A Study of Python Programming in Mathematical Modeling and Computation

Vivekanandam B.[1], Dinesh Rajassekharan[2]

[1,2]Associate Professor, School of AI Computing and Multimedia, Lincoln University College, Malaysia.

E-mail: [1]vivekanandam@lincoln.edu.my, [2]diraja@lincoln.edu.my

Abstract

This study focuses on the application of the programming language of Python in mathematical modeling in engineering and science. Mathematical modeling refers to a process of representing phenomena of reality in the form of mathematical models through computations. The formulations created in a mathematical model are computed in order to analyze and simulate them. The Python language has become popular because it is open source, contains libraries and is computationally powerful. This research study provides an outline of the process of mathematical modeling using Python by considering some libraries, such as NumPy, SciPy, and Matplotlib. Furthermore, a comparative analysis of Python along with some other programming languages, such as MATLAB and COMSOL, is provided to compare the performance, usability and scalability. This study reveals that although there are some advantages of Python in terms of customizability, data-driven modeling and availability, there are some challenges related to its execution speed and computational effectiveness. Therefore, Python programming language is suitable for developing flexible research models.

Keywords: Mathematical Modeling, Python Programming, Scientific Computing, Numerical Simulation, Computational Analysis, Engineering Applications, Data Visualization.

1. Introduction

Mathematical modeling represents the real-world systems using mathematical derivations. It plays a major role in both engineering and sciences since this technique could be used to model many engineering processes such as heat transfer, fluid flow, structural behavior, and biological systems. The transformation of real-world physical problem into

DOI: https://doi.org/10.36548/rrrj.2026.1.015
Received:22.05.2026, received in revised form:25.06.2026, accepted:10.07.2026, published: 22.07.2026

mathematical equations will allow engineers and researchers to model different cases to save the cost of performing experiments and obtain an optimized system. With increasing complexity of engineering systems nowadays, mathematical modeling becomes increasingly important as a powerful tool for analyzing the working of the system and making decisions based on data [2], [8]. Figure 1 shows that the process of mathematical modeling is iterative and consists of repetitive process involving formulation, analysis, interpretation, and refinement of real-world problems.

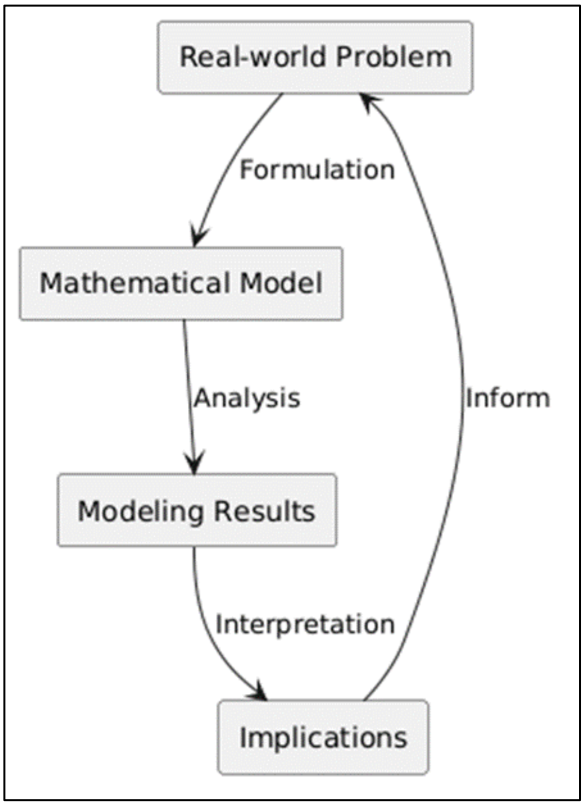

Figure 1. Conceptual cycle of the mathematical modeling process

Figure 1 shows the iterative process in the development of the mathematical model. The formulation of the problem in the real world gives the development of the mathematical model. Analysis of the mathematical model produces the outcome that will have certain implications on the problem formulated. These implications give insights into the understanding of the original problem.

