On the Use of the `ubcdiss` Template

by

Johnny Canuck

B. Basket Weaving, University of Illustrious Arts, 1991

M. Silly Walks, Another University, 1994

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

(Basket Weaving)

The University of British Columbia

(Vancouver)

April 2192

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the thesis entitled:

On the Use of the `ubcdiss` Template

submitted by **Johnny Canuck** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy** in **Basket Weaving**.

Examining Committee:

John Smith, Professor, Materials Engineering, UBC
Supervisor

Mary Maker, Professor, Materials Engineering, UBC
Supervisory Committee Member

Nebulous Name, Position, Department, Institution
Supervisory Committee Member

Magnus Monolith, Position, Other Department, Institution
Additional Examiner

Additional Supervisory Committee Members:

Ira Crater, Professor, Materials Engineering, UBC
Supervisory Committee Member

Adeline Long, CEO of Aerial Machine Transportation, Inc.
Supervisory Committee Member

Abstract

Graph structured data is used in a variety of applications because it naturally models ubiquitous concepts such as social networks, protein structures, and supply chains [6, 16]. This has motivated the development of Graph Processing Systems (GPS) whose aim is to process analytic queries such as PageRank, Shortest Paths, or Connected Components. Previous work accelerated graph processing by modifying the graph's data layout in memory [1, 5, 13] or on disk [10, 12]. Since we typically describe a graph G in terms of its Vertex and Edge sets using the notation $G(V,E)$, most GPS can be categorized as Vertex-Centric (VC) [8, 10, 18] or Edge-Centric (EC) [15] dependant on whether the systems implement analytical queries by applying a function over each vertex or each edge.

In the VC model, algorithms rely on user-defined vertex programs to compute analytic properties of an input graph. Vertex programs are run iteratively on every vertex in the graph. In each iteration, each vertex executes a user-defined vertex program, and messages are exchanged between neighbouring vertices to propagate updated vertex values. VC systems reorder the *vertices* of the graph to improve the locality of vertices that are expected to be frequently accessed together.

In the EC model, iteration occurs over the edges of the graph. For each edge in the graph, we apply an update to either the source or destination vertex incident on that edge. EC systems reorder the *edges* of the graph to mitigate the random-access pattern of incident vertices that is common to many graph processing kernels such as PageRank.

One way of representing a graph is as a square $N \times N$ matrix, A, where $N = |V|$. A non-zero element $A_{i,j}$ indicates the existence of an edge from vertex i to vertex j. Real-world graphs are typically sparse [4], meaning that the number of

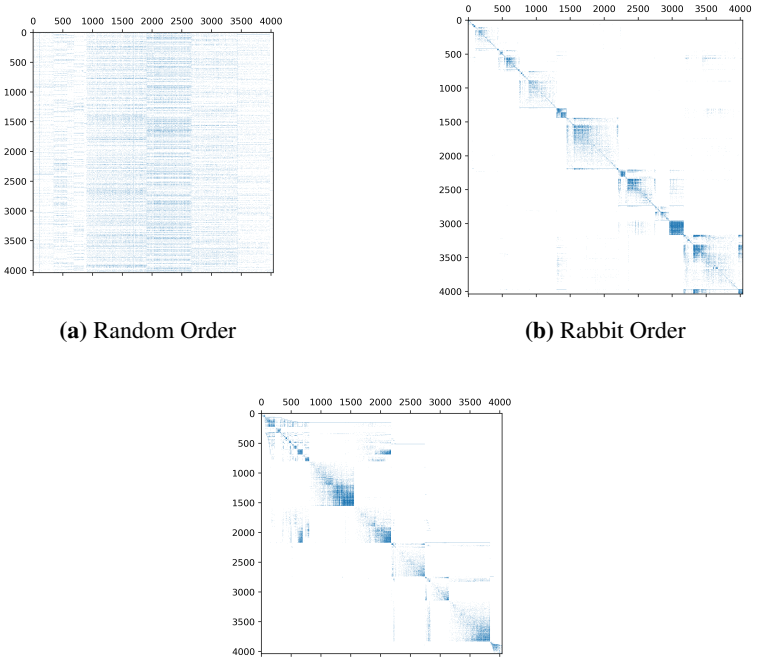
edges, $M = |E|$, is much smaller than the number of possible edges in the graph: $M \ll M_{\max} = \binom{N}{2}$.

