

CADApps: MATLAB Apps for Core Courses in Aerospace and Mechanical Engineering Curricula

Lina Battestilli¹, Larry Silverberg², Jeffrey Eischen² and Caleb Thomas²
*Department of Computer Science ¹/Department of Mechanical and Aerospace Engineering ²
at North Carolina State University*

Abstract:

A long-standing fundamental problem in engineering education is to offer effective design experiences in introductory engineering courses. The design barrier is the lack of Computer Aided Design (CAD) tools that are customized to core disciplinary courses. In this paper, we describe four MATLAB graphical user interface (GUI) Apps (called CADApps) that we have developed to help address this long-standing problem. Our focus has been on developing general-purpose, course-tailored CADApps: Trusses (for the statics course), Frames (for solid mechanics courses), Vibrations (for vibration and system dynamics courses) and Mechanisms (for dynamics courses). An essential requirement has been that students find the GUI interface intuitive and simple and therefore can start using them quickly with minimal training. Students and instructors can use the CADApps in homework problems, in mini design projects, and even in research and development.

Keywords

Technology, Instructional, Mechanical Engineering, Engineering Design, Software

Introduction

Increasing the amount of design in engineering programs, in general, is always a high priority of the Accreditation Board for Engineering and Technology (ABET). The reason for this is that analysis and design are central to the goals of engineering educations and design tends to be emphasized less than educators would like. As any instructor will attest, engineering design is difficult to teach ¹. Analysis is about determining the performance of a given engineering system while design is essentially the opposite. Design is about finding an engineering system that meets performance requirements. Analysis tends to have one correct solution, whereas design has many correct solutions that are found by iteratively changing the parameters in a system and analyzing it repeatedly until a satisfactory solution is found.

Exploration is necessary in engineering design and good design requires good exploration. In fact, exploration is necessary in any educational endeavor and effective learning requires that students explore. However, teaching exploration is difficult! Exploring a large number of designs and engineering concepts is particularly difficult because it is burdened by the complexity of engineering analysis. This is where computational exploration can be enabling.

Computer Aided Design (CAD) software can reduce the burden of engineering analysis and thus has the potential of providing students with an ability to explore more easily.

Realistic engineering design requires sophisticated CAD tools such as SolidWorks, AutoCad, or ANSYS. However, these tools are impractical in most classroom settings because they are complex, require tutorials and training and are not specific enough to the subject matter of a single course. These tools perform general structural analyses, thermal analyses, vibration analyses and fluid flow analyses. The amount of time and experience required to learn them has proven to be prohibitive for freshman, sophomore and junior level courses so one finds these tools being used only at the senior and graduate levels. Therefore, there is the need for simpler CAD tools that are specific to core engineering courses and can be used to engage the students early in the curricula.

The authors got together and set out to create four simple CADApps that could be used in core courses in Aerospace Engineering (AE) and Mechanical Engineering (ME) undergraduate curricula. Developing CAD tools for education is a popular topic of interest^{2,3,4}. What makes our apps unique is that we focused on creating a simple and intuitive Graphical User Interface (GUI) and tailored them to specific core courses around general-purpose problem types. We chose the MATLAB environment because of its popularity in engineering education. As this paper describes, CADApps bring customized exploration tools to early engineering education. They can be used to reinforce concepts and to assist with early engineering design experiences.

This paper presents the details and progress for the four CADApps that we have developed. These apps can help address this long-standing problem of teaching engineering design in early core classes.

CADApps for Early Undergraduate Mechanical/Aerospace Education

We began the development of the CADApps with the goal of closing the gap in the “design thread” found in AE/ME undergraduate programs. We designed the apps to be general-purpose but have course-specific applications. The apps have a GUI where the user can input geometric, physical, and loading parameters of a system. The output of the apps is a solution in the form of specific data points, plots and animation.

The four apps that we developed are called Trusses, Frames, Mechanisms and Vibrations. Each app satisfies the following requirements:

- Pertains to a type of system with wide coverage in an ME/AE core undergraduate course.
- It is general-purpose and can solve multiple types of problems within the type.
- It has intuitive and simple to use GUI that does not require specialized training. The student can learn how to use it in minutes once learning the underlying theory in a class.