Programming language bridges the theoretical modeling and computational modeling process. While theoretical modeling establishes a platform for deriving numerical solutions of mathematical equations, programming languages establish a tool through which numerical solutions are efficiently computed and simulated. Programming environments differ in their capabilities with respect to computations, ease of use and flexibility. For instance, MATLAB has rich built-in capabilities for numerical computation whereas lower-level programming languages can enable efficient simulation for very large problems. In consequence, there are various considerations that need to be considered while choosing programming languages as we transition from theoretical modeling to executable modeling. All these issues have a bearing on the efficiency and accuracy of the model [1], [2]. Mathematical modeling through computational implementation, as illustrated in Figure 2, includes algorithm development, program execution and result analysis.

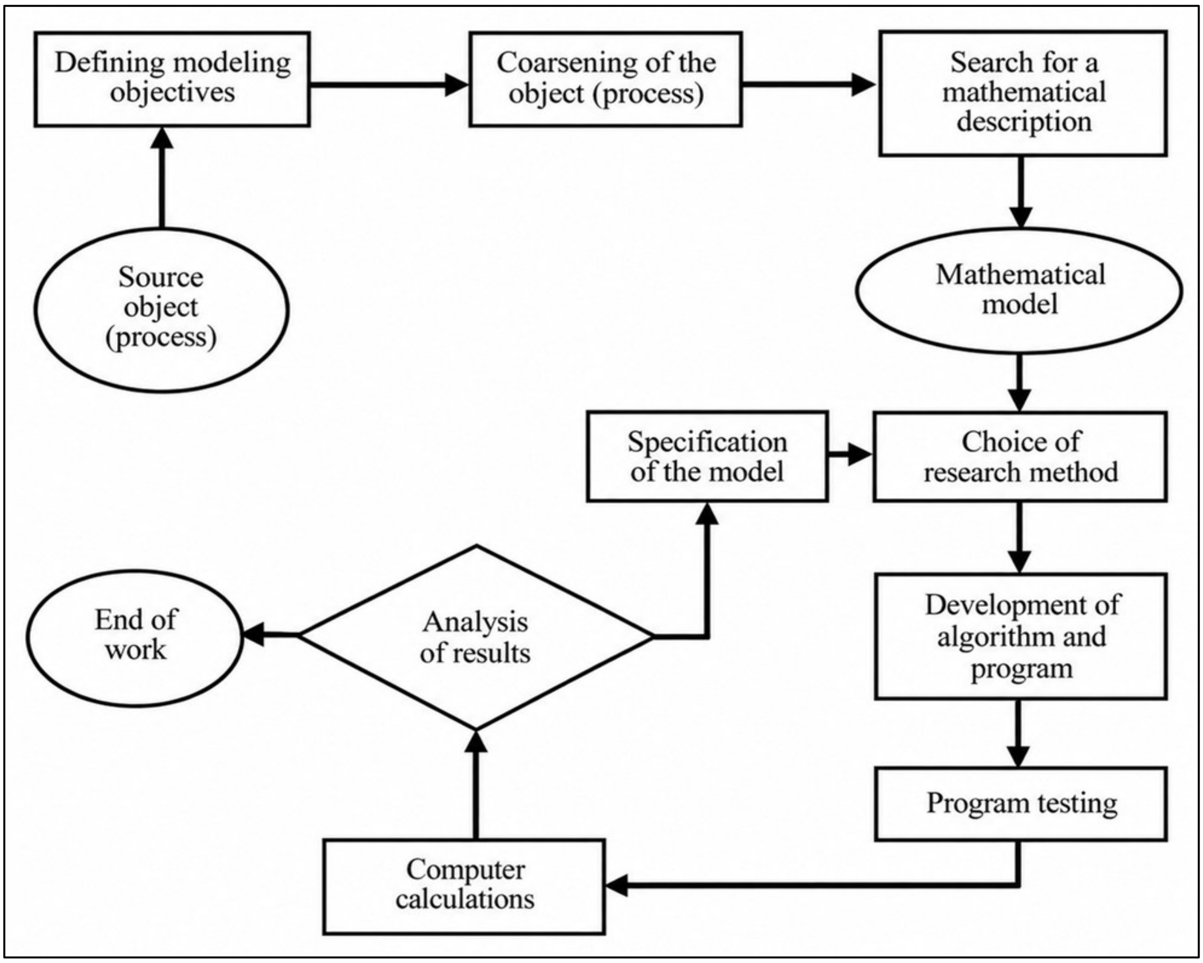

Figure 2. Computational framework for mathematical modeling and simulation [21]

