Teaching Structural Behavior with a Physics Engine

Kirk Martini[1]

[1]University of Virginia, Department of Architecture, PO Box 400122, Charlottesville VA 22904; PH 434-924-6445; email: martini@virginia.edu

Abstract

The paper describes a project which uses a novel non-linear computational method to teach structural topics ranging from elementary statics to plastic analysis of beams, and frame buckling. The project centers around a computer program, called *Arcade*, developed using a computational method, called a physics engine, which has been widely used in computer games. This method has two special characteristics with respect to structural behavior: it can model unstable structures, and it performs all calculations with respect to deformed geometry. In addition, the method can perform calculations in real time, so the on-screen structure responds instantly to loads and other actions induced by mouse clicks and key strokes. These characteristics allow the program to show the effect of forces on free-floating bodies, which is ideal for demonstrating free-body diagrams. Combined with elastic beam and truss elements, the method can model cable structures, elastic column buckling, and kinematic mechanisms that result from unstable configurations of supports and internal releases. Used with inelastic beam elements, the program can model plastic collapse mechanisms of beams and frames. The paper describes examples and exercises which have been used with the program in architecture and engineering courses at the University of Virginia.

Overview

To achieve visual realism, the fields of computer graphics and computer games have developed methods to model the physics of moving objects. The simplest approach involves modeling an object as a collection of point masses connected by springs, and then performing a time-step simulation, solving the differential equations of motion with each step. In computer games, the calculation loop runs in real time, so the on-screen model responds immediately to input from a game controller. This approach is commonly called a *particle system*, and the program that runs it is commonly called a *physics engine* (Hecker 1996, Witkin 1997).

Viewed from the perspective of structural analysis, the physics engine approach has two key features:

- **Large displacements**: All calculations are done with respect to the deformed geometry of the structure.

- **Unstable structures**: Since a physics engine does not assemble a global stiffness matrix, it can model structures that are unstable, including cables and kinematic mechanisms.

When structural engineering theory is used to extend the characteristics of the springs to account for flexural and axial material yielding, the approach can model a wide range of non-linear structural phenomena, including post-buckling behavior, and frame collapse mechanisms (Martini 2002).

The ability to model significant non-linear phenomena with real time interaction provides an opportunity to create a new kind of structural analysis program, which is particularly well suited to teaching non-linear structural behavior. This paper describes a project to develop such a program, called *Arcade*, along with associated teaching materials. The software and materials have been tested in structural design courses in the Department of Architecture and the Department of Civil and Environmental Engineering at the University of Virginia. The courses have ranged from a first course in structural design for architecture students, to an advanced course in structural steel design aimed at graduate students and fourth-year undergraduates in civil engineering. The following discussion provides examples of how the program has been applied in lecture presentations and student assignments.

