

TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Roman Yakovlev 206440IAIB

**VISUAL CALCULATOR FOR LINEAR ALGEBRA, LINEAR
AND NON-LINEAR PROGRAMMING**

Bachelor's Thesis

Supervisor: Mati Väljas
PhD

Tallinn 2024

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Roman Yakovlev 206440IAIB

**VISUAALNE KALKULAATOR LINEAARSE ALGEBRA,
LINEAARSETE JA MITTELINEAARSETE
PLANEERIMISÜLESANNETE JAOKS**

Bakalaureusetöö

Juhendaja: Mati Väljas
PhD

Tallinn 2024

Author's Declaration of Originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Roman Yakovlev

11.01.2024

Abstract

The goal of the given thesis is to implement a web-application for a visual and interactive calculator focused on linear algebra and mathematical programming. The system allows users to create and manipulate with matrices. Most manipulations happen via predefined functions.

In analysis, description for tasks that the calculator is intended to be used for is provided and technical analysis over what technologies are to be used can also be found.

Result of this thesis is a complete Vue.js application where user can create matrices in 3 different ways, and use 17 different functions to manipulate them.

The thesis is written in English and is 38 pages long, including 7 chapters, 1 figure and 0 tables.

Annotatsioon

visuaalne kalkulaator lineaarse algebra, lineaarsete ja mittelineaarsete planeerimisülesannete jaoks

Käesoleva lõputöö eesmärk on rakendada veebirakendus visuaalse ja interaktiivse kalkulaatori jaoks, mis keskendub lineaaralgebrale ja matemaatilisele programmeerimisele. Süsteem võimaldab kasutajatel luua ja manipuleerida maatriksitega. Enamik manipulatioone toimub etteantud funktsioonide kaudu.

Analüüsis on esitatud kirjeldus ülesannete kohta, milleks kalkulaatorit on mõeldud kasutada, ning tehniline analüüs selle kohta, milliseid tehnoloogiaid kasutada, on samuti leitav.

Lõputöö tulemus on täielik Vue.js rakendus, kus kasutaja saab luua maatriksid 3 erineval viisil ja kasutada 17 erinevat funktsiooni nende manipuleerimiseks.

Lõputöö on kirjutatud Inglise keeles ning sisaldab teksti 38 leheküljel, 7 peatükki, 1 joonist, 0 tabelit.

List of Abbreviations and Terms

UI	User Interface
GUI	Graphical User Interface
Desktop	Personal
Web	Synonym for Internet
Mobile	Smartphones and mobile phones
Front-end	Part of a web application that is presented to a user
Back-end	Part of a web application that stays on a server and allows connection to front-end
Framework	Software with generic functionality that can be changed by user code to create application specific software
JS	JavaScript programming language
TS	TypeScript programming language
Tech-stack	Selection of technologies used to build particular application
.XLSX	Spreadsheet file format introduced by Microsoft in 2006
.CSV	Simple text based format for storing tabular data

Table of Contents

1	Introduction	9
2	Existing Solutions	10
2.1	Education-oriented software	10
2.1.1	Symbolab	10
2.1.2	WolframAlpha	11
2.1.3	Matrixcalc	12
2.1.4	Mathway	12
2.1.5	Common traits	12
2.2	Professional-oriented software	12
2.2.1	MATLAB	13
2.2.2	Julia	13
2.2.3	Python with NumPy	13
2.2.4	Common traits and critique	14
3	Analysis	15
3.1	Problems that the system is intended to solve	15
3.2	Requirements	15
3.3	Technological choices	16
3.3.1	React	16
3.3.2	Angular	16
3.3.3	Vue.js	17
3.3.4	Final choice	17
3.3.5	Choice of back-end frameworks	17
4	Results	18
4.0.1	Creating matrices	18
4.0.2	Manipulating matrices	19
4.0.3	Exporting matrices	20
5	Testing	21
5.1	First testing round	21
5.2	Second testing round	21
5.2.1	Negatives	21
5.2.2	Suggestions	22
5.2.3	Fixes	22

6	Future Plans	23
7	Summary	24
	References	25
	Appendix 1 – Non-Exclusive License for Reproduction and Publication of a Graduation Thesis	28
	Appendix 2 – Corridor Testing Protocol	29
	Appendix 3 – Calculator Interface	31

List of Figures

1	<i>First iteration of Beale's algorithm expressed in MATLAB code</i>	14
---	--	----

List of Tables

1. Introduction

While studying or teaching linear algebra or mathematical programming, most common way to represent matrices or tables and calculation with them is by just writing them manually on paper or chalkboard. Despite tremendous flexibility and freedom, it has some significant disadvantages - time and cognitive effort. Due to that, mistakes also have a very high cost - being careless while manually calculating matrices might lead to having to do all the work again. These issues were recognised long before this thesis, so there are already some solutions on the market.

Modern tools, that are commonly used to simplify tasks of linear algebra and mathematical programming, are very versatile and able to solve wide variety of tasks, from simple operations on matrices to building complicated simulations of physical/economical/etc. processes with beautiful visuals and interactive interfaces. That kind of power makes these tools desirable by professionals (or often entire corporations or institutions) of the respective fields. But, with all the versatility and strength comes two corresponding drawbacks: cost and complexity. Software like MATLAB[2][3] or Artelys Knitro[1] require purchasing expensive licenses, as well as investing a lot of time into learning a tool itself. On the other hand, there are very simple, usually web-based applications that seem to target more general audience and that have form-factor of a search engine or a calculator. Examples of tools like that are Wolfram Alpha and Symbolab.com. They are very simple to use, but their free functionality might be insufficient for purposes of some problems of, for example, mathematical programming, because their built-in optimization instruments aren't flexible enough. As example, Minimize/Maximize/LinearProgramming[4] in Wolfram Alpha don't provide control over exact method that is going to be used for solving, and free version of Wolfram Alpha doesn't provide full solution. Symbolab.com[5] isn't able to do optimization with matrices at all. The main goal of this thesis, is to create a tool, that would be capable of solving variety of linear algebra problems, enable solving of multi-step problems with minimum effort for user, while staying simple enough to be easy for users to learn, and cheap and simple for developers to create and maintain.