EC systems that order the edges of the graph by ascending *Source* or *Destination* ID effectively iterate over A in Row-major or Column-major order, respectively. If we traverse the adjacency matrix of a graph in Row-major order, we will have excellent locality in the source vertices (since we will process all outgoing neighbours of a source vertex before moving on to the next source vertex), but our access to the destination vertex of each outgoing edge will corrsepond to near-random memory accesses of the vertex array. Prior work [13] addressed this concern by using an edge ordering defined by the Hilbert Space Filling Curve (HSFC), which is a way of assigning indices to the edges of a graph that produces locality in *both* the source and destination vertices.

VC systems can use vertex reordering as a preprocessing optimization to improve the memory access locality of the vertices. For example, certain VC systems [5, 17] use the degrees of the vertices of the graph to cluster and/or sort the vertices (e.g., sorting the vertices by descending order of degree). The rationale for this optimization is that high degree vertices (also known as "Hub" vertices) are, by definition, neighbours of a large number of vertices. These hubs will be frequently accessed when iterating over the edge set of the graph [2]. A descending degree sort colocates these hub vertices in memory and increases the likelihood of frequently accessed vertices being cached. Alternatively, Rabbit Order [1] relies on the observation that many real-world graphs such as social networks contain community structures, where vertices that belong to the same community share a larger number of edges than vertices that belong to separate communities. The goal of Rabbit Order is to order the vertices in such a way that consecutive vertex IDs correspond to meaningful communities. The algorithm first detects these communities and then labels the vertices according to those communities.

A vertex reordering can be defined as a function that maps each vertex ID to a new ID in the range: $[0,N)$. This mapping or relabeling is known as a *graph isomorphism*, since all edges between vertices are preserved and the structure of the graph is unchanged. However, observe that vertex reordering can impose "structure" on the adjacency matrix of the graph. Figure 1 shows the adjacency matrix of a Facebook social network graph that was constructed from the data of survey

participants using the Facebook App [11]. The graph has been reordered using 3 vertex orders: a random vertex ID assignment, and the IDs calculated using Rabbit and Cuthill-McKee orders.



(a) Random Order

(b) Rabbit Order

(c) Cuthill-McKee Order

Figure 1: Adjacency matrices of an undirected Facebook Social Network graph with 4,039 vertices and 88,234 edges. Each pixel denotes an undirected edge ("friend-of" relationship) between a pair of vertices (users) in the graph. Note the detected dense communities (submatrices) along the diagonal in (b) and the reduced bandwidth of the sparse matrix in (c).

This thesis is concerned in the interaction between Vertex and Edge ordering. Namely:

RQ1: It has been shown that using vertex and edge ordering to preprocess a graph confers performance benefits in a variety of applications. Is it possible to combine vertex and edge ordering as a preprocessing step such that the performance benefit gained by the combination of both is compounded (i.e., is greater

than using either separately)?

- Can edge-centric graph traversal be sped up by first introducing some structure into the adjacency matrix?

RQ2: Given an arbitrary input graph, is there a *vertex-and-edge* ordering combination that yields the best performance for an EC traversal (measured in speed of execution and number of cache misses)?

This work answers these questions by making the following contributions. We:

1. Perform a preliminary performance evaluation on different vertex and edge orderings on single-threaded EC traversals for a variety of graph datasets. We conclude that there *does not* exist a one-size-fits-all *vertex-and-edge* ordering that outperforms all others for all types of graphs.

2. Derive an analytic model of performance using a dataset of graph features (statistical measures that summarize a graph), to identify the characteristics of a graph that help us determine which vertex-and-edge ordering performs best for EC iteration.

