SAGE-LY SOLAR: ONLINE MATH TOOLS TO TEACH SOLAR ENERGY

Joseph Ranalli
Department of Engineering
Penn State Hazleton
76 University Drive
Hazelton, PA 18202
jar339@psu.edu

Jeffrey R. S. Brownson
Department of Energy & Mineral Engineering
The Pennsylvania State University
263 MRL Building
University Park, PA 16802
nanomech@psu.edu

ABSTRACT

Solar energy education at the University level is sparse, but student interest is driving new classes to be developed and taught. Solar energy is a calculation-intense, multiparameter systems field that relies on numerical tools. In the scaling up solar energy education to meet large numbers of interested students in science, engineering, and economics, we choose to develop numerical tools in the Python language to supplement courses for the new generation of young professionals. We demonstrate the fundamentals of assessing radiative transfer from the Sun, spherical trigonometry, and energy economics using open source software. We find that the core tools in solar energy project development can be explored in the classroom or remotely online using the Sage (System for Algebra and Geometry Experimentation) math suite available as an online server. The Sage mathematical software has capabilities to solve multiparameter problems, interactive plotting, and allows students to develop formalized *notebooks* to demonstrate a flow of work towards a final set of solutions. Integrated with Python, it can also be extended with additional mathematical libraries as needed. The Sage server can be installed on a University server and notebooks can be shared among students and faculty. Additionally, working in Sage allows students to develop basic programming skills that will be essential to future solar software development. The use of an online tool allows students to work on home computer devices or in computer labs on campus, makes the problem solving independent of major operating systems, and removes the barrier of purchasing proprietary software.

1. INTRODUCTION

With the exponential growth and expansion of solar technologies, deployed as photovoltaics, concentrating solar power, and solar hot water/air systems, there is a concomitant need for energy education to train design professionals in addition to training installer professionals. Solar energy education is changing, adapting to a growing cohort of students in higher education disciplines of energy engineering, economics, and entrepreneurship. This new cohort seeks the core knowledge to participate in project design, client/stakeholder consulting, and project finance and implementation.

One of our most advantageous efficiency improvements in modern society is common access to the computer. In practice, competitive professionals are called to reduce the number of hours required to accomplish a necessary task while working on a design project for the client. Hence, the longer-term outcome of using software in engineering education is exposing students to useful tools that will extend their employable skill set and create a more desirable employee in a design firm.

In the interest of preparing students for a future in designing solar projects, the course content has been formed around the core calculations of radiative transfer, spherical trigonometry, and energy economics. In the past, the multiparameter problems were addressed using software such as EES (Engineering Equation Solver) or MATLAB as paid-for tools, and Scilab or Octave as open source tools (each uses similar syntax to MATLAB).[1-3] The utility of these softwares is to direct the students focus to the content and learning objectives for the course content, rather than being distracted by the accounting and arithmetic encompassed in non-trivial multiparameter problem solving. While some design problems can be addressed using numerical spreadsheets, the act of programming in ASCII code can facilitate a more transparent record of data processing, such that a team of workers can pass useful information along and clearly demonstrate the repeatable methodology applied to arrive at the solution.

We investigated tools that could reduce the complexity of calculations but still provide interactivity, and that could be made available online to easily scale to a large group of online learners. The tool chosen for this project was Sage (System for Algebra and Geometry Experimentation).[4] Sage is an open-source mathematics suite based on the Python programming language. Sage provides a wide variety of mathematical functionalities, but those most apt for the present situation are symbolic and numerical manipulation, and interactive plotting capability.[5] Sage calculations are performed within notebooks. The notebooks allow information to be presented in the form of formatted HTML text (including LaTeX formatting via jsMath) interspersed with calculation blocks containing the Python/Sage code. Sage notebooks can developed and shared among students and faculty. In this way, presentation within Sage can resemble a sort of living textbook that can perform live calculations to accompany the text. The greatest advantage offered by Sage in this context is the interactive plotting capability that allows users to manipulate a graph by using a variety of common graphical interface widgets (e.g. sliders, text entry boxes) to manipulate the graph input variable values.

