International Transaction Journal of Engineering, Management, & Applied Sciences & Technologies

http://www.TuEngr.com, http://go.to/Research

Development of Computer Aided Learning Software for Use in Electric Circuit Analysis

Bradley Deken [a*] **and Chris Cowen** [a]

[a] *Department of Industrial and Engineering Technology, Southeast Missouri State University, USA*

ARTICLEINFO

Article history:
Accepted 26 August 2011
Available online
01 November 2011

Keywords:
Circuit analysis software;
Computer aided learning;
Electric circuit analysis;
Learning software;
SPICE.

ABSTRACT

Presently, instructors are required to teach more students with the same resources, thereby reducing the amount of time instructors have with their students. Because of this, examples may be omitted to be able to make it through all of the required material. This can be problematic with electric circuit analysis courses and other courses used as prerequisites. A lack of understanding in these classes will likely continue in future classes.

While software is often used in these classes, often it is analysis software not meant to teach concepts. Teaching software does exist, but may have only a preset number of problems or only provide the solution. Others provide a 'limitless' number of problems by changing component values, but each ends up being the same basic problem.

This paper introduces new learning software that addresses these shortcomings. The software provides a practically limitless number of problems by varying component values and circuit structure. Moreover, it provides both an answer and an explanation. Finally, it is designed so that students who need more help can get it, while those who do not can move on.

*Corresponding author (B. Deken). Tel/Fax: (573) 651-2659. E-mail addresses: bdeken@semo.edu. 2011. International Transaction Journal of Engineering, Management, & Applied Sciences & Technologies. Volume 2 No.5. (Special Issue). ISSN 2228-9860. eISSN 1906-9642. Online Available at http://TuEngr.com/V02/507-520.pdf

1. Introduction

As an educator in one of today’s classrooms, I am faced with two problems: an ever-growing body of knowledge and an educational system that is moving towards larger classrooms and reduced face-to-face time with students. Electrical circuit analysis courses (herein referred to as circuit courses) are typically relatively large classes and dense in the number of topics that are presented. Ideally, for each topic that is presented several examples would be presented to the class. However, in a traditional lecture environment these are time-consuming. Moreover, since circuit courses are typically entry-level, students may have a wide variety of backgrounds. Therefore, some students may understand topics right away while others may require many different examples. However, because of the number of topics, providing examples for each topic until each student understands is not a possibility. Equally, it is not prudent for students to ‘just get through’ a circuits course since it is often a prerequisite for most of their other courses. Ideally, an instructor or tutor could help any students who get ‘left behind’ by tutoring during their office hours or other help sessions. With the current state of our education system; however, instructors are required to teach more students with the same, or even fewer, resources. Software is often touted as a way to increase understanding outside of the classroom.

While software has its benefits, it is not a panacea for teaching. While software has its place in the classroom, one must keep in mind the purpose. For instance, SPICE and other circuit analyzing software are often used but they are designed to analyze, not teach analysis techniques. While the teaching of this type of software is often justified in circuit courses, the intent should be to learn the software not to teach analysis concepts. As an analogy, consider teaching a child arithmetic only using a calculator. They may learn how to use the calculator, but will neither learn how to perform the arithmetic nor gain insight into how the problems are solved. Other software does exist for the sole purpose of helping students learn the concepts. However, many are limited. For example, some have only a present number of problems or only provide the solution without specifying how it is computed. Others may provide a ‘limitless’ number of problems by changing component values, but each of these end up being the same basic problem.

This paper will introduce a newly developed, computer-aided, learning software that addresses these shortcomings. The software provides a practically limitless number of problems by not only varying component values but also the structure of the circuits. Moreover, the software provides not only an answer to each problem, but an explanation of how the student could compute it. Finally, the software is designed in such a way that students who need more help can get it while those who do not can move on. This software and an analysis of its use in a classroom will be presented in this paper.

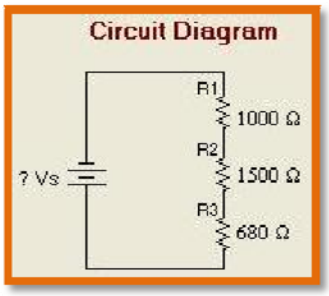
2. Types of Electric Circuit Analysis Software

There are three basic types of software packages used in introductory circuits courses: numerical analysis software, simulation software, and tutoring software. Each has a particular purpose for being included in a circuits course.

