

A New Kind of Software for Teaching Structural Behavior and Design

Kirk Martini

University of Virginia

Introduction

Structural analysis software has now been commonplace in teaching architectural structures for many years. The advantages in visualization and indeterminate analysis are widely recognized and accepted.

The most widely used analysis programs are intended for commercial engineering design: programs such as RISA (RISA Technologies) and Multiframe (Formation Design Systems). The great majority share two key characteristics:

1. **Interaction:** These programs are based on a three-stage model of interaction: 1) model the structure, 2) analyze, and 3) interpret results.
2. **Theory:** Most programs are limited to linear elastic theory. Even those that model aspects of non-linear behavior are still based on the direct stiffness method, which mathematically requires that the stiffness matrix be invertible. Physically, this means that the structure must be stable.

Experience has shown that a wide range of useful teaching can be done despite the limitations imposed by these characteristics.

This paper describes a program, called *Arcade*, which takes a different approach to the user interface and to the underlying theory to create a new kind of structural analysis program, which enables new modes of teaching with analysis software.

The interaction model and the computation methods of *Arcade* are both derived from computer games. For the interface, this means that the analysis and interpretation stages are merged, making it possible to interact with a model while an analysis is in progress. It is possible to make changes to the model and then see the effects instantly, the way that a game player sees a game respond to input from a controller.

Accordingly, *Arcade*'s computation method is one widely used in computer games to model the physics of moving objects with greater visual realism. The method is commonly called a *physics engine* or *particle system* (Hecker; Witkin). This method makes it possible to model non-linear large-displacement phenomena such as the changing shape of a hanging cable, or the buckling of a frame (Martini 2006).

The program was developed by the author and has been used in architecture courses at the University of Virginia (U.Va.) and Yale University, as well as engineering courses at U.Va. The following discussion describes examples and experiences in using *Arcade* in an introductory structures course for architecture students.

Elementary Statics Examples

It is widely accepted that students should learn statics before they begin working with structural analysis software (Black and Duff), based on the reasoning that without a knowledge of statics, students will be unable to understand what the program is doing.

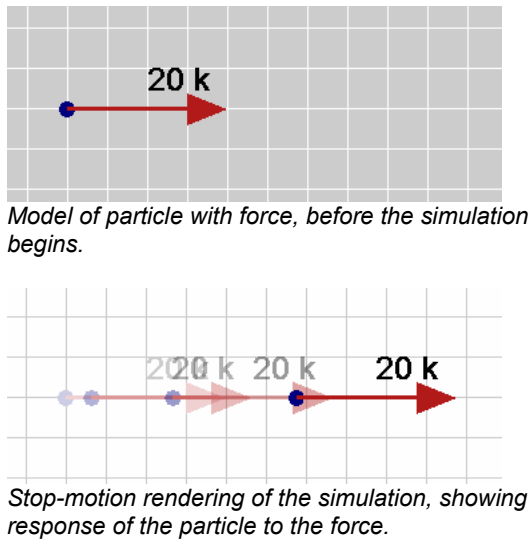


Figure 1. Simple particle mass with force.

Experience with Arcade has shown that it can be used to teach statics from the first day, because of the program's basis in the fundamental physics of $F=ma$. Figure 1 shows an Arcade example that is used in a lecture demonstration. The upper part of the figure shows the basic model: a single particle of mass with a force applied. The lower part of the figure shows a stop-motion rendering of the animated motion when the simulation is run, the particle accelerates in the direction of the force.

This primitive example illustrates some fundamental characteristics of a program:

- **Dynamic simulation:** The program models all phenomena as dynamic events unfolding over time.
- **Particle masses:** A structure is modeled as a collection of particle masses. The dynamic computation is done by stepping through time and solving $F=ma$ for each motion degree of freedom of each particle.

A structural framework is modeled by connecting particle masses with springs, in the form of elements similar to those used in conventional analysis.

