply	lyp	l:hpy
type=inproceedings									
title formatting			italic		italic			quotes	none
word in used	In	In	-	In	in	In:	In	in	In
conference formatting	italic	italic		italic			italic	italic	italic
pages	pages	pp.	pp.	pages	pp.		pages	pp	-
order: (h)ouse publishing	plyh	(ly)hp	(l)hyp	plyh	lyhp	lh(y)p	plyh	l:hyp	lhpy
type=book									
title formatting	italic	italic	italic	italic	italic		italic	italic	italic
location:publisher format								:	
type=article									
title formatting			italic		italic			quotes	none
journal	italic	italic		italic			italic		italic
, 15(10):1-3, 2006	+			+			+		+
15, 10 (2006), 1-3		+							
15 (2006), no. 10, 1-3			+						
15 (2006), pp. 1-3					+				
15(10) (2006) 1-3						+			
vol. 15, no. 10, pp. 1-3, 2006								+	

Figure 1: Comparative analysis of 9 styles with 5 reference types.

6.8 Duplicate bibtex keys (Author:Year) and their elimination

This might be not really trivial, however, JabRef can do this.

7 Document structure

7.1 Change default paper and font size

`\documentclass[a4paper,11pt]{article}` for A4 paper and 11pt font
`\documentclass[legalpaper,12pt]{article}` for US legal and 12pt font

7.2 Change page margin

The following produces 1 inch margins all around with no header or footer

```
\topmargin =0.mm      % beyond 25.mm
\oddsidemargin =0.mm
\evensidemargin =0.mm
\headheight =0.mm
\headsep =0.mm
\textheight =230.mm
\textwidth =165.mm
```

Play around with the settings here, there may be a better way of setting margins.

There's an easier way of handling margins if the user just wants to put as much text on a page as possible. All the user needs is to declare the `fullpage` package with an appropriate option. For example, `\usepackage[in]{fullpage}` will set all margins to 2.54cm. Using `empty` as option removes any spaces reserved for headers and footers. Other options exist. See the online documentation. The package requires the explicit declaration of paper size, i.e. legal or A4 as described in Section 7.1

7.3 Page headers

A simple piece of code is given below which creates a timestamp[8] with the word draft on the right hand side of the header on each page excluding the first one[9]:

```
\pagestyle{fancy}
\setlength\headheight{23pt}
\lhead{}
\chead{}
\rhead{\textsc{Draft}:\ \printtime\\
  \vspace{-0.7em} \dots\dots\dots\dots\dots\dots\dots\dots\dots}
\renewcommand{\headrule}{}
```

7.4 Footnotes

Just use `\footnote{Text goes here}` command. If more than one footnote reference at the same location we might want to insert a comma between them. Then just use `\usepackage[multiple]{footmisc}` in the preamble, everything else will be the same[10,11]. There is an incompatibility between `footmisc` and `hyperref` which might prevent this from working. See Section 13.5 for more info.

The package `footmisc` has another argument `para` which is if added to its declaration in the preamble will override the default latex setting of putting new references in the footnotes on a new line, and will append references immediately one after another.

Two remarks regarding multireference usage. First of all, the above usage of the `footmisc` package would not work if the `\VerbatimFootnotes` command from the `footmisc` package is used: reference numbers will be displayed without any separator (default latex option). See

[8] Requires the routines for time and date which can be found in section 12.1

[9] Requires package `fancyhdr`. Note that the package must be added to the preamble after `footmisc` as otherwise it will conflict with it by throwing a warning.

[10] Footnote one

[11] Footnote two

Section 8.6 for verbatim formatting production without this command. Secondly, `footmisc` may conflict with package `fancyhdr` by throwing a warning. Always put `footmisc` higher than `fancyhdr` in the preamble.

Package `footmisc` also conflicts with `setspace` with the latter had to be declared only after `footmisc`. Otherwise the error of Section 14.13 will be thrown.

Another problem with footnotes is that sometimes long url names do not get broken at all. The solution to this problem is discussed in 13.3.

If footnotes misbehave by being spillover to subsequent pages we may try to supress this behaviour by inserting `\interfootnotelinepenalty=10000` in the preamble. It penalises LaTeX for doing so. The disadvantage of this is that the compiler can throw `Underfull \vbox` warning messages. More information can be found on page called "splitfoot" on the official LaTeX FAQ website.

7.5 Page numbers

Generally pagination is done automatically, well at least for the 'article' style. However, the style of pagination can be altered by the `\pagenumbering{arabic}` command, with `arabic` being able to be changed to `roman, Roman, alph, Alph` if required.

Page numbering style can be reset for any part of text by calling a new style command following by the `\pagestyle` command anywhere in the text.

Other options include the use of the `\pagestyle{option}` with the following options:

- `plain` Just a plain page number.
- `empty` Produces empty heads and feet - no page numbers.
- `headings` Puts running headings on each page. The document style specifies what goes in the headings.
- `myheadings` Specify what is to go in the heading with the `\markboth` or the `\markright` commands.

If you want to specify a different numbering rule for a given page, for example, not to display a number for it, the `\thispagestyle{empty}` command can be used.

7.6 Notes on margins

There are two options: `marginpar` and `ed`. I find `ed` nicer for editorial notes as it provides for several lists (such as `ednote` and `issue`) and it also just marks an issue on the margin and puts its narrative to the footnote as a clearly identifiable editorial note. Just add `\usepackage[show]{ed}`, and use with `\ednote{text}` or `\issue{text}`.

Another option is the `\marginpar{}` command. There are a few possible customizations if required.

First, in order to place all margin paragraphs on the left (thus avoiding clashes with reference names produced when forcing LaTeX show all reference keys explicitly (see Section 13.2). This might help when keys of formulas appear too much on the right. Solution: add `\reversemarginpar` in the preamble.

Second, their width can be controlled, e.g., by `\setlength{\marginparwidth}{7.5em}`.

Third, `\marginpar` can be overwritten easily. For example, this command reduces the size of all margin paragraph to scriptsize

```
\let\myMarginpar\marginpar
\renewcommand\marginpar[1]{\myMarginpar[\scriptsize #1]{\scriptsize #1}}
```

Finally, two other options suggested in the newsgroup are `todonotes`, `fixme`. There are also packages for PDF- (or maybe rather Adobe Reader-)specific annotations such as `pdfmarginpar`, `fancytooltips`, and `cooltooltips`. This guy has only played around with them a bit. He likes that this kind of annotations act as an overlay to the document that does not disturb the layout of the document itself.

7.7 Title

Add the following in the preamble, i.e. not in inside the document:

```
\title{\LaTeX: Personal reference guide}
\author{Alexander Shorin \\
        Department of Computer Science\\
        University of Auckland\\
        PB 92019, Auckland, New Zealand\\~\\
}
\date{\today}  % your own text, a date, or \today
```

Then add `\maketitle` inside the document body.

7.8 Abstract

Just put it into the `abstract` environment with usual `begin` and `end` tags right after `\maketitle` in the document body.

7.9 Lines Fail to Wrap: Table of Content, List of Figures and Tables

Too long section titles or captions for figures and tables will fail to wrap in their summary pages. The problem is caused by the standard `dvips` package that does not support wrapped/broken links with `hyperref`. This is because entries in, e.g., ToC are treated as links by `hyperref`. Other drivers, e.g. `pdflatex` or `pdftex`, work with `hyperref` without any problem.

Three solutions are available for documents which require ToC, LoF or LoT. Most elegant is to define titles and captions short enough to fit in a single line. It improves readability and visual appeal.

The second approach should be employed when caption re-write is not acceptable. Simply import the `hypdvips` package after the import of `hyperref` for the `dvi` compilation profile implemented by the conditional `ifpdf` statement (Section 5.2). If `hypdvips` is used make sure that `hyperref` is imported without the option `dvips`. The usage of package `breakurl` is not affected by this modification. Finally, make sure that at least v.1.06 of 2009/03/25 is installed as earlier versions of `hypdvips` may conflict with `tabularx`.

Another solution is to import `hyperref` with the option which de-links captions/titles by, for example, linking to page numbers instead: `\usepackage[linktocpage]{hyperref}`. In this case `hypdvips` may not be required at all.

It was mentioned in many forums that one should discontinue using the `dvips` driver.

7.10 Sections and Table of Content

Suppress Section Numbering

Use *, e.g. `\subsubsection*{Some Chapter}`

A simple table of contents

Add `\tableofcontents` where the table should be in the text. No packages are necessary.

Unnumbered entries in ToC

If a legal section name (such as `\subsection` e.g.) is used for the text section then it needs to be starred to avoid inclusion to ToC. Then the entry needs to be coded directly as it is supposed to appear in text:

```
\subsection*{Dummy heading}
\addcontentsline{toc}{subsection}{\ \ Dummy heading}
```

obviously the names should be identical and the section legal name is to be the same as denoted by the command. The two blanks inserted in heading name are there for stylistic purposes if the current section to appear indented in ToC.

Other options is to use `\paragraph{}` instead of creating another starred section. Adding the paragraph to ToC is the same with probably only the `{subsection}` part to be called appropriately perhaps one level below the current section level where the paragraph appears.

The choice between the two options is purely stylistic.

Another option is to use one of `koma` classes (eg. `scrartcl`) and `\addsec`. This was not tested by me. Also see problems described in Section (14.11) for what can go wrong with `koma`.

7.11 A single appendix structure

Can be implemented using the `appendix` package. Put all appendices inside the `appendices` environment. The heading could be done as some `\huge` centered text or perhaps there are better ways of doing this which I have not investigated. Each new appendix begins with `\section`. Possible usage:

```
\begin{appendices}
\begin{center}{\huge Appendices}\end{center}
\section{Run MS Word macros from CL - ruby script\label{app:ruby}}
  ...
\end{appendices}
```

7.12 A simple index

- From package `makeidx`
- Words to be included into the index must be immediately followed by `\index` with curly braces containing the entry of this word in the index. For example, the word `superseding` needs to be changed to `superseding\index{supersede}`
- Whenever the index needs to be produced (e.g. at the end of the project) append `\makeindex` to the preamble and add `\printindex` to the place where the index is supposed to be. Otherwise comment these commands out to save processing time.
- Rebuilt a couple of times to generate the complete index

You might also read Section (7.14) to see how batch processing using `makeindex` can be carried out.

7.13 Manipulating definitions, illustrations and other objects[12]

Note this section describes the problem for which there might be a more straightforward solution offered by package `glossaries` described in Section 7.14. Other possible solutions to outputting lists of theorems and alikes can be provided with `ntheorem`, `thmtools` with its `\listoftheorems`. I have not tried them.

Sometimes an additional document-wide construct needs to be introduced, e.g. definitions, illustrations etc. The task is two fold. On the one hand we want it to be properly formatted, e.g. framed and shaded. On the other hand, we want to have a table of those constructs appearing together with, e.g. table of figures in the beginning of the document. An example of one possible implementation is given here.

Create and save a new `activity.sty` file and add to it:

```
\newcommand\floatc@greybox[2]{{\bf #1} #2\par}

\newcommand\fs@greybox{\let\@fs@capt\floatc@greybox
  \def\@fs@pre{\begin{center}\begin{minipage}{15cm} }%
  \def\@fs@mid{\kern5pt}%
  \def\@fs@post{\end{minipage}\end{center} }% }%

\let\@fs@iftopcapt\iftrue}
\floatstyle{greybox}
\newfloat{activity}{thp}{loa}[section]
\floatname{activity}{Activity}
```

Note that you can use `[section]` parameter here to define the level of numbering for the constructs, just replace it with `chapter` or `subsection` as required. The next step is to import two packages `float` and the newly created `activity` in the main tex file. Once it's done you can use it with

```
\begin{activity}[ht]
  Text of the activity goes in here...
  \caption{A nice activity caption}
  \label{dummylabel}
\end{activity}
```

to produce the desired output And finally generate the table of "activities" by adding the following line somewhere after the table of figures `\listof{activity}{List of Activities}`

[12]The solution was kindly provided by Jonathan Teutenberg

Activity 7.1: A nice activity caption

Text of the activity goes in here...

7.14 Glossary

α Install packages: `glossaries`, `xkeyval`, and `xfor`.

