POLITECNICO DI MILANO

Facoltà di Ingegneria

Dipartimento di Elettronica, Informazione e Bioingegneria

Master of Science in

Environmental and Land Planning Engineering

A template for master thesis at DEIB

Supervisor:

PROF. VILFREDO PARETO

Assistant Supervisor:

DR. PAUL KRUGMAN

Master Graduation Thesis by:

ANDREA COMINOLA
Student Id n. 111111

Academic Year 2011-2012

POLITECNICO DI MILANO

Facoltà di Ingegneria
Scuola di Ingegneria Civile, Ambientale e Territoriale
Dipartimento di Elettronica, Informazione e Bioingegneria

Corso di Laurea Magistrale in
Ingegneria per l’Ambiente e il Territorio

Un modello per tesi di laurea magistrale al DEIB

Relatore:

PROF. VILFREDO PARETO

Correlatore:

DR. PAUL KRUGMAN

Correlatore Alta Scuola Politecnica:

PROF. MY ASP SUPERVISOR

Tesi di Laurea Magistrale di:

ANDREA COMINOLA
Matricola n. 111111

EMANUELE MASON
Matricola n. 222222

Anno Accademico 2011-2012

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both LaTeX and LyX:

`http://code.google.com/p/classicthesis/`

Happy users of `classicthesis` usually send a real postcard to the author, a collection of postcards received so far is featured here:

`http://postcards.miede.de/`

Andrea Cominola and Emanuele Mason: *A template for master thesis at DEIB*, March 2013

To time, that do not go backwards

— A & E

ACKNOWLEDGMENTS

Thanks a lot to many people…

Big thanks to the scholarship "Politecnico di Milano - a.a. 2011/12 per tesi all'estero", we both could take advantage of.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LISTINGS

ACRONYMS

C++	C plus plus
DE	Differential Evolution
DEM	Digital Elevation Model
GDE3	Generalized Differential Evolution 3
MOEA	Multi Objective Evolutionary Algorithm
NFE	Number of Function Evaluations
XML	eXtensible Markup Language

ABSTRACT

An abstract is a brief of a research article, thesis, review, conference proceeding or any in-depth analysis of a particular subject or discipline, and is often used to help the reader quickly ascertain the paper's purpose. When used, an abstract always appears at the beginning of a manuscript or typescript, acting as the point-of-entry for any given academic paper or patent application. Abstracting and indexing services for various academic disciplines are aimed at compiling a body of literature for that particular subject.

SOMMARIO

Same as above but in italian

ESTRATTO

Breve riassunto in italiano della tesi da cui si capisca tutto.

PREFACE

A preface is an introduction to a book or other literary work written by the work's author. A preface generally covers the story of how the book came into being, or how the idea for the book was developed.

MOTIVATION

Part I

THEORY

You can put some informational part preamble text here.

1 INTRODUCTION

Science, my boy, is made up of mistakes, but they are mistakes which it is useful to make, because they lead little by little to the truth.

— Verne *Journey to the Center of the Earth*

This template is ready to be used when writing a thesis at Dipartimento di Elettronica, Informazione e Bioingegneria. It is a modified version of Classic Thesis by André Miede that can be found here `http://code.google.com/p/classicthesis/`. Refer to its manual for further informations. Any other package included have also its own manual that can be found by searching its name on Internet. Additional hints in italian can be found in "Introduzione allo stile Classic Thesis" and more general ones in "L'arte di scrivere con latex" both by Lorenzo Pantieri. If none of these answer your question, ask it on `http://tex.stackexchange.com/`.

1.1 SHOWCASE

Usefull features of the template are shown and explained.

1.1.1 *File structure*

The template is organized in multiple files. In the root folder, the main file which is to be compiled is `ClassicThesis_DEIB.tex`. You should just add the source files you want to include and any hyphenation you need to explictly specify. The `classicthesis-config.tex` contains options that can be chosen for this template. It contains also the definition for the title, the author and others stuff displayed in the titlepage. It is commented enough to be used. The other file present is `Bibliography.bib` which is the *Bibtex* database.

The other folders are:

A. `Chapters` contains source files for the main chapters of your thesis;

B. `CodeFiles` contains any code snippet you want to include in your thesis with the environment `listings`;

C. `FrontBackmatter` contains various files that are included to produce abstract, titlepages, acknoledgements, ... Modify them with your informations;

D. Images contains the .pdf or .png versions of the images of the thesis. A sources subfolder is also provided for keeping the original Inkscape, GIMP, Photoshop or whatever you use.