The value of CADApps is depicted in Figure 1. Each core class has a specific set of Core Learning Objectives. Typically, in early undergraduate courses the focus is on Basic Understanding and Simple Analytical Analysis. The reason for this is, in part, because of the introductory nature of the subjects but also because more complex problems are difficult to solve by hand. However, with the use of software tools, such as CADApps, it becomes possible to teach simple engineering design, even at the freshman and sophomore levels, and make the

conceptual connections between the subjects covered and the power of the subjects in their ability to apply to real-world problems.

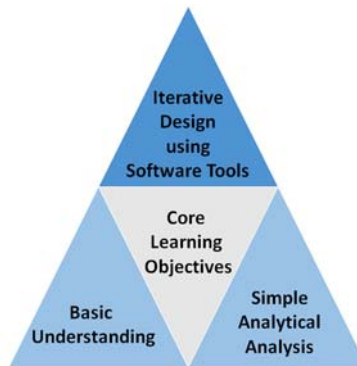


Figure 1: CADApps Value Proposition

The CADApps are unique because they are designed around general-purpose problem types, which broadens student and instructor appeal. Students can use the CADApps to check the answers in homework assignments and they can use them in small design projects. Another important benefit is student empowerment. Students gain control over how they explore the solution space.

CADApps also brings significant value to course instructors of ME/AE undergraduate core courses. They can make CADApps part of their preparation of class assignments/projects that require numerical modeling and engineering design. In addition to developing the CADApps software, we are also working on a companion manual that will contain problems of different levels of complexity.

CADApps Overview

The four CADApps we have developed are summarized in Table 1. We conjecture that the biggest learning benefit for the students would be if they used all four of the CADApps.

Table 1: CADApps Courses and Learning Objectives

CADApp	Core Courses	Learning Objectives
Trusses	Statics	Analyze and design planar truss structures (small and large displacement effects, internal forces).
Frames	Solid Mechanics	Analyze and design planar frame structures (small and large displacement/rotation effects, internal forces and moments).
Mechanisms	Dynamics	Analyze and design planar mechanisms, kinematic analysis of multibody systems bodies with pin and slider connections.
Vibrations	Vibrations	Perform transient and steady-state analyses of planar multi-DOF spring/mass/damper systems subject to harmonic excitation

Software Design

The development of each of the CADApps was separated into backend code (the theory) and a frontend (the GUI). Consider, for example, the backend code for the truss. It works as follows: For robustness, the truss problem was formulated as a nonlinear dynamics problem. Because the user is *not* interested in the dynamics problem and only the statics problem, i.e. is interested in the system response after the motion has died out, the code auto-selects damping properties in the system, tailoring them so that the motion damp out in 200 integration steps. This is computationally efficient while avoiding computational problems arising from matrix inversions. It also has the added benefit of being “nonlinear,” which means that the displacement do not need to be “small.” For example one can apply loads to a truss and make it buckle - a nonlinear phenomenon that a linear code would not be able to handle. The frame, vibration, and mechanisms Apps are likewise nonlinear, also allowing deformations that are “large.”

CADApps Truss Example

A truss is a structure capable of withstanding loads purely through axial forces in its members. It consists of load-bearing members joined together with pins. Figure 2a shows a 15 node truss that was created using the CADApps GUI. The properties of the nodes and members of the truss can be entered and edited in two ways; either via a popup GUI or via the Nodal and Member Property tables. The tables give the user finer control over the locations of the nodes and the specific properties. The steps to create this truss are:

- First, in the *Structure Mode* the user selects the nodes and connects them via members.
- Next, the user clicks *Boundary Conditions Mode* and places pins or rollers at specific nodes.
- Finally, the user clicks the *Load Edit Mode* and proceeds to apply forces at specific nodes.