β Add `\usepackage[toc]{glossaries}` and `\makeglossaries` to the preamble; also just where you want the glossary to be printed - `\printglossaries`. Make sure the option `toc` is entered otherwise, it will not appear in ToC.

γ Create the batch file and place it in the root of the project:

```
set file=%~1
echo %file%
makeindex -s %file%.ist -t %file%.glg -o %file%.gls %file%.glo
```

δ Configure all profiles of TeXnicCentre by going to their postprocessor tab and creating another executable. Link it to the batch file in the root, and set the argument to `"%bm"`.

ϵ To generate index you must compile it at least twice. All output from the executable will be appended to the end of the texniccentre's output window so you can inspect it if anything goes wrong. It supports MSDOS debugging commands such as echo.

ζ You can add a new entry into the glossary index with the following command. Note that the entry can be either in the preamble or in the text. Probably the best is to place it into a separate `tex` file.

```
\newglossaryentry{ref}{name=nameInText,description={descriptorText}}
```

η It is customary to use the description field `user1={text}` to add the capitalised name identical to the field `name`. This is because the package does not provide for a command with all capitalised words. Something which has to be used in, e.g., heading titles.

θ The term can be used in text by using its reference as follows. The page where it's used will be added to the glossary list by default.

```
\gls{ref}                    - for descriptor "name" to appear verbatim
\Gls{ref}                    - for descriptor "name" with 1st word capitalised
\glsentryuseri{ref}          - for descriptor "user1" to appear verbatim
\protect\glsentryname{rlsm} - in fields requiring protection
```

There's more info on this subject[13] including Section (7.13) and an excellent FAQ on common problems[14] such as using glossaries in section headings etc.

7.15 List of Notation, Multiple Glossaries etc

The approach outlined in Section 7.14 applies as `glossaries` allows to define multiple lists. So follow the procedures in that section plus additionally:

[13] Excellent writeup at: http://www.latex-community.org/index.php?option=com_content&id=263:glossaries-nomenclature-lists-of-symbols-and-acronyms

[14] http://theoval.cmp.uea.ac.uk/~nlct/latex/packages/faq/glossariesfaq.html

1. Define a new glossary `\newglossary{symbols}{sym}{sbl}{List of Symbols}`, where the unique name of the glossary is given by `symbols`, and `sym`, `sbl` are the names of file extensions created and used by this package

2. Add a new entry to the glossary just as before making sure you specify the glossary which is in this case `symbols` — `type=symbols`. If the type is ignored the entry is added to the main glossary.

```
\newglossaryentry{pi}{type=symbols,name={\ensuremath{\pi}},sort=pi,
 description={ratio of circumference of circle to its diameter}}
```

3. Referencing items, printing glossaries and finding help is described in Section 7.14.

8 Formatting text

8.1 Embedding All Fonts to PDF Documents

This is sometimes required by some conferences, e.g. IEEE's Xplore-compatible PDF productions. I think this might not be a problem if outputting directly with `pdflatex`, but if using `latex` we need to adjust the arguments passed to `ps2pdf`. If using `ps2pdf` directly just add type in the command line:

```
ps2pdf -dCompatibilityLevel#1.4 -dPDFSETTINGS#/prepress filein.ps fileout.pdf
```

Bear in mind though that fonts used will be different than produced by TexnicCenter's built-in DVI-PS-PDF profile which relies on GhostScript as a replacement for `ps2pdf`. In that case just make sure thaat the following arguments are added under profiles to GhostScript `-dPDFSETTINGS=/prepress`, so the complete line in TexnicCenter becomes (as one line right under `gswin32c.exe`):

```
-sPAPERSIZE=a4 -dCompatibilityLevel=1.4 -dPDFSETTINGS=/prepress -dSAFER -dBATCH
-dNOPAUSE -sDEVICE=pdfwrite -sOutputFile="\%bm.pdf" -c save pop -f "\%bm.ps"
```

Note that `#` becomes `=` under `gswin32.exe`.

8.2 Displaying special characters in text

Sometimes one needs to display a non alphanumerical character in text, e.g. the backslash. The common way of doing it outside mathematical environments is to use their escape sequences, e.g. `\textbackslash`. The complete list of characters is given on pp.455–463 of [6]. Notice, that in order for most of those escape sequences to be correctly, i.e. without warnings, recognized by the system, one must use the `T1` font encoding. Simply add `\usepackage[T1]{fontenc}` to the preamble right after the `documentclass` declaration.

8.3 Formatting fonts

Regular Text someSampleText

Various Sizes

\tiny \scriptsize \footnotesize \small \normalsize \large \Large \LARGE

Verbatim Text `someSampleText`
see Section 8.6 for details

Text color someSampleText
`{\color{green}someSampleText}`^{color}

Background color someSampleText
`\colorbox{green}{someSampleText}`^{color}

Custom rgb color
add `\definecolor{lgrey}{rgb}{0.95,0.95,0.95}`^{color} anywhere

Boxed text someSampleText
`\fbox{someSampleText}`, also see verbatim boxing in Section 8.6

Boxed text someSampleText
`\fcolorbox{red}{blue}{\color{white}someSampleText}`^{color}
thickness of the frame can also be setup[15]

Strike out ~~someSampleText~~
`\sout{someSampleText}`^{ulem}

Underline someSampleText
`\uline{someSampleText}`^{ulem}

Underwave someSampleText
`\uwave{someSampleText}`^{ulem}

Font Change `someSampleText`
`\usefont{T1}{pcr}{m}{n}someSampleText`[16]

Font Change Warnings
sometimes warning is issued if one of the parameters discussed above is incorrect. Please consult the list for the available alternatives.

[15] Add `\setlength{\fboxrule}{0.9pt}` immediately before the `\fcolorbox` command

[16] Three arguments that can be changed are font family (pcr), series (m), shapes (n). Available values of these parameters for standard font families supplied with LaTeX could be found on page 372 of [6]

8.4 Available fonts

Regular Fonts

The quick brwn fx jmps ovrlazydg 123!@*+,/?>([	cmr	Roman
The quick brwn fx jmps ovrlazydg 123!@*+,/?>([	ptm	Times
The quick brwn fx jmps ovrlazydg 123!@*+,/?>([	ppl	Palatino
The quick brwnfxjmps 123!@*+,/?>)]	cmtt	Type
The quick brwn fx jmps ovrlazydg 123!@*+,/?>([	cmvtt	Proprtn
The quick brwnfxjmps 123!@*+,/?>)]	pcr	Cour
The quick brwn fx jmps ovrlazydg 123!@*+,/?>([	cmss	Sans
The quick brwn fx jmps ovrlazydg 123!@*+,/?>([	phv	Hlvtca
The quick brwn fx jmps ovrlazydg 123!@*+,/?>((	pag	AvntGrd
The quick brwn fx jmps ovrlazydg 123!@*+,/?>([	cmfr	FnnyRmn
The quick brwn fx jmps ovrlazydg 123!@+,/?>([*	pzc	Zapf(it only)

Italic Fonts

The quick brwn fx jmps ovrlazydg 123!@+,/?>([*	cmr	Roman
The quick brwn fx jmps ovrlazydg 123!@+,/?>([*	ptm	Times
The quick brwn fx jmps ovrlazydg 123!@+,/?>([*	ppl	Palatino
The quick brwnfxjmps 123!@+,/?>)]*	cmtt	Type
The quick brwn fx jmps ovrlazydg 123!@+,/?>([*	cmvtt	Proprtn
The quick brwn fx jmps ovrlazydg 123!@+,/?>([*	cmfr	FnnyRmn

Slanted Fonts

The quick brwn fx jmps ovrlazydg 123!@+,/?>([*	cmr	Roman
The quick brwn fx jmps ovrlazydg 123!@+,/?>([*	ptm	Times
The quick brwn fx jmps ovrlazydg 123!@+,/?>([*	ppl	Palatino
The quick brwnfxjmps 123!@+,/?>)]*	cmtt	Type
The quick brwnfxjmps 123!@+,/?>)]*	pcr	Cour
The quick brwn fx jmps ovrlazydg 123!@+,/?>([*	cmss	Sans
The quick brwn fx jmps ovrlazydg 123!@+,/?>([*	phv	Hlvtca
The quick brwn fx jmps ovrlazydg 123!@+,/?>((*	pag	AvntGrd

SmallCaps Fonts

THE QUICK BRWN FX JMPS OVRLAZYDG 123!@*+,/?>([	cmr	Roman
THE QUICK BRWN FX JMPS OVRLAZYDG 123!@*+,/?>([	ptm	Times
THE QUICK BRWN FX JMPS OVRLAZYDG 123!@*+,/?>([	ppl	Palatino
THE QUICK BRWNFXJMPS 123!@*+,/?>)]	cmtt	Type
THE QUICK BRWNFXJMPS 123!@*+,/?>)]	pcr	Cour
THE QUICK BRWN FX JMPS OVRLAZYDG 123!@*+,/?>([	phv	Hlvtca
THE QUICK BRWN FX JMPS OVRLAZYDG 123!@*+,/?>((	pag	AvntGrd

Other fonts could be found in [6] pp. 354, 372-376, including `Bookman` (`pgk`), and `Schoolbook` (`pnc`). Both are sisters of `Roman` and `Times` but are bigger and thicker (`Bookman` is even more so). For usage see Section 8.5.

8.5 Common Font abbreviations

There are several categories of font classes describing visual variations within individual classes: weight, width and shape. These are presented in Figure 2, and their usage is `\usefont{T1}{pcr}{m}{n}someSampleText` where `pcr` is font family from Section 8.4, `m` is series (width or weight) depending on font, and `n` is shape.

#	Weight
ul	Ultra light
el	Extra light
l	Light
sl	Semi lIght
m	Medium
sb	Semi Bold
b	Bold
eb	Extra Bold
ub	Ultra Bold

(a) Weight classes

#	Width
uc	Ultra Condensed
ec	Extra Condensed
c	Condensed
sc	Semi Condensed
m	Medium(normal)
sx	Semi Expanded
x	Expanded
ex	Extra Expanded
ux	Ultra Expanded

(b) Width Classes

#	Shape
n	Normal
it	Italic
sl	Slanted
sc	Small Caps
ui	Upright It
ol	Outline

(c) Shape Classes

Figure 2: Abbreviations used for describing various classes of fonts

8.6 Verbatim text

This is a bit cumbersome in LaTeX and depends on where verbatim text is used and if it is multiline.

Using verbatim in main document body If verbatim is used in the main document body the simplest approach is to add the command `\MakeShortVerb{\|\}` from the `shortvrb` package anywhere beforehand (e.g. in the preamble or immediately before verbatim text). The symbol '|' in the command syntax is declared to be used as a special character delimiting verbatim text. Once no more verbatim text is expected '|' can be relieved for regular use by calling `\DeleteShortVerb{\|}`. And of course we are free to declare any other symbol a delimiter, e.g. '+'. Usage: |someSampleText| will produce `someSampleText`.

If a vertical bar needs to be used in verbatim text, an almost identical built-in alternative command can be used: `\verb/some|||||Text/` will produce `some|||||Text`. Notice as with the `shortvrb` package the delimiter '/' used above can be replaced with anything else, e.g. '+'. I do not know any other reason to use this package though.

An alternative method for main text body is just to surround text with `\texttt{}`. Notice however that not all characters can be by default inserted inside the env. Investigating right now.

For multiline formatting in the document body, e.g. for displaying source code, the previous methods will not work as well. Use the `Verbatim` environment from the `fancyvrb` package with optional font size and framing parameters:

`\begin{Verbatim}`^{fancyvrb}

```
    [fontsize=\relsize{-1},frame=single,framesep=3mm]
    ...
\end{Verbatim}
```

The added bonus of the `Verbatim` is that the chunk of text can be easily put in a frame, and it generates a box which is buffered with empty spaces from above and below text. However, if text also needs to be formatted further, e.g. with colors, we should not use the `Verbatim` (although exploration of the package documentation might resolve this difficulty). See Section 9.7 for an example of non `Verbatim` text.

Verbatim in footnotes Never use `\VerbatimFootnotes` from the `fancyvrb` package. It screws things up in a number of ways, e.g. multiple footnote references etc. This singles out the use of the `Verbatim` env. Neither the `\verb` command can be used in footnotes.