3. Develop a fully parallel implementation of the SlashBurn vertex ordering technique: **ParSB**.

4. Propose a novel, lock-free, multithreaded vertex-and-edge ordering technique that leverages the compressed graph representation given by ParSB and traverses the edges of the graph using the HSFC.

5. Evaluate our novel vertex-and-edge ordering and compare its performance against locking-based and merging-based multithreaded EC traversals.

Lay Summary

The lay or public summary explains the key goals and contributions of the research/ scholarly work in terms that can be understood by the general public. It must not exceed 150 words in length.

Preface

At University of British Columbia (UBC), a preface may be required. Be sure to check the GPS guidelines as they may have specific content to be included.

Table of Contents

List of Tables

List of Figures

Glossary

This glossary uses the handy `acroynym` package to automatically maintain the glossary. It uses the package's `printonlyused` option to include only those acronyms explicitly referenced in the LaTeX source. To change how the acronyms are rendered, change the `\acsfont` definition in `diss.tex`.

CTAN The Common TeX Archive Network

DOI Document Object Identifier (see http://doi.org)

GPS Graduate and Postdoctoral Studies

RCS Revision control system, a software tool for tracking changes to a set of files

URL Unique Resource Locator, used to describe a means for obtaining some resource on the world wide web

GPS Graph Processing System

VC Vertex-Centric

EC Edge-Centric

HSFC Hilbert Space Filling Curve

Acknowledgments

Thank those people who helped you.

Don’t forget your parents or loved ones.

You may wish to acknowledge your funding sources.

Chapter 1

Introduction

If I have seen farther it is by standing on the shoulders of Giants.
— Sir Isaac Newton (1855)

This document provides a quick set of instructions for using the `ubcdiss` class to write a dissertation in LaTeX. Unfortunately this document cannot provide an introduction to using LaTeX. The classic reference for learning LaTeX is Lamport's book [9]. There are also many freely-available tutorials online; Andy Roberts' online LaTeX tutorials[1] seems to be excellent. The source code for this docment, however, is intended to serve as an example for creating a LaTeX version of your dissertation.

We start by discussing organizational issues, such as splitting your dissertation into multiple files, in Section 1.1. We then cover the ease of managing cross-references in LaTeX in Section 1.2. We cover managing and using bibliographies with BibTeX in Section 1.3. We briefly describe typesetting attractive tables in Section 1.4. We briefly describe including external figures in Section 1.5, and using special characters and symbols in Section 1.6. As it is often useful to track different versions of your dissertation, we discuss revision control further in Section 1.8. We conclude with pointers to additional sources of information in Section 1.10.

[1] http://www.andy-roberts.net/misc/latex/

1.1 Suggested Thesis Organization

The UBC Graph Processing System (GPS) specifies a particular arrangement of the components forming a thesis.[2] This template reflects that arrangement.

In terms of writing your thesis, the recommended best practice for organizing large documents in LaTeX is to place each chapter in a separate file. These chapters are then included from the main file through the use of `\include{file}`. A thesis might be described as six files such as `intro.tex`, `relwork.tex`, `model.tex`, `eval.tex`, `discuss.tex`, and `concl.tex`.

We also encourage you to use macros for separating how something will be typeset (e.g., bold, or italics) from the meaning of that something. For example, if you look at `intro.tex`, you will see repeated uses of a macro `\file{}` to indicate file names. The `\file{}` macro is defined in the file `macros.tex`. The consistent use of `\file{}` throughout the text not only indicates that the argument to the macro represents a file (providing meaning or semantics), but also allows easily changing how file names are typeset simply by changing the definition of the `\file{}` macro. `macros.tex` contains other useful macros for properly typesetting things like the proper uses of the latinate *exempli gratiā* and *id est* (i.e., `\eg` and `\ie`), web references with a footnoted URL (`\webref{url}{text}`), as well as definitions specific to this documentation (`\latexpackage{}`).

1.2 Making Cross-References

LaTeX make managing cross-references easy, and the `hyperref` package's `\autoref{}` command[3] makes it easier still.