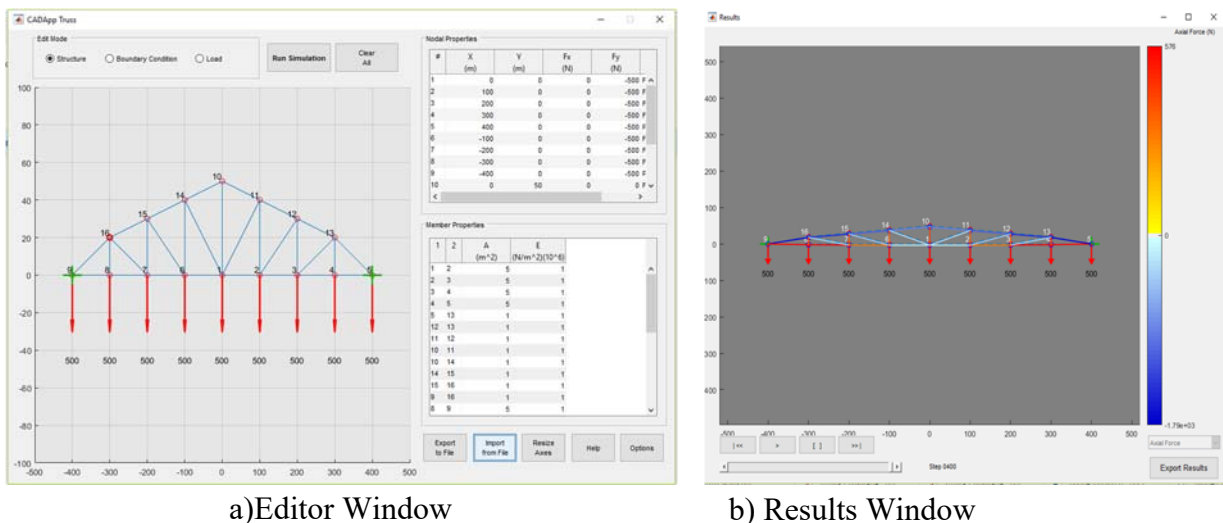


Figure 2. Truss App

After the truss is created, the user clicks the Run Simulation button and observes the effects of the forces on this particular structure. The user can also export the truss to a file so that it can be later imported back into CADApps. The resulting simulation for this particular truss is shown in Figure 2b. The different colors show compression (blue) and tension (red). When

hovering over a member with the mouse, the member properties appear and when hovering over a node, the nodal properties appear. A popup table appears that has information about them including their displacement (if any). The original configuration of the truss (dashed line) is shown under the deformed configuration of the truss. The user can then Export Results into a comma separated value (CSV) file if further analysis is desired.

CADApps Mechanism Example

The mechanisms app models rigid body motion - as commonly seen in undergraduate dynamics classes during discussion of kinematics. The rigid members can be connected and constrained by attaching them to a pinned connection or a slider. The sliders can either be external - a roller - or internal, where the slider slides along another member as shown in Figure 3. Node and member motion can also be defined to further constrain the problem. After a problem is fully constrained, the program can solve and animate the motion of the system through time for the user as well as provide instantaneous motion data (see Figure 4). This data can be used to solve problems commonly presented to undergraduates in their dynamics classes. Figure 3 shows the Mechanisms app system creation GUI with a fully constrained example problem.

