RationalFuncCalc User Manual

By: Kevin Chen, Ian Fox, Arthur Guo

Table of Contents

1.0 - General Information…………………………………………………………………………1

1.1 - System Overview

1.2 - Organization

2.0 - System Summary……………………………………………………………………………..2

2.1 - System Configuation

3.0 - Getting Started……………………………………………………………………………….3

3.1 - Graphical User Interface

4.0 - Using and Testing the System………………………………………………………………...4

4.1 - Parsing

4.2 - Field Values

4.3 - Functions - Polynomials

4.4 - Functions - Rational Functions

4.5 - Root Finding

1.0 General Information

This is where general information on the rational function calculator is called and explains the general purpose for the application.

1.1 System Overview

RationalFuncCalc is a rational function calculator that takes a user input of any rational function or polynomial and graphs it in a new window. It can find important information that will be discussed in the ‘features’ below. Currently, the status of this project is alpha testing. Not all the bugs have been reported and some still remain. If bugs are found, please report them to [KaiwenChenApple@gmail.com](mailto:KaiwenChenApple@gmail.com). The application operates on the coding language Java.

1.2 Organization

This user manual is split into 4 parts. Those parts are general information, system summary, getting started, and using and testing the system.

General Information discribes all the information needed to properly run the software as well as how the entire manual is organized. It also includes an email where bugs can be reported.

System Summary gives an summary of how the application works, the requirements of the software, and the access levels.

Getting Started helps you use the graphical interface properly.

Using and Testing the System gives an overview of the functions inside the applet itself, in case there is a bug or the user wants to access and change the app to their benefit.

2.0 System Summary

This section includes the summary of how the system works, requirements to run this software, and user access levels.

2.1 System Configuration

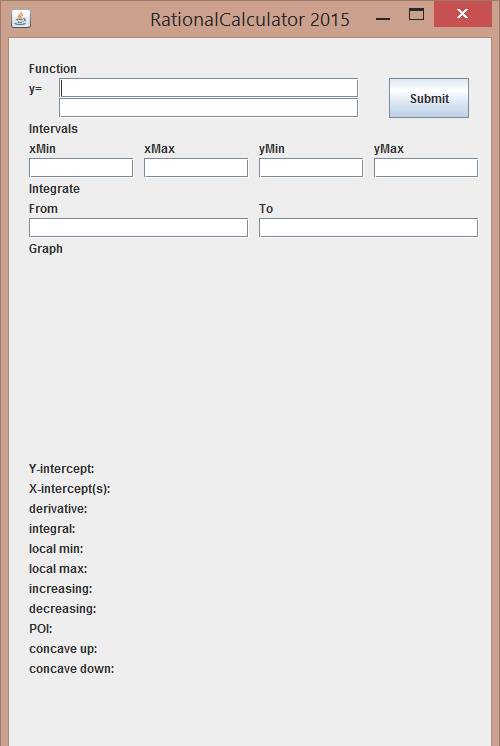
RationalFuncCalc operates on Java for Mac and PC. It is compatible with anything that can run a Java applet or has java installed. It requires no internet capabilities.

3.0 Getting Started

When RationalFuncCalc is opened, it displays the graphical user interface. This section will help you use the interface properly.

3.1 Graphical User Interface

The graphical user interface looks like this:



As shown, it includes user inputs at the top. These are where the user puts the polynomials or rational functions that they want to test. There is also the xMin, xMax, yMin, and yMax values. This is the window size for the graph. It plots everything inside the window size. The definite integral boxes are for the user if they want to find the area underneath the graph from a certain value to another value. Press the ‘Submit’ key for the program to analyze the inputted data and show the allotted outputs.

4.0 Using and Testing the System

This section helps explain exactly how the applet works and allows for easy changing of the program.

4.1 Parsing

The parsing function, located in Input.parse, first strips all spaces from the given string, then separates the string into factors by splitting at each set of brackets. Each factor is then parsed individually by collecting all of the coefficients along with their respective powers, and creating a new polynomial for each term. The resulting polynomials are multiplied together to obtain the final product.

4.2 Field Values and Constructors

Field values for a polynomial are located in the polynomial class. First, the constructor either takes a polynomial or various polynomials if it is a rational function (two polynomials: one on numerator, one on denominator). Since the polynomials can be given in two ways, factored and standard form, which both include their own benefits; there are separate constructors for both. Factored goes to a constructor that recognizes that the parts inside the brackets can be used to find the roots before multiplying it out. It saves those roots in the root field value under polynomial. After the multiplication, it then saves it as an array as the arguments of a polynomial. Standard takes all of the like terms, adds them together and saves it as arguments as the field values of a polynomial. There is also a third constructor which takes a string, parses it into the polynomial it represents and returns that.

Polynomials have a field value called arguments which is an array of coefficients, each one’s position corresponding to their respective power. For example, a 2.5 at the 0th position of the array represents 2.5x^0, whereas a -5 at index 3 would represent -5x^3. Rational functions simply have a field for the numerator, which is a polynomial, and a field for the denominator, another polynomial. Additionally, polynomials have a field which is an array-list of roots, and rational functions have an array-list of roots as well as an array-list of holes, and an array-list of asymptotes.

4.3 Functions: Polynomials

Functions that are included in polynomial are constructors, addition, subtraction, multiplication, division, derivative, root finding, showing polynomial as a standard string, definite integrals, local maxima and minima finding, increasing and decreasing interval finding, concavity, points of inflection, and evaluating the polynomial at a specific point.

4.4 Functions: Rational Functions

Functions that are included in the rational functions side of the project are the addition, subtraction, and multiplication of both other rational functions and polynomials. There is also an evaluate function for rational functions and function that puts a rational function into a string that calls the polynomial version of it for both the numerator and denominator.

4.5 Root Finding

A lot of functions circle around the root finding function. To start, the increasing/decreasing interval, local maxima and minima, concavity and point of inflection functions of both polynomial and rational functions all rely on the root function by finding roots of the first and second derivative. The function is also very complex. This function has 5 subsect root finding functions and categorizes polynomials depending on which one is best. For binomials, it finds the root simply the way we would, bringing the constant to the other side and dividing by lead coefficient to isolate ‘x’. Next, it is quadratic, it saves the roots using the quadratic formula. If neither, it first guesses all possibilities using the rational root theorem. If that still does not work, it uses Laguerre’s method and Newton’s method to solve for roots. After this is done, the roots are then found of the first and second derivatives to solve for the other important informations on the function, such as asymptotes, increasing/decreasing intervals, points of inflection, and concavity. That information is then sent to the screen.