Artificial Intelligence for Engineering Design, Analysis and Manufacturing (1998), *12,* 51–54. Printed in the USA.

POSITION PAPER

Stimulating appropriate uses of simulation in design

JAMES J. ROSENBERG
Department of Engineering, Harvey Mudd College, 301 E. 12th St., Claremont, CA 91711, U.S.A.

(Received June 11, 1997; Revised September 22, 1997; Accepted August 8, 1997)

Abstract

This article addresses the issue of educating undergraduate engineering students in the appropriate use of computer simulation in the design process. The premise that poorly designed assignments involving simulation can actually impair understanding is addressed. A set of goals for simulation-based exercises is suggested, and some tactics for meeting these goals are introduced. Finally, a specific example of a half-term assignment that is used to meet these goals is provided for illustration.

Keywords: Simulation; Design; Education

1. INTRODUCTION

At the outset, a distinction is drawn between the use of computers to educate students regarding design (an important topic), and educating students to use computers in design. It is the latter topic that is addressed here. Views on and experiences with teaching undergraduate engineering students how to (and how not to!) use computer simulation in performing design will be conveyed.

2. APPROPRIATE APPLICATION OF COMPUTER SIMULATION IN THE UNDERGRADUATE ENGINEERING CURRICULUM

Conventional wisdom suggests it is becoming increasingly important to include the use of computer simulations in the engineering undergraduate curriculum. While this is almost certainly true, it is extremely important to recognize that poorly chosen assignments involving computer simulation can actually *reduce* students' understanding rather than increase it.

It has been the author's experience that students will often jump at the opportunity to use simulation tools, even to perform simulations of physical systems for which exact closed-form solutions can be found. It seems easier than learning to apply modeling techniques and grind through algebra. Particularly in the early stages of their college careers, students do not yet appreciate that "getting an answer" to a problem is not useful in and of itself. They do not yet understand the utility (particularly in the context of doing design) of understanding the *general* behavior of a system and how that behavior varies with system parameters. This should not be too surprising because the educational paradigm in most high schools (and even, unfortunately, in some college courses) is that students are posed "closed-ended" problems that have a "right" answer, and the students are rewarded for finding that answer. In short, at the outset of their college careers they do not understand that while numerical solutions can provide an easy way to predict system behavior for one specific input and set of system parameters, this information is useful only in a narrow portion of the design process.

A typical example of this from the electronics area involves a situation in which students are given a topology for a circuit such as a simple amplifier, and asked to find values for components (e.g., resistors and capacitors) to achieve some design constraint, such as gain. Such simple problems are useful at the early stages of learning about circuits when students execute these problems by hand. When students start using circuit simulators *before* they learn about the behavior of simple mathematical circuit models, they

---

Reprint requests to: Prof. James Rosenberg, Department of Engineering, Harvey Mudd College, 301 E. 12th St., Claremont, CA 91711, U.S.A. Phone: (909) 607-3886; Fax: (909) 621–8967; E-mail: jim_rosenberg@hmc.edu

are prone to use the simulator to "tweak" component values until the desired behavior is obtained. When this happens, the students never develop a "feel" for how the circuit works. Among other things, this means that if a student is asked to redesign the circuit for a different value of gain, the first design iteration gives them no guidance for the redesign. More importantly, the student has developed no feel for the trade-offs involved, such as how the gain of the amplifier and its frequency response interact. It is my belief that incorporating computer exercises of this type produces poorer engineers, not better ones.

First, a set of goals that should be a part of what guides the incorporation of computer simulation in undergraduate courses is suggested. Ideally, exercises involving computer simulation should contribute to an understanding of:

- when computer simulation is useful, and when it is not;
- the interaction between approximate, closed-form solutions and numerical simulation;
- the hierarchy of tools—from 3D distributed model simulation down through simple lumped-element simulators and numerical "scratchpad" programs;
- the limitations of the simulation tools; and
- how to use visualization tools.

Structuring assignments to meet all of these goals simultaneously is difficult (perhaps impossible). It *is* possible, with some effort, to create assignments that realize several of these goals at a time, and to create sets of assignments spanning a course that achieve all these goals.

First, it is necessary to clarify the goals listed above. To understand when simulations are useful, students should understand under which circumstances a practiced engineer would use a computer simulation. Such situations include:

- when even an approximate closed-form solution that reasonably represents system behavior is simply unavailable;
- when in the later stages of the detailed design process an accurate (i.e., more accurate than a closed-form approximate solution can yield) analysis is needed;
- for visualizing continuous fields or distributions (e.g., stress/strain fields, electromagnetic fields, or temperature distributions); or
- for repetitive simple operations (e.g., topology verification for a circuit having thousands of nodes).