Figure 2 shows an extensive computational procedure of mathematical modeling that starts from defining modeling objectives and proceeds to model formulation, algorithm development, and program coding. This procedure involves testing, numerical computations, and result analysis with subsequent model improvement. It should be noted that this model demonstrates the importance of combining programming and computation approaches in implementing theoretical models as computer programs.

Python is a commonly used programming language for mathematical modeling as it is open source, user-friendly, and has various available libraries. The available libraries like NumPy, SciPy, and Matplotlib facilitate numerical calculations and simulations and also data visualization. In addition, Python has the capacity to be used in connection with advanced technologies like machine learning and data analytics. This makes it a very versatile language that can even be applied in developing custom-made models and applications.

This review presents an overview on the role of Python programming in Mathematical Modeling and Computing Applications. Comparison of different computing tools is made according to accuracy, computational efficiency, ease of use and flexibility. Further, the application of the programming languages has been discussed in various fields along with the strengths and weaknesses. The purpose of this paper is to discuss the applicability of Python in solving engineering/scientific problems.

2. Literature Review

Mathematical modeling is widely used for the representation of various systems of the real world in the engineering and science fields. Mathematical modelling allows to convert real-world phenomena into mathematical formulas so that computational systems could process these formulas [1], [8] to produce results that would be similar to those of the real life. Various industries employ mathematical models as well as its applications but in general, the use of mathematical models can be observed in the fields of manufacturing systems, biological systems, and energy systems, where mathematical models depict the development of these systems and optimize the output [10], [16]. According to recent studies, mathematical modelling can also be applied to solving large scale issues like the prediction of the results of pandemics and healthcare analysis [20], [17].

The programming languages are used in mathematical modeling have a significant impact on the effectiveness of the mathematical model. The programming language used a certain level of computational efficiency and abstraction. MATLAB programming language is considered to be one of the most popular programming languages for numerical computations due to its many built-in toolboxes which make it easy to create control systems [2]. However, Python is gradually emerging as the most preferred programming language since it allows flexibility in programming. Python programming language has many libraries that facilitate scientific computing and data analysis [1], [9]. The simpler programming languages will allow higher levels of computational efficiency but require programming skills.

Based on recent studies comparing programming languages, high-level programming languages provide a better access to allow faster prototyping using rapid prototyping methods. Nonetheless, when creating larger simulations that need performance optimizations, low-level programming languages are more efficiently used [1]. Hence, programming language plays a crucial role in developing new models.

Python is widely applied in mathematical modeling because of its versatility, openness, and variety of libraries (NumPy, SciPy, and Matplotlib), which contain lots of numerical computations, modeling, simulation, and graphical visualization routines [6], [9]. Additionally, they enable the application to easily build mathematical models (differential equations, optimization, and statistics). Moreover, Python is widely applied in computational physics and engineering applications, where it is often used for performing engineering simulations or helping solve engineering problems; by building modular/reusable code structures, Python increases the likelihood of finding a solution to an engineering problem [7], [12].

Various domain-specific frameworks have greatly enhanced the modeling abilities of Python. PyBaMM is an open-source framework that allows users to develop batteries using electrochemical models in Python. Similarly, PyMC allows users to use Bayesian statistical models to perform probabilistic programming and quantify uncertainty, making it a great framework for data-driven mathematical modeling applications [11]. In summary, the above frameworks show how Python can support both deterministic and probabilistic modeling approaches.

Python-based mathematical models have been successfully implemented in various fields like engineering, data analytics, and health care. Some examples of implementation of Python in engineering include thermal studies, compensation of errors in machine tools, and simulations for renewable energy resources [14], [15]. From the perspective of data analytics, Python also includes the possibility to develop machine learning models for consumer behavior prediction and analysis of huge datasets [13].

