

Faculty of  
Cognitive Science

# Bachelor Thesis

Romeo: Scripting Environment with interactive Visualizations



**Author:** Simon Danisch  
sdanisch@email.de

**Supervisor:** Prof. Dr.-Ing. Elke Pulvermüller

**Co-Reader:** Apl. Prof. Dr. Kai-Christoph Hamborg

**Filing Date:** 01.02.2014

# I Abstract

This bachelor thesis is about writing a simple scripting environment for scientific computing, with focus on visualizations and interaction. Focus on visualization means, that every variable can be inspected and visualized at runtime, ranging from a textual representation to complex 3D scenes. Interaction is achieved by offering simple GUI elements for all parts of the program and the visualizations. All libraries are implemented in Julia and modern OpenGL, to offer high performance, opening the world to scientists who have to work with large datasets. Julia is a novel high-level programming language for scientific computing, promising to match C speed, making it the optimal match for this project. -This section needs more work, and should probably be written in the end

## II Table of Contents

<b>I</b>	<b>Abstract</b>	<b>I</b>
<b>II</b>	<b>Table of Contents</b>	<b>II</b>
<b>III</b>	<b>List of Figures</b>	<b>IV</b>
<b>IV</b>	<b>List of Tables</b>	<b>V</b>
<b>V</b>	<b>Listing-Verzeichnis</b>	<b>V</b>
<b>VI</b>	<b>List of Abbreviations</b>	<b>VI</b>
<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Field of Research . . . . .	1
1.2	Problem . . . . .	1
1.3	Problem Solutions and Evaluation . . . . .	1
1.4	Outlook . . . . .	1
1.5	Motivation and Problem Description . . . . .	1
1.6	Used Technologies . . . . .	3
1.6.1	Julia . . . . .	3
1.6.2	OpenGL . . . . .	3
1.7	Similar Work . . . . .	3
1.7.1	Other Languages . . . . .	3
1.7.2	Ipython Notebook . . . . .	3
1.7.3	Matlab . . . . .	3
1.7.4	Mathematica . . . . .	3
1.7.5	Other Graphic acceleration APIs . . . . .	3
<b>2</b>	<b>Background</b>	<b>4</b>
<b>3</b>	<b>Design</b>	<b>5</b>
3.1	Architecture Overview . . . . .	5
3.2	OpenGL Wrapper . . . . .	5
3.3	Window System . . . . .	5
3.4	Event System . . . . .	5
3.5	Visualization System . . . . .	5
<b>4</b>	<b>Results and Analysis</b>	<b>5</b>
4.1	Performance Analysis . . . . .	5
4.2	Extendability Analysis . . . . .	5
4.3	Usability Analysis . . . . .	5
<b>5</b>	<b>Conclusion</b>	<b>5</b>
5.1	Discussion . . . . .	5
5.1.1	Performance . . . . .	5
5.1.2	Extensability . . . . .	5

5.1.3	Usability . . . . .	5
5.2	Future Work . . . . .	5
<b>Appendix</b>		<b>I</b>
<b>A</b>	<b>GUI</b>	<b>I</b>

### III List of Figures

Abb. 1	Volume Visualization . . . . .	1
Abb. 2	Prototype . . . . .	I

## **IV List of Tables**

## **V Listing-Verzeichnis**

## **VI List of Abbreviations**

**GUI**      Graphical User Interface

# 1 Introduction

This Bachelor Thesis is about writing a scripting environment for scientific computing. Its aims are to be intuitive to use, with deeply integrated visualizations and Graphical User Interface (GUI) elements, suitable for large datasets and with a modular, open source core, which is easy to extend. The introduction is structured in the following way. I will start with describing the field of research with its challenges. From the challenges I will identify the ones, that I want to solve in this Bachelor Thesis. Finally I will conclude with how I intend to solve the problems, how to measure the success and give an outlook on the structure of the entire Bachelor Thesis.

## 1.1 Field of Research

The research field is developing tools for scientific computing.

## 1.2 Problem

Having a single platform in which you can do the research.

## 1.3 Problem Solutions and Evaluation

Push code everyday!

## 1.4 Outlook

Make stuff even more awesome!

## 1.5 Motivation and Problem Description



Figure 1: different visualizations of  $f(x,y,z)=\sin(x/15)+\sin(y/15)+\sin(z/15)$ , visualized with Romeo. From left to right: Isosurface, isovalue=0.76, Isosurface, isovalue=0.37, maximum value projection

When one tries to visualize a volume described by  $f(x,y,z)=\sin(x/15f_0)+\sin(y/15f_0)+\sin(z/15f_0)$ , the limitations of the human mind becomes clear. While a computer doesn't have the



creativity of humans yet, the task of visualizing this volume is a no-brainer for a computer (see Figure 1). In science, there are many comparable problems, which are unsolvable without the aid of a computer. For example making sense of one petabyte of data, predicting the folding of a novel protein, simulating black holes and many more. This naturally leads to computers being an essential part of state of the art research. Researchers have unique demands for their tools, which makes writing software for scientific computing an interesting research field. Consider, that researchers mostly do not have a background in computer science, some problems need a lot of computational power and due to the novelty of some problems hand tailored solutions are needed. In addition, high interactivity is a crucial feature, since a lot of research is a creative process which involves exploring many alternative solutions. This bachelor thesis is about writing a scripting environment with advanced visualization capabilities, while keeping the previous mentioned demands of scientific computing in mind. The resulting demands for the scripting environment are as follows.