A thing to be cross-referenced, such as a section, figure, or equation, is *labelled* using a unique, user-provided identifier, defined using the `\label{}` command. The thing is referenced elsewhere using the `\autoref{}` command. For example, this section was defined using:

```
\section{Making Cross-References}
\label{sec:CrossReferences}
```

[2] See http://www.grad.ubc.ca/current-students/dissertation-thesis-preparation/order-components

[3] The `hyperref` package is included by default in this template.

References to this section are made as follows:

```
We then cover the ease of managing cross-references in \LaTeX\
in \autoref{sec:CrossReferences}.
```

`\autoref{}` takes care of determining the *type* of the thing being referenced, so the example above is rendered as

> We then cover the ease of managing cross-references in LaTeX in Section 1.2.

The label is any simple sequence of characters, numbers, digits, and some punctuation marks such as ":" and "–"; there should be no spaces. Try to use a consistent key format: this simplifies remembering how to make references. This document uses a prefix to indicate the type of the thing being referenced, such as `sec` for sections, `fig` for figures, `tbl` for tables, and `eqn` for equations.

For details on defining the text used to describe the type of *thing*, search `diss.tex` and the `hyperref` documentation for `autorefname`.

1.3 Managing Bibliographies with BibTeX

One of the primary benefits of using LaTeX is its companion program, BibTeX, for managing bibliographies and citations. Managing bibliographies has three parts: (i) describing references, (ii) citing references, and (iii) formatting cited references.

1.3.1 Describing References

BibTeX defines a standard format for recording details about a reference. These references are recorded in a file with a `.bib` extension. BibTeX supports a broad range of references, such as books, articles, items in a conference proceedings, chapters, technical reports, manuals, dissertations, and unpublished manuscripts. A reference may include attributes such as the authors, the title, the page numbers, the Document Object Identifier (DOI), or a Unique Resource Locator (URL). A reference can also be augmented with personal attributes, such as a rating, notes, or keywords.

Each reference must be described by a unique *key*.[4] A key is a simple sequence

[4] Note that the citation keys are different from the reference identifiers as described in Section 1.2.

of characters, numbers, digits, and some punctuation marks such as ":" and "–"; there should be no spaces. A consistent key format simiplifies remembering how to make references. For example:

last-name–*year*–*contracted-title*

where *last-name* represents the last name for the first author, and *contracted-title* is some meaningful contraction of the title. Then Kiczales et al.'s seminal article on aspect-oriented programming [7] (published in 1997) might be given the key `kiczales-1997-aop`.

An example of a BibTEX `.bib` file is included as `biblio.bib`. A description of the format a `.bib` file is beyond the scope of this document. We instead encourage you to use one of the several reference managers that support the BibTEX format such as JabRef[5] (multiple platforms) or BibDesk[6] (MacOS X only). These front ends are similar to reference managers such as EndNote or RefWorks.

1.3.2 Citing References

Having described some references, we then need to cite them. We do this using a form of the `\cite` command. For example:

```
\citet{kiczales-1997-aop} present examples of crosscutting
from programs written in several languages.
```

When processed, the `\citet` will cause the paper's authors and a standardized reference to the paper to be inserted in the document, and will also include a formatted citation for the paper in the bibliography. For example:

> Kiczales et al. [7] present examples of crosscutting from programs written in several languages.

There are several forms of the `\cite` command (provided by the `natbib` package), as demonstrated in Table 1.1. Note that the form of the citation (numeric or author-year) depends on the bibliography style (described in the next section). The `\citet` variant is used when the author names form an object in the sentence,

[5] http://jabref.sourceforge.net
[6] http://bibdesk.sourceforge.net

Table 1.1: Available `cite` variants; the exact citation style depends on whether the bibliography style is numeric or author-year.