Python has been extensively used by scientists in the creation of mathematical models for disease spread and interventions such as COVID-19, where Python helps health care experts and epidemiologists' model to know how diseases may potentially spread and how interventions may be effectively applied [3], [19].

There are many benefits for mathematical modeling using Python. Also, there are certain limitations too. The major limitations of using Python for mathematical modeling is the lower execution speed of Python in contrast to compiled languages like C++. This problem may cause problems in carrying out calculations which require more computing power. Another drawback is that more effort is required to carry out calculations with external libraries written specially for this purpose [4].

3. Python for Mathematical Modeling

Python is a very effective tool for mathematical modeling and scientific computations because of its flexibility, simplicity, and abundance of libraries. Unlike other languages where it requires some special syntax and previous skills to write the program, Python is easy to learn since it has a simple syntax that allows users to make use of the mathematical knowledge during programming. Open-source language like Python does not require extra expenses to access some special tasks because people can freely create, change and use them without having to purchase a license [6], [9].

Major advantages of using Python in mathematical modeling is the existence of various libraries for numerical computations, science, and visualization. In particular, there are libraries like NumPy that offer effective array operations and mathematical calculations as the foundation for mathematical modeling. In addition, features implemented in the SciPy library consist of a set of scientific algorithms that can be utilized for optimization, integration, interpolation, and solving differential equations. Finally, capabilities of the Matplotlib library allow for generating high-quality plots for the visualization of simulation results. It is important to note that there is also the panda's library for handling structured data [6], [9], [17].

Python supports many mathematical modeling approaches that are implemented with the use of the Python programming language. The methods involve the utilization of probabilistic and deterministic modeling approaches. With deterministic models, the most common application areas are to derive the solutions of ordinary and partial differential equations, and numerical simulation and optimization approach. These models are applied in many engineering applications in terms of thermal analysis, fluid dynamics and structural modeling, where there is a need to model the system behavior accurately. With the probabilistic modeling approaches, statistical analysis is carried out with the use of Monte Carlo and MCMC method by the implementation of PyMC framework [11], [7].

Another significant advantage of Python is its capabilities to integrate with recent technologies like machine learning and artificial intelligence. By combining the mathematical concepts with the currently used machine learning systems, Python enables the development of so-called hybrid models. These models represent the synergy of mathematical knowledge and data generated by insights. Hybrid models are more suitable for companies that require predictive, optimizing and real-time decisions. By using machine-learning mathematical models, companies can conduct research on consumer behavior, improve processes and make forecasts about markets and customers [13].

The modularity of Python's programming allows the development of scalable and reusable models that are necessary for solving complex computational problems. Domain specific libraries like PyBaMM illustrate how effective Python is for solving problems that are domain specific, such as battery modelling and energy systems simulations. Such libraries offer pre-built models and other sources that help with coding without sacrificing accuracy [5].

Although there are many advantages in using Python for mathematical modeling, there are some drawbacks of Python as well. The major disadvantage is the low speed of calculation in comparison to compiled programming languages like C++. Nevertheless, this drawback can be overcome by implementing optimized libraries, just-in-time compiling, and other high-performance computing approaches. In general, Python offers an excellent balance between flexibility, functionality, and usability, thus becoming a powerful tool for mathematical modeling in various fields [1], [4]. It can be concluded that Python is a highly flexible platform, allowing for performing mathematical modeling as well as modern computational methods.

Table 1. Major Python libraries for mathematical modeling

Library	**Primary Function**	**Mathematical Modeling Applications**	**Reference**
NumPy	Numerical computing and multidimensional arrays	Linear algebra, matrix operations, numerical computation	[6], [9]
SciPy	Scientific computing algorithms	Optimization, integration, differential equations, interpolation	[6], [7]

