



Development of Calculator Software for Evaluating Some Mathematical Functions

Ipeaveda, F. W¹, Binuyo, G. O² and Binuyo, A. O¹

¹Department of Physical Sciences, Ajayi Crowther University, Oyo, Nigeria.

²African Institute of Science Policy and Innovation, Obafemi Awolowo University, Ile-Ife, Nigeria.

ARTICLE INFO

Article history:

Received: 4 October 2012;

Received in revised form:

18 September 2015;

Accepted: 21 September 2015;

Keywords

Calculator,
Numerical Integration,
Polynomial solver,
Simultaneous equation solver,
Graph plotter.

ABSTRACT

In this paper, we present a design and implementation of a calculator software which applies the Microsoft visual studio and C-sharp (C#) for effective performance of some operations like polynomial solver of degree 10, simultaneous equation solver up to five systems of equations and evaluating the numerical integrations of some selected functions. It is expected that this application will help students and teaching staff in their daily doing of some mathematics related tasks.

© 2015 Elixir All rights reserved.

Introduction

Calculator can be defined as a hand held device that performs mathematical calculations. A calculator can also be a program on a computer that simulates a handheld calculator or offers more sophisticated calculation features [1].

Modern calculators generally have many more features than a standard four or five function calculator and the feature set differs between manufacturers and models [2]. However, the defining features of a scientific calculator includes: scientific notation, floating point arithmetic, trigonometric functions, logarithmic function using Base 10 and Base e, exponential functions and roots beyond the square root, Quick access to constants such as pi (π) and exponential (e).

A standard calculator is rectangular in shape and has a keypad through which numbers and operations are entered, as well as a display on which the entered numbers and the results of calculations are shown. Modern calculators can perform many types of mathematical computations, as well as permit the user to store and access data from memory. Common handheld calculators have the ability to use complicated geometric, algebraic, trigonometric, statistical and calculus functions. Many can also be programmed for specialized tasks. Calculators operate on electrical power supplied by batteries, solar cells, or standard electrical current. Modern calculators have digital displays, usually using some form of LCD [3].

Calculator programs are common accessories included with most personal computer operating systems. For example, both the Macintosh and Windows operating systems have a simple desktop calculator program [3, 4].

In this paper, we have developed scientific calculator software for effective performance of some mathematical operations. These operations include, polynomial solver (able to solve a polynomial equation of degree 10), simultaneous equation solver (able to solve systems of linear equation of

degree 5), graph plotter (able to plot the graph of basic mathematical functions on the real line axis) and evaluating the numerical integration of some selected functions.

Design Methodology

The Integrated Development Environment (IDE) used is Microsoft Visual Studio 2008 and the programming language used is C-sharp (C#) while Windows form is used for the Graphical user interface (GUI). C# is called an object-oriented language, which differs in approach from procedural languages like Visual Basic, Pascal, Cobol and so on.

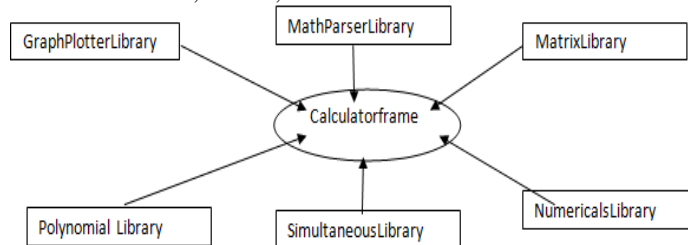


Figure 1. Improved Calculator Architecture

Software development is basically divided into two tasks: Graphical User Interface (GUI) design and coding, which is usually the code behind the GUI. None of the two is trivial though, while GUI design requires a sound creative mind and has to do with aesthetic properties, coding requires some technical skills and sound background in Mathematics, and has to do with functional properties of the software [5].

The following subheadings discuss each of the referenced namespaces (or libraries) that were created and used in the development of the calculator software:

Calculator Frame: It hosts the windows form used in the GUI design of the calculator

Simultaneous Library: It references the MatrixLibrary namespace and used it to solve the System of Linear Equations. It accomplishes this by first creating an instance of the

Simultaneous Class which contains reference to the MatrixClass library that contains implementation code for solving nth order simultaneous equation 'LU' decomposition method [6].

PolynomialLibrary: It contains two classes; the complex class that is used to analyse and represent both complex and real numbers that could result in the solution of any given polynomial equation, and the PolynomialClass that solves any polynomial equation given. The Polynomial class implements the 'Homer scheme' to evaluate any given polynomial and uses the Weierstrass iteration to compute the root of

Matrix Library: It contains the matrix class that is referenced in the SimultaneousClass, used to solve any simultaneous equation given. The matrix class used matrix operations (or simultaneous equation), using LU decomposition method. The class contains codes that implement LU decomposition method of solving system of linear equations. The traditional Gaussian Elimination method is not suitable in that it's prone to run-time errors, as the pivot element cannot be predetermined and the algorithm itself has some limitations [7].

