Received: 25 November 2022 | Revised: 17 February 2023 | Accepted: 12 March 2023 | Published: 24 April 2023

Volume 1(1) Pages 1–18 https://doi.org/10.56279/jicts.v1i1.17

JICTS
Journal of ICT Systems

STACK for interactive online numerical analysis tutorials: development, competence and performance

Idrissa S. Amour

Department of Mathematics, University of Dar es Salaam, Dar es Salaam, Tanzania

Corresponding author
Email: idrissa.amour@gmail.com
amour.idrissa@udsm.ac.tz

Keywords
Mathematics Digital Assessment
Mathematics e-Learning
Numerical Analysis
Online tutorials
STACK
University of Dar es Salaam

Abstract

Overseeing online tutorials in mathematics requires competency in Computer Algebra System. In this work, we study the application of System for Teaching and Assessment using Computer Algebra Kernel, a Moodle mathematics plugin. Our aim was to oversee online tutorials for Numerical Analysis, an undergraduate second-year course taught in the 2019/2020 and 2020/2021 academic years at the University of Dar es Salaam. Varieties of randomized questions were competently developed to cover the content of the course. The questions test different learning objectives, exhausting direct and iterative computations, analytical and guided proofs. Students got immediate feedback on their solutions and anticipated mistakes. Students' participation and performance in the tutorials and quizzes are promising. Students' final scores do not show any negative impact from the use of online tutorials. Overseeing online tutorials is important as the number of students enrolled every year is increasing disproportionally with the available teaching staff. This can also be an appropriate tool for both interactively delivering mathematics content and assessing mathematics learning, when meetings between teachers and students are limited (e.g., during the pandemic period) or not necessary (e.g., distance learning programmes).

1. Introduction

The use of a technology for learning and assessment in mathematics started to get momentum back at the beginning of the 21st century [1]. Since then, a number of mathematics engines, including NUMBAS [2], WeBWorK and MyOpenMath [3], and System for Teaching and Assessment using a Computer Algebra Kernel (STACK) [4], have been developed. These engines can easily be embedded into the Learning Management System (LMS), and require to be powered by Computer Algebra System (CAS), even for simple mathematics questions [5]. STACK is a powerful system for the online assessment of mathematics and related subjects [6]. STACK is available as a question type for Moodle and ILIAS systems. Thus, it receives full advantage of Moodle quiz management features. It can be used in other systems with the help of the Learning Tools Interoperability protocol [3, 6]. STACK executions incorporate Maxima programming as an open source CAS [4, 6].

Several works have been carried out to assess mathematics learning digitally. Sangwin [7] reported the successful use of STACK in the University Linear Algebra examination, whose results moderately correlated with version of paper examination. Digital assessment can also address the issue of large classes. Zerva [8] used STACK at the University of Edinburgh between 2017-2019 for over 600 students class in mathematics courses, where the cost of staff time was significantly reduced at the expense of one-time capital investment of developing the questions. At the University of Helsinki, Matti and Ellist at al. [9, 10] reported the use of STACK for an online Calculus course, where students got automatic, instant, and relevant feedback. Apart from assessment, Kinnear [11, 12] demonstrated how STACK could be used to interactively deliver mathematics content. The work of Davies et al. [13] reported the application of STACK in mathematics from lecturers at the University College London. However, the instructors admit their limitations to procedural questions over conceptual questions. Researchers are still looking for better performance of STACK by extending its functionalities. Eichhorn and Helfrich-Schkarbanenko [14] implemented short answers test which examines if two strings match by using the Damerau-Levenshtein metric. STACK may be applied beyond Mathematics; it has also been demonstrated in mechanics and electronics [15, 16, 17]. Orthaber et al. [16] created and accessed complex engineering problems electronically. In the region, STACK has got little attention. The work of Juma et al. [18, 19] reported the use of STACK at the University of Maseno for Calculus and Complex Analysis courses. In both cases, the authors only reported the behavior and performance of students.

Despite many advantages uncovered by researchers, the use of digital assessment in mathematics has unfortunately been not well accepted and, in some cases, not recognised as appropriate. One possible reason could be that the use of technology in learning and assessments in mathematics is subjected to several constraints. For example, the possibility of grading an alternative solution of a mathematical problem rather than the one set by the instructor. How can we make sure students are learning at the same pace even if some have little or no background in information and communication technology (ICT) and use of sophisticated ICT devices? Evidently we will need to invest our time to train students in writing basic CAS commands. The worrying setback in adapting online assessments in mathematics is the difficulty of constructing competent questions instead of easily automatic gradable questions [3]; the former compromises both the quality of the teaching process and learning outputs. Simplifying authoring of competent questions, Nakamura and Nakahara [20] created an item bank of STACK questions, where users could freely share their valuable questions.

Numerical Analysis is a branch of Mathematics that covers numerical approximations of mathematical problem solutions, whose analytical solutions are either impossible or difficult to obtain. Numerical Analysis (MT 274), an undergraduate course taught at the University of Dar es Salaam (UDSM) covers the topics in two categories: computational and analytical parts. In the computational part, procedural solutions are based on direct or iterative evaluation of numeric variables, while in the analytical part, the topics cover the analysis and limitations of the numerical methods. Most courses at UDSM are offered with weekly exercises (tutorials), where students get problems to be solved and discussed in arranged sessions. These sessions need to be interactively manageable. However, with the increased number of enrolled students, the groups are usually larger than the required capacity, thereby limiting students' learning capabilities.

Naturally, problems in numerical methods require significant effort in computation. The computations require high accuracy of floating point values to avoid error explosion. This demand is rather difficult for humans to handle efficiently. This reason makes numerical analysis different from other reported areas in mathematics. Overseeing this kind of course in a classical fashion of tutorials setup is ineffective. This suggests that the use of computer-aided systems is inevitable unless we limit students to solving toy problems, which usually require little human effort. Using STACK without competent questions, which resemble traditional paper-pen questions, has already shown negative impact on students, where results of online quizzes differed with the final results [19]. In this paper, we study the application of STACK and LMS (Moodle) to oversee the MT 274 tutorials and quizzes digitally. The study shall develops paper-pen-like competent and interactive STACK questions, where the intermediate steps are examined, graded, and instant feedback provided. The developed questions should address the concern of digital assessments in mathematics. Does the formulation give students the opportunity to critically think? Behaviour and performance of students engaged with online tutorials and quizzes are studied as well. Because assessment is not only about assigning numbers to define the success of students but also to make them learn and progress [21], this study focuses on assessment for learning. This technological intervention started in 2019 when Tanzania opened long-closed Universities due to the first Covid-19 outbreak; the UDSM management advised instructors to minimize contact with students.

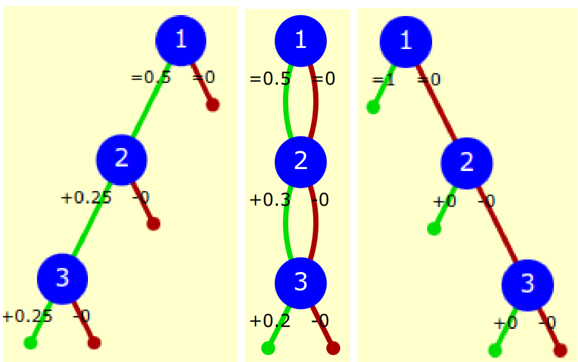
2. Methodology

2.1 Requisites of STACK

Developing competent STACK questions requires good skills in Maxima programming to handle both question-answer implementation and question variables randomization process. STACK uses MathJax to display mathematics symbols that are typeset with LaTeX. Basic LaTeX commands are necessary for typesetting mathematics expressions in STACK questions. To avoid the difficulty in solving the problem of copying hidden format characters, we choose the use of Moodle plain text editor with Hypertext Markup Language codes to manage both question and solution texts.

2.2 Basic Grading Algorithms

The real power of STACK comes from its design in Potential Response Tree (PRT) where the author can use it as a branching tool [4]. Figure 1 demonstrates three basic grading algorithms implemented by PRT. In Figure 1(a), PRT has been used to grade multipart dependent questions (graded in nodes 1, 2, and 3) and in Figure 1(b) to grade multipart independent questions (graded in nodes 1, 2, and 3). In Figure 1(c), PRT has been used to grade a question in node 1 and to provide students with specific feedback according to mistakes trapped in node 2 or node 3. The difference is that, in Figures 1(a) and 1(b), each node awards some points; however, in Figure 1(a), the grading stops at the very first wrong answer, while in Figure 1(c), full points are awarded at node 1, the rest are used to find anticipated mistakes for providing feedback to students. Smart use of PRT reduces the amount of codes an author has to write to achieve the same goal.



(a) (b) (c)

Figure 1: Grading algorithms for grading multipart questions: (a) dependent questions (b) independent questions, and (c) feedback provision for wrong answers.

2.3 Target Groups and Data Collection

Seventy nine, sixty four, and ninety one students were enrolled for MT 274 for the academic years 2018/2019, 2019/2020, and 2020/2021, respectively. Only the last two groups participated in the online tutorials. Studying the overseeing of online tutorials for MT 274, two categories of data were collected, namely qualitative and quantitative data.

