GVU-J.SHT *GVU Journal of Science, Health and Technology*

GVU-J. SHT 2025,**10**; **1**, 97–101, doi:10.5281/zenodo.15436479

WINDOW-BASED AUTOMATED MATHEMATICAL SOLVING APPLICATION MODEL

Odikwa, N. H.

[1]*Department of Computer Science, Abia State University, Uturu, Nigeria Corresponding Author: Ndubuisi.odikwa@abiastateuniversity.edu.ng, +2348102380567*

Received: *24 April 2025 []* ***Published:*** *16 May 2025*

The phobia exhibited in solving mathematical problems by many primary, secondary, and tertiary pupils and students has posed a significant challenge in society, compounded by the limited number of mathematics teachers in schools. This paper proposes a dynamic approach to automating mathematical solutions, making the system widely accessible for solving mathematical problems. The system integrates the capabilities of Python and Django programming languages to develop a window-based automated mathematical solving application model. The application supports both voice and text inputs, parsing mathematical expressions from the input string. The parsed input is preprocessed by a mathematical engine, which generates the required output. This model analyzes and procedurally presents the mathematical solutions in a user-friendly manner. When deployed, the system achieved an accuracy of 98% compared to other automated mathematical applications. The solving capabilities of various mathematical solution steps were evaluated, yielding positive results.

Keywords: Window-based, Parameters, Mathematics solver, Automated

Introduction

In a rapidly advancing world, the need to accelerate measurement and calculation processes is critical. While some calculations can be easily performed with a calculator, others involve complex strategies that require significant effort to master. Mathematical calculations typically rely on applying formulas to derive solutions. However, across the globe, many individuals struggle to comprehend mathematical computations. In primary, secondary, and even tertiary institutions, mathematics is often perceived as a formidable subject. Students frequently find it challenging to perform simple calculations, and the formulas required to solve mathematical problems can be particularly daunting. A recent study revealed that 75% of students expressed a dislike for mathematics, while only 25% showed interest in the subject (Mathematical Association of Nigeria, 2022). (Dongziang et al., 2019) noted that solving mathematical word problems is particularly difficult, likely due to the disconnect between the semantic meaning of such problems and the logical processes required to solve them.

Mathematical Automation

The automation of mathematical problem-solving methods offers a powerful tool to reduce solving time. Jollanda (2020) suggested that the primary objective of automated reasoning is the automation of mathematics. Automated systems leverage technology to perform mathematical tasks without human intervention For (2015) developed an automated assessment system for mathematics, enabling users to access current knowledge and applications in the field. This system incorporates a comprehensive mathematics database containing details of mathematical equations. Essentially, an automated system functions as an artificial intelligence, capable of independently performing tasks after learning. Ching (2013) argued that mathematics education should involve automated tasks, allowing students to learn independently of a teacher. In this context, (Olena et al., 2019) created a dynamic program for mathematical knowledge control, integrating scientific mathematical principles into an application that guides users in solving geometric problems. Additionally, to make mathematics more accessible to individuals with disabilities, Amjad & Shah(2023) developed a SymPy-based automated mathematical equation analysis and solver, aiding those with physical hand impairments in studying comfortably. Similarly, (Amjad et al., 2024) introduced an interactive learning system for mathematical expressions tailored for students with visual disabilities, utilizing Braille to facilitate learning, which has produced promising results.

Application of Artificial Intelligence in Mathematics

Recent advancements demonstrate that artificial intelligence can power innovative tools, such as AI-powered calculators (Neil et al., 2023). Artificial intelligence can also be applied to solve mathematical problems. (Olena et al., 2019), in an article titled “Automation of Mathematical Knowledge Control within Dynamic Mathematics Programs,” posited that computer aided tool can facilitate mathematical knowledge assessment even if not through formal verification.

This paper aims to address the widespread phobia of solving mathematical problems by developing an automated model that enhances accuracy and efficiency in solving a wide range of mathematical challenges

Materials and Methods

The architecture of the window-based automated mathematical solving application model is illustrated in Figure 1. The model is divided into five distinct modules.

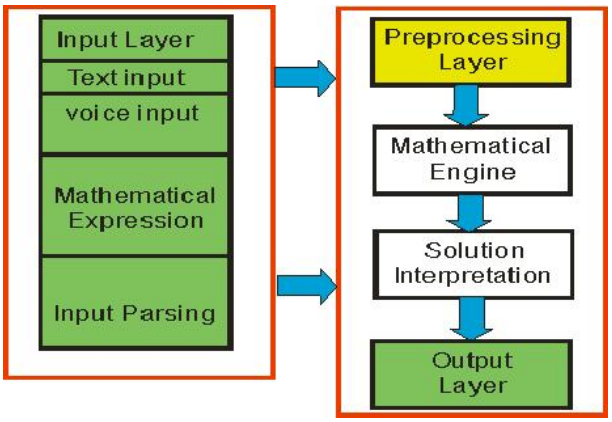

Figure 1 Architecture of the System

(1) **Input Layer:** The input layer comprises the graphical user interface (GUI), developed using the Java programming language, where users input mathematical problems via text or voice methods. This layer simultaneously handles mathematical expressions and input parsing.