2. Existing Solutions

Solver software can be divided into two groups: Professional-oriented and Education-oriented. This section describes of most advanced and distinct examples of software from each category.

2.1 Education-oriented software

Education-oriented software orients itself mostly towards audience of school/college students, who require answers for basic questions from given topics of science.

2.1.1 Symbolab

Symbolab was founded in 2011 in Israel, and was initially meant to be a "mathematical semantic search engine". Through only search bar, it provides users with ability to solve number of problems like: equations, simultaneous equations, inequalities, integrals, derivatives, limits, tangent line, trigonometric equations, functions. To some degree, it can understand text prompts[1]. Big accent in advertising of Symbolab is put on function of providing step-by-step solutions for problems (in free tier full step-by-step solutions may be unavailable[2]).

Apart from search, Symbolab also has a collection of specialized solvers and helpers, meant for graphing, practicing math, keeping notes, problem-specific calculators[3] and "Solutions" page which provides list of examples of prompts for search engine and allows to enter specialised symbols.

For matrices specifically, all calculations are defined by entering them in search bar. Also for matrices Symbolab supports finding diagonalization, eigenvalues, eigenvectors and gauss jordan[4]. Using answer of previous prompt in the next one is possible only by copy-pasting, which has its downsides since Symbolab can return matrix in a form that isn't valid for prompt. Symbolab allows for symbolic calculations within matrices.

Symbolab is a nice tool, but more focused on general mathematical problem solving for students, and not optimised for linear algebra. It doesn't support operations over selections of elements in matrices (required for manual evaluation of Gauss method[5], simplex algorithm[6] etc.), does not have built-in support for using answers from previous prompts,

and does not have ability to generate special matrices (like eye-matrix, ones-matrix, zeros-matrix).

2.1.2 WolframAlpha

WolframAlpha is a computational knowledge engine, launched in 2009.

Quote from WolframAlpha about page: 'Our mission is to collect and curate all objective data; implement every known model, method and algorithm; and make it possible to compute whatever can be computed about anything. Our work builds on the achievements of science and other systematizations of knowledge to provide a single source that can be relied on by everyone for definitive answers to factual queries.' [7]

It provides user with a prompt bar, where natural language can be used, specialized symbols as well as Wolfram Language[8]. Through the search bar, using one of the ways to define problem, numerous solvers from various areas of science can be accessed, starting from mathematical equation solving[9] and ending with completion of song lines[10]. For prompt, that returns matrix as a result (for example: $\{\{2, -1\}, \{1, 3\}\} \cdot \{\{1, 2\}, \{3, 4\}\}$), WolframAlpha would right away compute dimensions, trace, inverse, determinant, diagonalization, conditional number and draw a matrix plot. Thanks to support of Wolfram Language, among a lot of other things it supports symbolic calculations[11] and various optimisation methods, for example minimization with constraints[12]. This reliance on syntax of Wolfram Language gives great power to WolframAlpha, but also makes it impossible to use it without some degree of training and documentation referencing, which in turn places WolframAlpha closer to the group of professional-oriented software. Pro-membership allows to see step-by-step solutions and download/upload data in variety of formats[13]. Results of calculations can be copied as plain text for free.

If user has ability to write down intermediate results of calculations (for example, by using Windows Notepad), any problem involving matrices can be solved in WolframAlpha, although with a degree of discomfort.

Overall WolframAlpha is probably the most advanced education-oriented solver from all described in this paper, by functionality bordering with solvers oriented for professionals (and has some of negative sides that normally exist within professional-oriented software), but form factor of a prompt bar makes it uncomfortable for manual evaluation of some algorithms.

2.1.3 Matrixcalc

Matrixcalc[14] is only software in this section that purely specializes on matrices. It supports basic operations, as well as determinants, diagonalizations, row echelon form, LU and Cholesky decompositions, eigenvalues. It provides very rich step-by-step explanations for free, as well as built-in function of re-using results as arguments in consequent calculations, and supports symbols. Matrixcalc keeps history of previous calculations. It doesn't provide support to work with element selections from matrices.

Out of all solvers reviewed in this section, Matrixcalc is the most comfortable education-oriented software to work with when it comes to matrices, and probably, the closest to what this papers tries to achieve.

2.1.4 Mathway

Mathway is a tool that positions itself as best general mathematics as well as physics and chemistry solver for students[15]. Mathway has option for viewing step-by-step solutions. For matrices, it has a lot of built-in functions, but nesting functions doesn't work well in Mathway, because it tends to return incompletely computed result, and despite keeping history of computations, re-using results from previous operations is also uncomfortable because Mathway doesn't provide any way of doing so, and copy-pasting result returns matrix in invalid to input form[16].

2.1.5 Common traits

3 out of 4 reviewed solvers had similar for factor: aim for general mathematics, without particular focus on any area, feature smart prompt bar, capable of recognising variety of inputs, and have step-by-step solution description, which was often hidden behind paywall, and used as business-model. All 4 feature symbolic computations, in 2 out of 4 cases it is possible without any additional processing from user to re-use results from previous computations. None explicitly featured ability to operate over selections from matrices.

2.2 Professional-oriented software

Since the middle of 20-th century, when computers began to be prevalent, mathematical software was developed and released at consistent rates[17][18].

2.2.1 MATLAB

Numerical computing environment and a co-called programming language MATLAB first created in 1970's by mathematician Cleve Moler as a simple interactive calculator for matrices. Later, together with John L. Nittle and Steve Bangert they re-programmed MATLAB in C and created a specialised language for it[19]. Right now, MATLAB is a proprietary software used by 4 million people and 5000 universities, for science, economics and engineering[20]. Although MATLAB is a system with primary purpose in linear algebra, it also supports plotting, general programming, creation of user interfaces and supports interaction with other programming languages. Extensions that allow symbolic computing and model-based design can be added via toolboxes[21]. MATLAB influenced many other projects related to matrix computing, and syntax similar to that of MATLAB can be found in many other science-oriented programming languages.

