A Framework for Individualised Mathematical Assignments with Solutions in LaTeX

Bachelor Thesis Presentation

Hochschule Ruhr West University of Applied Sciences

Presented by: Thomas Flinkow (10009347)
Supervisors: Prof. Dr. rer. nat. Jürgen Vorloeper
Prof. Dr. rer. nat. Ioannis Iossifidis

21 April 2022

Introduction

Motivation

- individualised assignment benefits students and teachers
- creation of individualised tasks in LaTeX is time-consuming
- LuaLaTeX allows to embed Lua code directly into LaTeX documents

Goal

- simplify the process of creating individualised assignments
- perform (symbolic) computation directly in LaTeX documents, avoiding separation between calculation and content
- evaluate an approach making third-party libraries accessible from within LaTeX documents using LuaLaTeX

Existing Approaches – MATLAB

- MATLAB provides rich symbolic computation features
- MATEX[1] project offers a free web interface

```
clear all
fID = fopen('Aufgabe3.tex', 'wt');
%% Aufgabenstellung formulieren
Text = ['\\textbf{Aufgabe}: L\\"osen Sie die Gleichung \n \\[ \n '];
fprintf(fID, Text)
```

- LaTeX is merely the output

[1] https://lx4.mint-kolleg.kit.edu/MATeX/index.php

Existing Approaches – Python

- LaTeX packages on CTAN for integrating SymPy[2] and SageMath[3]
- feature-rich with an intuitive API

```
\documentclass{article}
\usepackage{sagetex}
\begin{document}
\[\sage{integrate(exp(-x**2), (x, -oo, oo))\]
\end{document}
```

Listing 1: Evaluating $\int_{-\infty}^{\infty} \exp(-x^2)\,dx$ in LaTeX with `sagetex`

- no separation between calculation and LaTeX
- requires multiple compilations (multiple LaTeX runs)

[2] `macros/latex/contrib/sympytexpackage`
[3] `macros/latex/contrib/sagetex`

Approach

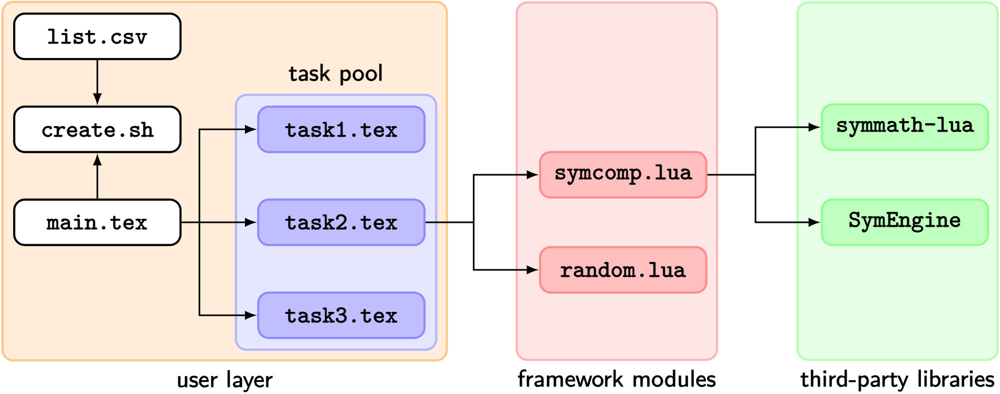


Figure 1: Overview of the framework

Symbolic Computation

- differentiation, integration, manipulation and evaluation of expressions
- framework module implemented in C++20 and Lua
- provides unified abstraction over multiple third-party libraries

```lua
local f = symcomp.expr("x^2-4")
local res = symcomp.solve(f, "x")               -- result: -2 and 2

local g = symcomp.expr("1/3*x^2")
local indefinite = symcomp.integrate(g, "x")     -- result: (x^3)/9
local definite = symcomp.integrate(g, "x", 0, 3) -- result: 3
```

Symbolic Computation – Abstraction

```
function symcomp.det(m)   [symcomp.lua]
```

```
static int det(::lua_State* L)   [wsymengine.cpp]
```

```
symcomp::ExprRep symcomp::Determinant(const std::string& what)   [mathematics.cpp]
```

```
RCP<const Basic> DenseMatrix::det() const   [SymEngine]
```

Figure 2: A framework-provided function eventually using SymEngine.