2.3.1 Qualitative Data

In this category, the items were evaluated by examining their quality as compared to the classical approach. The quality of the developed STACK question was described by its formulation as compared to the classic paper-pen questions. The quality of the grading algorithm was evaluated by examining the state-of-the-art of the grading technique and its correctness. The provision of specific feedback to students from their anticipated mistakes was also evaluated as high-quality practice for online learning activities and assessments for learning. Assistance in students' responses was measured by the availability of appropriate CAS syntax hints.

2.3.2 Quantitative Data

In this category, data were collected numerically from students' participation or performance. Data on students' performance of online tutorials and quizzes were collected from Moodle grade book, by working with the tutorials and quizzes scores, independently. Students' participation was measured from Moodle's tutorials/quizzes summary report, where participation was represented as a percentage of the involved students with respect to the whole class. Students engagement was measured by the eagerness of the students to acquire higher scores by reasonably re-attempting the tutorials, but also by working with the tutorials/quizzes before the deadline passes. On the other side, data on performance for the 2018/2019 students group were obtained from the 2018/2019 UDSM examination results.

2.4 Activities Design

It is easier to author a question from existing textbook questions. That is why most of the questions developed in this work were derived from the original questions of MT 274 reference books [22, 23, 24]. However, there are questions that were developed from scratch. In some cases, it proved difficult to make meaningful sampling of variables values. In such cases, we chose to author a number of similar questions with fixed variables values and then used Moodle quiz random question feature to sample one question from a set of similar questions. In addressing the issue raised by Gage [3] of authoring easily automatic gradable questions, we have adapted the paper-pen questions of the classical tutorials for the 2018/2019 academic year group, which were concrete with a reasonable amount of analytical tasks.

The developed questions were categorized into two types: tutorial questions (for mastering course content) and quiz questions (for formative assessment purposes). Ten tutorials and five quizzes as well as eleven tutorials and four quizzes were conducted in 2019/2020 and 2020/2021, respectively. In tutorials, students were allowed with unlimited time to repeatedly attempt the tutorials to improve their scores while learning through the specific and general feedback they received. It was different in quizzes, students were given between one- to two-hour intervals within two days to attempt the quizzes. Because students were re-attempting the tutorials as many times as they wished, randomization of question variables was necessary. The validity of authored STACK randomized questions is critical, and is achieved by studying mathematical analysis used for sampling question variables' values. A total of 90 questions were developed to cover the content of the course for both tutorials and quizzes.

A digitized mathematics learning activity should not become a barrier for students to learn mathematics. Students were given guidance and syntax hints whenever necessary. That is why the first tutorial was devoted aimed at empowering students with STACK command skills.

2.5 Data Analysis

Data were analyzed in different ways depending on their nature. The qualitative data were analyzed on their quality with respect to the classical approach of overseeing tutorials. Quantitative data were analyzed using the classical approach of numerical comparison, graphical visualization, and correlation analysis. The students' online scores were examined for correlation with the final scores by using Pearson's coefficient of correlation r for n data points pairs (x, y) defined by [24]

$$r = \frac{nS_{xy} - S_x S_y}{\sqrt{\left(nS_{x^2} - S_x^2\right)\left(nS_{y^2} - S_y^2\right)}}, \qquad (1)$$

where $S_{xy} = \sum_{i=1}^{n} x_i y_i$, $S_x = \sum_{i=1}^{n} x_i$, $S_y = \sum_{i=1}^{n} y_i$, $S_{x^2} = \sum_{i=1}^{n} x_i^2$ and $S_{y^2} = \sum_{i=1}^{n} y_i^2$. The coefficient of correlation uses a linear scale between 0 and 1; $r \in [0, 1]$. The value $r = 1$ means highly correlated and $r = 0$ means no correlation for data points x and y.

3. STACK Questions Development

A total of 80 STACK questions were developed for both tutorials and quizzes. Quizzes' questions were similar to those of tutorials, except that they added little challenges to students.

3.1 Competent STACK Question Components

3.1.1 Specific and General Feedback

Specific feedback is provided based on individual mistakes in the solution of the problem. Specific feedback is shown in Figure 2 with the question derived from the question by Burden and Faires [22], assisting a student on the inputs violating the Intermediate Value Theorem (IVT). The general feedback is given to all students as an extra resource for learning the intended concept or to provide a detailed solution of the problem. In this work, we chose the former as we wanted students to use the concept to solve the problems independently. Figure 3 shows general feedback describing the error bound for Taylor's polynomial approximation.

3.1.2 Floating Point Values Handling

By default, STACK forbids the use of floating point values, and the reason is that computers can not store floating point values precisely [25]. Maxima, which is behind the STACK execution, also stores floating point values differently [26], which makes comparison ambiguous. However, the author can allow the use of floating point values where necessary. Because the use of floating point values is the base of any approximation method, it was reasonable to activate this feature in MT 274 questions. STACK has several functions to handle floating point values (Table 1). Testing equality of floating point variables x and y is best achieved by

$$|x - y| < \epsilon \qquad (2)$$

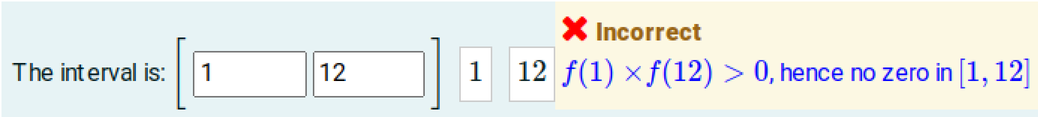


Figure 2: Specific feedback for inputs violating the Intermediate Value Theorem (IVT).

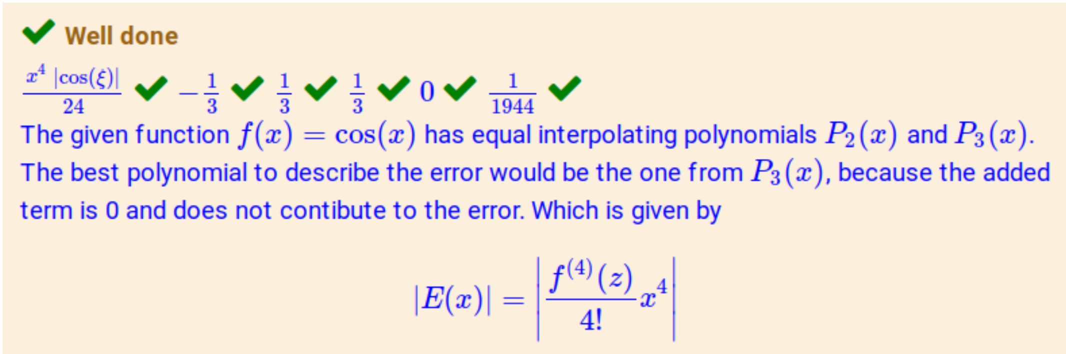


Figure 3: General feedback for Taylor polynomial approximation's error bound.

Table 1: Sample STACK functions for handling floating point values.

Function	Argument type	Task
decimalplaces(f,n)	float f, integer n	Rounds f into n decimal places.
significantfigures(f,n)	float f, integer n	Rounds f into n significant figures.
round(n)	integer n	Rounds n into the nearest integer.

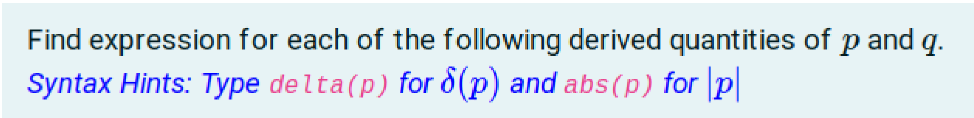


Figure 4: Syntax hints for writing mathematics symbols is provided in blue fonts.

where ϵ is the tolerance defined by the properties/errors of x and y,

3.1.3 Syntax Assistance

In any electronic learning resource, students should spend much of their time on the subject, and not on how to digitally present their responses. To address this issue, we provided syntax hints of the solutions when needed. This avoids disappointing students in their achievements with respect to digital mathematics. Figure 4 shows syntax hints in writing mathematics symbols, $\delta(p)$, and absolute value, $|p|$, which are required in presenting the solution in error analysis.

3.1.4 Randomization of Variables

Students were allowed to reattempt the tutorials as many times as they wish to improve their scores while learning the concepts. This practice is worthwhile because, in every attempt, different variant of the question is sampled, which makes student engaged in solving the problems. Constructing many variants of the question requires smart sampling of the problem variables, otherwise, useless samples would be included in the tutorial or quiz. To alleviate the useless sampling, we determined the algebraic solution of each question to learn a meaningful sample of each variable. Apart from creating meaningful variants, we avoided creating a question variant whose variable values may affect the difficulty level of the question. STACK provides a means to deploy a number of variants, which can then be inspected separately for any suspicious values of the variables. In every question with randomized variables, we deployed and inspected at least 50 variants. STACK comes with a bunch of randomization functions, some are listed in Table 2. The codes in Code 1 (Appendix) show a function to randomly sample one out of ten Initial Value Problems (IVP).

3.2 Selected STACK Question Cases

3.2.1 Guided Proof Question

Still, the biggest challenge in digital assessments in mathematics is the ability to ask reasoning and proof questions. One way to achieve this is to create a frame of the proof and ask students to respond with some mathematical expressions and reasoning arguments. Problem 1, is an example of a question in MT 274 demonstrating a proof question, whose student responses are shown in Figure 5.