2.2.2 Julia

Julia is an efficient general-purpose programming language, with accent on numerical analysis and statistics. Development has begun in 2009 by Jeff Bezanson, Stefan Karpinski, Viral B. Shah, and Alan Edelman[22]. Julia is used by many companies and organisations for variety of purposes, like engineering and computer science[23], medical development [24] and economical modelling [25]. One of the goals of Julia, was to make the language easy to use for scientists and mathematicians, so in many cases notation of Julia would be either reminiscent of mathematical notation, or familiar from older languages, like MATLAB. As a general purpose language, it is very flexible, and due to its similarity to already existing languages, it is simple to understand for newcomers with a programming or scientific computing background.

2.2.3 Python with NumPy

Development on general purpose language Python begun in 1989 by Guido van Rossum. Python language is oriented on ease of readability and writing, and is popular in scientific computing[26]. NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. NumPy is a result of an evolution of numerical computing packages for Python, that involved 4 stages of attempts to create a unified and efficient tool: matrix-sig, Numeric, Numarray and last, NumPy, with only the latter being actively maintained right now. Python and NumPy together allow to comfortably work with matrices, although restrictions posed by Python not being initially

designed for scientific computing make notation not very obvious for people with no previous Python or NumPy experience[27].

2.2.4 Common traits and critique

Workflow for processing matrices in all represented software is roughly the same: matrices are assigned to variables, and necessary operations are performed by passing variables to built-in function. To simplify the process, languages might feature syntactic sugar for quicker access to matrix cells, easier and more readable operations notation.

```

1      format rational
2      % initialize table M
3          % 1, -x1 -x2
4      M = [ 0, -1/2, 3; %1
5          -1/2, 1, -3/2; %-x1
6          3, -3/2, 3; %-x2
7          %-----
8          0, -1, 0; %a1
9          0, 0, -1; %a2
10         12, 4, 3; %a3
11         1, -1, 1] %a4
12     % append column 3 multiplied by -1 to columns 1 and 2
13     M = [M(:, 1:2), M(:, 3).*(-1)]
14     % add column 1 with column 3, subtract column 3 from column 2, leave
15     % column 3 as it is
16     M = [M(:, 1) + M(:,3), M(:,2)-M(:,3), M(:,3)]
17     % multiply row 3 by -1, leave all other rows as they are
18     M = [M(1:2,:); M(3,:) .* -1; M(4:end,:)]
19     % add row 3 to row 1, subtract row 3 from row 2, leave the rest of the rows
20     % as they are
21     M = [M(1,:) + M(3,:); M(2,:) - M(3,:); M(3,:); M(4:end,:)]
22

```

Figure 1. *First iteration of Beale's algorithm expressed in MATLAB code*

Main disadvantages for all systems listed above, and many other languages that were made for similar purpose comes from the nature of being a programming language. First disadvantage is that none of the programming languages can be used to its full extent from day 1, and ease of learning is even more important in case of specialised and scientific languages because people with no computer science background will need these languages as tools. Second disadvantage is due to code lacking necessary expressiveness for matrix manipulations, because form-factor of code lines being single-dimensional[28][29].

3. Analysis

This section describes the higher-level requirements for the system as well as the technological choices that are used for implementing the system.

3.1 Problems that the system is intended to solve

Below is the list of problems that the system should be able to solve after it is completed:

- Basic operations on matrices (addition, subtraction, multiplication, inversion, transposition)
- Gauss method
- Simplex method
- Two-Phase simplex method
- Beale method
- Leontjev system
- M-method

So it has to have following capabilities:

- Matrix row/columns/cell selection
- History of results of performed calculations
- Contain necessary basic functions
- Computing with symbols

3.2 Requirements

The system shall fulfill 2 following points:

- Make it possible to manually evaluate algorithms listed in previous section.
- Be simple and user-friendly, so user can use it without documentation and external assistance.

3.3 Technological choices

Considering requirements above, we can state that the application will need GUI. There are 3 forms of GUI applications: desktop-based, web-based and mobile-based. Desktop applications are applications that user would launch on their personal desktop computer, they don't necessarily require an internet connection to function, and can be big in size without having to download assets on every launch. Mobile-based application can have similar features, but can only be used on mobile. Web-based application don't need to be installed, but they should be sufficiently lightweight to work in a browser window. One of the advantages of web-based application is that it can be used on desktop as well as adapted to mobile devices, with only restriction being the presence of internet connection. Another advantage is that web application can be containerized using technologies like Electron and become standalone desktop or mobile apps.

Front-end parts of web applications can be created in multiple ways, but in current time they are usually written using one of the front-end development frameworks, and each framework comes with a set of advantages and drawbacks. For purposes of this thesis, main requirement for front-end frameworks is big community size, because it guarantees developed ecosystem, proper support and sufficient documentation. Based on this primary constraint 3 frameworks were chosen to be compared against each other: React, Angular and Vue.js .

3.3.1 React

React is a free and open-source front-end JavaScript library for building user interfaces based on components. It is maintained by Meta (formerly Facebook) and a community of individual developers and companies[30].

React is a choice of many big companies, has a lot of supporting tools and requires knowledge of medium to advanced JS concepts[31][32][33].

3.3.2 Angular

Angular is a TypeScript-based, free and open-source single-page web application framework led by the Angular Team at Google and by a community of individuals and corporations. Angular is a complete rewrite from the same team that built AngularJS[34].

Angular is a completely self contained framework, and does not require any extra tools to

function, but it also has the shrapest learning curve[31][32][33].

3.3.3 Vue.js

Vue.js is an open source JavaScript/TypeScript framework for reactive single-page application[35].