The next goal is "understanding the interaction between approximate closed-form solutions and numerical simulation." Here, the fact that the practiced designer will always strive to find simple analytical models that describe various aspects of the system behavior at the outset of the design process is referred. These models usually guide qualitative aspects of the design (and often coarse quantitative aspects), while the simulations will usually be used to carry out the refined quantitative analysis.

The goal of understanding the hierarchy of tools is similar to the goal of understanding the interaction of approximate closed-form solutions and numerical simulations. Often, a computationally intensive finite-element or finite-difference simulator is used only after a simpler, less accurate simulator is used to arrive at an approximate solution. It is important to understand that the computationally intensive programs can *also* be important at the *early* stages of design, where they can be used to help the designer construct simplified models of individual elements within a system. These simplified (typically lumped-element) models can then be used within a less computationally intensive simulation of the entire system of elements.

The goal of understanding the limitations of the simulation tool is fairly straightforward. Nonetheless, the typical procedures that practicing engineers take for granted in performing "sanity checks" on simulations tools (e.g., running a simulation on a configuration that *does* have a closed-form solution and then checking whether the simulated solution matches) must be taught to students. Furthermore, students should have the experience of seeing a simulation tool fail (e.g., not converge on a solution, or better yet, converge to an inaccurate answer because step size or mesh size was poorly chosen). It is important that students have experiences that force them to question simulator outputs, and that force them to learn something about how the simulator works.

3. STRATEGIES AND TACTICS FOR STIMULATING APPROPRIATE USES OF SIMULATION IN DESIGN

The first comment made here is a truism: students should learn by doing. One cannot learn a great deal about when simulation is helpful, how and when it can steer you wrong, and so on by listening to someone talk about it.

The following is a set of goals for a "good" simulation-based exercise that can be used as a yardstick to consider new assignments. The exercise:

- must be tractable;
- should be "conceptually nontrivial"—it should be impossible to complete by tweaking parameters in the simulation;
- should not be "contrived"—it should mimic an actual design process "in context" as much as possible; and
- it should illustrate the "hierarchy of tools" where possible.

Constructing examples that measure up well against all these objectives is difficult and time consuming for the instructor. Nonetheless, it *is* possible to construct such examples that are suitable in the undergraduate environment.

The author's experience is that it is easiest to meet these goals using project-based assignments. The goals of tractability and nontriviality are always at odds. Coming up with a typical "weekly" homework assignment involving simulation that is simultaneously tractable and nontrivial is nearly impossible. The extended time period of a half-term project is a real boon here.

Projects are also a much better environment in which to introduce an exercise that is similar to what a practicing engineer does for a living. Even though a half-term assignment still would represent only a small piece of a typical engineering project, it becomes possible to give the students a much more open-ended problem in which they will have to determine how to process certain inputs that come from some other part of a system (for which they are designing a subsystem), and produce certain types of outputs. Instead of just running a simulator to find "the answer" to a problem, the student now needs to first understand the problem itself, and then determine how simulation may (or may not) be helpful in solving the problem.

Projects also give the students a lot of opportunities to misuse simulations, and the time in which to make such mistakes. On a number of occasions, I have had students chase their tails for weeks on a project by trying to get at a solution by "tweaking," only to discover eventually that they needed to just sit down, and really understand the underlying principles. Following that, they usually can come upon an acceptable design with just an hour or two of simulation. This may be the most valuable lesson that they learn in their technical courses: the computer is no smarter than they are, and does not have "the answers." Of course, this is pointed out before the students begin the project, but it has much more lasting impact when the students discover this themselves. For this type of learning, there is no substitute for projects.

In terms of simulation tools, there always seems to be a trade-off between a tool's power and its ease of learning. In this trade-off, the easiest to learn tool that is sufficiently powerful for the project at hand is opted for. In the electronics area, there are so many computer-aided design (CAD) tools, and they change so rapidly that it hardly seems worth worrying about "training" students on any particular piece of software.

Allowing the students to learn to use the simulator while actually *doing* their design projects is recommended. This is goal-based learning—the students learn what they need to know *when* they need to know it, and are therefore much more engaged as they learn to use the simulator. There are two caveats here: one is that the instructor must be prepared to challenge the students about what the simulator is actually doing, and whether or not they are making valid use of it, and the other is that some students will need some help "getting over the hump" in getting started on the simulator. The tactic recommended for dealing with the latter problem is to give the students a "mini example" before giving them a project. The function of this example is to get them to do *something* with the simulator. It is the author's experience that once the students get past running the simulator once, they rarely need any "hand-holding" at all.

