Documentation of Shape Analyzer

Version 1.0

Emanuel Laude und Zorah Lähner

December 9, 2014

Contents

CHAPTER 1

Introduction

This LaTeX template is designed for the creation of thesis documents (bachelor, master, phd) and targets both beginner and experienced users of LaTeX. It supports all basic functionality and requirements of a technical document such as the inclusion of graphics, math, tables, references, bibliography and much more. In contrast to a standard LaTeX document this template not only loads all state of the art packages (`preamble/packages.tex`) to provide the best functions for each task, but also includes a separate document for the style/layout of the document (`preamble/style.tex`). It therefore tries to separate functionallity and layout as much as possible. And the best, everything is documented in the code and furthermore in a separate documentation file (`TemplateDocumentation.pdf`)

This document shows in **??** a general tutorial for LaTeX with links to the documentation for further tasks. You can view the underlying code in file `content/demo/latextutorial.tex` or in this document in **??**.

The code of the template itself is documented in `TemplateDocumentation.pdf`.

CHAPTER 2

Theory

Duis porta orci. Integer eu arcu at enim tempus facilisis. Pellentesque dignissim orci sed est. Etiam elementum laoreet mi. Donec nunc sapien, dictum in, tristique sed, aliquam vitae, massa. Morbi magna magna, vestibulum tempor, lobortis non, convallis nec, nibh. In sed nibh. Suspendisse adipiscing dictum pede. Suspendisse non augue. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Pellentesque lacinia, velit sed commodo convallis, diam dolor consequat ligula, a scelerisque quam neque et purus. Praesent vel augue. Sed lectus leo, dignissim eget, vulputate eu, auctor ut, nulla. Vivamus a quam. Nulla tellus. Pellentesque tempor pulvinar nunc.

2.1 Section heading

Fusce vitae quam eu lacus pulvinar vulputate. Suspendisse potenti. Aliquam imperdiet ornare nibh. Cras molestie tortor non erat. Donec dapibus diam sed mauris laoreet volutpat.

$$J_f(a) := \frac{\partial f}{\partial x}(a) := \frac{\partial(f_1, \ldots, f_m)}{\partial(x_1, \ldots, x_n)}(a) := \left(\frac{\partial f_i(a)}{\partial x_j}\right)_{i=1,\ldots,m;\ j=1,\ldots,n} \tag{2.1}$$

2.1.1 Subsection heading

Aliquam dignissim laoreet mi. Duis pulvinar nulla id velit euismod fringilla. In hac habitasse platea dictumst. Vestibulum dolor tellus, gravida a, condimentum nec, laoreet ut, nisi. Aenean lacus purus, tristique in, sagittis sit amet, pellentesque non, neque. Sed egestas nibh vitae velit. Nunc adipiscing. Donec sed lectus. Donec ultrices lacus nec orci. Fusce sit amet nulla. Suspendisse vulputate, mi nec nonummy sodales, ligula massa molestie est, at sagittis nisi est in leo.

Subsubsection heading

Mauris rutrum volutpat massa. Suspendisse potenti. Nam varius. Fusce nec leo. Morbi vestibulum augue ac justo. Vivamus in odio in turpis pharetra blandit. Mauris aliquet ullamcorper libero. Integer quam mi, venenatis ut, tristique ut, tempus at, ipsum. Donec malesuada. In quis tellus et ipsum hendrerit imperdiet. Vivamus sapien ipsum, suscipit sed, gravida a, lacinia laoreet, ipsum. Quisque augue. Nulla justo enim, auctor at, hendrerit nec,

porttitor non, urna. Duis tincidunt tincidunt dui. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Suspendisse potenti. Aenean sit amet mauris. In luctus purus nec lorem. Proin orci tortor, tempus sit amet, molestie hendrerit, placerat egestas, tellus.

CHAPTER 3

Customs

3.1 Exceptions

There are two types of exceptions you might want to throw: Exceptions (**??**) and Errors (**??**). The first one is kind of optional, it includes all normal exceptions C++ provides, the second one is for non-fatal problems that might occur and that can actually be handled.

3.1.1 Errors

3.1.2 Exceptions

- **Throw when** you have special cases that are due to previous faulty programming or unexpected errors
- **Documentation** not necessary

Any exception will be caught by the main program and showed within an error message. The program will then terminate. In order to make debugging easier a short description of the problem is useful.

```
if(s >= points_->size()) {
        throw invalid_argument("Source point (" + to_string(s) + ")
        larger than number of points (" + to_string(points_->size()) + ")
        in " + __PRETTY_FUNCTION__);
}
```

CHAPTER 4

Results

List of Figures

List of Tables

APPENDIX A

First chapter of appendix

A.1 Parameters

Todo list