No background in computer science puts a strong constraint on the used programming language. It should be high level and easy to learn as researchers should not be forced to waste their time with fixing memory corruptions or obscure language constructs. The entire suite should be open source and written in one language, to allow the researchers to modify every part easily, enabling them to create a solutions which is hand tailored to their particular problem. The two constraints of writing even the core libraries in one language and having also research areas with high demand for computational power means, that in addition to being high-level the chosen language must also be fast. As an example, the high level language Python can be substantially slower than a high-performance language like Fortran. In extreme cases, Python is up to 1155 times slower than Fortran [?], which means if a computation needs 1 day with Fortran, it could take 3 years within Python, rendering the problem intractable within Python.

Finally, the scripting environment should offer simple GUI elements like sliders, to enable the researcher a simple way to interact with the parameters in his script.

These considerations lead to the 3 main design choices: Julia, a novel scientific computing language, is chosen as the main language, Modern OpenGL for high performance graphic rendering and writing generic,Reusable open source code for a high amount of customizability. The feature set consists of parametrized scientific visualizations, simple GUIs for the parameters and a simple script editor.

-This section needs more quotation, to integrate it into an ongoing research problem. Also, some formulations have to be rewritten.

## **1.6 Used Technologies**

### **1.6.1 Julia**

### **1.6.2 OpenGL**

## **1.7 Similar Work**

### **1.7.1 Other Languages**

### **1.7.2 Ipython Notebook**

### **1.7.3 Matlab**

### **1.7.4 Mathematica**

### **1.7.5 Other Graphic acceleration APIs**

## 2 Background

## **3 Design**

### **3.1 Architecture Overview**

### **3.2 OpenGL Wrapper**

### **3.3 Window System**

### **3.4 Event System**

### **3.5 Visualization System**

## **4 Results and Analysis**

### **4.1 Performance Analysis**

### **4.2 Extendability Analysis**

### **4.3 Usability Analysis**

## **5 Conclusion**

### **5.1 Discussion**

#### **5.1.1 Performance**

#### **5.1.2 Extensability**

#### **5.1.3 Usability**

### **5.2 Future Work**

## Appendix

### A GUI

A nice Appendix.

#### Screenshot

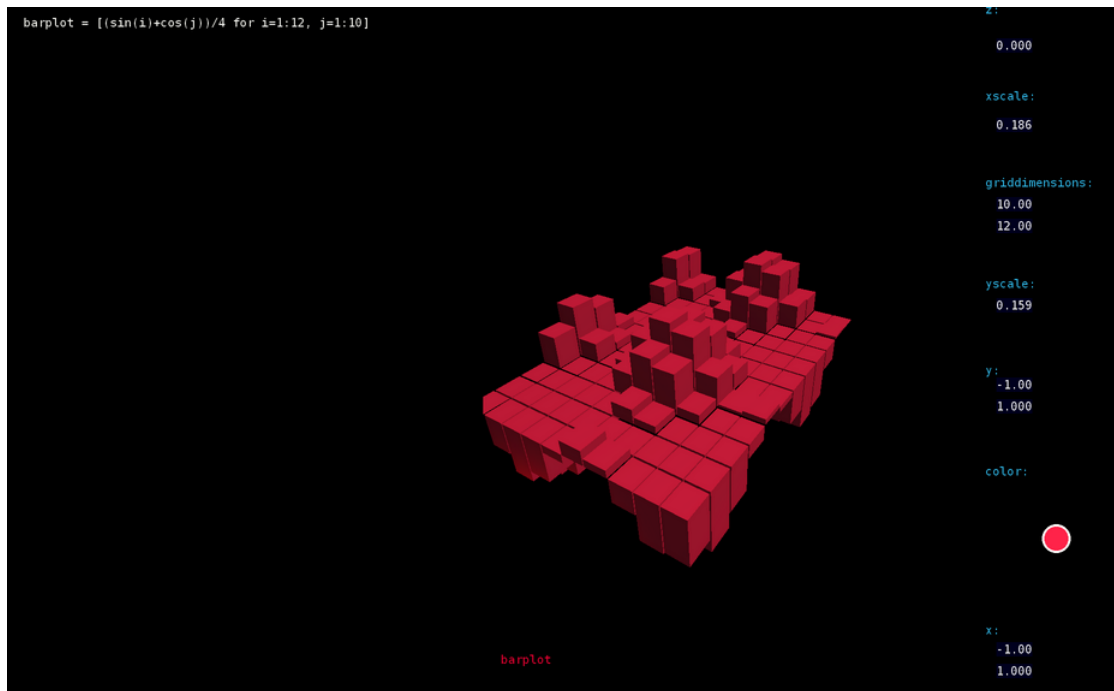


Figure 2: Screenshot of the prototype. Left: evaluated script, middle: visualization of the variable `barplot`, right: GUI for editing the parameters of the visualization

# Official Statement

I hereby guarantee, that I wrote this thesis and didn't use any other sources and utilities than mentioned.

Date:

.....

(Signature)