4. A SPECIFIC EXAMPLE

The project used in an Advanced Electronics class is an example of an exercise that measures up reasonably well against most of the author's objectives. In this project, the students (who work in teams of two or three) are asked to design the circuitry for an operational amplifier (opamp) that would be embedded in a large mixed analog/digital signal processing circuit. The inputs that the circuit will have to handle are specified, as are the required outputs and all specifications relating the inputs to the outputs (transient specifications, frequency response, power consumption, etc.). The students are also given the details of the transistor technology with which they will be working (which basically specifies the available components). The specifications are realistic for the type of application that is used as the motivation for the design.

The students are permitted to use any circuit configuration they like. Some typical configurations are covered in lecture, and the theory and simple approximate models are covered in class. The design is not tightly constrained, and there are many acceptable solutions. The students are permitted to use a circuit simulator called PSPICE, one of the industry standard packages.

This example meets the objective of providing a realistic example quite well. It has the useful feature with respect to discouraging tweaking that many of the parameters that the students can adjust in the circuit affect numerous aspects of circuit performance. Therefore, tweaking a parameter to improve one aspect of performance is likely to degrade some other aspect of performance. Without a systematic, orderly approach to the design (*preceding* the application of the simulation tool), success is nearly impossible. This helps students understand *when* in the design process the simulation is useful.

Additionally, the transistor technology available to the students in this design is optimized for the digital portion of the system (the students are working on a piece of the analog portion of the circuit). As a consequence, a standard technique for controlling the frequency response that the students have seen previously is inapplicable here in its simplest form. The inapplicability of the standard technique (as well as a workable modification to the simple technique) can be discovered easily by examination of an approximate closed-form model of a portion of the circuit. Failure to perform the approximate closed-form analysis can lead the student to struggle without success on the standard technique. This helps the students understand the interaction between closed-form approximate solutions and numerical simulation.

Finally, in performing the transient simulation on their circuits, the simulator can misbehave in a number of ways. The students are asked to look at the step response of their circuit. True step inputs (inputs having a voltage discontinuity) are not physically possible, nor is PSPICE happy about simulating them. Understanding the interaction between the time rate of change of inputs (and other signals) and the step size in the transient simulation is an important lesson that the student learn here. Additionally, experienced users of PSPICE are aware of the fact that the algorithm used for determining transistor currents does not result in a strict conservation of charge (this is mentioned in class). Consequently, the simulated final value of the output response to a step input may be just slightly different from the expected value. If one is trying to estimate 0.05% settling time of the system, these slight differences can be quite important. Consequently, this project gives the students some opportunity to bump up against limitations of the simulation tool.

Despite the complexities described here, the students are able to execute this project in half a semester. Naturally, there is a second project for the second half of the semester (the second project brings in issues of tool hierarchy).

5. CONCLUSION

In summary, there are three main points made. First, poorly planned simulation exercises can *reduce* rather than increase student understanding. Second, it is important to establish a set of goals for the introduction of simulation into design education, to try to have *each* exercise meet at least *several* of these goals, and to have the overall set of exercises in the curriculum address all of these goals. Finally, described is an exemplar project that meets most of the goals set forth by the author, in an attempt to show that it is possible to do so, but that it does take quite a bit of the instructor's time and effort to construct the assignment and to administer it.

James J. Rosenberg received the Sc.B. degree from Brown University in Providence, the M.S. degree from the University of California at Berkeley, and the Ph.D. from Columbia University, in New York. He has served at the Assistant and Associate Professor ranks at Brown University, was the Deputy Manager of the Microdevices Laboratory at the NASA Jet Propulsion Laboratory, served as Director of Engineering of Germanium Power Devices Corp., and currently is an Associate Professor of Engineering at Harvey Mudd College in Claremont, California. His research interests include semiconductor designs, analog and digital signal processing circuits, microwave amplification systems, and clock synchronization schemes for high-speed digital systems. Dr. Rosenberg has been awarded the NSF Fellowship for graduate study, the NSF Presidential Young Investigator Award, an IBM Faculty Development Award, and the 1988 Technical Analysis Corporation's President's Award for Excellence in Teaching. He is a member of Tau Beta Pi.