Graph Plotter Library: It contains the Graph Class that is referenced in the Calculator Frame namespace to plot graphs. Specifically, three types of graphs: Linear, Quadratic and Trigonometry. The codes in the GraphClass simply implement the standard formulae used in plotting these graphs in real life.

Math Parser Library: It contains a namespace called MathFunction which contains two classes: MathParser and SpecialFunc. MathParser is used to evaluate basic arithmetic functions while SpecialFunc is used to simplify trigonometry identities and is passed as the code behind the standard calculator.

NumericalLibrary: It contains a class called Integration. The integration class contains implementation code for our numerical integration (functions that return numerical computation of Simpson's rule and trapezoidal rule for some selected function used [8].

Experimental Design

This section shows Graphical User Interface design used in the paper. The paper opens into a six-tabbed windows application, having history pane on the extreme right for tracking calculations. The first tab contains the basic mathematical operations. It contains the functions that you will find in a normal calculator, but it gives the option to select whether the calculation is going to be purely arithmetic or standard, which will as a result disable and enable some functions. The second tab is Polynomial, it can solve up to 10th degree polynomial and displays the result in complex form if any complex roots results. The third tab is the Simultaneous, it can solve any order of simultaneous equation, but because of limited space on our user interface, it is programmed to solve up to 5th order simultaneous equation i.e. 5 system of equations in 5 unknowns. The fourth tab is Graphplotter, it is programmed to plot Linear, Quadratic and Trigonometry equations. You can plot the 3 types of graph on the same graph or clear the screen if you like. The interfaces look something like the ones shown below.

First Tab: Basic Operations

An arithmetic expression is typed in the text screen from the calculator's buttons and the result is given out just in time. There is no accumulation of result or monotonous computation as an ordinary calculator does.

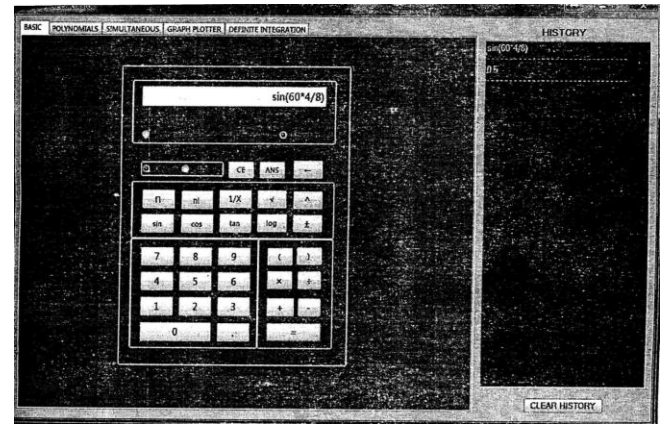


Figure 2. Diagram showing the basic arithmetic calculations
Second tab: polynomial operations

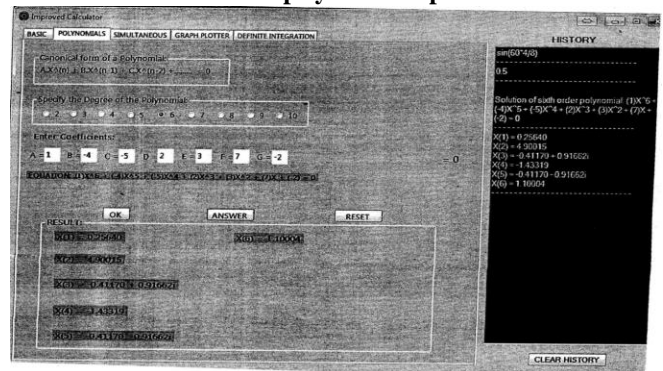


Figure 3. Diagram Showing The Polynomial Calculations Up To 10th Degree

Third Tab: Simultaneous Equations Operations

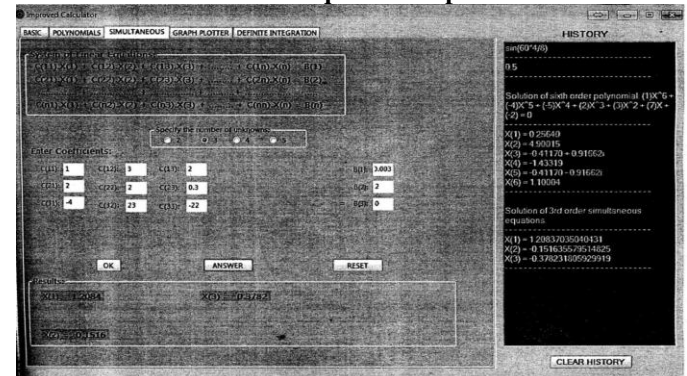


FIGURE 4. Diagram showing the simultaneous equation solver up to 5 systems of equation

Fourth Tab: Graph Plotting Operations

This helps to plot the graph of basic mathematical functions on the real line axis.

