2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC
Host Sponsor UNC Charlotte
ASEE AMERICAN SOCIETY FOR ENGINEERING EDUCATION
Paper ID #52820

Work-in-Progress: Use of Oral Exams in a Materials Engineering Numerical Methods and Programing Course

Jonathan R. Brown, The Ohio State University

Jonathan Brown (B.S., M.S. Mathematics, New Mexico Institute of Mining and Technology; Ph.D. Materials Engineering, New Mexico Institute of Mining and Technology) is an Assistant Professor of Practice in the Department of Materials Science and Engineering at The Ohio State University. His background is in computer simulations and theory of polymer glasses and block copolymers for energy applications. He teaches introduction to materials science and engineering and computational materials science courses.

Dr. Adithya Jayakumar, The Ohio State University

Dr. Adithya Jayakumar is currently a Senior Lecturer in the Department of Engineering Education at The Ohio State University (OSU). He received his Masters and PhD in Electrical and Computer Engineering from OSU.

Work in Progress: Use of Oral Exams in a Materials Engineering Numerical Methods and Programing Course

Abstract

At Ohio State University, Materials Science and Engineering (MSE) major sophomores take a numerical methods course. Having been introduced to basic MATLAB syntax and programming concepts as part of the general first year engineering curriculum, in the sophomore level MSE course, they deepen MATLAB skills and learn to apply additional statistics and numerical tools as they relate to the rest of the MSE curriculum.

Assessing programming work at this introductory level is difficult, because students are tempted to overuse external resources such as peers and large-language models for any assignments that are taken home, instead of taking enough time to fully engage and work through the assignment. Meanwhile, in-class exams don't allow time for the students to write and test sections of code or otherwise apply a variety of their skills and knowledge to explore and explain a complex topic. To allow for deeper assessments, here, we design exams with take-home and oral portions, based on successful implementation of a similar strategy in the first-year engineering curriculum. This format provides time for students to think through and work out the assigned problems at home, while also incentivizing them to understand the material well enough to defend it in the subsequent in-person oral exam. When switching to this format from in-class or fully take-home assessments, one crucial new challenge is in designing an oral assessment portion that can be fairly and consistently implemented at scale (in this case, with the help of teaching assistants).

The current paper details the design of the written and oral portions of the assessments and the strategy to implement them in the Spring semester 2026. The oral examination questions and format are created with special focus on 1. making them approachable for students who may not have experience with oral exams (for example, by having brief one-on-one discussions during computer lab time) and 2. maintaining fairness by using strict, detailed rubrics that are well aligned to course goals and can be applied consistently by different proctors. Example assignments, rubrics, training materials for teaching assistants, and logistics are reviewed and contrasted with typical prior assessments. The plans to assess the effectiveness of these strategies in future work are also discussed.

Introduction and Background

The rise of the use of large language models (LLMs) has added additional complexity to the perennial challenge of accurately assessing student skills and understanding, especially in areas such as scientific programming where LLMs can effectively complete common tasks for students [1, 2, 3, 4]. In light of these recent developments, we propose using structured oral exams as an assessment mechanism to promote fairness and effectiveness in assessing higher level skills on computers. Specifically, in a materials science and engineering computational laboratory course, we replaced in-class written examinations with two-part examinations that include a take-home portion and subsequent interactive oral portion, with concomitant changes in the course format to provide students practice in oral responses before the exam.

In this computational laboratory course, we aim to teach practical skills in applying statistical and numerical methods to simulate and analyze materials on the computer (see the Course Details section below). This places our goals in the middle levels of the revised Bloom's taxonomy, a commonly used framework to educational objectives [5]. Specifically, in the knowledge dimension (with categories factual, conceptual, procedural, and metacognitive), we focus on conceptual and procedural knowledge, and in the cognitive process dimension (with categories remember, understand, apply, analyze, evaluate, and create), we focus on applying, analyzing, and evaluating. Thus, assessments of these skills have relied on students being able to use their computers to complete complex tasks and evaluate their results. The computer-based assessments allow for more realistic multi-part application and analysis tasks than would be possible in a paper examination, such as writing a portion of code and then running it to analyze relevant material properties from a specific data set (with the student having the ability to adjust the code if they recognize it is not working as intended).

Methodology

We want to continue assessing higher-level skills by allowing computer use during exams, however, LLMs are readily available on computers, including within software that would otherwise be used on the exam (MATLAB), making it difficult to enforce rules against their use. Meanwhile, allowing students to use LLMs in a standard written exam would make it unclear the extent to which students have met the educational objectives in terms of their own understanding and skills. Importantly, our learning objectives do not require that LLMs not be used, but that the students can apply conceptual and procedural knowledge to perform computational tasks as well as analyze and critically evaluate results from calculations; in general, these processes could include the judicious use of LLMs.