Table 2: Some randomization STACK functions.

Function	**Argument type**	**Task**
rand(n)	integer n	Sample integer between 0 and n
rand(n.0)	integer n	Sample floating point value between 0.0 and n.0
rand_with_prohib(a,b,exl)	integer a, integer b, list of integers exl	Sample integer between a and b excluding those in list
random_permutation(lst)	list lst	Return a randomly permuted list

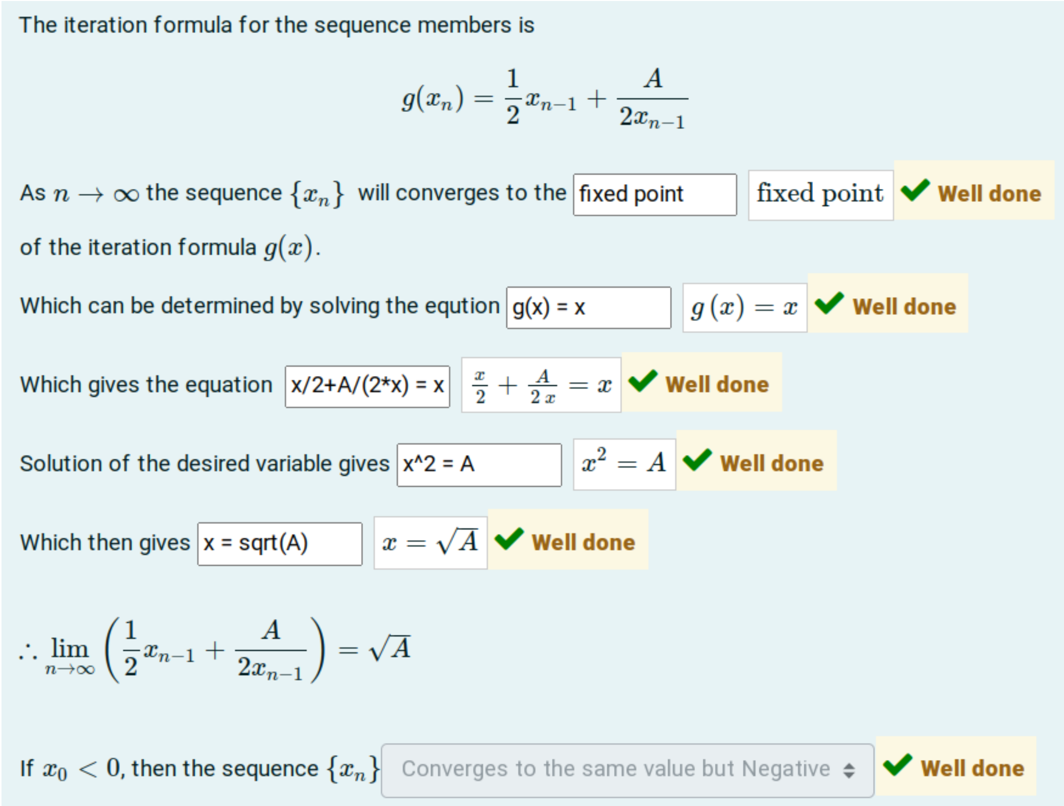


Figure 5: Guided proof frame for solution of Problem 1.

Problem 1: Show that if A is any positive number, then the sequence defined by

$$x_n = \frac{1}{2}x_{n-1} + \frac{A}{2x_{n-1}}, n \geq 1$$

converges to $\sqrt{A}$ whenever $x_0 > 0$. What happens if $x_0 < 0$.

3.2.2 Randomization for IVP Variables

IVPs are solved numerically by using the Euler method, Improved Euler Method, or Runge-Kutta (of order 4) method. The case of the IVP from Burden Faires [22] is shown in Problem 2. The IVP (5) is used to create a general STACK question by introducing variables a, b, c, t_0, t_f and y_0 to create a new problem

$$y' = \frac{ay^2}{b+ct}, \quad t_0 \leq t \leq t_f, \quad y(t_0) = y_0, \tag{3}$$

whose general solution is given by

$$y = -\frac{c}{a\ln(ct+b)+k} \tag{4}$$

where k is a constant of integration. The general solution (4) assists in the restriction of sampling in variables a, b, and c. For instance, values of a or c should not be zero, otherwise, it changes the problem to zero derivative function or derivative function independent of variable t, which affects difficult levels between question variants. Using initial condition $y(t_0) = y_0$, we get an expression (6) of the constant k as

Problem 2: Use modified Euler method with step size $h = 0.2$ to find the numerical solution of the initial value problem (5)

$$y' = \frac{y^2}{1+t}, \quad 1 \leq t \leq 2, \quad y(1) = -\frac{1}{\ln 2} \tag{5}$$

Compare your results numerically with the analytical solution $y = -\dfrac{1}{\ln(1+t)}$.

$$k = -\frac{c + ay_0 \ln(ct_0+b)}{y_0} \tag{6}$$

which tells which combinations of the random variables a, b, c, t_0 and y_0 would produce meaningful variant. It is clear that $y_0 = 0$ will blow the solution. The variable t_f would be sampled based on the number of iterations in the solution as it is free from the question setup. A possible set of variables sampling are shown in Table 3.

Table 3: Sampling the variables of the IVP (3).

Variable	Sampling
a	$a > 0$
c	$c > 0$
t_0	$t_0 \geq 0$
y_0	$y_0 \neq 0$
b	b such that $ct_0 + b > 0$

The Code 2 (in Appendix) shows the implementation of the sampling procedure described in Table 3. One variant generated by Code 2 is

$$y' = \frac{3\,y^2}{6\,t - 11},\ 3 \leq t \leq 6, y(3) = -\frac{1}{\ln{(3)}},\ h = \frac{3}{5}$$

whose question preview, grading, and PRT grading algorithm with marks distribution are shown in Figures 6(a), 6(b) and 6(c), respectively.

3.2.3 Executing with Iterative Procedures

Grading iterative procedure is somehow tricky as the number of iterations the process takes varies from one question variant to another. In this work, we randomly chose 5 iterations for grading, including the first and last iterations. This is justified by the fact, that if certain iteration values are correct, students deserve points for the previous iteration as well. The code implementing the selection of the rows for marking is shown in Appendix as Code 3.

3.2.4 Analytical Problem

Analytical problems share almost the same level of difficulty in development as the proof question because the responses are based on arguments, reasoning, and application of theorems. Here, we present a case of analyzing the best iteration formula as shown in Problem 3.

Problem 3: The function $f(x) = 2x^3 - 6x^2 + 3$ is known to have a zero inside the interval $(2, 3)$. Two iteration formulae

Formula 1: $x_{n+1} = 3\left(1 - \dfrac{1}{2x_n^2}\right)$

Formula 2: $x_{n+1} = \dfrac{x_n^2}{3} + \dfrac{1}{2x_n}$

have been suggested for finding the zero. Explain clearly the steps you would take in deciding which formula to use. Starting with $x_0 = 1.5$, calculate the iterates x_1, x_2 and x_3 (Round the iterate values to 6 decimal places).

Digitizing Problem 3 requires a student to follow some defined steps to end up with the conclusion of the best formula for approximating the zero. One can randomize the function $f(x)$ and zero's interval. However, in this work, only randomization of solution steps, formula names, initial guess x_0, and the number of decimal places were carried out. Figure 7 shows the guided solution and its grading. The code implementing solution of the Problem 3 is shown in Code 4 as an Appendix.

3.2.5 Graphical Feedback

Graphical feedback was provided to assist students in visualizing a hidden or difficult concept. In Problem 4, the student is required to compute the fourth derivative and second derivative functions to determine the local maxima of the functions in the interval of integration. The situation becomes worse if the function has local maxima between the limits of integration, instead of end limit points. Figures 8(a) and 8(b) show the general feedback and the graphs of the respective higher derivative function.

Problem 4: Determine the values of n and h required to approximate $\int_0^1 e^{2\,x}\,\sin{(3\,x)}dx$ within 10^{-3} by using the composite

(a) trapezoidal rule (b) Simpson's rule

Figures 8(a) and 8(b) provide visual support to realize the local maxima of the higher derivative absolute value functions $|f''(x)|$ and $|f^{(iv)}(x)|$, within the limits of integration $x \in [0, 1]$ of part (a) and part (b) of Problem 4, respectively, for $f(x) = e^{2x}\sin 3x$.