It has the smallest following and is not backed by any big tech companies, but is considered an easiest one to learn and best suited for small applications[31][32][33].

3.3.4 Final choice

Because of limited time and no previous experience in JS/TS, Vue framework was chosen because of claimed ease of learning. Even though Vue has weakest support and smallest community, its sufficient for purposes of this thesis.

3.3.5 Choice of back-end frameworks

Because complete web-application usually consists of two parts - client side (front-end) and back-end - the tech-stack for then latter also needs to be addressed. There are three candidates again: Java+Spring, Scala+Play and Python+Django. The final choice fell on Python+Django, for reasons similar to choice of Vue.js, it requires the least work to create an application and disadvantages of using lighter framework with dynamically typed language are irrelevant because of relatively small expected scope of the project and a single-member developer team.

As work on the project continued, it became evident that fully-functional application might be served as a static page, and does not require dedicated back-end, so no work on it beyond what is described in this subsection was done.

4. Results

Application is complete, it contains all features promised in section 3.1 and mostly caters to requirements in section 3.2. Most design decision and minor features that were considered important were also implemented. Despite all that, there is a wide front of possible improvements and advancements, some of which were planned to be done in the scope of this thesis, but for some reasons weren't. More information on that can be found in section 6.

Source code of the final application can be found at <https://gitlab.cs.ttu.ee/royako/vicalial-vue>. Screenshots of the application can be found in 7.

System is a web-application served like a static page. At the moment it does not have any features that require back-end side, so user cannot create an account and save work on a cloud. Once loaded, user is presented with default welcome page, same for all users, and once application is closed, all information that was not saved locally is completely lost.

Workflow of the system is centered around matrices. In context of the application matrix is defined as a 2 dimensional array of numerical values. User can upload or write matrices in several ways, and once matrices are created, they can be seen on the user interface. UI allows to perform many operations over matrices using pre-defined functions. Already created matrices are not lost if they were used in an operation, and can be used again, unless user deliberately deletes them. User can save individual matrices locally in several file formats.

Subsections below provide description of all functionality.

4.0.1 Creating matrices

To create a matrix user is presented with 3 options, all of which can be accessed through the same window.

- Manual Input - user can type width and height of a matrix. In presented template user can enter values one-by-one into each cell, all empty cells will be automatically filled with zeros. Input form can evaluate simple mathematical and symbolic expressions. Symbols are identified automatically.

- From File - Matrices can be uploaded in two formats: .XLSX and .CSV. Mathematical and symbolic expressions can be evaluated.
- Generator - user can type width and height of a matrix. There are 4 options that would automatically fill the template with values: Ones, Zeros, Eye and Range. Ones fills every template cell with value 1, Zeros fill every template cell with value 0. Eye fills all cells with zeros, except for the values on diagonal from upper-left corner to bottom-right corner, which are filled with 1. Range creates a matrix where each row is a sequence of numbers starting from 0 at the left side, and ending with value equal to matrix width - 1.

4.0.2 Manipulating matrices

Once a matrix is created and visible, following manipulations can be done:

- Deletion - a matrix can be permanently deleted.
- Renaming - each matrix is given a default name, but it can be changed by clicking on matrix name and editing it. Editing matrix name also automatically changes all occurrences of the name.
- History - closest relatives of the matrix can be viewed next to the name.
- Formatting - two options are available for formatting: "Long Float" and "Fraction". "Long Float" will display all non-integer values with 8 signs after comma, "Fraction" will display non-integers in form of a division of two whole numbers.
- Selection - by pressing a black square button in the upper right corner of the matrix, whole matrix can be selected, rows and columns can be selected by pressing a button next to them, single cells can be selected by clicking on them. Selecting a matrix or its part is a necessary step before applying operations to it.
- Deselection - user can either deselect everything by clicking a button on yellow taskbar, or only remove selection related to a particular matrix, by pressing deselect button under it.

Once a matrix or some part of it is selected, function selection menu is available. Through function selection menu, user can choose a function to manipulate numerical data in matrices, plot them, export, or change visual appearance of matrices. Functions are divided into three categories:

- Math
 - Add - adds two matrices/selections of compatible size.
 - Subtract - subtracts two matrices/selections of compatible size.
 - Multiply - multiplies two matrices/selections of compatible size.

- Transpose - transposes matrix or selection.
- Inverse - inverts square matrix or selection.
- Multiply by Constant - allows to multiply any selection by coefficient.
- Element-Wise Product - multiplies element-wise two matrices/selections of compatible size.
- Linear Programming: Minimize - takes matrix as if it is a simplex table, and finds minimum of a objective function. Objective function - is a first row of the matrix, constraint coefficients - first column of matrix. Solver treats constraints as if they are equalities.
- Code
 - Replace - replaces selection from first argument with selection from second argument. Returns whole parent of first selection with replacement.
 - Swap - swaps places of selections from first and second argument.
 - Size - returns 1-by-2 matrix where left element equals to amount of rows in selection/matrix, and right element is equal to amount of columns.
 - Select - takes any amount of selections as arguments, as long as they all are from the same matrix, and returns new matrix made out of these selections.
 - Append Rows - appends sideways two matrices/selections of same row amount.
 - Append Columns - appends vertically two matrices/selections of same columns amount.
- Other
 - Plot - plots matrix/selection.
 - Export - allows to export matrix or selection in downloadable formats.
 - Background Color - sets background color of selected cells to desired color.

4.0.3 Exporting matrices

Matrices can be exported through Export function. By using export, user can download a single matrix in three formats: .XLSX, .CSV and .LATEX .

5. Testing

User testing of application happened in two rounds: first in the middle and second at the end of development. Testing happened using Corridor testing technique, as well as interview of users (live and by e-mail). This chapter describes results of testing.

5.1 First testing round

First round was performed in a bit of chaotic fashion, was not being recorded and protocol from Appendix 27 has not yet been developed. Main takeout from this round was that its not very understandable what user is expected to do when they first land on web-page, as well as multiple point of confusion throughout the user flow. Also, users were confused when operations that couldn't yield valid result did nothing.