Thus, to focus on fair and accurate evaluation of higher-level cognitive skills, we have added interactive oral questioning into our exam process. Exams now include two parts: a take-home portion that requires code to be written/run and the results to be analyzed, and an interactive individual oral examination portion in which students are required to explain their take-home portion results as well as apply further analysis and evaluation skills to those problems. Students have access to their computers in both portions of the exam. The take-home portion questions are similar in format to prior in-class exam questions, but without the short answer questions that focused on evaluating the results. Instead, the students' analysis and evaluation abilities are assessed primarily through the oral portion.

Each of the two exam parts brings its own advantages and challenges. An important benefit of the untimed take-home portion is that it removes some of the pressure that could occur in class exams, which is especially relevant in a practical programming exam, where a typo or other coding issue might cause a student's entire program to not execute. This is especially useful since students come into the course with a diversity of prior programming experiences and preferences in using LLMs or other resources. With the take home portion, they can use their preferred resources. The obvious potential disadvantage is that students will over rely on such resources, which is why the oral portion is crucial as they will need to be able to explain and evaluate the results in the oral portion based on their experiences in the take home portion. This process more directly maps to real-world programming tasks that are part of a larger project. In the oral part of the exam, the proctor can ask pertinent questions about the student's thinking based on their current results and can ask follow-up questions if incomplete answers are provided. A major benefit is that the proctor can then provide further opportunities for students to demonstrate their higher-level thinking even if they exhibit a weakness in understanding an earlier part of the content (this is an advantage over written examinations, as it is difficult to accurately assess student skills from incomplete written answers). In this way, the accuracy of assessing the higher-level skills can be increased, while the concerns about student cheating with respect to LLM can be reduced (students will be allowed to use LLMs during the take-home portion).

In terms of new challenges of the oral portion of the exam, the flexibility to ask follow-up questions presents a concern regarding fairness across different students. This concern is also exacerbated for large classes where the additional instructor time required to proctor oral exams is prohibitive and co-instructors or teaching assistants must be used to proctor exams. Questions have been carefully designed with this in mind, and a clear set of proctor guidelines were developed alongside the questions that explain when and how to provide follow up information and additional questioning. Example exam questions and the related rubric are provided below. Furthermore, the rubric on scoring the responses in light of these guidelines must be robust to fairly evaluate a variety of different types of student responses and misunderstandings based on their demonstration of their abilities within the learning objectives. All proctors were trained in these materials and observed the instructor in performing examinations before proctoring their own exams.

Finally, a major new challenge from the student perspective is the potential stress or anxiety in being able to answer questions in this oral exam format which is different from many of their prior exam experiences. To mitigate this challenge, we communicated clearly and often about what to expect with this new format and provided smaller opportunities to orally discuss student work with instructors before the oral exams. Specifically, we included a “check-out” process for each lab period in which students briefly described an aspect of their work to the instructor or teaching assistant for a completion grade. For example, during a lab which included numerical integration, students calculated the work to inflate a balloon; the check-out question for this lab was, “How do you know your answer (7J) is reasonable? Why not a number 100 times larger or 100 times smaller?” which is similar to a question on the oral part of the exams, below. Additionally, the week of the oral exam, the instructor and teaching assistants modeled a mock exam in front of the class showing what is expected, and what both a strong and weak performance on the oral part of the exam looks like.

Course Details

The course considered in this work is a statistics and numerical methods programming course using MATLAB for applications in materials science/engineering. Typically, the course is taught in spring semester with an enrollment of 50-60 students (53 in SP26). Students are split into two sections for labs, which meet 110 minutes once per week, with a combined 55-minute lecture session once per week for the entire class. Generally, topics (see list below) are introduced first through assigned background reading, then discussed in the lecture, then applied in the in-person programming lab.

List of topics

First half (exam 1 material): MATLAB Review, Probability Basics, Probability Distributions, Statistics and Hypothesis Testing, Curve Fitting, LASSO Regression

Second half (exam 2 material): Linear Algebra and Transformations, Crystallography + Symmetry (using linear algebra / matrices), Crystallographic Calculations (using the metric tensor), Symbolic Calculations, Numerical Calculations (find roots, derivatives, integrals), Ordinary Differential Equations (overall technique, and using ode45)

Take-Home Exam

The following example is one of three problems of the take-home coding portion of the exam. This example is from the first exam and requires students to apply what they learned in the previous weeks to a new problem using MATLAB. Students have a week to complete the take home portion.

