

# Scientific Calculator Program in Octave

S.A.Abdulla

21 January 2025

# Contents

<b>Introduction</b>	<b>3</b>
<b>Prerequisites</b>	<b>3</b>
<b>Installation Instructions</b>	<b>3</b>
<b>How to Run the Calculator</b>	<b>3</b>
<b>UI of calculator</b>	<b>4</b>
<b>1 Usage Instructions</b>	<b>5</b>
<b>Usage Instructions</b>	<b>5</b>
1.1 Basic Usage . . . . .	5
1.2 Advanced Functionalities . . . . .	5
1.3 Equation Input and Display . . . . .	5
1.4 Memory Functions . . . . .	5
1.5 Statistical Calculations . . . . .	6
1.6 Number System Conversion . . . . .	6
1.7 Error Handling . . . . .	6
1.8 Example Usage . . . . .	6
Example Usage . . . . .	6
1.9 Program Structure . . . . .	7
Program Structure . . . . .	7
<b>2 Troubleshooting</b>	<b>9</b>
<b>Troubleshooting</b>	<b>9</b>
<b>3 Conclusion</b>	<b>10</b>
<b>Conclusion</b>	<b>10</b>

## Introduction

This scientific calculator program developed in Octave is designed to offer a wide range of mathematical functionalities, making it suitable for both basic and advanced calculations. The calculator allows users to perform standard arithmetic operations like addition, subtraction, multiplication, and division. However, its true strength lies in its ability to handle scientific and complex mathematical equations with precision and ease.

## Prerequisites

- **Software Required:**
  - GNU Octave version 9.0 or higher.

## Installation Instructions

- **Operating System:** The program works on Windows, macOS, and Linux.
- **Octave Version:** GNU Octave 5.0 or higher is required for full compatibility.
- **RAM and Processor:** The calculator is lightweight and runs efficiently on most modern systems with minimal resource requirements.

## How to Run the Calculator

1. Open Octave.
2. Download or copy the `calculator.m` file to a directory on your system.
3. In the Octave command window, navigate to the directory where `calculator.m` is saved using the `cd` command:

```
1  cd('path/to/directory')
2
```

