PhotoGraph

VoidWalkers
Meet Kathiriya (160050001)
Shreyas Pimpalgaonkar (160050024)
Phuntsog Wangchuk (160050109)

October 29, 2017

Contents

1 Abstract

We provide an android application that scans handwritten mathematical equations to produce graphs and stepwise solutions in real time. The application includes added features for useful operations on matrices. Many calculations that are rather tedious to be done by hand, can be solved in seconds by using our application.

2 Motivation

- Mathematical calculations often get very lengthy and time consuming, distracting students from real studying.
- Need for resource to allow students to focus more on exercising their minds on concepts and techniques rather than lengthy calculations.
- Better visualization of solutions through graphs.
- Resource needs to be easy to use and fit seamlessly into users solving process.

3 Problem statement

Create an Android Application that can read mathematical equations handwritten on paper, solve them and display step by step solutions. The application should include graphing capabilities as well as active, evolving template recognition environment that can be regularly updated.

4 Features

Features Promised

- Produce graphs and step wise solutions
- Create a login forum and enable facility of submitting templates

Features Delivered

- Produced graphs and stepwise solutions
- Didn't feel the need to enable login, submitting unknown templates was enough so implemented that only

5 Tools used

- Android Studio
- Java
- Javascript
- LaTeX
- HTML
- Mathpix API
- Desmos API
- javamail API
- JavaDoc

6 Introduction

6.1 Optical Character Recognition

We use the highly efficient Mathpix API for performing this task. When a photo is clicked, it gets stored into a temporary location and is sent to the Mathpix servers. A LaTeXstring is sent by the API which we later process by our own custom algorithms.

6.2 String Matching

We maintain a dictionary of known pattern types. Each pattern calls a function that dumps a step into a solution, converts the string into another known pattern 'closer' to the standard form. Then, the function corresponding to that pattern is called and this action is carried on recursively until we get the complete solution.
Conversion to templates is done by Regular Expressions in JAVA. We scan for keywords and locate patterns of variables and estimate their orientation in the latex string.

6.3 Graph plotting

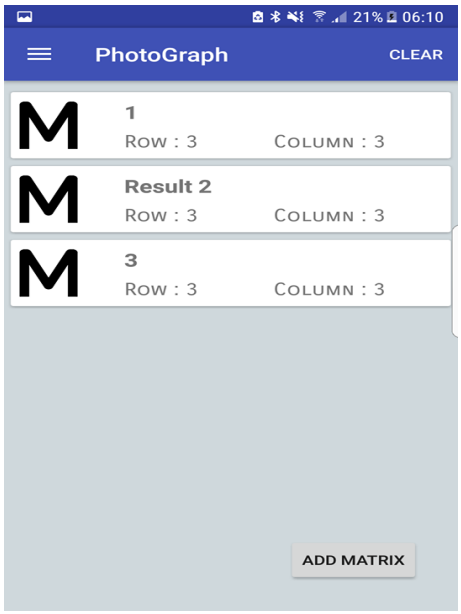
After receiving latex string we have plotted graph through desmos api and displayed it using java-script in webview container.

7 Implementation

7.1 Matrices

We use the same Mathpix API to scan matrices and convert to latex strings. We then obtain elements of matrices by parsing the latex strings.
After parsing these matrices we have added the following features.
Our UI contains of Menus, Drawers, ActionBars, CardViews to display and navigate.
We created a Matrix Class and added functions to facilitate the operations we performed. We also created a adapter class for matrix which displays matrices in a GridView.

We added the following functionalities



7.1.1 Addition

A fragment is displayed that contains a queue of matrices, list of added variables.
On clicking the elements of the CardView i.e. the matrices, we check whether the matrices can be added together.
So two or more matrices can be added to the queue and the total sum is then displayed which can be saved by the user. It then appears in the list of created variables. A and B be two mxn matrices then,

