



OPEN ACCESS INTERNATIONAL JOURNAL OF SCIENCE & ENGINEERING

EVALUATION OF OPEN-SOURCE MATHEMATICAL SOFTWARE FOR ENGINEERING APPLICATIONS

Mahesh Prabhakar Thakare

Gopikabai Sitaram Gawande Mahavidyalaya, Umarkhed, Dist. Yavatmal
mahesh1995mahesh@gmail.com

Abstract: *Open-source mathematical software has become an essential component of engineering education, scientific research, and industrial applications. Unlike proprietary software, open-source platforms provide free access to computational tools, enabling engineers, researchers, and students to perform numerical analysis, symbolic computation, optimization, simulation, visualization, and data analysis without licensing restrictions. Popular open-source mathematical software such as GNU Octave, Scilab, SageMath, Python (NumPy, SciPy, SymPy, Matplotlib), Maxima, and FreeMat have emerged as reliable alternatives to commercial packages like MATLAB, Mathematica, and Maple. This study evaluates the performance, usability, computational efficiency, functionality, documentation, community support, and engineering applicability of leading open-source mathematical software. A comparative evaluation framework based on engineering performance indicators is developed to analyze their suitability for various engineering disciplines. The findings indicate that Python-based scientific computing ecosystems provide the highest flexibility and scalability, while GNU Octave offers excellent compatibility with MATLAB. SageMath and Maxima demonstrate strong symbolic computation capabilities, whereas Scilab performs efficiently in numerical simulation and control system analysis. The study concludes that open-source mathematical software provides cost-effective, scalable, and technically robust solutions for engineering education, research, and industrial problem-solving.*

Keywords: *Open-Source Software, Engineering Mathematics, Scientific Computing, Numerical Analysis, Python, GNU Octave, Scilab, SageMath, Computational Engineering.*

I. Introduction

1. Introduction

Engineering increasingly depends on computational mathematics for solving complex analytical and numerical problems. Modern engineering disciplines—including mechanical, civil, electrical, electronics, aerospace, chemical, and computer engineering—require mathematical software to perform modeling, simulation, optimization, statistical analysis, signal processing, and data visualization.

Traditionally, commercial mathematical software such as MATLAB, Mathematica, Maple, and Mathcad has dominated engineering computation because of their comprehensive functionality and professional support. However, the high licensing costs of proprietary software often limit accessibility, particularly in educational institutions, developing countries, startups, and small research laboratories.

Open-source mathematical software has emerged as a practical alternative. These platforms provide unrestricted access to computational algorithms,

source code, and development tools while encouraging collaboration among researchers and software developers. The availability of freely distributable software has significantly expanded opportunities for engineering education and research.

Among the most widely used open-source mathematical software are GNU Octave, Scilab, SageMath, Maxima, Python scientific libraries (NumPy, SciPy, SymPy, Pandas, and Matplotlib), FreeMat, and R. These tools support numerical computation, symbolic mathematics, matrix operations, differential equations, optimization, statistical analysis, machine learning, and scientific visualization.

Engineering institutions increasingly incorporate these software packages into undergraduate and postgraduate curricula because they enable students to gain practical computational experience without financial barriers. Furthermore, open-source ecosystems continuously evolve through contributions from global developer communities, resulting in rapid improvements in functionality and

performance.

This study evaluates the capabilities of major open-source mathematical software packages and compares their effectiveness for engineering applications using multiple computational performance indicators.

Objectives

The objectives of this study are:

- To examine major open-source mathematical software used in engineering.
- To evaluate computational performance using engineering criteria.
- To compare functionality across different software platforms.
- To identify software best suited for specific engineering applications.
- To recommend appropriate software for engineering education and research.

2. Literature Review

The adoption of open-source software in engineering has increased considerably over the last two decades. Numerous studies emphasize its economic, educational, and technical advantages over proprietary software.

GNU Octave has become one of the most popular alternatives to MATLAB because of its similar programming syntax and extensive numerical computation capabilities. Researchers have demonstrated that Octave performs effectively in matrix computation, linear algebra, optimization, signal processing, and numerical solution of differential equations.