(2) **Preprocessing Layer:** The natural language of mathematical expressions, including complex notations, is modeled to extract the underlying mathematical problems. This is crucial as solutions are often expressed in English. Symbolic transformations are applied to problems with symbolic variables, which are interpreted and rewritten into standardized mathematical forms, such as algebraic notations and logical propositions.

Algorithms:

(a) **Area of Rectangle**
- Step 1: Input the length of the object.
- Step 2: Input the width of the object.
- Step 1: Add the lengths of all sides.

(b) **Area of Square**
- Step 1: Input the length of one side.
- Step 2: Multiply the length by itself.

(c) **Area of Triangle**
- Step 1: Input the base.
- Step 2: Input the height.
- Step 3: Multiply the base, height, and divide by 2

(d) **Perimeter of Objects**
- Step 1: Add the lengths of all sides.

(e) **Pythagoras (Right-Angle Triangle)**
- Step 1: Input the length of the opposite side.
- Step 2: Input the length of the adjacent side.
- Step 3: Square the length of the opposite side.
- Step 4: Square the length of the adjacent side.
- Step 5: Add the squared sides and take the square root to find the hypotenuse.

(f) Factorization (Quadratic Formula)
- Step 1: Input the coefficient of the first value.
- Step 2: Input the coefficient of the second value.
- Step 3: Input the constant.
- Step 4: Multiply the coefficient of the first value and the constant.
- Step 5: Square the coefficient of the second value.
- Step 6: Subtract the result from step 4 from the result of step 5 and take the square root.
- Step 7: Multiply the coefficient of the first value by 2.
- Step 8: Add a negative prefix to the coefficient of the second value, add the result from step 6, and divide by the result from step 7.
- Step 9: Add a negative prefix to the coefficient of the second value, subtract the result from step 6, and divide by the result from step 7.

(3) **Mathematical Engine:** The mathematical engine is the core of the automation system, responsible for solving mathematical problems. It includes a symbolic solver that simplifies algebraic equations and expressions, handling integrals, derivatives, and other symbolic expressions. It also supports linear algebra solvers for matrices and linear equations, as well as root-finding for factorizations.

Area of Rectangle

$$Area = LxW \tag{1}$$

Where L is the length (longer side) and W is the width (shorter side)

$$Area of Triangle = 1/2(a \times b) \tag{2}$$

Where a, is the length of the base and b is the length of the opposite side.

$$Area of Square = L^2 \tag{3}$$

Where L is the length of one side of the square

$$Area of Trapezium = \frac{1}{2}(a+b) * h \tag{4}$$

Where a, is the length of the base, b is the length of the opposite side and h is the height.

Pythagoras

$$x_1 = -b + \frac{\left(\sqrt{b^2} - 4 \times a \times b\right)}{(2 \times a)} \quad (5)$$

$$x_2 = -b - \frac{\left(\sqrt{b^2} - 4 \times a \times b\right)}{(2 \times a)} \quad (6)$$

Where b, is the coefficient of the second value of the expression, a is the coefficient of the first value of the expression.

Perimeter of Objects

- **Square**

Perimeter = 4 × L (7)

where L is the length of a side.

•Rectangle

Perimeter = 2 × (L + W) (8)

where L is the length and W is the width.

•Circle

Perimeter = 2 × π × r (9)

where π = 3.142 and r is the radius.

•Triangle

Perimeter = a + b + c (10)

where a, b, c are the lengths of the sides.

•Parallelogram

Perimeter = 2 × (L + W) (11)

where L is the length and W is the width.

•Trapezium

Perimeter = a + b + c + d (12)

where a, b, c, d are the lengths of the sides.

(4) **Solution Interpretation and Simplification:** This module interprets the computed results from the mathematical engine, presenting them in a user-friendly manner. For example, the perimeter of a trapezium is calculated by summing the lengths of its four sides, with results expressed in meters or centimeters. The perimeter of a circle (circumference) is computed by multiplying 2 by 3.142 (approximate value of π) and the radius. This approach extends to other mathematical problems.

Experimental Results and Discussion

Numerous experiments were conducted to validate the effectiveness of the window-based automated mathematical solving application model. These experiments aimed to assess the accuracy and time required to solve various mathematical problems.

The graphical user interface (GUI) of the automated mathematics solver model serves as a comprehensive platform listing all solvable mathematical problems. As shown in Figure 2, the application enables users to select a mathematical problem, displaying the step-by-step procedure for solving it. The application also presents the computed results upon completion.

