JCRINN Journal of Computing Research and Innovation
e-ISSN : 2600-8793

Available online at
https://jcrinn.com/
https://crinn.conferencehunter.com/

Journal of Computing Research and Innovation 10(1) 2025

Journal of Computing Research and Innovation

Nonlinear Equation Solver with MATLAB Graphical User Interface

Nur Idalisa Norddin[1*], Nurul Hafawati Fadhilah[2], Muhammad Azri Azman Shah[3], Nurul Hajar Mohd Yusoff[4], Nur Intan Syafinaz Ahmad[5]

[1,2]*Universiti Teknologi MARA Cawangan Terengganu Kampus Dungun, Terengganu, Malaysia*
[3]*Universiti Teknologi MARA Cawangan Terengganu Kampus Kuala Terengganu, Terengganu, Malaysia*
[4]*Faculty of Industrial & Manufacturing Technology & Engineering (FTKIP), Universiti Teknikal Malaysia Melaka (UTeM), Melaka, Malaysia*
[5]*Universiti Teknologi MARA Cawangan Johor Kampus Pasir Gudang, Johor, Malaysia*

ARTICLE INFO

Article history:
Received 24 October 2024
Revised 29 December 2024
Accepted 10 January 2025
Online first
Published 1 March 2025

Keywords:
MATLAB GUI
Nonlinear Function
System of Nonlinear Equations
Iterative Algorithms

DOI:
10.24191/jcrinn.v10i1.505

ABSTRACT

Nonlinear systems of equations, which is common in scientific and engineering fields, pose a significant difficulty because they may have numerous solutions and are complicated to be numerically solved. This study aims to improve the effectiveness and availability of solving these systems by creating a MATLAB Graphical User Interface (GUI) to compare different iterative methods. The GUI makes it easier to evaluate algorithm performance by providing a more accessible interface to input nonlinear functions, their Jacobians, and also their respective initial approximations. The article evaluates the differences and similarities of four commonly used iterative methods: Newton, Broyden, BFGS, and Steepest Descent. A set of ten typical nonlinear test functions were created specifically as a benchmark suite for assessing methods in different situations. Thorough testing and analysis helped determine the advantages and disadvantages of each approach, offering insights into their suitability for different problem characteristics. Researchers and practitioners who have tried to analyse and contrast the reliabilities of iterative methods in solving nonlinear systems may discover the MATLAB GUI to be beneficial. By utilizing a visual and interactive interface, the GUI allows users to gain deep understanding.

1. INTRODUCTION

This study concentrated on constructing a graphical user interface (GUI) for iterative numerical methods as solutions for the systems of nonlinear equations. As compared to numerous others, Newton, Broyden, BFGS, and Steepest Descent (SD) methods are the most widely used and effective current approaches. Newton's method, famous for its rapid convergence specifically when the initial guess is approximately near to real solution, involves the use of derivative (Jacobian) to refine guesses through iterations. However, relying on accurate Jacobian calculations could present a major disadvantage, especially for complex or problematic functions. Alternatively, if obtaining derivative information is challenging, Broyden's method,

[1*] Corresponding author. *E-mail address*: nuridalisa@uitm.edu.my
https://doi.org/10.24191/jcrinn.v10i1.505

a variation of Newton's method, offers a quasi-Newton approach that circumvents the need for directly computing the Jacobian through iterative updates. The BFGS method enhances this approach by maintaining the positive definiteness of the approximation, like another quasi-Newton method.

The GUI eases the use of iterative numerical methods, hence broadening their application in different fields. Users can enter necessary variables and get detailed results, including the number of iterations required for convergence and performance metrics, making the process to be more efficient. By filling a significant void in the application of these techniques, the GUI lessens the requirement for specialized programming expertise.

2. LITERATURE REVIEW

Many research studies have concentrated on the effectiveness of teaching and learning mathematics. In 2018, Kulina et al. reviewed the benefits of software in enhancing and boosting mathematical abilities. They emphasized the growth of various mathematical-related abilities, including problem-solving, analytical thinking, and computational skills. Collaborative and practical activities have been recognized as beneficial to increase student involvement, enhancing mathematical comprehensiveness, and developing practical abilities.

