



Apr. 19-22, 2017

# Software Development

Champion, PA



Physics Math Helper

# Table of Contents

---

| Item                        | Page                           |      |
|-----------------------------|--------------------------------|------|
| Research                    | 2                              |      |
| Description of Project      | 3                              |      |
| Plan of Work Log            | 4                              |      |
| Development Process         | Project Requirements           | 5    |
|                             | High-Level Software Design     | 6    |
|                             | Testing                        | 7-8  |
|                             | End-User Product Documentation | 9-12 |
| Evaluation and Future       | 13                             |      |
| References                  | 14                             |      |
| Student Copyright Checklist | 15                             |      |
| DVD                         | 16                             |      |

# Research

---

Research about the problem was not truly necessary. This issue was encountered by multiple member of our group while performing a variety of physics problems. Therefore, the desire for a program that minimizes the amount of time put into solving these equations was immediately evident. After much deliberation between the group members, we agreed to make this project our goal for this year.

After we had selected our goal, we had to pursue it. As such, we found many assorted articles that gave us crucial information in both coding and high-level physics. These resources (cited within this documentation) were key in solving the problem at hand. By combining both our knowledge of AP Physics and 6 years of programming experience with the information gathered in the cited pieces, we were able to implement full functionality.

We discovered that vectors could be easily summed by converting them to C++ vectors and using relevant equations. The line of best fit could be effortlessly found via the linear least squares regression line

$$y = \left( \frac{N * \sum xy - \sum x \sum y}{N * \sum x^2 - (\sum x)^2} \right) x + \frac{\sum x^2 \sum y - \sum x \sum xy}{N * \sum x^2 - (\sum x)^2}$$

Finally, solutions of systems of equations could be realized via basic linear operators.

# Description of Project

---

Problem: Problems and Labs in advanced physics often involve performing tedious calculations.

Solution: Develop a program to streamline solutions.

Educational and Social Value: This project allows education (within the realm of physics) rather simplistically. Having a calculator that a user may easily enter data into allows a student to check their work with ease. In addition, teachers could reasonably use this program to show the solutions to problems and then explain the process it uses. This project also grants society a few key boons, notably for professionals. Now those who would previously have to work out problems by hand can simply enter the data and get the result. As such, physicists will find this program to be a wonderful tool.



| <b>TECHNOLOGY STUDENT ASSOCIATION<br/>PLAN OF WORK</b> |   |                      |                                |  |
|--|---|----------------------|--------------------------------|--|
| <b>Date</b>  | <b>Task</b>                                       | <b>Time involved</b> | <b>Team member responsible</b> | <b>Comments</b>  |
| 12/9/16  | Brainstorming ideas, Fill in research and problem | 3h                   | 1,2,3                          | Forces, Images, Communication, Image Sorting                     |
| 12/11/16   | Main GUI  | 1h                   | 1                              | Program appears on screen  |
| 12/16/16   | Tabs, Classes, Vector Sums                        | 9h                   | 1                              | Control library integrated with tab events, vector/force summing |
| 12/16/16   | Data Tab, Fitting Curves                          | 3h                   | 1                              | Calculations, layout   |
| 1/1/17   | System Solutions, Linearization                   | 3h                   | 1                              | Calculations, layout   |
| 1/8/17   | Reformatting, Help Text                           | 3h                   | 1                              | Consistent layout, wrote info                                    |
| 1/19/17-1/26/17  | Documentation                                     | 6h                   | 1,2,3                          | Began describing process and function, icons, naming             |
| 3/2/17   | Documentation Redo                                | 2h                   | 1,2,3                          | Finishing required sections, refining                            |
| 3/9/17   | Bug Fixing  | 2h                   | 1,2,3                          | Display errors   |
| 3/16/17  | Discussions                                       | 2h                   | 1,2,3                          | LEAP, use cases, examples  |
| 4/16/17  | Finalization                                      | 1h                   | 2,3                            | Printing, burning  |
| Advisor signature                                      |   |                      |                                |  |

# Project Requirements

---

The program should address common problems not accessible with a scientific calculator or difficult to scale or access on a graphing calculator.

