**Calculator Instructions**

This is a math/matrix calculator. It supports functions for algebra and trigonometry.

The user can open a .txt file to calculate or save the calculation into a .txt file

Program functions support max 9\*9 Matrix calculations supports priority calculation.

Matrix and calculation can be read from json files for computing result.

To open .json file, click Menu then click open.

Please see SimpleData.json for the sample format. (other json file is design for testing purpose, each is

built base on its name. Result.json is sample result output.)

You can click '<' '>' or 'âˆš' to change from different operations that been read from the correct .json

files.

**Functions**

functions can be entered by press the button

The matrix must first set the row and column then press the button to do the calculate

The user must choose if the matrix calculates with one or two matrixes

The people can analyze an input sequence and return the max, min, and average value.

**Function Button Description**

ABS |x| Return the absolute value of x

Analyze AS Input with a sequence numbers, and return the max, min, and average

Value of the numbers, use space to separate different numbers

Space Space input a blank space to separate the numbers

Bracket () Set the priority of the calculation

**Matrix-Function Button Description**

Add + Add one matric and another

Minus - Subtract the second matrix from the first

Multiply \* Multiply the first matrix with the second

Determinant Det Return the determinant of the matrix

Inverse Inverse Return the inverse matrix of the given one

QR decomposition QR Return the decomposition two matrix

SVD decomposition SVD Return the decomposition two matrix

Matrix trace Trace Return the specific trace of the matrix

Powers of a matrix Power(n) matrix times n of itself

LU decomposition LU Return the decomposition two matrix

Gaussian Elimination Gauss-Elimination Return the matrix after row reduction

EiVector EiVector Return the eigenvector of the matrix

EiValue EiValue Return the eigenvalue of the matrix

**Arithmetic Operators**

Once you press the '=' button or press 'Enter', the expression is evaluated according to normal algebraic operator precedence. That is, parentheses first, followed by exponentiation, multiply, divide, add and subtract.

**Function Button**

Add +

Subtract -

Multiply ×

Divide ÷

Exponent exp

**Constants**

These two constants enter the value at high precision.

π Pi – approximately 3.142…

e e - approximately 2.718…

**BUG Report**

1. Cannot always find all the real eigenvalues of a Matrix.

Theoretically, a QR algorithm with Wilkinson-Shift will always converge for hessenburg matrices. See https://www.ams.org/journals/mcom/2002-71-240/S0025-5718-01-01387-4/S0025-5718-01-01387-4.pdf. However, although a QR algorithm with Wilkinson-Shift is implemented in this program, it seems sometimes, though very rare, the QR algorithm still does not converge. To deal with this, we tried to do some QR iteration with normal shift if the QR algorithm with Wilkinson-Shift doesn't converge in 1000 iterations, but still it cannot make sure that the algorithm always converges. If the QR algorithm does not converge in 4000 iterations, the QR algorithm will be stopped. This led to the fact that our algorithm cannot always find all the real eigenvalues of a Matrix. This bug might be fixed in the future by more studies on the QR algorithm and its implementation in the program.

2. Floating-point error might cause inaccuracies

In this program, since many calculations are operations with double, due to the floating-point error, the result might not be accurate. Sometimes, the floating-point error might be enlarged significantly by the Butterfly Effect in the complicated calculation processes. To deal with this, in order to decrease the impact of floating-point error, we now require the input matrix to be no larger than 9\*9, and the input numbers must be in the range [-100000, 100000] with no more than 3 decimal digits. This issue might be fixed in the future by changing some double type to java.math.BigDecimal to decrease the floating-point error.