Using visual and interactive representations of numerical algorithms in educational tools has been proven to improve student engagement and comprehension, as demonstrated by Li (2021). Computer Algebra Systems (CAS) like Maple, Mathematica, and MATLAB, are becoming more common in tertiary level of mathematical learnings (D. Jarvis et al., 2018). CAS integrates abilities for symbolic, numeric, and graphic calculations (Buteau et al., 2014). The selection of a system is frequently based on the particular requirements and users' option (Flores Salazar et al., 2022). According to D. Jarvis et al. (2022), CAS has been shown to improve teaching and learning by allowing students to delve deeper into mathematical concepts, creating a more engaging learning atmosphere, and offering instant feedback and personalized learning opportunities.

D. H. Jarvis et al. (2012) conducted a research project on the utilization and long-term viability of CAS of mathematics teaching in tertiary levels, exposing its global usage for improving instruction and offering sophisticated computational functions. Many educators are committed in using CAS despite the obstacles like software expenses, the requirements of continued professional learning, and challenges in aligning it with the curriculum. In a national survey of Canadian mathematicians, Buteau et al. (2014) also discovered that CAS is widely utilized in both research and teaching, with its role in research strongly impacting its usage in instruction. D. Jarvis et al. (2018) showed in a case study in Alberta, Canada, that CAS technology promoted interactive and dynamic learning environments, aided differentiated instruction, and improved problem-solving skills, yet challenges such as instructor preparation and equal access persisted.

In a research project carried out by Mezhennaya and Pugachev (2019), engineering students' views on three mathematical software programs - MATLAB, Mathematica, and Excel - were analyzed and compared. The research found that every scientific tool examined could be applied in educational settings, provided that software usage policies were implemented with caution. Nevertheless, the research indicated that a large number of students had limited practical experience with utilizing the software, specifically MATLAB and Mathematica, due to their perceived lack of user-friendliness. Therefore, the research determined that more courses should be offered to help students get ready to use software in their classes.

Another research by Khedekar et al. (2024) investigated how MATLAB improved the quality of education by enriching the teaching and learning process. MATLAB greatly enhanced students' learning experience by offering robust tools for computation, visualization, and simulation. Students discovered that using MATLAB was beneficial for comprehending difficult ideas and tackling mathematical issues. Nevertheless, the research also pointed out some obstacles in the utilization of MATLAB, including the

https://doi.org/10.24191/jcrinn.v10i1.505

necessity of sufficient training for teachers and students and potential problems with software access and compatibility. It was discovered that addressing these challenges is essential for a successful implementation.

Darweesh (2021) conducted a study on the incorporation of image processing and fundamental GUI concepts in teaching MATLAB programming. The research suggested a teaching method for MATLAB to students that included hands-on activities in image processing and GUI design. This method sought to increase the appeal and ease of programming by integrating interactive components. The study highlighted the significance of incorporating practical uses and easy-to-use interfaces in teaching MATLAB to assist students in comprehending intricate programming ideas and improving their learning journey.

Students' main deal is the high expense of commercial math software, even after rebates for academia (Han et al., 2016). Recent studies have concentrated on creating graphical user interfaces to support students in comprehending and solving non-linear equations. These resources are designed to enhance comprehension and participation in numerical analysis classes, which are frequently perceived as difficult (Han et al., 2016). GUI have been developed for a range of objectives, including simulating chaotic systems (Silva et al., 2018), compared diverse root-finding techniques (Alloqmani et al., 2022), and solving sets of non-linear equations (Caligaris & Laugero, 2022). These interfaces are usually designed in a way that is easy for users to navigate, necessitating only a small amount of interaction from students in order to produce solutions and visual representations. Different platforms have been utilized for their implementation, such as Excel VBA (Han et al., 2016), standalone applications (Silva et al., 2018), MATLAB (Alloqmani et al., 2022), and customized math programs (Caligaris & Laugero, 2022). Feedback from users and surveys has indicated that these tools effectively aid in learning and understanding concepts in numerical analysis.

Caligaris and Laugero (2022) utilized Newton's Method in their study to solve non-linear equations in the GUI. Newton's Method, or Newton-Raphson method, is a repetitive numerical method used to find estimated solutions to sets of non-linear equations through linearizing the equations at every stage. The MATLAB-created graphical user interface (GUI) makes it easier to apply this method by enabling users to enter equations, modify parameters, and observe the iterative procedure.

