Template for M.Sc. 1st Semester Paper

Jacob Møller†, No One‡, †Aalborg University, Sound and Music Computing, Dept. of Arch., Design & Media Tech., jmhh09@student.aau.dk ‡No where, No Where Land.

***Abstract*—When studying M.Sc.EE. at Aalborg University it is required to hand in a scientific paper on the 1st semester. Since this is the case, it is appropriatae to design a template for getting such a paper started. This is exactly what this document contains. The template is developed with purpose of focusing on studying more focused once the semester finally begins. By designing such a template now, the student will be able to focuse entirely on the mathematical subjects presented in lectures and in the consequential group work. In this paper, some random mathematical examples will be given, they are presented in a non-coherent context. Through examples, it is demonstrated that the sufficient packages is included in this project. Furthermore, this gives an indication of how the template looks and especially how math and tikz figures is presented. The code is available online.**

***Index Terms*—Paper, Template, IEEEtran, LaTeX.**

I. Introduction

THE original template is intended to serve as a "starter file" for IEEE journal papers produced under LaTeX using IEEEtran.cls version 1.8 and later [1]. Since this project is build from the IEEE-template which contains one "Master" LaTeX-file, it has been found convenient to divide this template into sub-files. The 1st semester paper template is build with the directories `main`, `set`, `sec`, `img` and `bibtex`. The `preamble.tex` has been cleaned from original comments and a few additional packages has been included. These changes are marked with comments inside the preamble. Feel free to ask any questions regarding this template, by email.

jmh
August 25, 2014

II. Content

In this paper, the syntax and typesetting is considered, for building a paper with LaTeX. By splitting this into sub-files it is easy to work with the project and flexible to change the content without a change of the structure. However, it might be easier to have the structure in another way, once the project evolves. In the following, each example will be defined in the subsection to this `content` section. Firstly, a short overview of the structure will be given followed by some tests which is carried out as examples of how it is possible to write a descent scientific paper.

A. File Structure

Since the template is divided into sub-files, these serves the following purpose. The root directory contains a Makefile, master file and two MATLAB test files. The master file contains all inputs to paper and also the "Title" and "Author" definitions, directly. The root dir also contains the following directories.

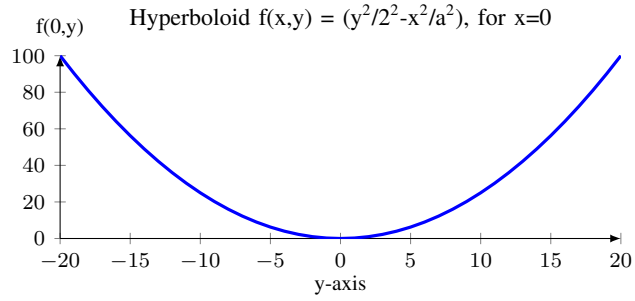

Fig. 1: Graph Plot

1) Main Directory: The directory `main` has been created to maintain the structure of a standardized paper. It contains the main files which are defined as inputs in the master file `master.tex`. This way it should be an easy "starter" for initializing the project and one can decide whether to input into these files or directly into the master file, when adding content.

2) Set Directory: The directory `set` contains a macro-file and the preamble. The macro file contains the desired syntactic changes to make things easier along the path of writing this paper. This IEEEtran-template [1] can be used for several different types of IEEE-papers, this can be changed by setting the document class in the preamble. The greatest change in the preamble has been implementing `Tikz`, and defining a `pgfplotssetup` for matching the template.

3) Bibtex Directory: The `bibtex` directory contains the necessary files for citations. The `sources.bib` is the place to add new litterature sources. Note that it might be necessary to run "make clean" followed by "bibtex master.tex" a few times, if it is not compiling as expected.

4) Img Directory: The `img` directory contains all figures. The figure format are defined in the preamble, from line 30. This is also the directory where Tikz-files are placed. Each Tikz-file is usually followed by a tex-wrapper with a similar name. This allows for simply applying an input of the wrapper into a file, when inserting a figure into the paper. This process is also automated by utilizing `matlab2tikz.m`. This is further elaborated in **??**. This directory contains another directory called `tikz` this is used as a "cache" for pre-compiled tikz figures, in order to reduce compile time. This makes it necessary to remove the `img/tikz/*` content in order to insert/compile new tikz figures.