Pandas	Structured data processing	Data preprocessing and statistical analysis	[9], [13]
Matplotlib	Data visualization	Scientific graphs, simulation result visualization	[6], [17]
PyMC	Probabilistic programming	Bayesian inference and uncertainty quantification	[11]
PyBaMM	Battery modeling framework	Electrochemical battery and energy system simulation	[5]
TensorFlow	Deep learning framework	Hybrid mathematical and machine learning models	[13]
PyTorch	Deep learning framework	Scientific computing and predictive modeling	[13]

Table 1 highlights the major Python libraries used for mathematical modeling. The combination of these libraries offers complete coverage of numerical computation, optimization, statistics, probability modeling, scientific visualization, and specialized simulations. The modular structure of these libraries facilitates the creation of scalable computational pipelines by combining mathematical models and machine learning [5], [6], [7], [9], [11], [13], [17].

4. Comparative Analysis

Analysis of the programming languages that are applied in mathematical modeling is very important in determining the effectiveness of these models. The common programming languages used in mathematical modeling include Python, MATLAB, and COMSOL. All these programming languages have their own advantages and disadvantages based on the application domain [1], [2].

Python is widely known for its flexibility and open-source nature; therefore, the creation of mathematical models designed by one particular person without any license is possible. Libraries, which exist in Python, make it possible to apply the programming language to many areas, including the numerical and data modeling. Nevertheless, the slow processing speed of Python in comparison to the compiled languages may be a drawback of it [1]. At the same time, the MATLAB environment may be regarded as a programming language with pre-installed toolboxes for numerical calculations, signal processing and control systems. Well-developed algorithms provide fast development and the reliability of MATLAB in engineering tasks. But MATLAB is the proprietary software due to expensive licensing costs [2].

COMSOL Multiphysics is a simulation tool used in carrying out multiphysics simulation tasks. COMSOL Multiphysics contains a graphical user interface that makes it easy to make a model and include different physics such as thermodynamics, mechanics, and fluid dynamics. However, it is very appropriate for large-scale industrial applications; nevertheless, it is less flexible in terms of customizations when compared with Python. To provide a clearer comparison, the table below shows some of the important features that differentiate Python from MATLAB and COMSOL [1], [2], [6], [7], [8], [17], [18].

Table 2. Detailed comparison of programming tools for mathematical modeling

Criterion	Python	MATLAB	COMSOL
Numerical Computation Approach	Array-based (NumPy), flexible scripting	Matrix-based optimized environment	Finite element method (FEM)-based
Model Development Style	Code-driven, highly customizable	Semi-code with built-in functions	GUI-driven model construction
Support for Differential Equations	SciPy (ODE/PDE solvers), extensible	Built-in solvers (ODE45, PDE toolbox)	Advanced Multiphysics PDE solvers
Integration with Data-Driven Models	Strong (ML, AI libraries like TensorFlow, PyTorch)	Limited ML integration	Minimal support for ML
Visualization Capabilities	Matplotlib, Seaborn (customizable)	Built-in plotting tools	Advanced 3D simulation visualization
Performance Optimization	Requires external tools (Numba, Cython)	Optimized internally	Highly optimized for physics simulations
Domain-Specific Extensions	Extensive (PyBaMM, PyMC, etc.)	Toolboxes (control, signal processing)	Physics-specific modules
Parallel & High-Performance Computing	Supported via libraries (Dask, MPI, GPU)	Parallel toolbox available	Built-in HPC for simulations
Industrial vs Research Suitability	Strong in research & prototyping	Strong in academia & engineering	Strong in industrial simulations

Table 2, Python is more flexible and integrated with data driven modeling techniques, while MATLAB provides structured numerics, and COMSOL specializes in physics modeling through finite elements. This brings to mind the importance of choosing the correct tools depending on the objectives of the modeling process.

The analysis indicates that Python is very suitable for dynamic modeling purposes and research where flexibility and customization capabilities are needed. MATLAB should be used for engineering purposes since it comes with different built-in toolboxes and easy usability features while COMSOL is better for Multiphysics modeling with a high degree of accuracy. As a result, there is a need to balance between different parameters like speed, cost, and flexibility.

Selecting an appropriate programming tool will depend on the particularities of the modeling process. The Python language can boast high flexibility and extendibility. therefore, it may serve well in research-related and data-focused problems. At the same time, MATLAB features a well-organized programming environment with pre-installed toolboxes and is good for engineering projects, while COMSOL is the right tool for multiphysics modeling tasks that demand high accuracy.