Numerical analysis software enables a user to create complex mathematical functions and algorithms, and then solve equations using those algorithms. MATLAB (derived from “Matrix Laboratory”), created by MathWorks, is numerical analysis software designed for engineering and other math-related fields (Attia, 1996). PTC’s MathCAD and the freely available FreeMat are both similar to MATLAB in functionality. Each of these software programs have the ability to perform complex mathematical computations. While these programs simplify the computations, they do have drawbacks. In lower-level electrical courses, many of the formulas used do not become complex enough to require computer-based software to analyze. MATLAB, and similar programs, can calculate diode characteristics, plot transistor response curves, and so much more (Attia, 1996). However, this is often not needed in low-level circuits courses. While learning the software may be an objective of the class, learning the numerical analysis software may not translate into an increased understanding of electric circuit analysis. Thus the implementation of numerical analysis software to teach electric circuit analysis may be impractical.

Simulation programs for electrical circuits enable a user to create a schematic of a circuit on a computer and then run a simulation on the circuit to analyze the behavior. Simulation Program Integrated Circuits Especially (SPICE) is one of the most common simulation engines used for electrical circuits (Chamas and Nokali, 2004). SPICE is a simulation engine developed at the University of California, Berkeley, and serves as a foundation for electronic simulation. Several programs use it, including XSpice (Georgia Technical Research Institute), PSpice

(MicroSim, now packaged by OrCAD), and MultiSim (Electronics Workbench). Simulating a circuit involves creating the circuit by placing parts and connecting with wires, running the simulation for the circuit, and then analyzing the results with probes, oscilloscopes, and meters. Simulation programs are very useful in the field of electronics. Having a simulation program eliminates the physical task of placing wires and parts on bread-boards. Simulation software also makes it possible to construct, test, and analyze a circuit without ever touching real parts, such as chips or resistors (Hart, 1993). The main benefit of this type of software being used in the classroom is that it is personalized to the students; they are able to build circuits at their own pace and with their own design (Lidgey, *et al.*, 1995). With the ability to store circuits on a computer for future reference, the student can create a portfolio of experiments without exhausting a supply of parts and space. Ultimately this could save money because the students do not need to buy this equipment or electronic parts (Castro, 2004). Unfortunately, unless already present the cost of the software and computers may be more costly than the simple electronic parts and equipment. Some simulation programs, such as PSpice, utilize a CAD structure that may not always have a simple user interface (Poole, 1994). If a large amount of time is needed to train a student on how to use the software, then time will be taken away from actual learning. For students in lower-level courses it may be easier to just build a circuit on a breadboard and test it with probes than spending time trying to learn how to do simple tasks with software. Despite being a good skill to have, learning simulation software does not exactly translate into learning circuit analysis. This software only provides the analysis; it does not teach the user how to obtain the results.



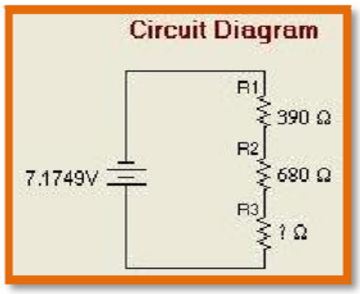


Figure 1: DC Circuits Challenge Layout Examples.

Tutoring software enables a user to learn by use of practice and example. Two examples of tutoring software are Web Tutor and DC Circuits Challenge. DC Circuits Challenge is a teaching program made by ETCAI that enables students to sharpen their skills on various topics in electrical circuits. Tutoring software, like DC Circuits Challenge, uses predetermined circuits for students to practice on (Baser, 2006). In DC Circuits Challenge, the values of the components change allowing students to practice the same type of circuit given various values. Tutoring software that uses predefined circuits, like DC Circuits Challenge, results in the student analyzing the same circuit over and over again with just the values of the components changing (ETCAI, 2008). This may help the student memorize an approach to solving a particular circuit, but may not help when the circuit changes. Figure 1 shows two examples of DC Circuits Challenge circuits. Notice that only the component values change between the two examples.

Web Tutor, developed by Benedykt Rodanski, is a program placed online where students can practice example exercises to increase knowledge and practice solving electrical circuits (Rodanski, 2006). Web Tutor generates a circuit within a predefined framework (Rodanski, 2008). Students using software that generates a random circuit will be subjected to many different circuit layouts. Web Tutor circuits are randomly generated having constraints placed on the size of the circuit and the orientation (Rodanski, 2008). Figure 2 shows examples of circuits generated by Web Tutor. While different, the circuits produced are similar visually.