With footnotes the first choice is `\texttt{}`[17] or the `shortvrb` package shortcuts alternative. Note, however, if using `shortvrb` there is its strange behavour in footnotes in which all special characters such as backslashes etc need to be escaped inside the verbatim text. So probably better to use `\texttt{}`.

Verbatim in section headings With section titles the situation is harder since even the `shortvrb` package will refuse to work. To deal with this there are a number of methods.

This (bad) complex method described in detail on pp.165–167 of [6] works only with dvi. Basically each phrase to be used verbatim in section commands needs to be declared in the preamble with the following declaration line: `\SaveVerb{verb1}=Some \phrase=`, where `verb1` is the name of the variable (container) holding the phrase `'Some \phrase'`. So in the heading the text can be inserted by using the following syntax: `\protect\UseVerb{verb1}`, which just calls the content of the variable `verb1`. Note that both commands, `\SaveVerb` and `\UseVerb`, require the package `fancyvrb`. More info on this is in http://www.tex.ac.uk/cgi-bin/texfaq2html?label=verbwithin. The problem with this approach is that it will only work with the dvi output, throwing multiple errors in pdflatex.

The better (but still bad) equivalent approaches involve using the `verbdef` package [18] or the built-in `\chardef` command [19] . Pdflatex will compile with those. More info about them is on the page given above. Since these methods are less straightforward they should only be used if the preferred method fails due to the complexity of the text to be displayed. Note, also that either of these two alternatives will generate warnings if compiled to a pdf. Solution is given in Section 8.15.

The preferred method is useful when nothing super fancy is required on section headings. Try `\texttt{}` which sets a typewriter font. Works perfectly in sections, footnotes, etc even with special characters if the font encoding is setup properly (see Section 8.2). If something special is required, try the methods given above.

Also, if text is not formatted properly then the compiler will generate errors/warnings. See Section 14.6 for more information on this.

[17] As `shown here`

[18] Definition is `\verbdef\mybackslash+\+`, usage `\section{Displaying \mybackslash vec}`

[19] It has the syntax `\chardef\mybackslash=‘\\` where ‘ is a character below the tilde, and the first \ is used as an escape character for the second \ we are defining. Use the backslash as usual, i.e. `\section{Displaying \mybackslash vec}` will produce the heading 'Displaying \vec'.

8.7 Breaks: spaces, words, lines, paras, pages

• There are several measuring systems for determining the size of the break they are described in Figure 3.

`mm`	millimetre $\approx 1/25$ inch	‖
`cm`	centimetre = 10 mm	⌊⌋
`in`	inch = 25.4 mm	⌊——⌋
`pt`	point $\approx 1/72$ inch $\approx \frac{1}{3}$ mm	‖
`em`	approx width of an 'M' in the current font	⌊⌋
`ex`	approx height of an 'x' in the current font	⌊⌋

Figure 3: LaTeX units

• \\ Starts a new paragraph with the new paragraph not being indented so for normal paragraph breaks use `\par` instead! **WARNING**: *Do not use to increase spacing between paragraphs, only to start another paragraph without indentation when the source code is rather kept without empty lines. Instead, use* `\vspace` *as explained below.* More information in section 14.2.

• `\\*` start a new line but not a new paragraph. The above warning is applied.

• `\linebreak` allow to break the line here.

• `\newline` request a new line.

• `\par` Creates a new paragraph with proper paragraph indentation. Use whenever possible instead of \\

• `\bigskip` Another alternative to the above. I think it skips 1 line of text in the current font size. For 1/2 line use `\smallskip`. The following paragraph begins unindented so use with `\par` for correct indentation. Alternatively force or suppress indentation as described in section 8.11.

• `\vspace{...}` is to insert vertical spacing between lines or paragraphs. Specifically, my convention for it is to define a new command `\newcommand\p{\vspace{3mm}}` in the preamble and insert the space whenever required by inserting `\p` in the text. Probably only good for home made documents as scientific publications are not supposed to have any spacing between paragraphs and lines. Notice, since `\p` is just a vertical line marker, an empty line must follow `\p`, otherwise it will not be inserted correctly. If an empty line does not follow `\p` then terminate `\p` with a \\. Careful about it anyway! More information is in section 14.2. Also see another (better) alternatives above.

• `\-` it's ok to hyphenate a word here.

• `\clearpage` plush all material and start a new page.

• `\cleardoublepage` flush all material and start a new page, start new odd numbered page.

• `\hyphenation` enter a sequence of exceptional hyphenations.

• `\newpage` request a new page.

- `\nolinebreak` no line break should happen here.
- `\nopagebreak` no page break should happen here.
- `\pagebreak` encourage page break.

8.8 Splitting text into columns

Usually helpful when working with exercises whenever a question, solution and explanation should go side by side.

Set up procedures. First, add the `parallel` package, then add the following command to the preamble[20]:

```
\newcommand\LR[2]{
  \usefont{T1}{cmss}{m}{sl}
  \ParallelLText{\noindent#1}
  \ParallelRText{\noindent#2}
  \ParallelPar\normalfont}
```

Now it can be used by adding the following code. Note the use of `[v]` as a flag for drawing a vertical line between the columns:

```
\begin{Parallel}[v]{}{}
  \LR{Text: left column part goes here}
     {Text: right column part goes here}
      Text: description goes here, it can be very very very long
\end{Parallel}
```

clearly works:

Text: left column part goes here | *Text: right column part goes here*
Text: description goes here, it can be very very very long

8.9 Tabbing

WARNING: this section is obsolete. The environment `tabular` needs to be used instead. Consult Mittelbach and Goossens [6] or the LaTeX cheetsheet for details.

One of the most convenient ways of using the tabbing environment is to declare a new command which specifies tabbing parameters in the preamble. One option could be

`\newcommand{\tab}{\hspace*{25.mm} \= \hspace*{25.mm} \kill }`

with as many `\hspace*{25.mm} \=` as required for each tabbing position. Then, using tabs is pretty straightforward:

```
\begin{tabbing} \tab
  column one \> column two \> column three \> etc. \\
  dates \> job title \> employer \> location
\end{tabbing}
```

[20] Using the package as a command is just one of the options. It seems to be the most convenient at the moment, however, if it lacks in flexibility explore other options with the package documentation or the printed material [6]

produces the following output:

```
column one  column two  column three  etc.
dates       job title   employer      location
```

Notice extra space added before and after the tabbing environment. At this stage I dont know any clean way of removing it except for specifying the negative `\vspace`.

Another alternative is to use the `Verbatim` env.

8.10 Stretched space: horizontal fill

To produce stretched space as in the example below use `\hfill`, `\dotfill` and `\hrulefill`:

Here's a stretched space
Here are two equal ones
Here'are a ..... dotted and _____ rulefill ones

8.11 Paragraph indentation

`\parindent 3pt` - changes paragraph indentation to 3pt. Can be either in the preamble or anywhere in the text
`\noindent` - suppresses it. Must be used in the beginning of the paragraph
`\indent` - creates paragraph indentation where it would otherwise be suppressed

8.12 Paragraph text alignment: justified vs rugged

Although it's a better looking presentation to have a justified alignment of text in most of documents, in some cases, such as very narrow page, we don't want LaTeXto break words and spread words evenly on the lines. In that case use `\raggedright` command in the preamble.

8.13 Lists: customisations and options

changing vertical spacing between items

The standard `itemize` or `enumerate` environmentswill produce vertical spacing larger than the one in the current text. To make them more compact use either `compactitem` for bullet lists or `compactenum` for numbered lists (requires the `paralist` package).

The `paralist` package environments are highly customizable. Specifically, the user can set up different defaults for vertical (and horizontal if required) spacing. For example the following command could be used either in the preamble or immediately before the list to set up a list with a customized spacing between individual items:

```
\setlength{\plitemsep}{.3ex}
```

where the usual default spacing is 0.

changing horizontal alignment of items

Another customization option is to stipulate indentation boundaries for sublists with the following command

`\setdefaultleftmargin{}{}{5ex}{}{}{}`

which sets the third nesting list indentation to 5ex and leaves the rest of them as set in defaults. This option can be set in the preamble as well as in the text as required.

changing standard bullet sign

Optional parameter is allowed to stipulate the style of bullets/numbering. A number of options are available, e.g. `\begin{compactitem}[$\star$]` will produce a star instead of a bullet point. Another option is `[i)]` which if used with `compactenum` will use roman small letters.

customising numbering to roman, numerical etc

If the entries are not gonna be referenced in the document this is accomplished by the command `\renewcommand\theenumi{\alph{enumi}}` where `<X>` in `enum<X>` will be one of i, ii, iii, iv and so on for various levels as required. Do not forget to surround it with the scope so that the changes are not pervasive for the entire document.

If referencing of list items is required then it pays to read pp.129–130 of the textbook [6] since an extra command is required if referencing is to appear correctly.

items with bold word heading for each entry

Another frequently used type of lists is to create lists with the `description` environment of the same compact nature using `compactdesc`. Good for definitions and so on. Also try `\paragraph{Heading goes here}` to produce the same result.

8.14 Setting up text width manually

Usually done in the `tabular` environment with the `\textwidth` command, however, could be set up in any place in the text. See Section 9.10 for an example.

8.15 Section headings with formulas, commands etc: $\sum$

If producing a pdf file with the `hyperref` package, one needs to recognize that a generated pdf file will contain pdf bookmarks corresponding to the section tree in the Latex project. Those bookmarks will have exactly the same text as section headings, but unfortunately no special formatting headings may contain is allowed for them. Hence if headings contain maths, commands (e.g. see Section 8.6) and so on, the pdf output will generate a lot of warnings. The solution is to use the `\texorpdfstring{$\sum$}{sum}` command whereas the first argument is the actual heading formatting, and the second one is its replacement plain text for its bookmark. The dvi output will simply ignore irrelevant information.

Notice, that basic font formatting with `\texttt{}`, for example, does not require special handling for pdf. However, commands like used by `\verbdef` or `\chardef` will require it (more in Section 8.6).

8.16 Section headings with frames or shades

Could be accomplished with the following code:

```
\fcolorbox{black}{lgrey}{
  \begin{minipage}[b]{.96\textwidth}{
    \textbf{someText\vphantom{p\^{E}}}
  }
  \end{minipage}
}
```

will produce the following output:

someText

Note, that this will not generate any numbering nor an entry in the contents page.

8.17 Accented characters and whitespaces in LaTeX and BibTeX

Sometimes it is required to display accented characters such as the german character umlaut ö. There are 3 ways of doing this. Read this in conjunction with Section 6.2 which describes generating bibliographies with these chars.

The preferred way is also most widely accepted when accented characters (not the white-space though) are represented with an escape sequence.

Description	Command
Backwards (grave) accent	\‘{x}
Forward (acute) accent	\’{x}
Caret symbol	\^{x}
Dieresis (umlaut)	\"{x}
Tilde	\~{x}

Still the complication with it is that the umlaut for example will have a problem with BIBLIOSCAPE and BibTex files since it has a double quote in it. The problem arises if double quotes (rather than braces) are used to delimit field values in BIBLIOSCAPE entries, and as a result the double quote in the umlaut is erroneously recognized as the end of the field value. However, this can be easily fixed as described in Section 6.2.

The second possible way which is robust across various platforms is to deploy the utf8 support. Add the following code to the preamble:

```
\usepackage[utf8]{inputenc}
\usepackage{textcomp}
\par\inputencoding{latin1}
```

Althought the complete syntax above is long and it uses the `utf8` file properly, a much shorter variation `\usepackage[latin1]{inputenc}` in the preamble is just as good if only the extended ASCII part of utf8 is required. Moreover it is included in the standard MikTex distribution. The `latin1` argument specifies which encoding for the extended ASCII to be used, in this case it will be ISO 8859-1. Then characters can be generated by simply specifying their ASCII number as, for example `^^f6` which will produce ö. A whitespace can be encoded similarly. Note that `f` (or other letters) must be of the lower case. The same approach would apply to BibTeX in which all characters must be encoded right in BIBLIOSCAPE. Note, that this encoding cannot be used with first authors as a bibtex key will be invalid if it contains the character `^`. See Figure 4 for the complete extended ASCII mapping of ISO 8859-1 (lower number in hex).