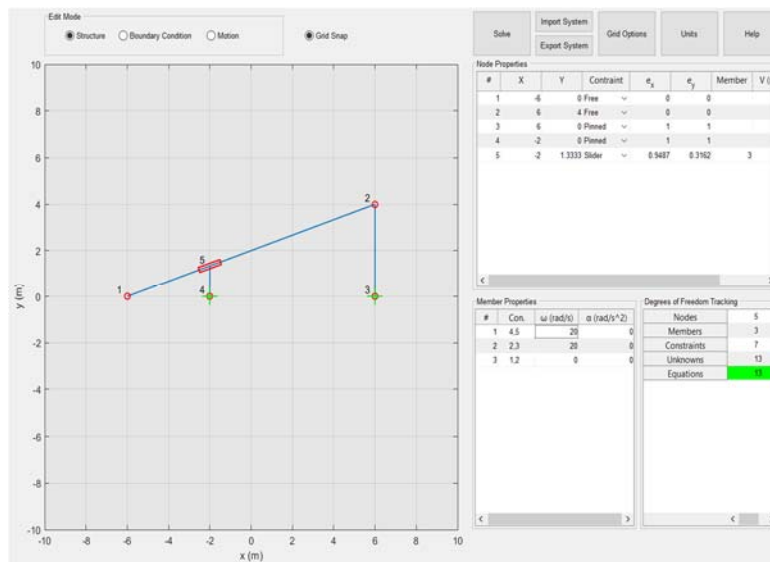


Figure 3: Mechanisms App Editor GUI

We can see in the bottom right corner of Figure 3 there is a table titled 'Degree of Freedom Tracking'. This table ensures that the system is neither under or over constrained. If the problem is not properly constrained, the user will be alerted and guided to fix it before the system can be solved. We can also see that nodes 3 and 4 are pinned, while node 5 is an internal slider that is sliding along member 3. Furthermore, the motion assignments of the members can be seen in the 'Member Properties' table. Figure 4 shows the solution to the problem depicted in Figure 3. The user can find the instantaneous solution data they would need for undergraduate problems by hovering over the nodes and members in question. In addition, the time response of the system is displayed through a time lapse line as well as a full animation, allowing the user to achieve a better understanding of the system's motion.

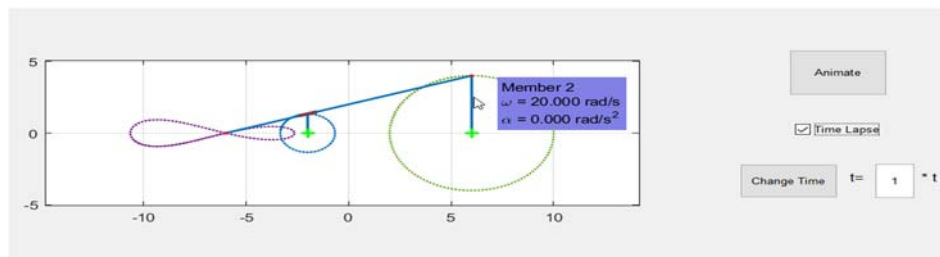


Figure 4: Mechanisms Solution

Study of Classroom Use of CADApps

We used CADApps for projects in two very different courses: Intro to Computing MATLAB Course (CourseA), and Solid Mechanics Course (CourseB). CourseA is basic computing requirement for ME/AE freshmen and sophomores students, while CourseB is typically taken by ME/AE juniors. In CourseA, the students have not been exposed to much engineering yet, if at all. In the project for CourseA, CADApps was used in as a data analysis project. By the time students take CourseB they have already attained some competency to analyze and perform some design in a few engineering domains.

After students worked on projects using CADApps in both courses, we wanted to find out the students experience and opinions of the software. We developed a uniform method of assessing CADApps and ran some preliminary surveys. The surveys focused on perception outcomes. The focus on perception outcomes correlates with student motivation, a widely accepted educational goal, and comparatively easier to assess than direct learning outcomes. The perception outcomes assess student appreciation of the role of computing in their program of study, the perceived improvement in their understanding of important course concepts, and their sense of the utility of CAD apps outside of the classroom (and the value of sophisticated computer tools). The surveys were done on two CADApps - Truss and Frames - because those were the two apps used in these particular courses. We had PRE and POST surveys consisting of 16 questions each. The results show trend lines associated with increasing interest in computing from the freshman year to the junior year. Figure 5 shows the results for the perceived perception of the student for the use of computers in Engineering Design. In the POST survey, after students have used CADApps, close to 60% of the junior students see value in using computers in contrast with only the 30% of the freshman/sophomores. This trend was observed in all of the questions in our surveys. By the time students are juniors, they are starting to see the value of computing and the use of CAD tools for engineering design.