After first round of testing major refactor were performed on the interface: were added multiple tooltips, floating snack-bars with explanations, and a button on a landing page suggesting the first operation. Also, logic was added for snack-bars to appear if performed operation yielded error.

5.2 Second testing round

Second round was done while tightly following Corridor Testing Protocol 7. Tests were done by three people: two first-year college students and one 3-rd year.

5.2.1 Negatives

Below is the list of troubles that users had while using the application.

1. All 3 users took long to figure out that matrices should be entered through 'ANY EXPRESSION' tab in Add Data Window. All of them at first attempted to edit pre-defined matrices in 'GENERATOR' tab.
2. One user generated a ones-matrix, and then tried to edit values in cells of it.
3. One user tried to select whole matrix by selecting individually each cell of matrix.
4. User tried to insert more that one matrix at a time into text field of 'ANY EXPRESSION' tab.
5. Some users had issue with understanding English words. Particular example, one

- user got stuck on E - A because he didn't know that the action is called 'subtraction'.
6. User didn't understand how selection works, and tried to insert names of matrices into coefficients text-fields.
 7. Users had trouble figuring out how to remove selection.
 8. Users tried to deselect by clicking outside statements and by clicking on already selected areas.

5.2.2 Suggestions

Below is the list of suggestions for features to add that testers think would make usage of the system easier.

1. tutorial.
2. localisation.
3. feedback option.
4. make it possible to choose function first, and then arguments.

5.2.3 Fixes

As a result of a second testing round, most problematic part of the interface, the Add Data window, was redesigned. Now, instead of having to write matrix in a .CSV format in one text field, users can select size of the matrix and then fill the empty matrix template with values.

6. Future Plans

Following there is a list of features and changes, that would benefit the project according to the personal opinion of the author.

- Right now statements can be of only 2 types: matrix and plot. It would be nice to rewire statement system to accommodate new types easier.
- Create back-end.
 - Add account system.
 - Enable ability to save work on server.
 - Right now pyodide and some other dependencies are fetched through jsdelivr, it would be faster and more optimal to serve own copy of pyodide.
- Add more different plot rendering functions.
- Add an option to change argument of an already executed function, and an option to propagate changes to all descending operations.
- Add functionality to save all calculation at once in a form of a LaTeX document or Python code.
- Add operation accumulator, to do several operations at once without having to create unneeded intermediate matrices.
- Create text statements, like Markdown containers in Jupyter Notebook.
- Because coding using own visual calculator statements might not be a feature to be introduced soon, project will definitely benefit from ability to define custom functions by writing them in python.
- Add if and for statements, and other tools to enable programming within calculator.
- Change component matrix viewer component, so that in case of a matrix with big number of cells, it still looks compact. It can be done by adding scroller and hiding some cells.

7. Summary

The goal of this thesis was to create an application that would fill the niche between two kinds of scientific software for matrix manipulation: simple to use but limited and flexible and powerful but very complicated. The created application fits the desired requirements by being visual and interactive, storing operations history, and not requiring any extra documentation to be used.

References

- [1] *Symbolab About Page*. [Accessed: May 2023]. URL: <https://www.symbolab.com/about>.
- [2] *Symbolab Upgrade Page*. [Accessed: May 2023]. URL: <https://www.symbolab.com/registration#flow=upgrade>.
- [3] *Symbolab Calculators*. [Accessed: May 2023]. URL: <https://www.symbolab.com/calculator/>.
- [4] *Symbolab Linear Algebra Calculator*. [Accessed: May 2023]. URL: <https://www.symbolab.com/solver/linear-algebra-calculator>.
- [5] *Wikipedia, Gaussian elimination*. [Accessed: May 2023]. URL: https://en.wikipedia.org/wiki/Gaussian_elimination.
- [6] *Wikipedia, Simplex algorithm*. [Accessed: May 2023]. URL: https://en.wikipedia.org/wiki/Simplex_algorithm.
- [7] *About Wolfram|Alpha*. [Accessed: May 2023]. URL: <https://www.wolframalpha.com/about>.
- [8] *Wolfram Language*. [Accessed: May 2023]. URL: <https://www.wolfram.com/language/>.
- [9] *WolframAlpha Equation Solving*. [Accessed: May 2023]. URL: <https://www.wolframalpha.com/examples/mathematics/algebra/equation-solving>.
- [10] *WolframAlpha Everyday Life Surprises*. [Accessed: May 2023]. URL: <https://www.wolframalpha.com/examples/everyday-life/surprises>.
- [11] *Wolfram Language Symbolic Calculations Tutorial*. [Accessed: May 2023]. URL: <https://reference.wolfram.com/language/tutorial/SymbolicCalculations.html>.
- [12] *Wolfram Language Minimize*. [Accessed: May 2023]. URL: <https://reference.wolfram.com/language/ref/Minimize.html>.
- [13] *Wolfram|Alpha Pro*. [Accessed: May 2023]. URL: <https://www.wolframalpha.com/pro/>.
- [14] *Matrix calculator*. [Accessed: May 2023]. URL: <https://matrixcalc.org>.
- [15] *Mathway About Page*. [Accessed: May 2023]. URL: <https://www.mathway.com/about>.