These will include adding an arbitrary number of vectors, fitting arbitrary-ordered polynomials to an arbitrary number of data points, and solving an arbitrary number of simultaneous equations for an arbitrary number of variables.

Each problem should be solved independently while using common algorithms for simplicity.

Each input format should avoid unnecessary formatting to increase productivity and output should be formatted the same.

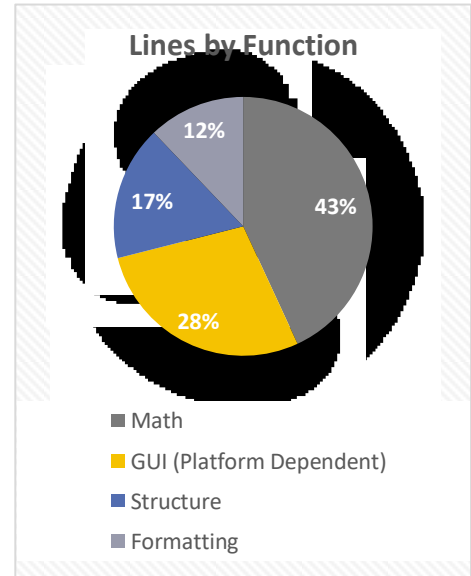
Each function/tab should use data which is separate, is stored independently, but can be copied (or pasted).

Each function/tab should process the GUI independently from calculation steps for troubleshooting.

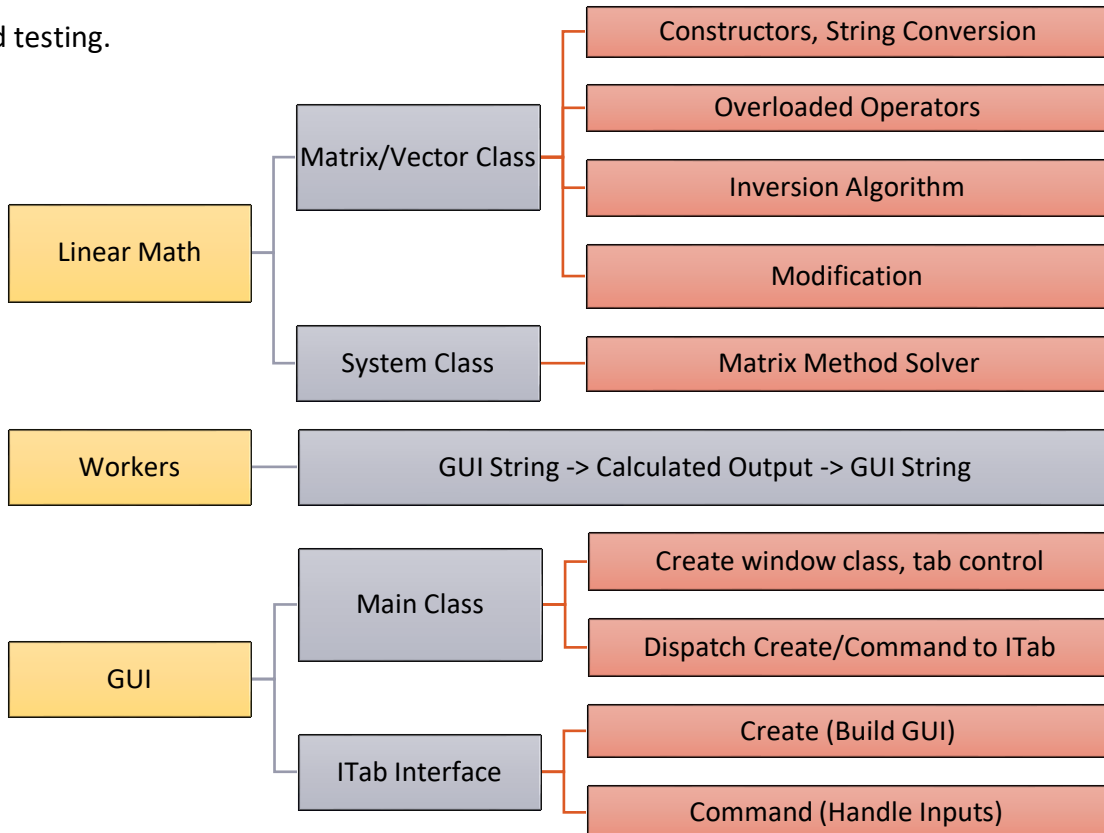
Each function/tab should be called in a streamlined manner via an interface.