The third approach is less desirable as it is likely not to be robust when moving to a non latin locale. Basically, one can try to access the extended part of the ASCII by using ALT+148 for the same character ö (note, it does not work in TEXTPAD and TEXNICCENTER but does work in BIBLIOSCAPE, NOTEPAD and WORD). Refer Figure 4 for other characters (upper number in decimal).

8.18 `lipsum`: Lorem Ipsum or Filler Text

Placeholder text (filler text) is often used to demonstrate the graphic elements of a document or visual presentation, such as font, typography, and layout. The lorem ipsum text, which is typically a nonsensical list of semi-Latin words, is a hacked version of a Latin text by Cicero, with words/letters omitted and others inserted, but not proper Latin. Just import package `lipsum`, and then insert text with `\lipsum`.

8.19 Math Mode `bbm` Fonts (e.g. `mathbbm`) are not Rendered in `pdf`

With certain fonts used in the math mode, e.g. `mathbbmm`, the `dvi`(latex) output produces correct rendering, but if it is further compiled from `dvi` to `ps`(dvips) to `pdf`(ps2pdf), the characters are either not displayed or displayed garbled.

The problem occurs when `dvi` is converted to `ps` using `dvips` as the default TeXnicCenter configuration specifies the argument `-P pdf "%Bm.dvi"` in the `Latex=>PS=>PDF` profile. To fix it, remove the flag `-P pdf`.

9 Figures, floats, tables etc

9.1 Common Errors with Figures, Floats, Tables etc.

- In Tables never use \\, instead use only `\tabularnewline`

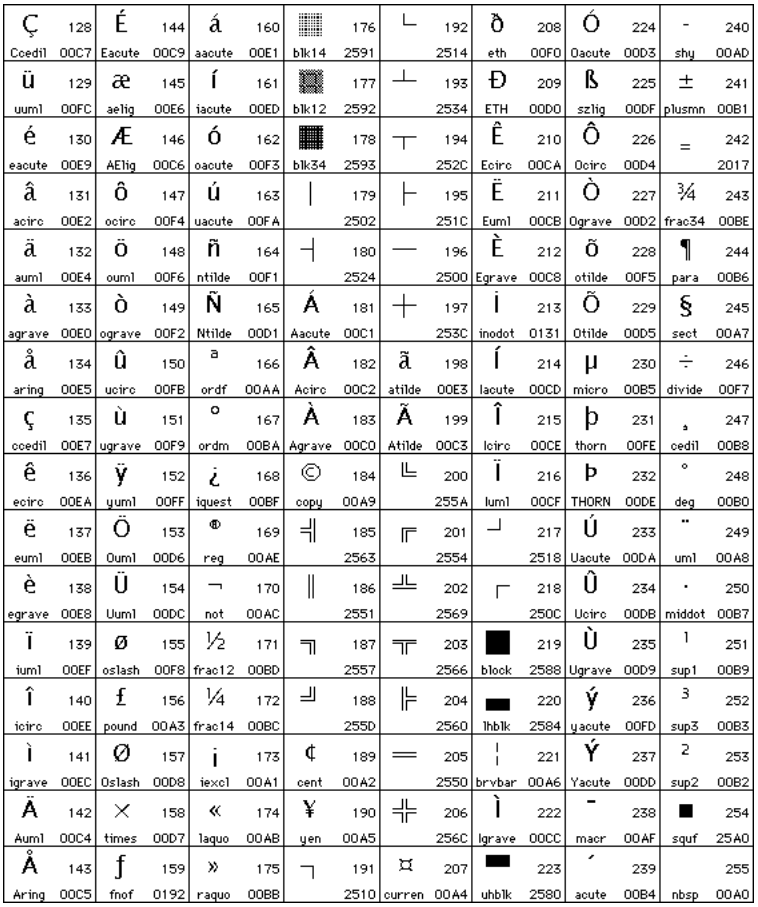


Figure 4: IBM 850 (Windows) and ISO 8859-1 (TeX latin1) extended ASCII characters sets. The upper number corresponds to a decimal number which can be entered with the ALT button (IBM 850). The lower number is the hex number of a character accessible with the hex shortcut (ISO 8859-1). See Section 8.17.

9.2 A simple figure

Makes the one with the left alignment:

```
\begin{figure}\flushleft \includegraphics{spirit}\end{figure}
```

9.3 A simple figure with a frame/box

Makes the one with a frame:

```
\begin{figure}
   \fbox{\includegraphics{spirit}}
\end{figure}
```

9.4 Available Packages to Display Algorithms

Two options: `algorithm2e` and `algorithmic`. Bear in mind that IEEE publications require `algorithmic`.

9.5 Placement specifiers with figures, tables, floats etc

These come in square brackets after the env declaration: `h`,`t`,`b`,`p`, or `!`. They are self explanatory with `p` being a special page at the end of the document, and `!` if put in front of the list meaning "do it even if it does not look that good". Note that the default placement specifier is `[tbp]`, and than `[!h]` will throw a warning and will be automatically replaced with `[!ht]`. Use non-default specifiers with caution.

Also if using subfloats, make sure that placement specifiers for those are correctly used. See Section 9.7.

Also, LaTeXcomes without an option to place floats at the bottom of a page. The way to override this is to use `stfloats` which introduces placement marker `b` so that `\begin{figure}[!b]` becomes possible. In addition you might want to add `\fnbelowfloat` from it to make sure that footnote appear below floats. The package is quite "dirty" but is accepted by IEEE. Check if placing floats at the bottom is acceptable by other publishers.

9.6 Incorrect numbering of references to figures and tables

In `figure` and `table` environments all floats must have `\label{}` **after** `\caption{}` commands for references to figures and tables to be inserted correctly. It is probably a good idea to scan through the entire printed document before completing its final draft to make sure that all references to tables and figures are correct. Another alternative would be to search through the source code and checking that every `\label{}` is preceded by `\caption{}`. If this requirement is not met, then referencing returns either an empty number, or a number corresponding to the lowest header `\ref{}` is currently in.

9.7 Vertical alignment of inhomogeneous sub-floats

If heights of individual subfloats in the `subfig` environment are different, then it's possible that more than one subfloat in one line would not be aligned as wanted. Specifically, if subfloats are built using `tabular`, `array` or `minipage` environments they will be centered by default. However, if they need to be aligned at the bottom by setting optional arguments `[t]` or `[b]` inside *those environments* and not in `figure` or `subfloat` declarations. Below is an example of including the argument with tabular subfloats:

```
\subfloat[Table 2]{\fbox{
  \begin{tabular}[b]{>{\bfseries}c|l}
      ...
```

For more information on positioning of floats see Chapter 5 of the `subfig` documentation file.

9.8 A simple table

```
\begin{table}[hbtp]
\centering
\caption{An example Table}
\begin{tabular}{||c|l|r|r||} \hline \hline
```

```
Title one & Title Two        & Title Three & Title Four \\ \hline \hline
equations ok & $ \alpha^3 $ & $ G_1^{3 +q} $ & $ \int_0^T x^2 $ \\ \hline
text ok & hello, & world & \\
numbers ok too & 123.45 & 567.89 & 123456 \\ \hline \hline
\end{tabular}
\label{tab:def}
\end{table}
```

Table 1: An example Table

Title one	Title Two	Title Three	Title Four
equations ok	α^3	G_1^{3+q}	$\int_0^T x^2$
text ok	hello,	world	
numbers ok too	123.45	567.89	123456

9.9 Another simple table

```
\begin{tabular}{lll}
        {\color{green}Regular Fonts}\\
        \hline
        \usefont{T1}{cmr}{m}{n}\fox& cmr& Roman\\
        \usefont{T1}{ptm}{m}{n}\fox& ptm& Times\\
        \hline
        \usefont{T1}{pzc}{m}{it}\fox& pzc& Zapf(it only)
\end{tabular}
```

Regular Fonts		
The quick brwn fx jmps ovrlazydg 123!@*+,/?>([	cmr	Roman
The quick brwn fx jmps ovrlazydg 123!@*+,/?>([	ptm	Times
The quick brwn fx jmps ovrlazydg 123!@+,/?>([*	pzc	Zapf(it only)

9.10 Tables with customizable width

Sometimes we need to setup a table with a specific width (I suppose it means that the width will be larger than the one set by default by LaTeX). In that case the `tabular*` environment needs to be used. For example:

```
\begin{tabular*}{0.5\textwidth}{l@{\extracolsep{\fill}}r}
  someText & someText \\
  someText & someText \\
\end{tabular*}
```

has the `{0.5\textwidth}` part which makes the table that 0.5 wide,
and the `{l@{\extracolsep{\fill}}r}` argument which inserts the rubber space between the two columns with `@{\extracolsep{\fill}}` command. The following table would be produced:

someText	someText
someText	someText

9.11 Multiple subfigures

```
\begin{figure}[ht]
\subfloat[Width Classes]{\fbox{
  \label{class:ex1}
  \begin{tabular}[b]{>{\bfseries}c|l}
    \textcolor{green}{\bfseries \#} & \textcolor{green}{\bfseries Width}\\\hline
    uc&  Ultra Condensed\\
    ec&  Extra Condensed\\
  \end{tabular}
}}\quad
\subfloat[Shape Classes]{\fbox{
  \label{class:ex2}
  \begin{tabular}[b]{>{\bfseries}c|l}
    \textcolor{green}{\bfseries \#} & \textcolor{green}{\bfseries Shape}\\\hline
    n& Normal\\
  \end{tabular}
}}
\captionsetup{singlelinecheck=false}
\caption{Abbreviations used for describing various classes of fonts}
\label{fig:ex3}
\end{figure}
```

#	Width
uc	Ultra Condensed
ec	Extra Condensed

(a) Width Classes

#	Shape
n	Normal

(b) Shape Classes

Figure 5: Abbreviations used for describing various classes of fonts

9.12 Left justifying captions in figures

In the `subfig` package there's a boolean option `singlelinecheck` which causes a caption that fit on one line to be centered below the figure (by default). Set it up to `false` makes a one line caption to be left justified. If a caption is multi-line it is left justified by default.
Usage: `\captionsetup{singlelinecheck=false}`^{subfig}
either in the `figure` environment or in the document preamble.

9.13 Formatting `subfig` labels/captions

The default label/caption formatting in subfigures includes their parenthesised numbering, e.g. (a) Text. If desired those can be formatted to remove their numbering. Add the command

```
\captionsetup[subfloat]{labelformat=empty}
```

before the float in which you want to suppress numbering. To revert to the default formatting repeat it again with `labelformat=parens`.

9.14 Caption wraps too short for graphics length with PDF

If using `jpeg` graphics in figures and subfigures multiline caption text can be far too short in length as compared to the width of its image. This is caused by LaTeX not being able to calculate bounding boxes corretly for `jpeg` files.

best solution solves the bounding box problem without any extra work such as calculating `bb` files and is described in Section 10.1. Generally, if the system is configured properly, there should not be any problem which is described here.

bad solution is to use a program (such as `ebb`) which calculates bounding boxes and save them as `*.bb` in the same location with graphics file. Another bad solution is to convert binary information of `jpeg` to `eps` with arbitrary tools which results in significant loss of quality.

bad solution is to use program `jpeg2ps.exe` which leaves binary information intact but wraps `jpeg` in the metadata describing its bounding boxes correctly. One of the implemented ms-dos scripts does it automatically. The problem with this approach is problematic as with other arbitrary tools. See Section 10.1 for details.

9.15 Want to wrap text around floats

This can be accomplished in non-list environmnets using one of `wrapfig`, `floatflt` etc. However, it's more like a hack and hence to be avoided regardless what environment it is in. I tried it and it looked bad. However, there's another package, `picinpar`, which allows to do this and is described in the LaTeX textbook [6] on pp.108–109. I have not tried it though, and again it does not work inside any lists or enumerations apparently.

Generally, scientific articles do not have a wrap in place at all. Plus, wrapped floats might look pretty bad anyway.

So the solution is if we want to include numerous small graphics or tables that are too narrow to reserve entire rows, we could use subfigures and put them to footnotes such as shown in Figure 2 for example.