The automated solver for simultaneous equations is depicted in Figure 3. The process uses alphabetic representations for coefficients in two equations: a_1 for the first value, b_1 for the second, and $c1$ for the third in the first equation; similarly, $a2$, $b2$, and

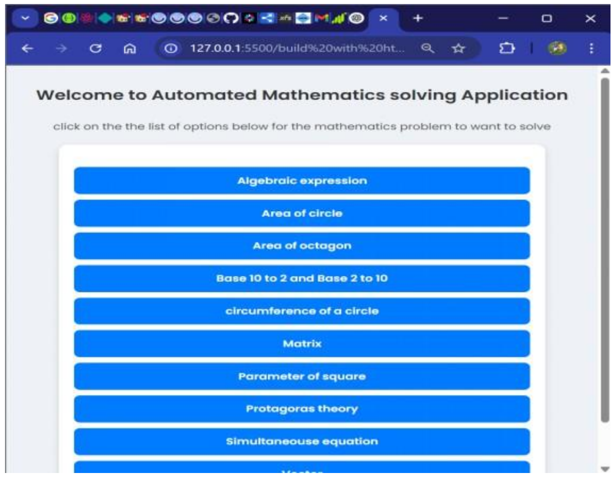

Figure 2 Automated Mathematics Solving Application

c_2 for the second equation. The steps are automated to display the rigorous procedures leading to the solution.

The automated vector solver, shown in Figure 4, handles calculations such as dot product, vector addition or subtraction, magnitude, unit vector, angle between vectors, and projection or components. The process is automated, displaying the steps taken to achieve the results. The matrix solver, illustrated in Figure 5, seamlessly addresses matrix operations, including addition, subtraction, multiplication, inverse, transpose, eigenvalue and eigenvector computation, and solving systems of equations using matrices. For Pythagoras problems, as shown in Figure 6, the solver considers a right-angle triangle with sides labeled a, b, and c (adjacent, opposite, and hypotenuse). The application calculates any side by taking the square root of the sum of the other sides squared, providing detailed steps.

Figure 7 illustrates the solver for perimeter calculations, using the square perimeter as an example to demonstrate the system's approach to solving perimeter problems.

Table 1 Comparison with other Automated Mathematics Application Models

Maths Solver	Accuracy in Analysis
Wolfram Alpha	80%
Microsoft Math Solver	90%
Photomath	85%
Symbolab	75%
GeoGebra	90%
Mathway	84%
Desmos	82%
Window-Based Automated Solver	97%

The window-based automated solver was compared with other automated mathematical solving applications. The analysis indicates that its accuracy and clarity in solving mathematical problems surpass other models, attributed to enhanced features like step-by-step procedural displays. Figure 8 presents the time (in seconds) required to solve various mathematical problems. The solver takes 1 second to solve problems such as the area of a circle, triangle, square, and perimeter of a square. It requires 2 seconds for the area of a trapezium, perimeter of a circumference, trapezium, and triangle. Factorization and Pythagoras problems are solved in 3 seconds, with the slightly increased time due to the involved steps. Matrix problems take 4 seconds, while geometry and integration problems require 5 seconds. Overall, the solving time ranges from 1 to 5 seconds, demonstrating improved performance over previous solvers, largely due to the integrated mathematical engine.