Even with these improvements, the requirement for tools that combine various iterative techniques into one cohesive app still endure. Many current tools either concentrate on particular techniques or offer restricted features. The goal of the application created in this research is to address this deficiency by combining Newton, Broyden, BFGS, and SD method in a single independent platform. This integration does not just make learning easier but also provides a complete tool for students and teachers.

3. METHODOLOGY

The first section of methodology is the selection of an appropriate research focus on numerical techniques for solving nonlinear equation systems, including Newton, Broyden, BFGS, and SD methods, along with MATLAB's `fsolve` function for comparison. The `fsolve` function in MATLAB can be used to solve these equations without requiring complicated analytical solutions or manual derivation (Abu-Alshaikh & Sahin, 2006). This initial step determined the study's focus, making sure that the chosen techniques were suitable for thorough examination considering their effectiveness, especially in terms of the necessary NOI and the gradient norm. A thorough review of literature will be carried out next based on previous studies to guide the development of the testing framework, consisting of ten specific test functions to evaluate the methods' efficiency.

The subsequent stage included the identification and preparation of the required mathematical equations for applying the chosen techniques in solving non-linear systems. MATLAB R2022a was utilized to create a user-friendly GUI where users can input functions, Jacobians, initial values, and tolerance levels. They can then choose a numerical method for calculating solutions, which includes metrics such as the NOI and NOG. The code was thoroughly tested and refined to guarantee its accuracy and consistency in all

https://doi.org/10.24191/jcrinn.v10i1.505

methods. Successful executions will proceed to the final stage of data collection, comparison, and analysis, which emphasize efficiency and provide a detail evaluation and discussion, in a more extensive circumstance of existing findings.

Fig. 1 shows the flowchart of the proposed MATLAB GUI.