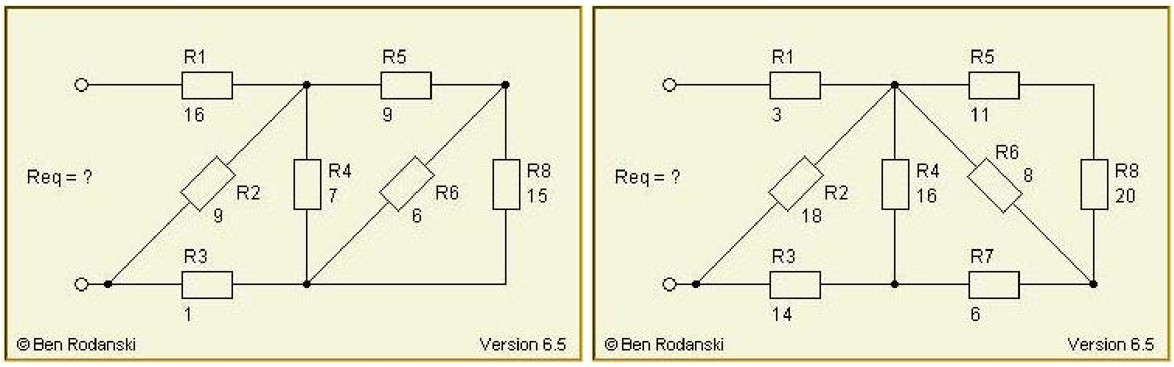


Figure 2: Web Tutor Layout Examples

By having the software itself create the circuit there can be many variations of circuits for any given topic. Students can learn by practicing as much as they want until they are comfortable. If a student grasps the concepts of a particular lesson then that student does not need to practice as much as another student who does not fully understand the lesson. If software allows for an infinite number of circuits, then students can practice as much as they need. In some cases tutoring software could replace book work and quizzes completely (Lidgey, *et al.*, 1995). Rodanski has proved that software tools can supplement a course with great results (Rodanski, 2006). Tutoring software can be used as a tool that forces students to look at examples and practice until the concepts become familiar. Tutoring software, however, does have drawbacks, specifically those found in the programming. The topics are limited by the amount of programming put into the software package. Some tutoring software may not cover all the topics covered by the course. Even though the concepts remain the same, the visual difference in some circuits may be confusing. Therefore, practicing on many different circuits will prepare the student more effectively (Baser, 2006). Students who practice on software that can generate any given number (and type) of circuits may have an advantage over the students who practice on one circuit where only the values of the components change. Therefore, the type of tutoring software that is used will determine the benefits.

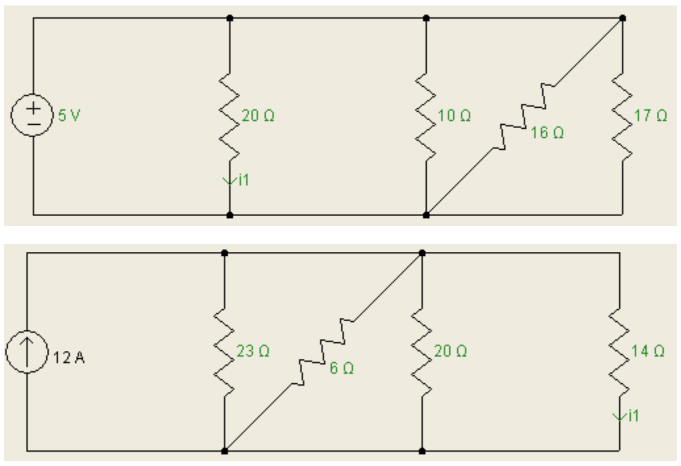
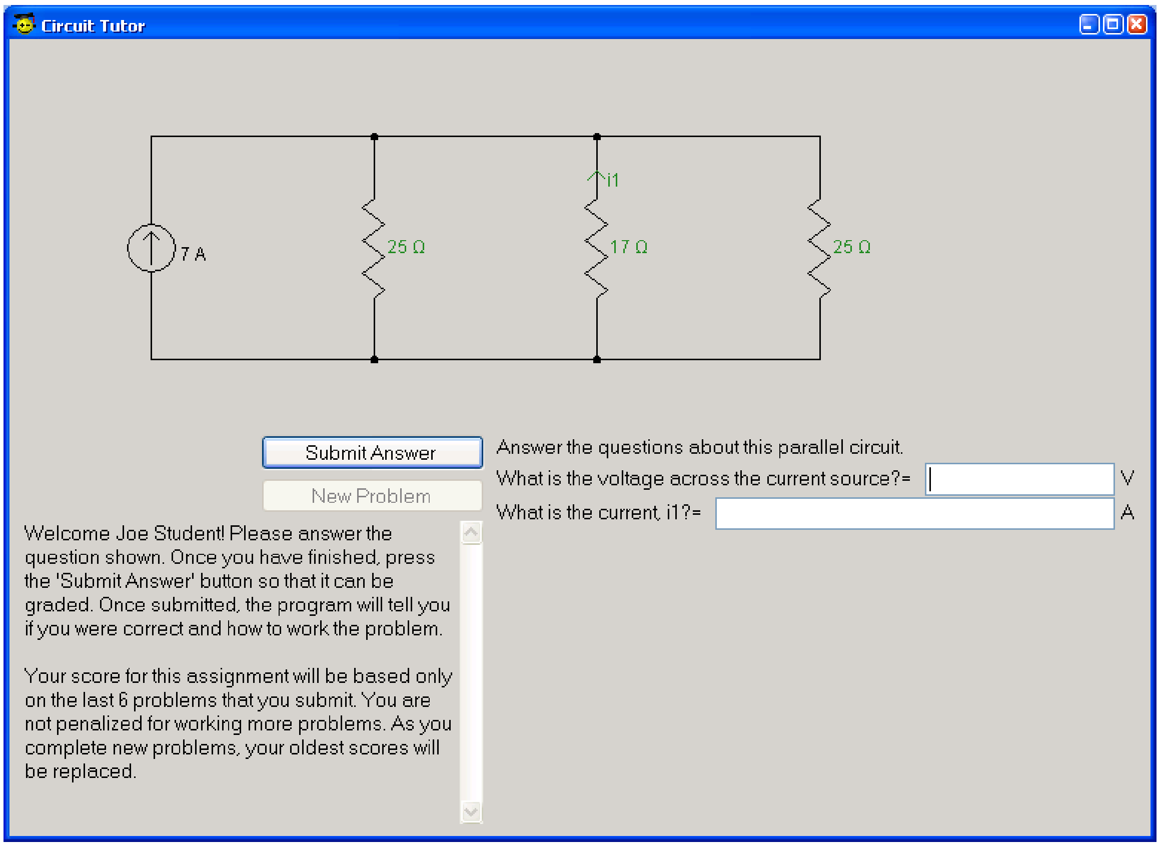