5. Applications of Python in Modeling

The versatile nature of Python and the vast number of libraries in existence, Python has found extensive use in modeling physical phenomena mathematically. In the field of engineering, there is extensive use of Python to create mathematical models of physical phenomena like thermal conduction, fluid dynamics, and structure analysis. There are various Python packages available that help in carrying out numerical calculations and solving ordinary differential equations that constitute the foundation of most mathematical models of physical phenomena. This can be seen in the application of thermal error compensation during machining using mathematical models created in Python and research in renewable energy systems [14], [15].

In health care and epidemiology, Python programming language has found its use in modeling diseases transmission Along with intervention strategies. In mathematical modeling that incorporates the features of Python programming language, it has been possible to predict outbreak scenarios of pandemic disease like the one witnessed in COVID-19 by making use of different models and simulations. Deterministic and stochastic approaches are among the two that can be used in modeling processes in the Python language due to the uncertainty that might be involved with the collected data.

Table 3. Application areas of Python in mathematical modeling

Domain	Application	Python Tools Used	Reference
Engineering	Thermal analysis, fluid simulation	NumPy, SciPy	[14], [7]
Healthcare	Disease spread modeling (COVID-19)	SciPy, PyMC	[3], 16], [19]
Data Science	Predictive modeling, analytics	Pandas, ensorFlow	[13]
Energy Systems	Battery and renewable modeling	PyBaMM	[5], [15]
Scientific Research	Physics simulations	SciPy, Matplotlib	[7], [17]

The Table 3 lists the different fields of Python usage in mathematical modeling based on the reviewed literature. The research clearly show that Python can be used for numerical simulation, probabilistic modeling, machine learning, and specialized computational approaches in the fields of engineering, medicine, energy, and science [3], [5], [7], [13]-[17], [19].

Python can also be used in the field of data science and model construction using machine learning, in this case the mathematical models can be used alongside data analysis. Using tools like TensorFlow, PyTorch, and Pandas, we can create models that will predict various situations, such as customers' behavior, financial forecasts, and process improvement, among others. One of the advantages of using Python in model creation is the use of both statistics and machine learning to make deductions from complex datasets [13].

The Python programming language is widely used in the fields of computational physics and scientific research. The language is extensively used for simulations and numeric experiments and also for testing theoretical models. With the help of the language framework, it becomes simple for scientists to conduct simulations and replicate their findings. Besides, the language has also found use in niche applications like simulations in batteries and energy systems using the PyBaMM library.

Overall, the flexibility of Python makes it easy for it to be applied in many industries such as engineering, healthcare, and research related to science and data. The ability of Python to enable both traditional modeling and data-based innovations is the reason why it is important in addressing complex challenges.

6. Discussion

This research study demonstrates that Python is a very flexible and adaptive platform for performing mathematical modeling in various domains. It can develop custom models and, thus, is applicable for research purposes when the iterative process of improvement is required. The use of structured platforms such as MATLAB and COMSOL is simple but they impose restrictions on their usage in complex and developing cases of modeling. There is one more important aspect revealed in the analysis, namely, the possibility to implement the usage of Python in the context of data-driven approach. In contrast to other software tools, MATLAB allows implementing relatively poor integration with modern algorithms of machine learning and artificial intelligence; therefore, hybrid models combining mathematical modeling and data analysis can be developed. This factor plays a very significant role in the context of modern applications using large amounts of data as well as real-time data processing.

On a computational perspective, a trade-off exists between flexibility and performance. As a programming language which interprets code, there will be cases where the performance of Python during simulation will be inferior to those obtained from specialized software such as MATLAB and COMSOL. However, this problem may be addressed through code optimization, parallel computing, and use of high-performance computing systems. This means that Python can still be employed in complex simulations using appropriate code optimization strategies.