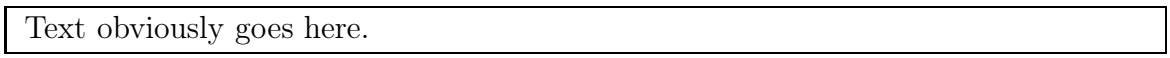
9.16 Changing default `Figure` to arbitrary name in figure captions

If `Display` is required instead - `\renewcommand{\figurename}{Display}`. And of course once it is no longer needed rename it back.

9.17 Inserting boxed displays into main text

The following float:
can be created with the package `color` and the code:

```
\renewcommand{\figurename}{Display}
\begin{figure}[!ht]
\begin{center}
```



Display 6: Example of Boxed Display

```
\fcolorbox{black}{white}{
   \begin{minipage}[b]{.9\textwidth}{
   Text obviously goes here.
   }
   \end{minipage}
}\end{center}
\caption{Example of Boxed Display}
\end{figure}
\renewcommand{\figurename}{Figure}
```

9.18 Displaying Figures in Footnotes

The normal approach with `figure` env will not work. Instead just add `\includegraphics{}` which will work, but I have not tried options for formatting.

9.19 Excel to LaTeX

There are several packages on CTAN. I tried `xl2latex`, but it does not support formatting of `multirow`, it does do colouring correctly though. It does automatic saving to a `tex` file which with the help of `\input` makes the workflow more elegant. The other method below requires handling clipboards.

A better choice is `xls2latex` which is not on CTAN, but on a Perdue Uni website (somebody's personal development)[21]. It's handy as it supports `multirow` and seems to be more adept in formatting tables. It does not deal with colours at all, and as the other package it cannot handle per cell alignment, only per column alignment. It does not seem to be able to save the file automatically as a tex, requires a workaround with copying go clipboards etc. The solution could be to merge both macros together to combine their functionalities as it suits me. Additionally it has options of importing ASCII data feed (like `csv` data) from the shortcut directly into the excel. I am not sure this is the best workflow, but is probably a good solution for certain situation when Excel struggles to pick up data correctly and put it into the right columns. It is saved in the local repo.

10 Graphics

10.1 Projects with both bitmap and postscript graphics

The problem is intrinsic to the current LaTeX version in which its dvi compiler (`latex`) requires exclusively `eps` graphics, while its pdf compiler (`pdflatex`) requires any of `pdf`, `jpg`, `png`, or `tif`.

[21] http://cobweb.ecn.purdue.edu/~zhang97/xls2latex/

best solution Differentiate between two disparate categories of *source* graphics files, bitmaps and postscript, and to store them in disparate sub-folders of a latex development project. Before graphics can be used, a script needs to be run converting all *source* files to their *derivative* counterparts (all original bitmap graphics to `eps`, and all original `eps` graphics to `pdf`). Once it's done all originals and derivatives are copied to the `graphics\img-src` folder as outlined in Sections 4.1 and 4.9 so each original source file appears in both possible formats in one flat resulting structure available to both `latex` and `pdflatex`.

bad solution As one of feasible workarounds the user can always compile to `ps` and then to `pdf`, but all choices come with a price (see Section (4.10)).

bad solution Using `jpeg2ps` does not help either as the tool cannot handle other non-`jpg` file formats and `eps` files it generates are quite unpredictable in size as the tool often fails to calculate bounding boxes correctly.

The adopted maintenance strategy is:

1. Copy original bitmaps to `graphics\bitmaps\` and original `eps` files to `graphics\eps` folders. Note that subfolders here are allowed to an arbitrary degree of branching/depth. Also note, that filenames (without extensions) of all source graphics must be unique or otherwise only the first encountered copy of the file will be used for generating of its counterpart.

2. Run script `mergeGraphics.bat` from the project root. It takes files in source subtrees `graphics\bitmaps\` and `graphics\eps\`, creates their copies in their compliment format, and then copies both of the versions to the `graphics\img-src` folder. Note there are two versions of the script, one which overwrites all existing files in `graphics\img-src` (slow if there is lots of graphics, but the only one which would work with `\3rdparty` files unless of course it is not required as with `laprint`), and one which only creates/copies files from the source if they do not exist in `graphics\img-src` (faster but potentially some graphics can be omitted).

10.2 `png` or `jpeg`?

Among raster graphics, `png` is a lossless highly compressible format which should be used for all scientific graphics with lines, charts etc (except pictures). `Jpg` (and `tif`) will necessarily degrade the quality of any fine graphics and should be avoided for scientific texts. Note, though that even `png` graphics more likely than not to be avoided and to be replaced by postscript graphics. An example of what can go wrong even with lossless `png` graphics see Figure (7a).

10.3 Bitmap or `eps`?

Clearly with the exception of photographic material where postscript will be a bad choice, most of, especially, drawn or automatically generated graphs, charts and figures must come as `eps` imports.

Similarly when dealing with MATLAB graphics, bitmap output should be avoided as well (e.g. printing from monitor screen; Figure (7a) demonstrates the unsatisfactory results of

this approach.). However, the problem with generating high quality graphics in MATLAB is a bit more complicated than just choosing between bitmap or raster graphics and is discussed in more detail in Section 10.8. The examples of graphics of various quality produced by different methods can be seen in Figure (7).

10.4 Resolution issue when converting from bitmaps to `eps`

The problem with `eps` format is that there's no strict consistency in its file specification which confuses (probably) every editor/viewer in the market. Part of the confusion is that the `eps` file header contains various sections such as Bounding Box, or Width/Height, or Resolution, or Dimensions which do not have to be consistent with each other. So in many cases if they are inconsistent the editor/viewer makes up its own decision with respect to which one to use for display purposes. The key here (which is rarely mentioned in literature) is `eps`'s native 72 dpi resolution, meaning that if you provide any other arbitrary resolution (or leave the resolution for an editor to decide) you are going to end up with dimensions of the generated `eps` file different from its source bitmap. For example, if you are converting a 200x200 `jpg` image to `eps` and specify say 100 dpi resolution as a command line argument (e.g. with `jpeg2ps`, or, e.g. with `pnmtops` command of Section 10.5) this will result in the bounding box specified in the header being 72/100 less then 200x200. The resulting image will almost certainly be displayed smaller as well. If you omit a resolution parameter altogether, the editor will pick up some random number which might be completely unpredictable.

10.5 Bitmap (`pdf`) <–> `eps` conversions

Choice of utility for bitmap to EPS conversions This step was automated by the batch script (see Section 4.8). This can be a really hard problem as many of them behave in a really strange way. For example, `jpeg2ps` adds 20 pixel boundary to the graphics whether you want it or not, while `convert` of IMAGEMAGICK might generate `eps` worse in quality then its original. The current choice is a UNIX package called NETPBM which has a working clone for Windows. After installation, a number of individual commands listed in its `\bin` folder can be used basically allowing to convert from any format to any format. For example the following commands could be used:

(`pnm` to `png`) `pnmtopng i.pnm > i.png`
(`pnm` to `eps`) `pnmtops -nocenter -equalpixels -dpi 72 i.pnm -noturn -rle > i.eps`
(`jpg` to `eps`) `jpgtopnm i.jpg | pnmtops -nocenter -equalpixels -dpi 72 -noturn -rle - > i.eps`

The first line is rather straightforward since conversion between those two types can be executed with just one command and no extra information is required by the editor to generate a `png` file. For the second line, it's also a one-command conversion, but since we convert to `eps`, extra info must be passed which appears as flags. Note the position of flags and filenames. In the third command we have to first uncompress `jpg` to lossless `pnm` (as required by NETPBM) and only then pipe it to the same `pnmtops` command shown on the second line. Note here that the rest of the syntax is almost identical.

Other type conversions to `eps` are possible if `jpgtopnm` is replaced with another appropriate command. For LaTeX we might need `tiftopnm` and `pngtopnm`.

Choice of utility for EPS to PDF conversion This step was automated by the script. See Section 4.8. There is an easy choice since the excellent application comes with the MikTex distribution, `epstopdf`. The utility is located in `\texmf\miktex\bin\` folder and can be easily envoked with any `eps` file as:

```
epstopdf i.eps
```

Additional information is available at http://mintaka.sdsu.edu/GF/bibliog/latex, where a host of helpful documents is located. Specifically the following pages are useful:

- `PSconv.html` - describes inherent problems when converting to `eps`, and introduces command syntax for possible successful solutions
- `LaTeXtoPDF.html` - discusses dvi/pdf production from source LaTex files and various strategies to handling it
- `figures.html` - gives more information about various graphics formats, and discusses `pbm` family of formats (which include `pgm`, `ppm`, and `pnm`) in more detail.
- other files at the same location can also be useful as background reading on the issue.

10.6 EPS files not recognized as graphics when outputing to PDF

The problem: when TeXnicCenter's output profile LaTeX=>PDF is selected, it rejects all `eps` graphics. So while there are no mistakes compiling to `dvi`, there will be mistakes compiling to `pdf` with `pdflatex`. This happens because different compilers use different file types.

Solution: see Section 10.1.

10.7 Save MATLAB Figures as Graphics

There are four ways to save Matlab figures exhibited in Figure 7.

best solution is described in Section 10.8

baddish alternative is `Save as` in Matlab with `eps`

baddish alternative is ot use package `savefig` available from Matlab Central[22] can output MATLAB graphics as `eps`. However, I discontinued its use. Example:

```
savefig('filename-without-ext', 'eps')
```

The exact equivalent of this is to use the following Matlab command once a figure has been created:

[22] http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=10889

```
print -deps2c dhist
```

which saves the currently in focus figure as `dhist.eps` file. See example of it in Fig. 7b. Note, that general quality in either case is comparable, but LaTeX strings are displayed a way better with the `print` command. So if possible avoid `savefig`. Also even with `print` note the limitation of Section 10.9.

However, bear in mind that the method explained in Section 10.8 produces far more superior overall quality of drawings (solid sharper graphics + support for LaTeX symbols - `print` is less flexible, eg with axis labellng etc). You might try to use it to generate all MATLAB graphics.

bad printscreen the picture and save as `png`

bad, not tested Another alternative is to use the package `pgfplots` in conjunction with MATLAB script `matlab2pgfplots`. According to the discussion in the latex newsgroup, this gives much more control and quality than `laprint`. However, two major disadvantages is time cost and its inability to generate 3D graphics.

10.8 Matlab and Latex With `laprint`

The method outlined here produces far more superior quality of MATLAB generated graphics then even the `Save as` option with `eps` in MATLAB or the method of Section 10.9. I would recommend using it every time some matlab drawings/graphics need to be generated [23] . Major disadvantage of `laprint` is that it is dependent on `psfrag` and hence cannot be used with `pdflatex`. More details are in this section plus in Section 4.10.

Matlab code to generate graphics from Matlab add-on `laprint` must be stored in directory `./matlab-laprint/` (see Section 4.12). The files which are generated by this Matlab code in that folder (for example, `tex` and `eps` files generated by `laprint`) should be automatically saved in the dedicated `./graphics/3rdparty/`.

Matlab can produce LaTeX compatible graphs with annotations formatted with full support for TeX. Basically, one needs `laprint.m` which is freely available on the mathworks website, and packages `graphics, color, psfrag` for it to work.

A sample matlab code:

```
set(0,'defaulttextinterpreter','none')
thirdParty = '..\graphics\3rdparty\';
fileName = 'matlaboutput';
theta = -pi:.1:pi;
y = sin(theta)./cos(theta);
plot(theta,y)
xlabel('$\theta$'); ylabel('$y$')
text(-1,20,'\LaTeX')
set(gca,'XTick',-pi:pi/2:pi)
set(gca,'XTickLabel',{'$-\pi$','$-\frac{\pi}{2}$','0', ...
```

[23] Another (bad) alternative is to use package `pgfplots` in conjunction with the `matlab2pgfplots` matlab script. According to a discussion in a latex newsgroup, this gives much more control and quality than even `laprint`. However, two major disadvantages is high time cost and its inability to generate 3D graphics.

```
                    '$\frac{\pi}{2}$','$\pi$'})
title('$y(\theta) = \frac{\sin \theta}{\cos \theta}$')
laprint(1,[thirdParty fileName],'width',5.5,'factor',.45) % Use LaPrint to save figure
```

will produce two files: `matlaboutput.eps` and `matlaboutput.tex`. Note the first line in the code above is compulsory for laprint to produce correct output. The next step is simply to copy the two generated files to the graphics folder for 3-rd party produced illustrations as described in Section 4.9. Finally, if the name of the `tex` file generated by laprint is `matlaboutput.tex` then one line `\input{../graphics/3rdparty/matlaboutput.tex}` is to be added to the current latex source file where the graphics is to appear, i.e. inside the `figure` environment.