4. Results

Tutorials and quizzes were overseen online successfully. Competent STACK questions were developed, including analytical and guided

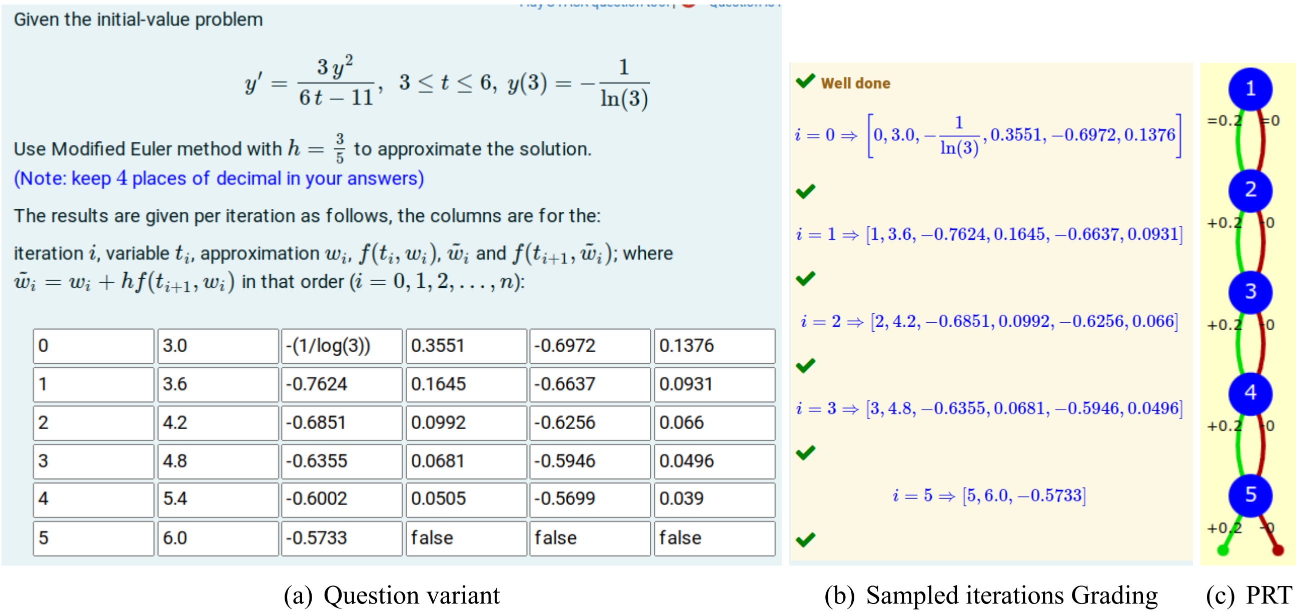


Figure 6: Question variant of the IVP generated by Code 2 of (3): (a) question variant with student inputs (b) sampled iterations grading (c) grading algorithm with fractional marks distribution.

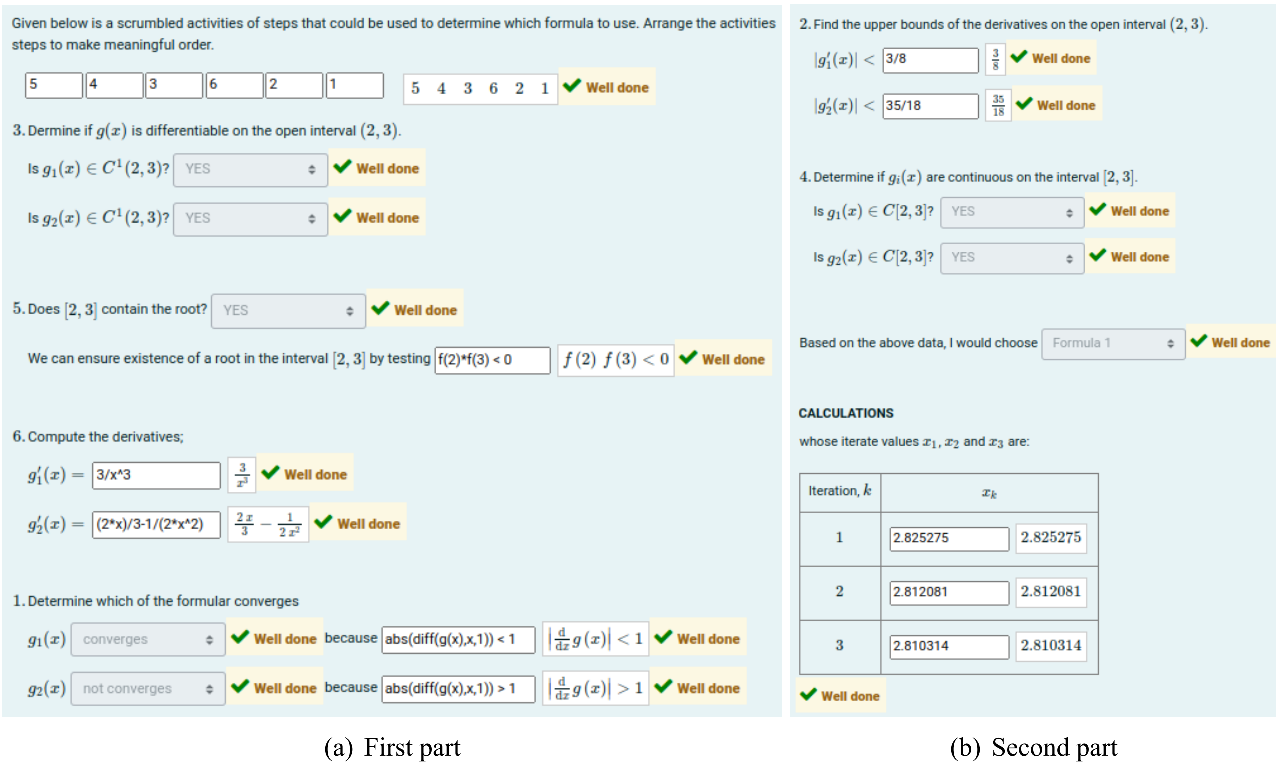
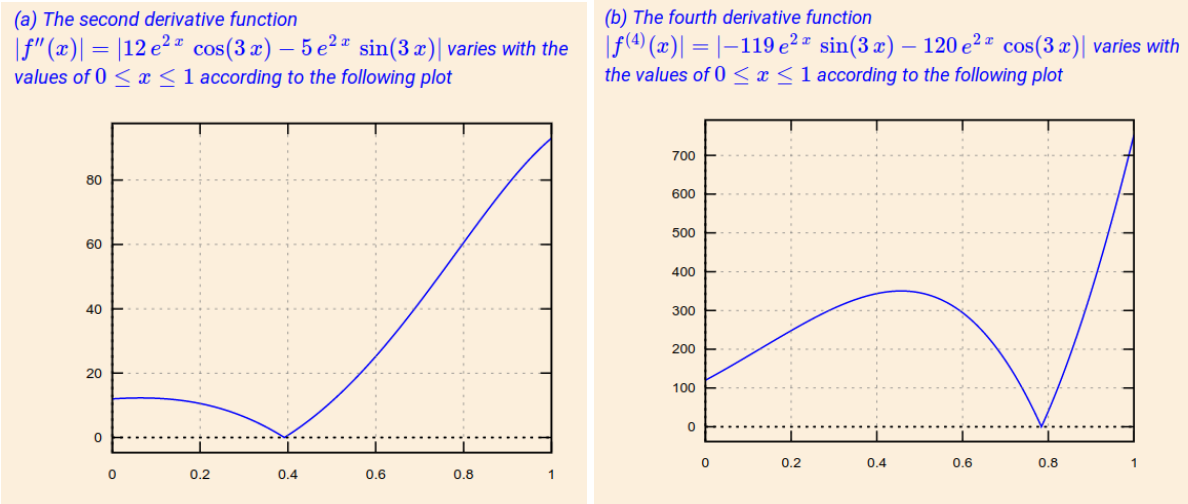


Figure 7: Grading guided solution of the analytical Problem 3.

proof questions. Computational questions with direct and iterative procedures were also developed. The questions engaged the students with both critical thinking and reasonable effort for their solutions. Students received general feedback to learn the respective concept and specific feedback from their anticipated mistakes. When needed, the feedback included graphs for better visualization of hidden concepts. The randomization of question variables was achieved competently by sampling question variables from the mathematically analyzed expressions.

Students worked independently without the assistance of the course instructors. Hence, engagement was higher compared to the classical sessions, in which students rely on colleagues' contributions or instructor's solutions. Table 4 shows good percentage participation of students in both tutorials and quizzes with exception of quiz 5 of 2019/2020 and tutorial 10 of 2020/2021.



(a) $y = |f''(x)|,\ 0 \le x \le 1$

(b) $y = |f^{(iv)}(x)|,\ 0 \le x \le 1$

Figure 8: Graphical general feedback for $f(x) = e^{2x}\sin 3x$ to realize the error bounds for composite (a) trapezoidal rule, and (b) Simpson's rule.

Table 4: Percentage participation for online tutorials and quizzes.

Serial	2019 / 2020		2020 / 2021	
	Tutorial	Quiz	Tutorial	Quiz
1	96.9%	95.3%	93.6%	97.9%
2	95.3%	93.8%	91.5%	94.7%
3	96.9%	96.9%	94.7%	93.6%
4	96.9%	95.3%	90.4%	88.3%
5	96.9%	68.8%	93.6%	84.0%
6	98.4%		68.1%	
7	95.8%		76.6%	
8	89.1%		85.1%	
9			74.5%	
10			56.4%	

Table 5 suggests performance for the academic years 2019/2020 and 2020/2021 are not alarming with reference to 2018/2019 classical tutorials. Actually, the pass rates are a little higher than that of 2018/2019.

Table 5: Final students' results for 2018/2019, 2019/2020 and 2020/2021 academic years.