Figure 3: Circuit Tutor Examples

3. Circuit Tutor Software

Tutoring software called Circuit Tutor is the focus of this paper. While it is similar to Web Tutor and DC Circuits Challenge in many respects, it is significantly different in others. Figure 3 shows examples of Circuit Tutor layouts. While the Circuit Tutor examples are similar to Web Tutor, the program does allow for more variation. A more significant difference between the three tutoring software packages is the ways in which they provide help. DC Circuits Challenge and Web Tutor give help by showing a generic example or by pointing to a specific theory on which the problem is based. Circuit Tutor, on the other hand, provides customized text showing step-by-step how a specific problem is solved. Figure 4 shows a problem as presented to the student.

Figure 5 shows the answers as submitted and the resulting explanation. The full text of this explanation is shown in Figure 6. This text is customized for each problem.

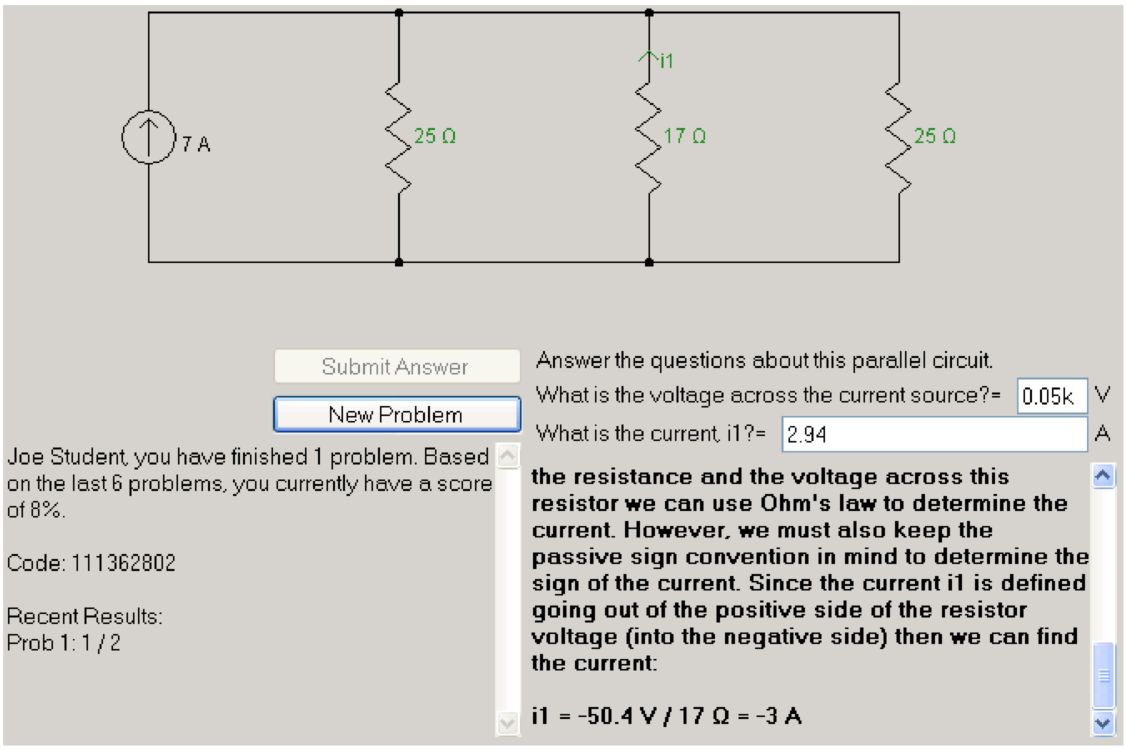


Answer the questions about this parallel circuit.
What is the voltage across the current source?= V
What is the current, i1?= A