Also to save your time by automatically copying `eps` files from `3rdparty` to `img-src` as described Section 4.9, the following line of code needs to be added to the MATLAB code above:

```
uploadEpsToImgSrc(fileName);
clear fileName;
```

The width of graphics in cm can be specified directly in the matlab code with the property `'width'` as shown above.

Note, however, that `psfrag` cannot be used with `pdflatex` as the latter does not accept `eps` imports, so only the `latex` compiler in MikTeX can be used. There is a number of problems resulting from being unable to use `pdflatex`. For more info see Section 4.10.

The advantage of this method is that the drawings produced with `psfrag` will be of much higher quality then the ones saved even as `eps` or `png`.

10.9 Displaying LaTeX when annotating MATLAB figures

When producing MATLAB figures, sometimes it's nice to have titles, axis labels etc to be labeled with LaTeX characters. This is to be done in your Matlab code of course. Just add the interpreter to the command used and surround the LaTeXexpression used in dollar signs:

```
title('$\mathcal{S}$','interpreter', 'latex')
```

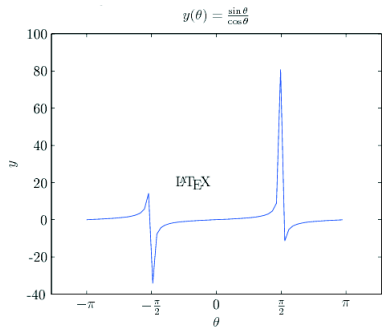
Alternatively, the interpreter can be configured only once for entire the Matlab project:

```
set(0,'defaulttextinterpreter','latex')
```

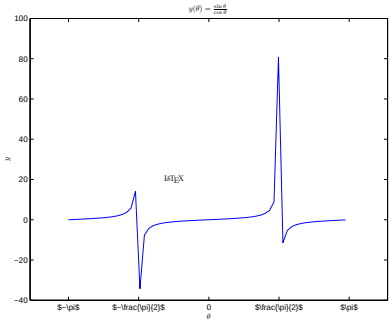
the only problem with this is that tick labels on axes etc are not interpreted in the same way as other text in matlab figures, e.g. π on the x axis in Figure 7b will be displayed as plain text with dollar signs. So the best way of handling it is to implement the approach of Section 10.8, i.e. avoid this current solution completely if possible (but it comes with its own issues). There is a solution (a matlab addon) available elsewhere though[24].

Another solution was proposed in a Matlab Central thread which is another version of a more general solution of Section 10.8. Apparently it's possible to put placeholders into `eps`

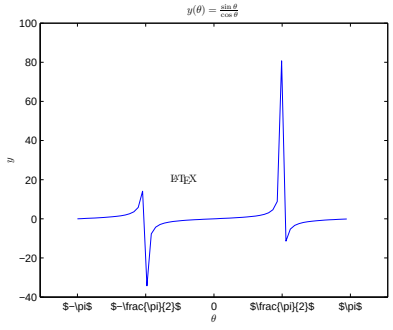
[24] A package called "Format Tick Labels" by Alexander Hayes is publicly available on Matlab Central: http://www.mathworks.com/matlabcentral/fileexchange/15986. I have not tried it but reviews are good.



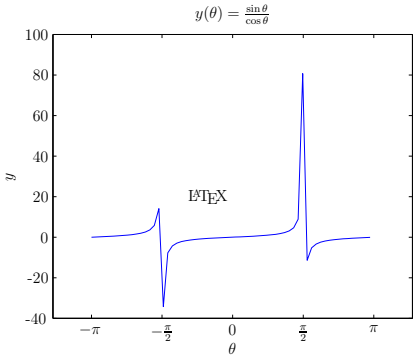
(a) Print screen from Section 10.3



(b) `savefig` or `print` to file from Section 10.7



(c) Using `Save as eps file` in Matlab



(d) Output from `laprint` with package `psfrag` and `latex`, using `\input`

Figure 7: Quality of Matlab graphics produced with various workflow options. Notice remarkably poor quality of graphs produced with lossless bitmap in (7a). Also, for `eps` graphics `print` and `Save as` will not support LaTeX in certain placeholders such as axes labels (7b),(7c). There's also less flexibility specifying placeholder's fontsize and line weight, plus some problem with latex symbols in axis labels (see Section 10.9. All of the above problems are resolved with `laprint` (7d). Note, however, `laprint` will not work with `pdflatex`.

produced by Matlab and then substitute their plain text labels for LaTeX code right in latex code[25]. Two problems with it. First, it's more laborious/cumbersome (i.e. more coding to be done and it's done in latex rather then in matlab). Finally, it will fail to work in both profiles as only `eps` can be supported (for more info see Section 10.1).

10.10 Importing hand made electronic drawings as graphics

Problem: a drawing made in MS Word (MS PPT) needs to be cropped and saved as an eps image file to be used by latex. Highest possible quality of reproduction is required.

preferred solution INKSCAPE, see Sections 4.4 and 10.11. Also investigate there's plenty of other choices on the fauskes website on page 7.

[25] http://www.mathworks.com/matlabcentral/newsreader/view_thread/251861

baddish solution Create a drawing in MS Word, MS PPT, MS Visio (for the largest toolbox of shapes) or any other application, then print it using PDFCreator to a pdf[26] file. Open the pdf in Adobe Acrobat, choose Document/Crop Pages to get rid of white margins and save as an eps file[27]. Done.

bad solution Draw everything in some desktop publishing application (e.g. CorelDraw, Illustrator), then export to eps. Disadvantages: no support for equations, special symbols plus very limited number of shapes to choose from.

bad solution Create a drawing in MS PPT, then save it as Windows Meta File (wmf), open with CorelDraw (or any other desktop publishing application) reposition if required and export to eps. A number of options could be changed at this stage, but the default ones work perfectly ok. Two major problems with it: the generated eps file is very large and there is some kind of the problem with texniccenter in which Latex→PDF profile does not see the file thus resulting in errors[28]. Other profiles are fine though.

bad solution Any intermediate importing to a graphics format (jpeg, png etc) destroys vector graphics thus degrading the quality when the scaling of the output pdf file from the latex platform is scaled to less than 100 percent.

10.11 Using Inkscape

This describes the workflow adopted for efficient graphics generation. Installation instructions are described in Section 4.4.

All files are stored according to Section 4.12. The root of Inkscape files must also contain the script file `_SvgToEps.bat` and the file `preamble.txt` with content required in the preamble of the otherwise equivalent standalone LaTeX code:

```
\usepackage{bm}
\usepackage{amsmath}
```

After `svg` graphics has been produced do not save it manually, instead run the script `_SvgToEps.bat` which converts all the files in the latex project Inkscape folder to `eps` and copies them over to `.\graphics\3rdparty\inkscape` where they are stored as `eps` source. It then converts them to `pdf`, moves `pdf`s and copies `epss` to `.\graphics\img-src` so no further graphics manipulation is required. Note that if `img-src` is emptied, it does not require the re-run of `_SvgToEps.bat` as `eps` files will be automatically picked up by `graphicsMerge` routines as described in Section 4.9.

[26] Any other output formats except pdf will result in the loss of quality

[27] if the pdf originates from MS Word drawing canvas, make sure that the canvas is tightly wrapped around the drawing as it will create a gentle outline in the final eps file

[28] the problem described in Section 10.6

10.12 Creating complex postscript graphics

The canonical solution is `pstricks` for which there is even a book writtern [2]. I stumbled on a highly praised package `TikZ/PGF` which has the best written manual and a website full of examples[29]. Looks impressive but also quite laborious.

11 Mathematics

11.1 Hard to Find Symbols Not in Symbols List

The only solution requiring package `amsfonts` which worked when I tried to produce the symbol ⩤ was:

```
\makeatletter
\newcommand*{\mathraise}[1]{%
\setlength\@tempdima{#1}\mathpalette\math@raise
}
\newcommand*{\math@raise}[2]{%
\setbox\z@\hbox{$\m@th#1{#2}$}\raise\@tempdima\box\z@
}
\makeatother
\newcommand*{\frowneqq}{%
\stackrel{\mathraise{2pt}{\smallfrown}}{\smash{=}}
}
```

And the usage is `\frowneqq` in math mode. Notice this is a standard way of creating a symbol out of individual components when it cannot be found in any LaTeX prefabricated packages.

I failed to make other suggested methods to work for me. One way to go is to try `inputenc`. Importing `\usepackage[utf8]{inputenc}`, while declarations could be either a single line `DeclareUnicodeCharacter{2014}{\mydash}` or robust:

```
\DeclareRobustCommand\dash{
  \unskip\nobreak\thinspace\textemdash\allowbreak\thinspace\ignorespaces}
```

The second suggested approach is to use package `XeTeX` which comes with MiKTeX. Again I did not succeed with it either.

11.2 Selecting environment for a particular task

LaTeX is full of various envs which come in the Math mode and in the standard `amsmath` package. The choices are abandant and confusing: align, aligned, falign, equation, case, etc etc. A very good reference book called "Math mode" [5] is available on CTAN which contains a large number of examples for pretty much every possible situation. Not only the envs are compared on an example by example basis, there is also a large section at the end which gives solutions to various formatting problems.

[29] Source http://www.texample.net/tikz/examples/

11.3 Common stylistic choices and errors with maths

Formatting of formulas Only special formatting commands are valid inside math environments. For example for bolding maths symbols use `\bm{}`, for roman text `\mathrm{}` or surround the text in `\mbox{}`. So the normal x_{max}, can be expressed as $x_{\max}$.

Mathematical and engineering rules for manuscripts Can be found here http://physics.nist.gov/cuu/Units/checklist.html.

11.4 Controlling Extensible Delimiters

Any extensible delimiters can be controlled with one of the 4 commands which provide for their fixed size: `big`, `Big`, `bigg`, and `Bigg`. They take a single parameter (that is a delimiter) and are used exactly the same was as `\left(`, e.g. `\bigg(`.

Now something like this is possible:

$$\left.\frac{ds}{di}\right|_i$$

11.5 Grouping equations with a vertically extensible delimiter

The following grouping:

$$\left.\begin{aligned}\mathcal{M}\mathbf{u} &= \mathbf{0}\\ \nabla\mathcal{M}^T\mathbf{u} &= \nabla Q\end{aligned}\right\} \quad \text{The KKT (partial) Conditions} \tag{1}$$

can be produced by

```
\begin{equation}
  \left.
  \begin{aligned}
    \mathcal{M}\textbf{u} = \textbf{0}\\
    \nabla\mathcal{M}^T\textbf{u} = \nabla Q
  \end{aligned}
  \right\rbrace
  \quad \text{The KKT (partial) Conditions}
\end{equation}
```

A few important remarks. We use a minipage environment `aligned` inside `equation` which allows us to group self-contained structures the way we wish. Moreover, the `aligned` part is delimited from the left and right by what is called *vertical extensions* which take as much space as required. On the right it takes the shape of a right brace `\rbrace`, on the left it, what is called, is a null delimiter, dot ".". Both delimiters must be preceded by either `\left` or `\right` respectively. For the full list of delimiters see p.498 of [6].

Its current unresolved limitation is inability to produce numbering of individual equations. It can produce numbering of the overall scope (which is set up default), if numbering to be avoided use the `\notag` command just before `\end{equation}`.