Grade	Classical	Online	
	2018/2019	2019/2020	2020/2021
A	2.6%	3.2%	1.1%
B+	11.7%	11.1%	11.2%
B	28.6%	17.5%	29.2%
C	37.7%	54.0%	48.3%
D	3.9%	3.2%	4.5%
E	15.6%	11.1%	5.6%
Pass	80.5%	85.7%	89.9%
Fail	19.5%	14.3%	10.1%

Students were allowed to re-attempt the tutorials, hence scores distributions are skewed to higher scores (Figures 9 – 12), with exception of Tutorial 3 of 2019/2020 and Tutorials 6 and 7 of 2020/2021. Figures 13 and 14 show that scores for quizzes are moderately normally distributed with the exception of Quiz 5 of 2019/2020 and Quiz 2 of 2020/2021. The distribution classifies the students into three performance groups; small low-performance group, large moderate-performance group, and small high-performance group.

Students' scores for online activities were compared with the final scores in Figure 15. Figures 15(a) and 15(b) show the correlation between online scores and final scores for the academic years 2019/2020 and 2020/2021, respectively. The red circular dots are fitted by the blue line in Figure 15(a), while the black line ignored the leftmost data point, which is considered as an outlier, whose coefficient of correlation (r) are, respectively, 0.4176 and 0.4611. Figure 15(b) shows linear correlation with $r = 0.7329$. The Figures suggest that students' online scores were correlated with their final University Examination results. The University Examinations are administered in a secure environment and therefore free from cheating. The results suggest that online tutorials were meaningful and reflected true students' performances.

5. Discussion

Online tutorials and quizzes offered great benefits to students by working remotely and received specific feedback based on their mistakes

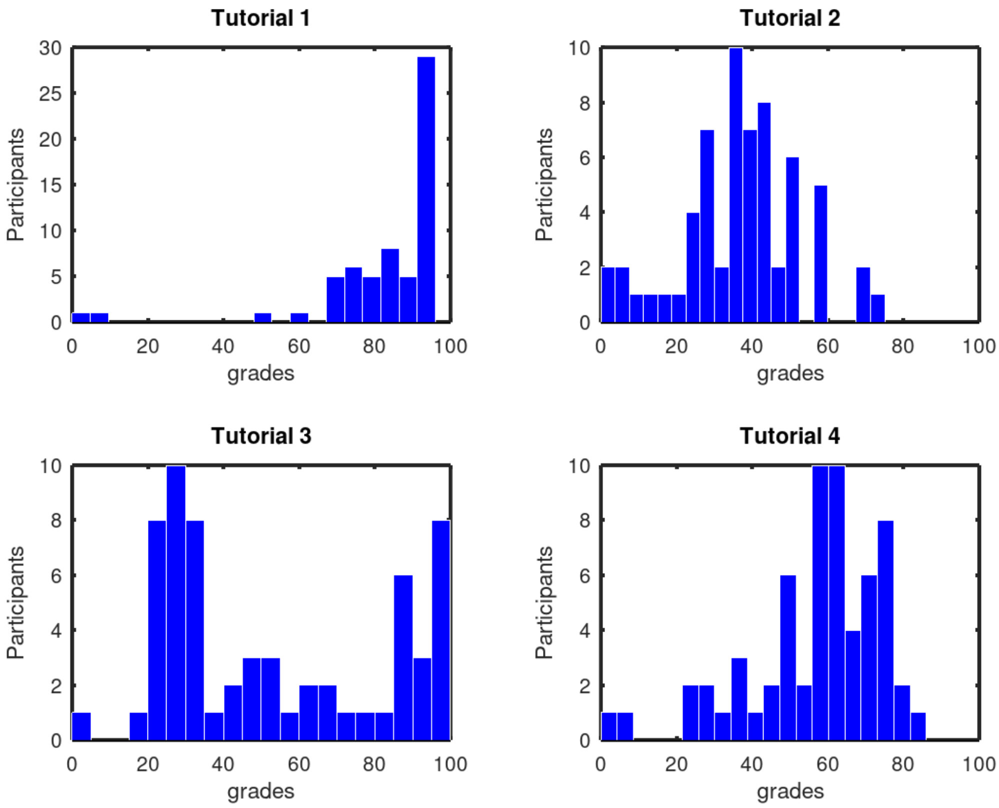


Figure 9: Distributions of scores of tutorials 1-4 for 2019/2020.

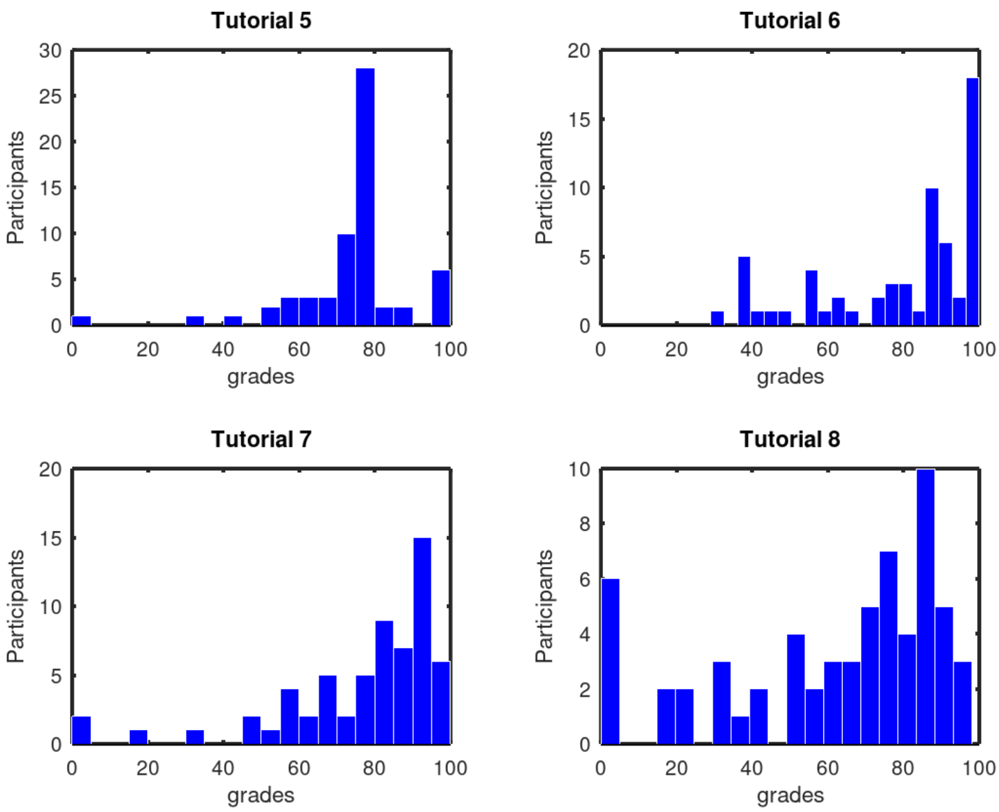


Figure 10: Distributions of scores of tutorials 5-8 for 2019/2020.

[27]. The behavior of students to cheat on online practice was expected as documented by Juma et al. and Kocdar et al. [18, 28]. The same has been observed in the current study, where students misused the system by submitting empty quizzes, just to receive correct answers

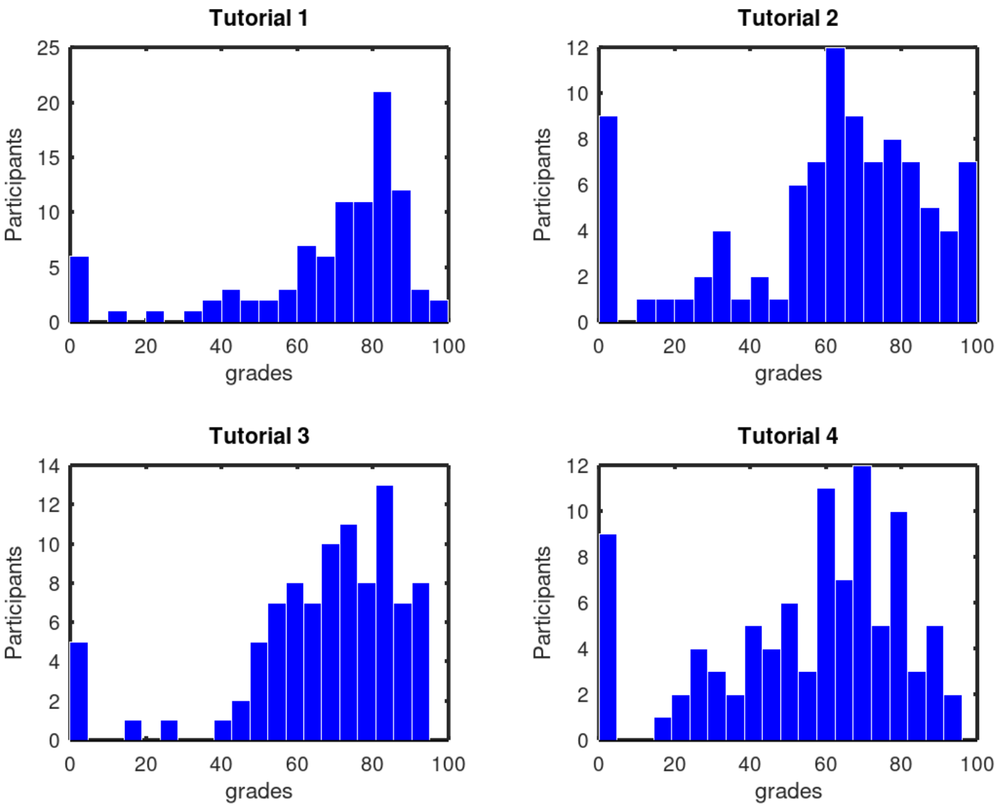


Figure 11: Distributions of scores of tutorials 1-4 for 2020/2021.