# High-Level Software Design

The built-in Microsoft Windows API is used for the GUI and is wrapped by the windows.h header. The function WinMain (an overridden main) is called to create the window and WindowProcedure dispatches messages to components. Each component/tab item overrides a virtual ITab interface with a create event to draw buttons and text, a show and hide event, and a command event called for all operations on the screen. When a button's ID appears as pressed, input strings are fed to workers which return an output string to display.



Workers and the Linear Math class are platform-independent and use C++ vectors and common linear algebra algorithms to perform calculations with functionality for almost all possible inputs. Classes share common definitions, but functionality is isolated for compilation and testing.



# Testing

---

## Solution accuracy:

Selected a few sets of inputs (7, 2.63, -5, -37.2, 15, 2) →  $m=2.0673$ ,  $b=-22.5714$ , (7, 2.63, -5, -37.2, 15, 2, 37, 1.234, 4.5555, -7) →  $m=0.695049$ ,  $b=-15.807$  to run through the program's data section. Confirmed accuracy by computing the same set of inputs within Wolfram Alpha ✓

If more than one pair of (x,y) coordinates are on a single line (within the data section), formatting error is displayed ✓

If an x or y value isn't included on a single line (within the data section), formatting error is displayed ✓

If systems do not have enough, have too many, or have a mixed number of equations or unknowns, the user is informed and nothing is calculated ✓

If no system solution exists, the program notifies the user, and does not attempt to divide by zero ✓

If nothing is entered, nothing is calculated ✓

## Module unit testing:

Impossible to crash program through arbitrary input like "1,2,test,,,-4|{};/?" due to validation and non-strict conversion ✓

Tested math on trivial problems along the way to locate minor errors like switching rows and columns ✓



# Testing

---

## Issues/Resolutions:

Tabs require Common Controls Library, fails to compile otherwise

No objects in tab hierarchy, so need to create single child window to show/hide

Buttons call parent's handler, but since child window has no handler, either need to redefine class, override all, use hooks, or final solution: make child of main window, so children of window have different coordinates than children of tabs

String methods were inconsistent in memory (delete[] necessary for buffers) so use std::stringstream class over cstdlib char\* conversion functions but COMPLEX("(" + string + ",0)") vs. COMPLEX(atof(string)) - speed advantage for atof

COMPLEX vs. double - accurate solutions vs. fast solutions with imaginary exception handling (no pun intended)

User formatting fixes for improperly sized inputs e.g. "1,2,3\r\n 1,2" means nothing

Try to display imaginary numbers as "a+ib" rather than default "(a,b)" with proper minus signs

Displaying cout << this->func().toString() requires func() and dependencies are constant expressions (does not edit \*this)

OPEN: WIN32 API still uses low-level long character pointers for strings with obfuscated typedefs. (Learn a better GUI library.)

# End-User Product Documentation

---

## Physics Math Helper

### General Information:

After starting the program, **SD1617.exe**, you will be met with the default Vectors tab.

The red x in the corner along with File → Exit will close the program.

Help → About will direct you to view this documentation

Data will be saved between tabs, but will be destroyed upon exiting the program.