B. Examples

1) Figures: .The handling of figures by means of plotting can be done in the usual way by importing the specified formats as figures inside the latex file. However, the approach

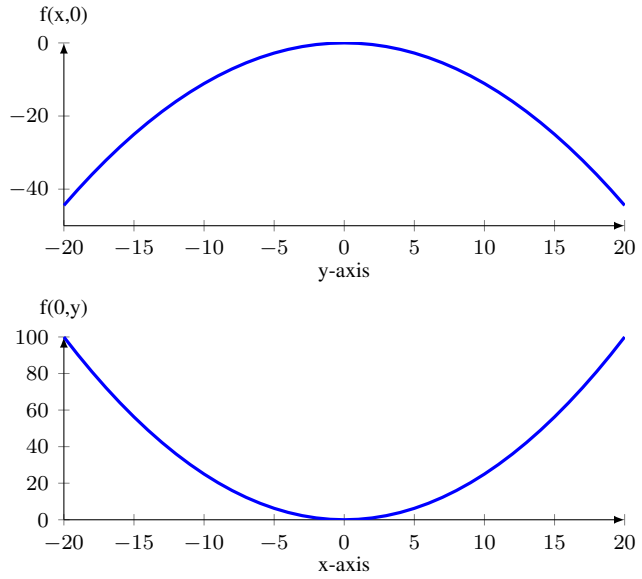

Fig. 2: Comparison of $f(x, y)$ when $x = 0$ and $y = 0$

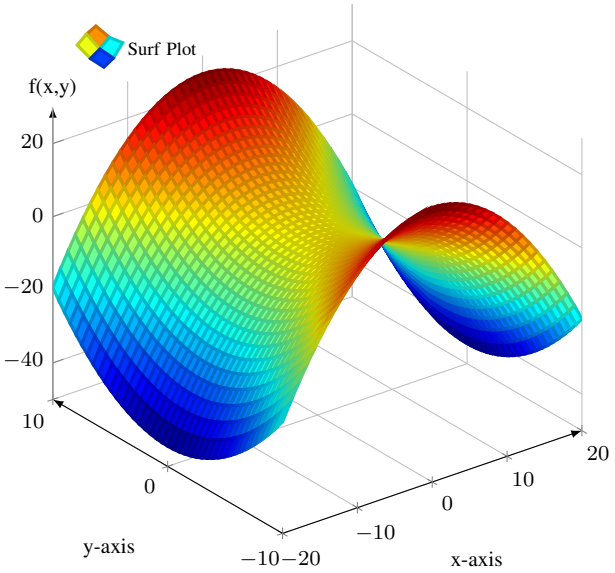

Fig. 3: $f(x, y)$ in 3 dimensions.

in this template is to utilize tikz and pgfplots for plotting. Furthermore, a wrapper is attached to this template for utilizing the MATLAB library called `matlab2tikz`. `matlab2tikz` has the ability to convert a MATLAB figure into a tikz-file [2]. It is possible to create the LaTeXfigure environment in a tex-file which inputs the generated tikz-figure. This can be done in one line from MATLAB, by using the file called `fig2tikz` [3]. `fig2tikz` creates a tikz-file and a tex-file with similar names. The tex-file is the only file that that needs to be input into LaTeXby simply typing
`input img/latexfile.tex`. A demo file which shows this procedure is attached to this project. The demo file is named `demofile_m2t.m` and is placed in the root directory along with `fig2tikz.m`. All the figures in this paper is created by `demofile_m2t.m`. Note that `matlab2tikz` has to be installed correctly for this procedure to work [2].

2) Math: The handling of math will be tested by applying a random example in the following. Consider the linear combination $\mathbf{a}' = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix}$, Since

$$\mathbf{a}'\mathbf{X} = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_4 \end{bmatrix} = X_1 \tag{1}$$

and

$$\mathbf{a}'\mu = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_4 \end{bmatrix} = \mu_1 \tag{2}$$

$$\mathbf{a}'\mathbf{\Sigma}\mathbf{a} = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \sigma_{1n} & \sigma_{2n} & \cdots & \sigma_n \\ \sigma_{12} & \sigma_{22} & \cdots & \sigma_{2n} \\ \cdots & \cdots & \ddots & \cdots \\ \sigma_{1n} & \sigma_{2n} & \cdots & \sigma_n \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_4 \end{bmatrix} = \sigma_{11} \tag{3}$$

III. Conclusion

So far everything seems to be working. There is some challenges with wide subplots which stretches over both columns. Some more testing is to be done.

Appendix A
Proof that there is no appendix

There is no text here.

Appendix B
Extra Appendix

This appendix is also empty.

Acknowledgment

The Author would like to thank Michael Shell who wrote the original template [1] and also Nico Schlömer who designed `matlab2tikz` [2].

References

[1] M. Shell. (2008) IEEEtran homepage. [Online]. Available: http://www.michaelshell.org/tex/ieeetran/

[2] N. Shlömer. (2012) Github homepage. [Online]. Available: https://github.com/nschloe/matlab2tikz/

[3] J. Møller. (2014) Github homepage. [Online]. Available: https://github.com/junilyd/LaTeX_article_template/

Jacob Møller Jacob Møller was born in Aalborg, Denmark, in September 1982. He received the B.Sc.EE degree from Aalborg University, Aalborg, Denmark. He is currently pursuing the M.Sc degree in Sound and Music Computing at the Department of Architecture Design and Media Technology, Aalborg University. He has a practical background within Music and Sound Engineering.