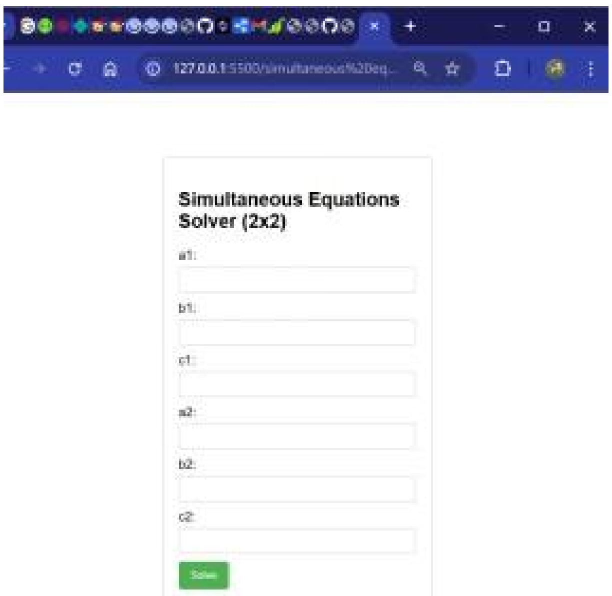

Figure 3 Interface Simultaneous Solver

Figure 4 Interface Vector Solver

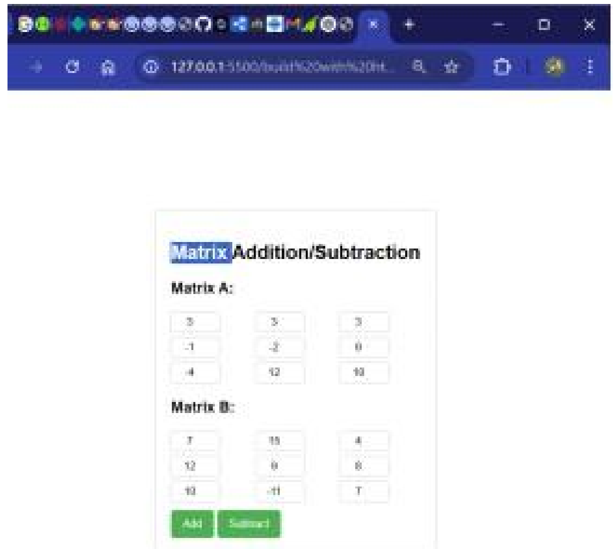

Figure 5 Interface Matrix Solve

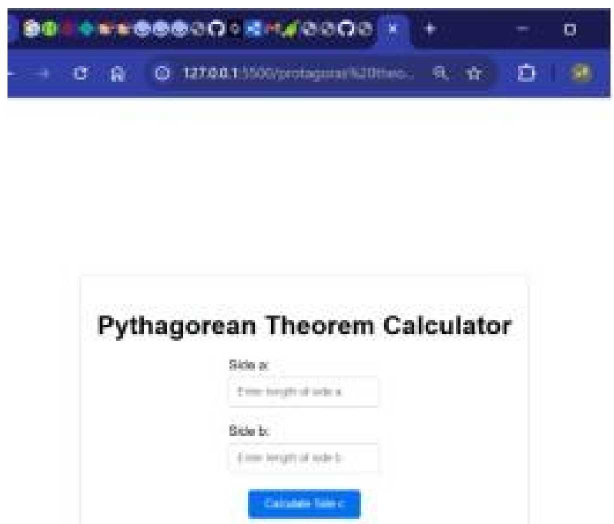

Figure 6 Interface Pythagoras Solver

Figure 7 Interface Square Perimeter Solve

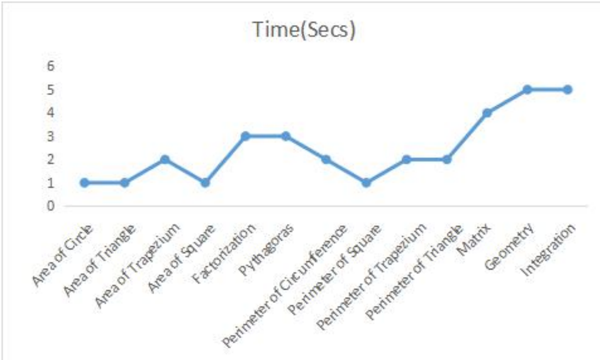

Figure 8 Time Determination in Solving Mathematical Problems

Conclusion

The computational ability of this system is tremendous and, when compared with other mathematical solving methods, demonstrates high accuracy. The primary motivation is its abiity to analyze how mathematical computations are performed while prioritizing automation. The system's robustness is evident in its capacity to solve virtually all computational problems in mathematics. This automated mathematical solving application model is recommended to mathematicians and individuals seeking accurate and timely solutions.

References

Amjad, A., & Shah, K. (2023). SA-MEA: Sympy-based Automated Mathematical Equation Analysis and Solver. Elsevier, 1–7.

Amjad, A., Shah, K., & Tahani, J. A. (2024). Accessible Interac- tive Learning of Mathematics Expressions for School Students with Visual Disabilities. PeerJ Computer Science, 4 (5), 1–14.

Ching-Sung, K. (2013). Automation of Mathematics Teaching. International Journal of Computer Science and Mobile Computing, 2 (7), 125–136.

Dongxiang Zhang, Lei Wang, Luming Zhang, Bing Tian Dai, & Heng Tao Shen. (2019). The Gap of Semantic Parsing: A Survey on Automatic Math Word Problem Solvers. arXiv.org.

Jollanda, S. (2020). The Automation of Mathematical Rea- soning. Available at: https://www.researchgate.net/publication/346425938, 1–12.

Neil, B. B., Khrisna, S., Angie, Y., Rowena, T., Sheila, M. L., & Kimberly, M. C. (2023). Artificial Intelligence-Powered Calculator Application Usage in Mathematics Summative Assessments. IRE Journals, 6 (10), 446–476.

Olena, S., Volodymyr, P., & Marina, D. (2019). Automation of Mathematical Knowledge Control Within Dynamic Mathemat- ics Programs. In Scientific Editor Eugenia Smyrnova-Trybulsk, 571–585.

Sergio, R. (2015). Automated Assessment in Mathematics. Institute of Electrical and Electronics Engineers Annual Computer Software and Applications Conference, 211–218.