In summary, the results indicate that while selecting tools for mathematical modeling, it should consider the particularities of each problem. Python programming language is necessary in problems where flexibility, cross-discipline approach, and ability to analyze data is needed, whereas MATLAB should be used in numerical analysis cases, while COMSOL is cited as the best option in multiphysics contexts. Therefore, there is no universal tool and the key point here is to find a balance between various parameters.

7. Challenges and Limitations

Although it is a very flexible language with broad applications, there are certain issues associated with the use of Python for computational and mathematical modeling purposes. The main disadvantage is the relatively low speed of execution compared to C++, among others. Therefore, it is rather hard for Python to handle with complex computations and massive amounts of data, especially because of performance requirements. However, there are ways to address the aforementioned concern but it requires some extra optimization [1]. Packages such as NumPy, SciPy, and PyMC have lots of features but the use of many packages can increase the level of complexity and possible incompatibilities. The difference from other computing environments like MATLAB or COMSOL is that they provide the user with an integrated environment where everything is already installed, while in the case of Python this is not the case [2].

Initial Python integration requires some additional effort in learning for people without prior experience in programming concepts, especially advanced modeling techniques, parallel computing, and optimization. Despite Python being one of the easiest languages to learn among all others, developing good mathematical models requires knowledge of programming and mathematics. This requirement can be seen as a challenge for those coming from non-technical background. Furthermore, while Python has the capability to perform parallel and high-performance computations, to maximize performance, more efforts are required. The use of third-party packages in distribution processing and hardware acceleration creates some difficulties. Nevertheless, specialized simulation tools such as COMSOL make solving hard multiphysics problems significantly easier with minimal knowledge of programming.

8. Conclusion and Future Scope

This research study revealed a detailed analysis of Python in relation to mathematical modeling in different contexts. According to the result of this review, Python is considered as a versatile tool for creating mathematical models due to the various libraries that are used, such as NumPy, SciPy, and Matplotlib. However, it is important to mention that Python differs from the other software solutions of MATLAB and COMSOL in terms of research-related mathematical modeling. Apart from the applicative evaluation of the software in relation to Python, the software proves to be effective in many disciplines, such as engineering, healthcare, data sciences, and scientific analysis. It is important to note the ability of the software to provide a solid basis for the deterministic and probabilistic modeling process. Although there are certain challenges related to the efficiency of the software solution, Python can be viewed as a promising option for modern computation modeling. In the future, the use of Python in emerging technologies such as artificial intelligence and digital twin technology, among others, will enhance its effectiveness.

References

[1] ‘G, G‘Aniyev Temur Komiljon. "The Role of Programming Languages in Mathematical Modeling." Eurasian Journal of Social Sciences, Philosophy and Culture, 2024, vol 4, no. 9: 129-135.

[2] Idoko, Idoko Peter, Gerald Chekwube Ezeamii, Christian Idogho, Enemali Peter, U. S. Obot, and V. A. Iguoba. "Mathematical Modeling and Simulations using Software Like MATLAB, COMSOL and Python." Magna Scientia Advanced Research and Reviews, 2024, vol 12: 62-95.

[3] Almosa, Nadia Ali Abbas, and Ahmed Sabah Al-Jilawi. "Python Optimization Code for Solving a Mathematical Modeling of COVID_19." In AIP Conference Proceedings, 2023, vol. 2591, no. 1: 050026.

[4] Aiyenitaju, Olaniyi, and Sarita Majhi. "Math with Python: A Context of Algebra." Bachelor’s thesis, 2024.

[5] Sulzer, Valentin, Scott G. Marquis, Robert Timms, Martin Robinson, and S. Jon Chapman. "Python Battery Mathematical Modelling (PyBaMM)." Journal of Open Research Software 2021, vol 9, no. 1: 1-8.

[6] Linge, Svein, and Hans Petter Langtangen. Programming for Computations-Python: A Gentle Introduction to Numerical Simulations with Python 3.6. Springer Nature, 2020: Second Edition.

[7] Landau, Rubin H., Manuel J. Páez, and Cristian C. Bordeianu. Computational Physics: Problem Solving with Python. John Wiley & Sons, 2024: Second Edition.