Symbolic Computation - Representing Mathematical Expressions

Goals

- manipulate and work with mathematical expressions

```lua
local A = symcomp.expr("[-3, 5] [2, -1]")
local l = symcomp.expr("lambda")

local res = symcomp.scalarMul(l, A)
```

- render mathematical expressions in LaTeX

```latex
\begin{equation*}
    \lambda\scprint{A}=\scprint{res}
\end{equation*}
```

$$\lambda \begin{bmatrix} -3 & 5 \\ 2 & -1 \end{bmatrix} = \begin{bmatrix} -3\lambda & 5\lambda \\ 2\lambda & -\lambda \end{bmatrix}$$

Symbolic Computation - Representing Mathematical Expressions

Solution

- internally represent mathematical objects as string tuples
- one internal representation for further manipulations and for interacting with third-party libraries (e.g. `f(1) == "sin((x^2)/2)"`)
- a LaTeX representation for display (e.g. `f(2) == "\\sin\\frac{x^2}{2}"`)

```lua
local f = symcomp.expr("sin((x^2)/2)")
local df = symcomp.diff(f, "x")

assert(df(1) == "x*cos((x^2)/2)")        -- internal representation
assert(df(2) == "x\\cos\\frac{x^2}{2}") -- LaTeX representation
```

Symbolic Computation – Choice of Libraries

	Advantages	Disadvantages
Symbolic Lua	Lua, symbolic integrals	slower, different API
SymPy	feature-rich	Python
SageMath	feature-rich	Python
SymEngine	C++, fast, feature-rich	under development
GiNaC	C++, fast	high-performance applications
		not well suited for teaching[4]
ViennaMath	C++	complex API, last release in 2012
		integrals cannot be sufficiently evaluated[5]
SymbolicC++	C++	no LaTeX output, last release in 2010

Table 1: Evaluation of third-party libraries for symbolic computation

[4] non-deterministic, non-simplified output

[5] ViennaMath uses Gaussian quadrature rules; only inbuilt rule is 1-point

Randomisation

- creation of randomised matrices, polynomials, numbers and arbitrary expressions
- reasonable bounds for numbers

```lua
local v = random.oneof({ "1/2*x", "2/3*x" }) -- choosing randomly from a set
local f = random.polynomial(4)              -- a polynomial p with deg(p) = 4
```

- generally: naïve text replacement

```lua
local e = random.create("a*x**n+b",
                        {
                           { "a", random.rational() },
                           { "n", random.integer(2, 5) },
                           { "b", random.oneof({ 4, 6, 10 }) }
                        })
```

- implemented in Lua only (sufficiently fast, no need for third-party libraries)

Randomisation

Problem

- not uncommon: LaTeX $\rightarrow$ biber $\rightarrow$ LaTeX $\rightarrow$ LaTeX to accommodate for bibliographic references, table of contents etc.
- Lua code is executed each time LuaLaTeX runs
- different random values each time!

Solution

- seeding Lua's PRNG with the respective student ID
- no need to maintain state (aux files) between compilations
- deterministic output

Quality Assurance

Code Quality

- C++20, C++ Core Guidelines and clang-tidy
- unit tests for most C++ functions
- unit tests for all symbolic computation Lua functions
- frequent assertions in code to check internal assumptions

Documentation

- code comments to explain intent
- function-level comments explaining what the function does
- additional Markdown documentation with descriptions, usage examples and possible errors

Examples – Linear Algebra Task

```
\begin{luacode*}
A = symcomp.matrix("[0, 1] [-2, -3]")
I2 = symcomp.identityMatrix(2)

lambdaI2 = symcomp.scalarMul("lambda", I2)
sub = symcomp.matrixSub(A, lambdaI2)
det = symcomp.det(sub)

ev = symcomp.eigenvalues(A)
\end{luacode*}

\question
Find all eigenvalues of the matrix \[A=\scprint{A}.\]

\begin{solution}
  The characteristic equation \[\det(A-\lambda I_2)=0\] is
  \begin{align*}
    &0=\det\left(\scprint{A}-\scprint{lambdaI2}\right)\\
    &0=\det\left(\scprint{sub}\right)\\
    &0=\scprint{det}.
  \end{align*}
  The two eigenvalues are
  \(\lambda_1=\scprint{ev(1)}\) and \(\lambda_2=\scprint{ev(2)}\).
\end{solution}
```