4. Run the calculator by typing:

```
1  octave calculator.m
2
```

## UI of calculator

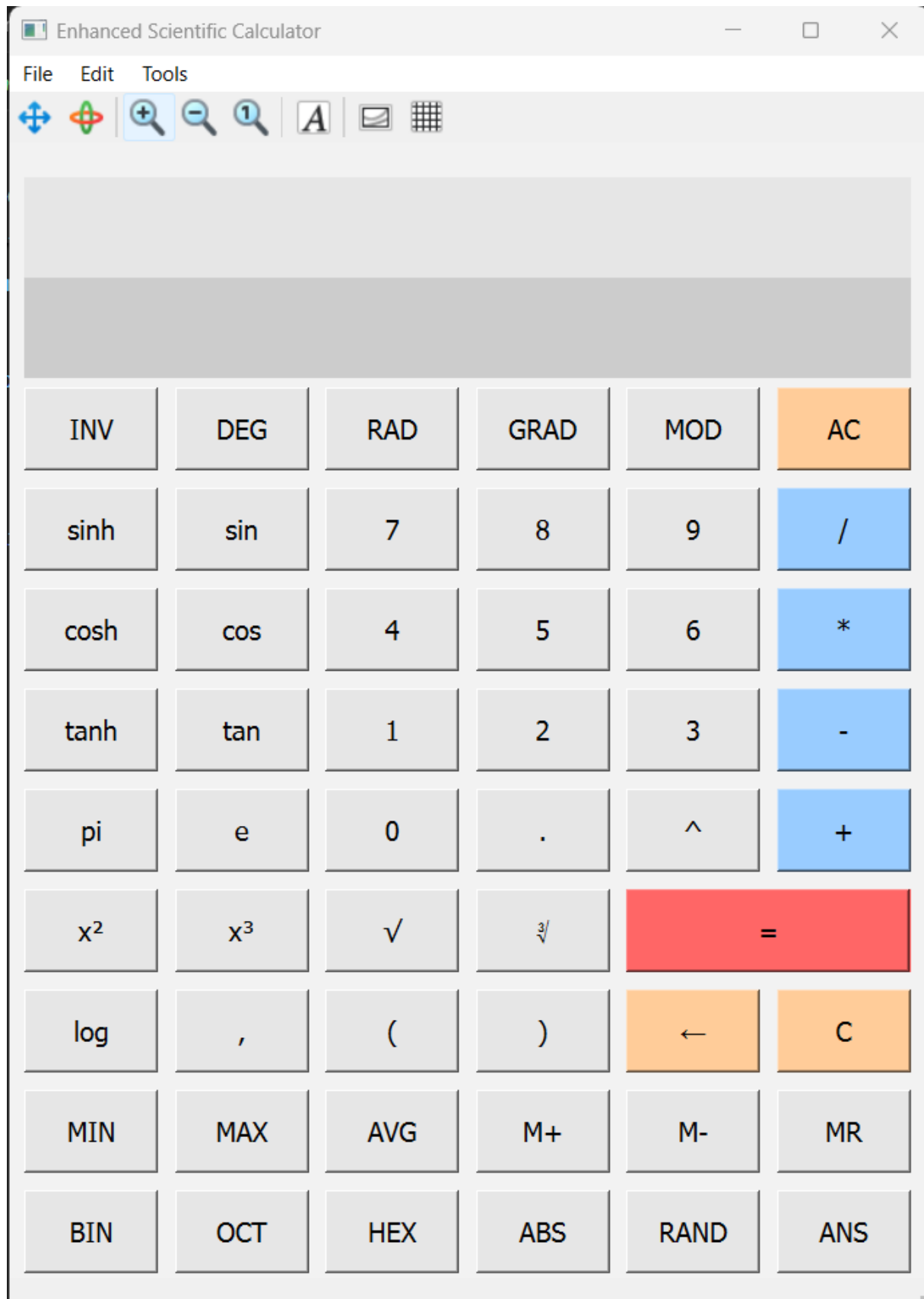


Figure 1: Overview of the Scientific Calculator Interface

# 1 Usage Instructions

a

Once the program starts, the calculator interface will be displayed, mimicking a physical scientific calculator. The user interacts with the calculator by selecting buttons for operations, entering numbers, and constructing equations. The interface dynamically displays the entered equation and results separately, ensuring clarity.

## 1.1 Basic Usage

The calculator supports basic arithmetic operations such as:

- **Addition (+):** Press the + button after entering the first number, followed by the second number, to see the result.
- **Subtraction (-):** Press the - button to subtract one number from another.
- **Multiplication (\*) and Division (/):** Use the \* and / buttons respectively for these operations.

## 1.2 Advanced Functionalities

The calculator extends beyond basic arithmetic with a comprehensive set of scientific operations and features:

- **Trigonometric and Hyperbolic Functions:** Calculate `sin`, `cos`, `tan`, and their hyperbolic counterparts (`sinh`, `cosh`, `tanh`) by pressing the corresponding buttons. Use angle mode buttons (`DEG`, `RAD`, `GRAD`) to toggle the input angle unit.
- **Powers and Roots:** Compute squares ( $x^2$ ), cubes ( $x^3$ ), square roots ( $\sqrt{\phantom{x}}$ ), and cube roots ( $\sqrt[3]{\phantom{x}}$ ) effortlessly.
- **Scientific Constants:** Use constants like `pi` and Euler's number (`e`) in your equations without manual input.
- **Logarithmic Functions:** The `log` button calculates the natural logarithm of a number, while other base-specific logs can be added to future versions.
- **Modulo Operation:** Use the `MOD` button for modulus calculations to determine remainders.

## 1.3 Equation Input and Display

The calculator allows users to construct complex equations with parentheses, ensuring the correct order of operations. For example:

1. Enter the equation:  $(5 + 3) * 2$ .
2. Press the = (`equals`) button to calculate the result.
3. The entered equation is displayed clearly at the top of the interface, while the result is shown separately below.