Variant	Result
`\cite`	Parenthetical citation (e.g., "[7]" or "(Kiczales et al. 1997)")
`\citet`	Textual citation: includes author (e.g., "Kiczales et al. [7]" or or "Kiczales et al. (1997)")
`\citet*`	Textual citation with unabbreviated author list
`\citealt`	Like `\citet` but without parentheses
`\citep`	Parenthetical citation (e.g., "[7]" or "(Kiczales et al. 1997)")
`\citep*`	Parenthetical citation with unabbreviated author list
`\citealp`	Like `\citep` but without parentheses
`\citeauthor`	Author only (e.g., "Kiczales et al.")
`\citeauthor*`	Unabbreviated authors list (e.g., "Kiczales, Lamping, Mendhekar, Maeda, Lopes, Loingtier, and Irwin")
`\citeyear`	Year of citation (e.g., "1997")

whereas the `\citep` variant is used for parenthetic references, more like an endnote. Use `\nocite` to include a citation in the bibliography but without an actual reference.

1.3.3 Formatting Cited References

BibTEX separates the citing of a reference from how the cited reference is formatted for a bibliography, specified with the `\bibliographystyle` command. There are many varieties, such as `plainnat`, `abbrvnat`, `unsrtnat`, and `vancouver`. This document was formatted with `abbrvnat`. Look through your TEX distribution for `.bst` files. Note that use of some `.bst` files do not emit all the information necessary to properly use `\citet{}`, `\citep{}`, `\citeyear{}`, and `\citeauthor{}`.

There are also packages available to place citations on a per-chapter basis (`bibunits`), as footnotes (`footbib`), and inline (`bibentry`). Those who wish to exert maximum control over their bibliography style should see the amazing `custom-bib` package.

1.4 Typesetting Tables

Lamport [9] made one grievous mistake in LaTeX: his suggested manner for typesetting tables produces typographic abominations. These suggestions have unfortunately been replicated in most LaTeX tutorials. These abominations are easily avoided simply by ignoring his examples illustrating the use of horizontal and vertical rules (specifically the use of `\hline` and `|`) and using the `booktabs` package instead.

The `booktabs` package helps produce tables in the form used by most professionally-edited journals through the use of three new types of dividing lines, or *rules*. Tables 1.1 and 1.2 are two examples of tables typeset with the `booktabs` package. The `booktabs` package provides three new commands for producing rules: `\toprule` for the rule to appear at the top of the table, `\midrule` for the middle rule following the table header, and `\bottomrule` for the bottom-most at the end of the table. These rules differ by their weight (thickness) and the spacing before and after. A table is typeset in the following manner:

```
\begin{table}
\caption{The caption for the table}
\label{tbl:label}
\centering
\begin{tabular}{cc}
\toprule
Header & Elements \\
\midrule
Row 1 & Row 1 \\
Row 2 & Row 2 \\
% ... and on and on ...
Row N & Row N \\
\bottomrule
\end{tabular}
\end{table}
```

See the `booktabs` documentation for advice in dealing with special cases, such as subheading rules, introducing extra space for divisions, and interior rules.

1.5 Figures, Graphics, and Special Characters

Most LaTeX beginners find figures to be one of the more challenging topics. In LaTeX, a figure is a *floating element*, to be placed where it best fits. The user is not

Figure 1.1: Proof of LaTeX's amazing abilities

expected to concern him/herself with the placement of the figure. The figure should instead be labelled, and where the figure is used, the text should use `\autoref` to reference the figure's label. Figure 1.1 is an example of a figure. A figure is generally included as follows:

```
\begin{figure}
\centering
\includegraphics[width=3in]{file}
\caption{A useful caption}
\label{fig:fig-label}   % label should change
\end{figure}
```

There are three items of note:

1. External files are included using the `\includegraphics` command. This command is defined by the `graphicx` package and can often natively import graphics from a variety of formats. The set of formats supported depends on your TeX command processor. Both `pdflatex` and `xelatex`, for example, can import GIF, JPG, and PDF. The plain version of `latex` only supports EPS files.

2. The `\caption` provides a caption to the figure. This caption is normally listed in the List of Figures; you can provide an alternative caption for the LoF by providing an optional argument to the `\caption` like so:

   ```
   \caption[nice shortened caption for LoF]{%
       longer detailed caption used for the figure}
   ```

   GPS generally prefers shortened single-line captions in the LoF: multiple-line captions are a bit unwieldy.