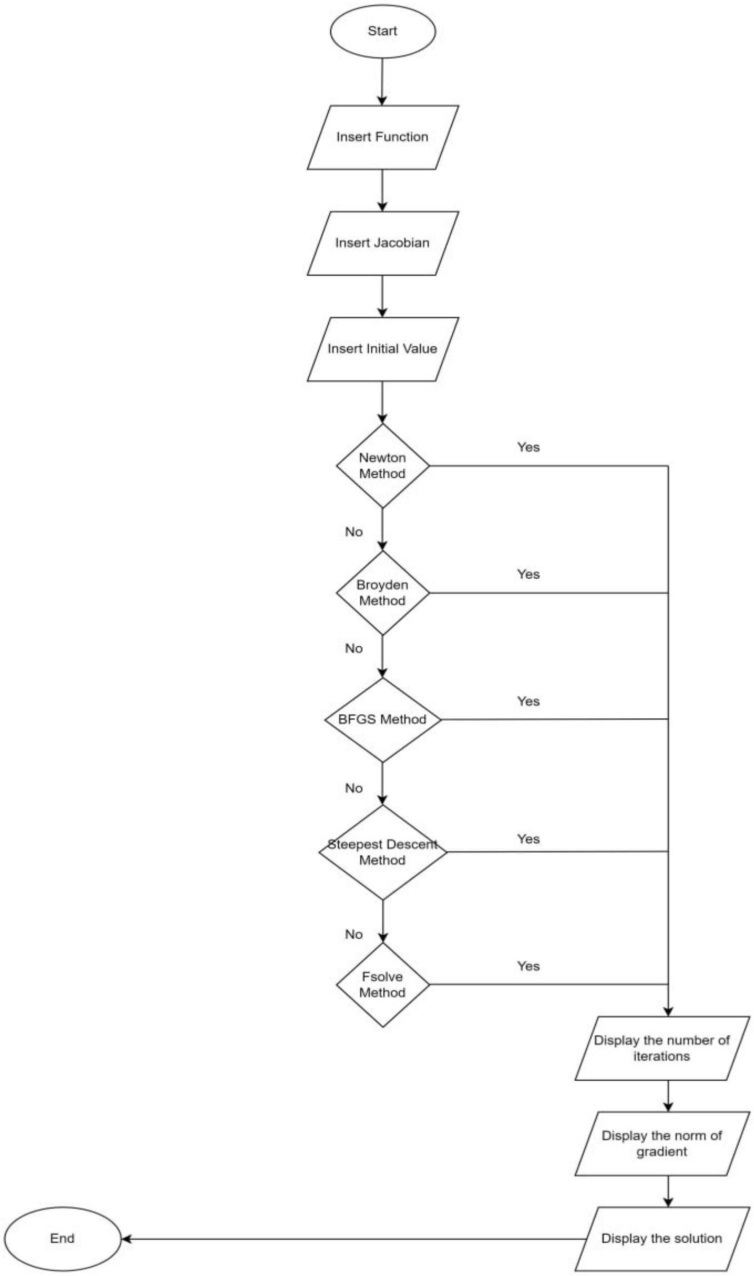

Fig. 1. Research procedure

https://doi.org/10.24191/jcrinn.v10i1.505

Fig. 2 shows the design of a solver created for tackling nonlinear systems of equations using MATLAB's App Designer in version R2022a. Next, users are asked to provide the essential information, such as function, its Jacobian (the function's derivative), and initial values. Once the inputs are given, users need to choose the preferred calculation method.

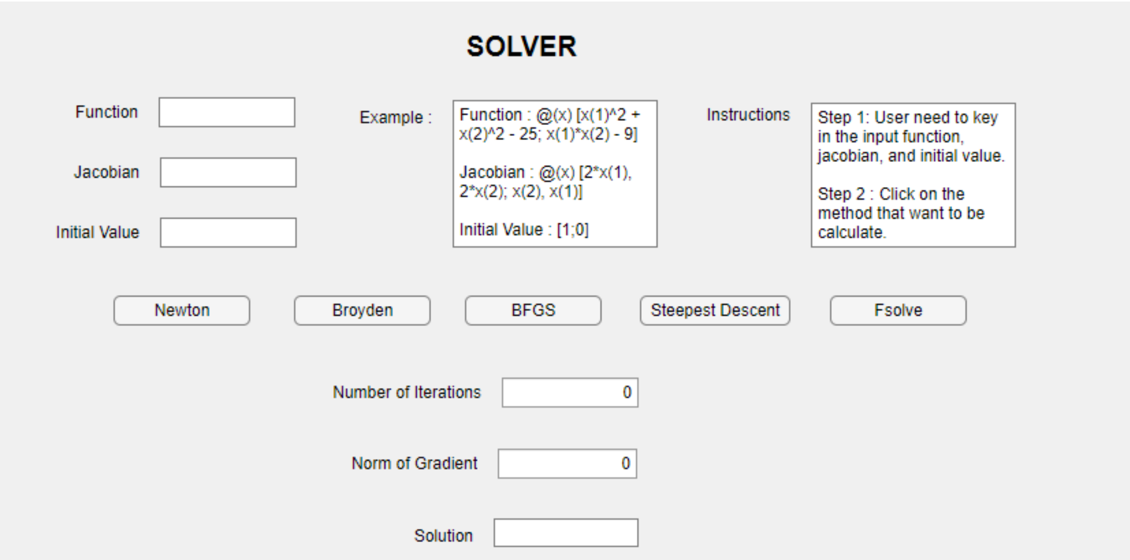

Fig. 2. GUI solver in MATLAB

Table 1 displays ten test functions, each paired with two distinct initial values. Ten typical nonlinear functions were chosen for testing, representing a variety of challenges and characteristics found in real-world scenarios.

Table 1. List of test functions

Function	Test Function	Initial Value
F1	$2x_1 + x_2 + 3x_3 - 10$ $x_1 + 5x_3 - 7$ $x_2 + x_3 - 4$	$(0,1,0)^t$, $(0,0,1)^t$
F2	$x_1^2 + x_2^2 - 25$ $x_1 x_2 - 9$	$(1,0)^t$, $(0,1)^t$
F3	$10x_1 - 2x_2^2 + x_2 - 2x_3 - 5$ $4x_2^2 + 4x_3^2 - 9$ $8x_1 x_3 + 4$	$(0.5,0.5,0.5)^t$, $(1,0,1)^t$

