

# Communicating scientific ideas: tutorials for professionally-styled laboratory reports

Kelley D. Sullivan<sup>1</sup>

<sup>1</sup>*Department of Physics and Astronomy, Ithaca College, 953 Danby Rd., Ithaca, NY, USA*

Written communication of scientific ideas is an important skill often honed through report-writing in physics laboratory courses. I present a series of tutorials that I developed to support students in learning to produce well-written and professionally-formatted laboratory reports. The tutorials include typesetting in L<sup>A</sup>T<sub>E</sub>X (with BibT<sub>E</sub>X), graphing in MATLAB, preparing diagrams using Inkscape, and writing abstracts, introductions, and conclusions. I evaluated student reports and found no statistical difference in the quality of reports written before or after the tutorials were introduced. Responses on end-of-course questionnaires, however, show that students overwhelmingly favor the tutorials and find them helpful in supporting their learning.

## I. INTRODUCTION

Advanced Physics Laboratory ("Advanced Lab") is a junior-level laboratory course at Ithaca College that challenges students to work with increasing independence to complete five to six advanced laboratory experiments. Six class periods in the first half of the semester are also devoted to supporting students in learning error analysis. Students write up the results of their experiments, including error analysis, in professionally-formatted reports styled after articles in the *American Journal of Physics* (*AJP*). Students are encouraged to use a variety of software to prepare their reports: L<sup>A</sup>T<sub>E</sub>X for typesetting, MATLAB for data analysis and graphing, and Inkscape for drawing diagrams. Our emphasis on professionally-styled scientific writing is intentional. Strong writing skills are highly marketable across a variety of job sectors into which physics majors venture after graduation [1, 2]. The AAPT recommends that students write lab reports that are authentic in style and format and that graphs and diagrams be used to present data (including uncertainty) and describe theoretical concepts [3]. This is echoed in the literature [4], where there is also general consensus that student scientific writing lacks clarity and coherence [5]. Students use the writing skills they gain in Advanced Lab in other courses at Ithaca College and when they write their senior theses, and several of our graduates have used reports from Advanced Lab in successful job applications.

The course outline I inherited for Advanced Lab did not include instruction on the recommended software or on report-writing. Students were provided with a L<sup>A</sup>T<sub>E</sub>X article template and were supported in learning all software and improving their writing during office hours and through instructor feedback on reports. I noticed that most students were eager to use the recommended software but struggled to gain proficiency in a timely fashion while also focusing on learning error analysis, completing their experiments, and writing their reports. Some students also balked at the "trial-by-fire" method of submitting their first report with a limited description of the expected written content and then improving via response to instructor feedback. To address these concerns, I developed a series of tutorials to facilitate student learning of the software fundamentals and to provide students with greater detail re-

garding my expectations for the content and style of several of the lab report sections. The writing tutorials further harness the power of peer-evaluation to improve writing skills [6]. Each tutorial requires about 30 min of in-class time and 30-60 min of out-of-class time to complete. To make time in class, I reduced instruction on error analysis, which students had expressed was largely superfluous given the clarity of the textbook [7]. I also reduced the amount of error analysis homework. I have not observed a decline in student understanding of error analysis as a result of these modifications.

I introduced the tutorials into Advanced Lab in spring 2013 to rave reviews from the students. On anonymous end-of-course questionnaires, five out of seven students chose the tutorials as the "one thing that should not change in the course." I present here a description of the tutorials and their implementation. I report on the impact of the tutorials on the quality of student work. I also provide an analysis of student questionnaire responses to demonstrate the sustained positive student perception of the value and effectiveness of the tutorials and the associated report-writing skills.

## II. SOFTWARE TUTORIALS

Each piece of software is introduced with a 10 min presentation of the user interface and demonstration of how to use the software to create text, graphs, or diagrams. I project my laptop screen for the class to see and the students follow along on their own laptops or classroom computers. The students then have 15-20 min to begin work on the tutorial, which is accompanied by written instructions and a tip sheet. The students are encouraged to use friends and online resources, as necessary, to help them complete each tutorial for homework and to expand their skills over the course of the semester.