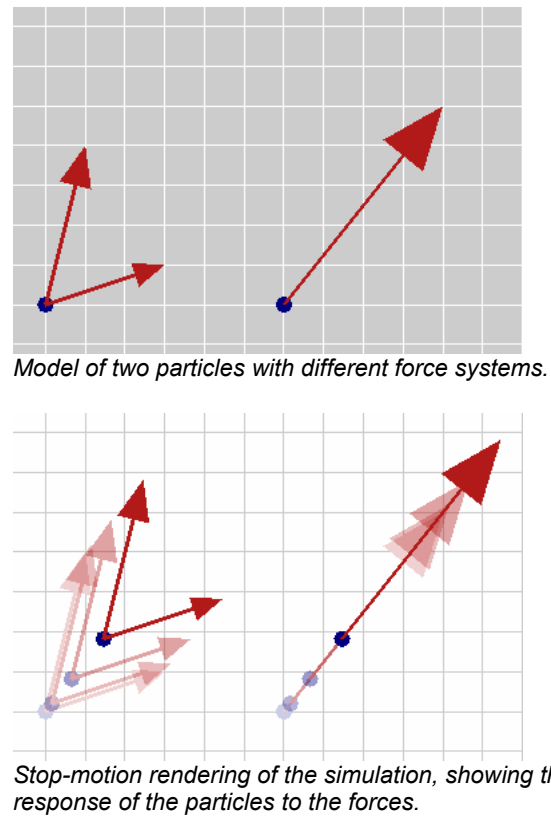
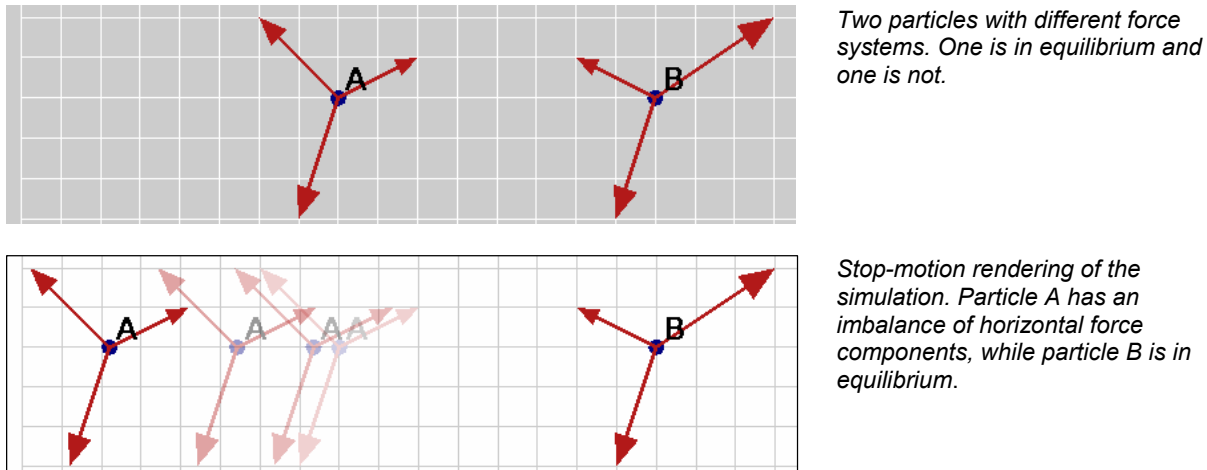


Figure 2. Two particles with equivalent systems.

Figure 2 shows another example from this series, this one illustrating the parallelogram rule.

The upper part of the figure shows the initial model. By counting squares of the background grid, it can be demonstrated that the single force on the right is on the diagonal of a parallelogram formed by the two forces on the left. When the simulation is run, the animation shows that the two particles move in exactly the same way.

Of course, $F=ma$ and the parallelogram rule are usually not among the most elusive concepts for architecture students, so it is reasonable to question the value of these demonstrations. The answer is that it enables a different perspective on force systems, shifting greater emphasis on thinking about how the particles will behave, with less emphasis on the values of the vectors. An example of this shift is shown in figure 3, with another slide from this series.



Two particles with different force systems. One is in equilibrium and one is not.

Stop-motion rendering of the simulation. Particle A has an imbalance of horizontal force components, while particle B is in equilibrium.

Figure 3. Predicting which particle will move, and in what direction.