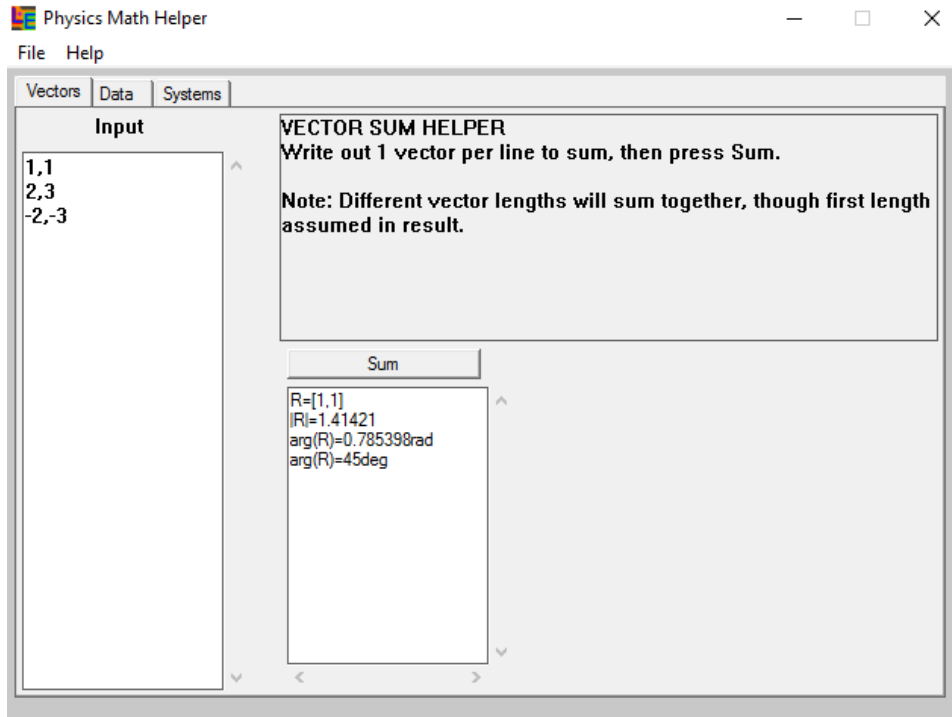
**Enter** will move to the next line

**Ctrl + C** will copy

**Ctrl + X** will cut

**Ctrl + V** will paste

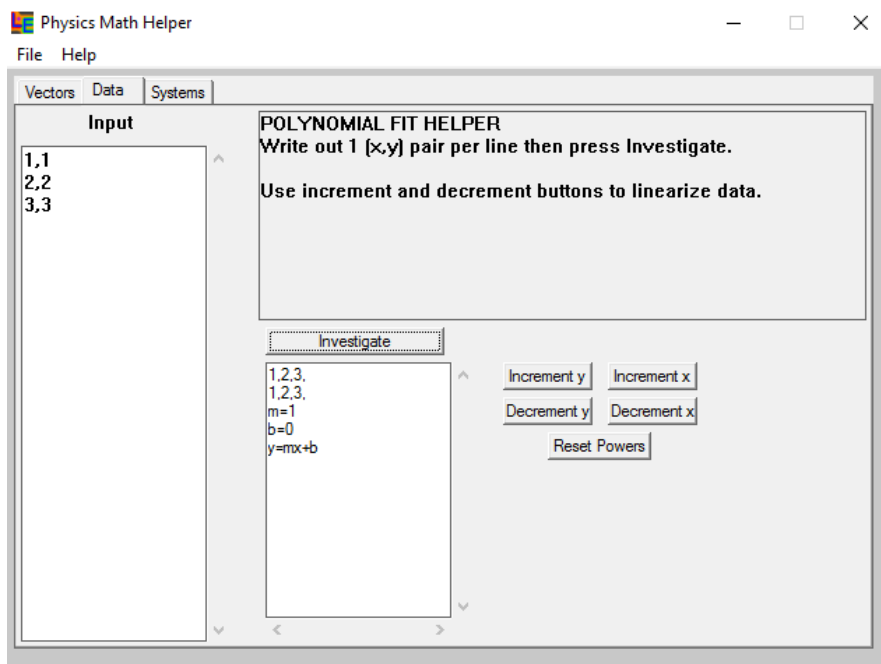
# End-User Product Documentation- Graphical Breakdown