Welcome Joe Student! Please answer the question shown. Once you have finished, press the 'Submit Answer' button so that it can be graded. Once submitted, the program will tell you if you were correct and how to work the problem.

Your score for this assignment will be based only on the last 6 problems that you submit. You are not penalized for working more problems. As you complete new problems, your oldest scores will be replaced.

Figure 4: Circuit Tutor Example as Presented to Student



Answer the questions about this parallel circuit.
What is the voltage across the current source?= 0.05k V
What is the current, i1?= 2.94 A

Joe Student, you have finished 1 problem. Based on the last 6 problems, you currently have a score of 8%.

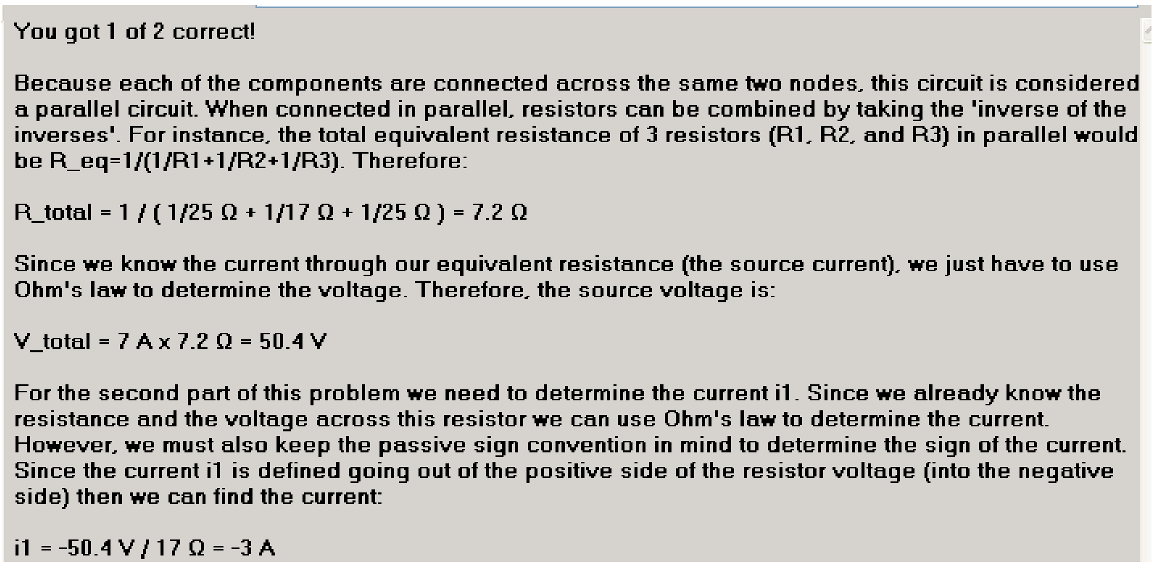
Code: 111362802

Recent Results:
Prob 1: 1 / 2

the resistance and the voltage across this resistor we can use Ohm's law to determine the current. However, we must also keep the passive sign convention in mind to determine the sign of the current. Since the current i1 is defined going out of the positive side of the resistor voltage (into the negative side) then we can find the current:

i1 = -50.4 V / 17 Ω = -3 A

Figure 5: Circuit Tutor Example Once Submitted by Student



You got 1 of 2 correct!