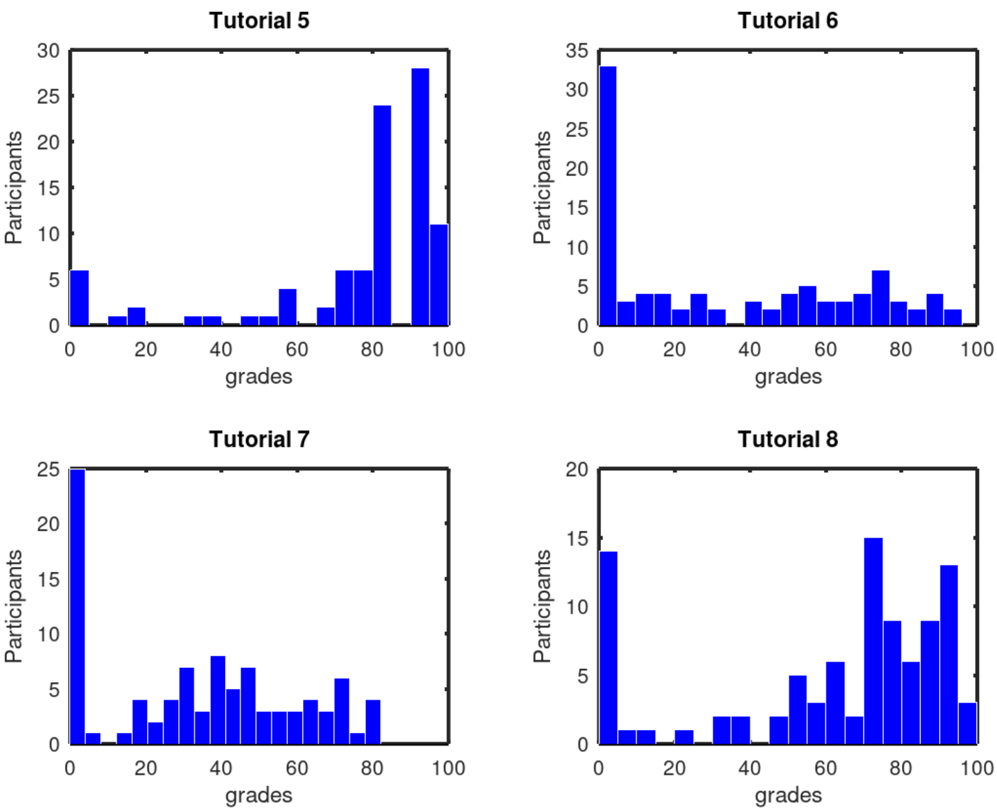


Figure 12: Distributions of scores of tutorials 5-8 for 2020/2021.

(feedback). Because every question had at least 50 variants, in most cases it positively helped them to learn the intended concept, as the probability of sampling the same question variant is at most 2%, which is very low. There are a few cases where students reported errors in

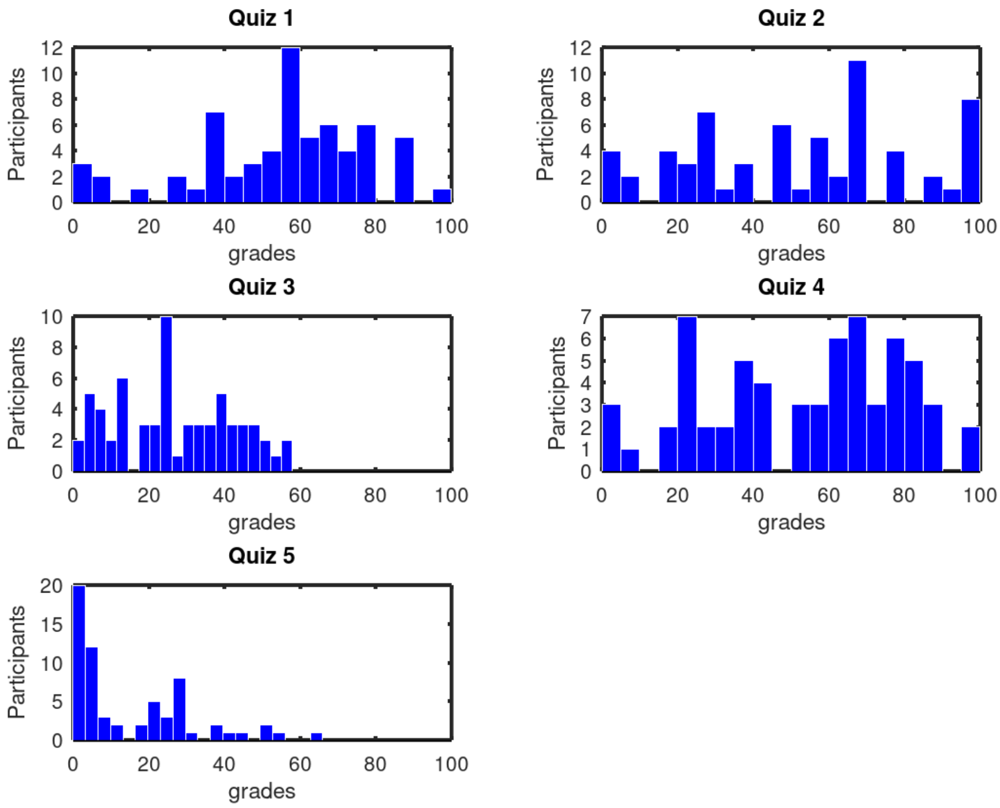


Figure 13: Distributions of scores of quizzes 1-5 for 2019/2020.

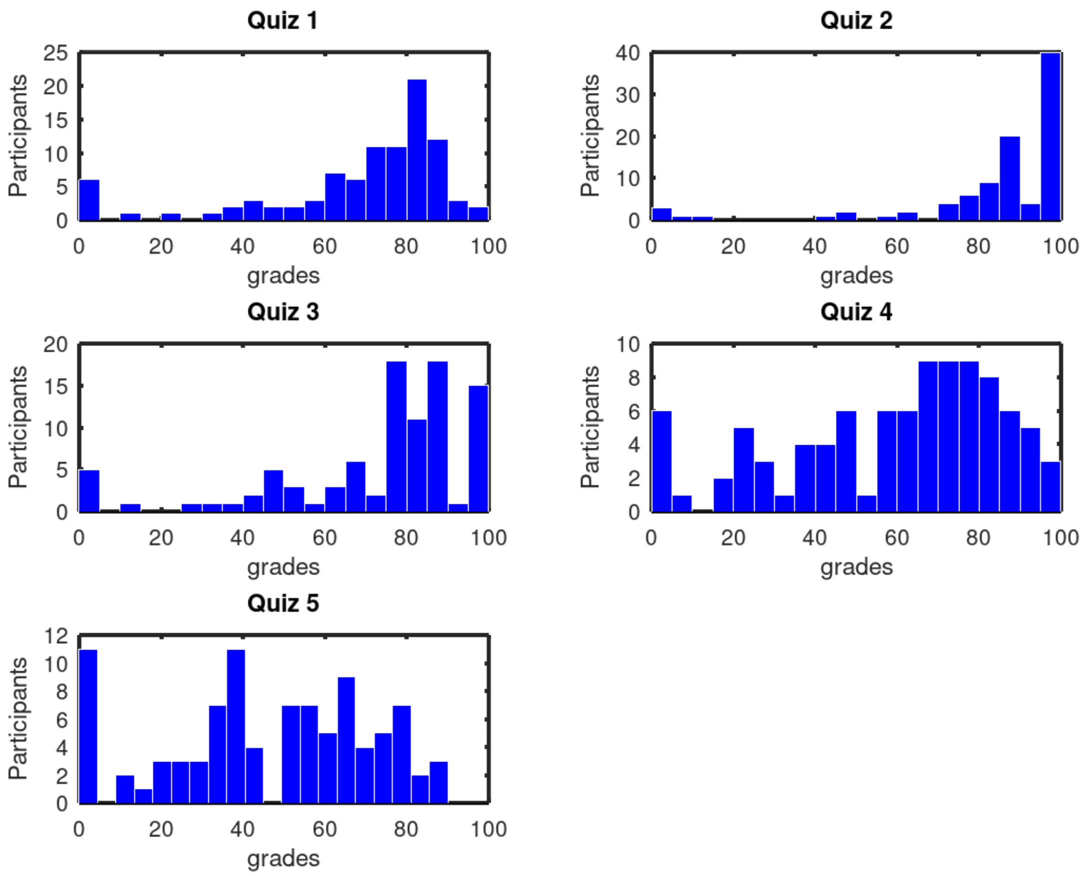
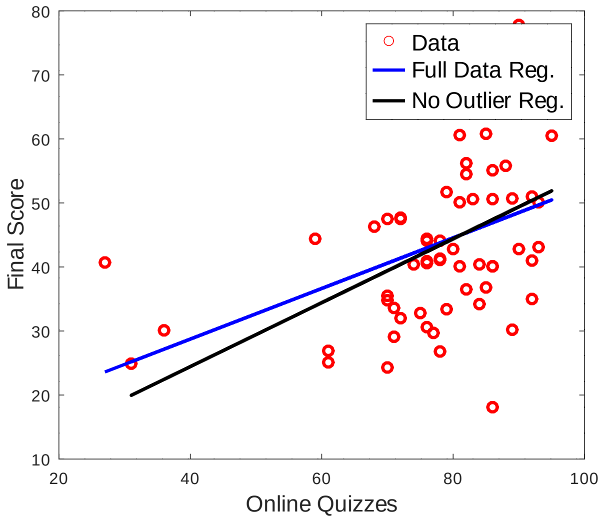
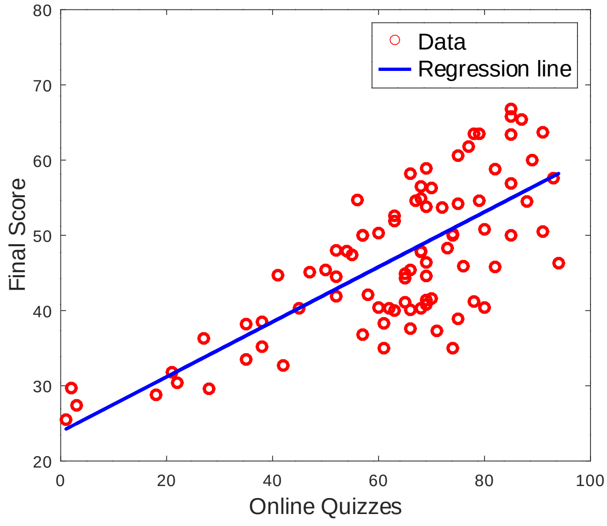