The upper part of the figure shows two particles. With this slide on the screen, students are put the following question:

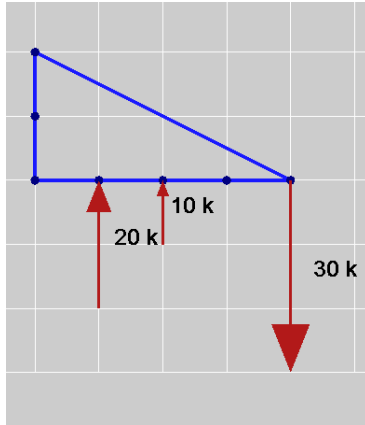
One of these particles will move when the forces are applied, while the other will remain stationary. Determine the one that moves. If you do that, then describe how it will move

Students are given a few minutes to discuss the problem in pairs, and most pairs correctly identify that the forces on particle A are unbalanced and will move the particle to the left. Students use the technique of counting grid squares to check the balance of horizontal and vertical components. The grid counting technique is of course not practical for more complex problems, but is effective in a lecture setting because it allows the discussion to focus on concepts of force balance rather than details of vector component calculation. More important, students are focusing on a question of physical behavior (How will the particles move?) rather than on one of mathematical properties (What are the vector resultants?).

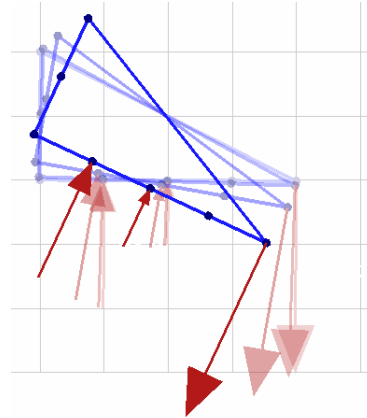
Figure 4 shows an extension of this approach to the statics of bodies. In this homework problem, students begin with an Arcade model where a body is represented

by particle masses connected by stiff beam elements. The force system is configured so that the horizontal and vertical components balance, but the body rotates under the action of the forces, as shown in the right part of the figure. Students are given the task of adding exactly two forces to the figure which put the body in equilibrium. The correct answer is any two forces that produce a counterclockwise couple of 800 kip-feet (the grid in the figure has a module of 10 feet).

As with the particle examples, this problem places the emphasis on the physical behavior rather than mathematical properties, although both must be understood. Free body diagrams are often a difficult concept for both architecture and engineering students, in part because students have little life experience dealing with free floating bodies with forces applied. The author has found that a group that has been taught statics with Arcade becomes quite quick to answer correctly when presented a free body with unbalanced forces and asked the question “how would this move?”; this is probably because these students have seen a dozen or more examples of free bodies moving under unbalanced forces and most begin to develop intuition based on that experience.



Body with rotationally unbalanced force system.



Stop-motion rendering of the simulation, showing the response of the body to the force system.

Figure 4. Body with unbalanced forces. Students must add two forces which put the body in equilibrium

Structural Frameworks Examples

Figure 5 shows an application of Arcade to a conventional truss structure. The rendering options are set so that the width of the member indicates the magnitude of the force, and the color indicates the sense of the force, with red indicating compression and blue tension.

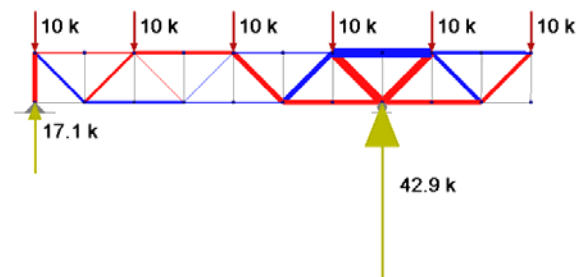


Figure 5. Truss rendered to indicate force magnitudes in members.

This rendering supports discussion of the overall patterns of forces and how the truss is working. It also shows clearly that some member have near zero force, which raises the traditional topic of zero force members, covered in most introductory texts on structures and statics. Traditionally, texts teach that in a system of three forces with two in-line, equilibrium requires the third force to be zero. This fact of statics is used to explain zero force members, and it is usually mentioned that the zero force

members are necessary for secondary bracing.