Because each of the components are connected across the same two nodes, this circuit is considered a parallel circuit. When connected in parallel, resistors can be combined by taking the 'inverse of the inverses'. For instance, the total equivalent resistance of 3 resistors (R1, R2, and R3) in parallel would be R_eq=1/(1/R1+1/R2+1/R3). Therefore:

R_total = 1 / (1/25 Ω + 1/17 Ω + 1/25 Ω) = 7.2 Ω

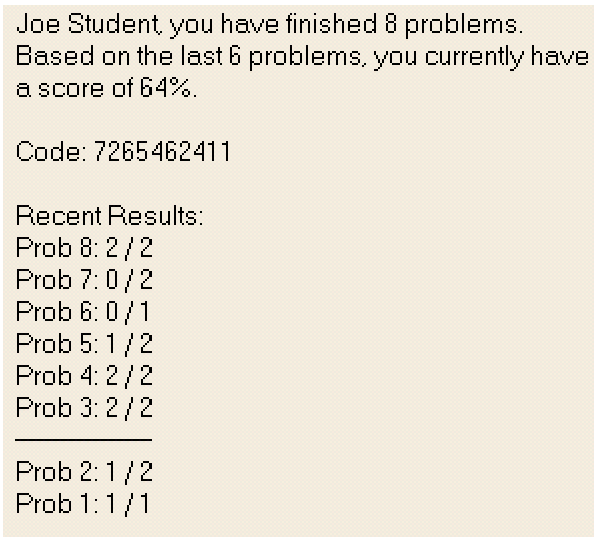
Since we know the current through our equivalent resistance (the source current), we just have to use Ohm's law to determine the voltage. Therefore, the source voltage is:

V_total = 7 A x 7.2 Ω = 50.4 V

For the second part of this problem we need to determine the current i1. Since we already know the resistance and the voltage across this resistor we can use Ohm's law to determine the current. However, we must also keep the passive sign convention in mind to determine the sign of the current. Since the current i1 is defined going out of the positive side of the resistor voltage (into the negative side) then we can find the current:

i1 = -50.4 V / 17 Ω = -3 A

Figure 6: Circuit Tutor Example, Explanation Close-up



Joe Student, you have finished 8 problems.
Based on the last 6 problems, you currently have a score of 64%.

Code: 7265462411

Recent Results:
Prob 8: 2 / 2
Prob 7: 0 / 2
Prob 6: 0 / 1
Prob 5: 1 / 2
Prob 4: 2 / 2
Prob 3: 2 / 2

Prob 2: 1 / 2
Prob 1: 1 / 1

Figure 7: Circuit Tutor Scoring Example

The software is designed to increase students' critical thinking abilities. When students see a circuits question on an exam they must first determine a method to solve the problem. This is by no means trivial. Even electric circuits that look extremely similar may be analyzed in very different ways. For this reason, memorizing theories is not enough. Students must comprehend the theories and determine their applicability to specific circuits. Unfortunately, this comprehension cannot be 'taught'. While instructors can provide resources and guide the students, we are not able to 'flip the switch' of understanding. That is up to the students. One way to get students to 'see' the answer is through the use of examples. Most software used in circuits class will analyze a circuit but not show how it is done. The Circuit Tutor software, however, attempts to do this. This software challenges a student with a problem, checks their answer, then shows the student how they could approach the problem. Like most other tutor software, it has a ‘practice mode’ that repeats the same type of problem over and over to build familiarity with how the specific problem type is solved. A difference with this software is that it also has a ‘test mode’. In this mode random types of problems are given so students must learn to recognize the type of problem in addition to knowing how it is solved.

The software developed as a part of this project can be used to supplement topics that students feel they need more time with. The software currently has 3 topics and each topic has at least 2 subtopics. The software was designed so that it can easily be expanded with more topics. Furthermore, students are able to spend as much (or as little) time with a topic as they (or the instructor) see fit. The software has a special scoring system built in that encourages this. It scores only the most current problems. Therefore, if a student knows a topic they can work the minimum number of problems and get a good score. Alternatively, a student who is struggling can keep doing additional problems until they understand the topic. Since only the most current problems are used to determine the score, a good score is always possible. An example of this is shown in Figure 7. For this particular example only 6 problems are used in scoring. Since the student worked 8 problems the first two are ignored. This method is meant to encourage those who need extra work to do it while not penalizing those who do not need it.

4. Implementation of Software

The purpose of this project was to find a way to enhance classroom pedagogy in the field of electrical circuits. This was quantified using test scores and surveys from students. This project suggested ways to help relieve time strains placed on the educator and increase conceptual understanding for the students. The key concern of the software is to improve the skills of students in a time-efficient manner.

This project involved the students in a lower-level electrical course in the School of Polytechnic Studies at Southeast Missouri State University. The specific course is ET160: Basic Electricity and Electronics. The class involved has thirty-two students. However, not all thirty-two students participated in the project. Aside from the thirty-two potential students in ET160, there was a potential five students from higher-level courses that had already taken either ET160 or a similar course. These upper-level students were used to provide feedback on the software. The project relied on student surveys for information regarding the benefits and opinions of the software when used in the class. A survey was also given to the higher-level students.