**L<sup>A</sup>T<sub>E</sub>X:** I begin with the L<sup>A</sup>T<sub>E</sub>X tutorial, which is built around a completely redesigned template that includes brief descriptions of each required report component under appropriate section headings. Sample equations, a table, and commands for figures and a bibliography are also included. I recommend TexMaker [8] as an editor/compiler because it is free and cross-platform, but students are welcomed to use any editor of their choosing.

I use the template to demonstrate the structure of a  $\LaTeX$  document including the document class, preamble with common packages, and the main body of the document with sectioning. I highlight examples of inline and display equations and tables, as well as describe how referencing works. Finally, I show the students how to change common user preferences in TexMaker and how to compile their document.

The tutorial guides students through editing the  $\LaTeX$  template to match a provided document. The text edits emphasize common formatting elements including left quotation marks, bold and italic emphasis, and special characters. The students then replace kinematics equations with an electromagnetism derivation involving integrals, vectors, expanding parentheses, and variables with superscripts and subscripts. Finally, the students are asked to fill in missing detail from a table including numbers in scientific notation and units. Students must also supply labels for equations and the table and reference them appropriately in the text.

**Inkscape:** I begin the tour of Inkscape by showing students how to set document preferences. I then draw a few shapes, lines, and curves using the Bezier tool and add text including using unicode to make a Greek letter. I open the fill and stroke menu to show students how to create arrow heads and set fill and stroke colors. Finally, I show students how to save drawings or selected objects as .eps files. I then return to the  $\LaTeX$  template and we discuss how to incorporate figures into the document, including how to size figures and how to properly reference and caption them.

The tutorial provides instruction to help students reproduce a landscape using common shapes, curves, arrows, and text labels. The students are then tasked with creating an equipment diagram for their ongoing lab and adding their image, with a caption and reference, to the  $\LaTeX$  template. Having the students draw a diagram for their ongoing lab effectively reduces the tutorial workload and offers students the opportunity to receive feedback on their diagram before the first lab is due.

**MATLAB:** Physics students are introduced to plotting in MATLAB in our required sophomore-level lab course. Therein, students are taught how to write a script to define data arrays and plot their data. I expand upon this introduction and show students how to import data using the wizard and plot using a script that calls data from the workspace. I then open the  $\LaTeX$  template and show how the MATLAB default plot settings result in a hard-to-read figure in the small space offered by a two-column report. I return to MATLAB and introduce the figure GUI as a visual tool to modify their plots for better readability.

For the MATLAB tutorial exercise, students take a data set (provided in advance), import it through the wizard, plot the data and min/max fit lines using a script, then adjust the graph using the figure GUI to produce an easy-to-read figure in the  $\LaTeX$  template report. The students must also label and reference their figure and provide an appropriate caption.

**BibTeX:** I demonstrate BibTeX by creating a bibliography file in real time using the TexMaker bibliography wizard

and by cutting and pasting bib entries from an online journal database search. I return one last time to the  $\LaTeX$  template and show students how to create citations in the text and which commands set the bibliography style and call the .bib document. For the tutorial exercise, students cite a provided book, journal article, and website in the  $\LaTeX$  template. The students must also create a bibliography for their ongoing lab.

### III. WRITING TUTORIALS

The writing tutorials provide detailed guidance for writing conclusions, abstracts, and introductions. I chose to focus on these report sections because my experience showed that our students struggle most with their content compared to the content of the theory, experiment, and results sections. I also changed the report requirements such that students need to include only the goals of their experiment plus theory, experiment, and results sections in the first report of the semester. The students include the remaining sections in future reports once the associated tutorials are complete.

The writing tutorials are completed entirely in class. The conclusion and abstract tutorials begin with a discussion of the detailed description and scoring rubric that I devised for each report section. I then pass around copies of former student work for the students to read and score per the rubric. The examples are drawn from reports on a single laboratory and include work that I consider to be fair, good, and excellent quality in terms of the content and writing style. I collect the students' scores on the board and we discuss the negative and positive aspects of each example.