$$Sum[i][j] = A[i][j] + B[i][j]$$

7.1.2 Subtraction

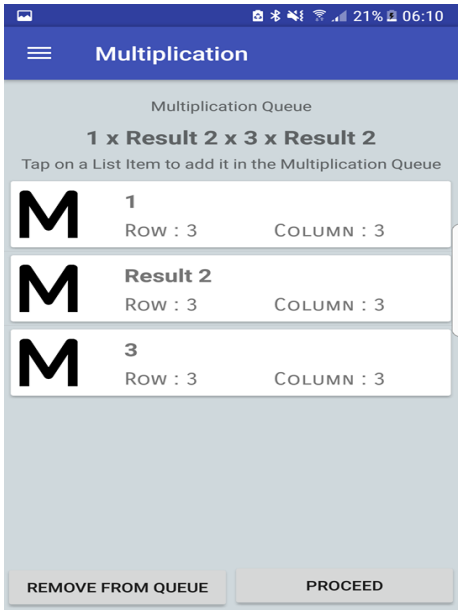
A fragment same as addition is displayed. Has same checks for subtraction of matrices. One or more matrices can be subtracted from a single matrix.

$$Sub[i][j] = A[i][j] - B[i][j]$$

7.1.3 Multiplication

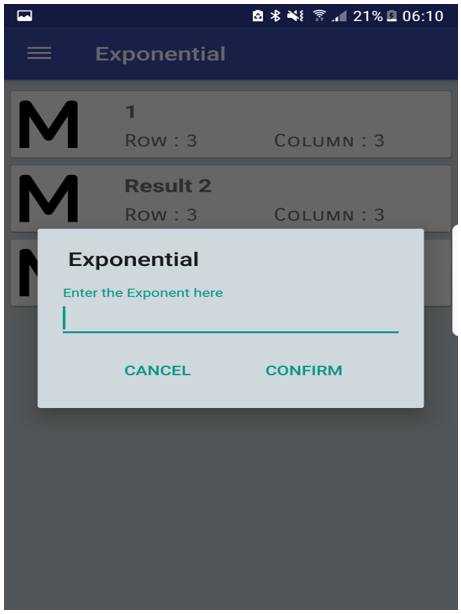
Same as addition again. On clicking the next matrix, the program checks whether the matrices are multipliable or not, else gives an error.

$$Mult[i][j] = \sum_{k=1}^{n} A[i][k] + B[k][j]$$



7.1.4 Exponent

Calculates the exponent of a matrix, uses the algorithm for computing the products in series



7.1.5 Transpose

Calculates the transpose of a square matrix by using the relation

$$A^n = A * A^{n-1}$$

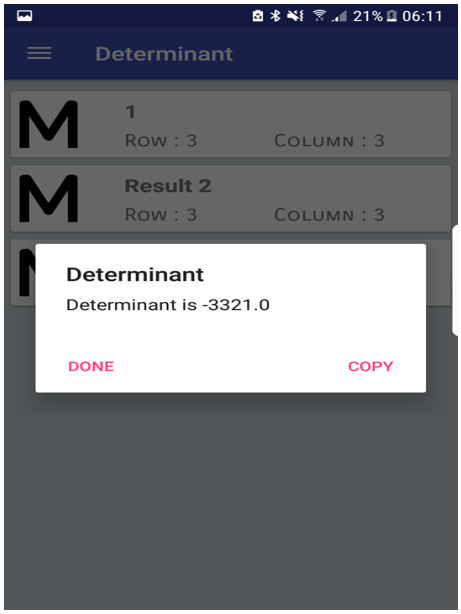
Stores it in a buffer matrix and displays the result.

7.1.6 Determinant

Calculates the determinant of a square matrix recursively

$$Det(A) = \sum_{i=0}^{n} (-1)^i M_{1i}$$

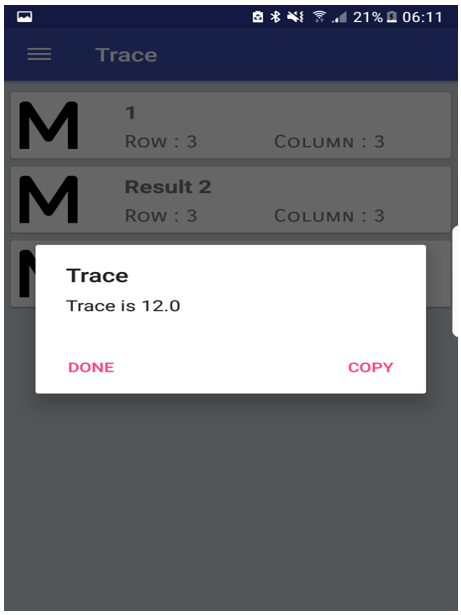
where, M_{ij} is the determinant of the matrix excluding i^{th} row and j^{th} column