The goal of this project was to create a suite of educational tools for solar power that would complement the classroom experience and aid in student independent learning and homework assignments. We have explored preliminary examples with the students of EGEE 437: *Design of Solar Energy Conversion Systems* (26 of 60 total), comparing Sage with an open source MATLAB clone, Scilab. Much like problem solving with MATLAB/Scilab, working in Sage allows students to expand their basic programming skills for numerical problem solving. The use of an online tool allows students to work on home computer devices or in computer labs on campus, makes the problem solving independent of major operating systems, and removes the barrier of purchasing proprietary software.

In this study, we considered tools that can assist students with calculations related to spectral intensity of ideal black-body radiation. A common pedagogical task in our course is calculation of the black body emissive power within a spectral band of interest, requiring integration of Plancks Law between two wavelength bounds. Traditional textbook approaches to these problems utilize tabulated values to compute the integral using a normalized form of the variables (wavelength and temperature).[6] Qualitatively, students were observed to have difficulty performing the calculations using this method. The wide availability of advanced computational tools provides the opportunity for alternative methods that may be more natural to students.

2. <u>METHODS</u> An introductory pair of Sage notebooks were developed to familiarize students with Sage, one demonstrating Sages programming syntax and the second Sages text markup capabilities. In addition to these, a third notebook was then developed, providing a brief technical treatment of black body radiation theory along with several illustrative plots. The technical notebook culminated with an interactive plot able to calculate and visualize the integral of the black body radiation curve, allowing users to specify the temperature and a wavelength integration band (e.g. the visible spectrum). Finally, the notebook provided a text-only version of this calculator for students to simply perform repeated the spectral emissive power calculations.

The students were presented with the Sage notebooks as an alternative option for their first homework assignment. For their homework, students are required to turn in their answers in a report format, which they produced using the LaTeX typesetting software. Two problems on the first homework assignment required students to perform calculations regarding energy emitted from a graybody source within a given spectral band. Students in this course have all previously taken a programming course using MATLAB, or a combination of MATLAB and C++. On the homework assignment, they were required to use MATLAB or SciLab to generate plots and numerical answers as solutions to the two problems on spectral radiation. For these two problems, students were also provided with an opportunity to evaluate the Sage notebooks as an alternative. A 10% bonus point incentive on the assignment score was offered for those students who tested Sage and completed a survey about their experience. Of the 60 registered students, a total of 26 participated in the Sage portion of the assignment.

The effectiveness of the Sage notebooks was evaluated by student self-reported perceptions about using the Sage software on its own merits, and as compared to

MATLAB/SciLab. The questions from the survey are shown below. Questions were written to probe student confidence in solving problems using these methods and whether students felt that they could modify the tools for other problems. A free response section for student comments was also provided. The survey was administered electronically using the ANGEL course management system.