[8] Velten, Kai, Dominik M. Schmidt, and Katrin Kahlen. Mathematical Modeling and Simulation: Introduction for Scientists and Engineers. John Wiley & Sons, 2024: Second Edition.

[9] Nagpal, Abhinav, and Goldie Gabrani. "Python for Data Analytics, Scientific and Technical Applications." In 2019 Amity International Conference on Artificial Intelligence (AICAI), IEEE, 2019: 140-145.

[10] Ilyushin, Yu V., and V. A. Nosova. "Development of Mathematical Model for Forecasting the Production Rate." Journal: International Journal of Engineering, 2025, vol. 38, no. 8: 1749-1757.

[11] Abril-Pla, Oriol, Virgile Andreani, Colin Carroll, Larry Dong, Christopher J. Fonnesbeck, Maxim Kochurov, Ravin Kumar et al. "PyMC: A Modern, and Comprehensive Probabilistic Programming Framework in Python." PeerJ Computer Science, 2023, vol 9: e1516.

[12] Avanzini, Stefano, David M. Kurtz, Jacob J. Chabon, Everett J. Moding, Sharon Seiko Hori, Sanjiv Sam Gambhir, Ash A. Alizadeh, Maximilian Diehn, and Johannes G. Reiter. "A Mathematical Model of CTDNA Shedding Predicts Tumor Detection Size." Science Advances 2020, vol 6, no. 50: eabc4308.

[13] Chaudhary, Kiran, Mansaf Alam, Mabrook S. Al-Rakhami, and Abdu Gumaei. "Machine Learning-based Mathematical Modelling for Prediction of Social Media Consumer Behavior using Big Data Analytics." Journal of Big data, 2021, vol 8, no. 1: 73.

[14] Mareš, Martin, Otakar Horejš, and Lukáš Havlík. "Thermal Error Compensation of a 5-Axis Machine Tool using Indigenous Temperature Sensors and CNC Integrated Python Code Validated with a Machined Test Piece." Precision Engineering, 2020, vol 66: 21-30.

[15] Hofmann, Fabian, Johannes Hampp, Fabian Neumann, Tom Brown, and Jonas Hörsch. "Atlite: A Lightweight Python Package for Calculating Renewable Power Potentials and Time Series." Journal of Open Source Software 2021, vol 6, no. 62: 3294.

[16] Peter, Olumuyiwa James, Hasan S. Panigoro, Afeez Abidemi, Mayowa M. Ojo, and Festus Abiodun Oguntolu. "Mathematical Model of COVID-19 Pandemic with Double Dose Vaccination." Acta Biotheoretica, 2023, vol 71, no. 2: 9.

[17] Sial, Ali Hassan, Syed Yahya Shah Rashdi, and Abdul Hafeez Khan. "Comparative Analysis of Data Visualization Libraries Matplotlib and Seaborn in Python." International Journal, 2021, vol 10, no. 1: 277-281.

[18] Bodner, Korryn, Chris Brimacombe, Emily S. Chenery, Ariel Greiner, Anne M. McLeod, Stephanie R. Penk, and Juan S. Vargas Soto. "Ten Simple Rules for Tackling your First Mathematical Models: A Guide for Graduate Students by Graduate Students." PLoS Computational Biology, 2021, vol 17, no. 1: e1008539.

[19] Oliveira, Juliane F., Daniel CP Jorge, Rafael V. Veiga, Moreno S. Rodrigues, Matheus F. Torquato, Nivea B. da Silva, Rosemeire L. Fiaccone et al. "Mathematical Modeling of COVID-19 In 14.8 Million Individuals in Bahia, Brazil." Nature Communications 2021, vol. 12, no. 1: 333.

[20] Raihan, Nishat, Mohammed Latif Siddiq, Joanna CS Santos, and Marcos Zampieri. "Large Language Models in Computer Science Education: A Systematic Literature Review." In Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 1, 2025: 938-944.

[21] Zhuzbayev, S., D. Sabitova, and B. Sarsenov. "Computer Mathematical Modeling of Wave Processes." 4th International Conference on Computer and Technology Applications (ICCTA), IEEE, 2018, 1-5.