1.1.2 *Environments*

The command graffito is used to put some text here, usefull to underline important things in long paragraphs.

In addition to common LaTeX environments, this thesis is set to use:

- \begin{aenumerate} to produce an \enumerate with letters instead of numbers, as above;
- \blockcquote[][]{}{} to "produce a citation with reference to author and page" [see 1, p. 111]. If the citation is longer than two rows is indented. This is provided by package csquotes, which settings are in classicthesis-config.tex. The package also provides \enquote{citation} that produces "correct citation style" according to the language in use.
- \ac{} and its variations, defined by package acronyms, provide nice handling for acronyms, like eXtensible Markup Language (XML). They are listed after figures and tables before the abstract.
- the so called semi-dynamic referencing for chapter, sections, figures, etc. A set of command like \myChap{label_key} are provided to produce things like chapter 1. There are also capital versions of the commands (\MyChap{} produces Chapter 1) and commands for equations (\myEq{} produces equation (1.1)).
- references to bibliography are produced in the usual way with \cite{bib_key} [1] and its variations \citeauthor{bib_key}, \citetitle{bib_key} and others.
- figures are handled usually with the code

  ```
  \begin{figure}
  \centering
  \includegraphics[width=\columnwidth]{Images/name.pdf}
  \caption[Short description]{Long description.}
  \label{fig:massConstraintFeasibility}
  \end{figure}
  ```

  which produces figure 1.1.
- tables are produced with

  ```
  \begin{table}[tb]
  \footnotesize
  \centering
  \begin{tabularx}{0.8\textwidth}{llrcl}
  ```

ALGORITHM	PARAMETER	SUGGESTED VALUES		
Any	NFE	10 000	÷	200 000
	Population Size	10	÷	1000
GDE3	DE step size	0.0	÷	1.0
	Crossover rate	0.0	÷	1.0

Table 1.1: Parameters needed for each MOEA used: if the algorithm requires the parameter, a range is given as in literature.

```
\toprule
\tableheadline{l}{Algorithm} &
\tableheadline{l}{Parameter} &
\tableheadlineMore{3}{c}{Suggested Values} \\
\midrule
\tablefirstcol{l}{Any}
& \acs{NFE}  & $10\,000 $ & $ \div $ & $ 200\,000$ \\
& Population Size  &  $10 $ & $ \div  $ & $ 1000$ \\
\midrule
\tablefirstcol{l}{\ac{GDE3}}
& \ac{DE} step size & $0.0 $ & $\div $ & $ 1.0$ \\
& Crossover rate  & $0.0$ & $ \div $ & $ 1.0$ \\
\bottomrule
\end{tabularx}
\caption[Short description]{Long description.}
\label{tab:MOEAandParameters}
\end{table}
```

which produces table 1.1. `\myfloatalign`, `\tableheadline{}{}` and its variation `\tableheadlineMore{}{}{}` and `\tablefirstcol{}{}` are used to give a common style to all tables in the document. They are defined in `classicthesis-config.tex`.[1]

$$\nabla \mathbf{q_s} = U(x,y) - b_t \tag{1.1}$$

These are the main commands used: others are missing and many can be improved or refined. Suggestion are welcome at http://code.google.com/p/

[1] Also do not forget footnotes, which should be placed after the punctuation mark.

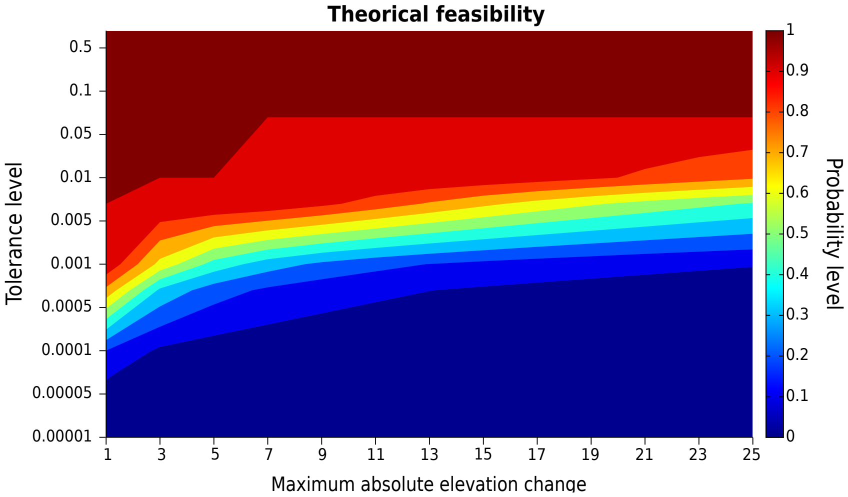

Figure 1.1: Theoretical probability of randomly choose a surface respecting the DEM elevations sum constraint: the blue area represents low probabilities, the red higher ones. On the y-axis there is the tolerance on the accepted mass variation. On the x-axis there is the range of possible variations in meters, e. g. 5 means that each cell can vary from its initial condition of ±5 meters. DEM dimensions are 51×51.