https://doi.org/10.24191/jcrinn.v10i1.505

F4	$3x_1 + 2x_2 + x_3 - 10$ $5x_1 + x_3 - 7$ $x_2 + x_3 - 4$	$(0,1,0)^t, (0,0,1)^t$
F5	$x_1 + x_3 x_2^2 - 10$ $3x_1^2 - x_2 + 2x_3 + 2$ $x_2^3 + x_3 - 7$	$(0,0,1)^t, (1,0,1)^t$
F6	$15x_1 + x_2^2 - 4x_3 - 13$ $x_1^2 - 10x_2 - x_3 - 11$ $x_2^3 - 20x_3 + 22$	$(0,0,1)^t, (1,0,1)^t$
F7	$x_1 + 3x_2 + 2x_3 - 10$ $x_1 + 5x_3 - 7$ $x_2 + x_3 - 4$	$(0,1,0)^t, (0,0,1)^t$
F8	$x_1(x_2 - 2)^2 + 4x_2 - 3x_3 - 10$ $-2x_1 + x_2^2 + 4x_3 + 1$ $x_1 + 6x_2 - x_2 x_3 - 15$	$(1,0,1)^t, (0,0,1)^t$
F9	$2x_1^2 + 2x_2^2 - 25$ $2x_1 2x_2 - 9$	$(0,1)^t, (1,0)^t$
F10	$2x_1 + x_2 + 3x_3 - 10$ $x_1 + 5x_3 - 7$ $x_2 + x_3 - 4$	$(0,0,1)^t, (1,0,1)^t$

4. RESULTS AND DISCUSSIONS

This part discusses the MATLAB GUI solver's testing stage and highlights the theoretical analysis. The testing stage aims to assess whether the goals have been met, as well as to evaluate the solver's functionality. To ensure all buttons and components in the GUI solver are working correctly, a sample equation and inputs are utilized for verification.

The tester equation of the GUI solver utilizes the nonlinear equations F1 from Table 1 with the initial value $(0,1,0)^t$. Once all the data has been entered into the solver, users must select the desired method to present the results. For example, Fig. 3 displays the GUI solver's output interface when the Newton button is clicked.

https://doi.org/10.24191/jcrinn.v10i1.505

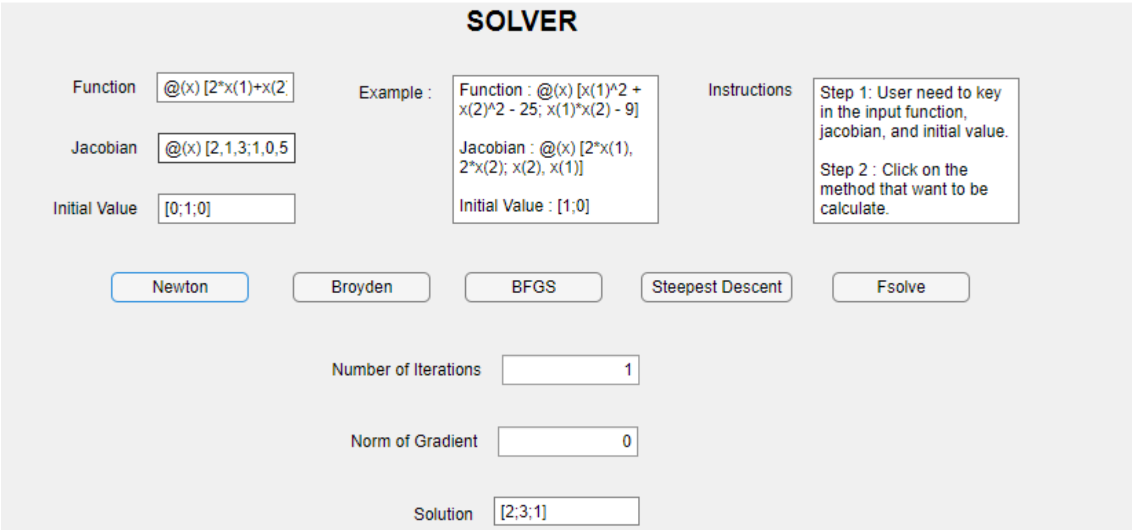

Fig. 3. Output interface of GUI solver

By utilizing this graphical user interface solver, quick responses of results such as iteration count, gradient norm, and solution point is possible. It is evident from Fig. 3 that all anticipated results are clearly visible, indicating the button is functioning correctly. If the inputs stay unchanged, clicking on the button for other methods (Broyden, BFGS, SD, `fsolve`) will produce their results as well. During the analysis stage, these five techniques were evaluated based on NOI and NOG using the performance profile method developed by Dolan and Moré (2002). The performance profile is a widely used method to assess the effectiveness and stability of techniques, while also confirming the reliability and stability of optimization solutions.