1. You were required to use Scilab or Matlab for the entire Homework. How confident did you feel as a user of Scilab/Matlab? Rate your confidence using Scilab as compared to using pencil, paper, and a hand calculator. [Five level confidence rating scale]
2. You were given the option to use Sage from prepared worksheets with code and explanatory text (specifically the "EGEE 437: PlancksLaw" workbook). How confident did you feel as a user of Sage to address problems 3.5 and 4.2? Rate your confidence using Scilab as compared to using pencil, paper, and a hand calculator. [Five level confidence rating scale]
3. I feel confident in the answer I generated using the worksheet "Sage EGEE 437: PlancksLaw". [Two level rating scale (Agree/Disagree)]
4. Using Sage helped me to generate my report for the final submission. [Two level rating scale (Agree/Disagree)]
5. Using Scilab/Matlab helped me to generate my report for the final submission. [Two level rating scale (Agree/Disagree)]
6. I feel I could modify code to solve other problems using Sage. [Two level rating scale (Agree/Disagree)]
7. I prefer using Sage to doing homework by hand. [Two level rating scale (Agree/Disagree)]
8. I prefer using Scilab/Matlab to doing homework by hand. [Two level rating scale (Agree/Disagree)]
9. I would like to learn more about using Sage for homework. [Two level rating scale (Agree/Disagree)]
10. Given a range of opinion rate your preference for either Scilab/Matlab vs. Sage for answering engineering problems in this homework? [Five level preference rating scale]
11. Did you use additional resources such as the provided "EGEE 437: Programming and Sage.sws," or materials you found on the web, to learn more about Sage while answering the questions? [Four choices: I used the Programming and Sage Tutorial; I used other resources; I used the Programming and Sage tutorial AND other resources; I did not use any additional resources]
12. Which software was easier for you to generate your solution in? [Choice between Matlab/SciLab and Sage]
13. Which software method do you feel gives you more confidence in your solution? [Choice between Matlab/SciLab and Sage]
14. Which software do you think would be easier to generate a solution to a different problem using? [Choice between Matlab/SciLab and Sage]
15. Which software would you rather use in future assignments? [Choice between Matlab/SciLab and Sage]

3. RESULTS & DISCUSSION

Students completed the assignment using both Scilab and Sage, and were required to complete the post-survey to receive the extra credit. The number of responses for all questions was 26. The student response data by question appears in Table 1, and the responses for Questions 1 and 2 are listed in Figure 1.

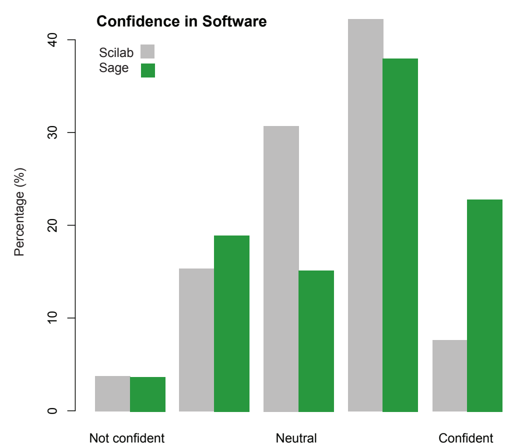

Figure 1: Plot of student reported confidence using Scilab vs. Sage for the homework assignment.

On questions asking for non-exclusive responses about the software tools, students expressed a preference for using both of the computer tools as compared to doing their homework by hand. Students also felt that both software options helped them to complete their assignment. Students expressed confidence in their ability to generate solutions using either tool, with slightly stronger confidence in the Sage tool. In Figure 2 the range of preference demonstrates the slight preference for using Sage, with the exception of 19% of the students who strongly preferred Scilab. This result is further supported by the exclusive responses that asked students to select their preference between the two tools. On each of these questions, students indicated that they preferred the Sage tools over Scilab/Matlab to generate the solutions here and for future problems, were more confident in Sage-generated answers, and would rather use Sage as a tool for future assignments. A strong majority (80%) of students noted that they would be interested in learning more about using Sage for their other homework assignments.

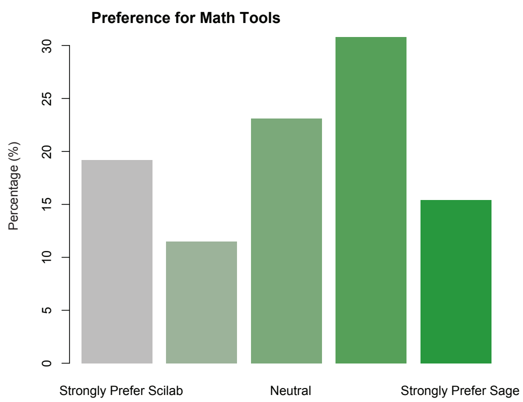

Figure 2: Plot of student reported preference using Scilab vs. Sage for the homework assignment.