Arcade allows the topic of zero force members to be cast in a new light, with its "Element disable tool" (more commonly known as the "bomb tool.")

When this tool is clicked on an element during a simulation, the element is removed from the model immediately and the structure responds accordingly. Figure 6 shows a stop-motion rendering of the truss after one of the "zero force" members is removed.

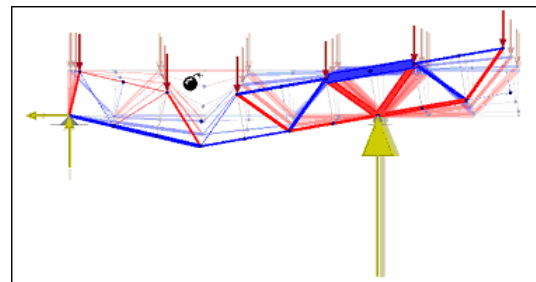


Figure 6. Stop-motion rendering of truss after a "zero force" member is removed.

During a lecture presentation, the bomb tool is used to remove different zero force members, and it becomes clear that removing some members causes collapse, while removing others does not. The class

is asked to explain why that is. Before long, an astute student will note that in the case where the zero force member meets two in-line compression members, the structure collapses, but when the in-line members are in tension, the truss does not collapse.

This observation allows the traditional textbook explanation to be elaborated as follows:

While it is true that for three forces with two in-line, equilibrium requires that the third force be zero, in a real structure, nothing is exactly in line, because the structure deforms under load, and probably wasn't built precisely in line to begin with. When the off-line "zero force" member is removed, compression in the remaining two will make them fold together, leading to collapse, while tension will make the members straighten out and stabilize.

This interpretation gives deeper insight into truss behavior than the traditional treatment of zero force members, and underscores the importance of small members that brace major compression members.

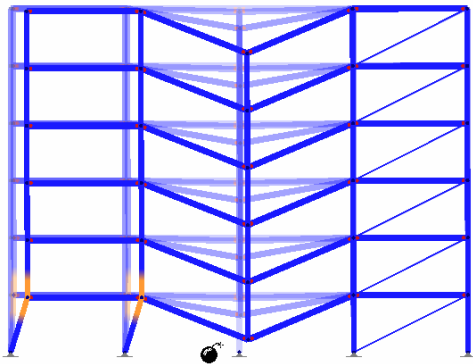


Figure 7. Stop-motion rendering a frame after a column is removed.

The bomb tool is also useful for illustrating broad aspects of frame behavior. Figure 7 shows a stop-motion rendering of a conventional building frame after a first-story column is removed with the bomb tool.

The lecture presentation goes on to compare the behavior of this frame of that with more redundant frame configurations that can withstand the sudden removal of a first-story column

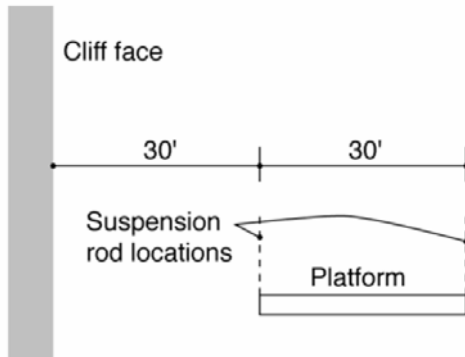
Brief Design Exercise

Ultimately, the reason for teaching structures to architecture students is to instill the ability to make informed decisions that recognize the interactions between structural form and behavior. Such teaching requires generative exercises, where students create original solutions in a design context with objectives and constraints. Despite the importance of generative exercises, the demands of studio projects on student time often make it difficult to include them in a structures course.

The following discussion outlines a brief (90-minute) computer-based exercise done in a computer lab setting where students generate a structural form using a highly simplified design context, objectives, and constraints. The outline of this project was published as an essay in the *Connector* newsletter in 2001. At that time, the RISA program was used for analysis [RISA 2006], and it could be carried out with that or other commercial software. The presentation here includes new aspects of the project that have emerged with the use of Arcade.