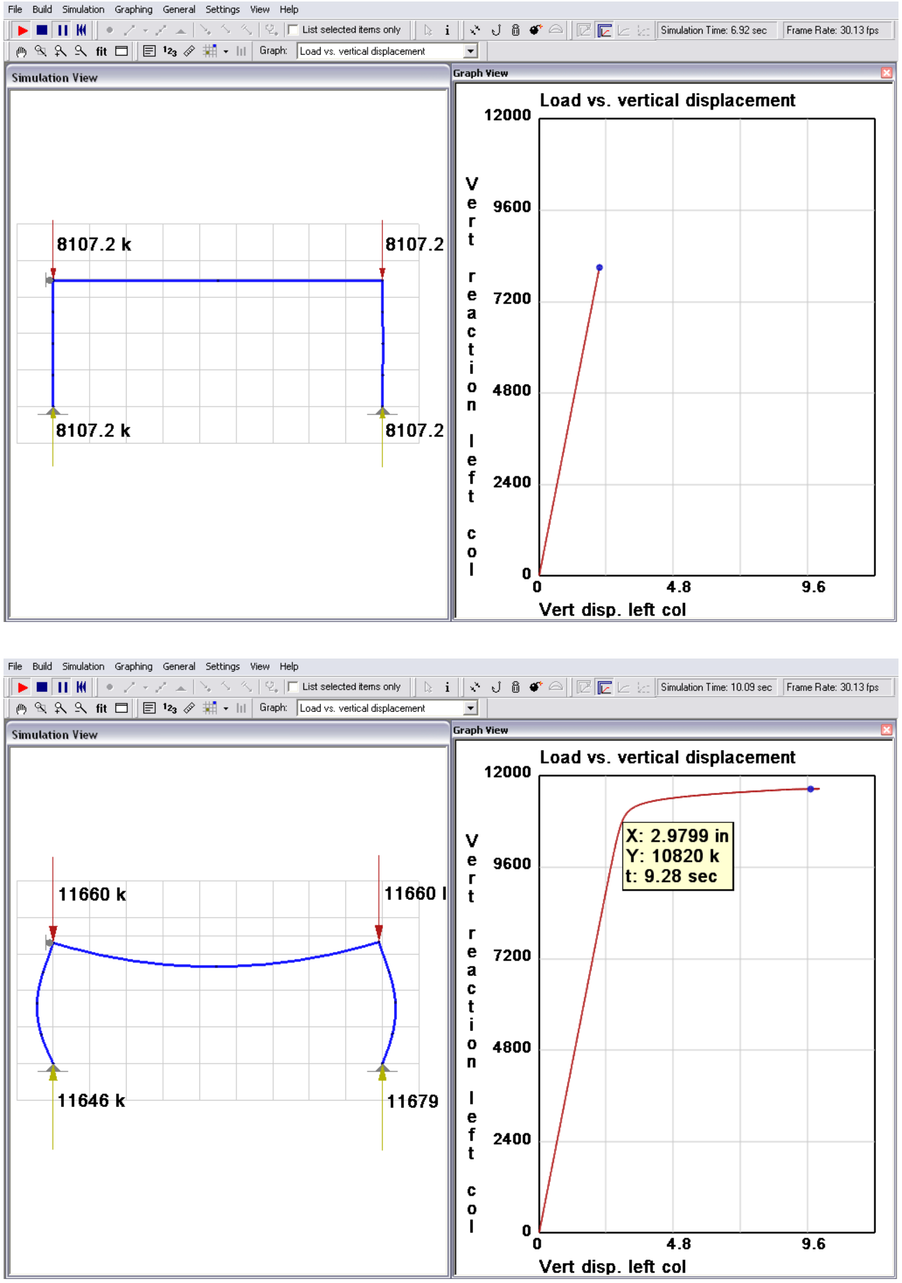
Arcade in Lecture Presentations

One of the special features of Arcade as a lecture presentation tool is the ability to organize a collection of input files into a slide show, so that a series of models can be shown in sequence to illustrate some point of structural behavior. In this mode, the program mimics PowerPoint, except that each page is an animated structural model that responds to input in real time. The following discussion provides three examples of such “slide sets”, concerning frame buckling, inelastic frame behavior, and two-dimensional rigid body statics.

Frame Buckling

Figure 1 shows an Arcade model of a portal frame from a steel textbook (Salmon 1996). In the model, the columns are modeled using 4 elastic beam elements, with the nodes positioned to give an out-of straightness of L/1000. The book uses the frame to demonstrate the calculation of a column buckling load using alignment charts and effective length. The figure shows two screen snapshots, each includes the animation of the frame on the left and a graph which shows the vertical reaction of the left column on the Y axis and the vertical displacement of the top of the column on the horizontal axis. The upper screen shows the frame at about 80 percent of the buckling load, and the lower frame shows the frame at about 110 percent of the buckling load. The analysis runs as a continuous animation, with the graph rising as the load increases.

The lower graph includes a popup window which is summoned by right-clicking on the graph at the location indicated by the upper left corner of the popup. The point where the graph departs from linear corresponds to Euler buckling. The graph shows that this point corresponds to approximately 10,820 kips (48,130 kN), while the calculated Euler load is 10,600 kips (47,150 kN), a difference of 2 percent.



Simulation Time: 6.92 sec
Frame Rate: 30.13 fps
Graph: Load vs. vertical displacement

Simulation Time: 10.09 sec
Frame Rate: 30.13 fps
Graph: Load vs. vertical displacement

Figure 1. Screen shots of a frame buckling analysis.