11.6 Equations not aligned but centered

Sometimes instead of alignment against a particular part of an equation, it's better to center various parts of equations to have it appear just like a table:

$$\begin{array}{ccccc} \Delta_{\min;x,x-1;y} & \le & h_{x;y} - h_{x-1;y} & \le & \Delta_{\max;x,x-1;y}; \\ \Delta_{\min} & \le & h_{x;y} & \le & Q-1 \end{array}$$

Notice the nice centered way the elements in the second line appear. The code

```
\begin{equation}
  \begin{array}{ccccc}
    \Delta_{\min;x,x-1;y}&\le &h_{x;y}-h_{x-1;y}&\le &\Delta_{\max;x,x-1;y}; \\
    \Delta_{\min} & \le & h_{x;y} & \le & Q-1
  \end{array}
  \label{eq:eq1}
\end{equation}
```

11.7 Formulas in running text exceed right column margin

It was pointed out that LaTeX has a pretty rigid system for determining the way how paragraphs are typeset. For more on this read the manual for the `microtype` package and Section (4.10) on some remarks regarding `microtype`. Especially it is a problem for the first line in the paragraph. So if you are unlucky to have a running text formula using $ separators on the line it may exceed its boundaries. Hence the `Overfull \hbox` warning is thrown.

Currently there is no solution for the `latex` compiler with output to `dvi` and `ps`, but for `pdftex` the following solves it pretty nicely: just declare the package `microtype` in the preamble[30]

So all proposed solutions include[31]:

- As above, it will just pack letters nicer so the formula will have more space on the line
- Probably the best solution is to rewrite the piece of text. Normally if LaTeX struggles with proper line breaks it means that it faces dilemma: either to exceed the line boundaries or have really ugly line with big breaks between individual words. If the above does not help it means (really) that rewrite is the only option. Even consider rewrite as the first solution if possible.
- Another solution (which will work with `latex`, i.e. dvi, is to use the `\tolerance=500` command which specifies how loosely LaTeX can typeset the paragraphs. The values could be somewhere between 50 to 9999, do not use 10000 though as very (ugly) loosely looking lines would appear. Again as above if 9999 does not work probably the only solution is to rewrite the text. Also be careful with some arbitrary assignments as it might screw the rest of your document.

[30] Note, this package has numerous dependancies which are not included in the distributive files. You will need the following packages to download and install beforehand: `letterspace`, `ifpdf`, `hypdoc`, `holtxdoc`, `booktabs`. If the above `sty` files are properly loaded then just run `pdftex.exe microtype.ins` first. When it's done run `pdftex.exe microtype.dtx` and ignore any warnings/errors. After this the code using `microtype` should run smoothly.

[31] For more background info check out pages 102-103 and 939 of [6]

- If you must suppress it surround the paragraph with `\begin{sloppypar}` environment, this pushes the formula to the next line. It's a hack anyway and if the block you are trying to push to the next line is too big then it will result in ugly lines. Still can try since it worked for me.
- Ignore under 3pt warning completely using `\hfuzz=3pt`, however when document is finished and you want to correct/rewrite it you might want to remove it and mark all bad boxes with a black box in the pdf document[32]
- Do not use the command `\sloppy` since it is a global parameter and it hacks the typesetting algorithm

11.8 Do not use eqnarray for displaying formulas

Instead use `align` from the `amsmath` package. It was pointed out to someone in a forum thread.

11.9 Matrix macro

This method will not for horizontal vectors though, just use squared brackets.

A more convenient method for typesetting matrices is to declare a command `\mat`:

```
\newcommand{\mat}[2][rrrrrrrrrrrrrrrrrrrrrrrrr]{
  \left[
    \begin{array}
      {#1}#2\\
    \end{array}
  \right]}
```

and then just use it as `$\mat{0&1\\2&3}$` producing $\left[\begin{array}{rr} 0 & 1 \\ 2 & 3 \end{array}\right]$. A compact $\begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix}$ will be possible with a slight variation, which I call `\matc`:

```
\newcommand{\matc}[2][]{
  \begin{bmatrix}
    {#1}#2\\
  \end{bmatrix}}
```

11.10 Sub(super)script sizes are too large for font size 11

Font sizes for subscripts would work with size 12 documents, however changing a document to size 11 will make subscripts too large. It relates to the fact that math mode calculates sizes for 11 as $10 * sqrt(1.2) \approx 10.95$, generating some sort of internal problem.

One way of dealing with it is to add to the preamble (only once):

```
\makeatletter
 \DeclareMathSizes{\@xipt}{\@xipt}{6}{5}
\makeatother
```

[32] see p.939 of [6] just add to .

whenever dealing with size 11 documents. Manual encoding of sizes is possible in which `\@xipt` can be simply replaced with `10.95`. The syntax of the command is described as: `\DeclareMathSizes{ts}{ms}{ss}{sss}`, where `ts` is the size of surrounding text (needs to be kept at the document size), `ms` math mode size, `ss` subscript size, `sss` subscript of subscript size (for nested subscripts).

Note that the following solution will only work with size 11 documents. The above declaration will have to be commented out if shifting to a different document size.

Also note, that the resulting presentation of subscript can be unpleasant to an eye as horizontal spacing between characters in subscript will be too large. This is especially noticeable when using punctuation characters in subscripts such as commas.

11.11 `bbm` Fonts are not Rendered in `pdf`

When importing package `bbm` make sure that TeXnicCenter is configured correctly as outlined in Section 8.19.

11.12 Produce formulae with transparent (not white) background

Sometimes output which is easy to generate with LaTeX is required for 3rd party applications such as MS PowerPoint. When the background of the container is not white it might be important to produce formulae etc with transparent background. One solution which works under Windows is a free PowerPoint plugin called IguanaTeX which allows generation of any LaTeX output directly in a presentation `ppt` file.

11.13 Transpose: display correct math symbol

It should be implemented with `X^{\mathsf{T}}` as X^{T}. Other alternatives include:

X^T with `X^T`

$X^\intercal$ with `X^{\intercal}` from `amssymb`

$X^{\trans}$ with `\trans{X}`

$X^{\tr}$ with `X^{\tr}`. g1 The declarations for the last two commands are given below:

```
\newcommand{\trans}[1]{{#1}^{\ensuremath{\mathsf{T}}}}
\newcommand\tr{{\mathpalette\raiseT{\intercal}}}
\newcommand\raiseT[2]{\raisebox{0.25ex}{$#1#2$}}
```

12 Miscellaneous

12.1 How to display date and time

Add the following code to the preamble. The time can be displayed anywhere in the document using the command `\printtime`[33].

[33] Requires package `calc`

```
\newcounter{hours}\newcounter{minutes}
\newcommand\printtime{\today\ \setcounter{hours}{\time/60}
   \setcounter{minutes}{\time-\value{hours}*60}
   \thehours:\theminutes}
```

12.2 Suppress left bookmarks pane in Adobe with pdf files

If one does not want a pdf file generated by Latex to be displayed in Adobe products with the left bookmarks pane open by default, one can add one of the following to the preamble:

```
\usepackage[pdfpagemode=UseNone]{hyperref}
```

or

```
\hypersetup{pdfpagemode=UseNone}
```

13 Referencing, links and hyperlinks

13.1 References in Figure/Float/Etc Captions

Always use protect in captions of any floats: `\caption{Eq.~\protect\ref{eq}}`. Otherwise, you might have a hard to replicate, inconsistent problem appearing elsewhere in the text. The way this problem was detected was that on the 3-4th compilation there would be a message about undefined messages (however, if either `hyperref` or `showkeys` are removed the problem will just disappear).

13.2 Replace all references with their actual labels

All raw references in text using `\label`, `\ref` and other commands will be inserted into the output if the package `showkeys` is simply imported. There are other options in which certain artifacts can be suppressed. See its documentation. Other packages which deal with this very problem are `showlabels` and `pukool`.

In `\showkeys` the size of font can be controlled only directly in its `sty` file which is located in the local MikTeX distribution folder. Simply search for the string `ttfamily` and you will find two locations, one for refs, one for cites, where the size can be adjusted. I currently use `\tiny`.

13.3 Urls

employment

The best way to add urls is to employ packages `url` and `hyperref`. Usage (requiring packages `url` and `hyperref`:

```
{\color{blue}\url{http://www.ctan.org}}
```

The url's fontsize can be set `\color{blue}\scriptsize\url{http://www.ctan.org}}`, but avoid doing it if the url is in the footnote as it will be reduced in size there automatically. For a possible problem of this useage with footnotes see a description below.

long urls

I think this one is resolved if using `url` *with* `hyperref` *and compiling with* `pdflatex`*. If the problem appears in* `dvi` *just import* `breakurl`*. However, read other subsections in this section.*

Note that it is hard to handle long url names properly in LaTeX. Either bad boxes will be generated or the link in the output pdf file will not go to the correct address if clicked. One possible bad hacky solution is to place the url into a parbox, i.e. `\parbox{15cm}{ \color{blue}\url{http:...}}` and adjust the width of the parbox accordingly from the 15cm given in this example to fit. So it fixes the url, however visually, the string still exceeds the right margin of the parbox and thus the user will still get a overfill warning. Additionally, the spacing between the lines of the url inside the parbox will be a bit bigger than the rest of the text and thus visually unpleasant. So both problems need to be solved if decided to use the `url` package for long names.

Extra info on settings can be found on the CTAN site:[34].

package hyperref may not wrap urls properly

bad solution Even when packages `url` and `hyperref` are set up properly this problem may cause overfull boxes for long urls. It arises in the dvi profile only (i.e. `pdflatex` is fine) and may be present when updating from an old version of `hyperref`[35]. The `pdflatex` output still appears correct. The way to correct is in the `dvi` profile to replace the signature of the package import from something like this:

```
\usepackage[hyperfootnotes=false]{hyperref}
```

to add `hypertex` driver:

```
\usepackage[hypertex,hyperfootnotes=false]{hyperref}
```

One way to do this is with conditional compilation described in Section 5.2. This method worked. See also Section (13.4).

good solution Unfortunately the above solution will not work if using `breakurl` package as described in this section (`breakurl` and `hypertex` are apparently uncompatible and code will throw multiple strange errors. So a better approach is to import at the minimum the following lines for a conditional compile to `dvi`:

```
\usepackage[dvips,hyperfootnotes=false]{hyperref}
\usepackage{breakurl}
```

However, the use of `dvips` is discouraged. See Sections 7.9 and 5.2 for more info.

[34] http://www.tex.ac.uk/cgi-bin/texfaq2html?label=setURL

[35] Apparently long ago the default in `hyperref.cfg` has changed from `hypertex` to `dvips`.

long urls in footnotes

There is a similar problem in which under certain conditions long urls in footnotes are not broken up at all. This happens when both conditions are met (the AND condition): (a) the width of the document is hardcoded in its preamble, i.e. `\textwidth=165.mm` and (b) the local font size declared to `\scriptsize` immediately before `\url` command is used. Changing to any other size (non-scriptsize) is enough to fix it, or alternatively, dropping the document's fixed text width.

change text of hyperlink

Package `hyperref` allows a display of a hyperlink shortcut with labeling different from the actual hypertext, it is worth using it. See syntax in Section 13.4.

urls with psfrag

Note that the `psfrag` package and its matlab helper routine `laprint` do not work with `pdflatex` so no special handling such as url-specific line breaks will be possible if they are used. There's a workaround involving `\breakurl` with `dvips` driver see Section 10.8.

urls in citations

See Section 6.2.

13.4 Add hyperreferences to contents

Just add `\usepackage{hyperref}`. This will create links both in dvi and pdf files for sections in the contents and urls as well. The appearance in pdf will not be too pretty due to using red boxes in contents and blue boxes for urls. Hence, it's best to the `\url` package for url links[36]. In addition it automatically opens a pdf viewer with the left hand side toolbar with the tree structure of the document. Might want to read more about this package to avoid those unpleasant by-products.

However, if the pdf is created via LaTeX=> PS => PDF option in the texniccenter output, then no red/blue boxes will be created. Only the links to url will be kept.

You can also tune the PDF file with the hyperref package (see manual for details). The following customisations can be used[37]:

1. Options to use when loading the package:

`\usepackage[argument1, argument2, ...]{hyperref}`

`bookmarks=true` create a linked ToC, as well as links for the cross-references such as calls to figure and page numbers, `colorlinks=true`, `backref` adds backlink text to the end of each item in the bibliography, to allow for `backref` to add page numbers you will also need `pagebackref=true`, `hyperfootnotes=false` see Section (13.5).