Figure 8 shows an abridged version of the problem statement. Students work in pairs at their own pace while the instructor and teaching assistants roam the room, assisting pairs that ask for help. Figure 9 shows a photograph of the lab room while the exercise is underway, and figure 10 shows a photograph of a student pair at work. During the exercise, there is a buzz of conversation in the room as pairs discuss their work.

Working in teams of two people at one computer, consider the conceptual design of a structure to suspend a platform from a cliff face, according to the geometry shown below.



The platform is to be supported by suspension rods that attach to the structure you will design. Assume each hanger rod exerts a downward force of 20 kips.

CRITERIA

Use Arcade to design the structure using the following highly simplified criteria.

- The maximum downward deflection of the structure should not exceed 2 inches.
- No single horizontal reaction should exceed 60 kips.
- No single vertical reaction should exceed 30 kips.
- All members should be steel, with a cross area of 20 in^2 , and a moment of inertia of 700 in^4 .

You should try to minimize the weight of your structure while meeting the performance criteria.

Figure 8. Abridged version of problem statement for design exercise.



Figure 9. Brief design exercise in progress

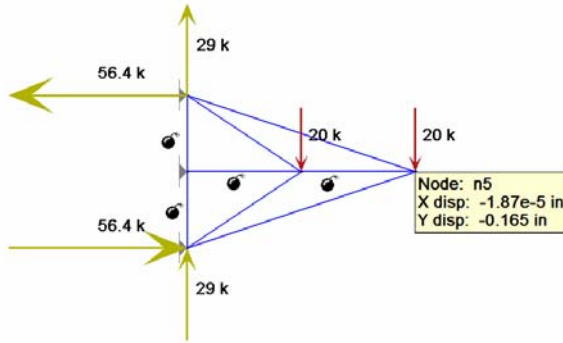


Figure 10. A student pair discusses design decisions.

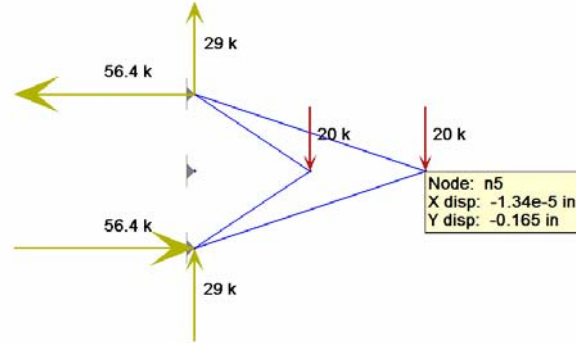
One of the key features of this problem is that it does not involve member sizing, which is completely unrealistic, but forces students to focus on the relationship between form and behavior, where the consideration of behavior is artificially and deliberately limited to reactions and displacement.

One common trend in this exercise is that students begin with structures that have far more members than are required for the narrowly defined problem. As the instructor and teaching assistants talk with pairs, they can ask whether there are any members can be eliminated, and then use the bomb tool to remove the members and check the response. Figure 11 shows an example.

The left part of the figure shows a student model in progress. The vertical members are ineffective because they have a support at each end, the horizontal members appear ineffective because there is no horizontal reaction at the middle support. Four clicks of the bomb tool quickly reveal that the members are in fact unnecessary. Removing the members makes clear the primary behavior of the structure, which in the case of figure 11 is effectively two independent chevron-shaped structures, one for each load.



A student design with ineffective members, indicated with the bomb tool icon.



The structure after the members are removed with the bomb tool, the reactions and displacement are still acceptable.

Figure 11. Illustrating the use of the bomb tool to identify ineffective members in a framework.