## Vectors Tab

- **Input**
  - Format: Vector components separated by commas
  - Example: if  $\vec{a} = 7\hat{i} - 6\hat{j} + 8.5\hat{k}$ , input " 7,-6,8.5 "
  - Note: If  $\vec{a}$  has N components, and subsequent entries have less, the rest are assumed to be 0.
- Press **Sum** to calculate outputs
- **Output**
  - R is the resultant vector (the sum of the inputs) in as components separated by commas all in brackets.
  - $|R|$  is the magnitude of the resultant  $|R| = \sqrt{R_x^2 + R_y^2 + R_z^2 + \dots} = \sqrt{\sum_i R_i^2}$
  - $\arg(R)$  is displayed if all vectors are all 2D and gives the angle from the +x axis counterclockwise to the resultant where  $\arg(R) = \arctan2(R_y, R_x)$  which puts signs on  $\arctan R_y/R_x$  outside  $(-\frac{\pi}{2}, \frac{\pi}{2})$ .

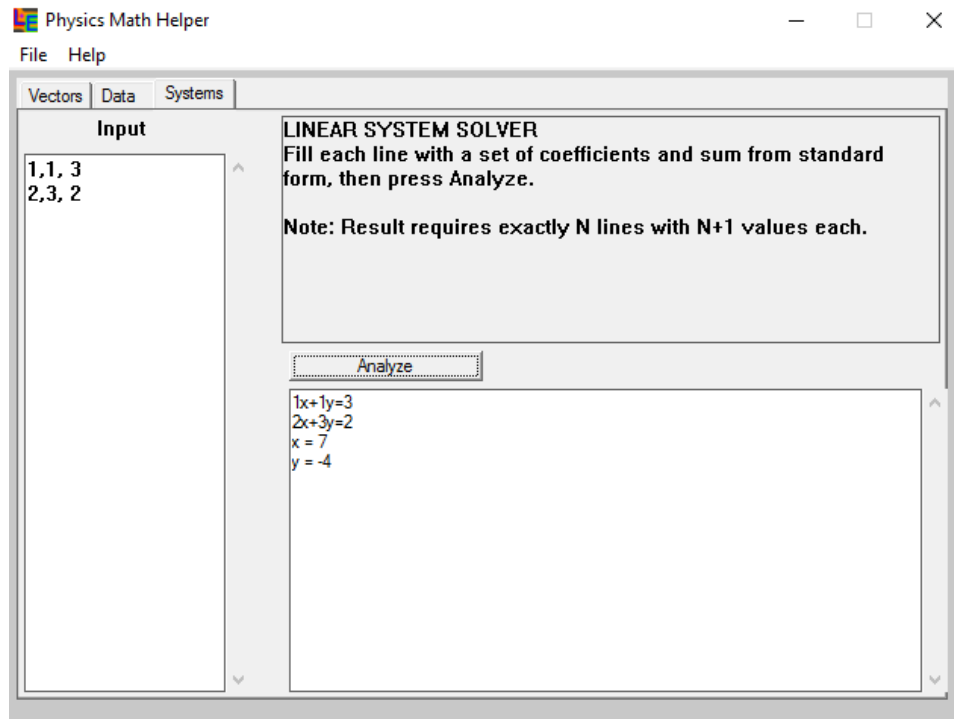
# End-User Product Documentation- Graphical Breakdown



## Data Tab

- **Input**
  - Format: coordinate pairs to determine correlation and fit of as the x coordinate and y coordinate separated by a comma
  - Example: (4,-3.1) as “ 4,-3.1 ”
- Press **Investigate** to calculate outputs
- Use **Increment**, **Decrement**, and **Reset** buttons to vary powers for a polynomial fit
- **Output**
  - a list of x values (transposed from first column for verification)
  - a list of y values (transposed from second column for verification)
  - m – the slope of the linear least squares line through the data set
  - b – the y-intercept of the linear least squares line through the data set
  - An equation for the line using variables and ^ for exponentiation
  - $m = \frac{N \cdot \sum xy - \sum x \sum y}{N \cdot \sum x^2 - (\sum x)^2}$  and  $b = \frac{\sum x^2 \sum y - \sum x \sum xy}{N \cdot \sum x^2 - (\sum x)^2}$
- Linearization allows nonlinear data sets to be fit by varying the powers of x or y data before using the above equations.