3. The `\label` command provides for associating a unique, user-defined, and descriptive identifier to the figure. The figure can be can be referenced elsewhere in the text with this identifier as described in Section 1.2.

Table 1.2: Useful LaTeX symbols

LaTeX	Result	LaTeX	Result
`\texttrademark`	™	`\&`	&
`\textcopyright`	©	`\{ \}`	{ }
`\textregistered`	®	`\%`	%
`\textsection`	§	`\verb!~!`	~
`\textdagger`	†	`\$`	$
`\textdaggerdbl`	‡	`\^{}`	^
`\textless`	<	`\_`	_
`\textgreater`	>		

See Keith Reckdahl's excellent guide for more details, *Using imported graphics in LaTeX2e*[7].

1.6 Special Characters and Symbols

LaTeX appropriates many common symbols for its own purposes, with some used for commands (e.g., `\{}&%`) and mathematics (e.g., `$^_`), and others are automagically transformed into typographically-preferred forms (e.g., `--''`) or to completely different forms (e.g., <>). Table 1.2 presents a list of common symbols and their corresponding LaTeX commands. A much more comprehensive list of symbols and accented characters is available at: http://www.ctan.org/tex-archive/info/symbols/comprehensive/

1.7 Changing Page Widths and Heights

The `ubcdiss` class is based on the standard LaTeX `book` class [9] that selects a line-width to carry approximately 66 characters per line. This character density is claimed to have a pleasing appearance and also supports more rapid reading [3]. I would recommend that you not change the line-widths!

[7] http://www.ctan.org/tex-archive/info/epslatex.pdf

1.7.1 The `geometry` Package

Some students are unfortunately saddled with misguided supervisors or committee members whom believe that documents should have the narrowest margins possible. The `geometry` package is helpful in such cases. Using this package is as simple as:

```
\usepackage[margin=1.25in,top=1.25in,bottom=1.25in]{geometry}
```

You should check the package's documentation for more complex uses.

1.7.2 Changing Page Layout Values By Hand

There are some miserable students with requirements for page layouts that vary throughout the document. Unfortunately the `geometry` can only be specified once, in the document's preamble. Such miserable students must set LaTeX's layout parameters by hand:

```
\setlength{\topmargin}{-.75in}
\setlength{\headsep}{0.25in}
\setlength{\headheight}{15pt}
\setlength{\textheight}{9in}
\setlength{\footskip}{0.25in}
\setlength{\footheight}{15pt}

% The *sidemargin values are relative to 1in; so the following
% results in a 0.75 inch margin
\setlength{\oddsidemargin}{-0.25in}
\setlength{\evensidemargin}{-0.25in}
\setlength{\textwidth}{7in}       % 1.1in margins (8.5-2*0.75)
```

These settings necessarily require assuming a particular page height and width; in the above, the setting for `\textwidth` assumes a US Letter with an 8.5" width. The `geometry` package simply uses the page height and other specified values to derive the other layout values. The `layout` package provides a handy `\layout` command to show the current page layout parameters.

1.7.3 Making Temporary Changes to Page Layout

There are occasions where it becomes necessary to make temporary changes to the page width, such as to accomodate a larger formula. The `chngpage` package provides an `adjustwidth` environment that does just this. For example:

```
% Expand left and right margins by 0.75in
\begin{adjustwidth}{-0.75in}{-0.75in}
% Must adjust the perceived column width for LaTeX to get with it.
\addtolength{\columnwidth}{1.5in}
\[ an extra long math formula \]
\end{adjustwidth}
```

1.8 Keeping Track of Versions with Revision Control

Software engineers have used Revision control system (RCS) to track changes to their software systems for decades. These systems record the changes to the source code along with context as to why the change was required. These systems also support examining and reverting to particular revisions from their system's past.