Exercise Follow-up

An essential part of the exercise is a follow-up session, done as part of a lecture during the following week. In this session, the following points are projected on the screen and discussed:

The problem highlighted some important aspects of design

- *Design is a process of search within a constrained space.*
- *Design is iterative and cyclic, not linear and direct.*
- *Some things just don't work.*

The problem had good and bad points:

- **Good:** *Allows broad exploration of design space, focusing on behavior rather than calculation details.*
- **Bad:** *The problem was pure model manipulation without any sense of a real structure and how it would be detailed for construction, braced in three dimensions, or how the structure would look in an architectural context. It was a "stress invaders" video game.*
- **Good:** *The problem allowed you to discover why trusses are used, and how to look carefully at the assumptions and constraints of a problem.*
- **Bad:** *The constraints were very incomplete, since they did not include*

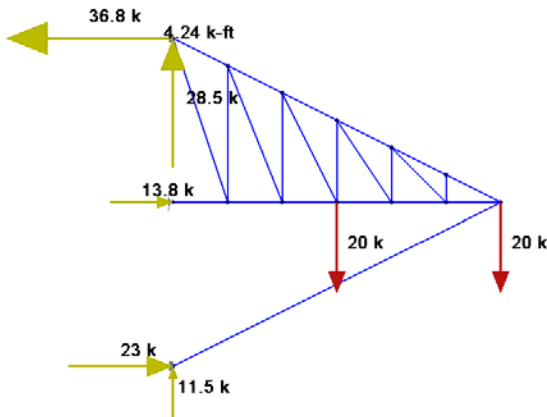
checking the strength of the members, particularly buckling.

- **Good:** *The problem offers an opportunity to show the value of back-of-an-envelope calculations to make decisions about overall form.*

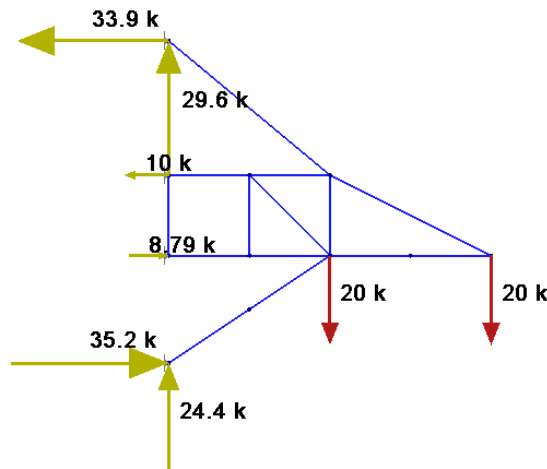
The purpose of this review is to make sure that students understand the significant limitations of the exercise.

The follow-up session also includes a review of some of the typical design approaches in order to learn more general lessons. Figure 12 shows a few such examples, organized in a progression from worst adapted to best. Example a) shows a common approach in design practice: take a conventional structural configuration, and then modify it (in this case by adding a diagonal braces) so that it meets the criteria, resulting in an inefficient design. Example b) is slightly better than a), with example c) illustrating a good solution.

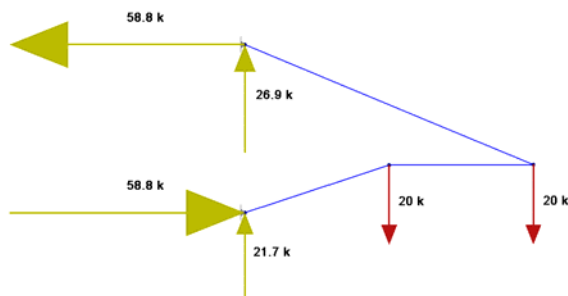
The lecture then derives the moment diagram for the load and span condition and notes that the overall profile of solutions such as that in figure 12c) is similar to the shape of the moment diagram in that its depth varies as the moment diagram varies, although the structure has both legs sloped, which is necessary to keep the individual vertical reactions within the constraints.



a) An ineffective design that adapts a conventional solution. Self weight 26 kips



b) A design that is somewhat better adapted to the problem condition. Self weight 18 kips



c) A design that adapts well to the constraints and objectives of the problem. Self weight 8.6 kips

Figure 12. Sample results from the brief design exercise.

The moment diagram can also be used to calculate the required depth of the structure at the supports. That depth times the maximum allowable horizontal reaction should equal the maximum moment for the cantilever. This calculation is an example of a back-of-the-envelope calculation that an experienced engineer would do, interpreting the global statics of the situation to estimate bounds and overall form.