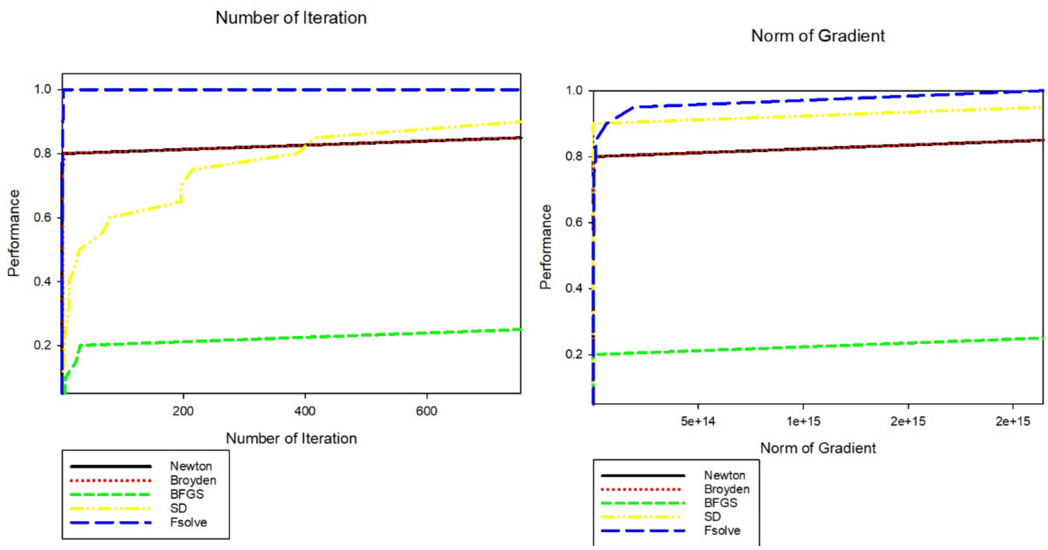

Fig. 4. (left) Performance profile of NOI; (right) Performance profile of NOI

https://doi.org/10.24191/jcrinn.v10i1.505

Fig. 4 (left) demonstrated that `fsolve` stands out as the most efficient approach, as the optimal solutions were rapidly attained after few iterations. Newton, Broyden, and SD methods showed similar performance levels. On the other hand, BFGS fells behind in terms of both speed and performance. Fig. 4 (right) displays that the `fsolve` method consistently outperformed other methods in terms of NOG, showing faster and superior performance through a quick initial increase and high stabilization level. After SD method, Newton, Broyden were used, with BFGS method showing the poorest performance. In summary, `fsolve` proved to be the most efficient in terms of iterations and the most successful in optimizing the norm of the gradient.

Overall, the MATLAB GUI solver's testing phase demonstrates its precision and efficiency in solving challenging nonlinear equations. The analysis has offered important insights into the effectiveness of the MATLAB GUI solver, thus confirming its trustworthiness for future research.

5. CONCLUSION AND RECOMMENDATIONS

Both students and instructors in academic environments encounter difficulties in comprehending and utilizing iterative methods efficiently. Manual calculations and conventional teaching methods are both time-consuming and prone to mistakes, consequently, could prevent the learning process and limit a full grasp of these methods (Liu, et. al, 2020).

Developing an application aimed at easing the process of solving systems of nonlinear equations using iterative methods is a significant step to tackle these difficulties. This application improves both the computational process and the learning experience by offering an interactive platform for both students and teachers. Through automating the repetitive computations and displaying the progress towards convergence, the software functions as a teaching and practical tool for teachers, closing the divide between theory and real-world use.