Part II

PRACTICE

Dat show!

2

FINAL REMARKS: OUR CONTRIBUTION AND SOME OPEN QUESTIONS

May the principle be used to represent the landscape evolution process in time and space using Digital Elevation Models?

Would a multi-objective framework be suitable to take advantage of the existing knowledge and to assess the trade-offs among different simplified version of the "optimality" criterion in order to find the proper expression for 3D modeling?

The previous two questions were written in the introduction as the goal of this thesis. Now, after having commented the results, are we able to answer?

2.1 OUR CONTRIBUTION TO THE FIELD OF STUDY

This thesis contributes to the field studying landscape and river evolution as follows:

- as for modeling purposes, it is the first attempt that provides a model to reproduce the 3D features of landscapes and river networks based on multiple criteria optimization;
- as for sistem understanding purposes, it deepened the knowledge about the different formulations of least action principle and the way they affect the features of river networks developing on landscapes.

2.2 PROPOSED IMPROVEMENTS AND OPEN QUESTIONS

Given the large amount of topics touched by this thesis, some relevant issues are proposed to future development.

BIBLIOGRAPHY

[1] R. Bringhurst. *The Elements of Typographic Style*. Version 3.2. Point Roberts, WA, USA: Hartley & Marks Publishers, 2008 (cit. on pp. 4, 16).

[2] D. E. Knuth. “Computer Programming as an Art.” In: *Communications of the ACM* 17.12 (1974), pp. 667–673 (cit. on p. 15).

[3] J. Verne. *Journey to the Center of the Earth*. Classics illustrated. Huge Print Press, 1957. ISBN: 9780758311993 (cit. on p. 3).

Part III

APPENDIX

A APPENDIX EXAMPLE

We have seen that computer programming is an art,
because it applies accumulated knowledge to the world,
because it requires skill and ingenuity, and especially
because it produces objects of beauty.

— Knuth, "Computer Programming as an Art," 1974

A.1 HOW TO WRITE GOOD SOFTWARE: TESTING AND EVALUATING

To test the software we wrote, we rely on the Google C++ Testing framework. The site says that it is the

> Google's framework for writing C++ tests on a variety of platforms (Linux, Mac OS X, Windows, Cygwin, Windows CE, and Symbian). Based on the xUnit architecture. Supports automatic test discovery, a rich set of assertions, user-defined assertions, death tests, fatal and non-fatal failures, value- and type-parameterized tests, various options for running the tests, and XML test report generation.

It is available under the "New BSD License"[1] at `http://code.google.com/p/googletest/`. Within the testing framework, we performed eighteen tests on the core functionalities of the software. 658 assertions have been evaluated to ensure code correctness both while writing is and when deployment has been done.

The model relies on a certain amount of data which must be produced and held during the execution of the software. An optimization run usually lasts for tens of hours and therefore the memory management becomes an important issue. Assessment of its correctness and coherence has been performed using Valgrind.

> Valgrind is an instrumentation framework for building dynamic analysis tools. There are Valgrind tools that can automatically detect many memory management and threading bugs, and profile your programs in detail. It runs on the following platforms: X86/Linux, [. . .]. Valgrind is Open Source / Free Software, and is freely available under the GNU General Public License, version 2.

Performance is also a critical issue: during the optimization, the operations performed by the model are executed millions of times.

[1] `http://opensource.org/licenses/BSD-3-Clause`

Therefore we analyzed performance with a profiler, *gprof*. Profiling allows you to learn where the program spent its time and which functions called which other functions while it was executing. This information can show which pieces of your program are slower than expected and might be candidates for rewriting to make the program execute faster.

The profiling analysis showed that the model spend most of the execution time in extraction of flow direction. Some time is spent also during the depression filling: the amount spent is influnced greatly by the smoothness of the landscape and by the dimension of the depressions. After some improvements, the average perfomance in first experiment has been about 40 ms per function evaluation i. e. per depression filling, flow routing and objectives evaluation.