- [16] *Mathway Linear Algebra*. [Accessed: May 2023]. URL: <https://www.mathway.com/LinearAlgebra>.
- [17] *List of COIN-OR projects*. [Accessed: May 2023]. URL: <https://www.coin-or.org/projects/#ffs-tabbed-11>.
- [18] Julian Hall. *HiGHS High-performance open-source software for linear optimization*. [Accessed: May 2023]. URL: https://co-at-work.zib.de/slides/Montag_21.9/HiGHS%20slides%20%281%29.pdf.
- [19] N. Chonacky and D. Winch. “Reviews of Maple, Mathematica, and Matlab: Coming Soon to a Publication Near You”. In: 2005.
- [20] *Current number of matlab users worldwide*. [Accessed: December 2023]. URL: <https://www.mathworks.com/matlabcentral/answers/77436-current-number-of-matlab-users-worldwide>.
- [21] *Get Started with MATLAB*. [Accessed: December 2023]. URL: https://www.mathworks.com/help/matlab/getting-started-with-matlab.html?s_tid=CRUX_lftnav.
- [22] *Why we created Julia*. [Accessed: December 2023]. URL: <https://webcitation.org/6Gqdx1FXF?url=http://julialang.org/blog/2012/02/why-we-created-julia/>.
- [23] *Introducing Braket.jl - Quantum Computing with Julia*. [Accessed: December 2023]. URL: <https://forem.julialang.org/kshyatt/introducing-braketjl-10f2>.
- [24] *Julia: Case Study*. [Accessed: December 2023]. URL: <https://juliahub.com/case-studies/>.
- [25] *How A New Programming Language Created By Four Scientists Now Used By The World's Biggest Companies*. [Accessed: December 2023]. URL: <https://www.forbes.com/sites/suparnadutt/2017/09/20/this-startup-created-a-new-programming-language-now-used-by-the-worlds-biggest-companies/>.
- [26] *Python® – the language of today and tomorrow*. [Accessed: December 2023]. URL: <https://pythoninstitute.org/about-python#:~:text=Python%20was%20created%20by%20Guido,called%20Monty%20Python's%20Flying%20Circus..>
- [27] Gaël Varoquaux Stefan Van Der Walt S. Chris Colbert. “The NumPy array: a structure for efficient numerical computation”. In: 2011.

- [28] *Convincing colleagues to use Python instead of Matlab/C for a basic computer science class.* [Accessed: December 2023]. URL: https://www.reddit.com/r/Python/comments/2khhnq/convincing_colleagues_to_use_python_instead_of/.
- [29] *Does MATLAB get a bad rap?* [Accessed: December 2023]. URL: https://www.reddit.com/r/matlab/comments/oelui/does_matlab_get_a_bad_rap/.
- [30] *The History of React.js on a Timeline.* [Accessed: December 2023]. URL: <https://blog.risingstack.com/the-history-of-react-js-on-a-timeline/>.
- [31] Aris Pattakos. *Angular vs React vs Vue 2023.* [Accessed: May 2023]. URL: <https://athemes.com/guides/angular-vs-react-vs-vue/>.
- [32] Shaumik Daityari. *Angular vs React vs Vue: Which Framework to Choose.* [Accessed: May 2023]. URL: <https://www.codeinwp.com/blog/angular-vs-vue-vs-react/>.
- [33] Mohit Joshi. *Angular vs React vs Vue: Core Differences.* [Accessed: May 2023]. URL: <https://www.browserstack.com/guide/angular-vs-react-vs-vue>.
- [34] *AngularJS History: Timeline, Previous Version, Latest News.* [Accessed: December 2023]. URL: <https://www.knowledgehut.com/blog/web-development/angular-version-history#features-of-angular%C2%A0>.
- [35] *What is Vue.js?* [Accessed: December 2023]. URL: <https://www.startechup.com/blog/what-is-vue-js/>.

Appendix 1 – Non-Exclusive License for Reproduction and Publication of a Graduation Thesis¹

I Roman Yakovlev

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis “Visual Calculator for Linear Algebra, Linear and Non-linear Programming”, supervised by Mati Väljas
 - 1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
 - 1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons’ intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

11.01.2024

¹The non-exclusive licence is not valid during the validity of access restriction indicated in the student’s application for restriction on access to the graduation thesis that has been signed by the school’s dean, except in case of the university’s right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.

Appendix 2 - Corridor Testing Protocol

To ensure that every user is being tested the same way and no point is being missed, following steps were performed during each corridor testing session in second testing round:

1. Record screen where system is being tested and voices of interviewer and interviewee.
2. Explain idea of corridor testing and what user has to do, so to avoid confusion and unnecessary actions from user.
3. Explain what the system is about, and what it is meant for.
4. Tell user a following backstory: "You are a student in college, and you are taking linear algebra class. Recently, you have seen an ad of this application, and came to this web-page expecting it to assist you with your assignments."
5. Give access to the system, by providing user with a link to the system's web-page.
6. Give user proposal to think how he might use the application, give opportunity to explore it in their own.
7. Give following problem to a user and ask to solve it:

$$A = \begin{bmatrix} 0.1 & 0.05 & 0.2 \\ 0.3 & 0 & 0.15 \\ 0.2 & 0.4 & 0 \end{bmatrix}$$

$$Y = \begin{bmatrix} 100 \\ 50 \\ 80 \end{bmatrix}$$

$$E = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$X = ((E - A)^{-1}) * Y$$

Objective is to find value of X.

8. Ask questions:
 - (a) "What would you add, to make your life easier?"
 - (b) "Maybe you know some software, that implements same functionality in a better way?"

9. Propose user to solve same problem using Symbolab, Mathway and other software, and compare.

Appendix 3 - Calculator Interface

Below you can find screenshots of the created application being used to solve an operational analysis problem. Screenshots of some calculation steps were not taken, because they do not show any new functionality of the application. Number after letter M in the names of matrices is not indicative of the number of steps performed. Last screenshot, containing the final result is displayed in "Long Float" format to show corresponding functionality.

Calculator

The calculator interface displays a matrix with the following values:

6	-1/2	3
-1/2	1	-3/2
3	-3/2	3
0	-1	0
0	0	-1
12	4	3
1	-1	1

The matrix is labeled $M1$. The interface includes tabs for MATH, CODE, and OTHER. The BACKGROUND COLOR tab is active, showing a color selection tool with a pink circle and the text "SELECTED COLOR: PINK".