7.1.7 Trace

Calculates the trace of the square matrix, sum of the diagonal elements.

$$Trace = \sum_{i=0}^{n} A[i][i]$$

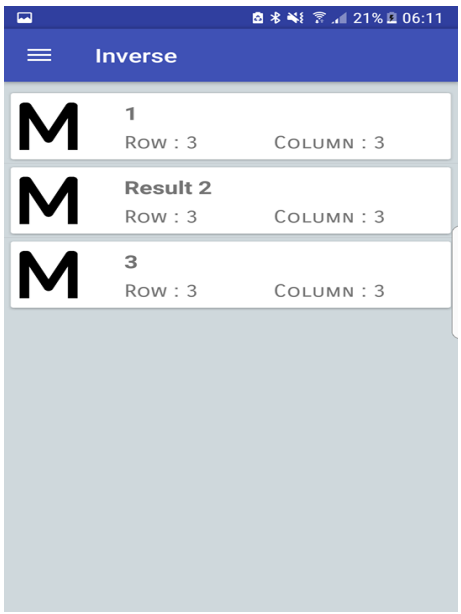


7.1.8 Adjoint

Calculates adjoint of the matrix, which is the transpose of cofactor matrix

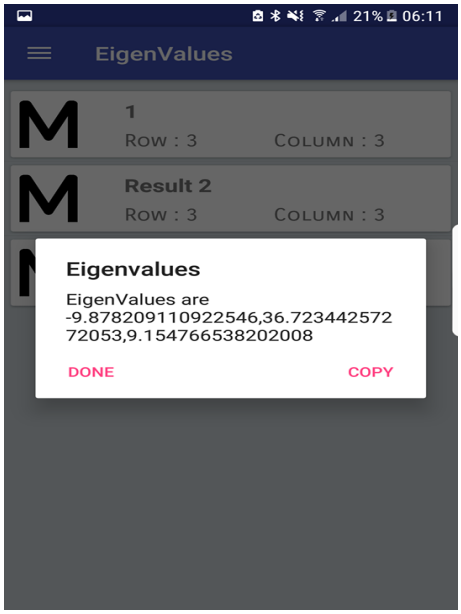
7.1.9 Inverse

Calculates Inverse by first calculating Adjoint



7.1.10 Eigenvalues

We have used apache library for carrying out this operation for time efficiency. This operation gives the eigenvalues values of the square matrix selected.



7.1.11 Eigenvectors

We have used apache library for carrying out this operation for time efficiency. This operation gives the eigenvectors of the matrix selected.

7.2 Linear Equations

The latex string always contains a comma which can be trivially found. Then we follow the following algorithm

1. Split the string with the separator ","
2. Remove all the whitespaces
3. Convert to templates
4. Call respective functions for each of the equations that give the stepwise solutions
5. Plot the graph

This will essentially generate the plot. We are using Desmos Engine for handling the plots. Given the latex string, it presents us the plot which we display it on the screen.
As our essential aim was to provide stepwise solutions, we call functions from the database of templates as follows
As we know our latex string contains a linear equation. It should contain two variables (currently supporting for two variables)

1. Search for all variables "x" , "y" , and replace the strings as "cx" and "cy"
2. Search for all types of strings "
   frac{a,b}" and replace them with the value of "a/b"
3. Replace all consecutive digits with c, and consecutive c's with one c
4. Replace all - with +

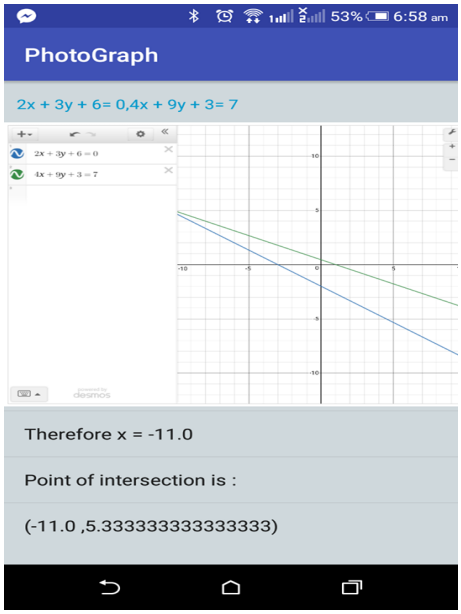
We call the resulting string a template. The equations