Figure 14: Distributions of scores of quizzes 1-5 for 2020/2021.

grading. Thanks to Moodle regrading feature, we addressed the issue by regrading the quizzes after



(a) 2019/2020; $r = 0.4176$, $r = 0.4611$



(b) 2020/2021; $r = 0.7329$

Figure 15: Correlation of online quizzes with final scores.

the solution and programming errors were fixed. The practice assisted well a group of students with limited social interaction to discuss with the instructors/colleagues in classical tutorials, as discussed by Robinson et al. [29].

Ninety competent STACK questions have been developed covering the content of MT 274. The questions enable effective and efficient overseeing of the online tutorials. However, developing competent STACK questions requires a significant amount of time. As discussed by Nakamura and Nakahara [20], it may be difficult for instructors to be motivated. It involves critical thinking of problem theory, scenario, and its valid solution, especially when randomization is required. However, this worthy time investment is useful to enrich course digital resources for the coming years and the public in general, as supported by Zerva [8]. That is why Nakamura and Nakahara [20] have created an item bank system where educators can publicly share their valuable STACK questions.

Students' performance in online practice moderately correlated with their final UDSM examination results, this is very similar to the findings by Sangwin [7]. However, the result of online quizzes may mislead our decision-making, as some students do not work on their own [18, 28]. Some mechanisms, such as, use of cameras, may be needed to ensure the security of the practice. An alternative way is to administer the online assessment in UDSM lecture halls; however, there may be an issue with Moodle handling a large number of users concurrently. Secured practice with appropriate resources can be employed in summative assessments to save supervision resources and human efforts in marking.

6. Conclusion

Online tutorials and quizzes were successfully administered for MT 274 at UDSM for the academic years 2019/2020 and 2020/2021. Ninety competent STACK questions had been developed to cover the content of the course. The flexibility of STACK permitted authoring of direct computation, iterative, analytical, guided proof questions as well as the objective types classical questions; drop-down list, multiple choice, and true/false questions [30]. These objective STACK questions offer more advantage for mathematics and related subjects than their Moodle objective question types counterparts.

Students' participation and engagement were reasonably good with exception of the activities at the end of the semester, where students were preparing for their final examinations. This may address the issue of both poor attendance and individual participation in classical tutorials [31]. The distribution of students' scores was evidently divided into the small low-performance group, large moderate-performance group, and small high-performance group. There is no evidence of any negative impact of the online tutorials' content or design on students' performance. Students' final scores show a reasonable correlation with the online practice scores, which suggests meaningful online assessments.

Developing competent STACK questions

requires a significant amount of time investment, in return, we benefit from the digital resources for the next several years to come. The growing need of introducing online courses in Mathematics is hindered by the lack of effective and competent online assessments. The success of online tutorials and quizzes is a good step towards not only online but also smart assessment systems in Mathematics. The competent use of STACK shall revolutionize the existing barrier of not being able to digitally access Mathematics learning, delivering interactive mathematics content, and conducting online Mathematics programmes. Currently, many Mathematics and Engineering Departments can not offer online degree programmes. This opens scholarly discussion on the possibility of introducing online programmes, where students will interact with content and accessed electronically and competently.

References

[1] Sangwin, C., *Assessing mathematics automatically using computer algebra and the internet*, Teaching Mathematics and Its Applications: International Journal of the IMA, **23**(1): p. 1, 2004, doi: http://dx.doi.org/10.1093/teamat/23.1.1.

[2] Docherty, F., Petrie, E., Lawson-Perfect, C., *NUMBAS Online Assessment - an Introduction to Creating Exams and Resources and Editing Questions*, in *14th Annual Learning and Teaching Conference, Glasgow, UK, 18 June*, 2021.

[3] Gage, M.E., *Methods of Interoperability: Moodle and WeBWork*, Journal of Learning Analytics, **4**(2): p. 22, 2017, doi: http://dx.doi.org/10.18608/jla.2017.42.4.

[4] Sangwin, C., *Computer Aided Assessment of Mathematics*, University Press Oxford, UK, 2013, ISBN 978-0-19-966035-3.

[5] Drijvers, P., *Digital assessment of mathematics: Opportunities, issues and criteria*, Mesure et évaluation en éducation, **41**(1): p. 41, 2018, doi: http://dx.doi.org/https://doi.org/10.7202/1055896ar.

[6] Lowe, T., Sangwin, C., Jones, I., *Getting started with STACK*, Loughborough University, 2019, doi: https://docs.stack-assessment.org/content/2019-STACK-Guide.pdf.

[7] Sangwin, C., *Developing and evaluating an online linear algebra examination for university mathematics*, in *Eleventh Congress of the European Society for Research in Mathematics Education*, volume TWG21(15), Utrecht University, Utrecht, Netherlands, 2019, doi: https://hal.archives-ouvertes.fr/hal-02430556.

[8] Zerva, K., *Developing STACK assessments in Edinburgh 2017-2019*, in *In: Contributions to the 1st International STACK conference 2018, Friedrich-Alexander-Universität Erlangen-Nürnberg: Fürth, Germany*, 2018, doi: http://dx.doi.org/DOI:10.5281/zenodo.2585816.

[9] Matti, P., *Calculus courses' assessment data*, Journal of Learning Analytics, **4**(2): p. 12, 2017, doi: http://dx.doi.org/10.18608/jla.2017.42.

[10] Ellis, J., Hanson, K., Nuñez, G., Rasmussen, C., *Beyond Plug and Chug: an Analysis of Calculus I Homework*, International Journal of Research in Undergraduate Mathematics Education, **1**(2): p. 268, 2015, doi: http://dx.doi.org/10.1007/s40753-015-0012-z.

[11] Kinnear, G., *Delivering an Online Course Using STACK. In: Contributions to the 1st International STACK Conference*, International STACK Conference, 2019, doi: doi:10.5281/zenodo.2565969.

[12] Kinnear, G., *Using JSXGraph for diagrams and interactivity*, (Demonstration slides), School of Mathematics, The University of Edinburgh, 2020, doi: https://eams.ncl.ac.uk/sessions/2020/using-jsxgraph-for-diagrams-and-interactivity/slides.pdf.

[13] Davies, B., Smart, T., Geraniou, E., Crisan, C., *STACKification: automating assessments in tertiary mathematics. In: Proceedings of the Twelfth Congress of the European Society for Research in Mathematics Education. European Society for Research in Mathematics Education*, 2022, doi: https://discovery.ucl.ac.uk/id/eprint/10145131.

[14] Eichhorn, A., Helfrich-Schkarbanenko, A., *Question Answering in STACK Applying String*

Similarity, International Journal of Emerging Technologies in Learning (iJET), **17**(23): p. 56, 2022, doi: http://dx.doi.org/10.3991/ijet.v17i23.35893.

[15] Klischat, C., Becker, P., Vasko, M., *STACK is more than Maths - Development of Online Problems for Mechanics and Electrotechnics*, in *In Contributions to the 1st International STACK conference 2018 in Furth, Germany*, 2019.

[16] Orthaber, M., Stütz, D., Antretter, T., Ebner, M., *Concepts for E-Assessments in STEM on the Example of Engineering Mechanics*, International Journal of Emerging Technologies in Learning (iJET), **15**(12): p. 136, 2020, doi: http://dx.doi.org/10.3991/ijet.v15i12.13725.

[17] Ellonen, O., Vesapuisto, M., Vekara, T., *Experiences on development and design of stack problems for circuit analysis*, Athens Journal of Technology & Engineering, **7**(3): p. 185, 2020, doi: http://dx.doi.org/10.30958/ajte.7-3-2.