Tests were used to gather data indicating the levels of understanding for students in ET160. To get a better understanding of the usefulness of the software 4 tests were administered: pre-test A, post-test A, pre-test B, and post-test B. Both of the pre-tests were given before use of the software. Both of the post-tests were given after use of the software. The pre-test A and post-test A both used a randomly generated problem taken verbatim from the Circuit Tutor software. Pre-test A was given approximately one month before students used the software. Post-test A was different than the pre-test, but similar. It was given immediately after students worked with the software. Pre-test B and post-test B both consisted of problems covering the same concepts as those found in the software. However, the way in which the concept is presented is significantly different than that found in the software. Pre-test B was given approximately one month before students used the software. Post-test B was different than the pre-test, but similar. Post-test B was given a week after students worked with the software.

Test set A was designed to see if the software helped the understanding of a very specific problem type over a very short time period. Test set B was designed to see if the software could help with other types of problems (covering the same concepts) over a longer period of time. While changes were present from the pre-test to the post-test for both test sets, the overall difficulty of the problem was meant to be consistent. These differences were introduced to ensure that students didn’t memorize answers without understanding the underlying concept. It should also be stated that the timing and concepts of both pre-tests were introduced to the students ahead of time. The concept and timing of the post-test A was also announced. Neither the timing nor the specific problem type of post-test B was announced ahead of time. The concepts covered by the software and tests were all covered in lecture before any tests were given.

5. Results of Software Testing

The test data gathered was analyzed using standard statistical processes. A one-tailed t-test was used to determine if the students’ grades increased from pre-test to post-test for each test set. While there was over 30 students in the class, not all students participated in all parts of the project. For example, some students submitted pre-test A, but did not submit post-test A. Some students also took parts of the pre-test and post-test without using the software. Absent of the software used, it is assumed that insignificant improvement was made between pre-testing and post-testing. A limited sample size prevented sub-dividing the group into two parts for control and experiment. Thus the assumption has been made that all improvement were due to the software.

Table 1 lists the results for each test set, the sub-questions of each problem set, the overall results, and the results from 4 students who were not able to use the software because of absence. The P-values shown are the result of the t-test and provides a measure of the likelihood that the result could have been achieved through chance. If the P-value is less than the set alpha level (typically 0.10 or 0.05), the data is considered to be statistically significant. The sample sizes are also shown. Paired t-tests require a pair of pre- and post-tests. Therefore, if one student submitted the pre-test but not the post-test then the data was omitted. The table illustrates whether or not the student achievement, based on grades from pre-tests and post-tests, is enough to assume that the software has a statistically significant effect on the education of students. With the alpha level set at 0.05 the overall test results shows significant improvement since the P-value is less than the alpha level ($0.0005 < 0.05$). Many of the individual problems and questions also show a statistically significant improvement. It should be noted, however, that in Question 2a in Test Set B the average score actually decreased (even if it is statistically insignificant at a P-value of 0.1641). The improvements made on most of the post-test questions, however, make the overall test improvement statistically significant.

Because of absences, a small sample of students (4) from ET160 were given pre-test B and post-test B without using the software. Data from these students is listed in the table as ‘Test Set B Control’. Although the sample size was small, the data gathered did indicate that the post-test results showed no improvement over the pre-test results when the students did not use the software. Thus the assumption that no significant improvement would naturally be made from pre-test to post-test is supported.

In addition to analyzing test results, students were also given an anonymous survey so that they could provide feedback on the software. Most of the responses from the survey are positive regarding the use and benefit of Circuit Tutor. One question, question 10, had a unanimous response that the program would be beneficial for students taking the course in the future. Comments made at the end of the survey were mostly positive. Expressions such as “very helpful,” “very beneficial,” “simple and easy to use,” and “pretty amazing” were all used to describe the software. Circuit Tutor “supported what was taught in the class well,” “definitely helped,” was “good practice with immediate feedback,” “helped to better understand the material,” “enhanced understanding,” and “reinforced the concepts” according to the surveys. Some of the feedback identified Circuit Tutor to be a “good study tool,” and “really helpful when studying” as well as “helpful when preparing for tests and understanding homework.” Not all the comments, however, were positive. Some comments were constructive and provided suggestions on improvement. Overall there were no negative remarks, just positive evaluations and helpful suggestions.