An RCS can be used to keep track of changes to things other than source code, such as your dissertation. For example, it can be useful to know exactly which revision of your dissertation was sent to a particular committee member. Or to recover an accidentally deleted file, or a badly modified image. With a revision control system, you can tag or annotate the revision of your dissertation that was sent to your committee, or when you incorporated changes from your supervisor.

Unfortunately current revision control packages are not yet targetted to non-developers. But the Subversion project's TortoiseSVN[8] has greatly simplified using the Subversion revision control system for Windows users. You should consult your local geek.

A simpler alternative strategy is to create a GoogleMail account and periodically mail yourself zipped copies of your dissertation.

1.9 Recommended Packages

The real strength to LaTeX is found in the myriad of free add-on packages available for handling special formatting requirements. In this section we list some helpful packages.

1.9.1 Typesetting

enumitem: Supports pausing and resuming enumerate environments.

[8] http://tortoisesvn.net/docs/release/TortoiseSVN_en/

ulem: Provides two new commands for striking out and crossing out text (`\sout{text}` and `\xout{text}` respectively) The package should likely be used as follows:

```
\usepackage[normalem,normalbf]{ulem}
```

to prevent the package from redefining the emphasis and bold fonts.

chngpage: Support changing the page widths on demand.

mhchem: Support for typesetting chemical formulae and reaction equations.

Although not a package, the `latexdiff`[9] command is very useful for creating changebar'd versions of your dissertation.

1.9.2 Figures, Tables, and Document Extracts

pdfpages: Insert pages from other PDF files. Allows referencing the extracted pages in the list of figures, adding labels to reference the page from elsewhere, and add borders to the pages.

subfig: Provides for including subfigures within a figure, and includes being able to separately reference the subfigures. This is a replacement for the older `subfigure` environment.

rotating: Provides two environments, sidewaystable and sidewaysfigure, for typesetting tables and figures in landscape mode.

longtable: Support for long tables that span multiple pages.

tabularx: Provides an enhanced tabular environment with auto-sizing columns.

ragged2e: Provides several new commands for setting ragged text (e.g., forms of centered or flushed text) that can be used in tabular environments and that support hyphenation.

[9] http://www.ctan.org/tex-archive/support/latexdiff/

1.9.3 Bibliography Related Packages

bibunits: Support having per-chapter bibliographies.

footbib: Cause cited works to be rendered using footnotes.

bibentry: Support placing the details of a cited work in-line.

custom-bib: Generate a custom style for your bibliography.

1.10 Moving On

At this point, you should be ready to go. Other handy web resources:

- Common TEX Archive Network (CTAN)[10] is *the* comprehensive archive site for all things related to TEX and LATEX. Should you have some particular requirement, somebody else is almost certainly to have had the same requirement before you, and the solution will be found on CTAN. The links to various packages in this document are all to CTAN.
- An online reference to LATEX commands[11] provides a handy quick-reference to the standard LATEX commands.
- The list of Frequently Asked Questions about TEX and LATEX[12] can save you a huge amount of time in finding solutions to common problems.
- The teTEX documentation guide[13] features a very handy list of the most useful packages for LATEX as found in CTAN.
- The color[14] package, part of the graphics bundle, provides handy commands for changing text and background colours. Simply changing text to various levels of grey can have a very dramatic effect.
- If you're really keen, you might want to join the TEX Users Group[15].

[10] http://www.ctan.org
[11] http://www.ctan.org/get/info/latex2e-help-texinfo/latex2e.html
[12] http://www.tex.ac.uk/cgi-bin/texfaq2html?label=interruptlist
[13] http://www.tug.org/tetex/tetex-texmfdist/doc/
[14] http://www.ctan.org/tex-archive/macros/latex/required/graphics/grfguide.pdf
[15] http://www.tug.org

Bibliography

[1] J. Arai, H. Shiokawa, T. Yamamuro, M. Onizuka, and S. Iwamura. Rabbit order: Just-in-time parallel reordering for fast graph analysis. In *2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 22–31. IEEE, 2016. → pages iii, iv