1. Find all eigenvalues of the matrix

$$A = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}.$$

Solution: The characteristic equation

$$0 = \det(A - \lambda I_2)$$

is

$$\begin{aligned} 0 &= \det\left(\begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}\right) \\ 0 &= \det\left(\begin{bmatrix} -\lambda & 1 \\ -2 & -3-\lambda \end{bmatrix}\right) \\ 0 &= \lambda^2 + 3\lambda + 2. \end{aligned}$$

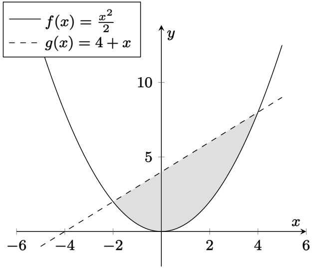
The two eigenvalues are $\lambda_1 = -2$ and $\lambda_2 = -1$.

Examples – Calculus Task

2. How big is the parabolic segment between the parabola $f(x) = \frac{x^2}{2}$ and the line $g(x) = 4 + x$?
Sketch a graph to visualize the desired area.

Solution: The functions intersect at $P_1 = (-2, 2)^T$ and at $P_2(4, 8)^T$. Thus, the area is

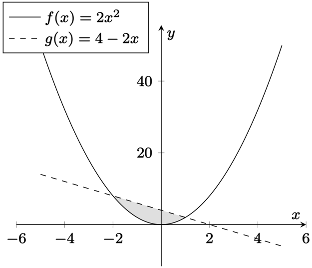
$$
\begin{aligned}
A &= \int_{-2}^{4} g(x) - f(x)\,\mathrm{d}x \\
&= \int_{-2}^{4} 4 + x + \frac{-1}{2}x^2\,\mathrm{d}x \\
&= \left[\frac{1}{6}x(24 + 3x - x^2)\right]_{-2}^{4} = 18.
\end{aligned}
$$



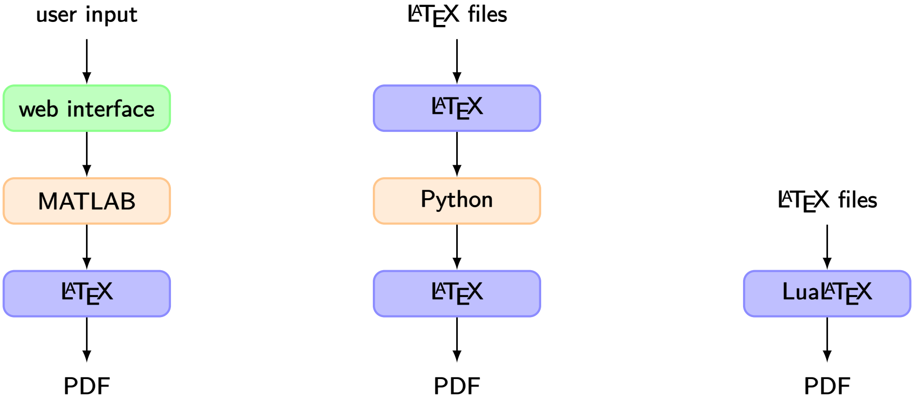
2. How big is the parabolic segment between the parabola $f(x) = 2x^2$ and the line $g(x) = 4 - 2x$?
Sketch a graph to visualize the desired area.

Solution: The functions intersect at $P_1 = (-2, 8)^T$ and at $P_2(1, 2)^T$. Thus, the area is

$$
\begin{aligned}
A &= \int_{-2}^{1} g(x) - f(x)\,\mathrm{d}x \\
&= \int_{-2}^{1} 4 - 2x + 2x^2\,\mathrm{d}x \\
&= \left[x^2(4 + x - \frac{2}{3}x^2)\right]_{-2}^{1} = 9.
\end{aligned}
$$



Improved Workflow



(a) MATEX (b) SymPy / SageMath (c) ours

Figure 3: Comparison of workflows

Future Improvements

- extending symbolic computation functionality
- support solving systems of linear equations
- add more tasks to the task pool
- add easier plotting
- integration with e-learning platforms
- port the framework to Windows
- gather feedback from students and teachers