Lourenço et al. (2024) explored the revolutionary effect of artificial intelligence (AI) platforms on the field of mathematics education. They investigated the ways in which AI-powered tools improved the educational experience in college math classes. They found that AI platforms played a crucial role in delivering mathematics course lectures, providing chances for enhanced, streamlined, and enjoyable learning journeys.

6. ACKNOWLEDGEMENTS/FUNDING

The authors would like to acknowledge the support of Universiti Teknologi Mara (UiTM), Cawangan Negeri Terengganu, Malaysia for providing the facilities.

7. CONFLICT OF INTEREST STATEMENT

The authors agree that this research was conducted in the absence of any self-benefits, commercial or financial conflicts and declare the absence of conflicting interests with the funders.

8. AUTHORS' CONTRIBUTIONS

Nur Idalisa: Conceptualisation, supervision, and writing-original draft; **Nurul Hafawati Fadhilah**: Conceptualisation, writing- review and editing, and formal analysis; **Muhammad Azri Azman Shah**: Formal analysis, investigation and validation; **Nurul Hajar**: Conceptualisation, methodology, and validation. **Nur Intan Syafinaz**: Conceptualisation, writing- review and editing, and validation.

https://doi.org/10.24191/jcrinn.v10i1.505

9. REFERENCES

Abu-Alshaikh, I., & Sahin, A. (2006). Two-point iterative methods for solving nonlinear equations. *Applied mathematics and computation*, 182(1), 871–878. https://doi.org/10.1016/j.amc.2006.04.064

Alloqmani, A., Alsaedi, O., Bahatheg, N., Alnanih, R., & Elrefaei, L. A. (2022). Design principles-based interactive learning tool for solving nonlinear equations. *Comput. Syst. Sci. Eng.,* 40(3), 1023–1042. https://doi.org/10.32604/csse.2022.019704

Buteau, C., Jarvis, D. H., & Lavicza, Z. (2014). On the integration of computer algebra systems (CAS) by Canadian mathematicians: Results of a national survey. *Canadian Journal of Science, Mathematics and Technology Education*, 14, 35-57. https://doi.org/10.1080/14926156.2014.874614

Caligaris, M. G., & Laugero, L. F. (2022). Tools for solving systems of non-linear equations. *Global Journal of Computer Sciences: Theory and Research*, 12, 62–77. https://doi.org/10.18844/gjcs.v12i2.8037

Darweesh, M. (2021). Teaching MATLAB Programming by Utilizing Image Processing and Graphical User Interface Basics. In *Interactive Mobile Communication, Technologies and Learning* (pp. 824-832). Springer. https://doi.org/10.1007/978-3-030-96296-8_74

Dolan, E. D., & Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical programming*, 91, 201-213. https://doi.org/10.1007/s101070100263

Flores Salazar, J. V., Gaona, J., & Richard, P. R. (2022). Mathematical work in the digital age. Variety of tools and the role of geneses. In *Mathematical Work in Educational Context: The Perspective of the Theory of Mathematical Working Spaces* (pp. 165-209). Springer. https://doi.org/10.1007/978-3-030-90850-8_8

Han, C. T., Gaik, T. K., & Kahar, R. A. (2016). Nonlinear equation graphical user interface solver using excel VBA programming. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 8(2), 59–63.

Jarvis, D., Buteau, C., Doran, C., & Novoseltsev, A. (2018). Innovative cas technology use in university mathematics teaching and assessment: Findings from a case study in Alberta, Canada. *Journal of Computers in Mathematics and Science Teaching*, 37(4), 309–354.

Jarvis, D., Dreise, K., Buteau, C., LaForm-Csordas, S., Doran, C., & Novoseltsev, A. (2022). CAS use in university mathematics teaching and assessment: Applying oates' taxonomy for integrated technology. In *Mathematics education in the age of artificial intelligence: How artificial intelligence can serve mathematical human learning* (pp. 283–317). Springer. https://doi.org/10.1007/978-3-030-86909-0_13

Jarvis, D. H., Buteau, C., & Lavicza, Z. (2012). Computer algebra system (CAS) usage and sustainability in university mathematics instruction: Findings from an international study. *Paper presented as part of Topic Study Group*, 18.