# End-User Product Documentation- Graphical Breakdown



## Systems Tab

- **Input**
  - Format: STANDARD FORM coefficients on variables followed by the sum. The last number in each row is assumed to be the sum
  - Example: if  $2x + 3y - 5z = 1.2$ , enter " 2,3,-5,1.2 "
- Press **Analyze** to calculate outputs
- **Output**
  - Each equation reconstructed with variables
  - A list of each variable's solution
  - OR an explanation of why a solution could not be found
  - $Ax = b \quad x = A^{-1}b = \frac{1}{|A|} [(-1)^{i+j} M_{ij}]^T b$
  - $O(N^3)$  in arithmetic

# Evaluation and Future

---

The current iteration of our program is definitely up to snuff. It performs all of the calculations that we set out to do in a rapid pace. But we are never truly done improving. A few ideas for the future of the program includes adding units, graphing the equations presented, solving linear ordinary differential equations, automatic minimization for possible error in large-scale operations, and changing the GUI library to allow cross-platform use. Each of these ideas would certainly require extra work, but they are absolutely wise choices to make.

# References

---

"Complex number class." *C++ Reference*. cplusplus.com, 2015 Web. 10 Dec. 2016

Mattuck, Arthur. "Lecture 25: Homogeneous Linear Systems with Constant Coefficients: Solution via Matrix Eigenvalues (Real and Distinct Case)." MIT OCW: 18.03 Differential Equations. MIT, Spring 2003. Lecture.

Siek, Jeremy G., and Andrew Lumsdaine. "The matrix template library: A generic programming approach to high performance numerical linear algebra." *Computing in Object-Oriented Parallel Environments*. Springer Berlin Heidelberg, 1998. 59-70. Web. 10 Dec. 2016

Strang, Gilbert. MIT OCW: 18.06 Linear Algebra. MIT, Spring 2010. Lecture.

Taussky, Olga. "On matrix classes corresponding to an ideal and its inverse." *Illinois Journal of Mathematics* 1.1 (1957): 108-113. Web. 13 Jan. 2017

"Windows API Index." *Microsoft Developer Network*. Microsoft, n.d. Web. 29 Dec. 2016.

## Student Copyright Checklist



DVD

# TSA LEAP LEADERSHIP RESUME – TEAM EVENT

The resume must be typed using 11pt Arial or Calibri font. For more information about how to complete the resume, visit this link: (<http://www.tsaweb.org/LEAP-competition-engagement>)

## **TEAM IDENTIFICATION**

**Team ID:** 2103-901

**Competitive event:** Software Development

**Level:** High School

## **LEADERSHIP EXPERIENCES** (specific to a competitive event)

Set up portable development environment to work during free time anywhere. (Be)

Researched common control API to implement tabs. (Know)

Broke down program into parts implementing everyone's ideas. (Do)

Personally thanked local college's math professor/department chair for details learned in his course. (Do)

**LEADERSHIP EXPERIENCES** (connected to one or more of these categories: *Leadership Roles; Community Service/Volunteer Experiences; Leadership Development/Training; College/Career Planning*)

### ***Leadership Roles***

Took lead in programming, constantly adopting feedback from the team. (Do)

Encouraged others to attend TSA work days (Do)

Worked as club officers within the school (Do)

Discussed principles and standards of each team member (Be)

### ***Leadership Development/Training***

Shared experience in C++ from interactive art projects, electronic ugly Christmas sweater, and previous entries. (Know)