Example Problem

> A study is performed on titanium specimens, some of which contained fatigue cracks generated by a three-point bending process.
>
> The cracks in the samples are analyzed and found to have length (in mm) as follows:
>
> `crack_length = [`*`provided vector of 37 numerical values`*`]`
>
> a) Determine if the data is normally distributed.
>
> *Answer:*
>
> Regardless of your actual answer to part (a), you may assume that the data is normal for parts (b) and (c).
>
> b) You wish to know if the data is consistent with an average crack size of 3.7mm. Write null and alternative hypotheses for this:
>
> *Null:*
>
> *Alternative:*
>
> c) Perform the appropriate statistical test to determine if you should accept or reject the null hypothesis with a 90% confidence level.
>
> *Answer:*

Oral Exam

During the lab period the week of the exam, the instructor and a team of two teaching assistants (TAs) administered oral exams with students in one half of a divided classroom while the students not taking the exam waited in the other half of the room. Each student had an average of 8 minutes for the oral part of the exam, and each student was asked about one of the three exam problems, chosen randomly.

For the randomly chosen problem, the proctor (the instructor or a TA) would ask three questions. The general form of these questions (and the rubric for how they were graded) were provided to the students two weeks before the first oral exam. The questions were:

1. **Explain in simple terms**: In layperson's terms, what problem are you solving and what did you do to solve it?
2. **Evaluate result**: Explain how you know your answer is reasonable, or if you think your answer is wrong, explain how you know that.
3. **Modify code**: Modify your code to do [*something*].

In practice, questions 2 and 3 were specific to the problem the student was being asked about. If a student were asked about the example problem above, the proctors were provided with options that they could ask (these options were not provided to students):

Explain result:

- In part (a), how do you know the data is normal (or not normal)?
- In part (c), how can you tell that your test result is reasonable?

Modify code:

- Modify your test to use a 99% confidence level. What is the result now?
- Modify your test to be a right-tailed test. What is the result now?
- Modify your test to be a left-tailed test. What is the result now?

To train the TAs in performing the exams, the rubric, the questions, and expectations for each question were discussed in detail during an instructor-TA meeting. The TAs were encouraged to ask follow-up questions to students in a clear way that does not imply they are 'trick' questions and focus on getting a fair assessment of student knowledge. They were told to move on as soon as they could fairly place the student on the rubric, and to point students in the right direction without giving away the answer if they were stuck. In future years, role-playing practice will be incorporated into this training.

The rubric was available to students a week before they had to take the exam. In class on the review day (the Monday lecture the week of the exam, where the exam was in lab sections later in the week), the instructor and a TA demonstrated what a "10 Point" response would look like in practice, where the instructor acted as the proctor and the TA acted as the student. The rubrics for Exam 1 and Exam 2, the latter of which was revised for clarity, are provided on the following pages.

Student Name:____________________________ Proctor: ______________________________

MSE 2321 SP26 Exam 1 Oral-portion Rubric

Question:	**15 points**	**10 Points**	**5 Points**	**0 Points**	**Score**
Prepared	N/A	Completed all problems in take-home portion. AND When asking conceptual questions, looked at notes sparingly. Did not rely on them.	Take-home portion incomplete OR Student is unable to answer conceptual questions without using notes (or unable to answer questions at all)	Take-home portion incomplete AND Student is unable to answer conceptual questions without using notes (or unable to answer questions at all)	
Explain in simple terms	Shows a deep/robust understanding of topic, able to explain correctly in simple terms	Shows a limited understanding of topic, able to explain correctly but not always in simple terms	Shows a superficial understanding of topic, explanation is not always correct, and not always in simple terms	Shows no understanding of topic, explanation is wholly incorrect	
Explain result	Supports result with sufficient evidence that is both accurate and relevant AND if incorrect, their code must still attempt the whole problem	Supports result with evidence that is *mostly* relevant or accurate OR presents *limited* evidence without prompting AND if incorrect, their code must still attempt most of the problem	Supports result with evidence that is *inaccurate* or *irrelevant* but corrects when prompted OR does not present *any evidence* unless prompted AND if incorrect, their code must still attempt most of the problem	Presents only *inaccurate* or *irrelevant evidence*, even when prompted OR does not present *any evidence* OR code is totally incorrect (there is little or nothing to support)	
Modify code	Modifies code to accomplish task with little to no prompting	Has a reasonable strategy to accomplish task, but prompting needed to get it to work	Has a reasonable strategy to accomplish task, but is unable to complete it even with prompting	Does not know how to complete task	
					Total:

Student Name:___________________________ Proctor: ____________________________

MSE 2321 SP26 Exam 2 Oral-portion Rubric

Question:	**15 points**	**10 Points**	**5 Points**	**0 Points**	**Score**
Prep.	All three of: (1) Completed all take-home problems (2) looked at notes/code sparingly (3) required minimal prompting	Two of: (1) Completed all take-home problems (2) looked at notes/code sparingly (3) required minimal prompting	One of: (1) Completed all take-home problems (2) looked at notes/code sparingly (3) required minimal prompting	None of: (1) Completed all take-home problems (2) looked at notes/code sparingly (3) required minimal prompting	
Explain in simple terms	Explanation is all of: (1) correct (2) comprehensive but concise (3) in simple terms	Explanation has significant deficiencies in one of: (1) correct (2) comprehensive but concise (3) in simple terms	Explanation has significant deficiencies in two of: (1) correct (2) comprehensive but concise (3) in simple terms	Unable to explain in simple terms or otherwise	
Evaluate Result	Supports result with sufficient evidence that is both accurate and relevant AND if incorrect, their code must still attempt the whole problem	Supports result with evidence that is *mostly* relevant or accurate AND if incorrect, their code must still attempt most of the problem	Supports result with evidence that is *inaccurate* or *irrelevant* but corrects when prompted AND if incorrect, their code must still attempt most of the problem	Presents only *inaccurate* or *irrelevant evidence*, even when prompted OR does not present *any evidence* OR code is totally incorrect (there is little or nothing to support)	
Modify Code	Modifies code to accomplish task	Has a reasonable and detailed strategy to accomplish task, but unable to get it working in time	Has a general idea of how to accomplish task, but unable to get it working in time	Does not know how to complete task	
					Total:

Preliminary Results and Future Work

This new exam format was implemented in spring semester 2026 (ongoing as of this writing). So far, students seemed to adapt well and generally like the exam format. Specifically, in an informal mid-semester feedback form administered the week after the first exam, most comments mentioning the exam were positive.

The main lesson learned was that the design of the rubric is a crucial factor in implementation of oral exams. The rubric must be carefully written with sufficient detail to ensure the correct skills are fully assessed and are assessed accurately. For example, the description of full credit for the "Explain in simple terms" question was "Shows a deep/robust understanding of topic" for the Exam 1 rubric, which is not detailed enough because it could be hard to identify what this looks like in practice. For this question, the best response should be accurate, in simple terms, complete, and concise. The latter two items were not mentioned in the original rubric for Exam 1 and are made explicit in the revised rubric for Exam 2. An updated rubric (on the previous page) was used for the second exam to address these issues.

Another issue related to rubric design was that different proctors graded differently. For Exam 1, students were randomly sorted to be graded either by the instructor or by a team of two TAs, but the initial grades given by the instructor were significantly lower than the ones given by the TAs (t-test $p = 1.6 \times 10^{-5}$). Also, the TAs graded differently than each other but were overall more aligned (ANOVA $p = 0.012$). This was remedied by increasing the scores of students graded by the instructor to have the same mean as the average score given by the TAs.

In future offerings, we will continue to revise the rubric to be more specific and actionable and make the TA training more detailed. Specifically, the instructor will go through the improved rubric in greater detail with the TAs and will have TAs practice with example responses and role-play with the instructor. Additionally, in future semesters, the strategy will be assessed in a more robust manner.

References

[1] B. Chen, C. M. Lewis, M. West, and C. Zilles, "Plagiarism in the Age of Generative AI: Cheating Method Change and Learning Loss in an Intro to CS Course," in *Proceedings of the Eleventh ACM Conference on Learning @ Scale*, in L@S '24. New York, NY, USA: Association for Computing Machinery, Jul. 2024, pp. 75–85. doi: 10.1145/3657604.3662046.

[2] Z. Stephenson, N. Johnson-Glauch, and S. Cruchley, “Interventions and facilitators of oral assessment performance in higher education: a systematic review,” *Assess. Eval. High. Educ.*, vol. 50, no. 7, pp. 1140–1153, Oct. 2025, doi: 10.1080/02602938.2025.2504621.

[3] D. R. E. Cotton, P. A. Cotton, and J. R. Shipway, “Chatting and cheating: Ensuring academic integrity in the era of ChatGPT,” *Innov. Educ. Teach. Int.*, vol. 61, no. 2, pp. 228–239, Mar. 2024, doi: 10.1080/14703297.2023.2190148.

[4] P. Eachempati, R. Komattil, and A. Arakala, “Should oral examination be reimagined in the era of AI?,” *Adv. Physiol. Educ.*, Feb. 2025, doi: 10.1152/advan.00191.2024.

[5] D. R. Krathwohl, “A Revision of Bloom’s Taxonomy: An Overview,” *Theory Pract.*, vol. 41, no. 4, pp. 212–218, Nov. 2002, doi: 10.1207/s15430421tip4104_2.