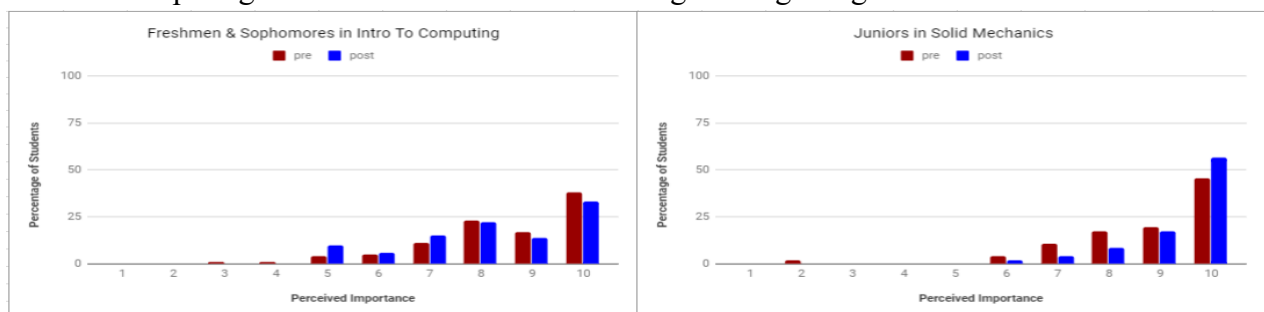


Figure 5: Use of Computers in Engineering Design

Conclusion

The CADApps presented in this paper can assist in addressing the long-standing problem of design in early engineering education, addressing a variety of issues on both sides of the equation – the performance side relating to the field of engineering design, and the cost side relating to such constraints as compatibility with current lesson plans, preparation overhead, etc. CADApps could strengthen the computing thread, in particular the use of MATLAB, throughout engineering curricula. Presently, we are completing the software testing phase of the CADApps and plan to release the code for open use. We also hope that the companion manual will help instructors integrate CADApps assignments in their ME/AE courses. Our early results are promising and students are successful at picking up CADApps and using it with minimal help. Our future work is to open source the CADApps code and promote dissemination of the tools more broadly.

References

1. Dym, Clive L., et al. "Engineering Design Thinking, Teaching, and Learning." *Journal of Engineering Education*, vol. 94, no. 1, 2005, pp. 103–120, [doi:10.1002/j.2168-9830.2005.tb00832.x](https://doi.org/10.1002/j.2168-9830.2005.tb00832.x)
2. Wolfsteiner, P. (2012, June), Teaching Multibody System Simulation: An Approach with MATLAB Paper presented at 2012 ASEE Annual Conference & Exposition, San Antonio, Texas. <https://peer.asee.org/22009>
3. Rhudy, M., & Nathan, R. (2016, June), Integrated Development of Programming Skills Using MATLAB within an Undergraduate Dynamics Course Paper presented at 2016 ASEE Annual Conference & Exposition, New Orleans, Louisiana. 10.18260/p.25770
4. Zuo, Wenjie, et al. "EFESTS: Educational Finite Element Software for Truss Structure. Part I: Preprocess." *International Journal of Mechanical Engineering Education*, vol. 42, no. 4, 2014, pp. 298–306., [doi:10.1177/0306419015574637](https://doi.org/10.1177/0306419015574637).

Biographical Information

Lina Battestilli, North Carolina State University (NCSU)

Lina Battestilli is an Assistant Professor (Teaching Track) in the Department of Computer Science. Upon receiving her PhD in the area of Performance Evaluation of Data Communication Networks, she worked in industry for seven years including IBM Research. Currently, her research interests are in innovation of Computer Science Education, effective ways of teaching Computing to non-majors and increasing the diversity in engineering education. She teaches undergraduate courses in programming and computer networking and advises the Computer Science capstone course.

Larry Silverberg, North Carolina State University

Larry Silverberg is a Professor in the Mechanical and Aerospace Engineering Department. His area of research is dynamics. It's about how and why things move - from the design of aerial vehicles that fly over animal wildlife in the African savanna, to fundamental questions about the motion of celestial bodies. This paper addresses an educational barrier in early undergraduate education with getting more design/exploration into the classroom and offers an approach to doing this.

Jeffrey Eischen, North Carolina State University

Jeffrey Eischen is an Associate Professor of Mechanical and Aerospace Engineering. He also serves as Director of Undergraduate Student Affairs in his department. His research interests include: linear and nonlinear finite element analysis, multi-body kinematics/dynamics/control, fabric mechanics, and stress analysis in microelectronic devices. He teaches undergraduate courses in solid mechanics and mechanical design and a graduate course in fracture mechanics.

Caleb Thomas, North Carolina State University

Caleb Thomas is a Senior undergraduate student studying Aerospace Engineering. His main research area is computational fluid dynamics and has worked with the NC State's Aerospace Engineering Computational Fluid Dynamics Laboratory and the Kirtland Air Force Base Research Laboratories. He has also been heavily involved with MATLAB programming since the start of his Junior year and has developed graphical user interfaces for structural and fluids analysis and the CADApps project.