[18] Juma, Z.O., Ayere, M.A., Oyengo, M.O., Osang, G., *Evaluating Engagement and Learning Based on a Student Categorization using STACK, Exam Data, Key Informant Interviews, and Focus Group Discussions*, International Journal of Emerging Technologies in Learning (iJET), **17**(23): p. 103, 2022, doi: http://dx.doi.org/10.3991/ijet.v17i23.36619.

[19] Juma, Z.O., Okoth, I.O., Oyengo, M.O., *Delivery of Integral Calculus at Maseno University: Is STACK Really Playing an Integral Part?*, International Journal of Emerging Technologies in Learning (iJET), **17**(23): p. 64, 2022, doi: http://dx.doi.org/10.3991/ijet.v17i23.36621.

[20] Nakamura, Y., Nakahara, T., *Development of an Item Bank System for the Mathematics e-Learning System STACK*, 2013.

[21] Black, P., Harrison, C., Lee, C., Marshall, B., William, D., *Assessment for Learning- putting it into practice*, Open university Press, Maidenhead, U.K., 2003, doi: http://oro.open.ac.uk/24157/.

[22] Burden, R.L., Faires, J.D., *Numerical Analysis*, Brooks/Cole, 9th edition, 2011.

[23] Boyce, W.E., Diprima, R.C., Meade, D.B., *Elementary Differential Equations and Boundary Value Problems*, John Wiley & Sons, Inc., New York, 11th edition, 2017, ISBN ISBN-13 978-1-119-25600-7.

[24] Kreyzig, E., *Advanced Engineering Mathematics*, John Wiley & Sons, INC, 10th edition, 2011.

[25] Pine, D.J., *Introduction to Python for Science and Engineering*, CRC Press, Taylor & Francis Group, 2019, ISBN ISBN-13: 978-1-138-58389-4.

[26] *Maxima - A computer Algebra System: Maxima Reference Manual*, (accessed on November 24, 2022), doi: "https://maxima.sourceforge.io/documentation.html".

[27] Akugizibwe, E., Yong, J.A., *Perspectives for Effective Integration of e-Learning tools in University Mathematics Instruction for Developing Countries*, Education and Information Technologies, **25**: p. 889, 2020, doi: http://dx.doi.org/10.1007/s10639-019-09995-z.

[28] Kocdar, S., Karadeniz, A., Peytcheva-Forsyth, R., Stoeva, V., *Cheating and Plagiarism in E-Assessment: Students' Perspectives*, Open Praxis, **10**(3): p. 221, 2018, doi: http://doi.org/10.5944/openpraxis.10.3.873.

[29] Robinson, M., Loch, B., Croft, T., *Student Perceptions of Screencast Feedback on Mathematics Assessmen*, Int. J. Res. Undergrad. Math. Ed., **1**: p. 363, 2015, doi: https://www.tandfonline.com/doi/full/10.1080/10618600.2019.1573687.

[30] Bach, S., *Using multiple choice questions in STACK - reasons and examples*, 2020, doi: http://dx.doi.org/10.5281/zenodo.3945809.

[31] Snopce, H., Alija, S., *The Effects of Students Attendance in the Success of Undergraduate Mathematical Courses-the Case of the See-University*, ICEMST 2015: International Conference on Education in Mathematics, Science & Technology. The Eurasia Proceedings of Educational & Social Sciences (EPESS), 2015, **2**: p. 101, 2015, doi: http://www.epess.net/en/pub/issue/30319/332890.

Appendix

Code 1: Randomization of IVP to be embedded in the same stem of question.

```
/* Ten IVP from Burden and Fares */
ivpsampling(n):= block(
    if is(n=1) then
        [2/t*y+t^2*exp(t),1,2,0,t^2*(exp(t)-exp(1))]
    elseif is(n=2) then
        [1/t^2-y/t-y^2,1,2,-1,-1/t]
    elseif is(n=3) then
        [t*exp(3*t)-2*y,0,1,0,1/5*t*exp(3*t)-1/25*exp(3*t)+1/25*exp(-2*t)]
    elseif is(n=4) then
        [1+y/t, 1,2,2,t*log(t)+2*t]
    elseif is(n=5) then
        [1+(t-y)^2,2,3,1, t+1/(1-t)]
    elseif is(n=6) then
        [cos(2*t)+sin(3*t),0,1,1,1/2*sin(2*t)-1/3*cos(3*t)+4/3]
    elseif is(n=7) then
        [exp(t-y),0,1,1,log(exp(t)+exp(1)-1)]
    elseif is(n=8) then
        [-y+t*y^(1/2),2,3,2,(t-2+sqrt(2)*exp(1)*exp(-t/2))^2]
    elseif is(n=9) then
        [t^(-2)*(sin(2*t)-2*t*y),1,2,2,1/2*t^(-2)*(4+cos(2)-cos(2*t))]
    else [y^2/(1+t),1,2,-log(2)^(-1),-1/log(t+1)];
)
```

Code 2: Sampling of variables as suggested in Table 3.

```
/* sampling the problem variables */
a:1+rand(8);
c:1+rand(8);
t_0:1+rand(3);
y_0:-1/ln(rand([2,3,4,5]));
b:-c*t_0+1+rand(8);
t_f:t_0+rand([1,2,3]);
/* number of decimal places */
dp:rand([4,5,6]);
/* integration contant */
k:-(c+a*y_0*log(c*t_0+b))/y_0;
/* problem parameters */
[fty,t0,tn,y0]:[a*y^2/(b+c*t),t_0,t_f,y_0];
/* sample number of slices */
n:rand([5,6,7,8,9,10]);
if is(n>=8) then tn:tn+1;
/* integration step */
h:(tn-t0)/n;
```

Code 3: Random rows selection for grading the IVP (3).

```
/* sample 5 rows to mark including the first and the last */
randmark:random_permutation(makelist(i,i,2,n));
randmark:sort(makelist(randmark[u],u,1,3));
```

Code 4: Analytical problem solution procedure.

```
/* number of decimal places */
ndp:rand([4,5,6]);
/* The interval */
x1:2; x2:3;
/* correct order of the activities */
list_orders:random_permutation([1,2,3,4,5,6]);
ord:makelist(0,i,1,6);
ord[1]:list_orders[2]; ord[2]:list_orders[6];
ord[3]:list_orders[1]; ord[4]:list_orders[3];
ord[5]:list_orders[5]; ord[6]:list_orders[4];
ordt:matrix(ord);
yn:[[0,false,"NO"],[1,true,"YES"]];
/* Intermediate Value Theorem test */
tivt:f(x1)*f(x2) < 0;
/* the iteration formulae */
iforms:random_permutation([1/3*x^2+1/(2*x),3*(1-1/(2*x^2))]);
g1:iforms[1];
g2:iforms[2];
/* derivative of the iteration formulae */
tdg1:diff(g1,x);
tdg2:diff(g2,x);
/* compute the bounds for the derivatives */
tla1:max(abs(ev(tdg1,x=x1)),abs(ev(tdg1,x=x2)));
tla2:max(abs(ev(tdg2,x=x1)),abs(ev(tdg2,x=x2)));
/* choices for convergence of the two formulae */
if is(tla1<1) then
   (cf1:[[1,true,"converges"],[0,false,"not converges"]],
   tdb1:abs(diff(g(x),x,1))< 1,bf:g1)
else (cf1:[[1,true,"not converges"],[0,false,"converges"]],
   tdb1:abs(diff(g(x),x,1)) > 1);
if is(tla2<1) then
   (cf2:[[1,true,"converges"],[0,false,"not converges"]],
   tdb2:abs(diff(g(x),x,1)) < 1,bf:g2)
else (cf2:[[1,true,"not converges"],[0,false,"converges"]],
   tdb2:abs(diff(g(x),x,1)) > 1);
/* the best formula choice */
f1: if is(tla1<1) then 1 else 0;
f2: if is(tla2<1) then 1 else 0;
tchf: if is(f1=1) then [[1,true,"Formula 1"],[0,false,"Formula 2"]];
else [[0,false,"Formula 1"],[1,true,"Formula 2"]];
tchf:random_permutation(tchf);
/* initial guess for Iteration value calculations */
tx0:decimalplaces(2.0+rand(1.0),2);
/* iteration value calculations using the best Formula */
tx1:decimalplaces(ev(bf,x=tx0),ndp);
tx2:decimalplaces(ev(bf,x=tx1),ndp);
tx3:decimalplaces(ev(bf,x=tx2),ndp);
```

Code 5: Iterative processes initiated for solving IVP (3) with the modified Euler's method.

```
/* initialize the lists */
odesol:makelist([i,decimalplaces(t0+i*h,dp),false,
                 false,false,false],i,0,n);
odesol[1][3]:y0;
/* modified Euler Method main loop */
for i:1 thru n do block(
  odesol[i][4]:decimalplaces(ev(fty,t=odesol[i][2],
                    y=odesol[i][3]),dp),
  odesol[i][5]:decimalplaces(odesol[i][3]+h*odesol[i][4],dp),
  odesol[i][6]:decimalplaces(ev(fty,t=odesol[i+1][2],
                    y=odesol[i][5]),dp),
  odesol[i+1][3]:decimalplaces(
          odesol[i][3]+h/2*(odesol[i][4]+odesol[i][6]),dp)
);
```