[36] Otherwise use the following syntax for urls: `\color{blue}\href{http://www.ctan.org}{webpage}`. Note that the font in which url is displayed will also need to be manually changed to some typed one.

[37] This and below is from http://merkel.zoneo.net/Latex

Now in addition, if using both `pdflatex` and `latex` you will have to specify conditional drivers to them; drivers are specified just like any other argument. So `latex` requires argument `dvips`, while `pdflatex` no argument (Note that `dvips` is strongly discouraged as described in Sections 7.9 and 5.2. The workaround is given in Section 7.9 which means that no separate import of `dvips` is required at all.). To implement it in code, you will need conditional loading with `ifpdf` which is described in Section (5.2). For an example of why a driver might be required with `hyperref` see Section (13.3). If we used driver `dvips` with `pdflatex` this will result in a warning.

2. You can also add information to the properties of your pdf document:

```
\hypersetup{
    pdftitle = The title of my PDF,
    pdfauthor = My name,
    pdfsubject= The subject,
    pdfkeywords =  keyword1 keyword2 keyword3}
```

3. Finally, I also used colors:

```
\hypersetup{
    colorlinks = true,
    linkcolor = red,
    anchorcolor = red,
    citecolor = blue,
    filecolor = red,
    pagecolor = red,
    urlcolor = red}
```

That will put some colors in the PDF documents to make the links more obvious to the reader.

4. Close/suppress Adobe bookmarks pane for pdfs created in latex - described in Section 12.2.
5. On what to do if urls are not broken correctly with `hyperref` see Section (13.3).

13.5 Problems to Avoid When Using `hyperref`

Due to some compatibility issue between `footmisc` and `hyperref`, if `hyperref` is loaded with `hyperfootnotes=true` option (which is the default unless specifically overriden with `false` in the preamble, see Section 13.4), we will not be able to use option `multiple` of `footmisc`. Specifically, adjacent footnotes are not going to be automatically separated by commas as outlined in Section 7.4. To solve this just use `hyperfootnotes=false`. This will stop refs to footnotes being clickable, but will result in the proper separation.

A suggestion was made to use `zref.sty` as a solution if clickability is required. I have not tested if this works yet.

14 Errors, warnings and other undesirables

14.1 Package hyperref: Token not allowed in PDFDocEncoded

Most probably refers to the situation described in Section 8.15.

14.2 Underfull hbox (badness 10000) in paragraph at ...

Sometimes, and it is not really clear exactly why, this warning appears. There is no adverse effect on the output produced. One of the causes encountered was when trying to terminate a line with two \\ new line characters. Another instance was when using the new line character \\ in the presence of space between the lines. In either case, it appears to be caused by the apparent problem with the system detecting spaces as valid characters, i.e. there are spaces but they are ignored and the system thinks there's no line to terminate. The quick fix is to include an empty verbatum string (using | for example if defined) between the two \\ to force creation of the line and thus make latex recognize it. More information on this is in Section 8.7.

The second reason why this very warning may happen is when one hacks the typesetting process by using one of the methods described in section 11.7. The informal explanation of this error then is that the compiler sees that the line is really badly and loosely typeset, so it notifies the user. You can see the problem if you inspect the line. Read that section to see how this problem can be corrected.

14.3 Package amsmath: Unable to redefine math \vec accent

This warning is given if the package `amsmath` is used in conjuction with the Springer's class `llncs` most probably declared in the preamble as `\documentclass{llncs}`.

`\vec` is normaly a math accent that puts an arrow over the symbol and it comes in the standard bundle and as the `amsmath` package with the same definition, while Springer define it as a bold symbol. Apparently, Springer's class redefines `\vec` so that it no longer is a plain `\mathaccent`. Amsmath detects this and prints a message that it cannot perform its magic trick on `\vec`. Amsmath calling this a warning is excessive and the warning can be ignored.

WARNING: do not do this if the Springer's behavior is to be preserved. If the default meaning of `\vec` is to be restored and the warning to be removed permanently the following fix can be done (source - newsgroups; tested):

```
\let\accentvec\vec              %add before \documentclass
\documentclass{llncs}
\let\spvec\vec                  %add before \usepackage{amsmath}
\let\vec\accentvec
\usepackage{amsmath}
\begin{document}
 $\vec{x} \quad \spvec{x}$
\end{document}
```

I do not know exactly what the command `\let` does, but it seems as it copies the behavior of the second argument to the first one. So when it is called before the class declaration it stores the default one in `\accentvec` and then assigns the Springer's one to `\spvec` and the default one back to `\vec`.

By default (and with `amsmath`) `\vec` produces $\vec{x}$, but Springer's `llncs` changes it to $\boldsymbol{x}$. After implementation of the above code `\vec` will generate the default $\vec{x}$, while `\spvec` will produce Springer's $\boldsymbol{x}$.

14.4 Overfull hbox (XX.XXXXpt too wide) in paragraph at ...

This might be caused by a too long inline formula or a too long inline verbatim text. For possible solutions see Section 11.7.

14.5 Underfull vbox ...

Although have not seen this error yet, the command supressing splitting footnotes described in Section 7.4 may cause this. Read references given in that section for any help.

14.6 Table of contents misbehaves

Sometimes out of the blue there will be a few douzen errors which could be tracked down to `\tableofcontents` command. Indenting this command out might stop errors appearing. The possible cause is the use of `Verbatim` environment in headings. Correct any headings recently edited, delete all the files from the source directory (!) and rerun again.

14.7 No fancyvrb.cfg found

Somehow this particular file does not come with the `fancyvrb` package. You need to download it manually from the `ctan` website.

14.8 pdfTeX warning: name(Hfootnote.xx) has been referenced...

The warning may appear only when generating an output in the LaTeX => PDF profile. The DVI mode will never have this type of error. The cause - footnotes contain references. Two solutions. (a) Adding `[hyperfootnotes=false]` option to `hyperref` package declaration disables clickable hyperlinking of footnotes. (b) Remove any references from footnotes. (c) Not using LaTeX => PDF profile. The first solution is the most elegant and works perfectly fine (I think even hyperlinked references are left intact).

14.9 PDF output is too light

This happens only if T1 font is used as described in Section 8.2 and the system did not have T1 (Type 1) version of the EC fonts installed (whatever this means). In the current uni TeX installation I apparently have the older version of MikTeX for the reasons explained in Section 4.1 and probably this is the reason for having this problem. So it should not really be a problem with other computers.

Solution: download and install the package `cm-super` (appr. 64Mb). There might be problems configuring it in MikTeX 2.4. Try the following steps after unzipping the package (which worked).

1. Copy `*.pfb` files to `C:/Program Files/texmf/fonts/type1/public/cm-super/`
2. Copy all files in `dvips` to `C:/Program Files/texmf/dvips/cm-super/`
3. Add the following lines to the file `C:/Program Files/texmf/web2c/updmap.cfg`

```
# cm-super
MixedMap cm-super-t1.map
MixedMap cm-super-t2a.map
MixedMap cm-super-t2b.map
MixedMap cm-super-t2c.map
MixedMap cm-super-ts1.map
MixedMap cm-super-x2.map
```

4. Run the following commands in `C:/Program Files/texmf/miktex/bin`

```
initexmf -u
initexmf initexmf --mkmaps
```

5. Restart the computer

If the above did not work alone the following should fix it.

1. Add to `C:/Program Files/texmf/dvips/config/config.ps`

```
# cm-super
p +cm-super-t1.map
p +cm-super-ts1.map
p +cm-super-t2a.map
p +cm-super-t2b.map
p +cm-super-t2c.map
p +cm-super-x2.map
```

2. Add to `C:\Program Files/texmf/pdftex/config/pdftex.cfg`

```
# cm-super
map +cm-super-t1.map
map +cm-super-ts1.map
map +cm-super-t2a.map
map +cm-super-t2b.map
map +cm-super-t2c.map
map +cm-super-x2.map
```

3. Add to file `C:/Program Files/texmf/dvipdfm/config/config`

```
f cm-super-t1.map
f cm-super-ts1.map
f cm-super-t2a.map
f cm-super-t2b.map
f cm-super-t2c.map
f cm-super-x2.map
```

4. Run the following commands in `C:/Program Files/texmf/miktex/bin`

```
initexmf -u
initexmf initexmf --mkmaps
```

5. Restart the computer

14.10 pdfTeX warning: destination with the same identifier...

The full message: "pdfTeX warning: destination with the same identifier (namefigure.1) has been already used, duplicate ignored", appears when using `float` and `hyperref` packages in the same document. Just make sure that `float` is declared first.

14.11 Problem with `koma` after upgrading to v.3.0

Upgrading to a new koma distribution brought a lot of trouble under MikTeX 2.7. One problem is that the documentclass statement with `scrbook` must be changed from something like this:

```
\documentclass[11pt, listof=totoc, twoside, footsepline, headsepline, openright,
               a4paper, bibtotoc, idxtotoc, pointlessnumbers]{scrbook}
```

to a new version:

```
\documentclass[11pt, twoside, footsepline, headsepline, openright, a4paper,
              listof=totoc, bibliography=totoc, index=totoc, numbers=noenddot]{scrbook}
```

that is because some of the parameter names have changed.

The other problems with koma are described in Sections (14.12) and (14.13).

14.12 Class scrreprt Warning: \float@addtolists detected!

This warning appears after update to koma-script v.3.0, e.g. when the entire MikTex distribution got updated from the scratch. The strange warning message is generated by koma-script package.

```
Class scrreprt Warning: \float@addtolists detected!
(scrreprt) You should use the features of package 'tocbasic'
(scrreprt) instead of \float@addtolists.
(scrreprt) Support for \float@addtolists may be removed from
(scrreprt) 'scrreprt' soon .
```

Importing package \usepackage{scrhack} which hacks koma helps. Bear in mind though that the following order must be observed for this to work correctly: scrhack, footmisc must appear first in the preamble right after the documentclass statement, and hyperref at the end of the preamble. Also see Section (14.11).

14.13 LaTeX Warning: Command \@footnotetext has changed

This problem appears either after an update to MikTeX 2.7 or after an upgrade to koma as described in Section (14.11). The compiler gives the following warning messages:

```
LaTeX Warning: Command \@footnotetext has changed.
               Check if current package is valid.
```

The problem has been traced back to the fact that now it's required to import hyperref only after footmisc, and keep the order described in Section (14.12).

Another instance when this warning can appear is the conflict between footmisc and setspace described in Section 7.4.

14.14 Document does not Hyphenate

Sometimes hyphenation does not come through, for example if using the package babel with the option british, or if using the scrbook document class. To quickly check if there is a problem with hyphenation, in the preamble type \showhyphens{propaganda}. If the word is not hyphenated in the compilation output, it is worth trying to import babel with the option english.

Appendices

A Example of Appendix

References

[1] M. Shell, *How to Use the IEEEtran BibTeX Style*. Online, accessed 12 Feb 2009, http://www.cs.auckland.ac.nz/~al/ieee_bibtex.pdf.
Referenced on page: 20, 21, 24

[2] M. Goossens, F. Mittelbach, and S. Rahtz, The LATEX Graphics Companion: Illustrating Documents with TEX and PostScript. Reading, Ma.: Addison-Wesley, 2005.
Referenced on page: 8, 57

[3] S. Pakin, *The Comprehensive LaTeX Symbol List*. Online, accessed 15 Aug 2008, http://www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-a4.pdf.
Referenced on page: 8, 16, 22

[4] M. Downes, *Short Math Guide for LaTeX: American Mathematical Society*. Online, accessed 09 Dec 2009, ftp://ftp.ams.org/pub/tex/doc/amsmath/short-math-guide.pdf.
Referenced on page: 8

[5] H. Voß, *Math mode*. Online, accessed 18 Aug 2008, http://www.ctan.org/tex-archive/info/math/voss/mathmode/Mathmode.pdf.
Referenced on page: 8, 57

[6] F. Mittelbach and M. Goossens, The LaTeX companion, 5th ed. Boston: Addison-Wesley, 2006.
Referenced on page: 6, 8, 16, 33, 34, 35, 37, 39, 39, 41, 48, 58, 59, 60