Scilab has gained recognition for engineering simulation, control systems, optimization, image processing, and scientific visualization. Its graphical modeling environment (Xcos) provides simulation capabilities comparable to Simulink, making it valuable for control engineering education.

Python has become the dominant programming language for scientific computing due to its extensive ecosystem of open-source libraries. NumPy supports efficient numerical arrays, SciPy provides advanced scientific algorithms, SymPy enables symbolic mathematics, Matplotlib facilitates visualization, while Pandas simplifies data analysis. The integration of machine learning libraries further expands Python's engineering applications.

SageMath combines numerous mathematical packages into a unified computational environment supporting algebra, calculus, number theory, optimization, graph theory, statistics, and symbolic computation. Maxima specializes in symbolic mathematics, including differentiation, integration, matrix operations, and equation solving.

Comparative studies consistently report that open-source software offers comparable computational

accuracy to commercial software while significantly reducing financial costs. However, differences remain in documentation quality, technical support, graphical interfaces, and specialized engineering toolboxes.

3. Research Methodology

This research adopts a descriptive comparative evaluation methodology.

Evaluation Criteria

The software packages are evaluated using ten engineering performance indicators.

- Numerical computation
- Symbolic computation
- Engineering simulation
- Data visualization
- Programming flexibility
- Community support
- Documentation quality
- Computational efficiency
- Ease of learning
- Cross-platform compatibility

Rating Scale

Each software package is evaluated using a five-point scale.

Score	Performance
1	Very Poor
2	Poor
3	Average
4	Good
5	Excellent

4. Overview of Major Open-Source Mathematical Software

4.1 GNU Octave

GNU Octave is a high-level programming environment designed primarily for numerical computation. Its syntax is largely compatible with MATLAB, enabling engineers to migrate existing programs with minimal modification. Octave supports matrix operations, optimization, numerical integration, signal processing, control systems, and differential equations.

Major engineering applications

- Numerical linear algebra
- Mechanical engineering
- Electrical engineering
- Finite element computation
- Signal processing

4.2 Python Scientific Computing Ecosystem

Python has evolved into the world's leading open-source scientific programming language.

Important libraries include:

- NumPy
- SciPy

- SymPy
- Matplotlib
- Pandas
- Scikit-learn

Python supports numerical computation, symbolic mathematics, optimization, machine learning, artificial intelligence, image processing, scientific visualization, robotics, and engineering simulations. Its extensive community support and continuous development make it one of the most versatile computational environments available.

4.3 Scilab

Scilab is designed specifically for scientific computing and engineering simulation.

Its major strengths include

- Matrix computation
- Control system analysis
- Optimization
- Numerical simulation
- Signal processing
- Xcos simulation environment

Scilab is widely used in engineering laboratories because of its graphical modeling capabilities.

4.4 SageMath

SageMath integrates more than one hundred mathematical software libraries into a unified

computational platform.

Major capabilities include

- Symbolic algebra
- Calculus
- Number theory
- Linear algebra
- Statistics
- Optimization
- Graph theory

It is particularly valuable for higher mathematics and engineering research requiring symbolic computation.

4.5 Maxima

Maxima is one of the oldest open-source computer algebra systems.

It provides excellent symbolic computation capabilities including

- Differentiation
- Integration
- Matrix operations
- Differential equations
- Polynomial algebra
- Laplace transforms

Its lightweight architecture makes it suitable for educational environments.

5. Comparative Evaluation

Table 1. Comparative Performance of Open-Source Mathematical Software

Parameter	Python	Octave	Scilab	SageMath	Maxima
Numerical Computation	5	5	5	4	3
Symbolic Mathematics	4	3	3	5	5
Engineering Simulation	5	4	5	3	2
Visualization	5	4	4	4	2
Programming Flexibility	5	4	4	4	3
Machine Learning	5	2	2	3	1
Documentation	5	4	4	4	3
Community Support	5	4	4	4	3
Ease of Learning	4	4	3	3	3
Cross-platform Support	5	5	5	5	5

Preliminary Analysis

The comparative evaluation indicates that Python provides the highest overall performance because of its comprehensive scientific libraries, extensive community support, and versatility across engineering disciplines. GNU Octave demonstrates excellent numerical computing capabilities and strong compatibility with MATLAB, making it particularly attractive for engineering education. Scilab performs well in engineering simulations and control systems, while SageMath and Maxima excel in symbolic mathematics and theoretical computation.