Add data Deselect Format as: Long Float Fraction

Calculator

The calculator interface displays the same matrix as above:

6	-1/2	3
-1/2	1	-3/2
3	-3/2	3
0	-1	0
0	0	-1
12	4	3
1	-1	1

The matrix is labeled $M1$. The interface includes tabs for MATH, CODE, and OTHER. The BACKGROUND COLOR tab is active, showing a color selection tool with a red circle and the text "SELECTED COLOR: RED".

Add data Deselect Format as: Long Float Fraction

Calculator

$M1 =$

0	-1/2	3
-1/2	1	-3/2
3	-3/2	3
0	-1	0
0	0	-1
12	4	3
1	-1	-1

DESELECT

MATH CODE OTHER
choose function
ADD SUBTRACT MULTIPLY TRANSPOSE INVERSE
More functions
Multiply by Constant

Multiply by Constant DESCRIPTION

you can write coefficients in white boxes
(-1 M1[2])

In parent

APPLY

Add data Deselect Format as: Long Float Fraction

Calculator

$M2 = (-1 * M1[2])$

0	-1/2	-3
-1/2	1	3/2
3	-3/2	-3
0	-1	0
0	0	1
12	4	-3
1	-1	-1

DESELECT

MATH CODE OTHER
choose function
ADD SUBTRACT MULTIPLY TRANSPOSE INVERSE
More functions

Add DESCRIPTION

you can write coefficients in white boxes
(M2[0]) + (M2[2])

In parent

APPLY

Add data Deselect Format as: Long Float Fraction

$M4 = M2[0] + M2[2]$

-3	-1/2	-3
1	1	3/2
0	-3/2	-3
0	-1	0
1	0	1
9	4	-3
0	-1	-1

DESELECT

MATH CODE OTHER
choose function
ADD SUBTRACT MULTIPLY TRANSPOSE INVERSE
More functions

Subtract DESCRIPTION

you can write coefficients in white boxes
(M4[1]) - (M4[2])

In parent

APPLY

Add data Deselect Format as: Long Float Fraction

$M4 = M2 \cdot 0 + M2 \cdot 2$

0	-1	0
0	0	1
12	4	-3
1	-1	-1

-3	-1/2	-3
1	1	3/2
0	-3/2	-3
0	-1	0
1	0	1
9	4	-3
0	-1	-1

MATH CODE OTHER
choose function
ADD SUBTRACT MULTIPLY TRANSPOSE INVERSE
More functions
Multiply by Constant

Multiply by Constant DESCRIPTION
you can write coefficients in white boxes
(M52,)
 in parent
APPLY

Add data Deselect Format as: Long Float Fraction

$M5 = M4 \cdot 1 - M4 \cdot 2$

0	-1	0
1	0	1
9	4	-3
0	-1	-1

-3	5/2	-3
1	-1/2	3/2
0	3/2	-3
0	-1	0
1	-1	1
9	7	-3
0	0	-1

MATH CODE OTHER
choose function
ADD SUBTRACT MULTIPLY TRANSPOSE INVERSE
More functions
Multiply by Constant

Add DESCRIPTION
you can write coefficients in white boxes
(M50,) + (M52,)
 in parent
APPLY

Add data Deselect Format as: Long Float Fraction

$M6 = (-1 \cdot M52 \cdot 1)$

0	-1	0
1	-1	1
9	7	-3
0	0	-1

-3	5/2	-3
1	-1/2	3/2
0	-3/2	3
0	-1	0
1	-1	1
9	7	-3
0	0	-1

MATH CODE OTHER
choose function
ADD SUBTRACT MULTIPLY TRANSPOSE INVERSE
More functions
Multiply by Constant

Subtract DESCRIPTION
you can write coefficients in white boxes
(M71,) - (M72,)
 in parent
APPLY

Add data Deselect Format as: Long Float Fraction

$$\begin{bmatrix} 1 & -1 & 1 \\ 9 & 7 & -3 \\ 0 & 0 & -1 \end{bmatrix}$$

$$\begin{bmatrix} -3 & 1 & 0 \\ 1 & -1/2 & 3/2 \\ 0 & -3/2 & 3 \\ 0 & -1 & 0 \\ 1 & -1 & 1 \\ 9 & 7 & -3 \\ 0 & 0 & -1 \end{bmatrix}$$

$$M7 = M9[0, 1] + M8[2, 1]$$

$$\begin{bmatrix} -3 & 1 & 0 \\ 1 & -3/2 & 3 \\ 0 & -3/2 & 3 \\ 0 & -1 & 0 \\ 1 & -1 & 1 \\ 9 & 7 & -3 \\ 0 & 0 & -1 \end{bmatrix}$$

$$M10 = M7[1, 1] - M7[2, 1]$$

MATH CODE OTHER
 choose function
 PLOT EXPORT BACKGROUND COLOR
 More functions

Background Color
 background color (Selected: 1)
 SELECTED COLOR: RED
 APPLY

Add data Deselect Format as: Long Float Fraction

$$\begin{bmatrix} 0 & -1 & 0 \\ 1 & -1 & 1 \\ 9 & 7 & -3 \\ 0 & 0 & -1 \end{bmatrix}$$

$$\begin{bmatrix} -3 & 1 & 0 \\ 1 & -1/2 & 3/2 \\ 0 & -3/2 & 3 \\ 0 & -1 & 0 \\ 1 & -1 & 1 \\ 9 & 7 & -3 \\ 0 & 0 & -1 \end{bmatrix}$$

$$M7 = M9[0, 1] + M8[2, 1]$$

$$\begin{bmatrix} -3 & 1 & 0 \\ 1 & -3/2 & 3 \\ 0 & -3/2 & 3 \\ 0 & -1 & 0 \\ 1 & -1 & 1 \\ 9 & 7 & -3 \\ 0 & 0 & -1 \end{bmatrix}$$

$$M10 = M7[1, 1] - M7[2, 1]$$

MATH CODE OTHER
 choose function
 ADD SUBTRACT MULTIPLY TRANSPOSE INVERSE
 More functions