Khedekar, M., Aher, S., Mandale, M., Bobalade, D., Patil, V., & Yadav, S. (2024). Enhancing the teaching-learning experience with the implementation of Matlab tool: A case study. *Educational Administration: Theory and Practice*, 30(5), 10891–10903.

Kulina, H. N., Gocheva-Ilieva, S. G., Voynikova, D. S., Atanasova, P. K., Iliev, A. I., & Ivanov, A. V. (2018). Integrating of competences in mathematics through software—case study. *2018 IEEE Global Engineering Education Conference (EDUCON)*, 1586–1590. https://doi.org/10.1109/EDUCON.2018.8363424

https://doi.org/10.24191/jcrinn.v10i1.505

Li, X. (2021). Application of computer and mathematics software in university mathematics teaching. *2021 2nd International Conference on Information Science and Education (ICISE-IE*), 1256–1259. https://doi.org/10.1109/ICISE-IE53922.2021.00281

Liu, X., Deng, M., Meng, Q., & Yan, Y. (2020). History and pedagogy mathematics in mathematics education: On the causes of learning difficulties in advanced mathematics. *2020 International Conference on Modern Education and Information Management (ICMEIM),* 730–733. https://doi.org/10.1109/ICMEIM51375.2020.00163

Lourenço, J., Milczarski, P., Lucas, C., Paulo, J., & Morais, J. (2024). Mathematics course lecturing: Ai platform as an enabler. *Future of Information and Communication Conference*, 23–29. https://doi.org/10.1007/978-3-031-53963-3_3

Mezhennaya, N. M. & Pugachev, O. V. (2019) On perception of computer algebra systems and Microsoft Excel by engineering students. *Problems on Education in the 21st Century*, 77(3), 379-395. https://doi.org/10.33225/pec/19.77.379

Silva, P. H. O., Nardo, L. G., Martins, S. A. M., Nepomuceno, E. G., & Perc, M. (2018). Graphical interface as a teaching aid for nonlinear dynamical systems. *European Journal of Physics*, 39(6), 065105. https://doi.org/10.1088/1361-6404/aae35c

About the Authors

Nur Idalisa, PhD, is a senior lecturer at the Pusat Pengajian Matematik, Universiti Teknologi MARA Cawangan Terengganu. Her primary research interest is in optimization, specifically conjugate gradient methods. Her interests also include machine learning and data analysis. She teaches Calculus, Engineering Mathematics, Introduction to Statistics, Numerical Analysis and Algebra. She may be reached by email at nuridalisa@uitm.edu.my.

Nurul Hafawati Fadhilah is a PhD candidate in Universiti Teknologi MARA. Her main research is in the area of optimization specifically in the conjugate gradient method. She can be reached through her email at nurulhafawatifadhilah@gmail.com

Muhammad Azri Azman Shah is currently a final-year student pursuing a Bachelor of Science (Hons.) in Mathematical Modelling and Analytics at UiTM Kuala Terengganu. He is undertaking an internship at Cakna Travel & Tours Sdn Bhd located in Kuala Kangsar, Perak, where his duties involve managing customer data within the company's database, conducting data analysis, summarizing customer insights, and preparing monthly reports. He can be contacted at azrimuhammad1620@gmail.com.

Nurul Hajar , PhD, is a lecturer at the Faculty of Industrial and Manufacturing Technology and Engineering at Universiti Teknikal Malaysia Melaka (UTeM). With a passion for making complex concepts accessible, she teaches a diverse range of mathematics courses, including business mathematics, statistics, engineering mathematics, and differential equations. With a strong focus on real-world problem-solving, her research revolves around optimization and applied mathematics. For collaborations or inquiries, she can be reached at nurulhajar@utem.edu.my.

https://doi.org/10.24191/jcrinn.v10i1.505

Nur Intan Syafinaz is a senior lecturer in the Department of Mathematics, College of Computing, Informatics, and Mathematics. Her primary areas of research include algebra, graph theory, and numerical analysis, reflecting her expertise and contributions to the field of mathematics. Dedicated to advancing knowledge in these areas, she actively engages in academic research and teaching. She can be contacted via email at nurin395@uitm.edu.my.

https://doi.org/10.24191/jcrinn.v10i1.505