6. Engineering Applications of Open-Source Mathematical Software

Open-source mathematical software has become indispensable across engineering disciplines due to its flexibility, cost-effectiveness, and extensive computational capabilities. Different software packages are optimized for specific engineering tasks.

6.1 Mechanical Engineering

Mechanical engineers frequently perform stress analysis, vibration analysis, finite element modeling, optimization, and thermodynamic simulations.

Python, GNU Octave, and Scilab provide efficient numerical solvers for matrix computations, differential equations, and optimization problems.

Typical applications include:

- Heat transfer analysis
- Structural mechanics
- Fluid dynamics
- Machine design optimization
- Numerical solution of engineering equations

6.2 Civil Engineering

Civil engineering applications involve structural analysis, transportation modeling, surveying computations, and environmental engineering. Open-source software supports finite element analysis, statistical modeling, and optimization of construction projects.

Common applications include:

- Structural load analysis
- Bridge design
- Traffic flow simulation
- Earthquake engineering
- Water resource management

6.3 Electrical and Electronics Engineering

Electrical engineers use mathematical software for signal processing, control systems, communication systems, and power system analysis.

Scilab with Xcos and Python provide efficient environments for:

- Digital signal processing
- Control system simulation
- Image processing
- Communication engineering
- Embedded system development

6.4 Computer Engineering and Artificial Intelligence

Python dominates computer engineering because of its integration with artificial intelligence, machine learning, and data science libraries.

Engineering applications include:

- Machine learning
- Computer vision
- Robotics
- Neural networks
- Big data analytics
- Internet of Things (IoT)

6.5 Chemical Engineering

Chemical engineers employ mathematical software for:

- Process optimization
- Reaction kinetics
- Numerical modeling
- Process control
- Statistical quality control

Python and Scilab provide powerful numerical methods for solving nonlinear engineering equations

encountered in chemical process design.

7. Performance Evaluation

The software packages were evaluated using the ten engineering performance indicators introduced in Part 1. The average scores are summarized below.

Table 2. Overall Performance Scores

Software	Total Score	Mean Score
Python	48	4.80
GNU Octave	39	3.90
Scilab	39	3.90
SageMath	39	3.90
Maxima	30	3.00

Python achieved the highest overall score because of its extensive ecosystem, programming flexibility, machine learning support, and strong community contributions. GNU Octave, Scilab, and SageMath demonstrated comparable performance with different areas of specialization, while Maxima remained highly effective for symbolic mathematics despite offering fewer engineering-specific features.

8. Comparative Analysis

Table 3. Software Suitability for Engineering Tasks

Engineering Task	Best Software
Numerical Analysis	Python / GNU Octave
Symbolic Mathematics	SageMath / Maxima
Control Systems	Scilab
Machine Learning	Python
Data Science	Python
Optimization	Python / Scilab
Engineering Simulation	Python / Scilab
Linear Algebra	GNU Octave
Scientific Visualization	Python
Educational Laboratories	GNU Octave / Scilab

The comparative analysis demonstrates that no single software package is optimal for every engineering application. Selection should be based on project requirements, computational complexity, available expertise, and institutional resources.

9. Graph-Ready Data

Figure 1. Mean Performance Scores

Software	Mean Score
Python	4.80
GNU Octave	3.90
Scilab	3.90
SageMath	3.90
Maxima	3.00

Figure 2. Computational Capability Comparison

Feature	Python	Octave	Scilab	SageMath	Maxima
Numerical	5	5	5	4	3

Feature	Python	Octave	Scilab	SageMath	Maxima
Computing					
Symbolic Computing	4	3	3	5	5
Simulation	5	4	5	3	2
Visualization	5	4	4	4	2
Machine Learning	5	2	2	3	1

10. Discussion

The evaluation highlights that open-source mathematical software has matured into a practical alternative to proprietary engineering tools. Python's extensive ecosystem of scientific libraries, excellent documentation, and active developer community make it highly suitable for multidisciplinary engineering projects. Its ability to integrate numerical computation, symbolic mathematics, machine learning, visualization, and data analytics within a single environment is a major advantage.