Table 1: Statistical Analysis of Test Results

Test Set A, N=25			
	P-value	*Pre-Avg*	*Post-Avg*
Question 1	0.0159	72.80%	84.00%
Question 2	0.2872	76.00%	79.20%
Total	**0.0476**	**74.40%**	**81.60%**
Test Set B, N=24			
	P-value	*Pre-Avg*	*Post-Avg*
Question 1a	0.0654	83.33%	92.75%
Question 1b	0.0393	43.48%	59.78%
Total for Q1	**0.0162**	**67.39%**	**79.57%**
Question 2a	0.1641	94.35%	93.04%
Question 2b	0.0363	76.09%	93.48%
Question 2c	0.0928	63.04%	76.09%
Total for Q2	**0.0958**	**87.27%**	**90.68%**
Both Test Sets, N=23			
	P-value	*Pre-Avg*	*Post-Avg*
Overall Test	**0.0005**	**73.18%**	**79.88%**
Test Set B Control, N=4			
	P-value	*Pre-Avg*	*Post-Avg*
Overall Test	**0.094**	**98.00%**	**91.00%**

Overall the Circuit Tutor software did improve grades by a statistically significant amount. Not only did Circuit Tutor improve the grades, it also received good reviews by the students themselves. Although the sample size for the study was small, the results were clear; tutoring software does have an impact on the level of understanding the students have regarding electrical circuits.

6. Conclusion

The data gathered from the tests indicate that Circuit Tutor software did have a positive impact on the level of understanding regarding electrical circuits. The data gathered from the surveys show that students liked the idea of having tutoring software available to supplement the lessons as well as replacing homework and quizzes. As a result of the analysis of the data, the pedagogy in the courses on electrical circuits at Southeast Missouri State University may be improved.

The software was also found to provide a time savings. The time it takes to assign and grade homework, assign and grade quizzes, review for tests, and explain how to solve circuits can be partially eliminated with the use of this tutoring software. The students will be able to practice as much as needed with unlimited examples and without the assistance of a teacher, be able to work on homework at their own pace, and be able to study for tests without dedicated review.

Currently, my students are given access to the software and encouraged to use it, but are not required to use it. I am currently working on enhancements to the software to make it more user-friendly. In the future, more topics will be added. Once these modifications are complete, a second, larger study of Circuit Tutor will be performed. With a larger sample size and more time another study may compliment this one with improved results. Although Circuit Tutor can be expanded and improved and more studies performed on the effectiveness of the software, this project was successful in its examination of enhancing classroom pedagogy in circuit analysis courses.

7. References

Attia, J. (1996). "Teaching electronics with MATLAB," *Frontiers in Education Conference*, 2, 609-611.

Baser, M. (2006). “Promoting conceptual change through active learning using open source software for physics simulations,” *Australasian Journal of Educational Technology*, 22 (3), 336-354.

Castro, M. (2004). “Work in progress - integration of new tools and technologies in electronics teaching,” *Frontiers in Education*, 10-11.

Chamas, I. and Nokali, M. (2004). “Automated PSpice simulation as an effective design tool in teaching power electronics,” *IEEE Transactions on Education*, 47 (3), 415-421.

ETCAI. (2008). DC circuits challenge, [Computer Program] (5). Ashland, MS: ETCAI Product.

Hart, D. (1993). “Circuit simulation as an aid in teaching the principles of power electronics,” *IEEE Transactions on Education*, 36 (1), 10-16.

Lidgey, F., Rawlinson, M., Pidgeon, M., and Alshahib, I. (1995). “Effective CBL design for electronics education and student feedback,” *IEE Colloquium on Computer-Based Learning in Electronic Education*, 4.1-4.8.

Poole, N. (1994). “The application of simulators in teaching digital electronics,” *Engineering Science and Education Journal*, 3, 177-184.

Rodanski, B. (2006), “Dynamic web-based tutorial tool,” *Information Technology Based Higher Education and Training*, 67-70.

Rodanski, B. (2008), Web tutor. [Computer Program] (6.5). Sydney, AU: Ben Rodanski.

Dr.B. DEKEN is currently an Assistant Professor in the Industrial and Engineering Technology Department at Southeast Missouri State University. He has a PhD in Electrical Engineering from Purdue University with a research focus in machines and power systems. Dr. Deken combines this with an interest in education and computer programming. He has been a member of IEEE since 2000.

CHRIS COWEN is currently working in the PCB manufacturing industry in Dallas, TX. He has a MS degree from Southeast Missouri State University in Industrial Management with a focus in Teaching and Training. Mr. Cowen has a strong drive for education and the utilization of technology.

Peer Review: This article has been internationally peer-reviewed and accepted for publication according to the guidelines given at the journal’s website. Note: This article was accepted and presented at IAJC-ASEE International Joint Conference on Engineering and Related Technologies sponsored by IAJC, ASEE, IEEE, venue at University of Hartford, USA during April 29-30, 2011.