The introduction tutorial is similar except that I begin by having the students read introductions from two *AJP* articles, then I ask the students what type of information each (very different) introduction conveys. Invariably, their answers mirror the section description and rubric that I then pass around for them to use to score a set of student-written introductions.

Critiquing the work of their peers provides an opportunity for the students to recognize what common errors and omissions students make when writing their reports. The students are also empowered by the examples of excellent peer writing. The students' scores for the example report sections are typically in line with each other's scores and with mine. This impresses upon the students that they have the ability to appropriately evaluate the quality of their own writing.

### IV. STUDENT QUESTIONNAIRES

For the first four semesters in which the tutorials were implemented, I included questions on anonymous end-of-course questionnaires to solicit feedback from the students regarding their experience with the tutorials. Separate prompts provided an opportunity for students to "comment on the effectiveness" of each software tutorial individually and of the writing tutorials as a group. I also asked students to offer "additional con-

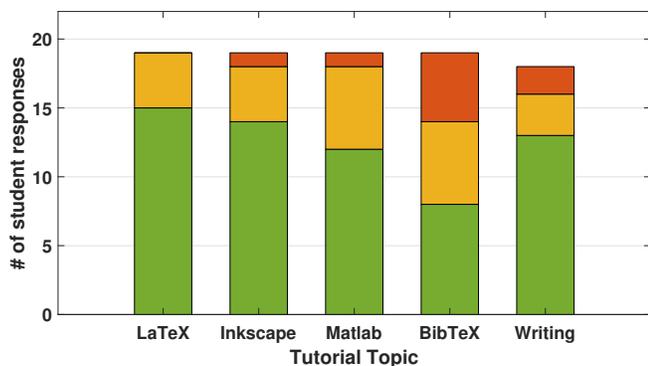


FIG. 1. The number of students responding favorably (bottom), neutrally (middle), or unfavorably (top) to questions about the tutorials on anonymous end-of-course questionnaires ( $N = 21$  total respondents). Students overwhelmingly found the tutorials to be useful in helping them learn report-writing skills.

structive suggestions for improvement," which I have used along with my own observations to edit the tutorials annually.

Twenty-one students filled out questionnaires during this time period. I collected their responses and categorized them as favorable, neutral, or unfavorable. Neutral responses included ambiguous comments, such as "No complaints here." and non-answers, such as "N/A – already familiar with L<sup>A</sup>T<sub>E</sub>X." Two responses were left blank for each of the software tutorial prompts and three for the writing tutorial prompt. The blank responses were removed from the analysis. The quantification of responses for each tutorial is shown in Fig. 1.

The response from the students was overwhelmingly positive. The L<sup>A</sup>T<sub>E</sub>X tutorial was the most well-received, with 15 students (79%) responding favorably and no students responding unfavorably. The students felt that the tutorial was "an effective way to get a good understanding for a beginner."; "Perfect. Set me up to write reports successfully."; and "Great. Easy to follow."

For the Inkscape tutorial, 14 students (74%) responded favorably and 1 student (5%) responded unfavorably. Students again found this tutorial "Really helped!! It was also very fun." and "Taught skills needed for the course." The unfavorable comment focused on what one student felt was a difference in expectations between the tutorial (more strict) and the lab reports (more lenient). I used this, and other constructive suggestions, such as "I wish more class time was dedicated to the Bezier tool." to improve the tutorial over time.