[2] V. Balaji and B. Lucia. When is graph reordering an optimization? In *IEEE International Symposium on Workload Characterization (IISWC)*, 2018. → page iv

[3] R. Bringhurst. *The Elements of Typographic Style*. Hartley & Marks, 2.5 edition, 2002. ISBN 0881791326. → page 8

[4] M. Danisch, O. Balalau, and M. Sozio. Listing k-cliques in sparse real-world graphs. In *Proceedings of the 2018 World Wide Web Conference*, pages 589–598, 2018. → page iii

[5] P. Faldu, J. Diamond, and B. Grot. A closer look at lightweight graph reordering. In *2019 IEEE International Symposium on Workload Characterization (IISWC)*, pages 1–13. IEEE, 2019. → pages iii, iv

[6] S. Heidari, Y. Simmhan, R. N. Calheiros, and R. Buyya. Scalable graph processing frameworks: A taxonomy and open challenges. *ACM Computing Surveys (CSUR)*, 51(3):1–53, 2018. → page iii

[7] G. Kiczales, J. Lamping, A. Mendhekar, C. Maeda, C. Lopes, J.-M. Loingtier, and J. Irwin. Aspect-oriented programming. In *Proceedings of the European Conference on Object-Oriented Programming (ECOOP)*, volume 2591 of *Lecture Notes in Computer Science*, pages 220–242, 1997. → pages 4, 5

[8] A. Kyrola, G. Blelloch, and C. Guestrin. GraphChi:Large-Scale graph computation on just a PC. In *10th USENIX Symposium on Operating*

Systems Design and Implementation (OSDI 12), pages 31–46, 2012. → page iii

[9] L. Lamport. *LaTeX: A Document Preparation System*. Addison-Wesley, 2 edition, 1994. ISBN 0201529831. → pages 1, 6, 8

[10] E. Lee, J. Kim, K. Lim, S. H. Noh, and J. Seo. Pre-select static caching and neighborhood ordering for bfs-like algorithms on disk-based graph engines. In *2019 USENIX Annual Technical Conference (USENIX ATC 19)*, pages 459–474, 2019. → page iii

[11] J. Leskovec and J. Mcauley. Learning to discover social circles in ego networks. *Advances in neural information processing systems*, 25, 2012. → page v

[12] S. Maass, C. Min, S. Kashyap, W. Kang, M. Kumar, and T. Kim. Mosaic: Processing a trillion-edge graph on a single machine. In *Proceedings of the Twelfth European Conference on Computer Systems*, pages 527–543, 2017. → page iii

[13] F. McSherry, M. Isard, and D. G. Murray. Scalability! but at what COST? In *15th Workshop on Hot Topics in Operating Systems (HotOS XV)*, 2015. → pages iii, iv

[14] J. K. Rowling. *Harry Potter and the Philospher's Stone*. Bloomsbury, 1997. ISBN 0747532699.

[15] A. Roy, I. Mihailovic, and W. Zwaenepoel. X-stream: Edge-centric graph processing using streaming partitions. In *Proceedings of the Twenty-Fourth ACM Symposium on Operating Systems Principles*, pages 472–488, 2013. → page iii

[16] Y. Yan, S. Zhang, and F.-X. Wu. Applications of graph theory in protein structure identification. *Proteome science*, 9(1):1–10, 2011. → page iii

[17] Y. Zhang, V. Kiriansky, C. Mendis, S. Amarasinghe, and M. Zaharia. Making caches work for graph analytics. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 293–302. IEEE, 2017. → page iv

[18] D. Zheng, D. Mhembere, R. Burns, J. Vogelstein, C. E. Priebe, and A. S. Szalay. FlashGraph: Processing Billion-Node graphs on an array of commodity SSDs. In *13th USENIX Conference on File and Storage Technologies (FAST 15)*, pages 45–58, 2015. → page iii

Appendix A

Supporting Materials

This would be any supporting material not central to the dissertation. For example:

- additional details of methodology and/or data;
- diagrams of specialized equipment developed.;
- copies of questionnaires and survey instruments.