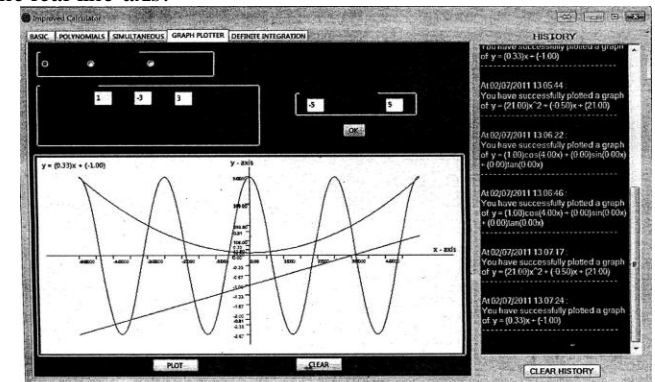


FIGURE 5. Diagram showing the graph plotter

Fifth Tab: Definite Integration (A)

This helps to evaluate the numerical integration of some selected functions using trapezoidal rule.

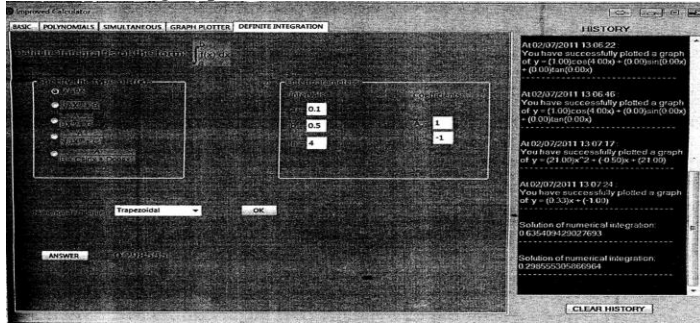


FIGURE 6. Diagram showing the numerical solutions using the trapezoidal method.

Sixth Tab: Definite Integration (B)

This helps to evaluate the numerical integration of some selected functions using Simpson's rule.

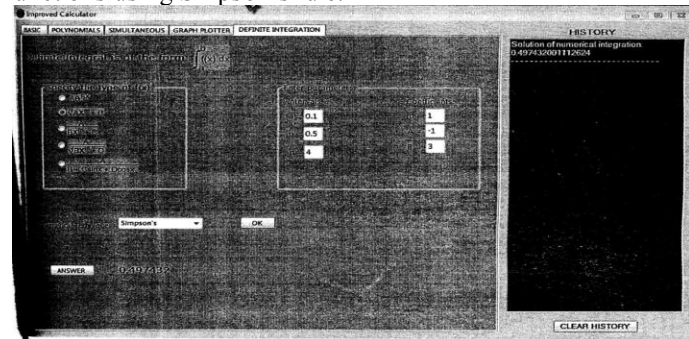


FIGURE 7. Diagram showing the numerical solutions using the Simpson's rule.

Conclusion

This calculator software has been developed, tested and found to meet the intended requirements. It is our expectation that this application will help students, academic and non-academic staff in their daily doing of mathematical related tasks. This software is meant to be a suitable substitute for similar software in the sense that it reduces the tedium of manually

computing some of the formulas we use in advanced Mathematics like Simpson's rule and trapezoidal rule for numerical integration.

References

1. N. Reznichenko. "Learning with a Graphing Calculator (GC): GC as a cognitive tool". Paper presented at the Annual EERA Conference, Clearwater, FA, February 2007.
2. C. Bardini, R.U. Pierce & K. Stacey. "Teaching Linear functions in context with graphics calculators: Students' responses and the impact of the approach on their use of algebraic symbols". *International Journal of Science and Mathematics Education*, vol 2, pps 353-376, 2004,
3. D. Guin and L. Trouche. "The complex process of converting tools into mathematical instruments: The case of calculators". *International Journal of Computers for Mathematical Learning*, Vol.3 (3), 195-227, 1999.
4. A.T. Graham. "Building a Versatile Understanding of Algebraic Variables with a Graphic Calculator". *Educational Studies in Mathematics*, Vol. 41, 265-282, 2000.
5. L. Ball & K. Stacey. "Good CAS Written Records: Insights from teachers". In H. Chick et al. (Eds.). *Proceedings of the 29th Conference of the International Group for the Psychology of Mathematics Education*, Vol.2, pp.113-120, Melbourne: PME, 2005
6. M. A. Mariotti. "Influencies of technologies advances in students' math learning". In L. D. English (Ed), *Handbook of International Research in Mathematics Education*, pp. 757-786. Mahwah, New Jersey: Lawrence Erlbaum Associates Publishers, 2002.
7. Y. Todd & M. J. Mohlenkamp. "Introduction to Numerical Methods and MATLAB programming for Engineers". Department of Mathematics, Ohio University, Athens, OH45701, 2011.
8. B. D. Bunday & H. Mulholland. "Pure Mathematics for Advanced Level". 9th edition, Published by Heinemann Educational Books limited, Halley Court, Jordan Hill, Oxford OX2 8EJ, UK, 1982.