3x+4y=5 and 3y+4x=5

will have different templates

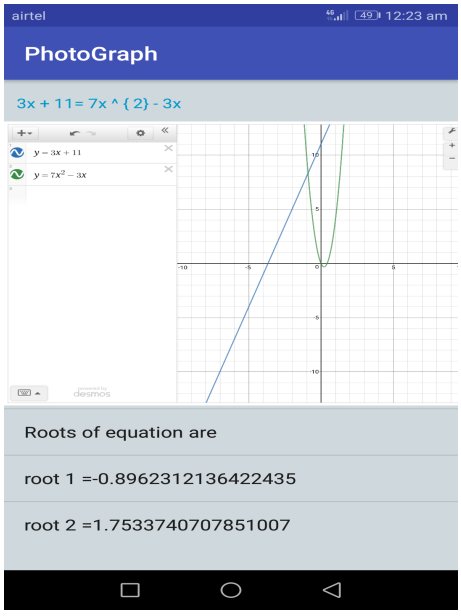
cx+cy=c and cy+cx=c

Once we get the templates, we call a function from the HashMap corresponding to that pattern. If the string is in standard form, we take out the coefficients and call the final two functions which solve the equations given the coefficients.
In the final answer, the input LaTeXstring is displayed as a TextView, the Graph is displayed in a WebView and the steps of the solution are written in a ListView inside a ScrollView.



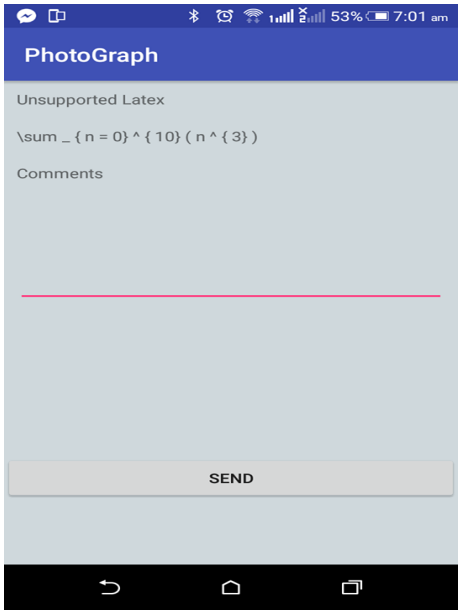
7.3 Quadratic Equations

For more versatility, we chose to do this by a traditional way. We created a function that directly gives coefficients from given latex string and another function that gives final stepwise outputs. To be able to do this we did the following things,



7.4 Other Templates

We have mainly implemented functionalities for Linear and Quadratic Equations. If a user enters equations not known by the application, that is the equations, which cannot be classified as any template, we have added the functionality of sending it to the admins of the application. A form appears in which user can add a comment and send an email to the admin directly.
To help the user get the answer immediately, we redirect the user's query to Wolfram Alpha, and display it right away within the app.



References

[1] https://www.regular-expressions.info/java.html

[2] http://www.vogella.com/tutorials/JavaRegularExpressions/article.html

[3] https://commons.apache.org/proper/commons-lang/javadocs/api-2.6/org/apache/commons/lang/StringUtils.html

[4] http://commons.apache.org/proper/commons-math/

[5] https://developer.android.com/guide/topics/manifest/uses-permission-element.html

[6] https://www.youtube.com/user/ProgrammingKnowledge

[7] https://https://thenewboston.com/

[8] https://github.com/Mathpix/api-examples

[9] https://docs.mathpix.com

[10] https://github.com/coder3101

[11] https://www.ics.uci.edu/ alspaugh/cls/shr/javadoc.html

[12] https://github.com/steveholt55/JavaMail-API-Android

[13] https://www.desmos.com/calculator