A.2 CONSTRAINT FEASIBILITY

The model features a constraint called "tectonic condition" based on the hypothesis that the mass gained by the uplift is the same as the total loss of sediment mass from the whole landscape. This requirement also means that the sum of elevations is still the same during the optimization process.

A.2.1 *Technical side*

We were asked to use the LaTeXsystem, and we would have chosen it anyway. As the website http://latex-project.org/ states,

> LaTeXis a document preparation system for high-quality typesetting. It is most often used for medium-to-large technical or scientific documents but it can be used for almost any form of publishing. LaTeXis not a word processor! Instead, LaTeXencourages authors not to worry too much about the appearance of their documents but to concentrate on getting the right content. [It] is based on the idea that it is better to leave document design to document designers, and to let authors get on with writing documents. LaTeXis based on Donald E. Knuth's TeXtypesetting language or certain extensions. LaTeXwas first developed in 1985 by Leslie Lamport, and is now being maintained and developed by the LaTeX3 Project.

We can confirm every single word. Indeed, we let the design of the appearance to "document designers". Particularly we chose the style "Classic Thesis" available at http://code.google.com/p/classicthesis/. It's "an homage to the elements of typographic style" and is inspired by the work of Bringhurst *The Elements of Typographic Style* [1].

A little workflow research on the Internet resources was done to find a simple yet powerful bibliography management system. We ended up choosing *zotero* because of its easy connection with most of articles database and with bibtex. As the website states at http://www.zotero.org/,

> Zotero [zoh-TAIR-oh] is a free, easy-to-use tool to help you collect, organize, cite, and share your research sources. It lives right where you do your work — in the web browser itself.

It has also the capability to retrieve the article itself and to link the file with its database, which effectively become our library.

The word "BibTeX" stands for a tool and a file format which are used to describe and process lists of references, mostly in conjunction with LaTeXdocuments. It is supported by *zotero* as well as Google Scholar, Web Of Science and many other research related resources.

Last but not least, we rely on the plugin structure of *Eclipse* to use it as a LaTeXeditor, thanks to the TeXlipse plugin available at http://texlipse.sourceforge.net/. With the *pdf4Eclipse* plugin we were also able to see the changes in the document appearance each time we saved the TeXfiles, thanks to the automatic compilation that can be triggered in *Eclipse*. We could also rely on the *Google Code* repository to take care of merging the work of both of us and prevent any losses.

Listing A.1: Code snippet with the recursive function to evaluate the pdf of the sum Z_N of N random variables equal to X.

```
std::vector<int> values_of_x(number_of_values_of_x,
    min_value_of_x);
for (unsigned int i = 1; i < number_of_values_of_x; i++) {
    values_of_x[i] = values_of_x[i - 1] + 1;
}
prob_x = 1.0 / number_of_values_of_x;
std::vector<std::vector<double> > p_z;
for (unsigned int idx = 0; idx < p_z.size(); idx++) {
    p_z[idx] = std::vector<double>(
        (max_value_of_x * (idx + 1) - min_value_of_x
            * (idx + 1)) + 1, INIT_VALUE);
}

double prob(int Z, int value_of_z) {
    if (value_of_z < min_value_of_x * Z ||
        value_of_z > max_value_of_x * Z) {
            return 0.0;
    }
    if (value_of_z < min_value_of_z ||
        value_of_z > max_value_of_z) {
            return 0.0;
    }
    int idx_value_of_z = -(min_value_of_z - value_of_z);
    int idx_N = Z - 1;
    if (p_z[idx_N][idx_value_of_z] == -2.0) {
        if (Z > 1) {
            double pp = 0.0;
            for (unsigned int i = 0; i < number_of_values_of_x; i
                ++) {
                pp += prob(Z - 1, value_of_z - values_of_x[i], p)
                    ;
            }
            p_z[idx_N][idx_value_of_z] = prob_x * pp;
        } else {
            if (Z == 1) {
                for (unsigned int j = 0; j <
                    number_of_values_of_x; j++) {
                    if (value_of_z == values_of_x[j]) {
                        p_z[idx_N][idx_value_of_z] = prob_x;
                        break;
                    }
                }
            }
            if (p_z[idx_N][idx_value_of_z] == INIT_VALUE) {
                p_z[idx_N][idx_value_of_z] = 0.0;
            }
        }
    }
    return p_z[idx_N][idx_value_of_z];
}
```