Regarding the Matlab tutorial, 12 students (63%) left favorable comments and 1 student (5%) left an unfavorable comment. Students found the tutorial "Soo [*sic*] useful!" and claimed it "taught me things I didn't know!" One student asked, "Why did nobody teach me the plot editor before?" The one unfavorable comment astutely summed up the challenge with using MATLAB in this course in general: "I don't think the coverage was comprehensive enough to be useful. The trouble is that students are all at different places with an understanding of MATLAB." The MATLAB tutorial is in-

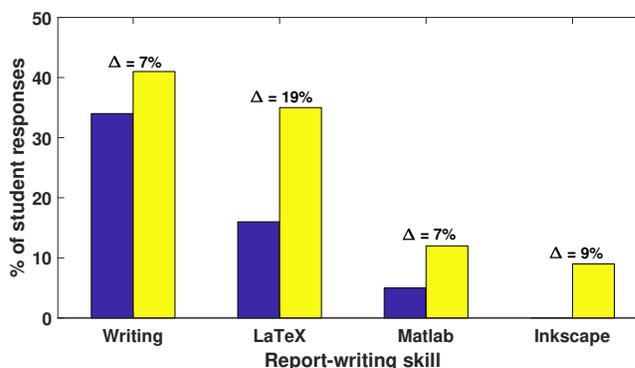


FIG. 2. The percent of student responses on anonymous end-of-course questionnaires from classes before the tutorials were introduced (left) and after (right) that included scientific writing, L<sup>A</sup>T<sub>E</sub>X, MATLAB, or Inkscape under the question prompt "What is the most significant/useful thing you learned in this course?"

tended to help level the playing field by re-introducing basic graphing skills, however some of our students require support beyond the tutorial.

The BibTeX tutorial received the most mixed reviews, with 8 students (42%) responding favorably and 5 students (26%) unfavorably. The large number of neutral responses (6) points to the fact that several students found this tutorial "very simple, almost unnecessary." Revealing the dichotomy of student comfort with programming, another student commented, "If you want us to use bibtex, then we need much more instruction on it."

The students found the writing tutorials to be helpful also, with 13 students (72%) responding favorably and 2 students (11%) unfavorably. One student "liked that past student examples were used as it gave a feeling for how the sections should be written and how other students approach writing them." Another student remarked, "I thought the writing tutorials helped me when writing lab reports. I also learned how I would be graded, which I think is very fair!"

In addition to the comments left under the tutorial-specific prompts, comments regarding the tutorials and associated report-writing skills were also left spontaneously under other prompts. The responses to one question in particular stood out: "What was the most significant/useful thing you learned in this course?" I gathered student responses from 6 sections of the course taught before tutorials were added ( $N = 44$ ) and from 7 sections after the tutorials were added ( $N = 34$ ) and calculated the percent of responses that included writing, L<sup>A</sup>T<sub>E</sub>X, MATLAB, or Inkscape. The results, displayed in Fig. 2, show significant increases across all categories after the tutorials were introduced. The evidence shows that students not only find the tutorials effective for learning report-writing skills, but also since the introduction of the tutorials a larger number of students feel that the individual skills themselves are significant.

## V. LAB REPORT QUALITY

I evaluated student reports written in semesters before and after the tutorials were introduced to determine if the tutorials were having any effect on the quality of student work. I focused on individual report elements that reflect a subset of the skills taught in the tutorials: data plots (MATLAB), equipment and theory diagrams (Inkscape), and abstracts (writing). The elements were extracted from the last or second-to-last report written by different random samples of students enrolled in one of the 8 sections of Advanced Lab offered before ("pre") or 8 sections offered after ("post") the tutorials were introduced.

I created rubrics to evaluate the data plots and the equipment and theory diagrams using criteria derived from the MATLAB and Inkscape tutorials, respectively. Each criterion was rated on a 1-3 scale and included basic aesthetic elements such as font sizes, line widths, and sizing. Any criteria that were not applicable were removed from the total score. If a report had more than one plot or diagram then all plots or diagrams were evaluated and an average score was recorded for that report. I scored the abstracts using the rubric from the abstract tutorial. The mean, standard deviation, and median were calculated for each set of elements pre- and post-tutorials and a Student's t-test was used to determine if the two populations had statistically different means.