Subtract
 DESCRIPTION
 you can write coefficients in white boxes
 () - ()
 in parent
 APPLY

Add data Deselect Format as: Long Float Fraction

$$\begin{bmatrix} 0 & -1 & 0 \\ 1 & -1 & 1 \\ 9 & 7 & -3 \\ 0 & 0 & -1 \end{bmatrix}$$

$$\begin{bmatrix} -4 & 1 & 0 \\ 0 & 1 & -3/2 \\ 3/2 & -3/2 & 3 \\ 1 & -1 & 0 \\ 2 & -1 & 1 \\ 2 & 7 & -3 \\ 0 & 0 & -1 \end{bmatrix}$$

$$M11 = M10[1, 0] - M10[1, 1]$$

$$\begin{bmatrix} -4 & 1 & 3/2 \\ 0 & 1 & 0 \\ 3/2 & -3/2 & 3/4 \\ 1 & -1 & -3/2 \\ 2 & -1 & -1/2 \\ 2 & 7 & 15/2 \\ 0 & 0 & -1 \end{bmatrix}$$

$$M12 = M11[1, 2] + (3/2 * M11[1, 1])$$

MATH CODE OTHER
 choose function
 ADD SUBTRACT MULTIPLY TRANSPOSE INVERSE
 More functions

Subtract
 DESCRIPTION
 you can write coefficients in white boxes
 () - ()
 in parent
 APPLY

Add data Deselect Format as: Long Float Fraction

1	-1	0
2	-1	1
2	7	-3
0	0	-1

-4	1	3/2
0	1	0
3/2	-3/2	3/4
1	-1	-3/2
2	-1	-1/2
2	7	15/2
0	0	-1

M12 = M11[., 2] + (3/2 * M11[., 1])

-4	0	3/2
0	1	0
3/2	-3/2	3/4
1	-1	-3/2
2	-1	-1/2
2	7	15/2
0	0	-1

M13 = M12[0, 1] - M12[1, 1]

MATH CODE OTHER

choose function

ADD SUBTRACT MULTIPLY TRANSPOSE INVERSE

More functions

Add

DESCRIPTION

you can write coefficients in white boxes

(M13[2, 1]) + (3/2 M13[1, 1])

In parent

APPLY

Add data Deselect Format as: Long Float Fraction

1	-1	-3/2
2	-1	-1/2
0	0	-1

-4	0	3/2
0	1	0
3/2	-3/2	3/4
1	-1	-3/2
2	-1	-1/2
2	7	15/2
0	0	-1

M13 = M12[0, 1] - M12[1, 1]

-4	0	3/2
0	1	0
3/2	0	
1	-1	-3/2
2	-1	-1/2
2	7	15/2
0	0	-1

M14 = M13[2, 1] + (3/2 * M13[1, 1])

MATH CODE OTHER

choose function

ADD SUBTRACT MULTIPLY TRANSPOSE INVERSE

More functions

Multiply by Constant

Multiply by Constant

DESCRIPTION

you can write coefficients in white boxes

(3/2 M14[., 2])

In parent

APPLY

Add data Deselect Format as: Long Float Fraction

1.4	0.4000000000000013	0.2
2.133333333333333	-0.533333333333333	0.0666666666666667
0	0	-1
0.2666666666666666	0.933333333333333	0.133333333333333

-4.746666666666667	-1.213333333333334	-0.173333333333334
0	1	0
-0.173333333333334	0.093333333333334	0.013333333333334
1.4	0.4000000000000013	0.2
2.133333333333333	-0.533333333333333	0.0666666666666667
0	0	-1
0.2666666666666666	0.933333333333333	0.133333333333333

M20 = M19[0, 1] + (2 * M19[2, 1])

-4.746666666666667	-1.213333333333334	-0.173333333333334
-1.213333333333334	0.093333333333334	0.093333333333334
-0.173333333333334	0.093333333333334	0.013333333333334
1.4	0.4000000000000013	0.2
2.133333333333333	-0.533333333333333	0.0666666666666667
0	0	-1
0.2666666666666666	0.933333333333333	0.133333333333333

M21 = M20[1, 1] + (7 * M20[2, 1])

Add data Deselect Format as: Long Float Fraction

	1.6935483878967742	0.24193548387896783	0.1774193548387897	<input type="checkbox"/>
	1.7419354838789677	-0.3225886451612983	0.8967741935483871	<input type="checkbox"/>
	0	0	-1	<input type="checkbox"/>
	0.9516129832258865	0.5645161298322581	0.88864516129832256	<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> M20	$= M20[0,] + (0/175 * M20[1,])$			
	-5.637896774193549	0	-0.10483878967741936	<input type="checkbox"/>
	0	0.6848387896774194	0	<input type="checkbox"/>
	-0.10483878967741934	0.85645161298322581	0.88864516129832258	<input type="checkbox"/>
	1.6935483878967742	0.24193548387896783	0.1774193548387897	<input type="checkbox"/>
	1.7419354838789677	-0.3225886451612983	0.8967741935483871	<input type="checkbox"/>
	0	0	-1	<input type="checkbox"/>
	0.9516129832258865	0.5645161298322581	0.88864516129832256	<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/> M27	$= M20[2,] - (7/75 * M20[1,])$			
	-5.637896774193549	0	-0.10483878967741936	<input type="checkbox"/>
	0	0.6848387896774194	0	<input type="checkbox"/>
	-0.10483878967741934	0	0.88864516129832258	<input type="checkbox"/>
	1.6935483878967742	0.24193548387896783	0.1774193548387897	<input type="checkbox"/>
	1.7419354838789677	-0.3225886451612983	0.8967741935483871	<input type="checkbox"/>
	0	0	-1	<input type="checkbox"/>
	0.9516129832258865	0.5645161298322581	0.88864516129832256	<input type="checkbox"/>

Add data Deselect Format as: Long Float Fraction