Of course, the primary benefits of this exercise do not rely on the special features of Arcade, and can be achieved with conventional structural analysis software. Arcade brings two primary advantages:

1. **Familiarity:** Students begin using Arcade from the earliest stages of the course (e.g. statics of a particle), and those initial assignments serve as introductory tutorials so that students become familiar with the program without needing a special tutorial session to learn to operate the program.
2. **Interaction:** As discussed above, the ability to remove members and immediately see the resulting effects is highly effective in illustrating the load-carrying action of a framework. This is a key benefit of Arcade's game-like interaction.

Limitations of Arcade

The limitations of the Arcade program in teaching include the following:

- **2 Dimensions:** The program is limited to 2-dimensional analysis; this is not a limitation of the computation method, but of Arcade's current implementation of it. A 3-dimensional version is planned.
- **Small problems:** Because the analysis runs in real time, it is computationally demanding, so that the program is limited to small scale

problems (typically something less than 100 nodes, depending on the mass and stiffness properties of the structure). This is a significant limitation for commercial application, but is less of a limitation for teaching.

- **No section library or code check:** The program does not include common commercial features such as standard section libraries and code checks. Code checking would not be appropriate for Arcade, since the focus of the program is realistic behavior, but section libraries would be a useful addition.

In general, Arcade is designed toward supporting the work of teachers and students in a classroom rather than the work of an engineer at a desk, since there is already a wide range of high-quality software to support working engineers. Arcade is not intended to supplant the useful role that commercial programs can play in teaching, but rather to complement that role with capabilities that are not easily accessible with commercial programs.

Conclusions

The application of non-linear dynamic analysis in teaching elementary statics challenges two long-standing teaching assumptions for both architecture and engineering students. The first assumption is that students should learn statics before they learn computer-based structural analysis. Experience with Arcade has demonstrated that computer-based analysis can be an effective tool in teaching the most fundamental concepts of statics.

The second assumption is that when students do begin learning computer-based structural analysis, they should begin with linear elastic static analysis, and then progressively work toward mathematically more complex methods. Experience with Arcade demonstrates that its non-linear

dynamic analysis method can be an appropriate entry point for learning structural behavior and analysis.

The key is that although non-linear dynamic analysis is mathematically more complex, it is physically more intuitive. Physical phenomena in real life are not linear, they are not ideally elastic, and they are not strictly static. The real-time interaction of Arcade's non-linear models is closer to student's physical experience than the more stark abstractions used in conventional analysis methods. The greater mathematical complexity of non-linear dynamic analysis combined with real-time interaction is exactly what makes its results more accessible to novice students.

The development of Arcade is ongoing, and the program can be freely downloaded from the project web site [Martini 2006].

Acknowledgement

This material is based upon work supported in part by the National Science Foundation under Grant No. 0230573. That support is gratefully acknowledged. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

References:

- Black, R G and Duff, S 1994. *A model for teaching structures: finite element analysis in architectural education*. Journal of Architectural Education 48(1): 38-55.
- Hecker, Chris. 1996. "Physics, The Next Frontier." *Game Developer*, October/November, 12-20.

Martini, Kirk. 2005. "Teaching Structural Behavior with a Physics Engine." *American Society of Civil Engineers Structural Congress 2005*, New York, CD-ROM publication. Also available from <http://www.arch.virginia.edu/arcade/pubs/arcade-asce-2005.pdf>

Martini, Kirk. 2006. *Arcade: Interactive Non-linear Structural Analysis and Animation*; [cited July 9, 2006] Available from <http://www.arch.virginia.edu/arcade>

Formation Design Systems. 2006. *Multiframe Integrated Structural Engineering Software*. [cited July 9, 2006] Available from <http://www.formsys.com/multiframe>

RISA Technologies. 2006. *Structural Engineering Software*; [cited July 7, 2006] Available from <http://www.risatech.com>.

Witkin, Andrew. 1997. *Physically Based Modeling: Principles and Practice: Particle System Dynamics* [cited July 9, 2006] Available from <http://www.cs.cmu.edu/afs/cs/user/baraff/www/sigcourse/notesc.pdf>