The population statistics for the data plots were nearly identical pre- and post-tutorials, with the average, standard deviation, and median changing from 79, 9, and 79 to 80, 8, and 79. The statistics for the equipment and theory drawings also changed little, with values of 91, 9, and 93 pre-tutorials and 93, 8, and 97 post-tutorials. The results for the abstracts showed much greater changes, with values of 61, 14, and 61 pre-tutorials and 68, 20, and 73 post-tutorials. The differences in means for all elements pre- and post-tutorials, however, were not statistically significant – data plots ( $p = 0.82$ ), diagrams ( $p = 0.47$ ), and abstracts ( $p = 0.36$ ).

These null results are not wholly surprising. Many forces govern the quality of student work, including the work ethic of each student and the feedback provided by the instructor. Motivated students taking the class before the tutorials were introduced would have put in the effort to learn how to use the recommended software and may have passed their learn-

ing on to their peers. Moreover, the report elements I evaluated were drawn from the last or second-to-last reports of the semester. By this time, the students would have received instructor feedback on three or four previously-submitted reports and improved their figures and writing in response. This suggests that instructor feedback alone is an effective, if less efficient and supportive, method to help students improve their report-writing skills.

## VI. CONCLUSION

The use of software and writing tutorials in our Advanced Laboratory course makes students feel supported in learning the skills necessary to produce well-written and professionally-styled laboratory reports. Although the quality of student work has not improved with the implementation of the tutorials, students overwhelmingly find the tutorials helpful and appreciate being provided with instruction to jump-start their learning. A recent student wrote on their end-of-course questionnaire, "Having tutorials is definitely useful as I don't think anyone likes just being thrown in the deep end." Even students who come to class familiar with the software presented in the tutorials admit they "learned shortcuts & other things."

The tutorials require a total of about 3.5 hours of in-class time and approximately 5-6 hours total of out-of-class time to complete. Anecdotal evidence suggests that the loss of out-of-class time is mitigated by the fact that students gain proficiency with the software more rapidly with the support of the tutorials, which also include exercises that provide content for their ongoing labs. Nonetheless, minor adjustments to course structure are necessary to incorporate the tutorials.

Use of the tutorials has proliferated to other courses in our department, including to Senior Thesis and Senior Project. The tutorials were also adapted by a colleague in computer science for a course focused on reading and writing technical reports. This colleague found "a huge improvement in the visual and organizational quality of student technical reports compared to the previous offering" in which the tutorials and L<sup>A</sup>T<sub>E</sub>X template were not used. Moreover, the students' "overwhelming response to the tutorials was that they made L<sup>A</sup>T<sub>E</sub>X easy to understand and use." The tutorials are available on Compadre at [www.compadre.org/portal/items/detail.cfm?ID=14705](http://www.compadre.org/portal/items/detail.cfm?ID=14705).

- 
- [1] R. Czujko, K. Redmond, T. Sauncy, and T. Olsen, Tech. Rep., AIP Statistical Research Center (2014).
- [2] P. Mulvey and J. Pold, Tech. Rep., AIP Statistical Research Center (2017).
- [3] J. Kozminski, N. Beverly, D. Deardorff, R. Dietz, M. Eblen-Zayas, R. Hobbs, H. Lewandowski, S. Lindaas, A. Reagan, R. Tagg, et al., Tech. Rep., American Association of Physics Teachers (2014).
- [4] B. Zwickl, N. Finkelstein, and H. J. Lewandowski, in *Physics Education Research Conference 2011* (Omaha, Nebraska, 2011), vol. 1413 of *PER Conference*, pp. 391–394.
- [5] C. Haagen-Schuetzenhoefer, *The Physics Teacher* **50**, 430 (2012).
- [6] D. Nicol, A. Thomson, and C. Breslin, *Assessment & Evaluation in Higher Education* **39**, 102 (2014).
- [7] J. R. Taylor, *An Introduction to Error Analysis* (University Science Books, 1997), 2nd ed.
- [8] P. Brachet, <http://www.xmlmath.net/texmaker/doc.html>, Free Software Foundation (2003–2018).