## 1.4 Memory Functions

The calculator includes memory functionality to store and manage values:

- **M+:** Add the displayed result to memory.
- **M-:** Subtract the displayed result from memory.

- **MR (Memory Recall):** Retrieve the stored value for use in calculations.
- **Clear Memory:** If the memory is empty or needs resetting, the calculator will display `Memory Empty`.

## 1.5 Statistical Calculations

For statistical data analysis, the calculator provides:

- **MIN:** Calculate the minimum of a set of numbers.
- **MAX:** Determine the maximum value.
- **AVG:** Compute the average of a set of values.

## 1.6 Number System Conversion

The calculator supports conversions between number systems:

- **BIN, OCT, and HEX** buttons convert values to binary, octal, and hexadecimal formats respectively.

## Random Number Generation and Special Functions

- **RAND:** Generate random numbers.
- **ABS:** Calculate the absolute value of a number.
- **ANS:** Retrieve the most recent result for use in subsequent calculations.

## 1.7 Error Handling

The calculator is equipped to handle errors gracefully:

- If an equation is invalid (e.g.,  $5 / 0$ ), the calculator displays a clear error message like `Error: Division by Zero`.
- For empty memory or invalid operations, a helpful message such as `Memory Empty` or `Invalid Input` is shown.

## 1.8 Example Usage

Here are some examples demonstrating the diverse functionalities of the scientific calculator. For each equation, the input is displayed along with the corresponding result.

### 1. Basic Arithmetic:

- Equation:  $(3 + 5) * 2$
- Result: 16

### 2. Trigonometric Functions (Degrees):

- Equation:  $\sin(30) + \cos(60)$
- Result: 1.0

### 3. Trigonometric Functions (Radians):

- Set the angle mode to RAD.
- Equation:  $\sin(\pi / 4) + \tan(\pi / 6)$
- Result: 1.366

### 4. Hyperbolic Functions:

- Equation:  $\sinh(2) - \cosh(1)$

- Result: 2.350
- 5. **Exponents and Roots:**
  - Equation:  $x^3 + \sqrt[3]{27}$
  - Input:  $x = 3$
  - Result:  $27 + 3 = 30$
- 6. **Logarithms:**
  - Equation:  $\log(e)$
  - Result: 1.0
- 7. **Modulo Operation:**
  - Equation:  $15 \text{ MOD } 4$
  - Result: 3
- 8. **Statistical Functions:**
  - Equation:  $\text{MIN}(4, 12, 9, 20), \text{MAX}(4, 12, 9, 20), \text{AVG}(4, 12, 9, 20)$
  - Result:  $\text{MIN} = 4, \text{MAX} = 20, \text{AVG} = 11.25$
- 9. **Number System Conversion:**
  - Equation:  $\text{BIN}(10), \text{HEX}(255), \text{OCT}(8)$
  - Result:  $\text{BIN} = 1010, \text{HEX} = \text{FF}, \text{OCT} = 10$
- 10. **Random Number and Absolute Value:**
  - Equation:  $\text{RAND}()$  (generates a random number)
  - Result:  $\text{RAND} = 0.842$  (example)

Each of these examples illustrates a unique functionality of the calculator. You can mix and match these features to create more complex calculations, leveraging the full power of the scientific calculator.

## 1.9 Program Structure

The calculator program is composed of several modular components and functions, each dedicated to a specific task. Below is a detailed overview of its structure:

- **Main Calculator Function:**
  - The program begins with a primary function, `calculator`, which is responsible for creating the base frame of the calculator.
  - This includes setting up the display screen and generating all buttons for the user interface.
- **Button Logic:**
  - Each button is assigned specific functionality, and its behavior is handled by dedicated code blocks.
  - Buttons like "=", "M+", and "AC" have unique implementations to ensure their operations are accurate and responsive.
- **Backspace Logic:**
  - A separate function manages the behavior of the backspace button.
  - It ensures the correct number of characters is removed from the input field for each press, accounting for varying input types (e.g., symbols, numbers, or words).
- **Formatted Equation Handling:**
  - Complex equations are processed through a dedicated `formatted_equation` function.