GNU Octave continues to be one of the best choices for institutions transitioning from MATLAB because of its similar syntax and reliable numerical capabilities. Scilab provides a strong environment for control systems, optimization, and simulation through its Xcos modeling platform. SageMath and Maxima remain valuable for advanced symbolic computation, theorem verification, and mathematical research.

Open-source software also promotes reproducible research because source code can be freely examined, modified, and shared. Furthermore, the absence of licensing costs allows universities, research institutions, and startups to deploy advanced computational tools on a large scale.

Despite these strengths, challenges remain. Some specialized engineering toolboxes available in commercial software may not have direct open-source equivalents. Additionally, beginners may require training to effectively utilize command-line interfaces and open-source development environments.

11. Conclusion

Open-source mathematical software has become a powerful and economically viable solution for engineering education, research, and industrial applications. The comparative evaluation conducted in this study demonstrates that Python, GNU Octave, Scilab, SageMath, and Maxima each possess unique strengths that support a wide range of computational engineering tasks.

Python achieved the highest overall performance because of its versatility, extensive scientific libraries, and integration with emerging technologies such as artificial intelligence, machine learning, robotics, and big data analytics. GNU

Octave provides excellent compatibility with MATLAB and remains particularly suitable for numerical engineering applications. Scilab offers strong capabilities in engineering simulation and control systems, while SageMath and Maxima excel in symbolic computation and mathematical analysis. The findings suggest that engineering institutions should adopt open-source mathematical software as a cost-effective complement—or, in many cases, an alternative—to proprietary platforms. Such adoption can enhance computational skills, promote reproducible research, and broaden access to advanced engineering tools. Future work may include benchmarking these software packages using large-scale engineering datasets, parallel computing environments, cloud-based workflows, and AI-assisted computational methods.

References

1. Chapra, S. C. (2020). *Applied Numerical Methods with MATLAB for Engineers and Scientists* (4th ed.). McGraw-Hill.
2. Eaton, J. W., Bateman, D., & Hauberg, S. (2022). *GNU Octave Manual*. Free Software Foundation.
3. Gilat, A., & Subramaniam, V. (2020). *Numerical Methods for Engineers and Scientists*. Wiley.
4. Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90–95.
5. Langtangen, H. P. (2016). *A Primer on Scientific Programming with Python*. Springer.
6. Millman, K. J., & Aivazis, M. (2011). Python for scientists and engineers. *Computing in Science & Engineering*, 13(2), 9–12.
7. Oliphant, T. E. (2015). *Guide to NumPy* (2nd ed.). CreateSpace.
8. Pérez, F., & Granger, B. E. (2007). IPython: A system for interactive scientific computing. *Computing in Science & Engineering*, 9(3), 21–29.
9. Quarteroni, A., Sacco, R., & Saleri, F. (2014). *Numerical Mathematics* (2nd ed.). Springer.
10. Ramalho, L. (2022). *Fluent Python* (2nd ed.). O'Reilly Media.
11. Scilab Enterprises. (2021). *Scilab Documentation*. Scilab Enterprises.
12. SageMath Developers. (2024). *SageMath Documentation*. SageMath Project.
13. Severance, C. (2016). *Python for Everybody*. CreateSpace.
14. SymPy Development Team. (2023). *SymPy Documentation*. SymPy Project.
15. The NumPy Developers. (2024). *NumPy User Guide*. NumPy Community.

16. The SciPy Community. (2024). *SciPy Reference Guide*. SciPy Community.
17. Van Rossum, G., & Drake, F. L. (2009). *Python 3 Reference Manual*. CreateSpace.
18. Virtanen, P., et al. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3), 261–272.
19. Weisstein, E. W. (2024). *MathWorld*. Wolfram Research.
20. Wilkinson, J. H. (2013). *The Algebraic Eigenvalue Problem*. Oxford University Press.