In the free response section, many students expressed somewhat mixed feelings between the two software options. An illustrative example is the following quote from one student:

> "I feel that with some work I could get used to Sage and maybe in the future use it for final reports and to solve homework problems. I did find the code somewhat difficult to follow but the output answers were a lot clearer than in Scilab.

Many expressed having developed an interest in Sage and identified several advantages they observed by using Sage. Another common aspect, even among those who enjoyed the experience with Sage, was reservations about their capability to use Sage in the long term, especially as compared to the MATLAB skills that they may have developed in other classes. Several students expressed some confusion about the operation of the Sage documents, which a few attributed to their lack of prior experience.

4. CONCLUSION

Sage is an open-source computational mathematics package that can be accessed via the internet. Educational notebooks were created within Sage to demonstrate mathematics related to radiative heat transfer for junior level engineering students in a course about solar energy conversion systems. Based on student responses, these notebooks were successful in communicating these topics. Sage was viewed favorably by a majority of the students as an aid in completing their homework assignments, and students expressed a desire to continue using Sage in the future. This can be considered a preliminary success relative to the initial project goal of engaging students through use of computer-based calculation tools. We believe that Sage represents an excellent platform for continued development of educational tools. We also consider it to be a viable avenue for helping students to develop an appreciation of the need for professional computational skills.

5. ACKNOWLEDGMENTS

Thanks to the Spring 2013 class of EGEE 437: *Design of Solar Energy Conversion Systems* at the Pennsylvania State University–University Park Campus for their participation in the survey.

6. REFERENCES

(1) H. Fangohr, 2004, "A Comparison of C, MATLAB, and Python as Teaching Languages in Engineering," M. Bubak et al. (Eds.): ICCS 2004, LNCS 3039, pp. 12101217. (2) N. Sharma and M. K. Gobbert, 2010, "A Comparative Evaluation of Matlab, Octave, Freemat, and Scilab for Research and Teaching" Technical Report No. HPCF-2010-7, UMBC High Performance Computing Facility, University of Maryland, Baltimore County. (3) S. A. Klein, 2012, EES: Engineering Equation Solver for Microsoft Windows Operating Systems (4) Open Source Mathematics Software, URL: `http://www.sagemath.org/`, Retrieved on 2/22/2013 (5) B. Eroöcal and W. Stein, 2010, "The Sage Project: Unifying Free Mathematical Software to Create a Viable Alternative to Magma, Maple, Mathematica and MATLAB". Third International Congress on Mathematical Software, Kobe, Japan, September 13-17. Lecture Notes in Computer Science Vol. 6327 (6) M. F. Modest, 2003, *Radiative Heat Transfer*, 2nd. Ed., Academic Press

Table 1: Tabulated responses from each question in the class survey of mathematical tools in solar energy. Responses divided into representative percentages (%).

Question	**Responses**				
	Not Confident	**Somewhat Not Conf.**	**Neutral**	**Somewhat Conf.**	**Confident**
1	3.8	15.4	30.8	42.3	7.7
2	3.8	19.2	15.4	38.5	23.1
	Agree	**Disagree**			
3	76.9	23.1	-	-	-
4	69.2	30.8	-	-	-
5	88.5	11.5	-	-	-
6	80.8	19.2	-	-	-
7	69.2	30.8	-	-	-
8	65.4	34.6	-	-	-
9	80.8	19.2	-	-	-
	Strongly Prefer Scilab	**Prefer Scilab**	**Neutral**	**Prefer Sage**	**Strongly Prefer Sage**
10	19.2	11.5	23.1	30.8	15.4
	Used Tutorials	**Used Other Resources**	**Used Other Resources and Tutorials**	**Did Not Use Any Other Resources**	
11	38.5	15.4	23.1	23.1	-
	Scilab/Matlab	**Sage**			
12	34.6	65.4	-	-	-
13	34.6	65.4	-	-	-
14	42.3	57.7	-	-	-
15	34.6	65.4	-	-	-