- This function handles the parsing and formatting of equations, ensuring that inputs are syntactically correct and ready for evaluation.
- **Symbol and Keyword Handling:**
  - Specific functions handle mathematical constants and keywords.
  - For example, a function replaces the word "pi" with the actual constant value of  $\pi$ .
  - Similarly, operations like factorial ("!") are converted into their equivalent method calls.
- **Parentheses Management:**
  - A function is designed to track the usage of parentheses in equations.
  - It automatically adds missing parentheses to ensure completeness, reducing the likelihood of errors caused by forgotten or mismatched brackets.
- **Modulus Function Conversion:**
  - The word "mod" is automatically replaced with a functional representation, "mod()", and the appropriate parameters are passed to it.
- **Cuberoor Calculation:**
  - The program includes a dedicated function to handle cube root calculations, ensuring seamless integration with other operations.
- **Factorial Calculation:**
  - A specific function is implemented to calculate factorials, converting the symbol "!" into the corresponding factorial operation.
- **Helper Functions for Basic Arithmetic:**
  - Separate helper functions are used for fundamental operations such as addition, subtraction, multiplication, and division.
  - These functions are modular and reusable, contributing to the overall clarity and maintainability of the code.

This modular structure ensures that the program is organized, robust, and easy to maintain. Each function plays a distinct role in handling various aspects of the calculator's functionality, making it both efficient and user-friendly.



## 2 Troubleshooting

Despite its robust design, users may encounter occasional issues while using the calculator program. Below are some common problems and their solutions:

- **Problem: The program doesn't run or returns an error upon execution.**

**Solution:**

- Verify that **GNU Octave** is installed correctly and is accessible from your system.
- Ensure that the `calculator.m` file is saved in the correct directory.
- Use the `cd` command in the Octave command window to navigate to the directory where `calculator.m` is located before running the program:

```
cd('path/to/directory')
```

- **Problem: Incorrect results for operations involving trigonometric functions.**

**Solution:**

- Confirm that the angle mode (e.g., `DEG`, `RAD`, `GRAD`) is correctly set before performing calculations. The angle mode significantly affects trigonometric computations.
- Use the appropriate buttons (`DEG`, `RAD`, `GRAD`) to switch between modes.

- **Problem: Errors in equations with parentheses or complex syntax.**

**Solution:**

- Ensure that all parentheses in the equation are properly matched.
- If the equation is incomplete, the program attempts to autofill missing parentheses; however, for complex calculations, verify the equation manually.
- Use the backspace (`←`) and clear (`C`) buttons to correct any mistakes.

- **Problem: Memory functions do not work as expected.**

**Solution:**

- Ensure that a valid result has been stored using `M+`.
- Use `MR` to recall the stored value and confirm it is displayed correctly.
- If necessary, reset the memory by using `M-`.

- **Problem: Unexpected behavior with advanced functions (e.g., logarithms, roots).**

**Solution:**

- Check that inputs for functions like `log` or  $\sqrt[3]{\phantom{x}}$  are valid (e.g., no negative inputs for square roots).
- Refer to the usage instructions to ensure the correct syntax is being used.

### 3 Conclusion

The scientific calculator program developed in Octave offers a versatile and user-friendly solution for performing a wide range of mathematical computations. From basic arithmetic to advanced operations like trigonometric functions, powers, roots, and memory-based calculations, the program is designed to meet the needs of both casual and advanced users.

Key highlights of the program include:

- A comprehensive set of functionalities resembling those of a physical scientific calculator.
- A modular structure that ensures the program is both robust and easy to maintain.
- User-friendly features like memory storage (**M+**, **M-**, **MR**), advanced function handling (e.g., **MOD**, **BIN**, **HEX**), and error prevention mechanisms (e.g., parentheses management).

This program provides a solid foundation for further enhancements. Potential future developments include:

- Support for graphical plots and visualization tools.
- Integration of additional scientific functions like statistical operations and calculus.
- Development of a graphical user interface (GUI) to enhance usability further.

In its current form, the calculator serves as a reliable and efficient tool, whether for academic purposes, everyday calculations, or advanced mathematical problem-solving.