After seeing the analysis above, students can begin to gain confidence in the close agreement of the program and the alignment chart method of predicting buckling loads. While it is important for students to gain this confidence in their analytic methods, it is also important for them to understand the limits of those methods. The following series of figures shows excerpts from an Arcade slide set on the topic of the alignment chart method and its base assumptions and limitations. These slides were used in an engineering course in steel design.

Figure 2 shows the first slide of a simple example frame composed of two columns and two girders. Calculation by alignment chart predicts an Euler buckling load of 335 kips (1490 kN), with the Arcade model producing highly visible buckling at 342 kips (1522 kN), 2 percent higher. The deflected shape illustrates one of the base assumptions of the alignment chart method: beams in braced frames buckle in single curvature.

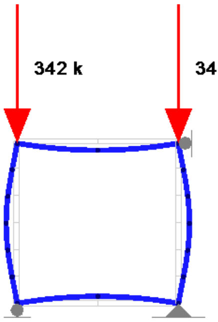


Figure 2: Alignment charts assume beams in braced frames buckle in single curvature.

The next slide illustrates a key assumption of the alignment chart method: all columns buckle simultaneously. The model shows that when the load on the right column is reduced, the left column can then be loaded without buckling to 7 percent above the buckling load of the preceding model.

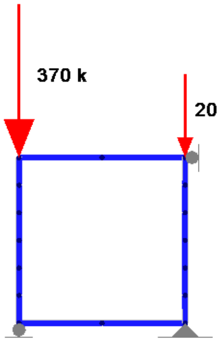


Figure 3: Alignment charts assume columns buckle simultaneously.

The next slide illustrates another key assumption of the alignment chart method: girders are not in compression. The slide shows that when a compression force of 100 kips (445 kN) is added to the girders, the columns buckle at 88 percent of the load that produced buckling without compression.

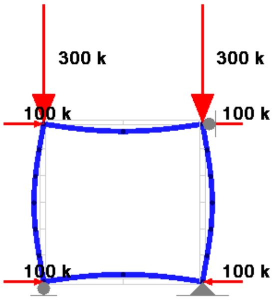


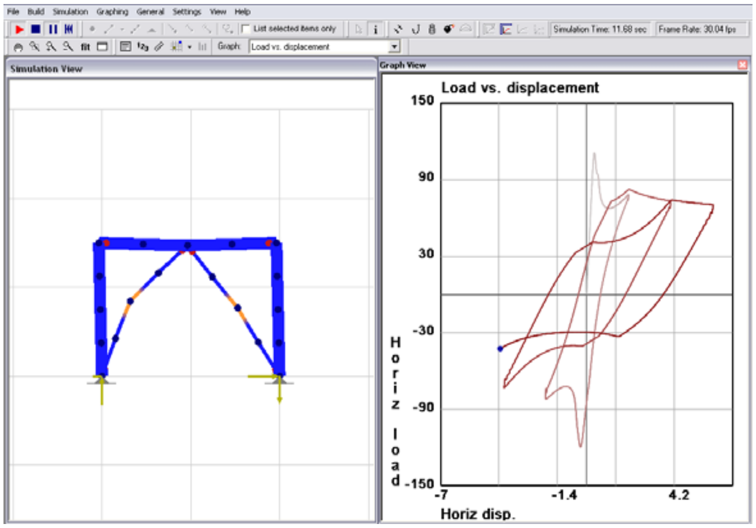
Figure 4: Alignment charts assume beams are not in compression.

The discussion of the slides can point out that while the assumption that columns buckle simultaneously is conservative, the assumption that girders are not in compression may not be conservative. The overall point is to emphasize that the alignment chart method, like any calculation method, has inherent assumptions and limitations and it is the engineer's responsibility to recognize situations where the underlying assumptions do not apply.

This slide set continues, addressing issues in the behavior of sidesway frames, but rather than continuing that slide set, the discussion will now turn to another advanced topic: inelastic frame behavior.

Inelastic Frame Behavior

Figure 5 shows an Arcade model of a chevron-braced frame subjected to increasingly large cyclic displacements, with an accompanying load-displacement graph. The graph is rendered so that the color of the plot is drawn in darker shades for more recent movement, so that the initial movement is drawn in pale shades.


☐ List selected items only

Simulation Time: 11.68 sec

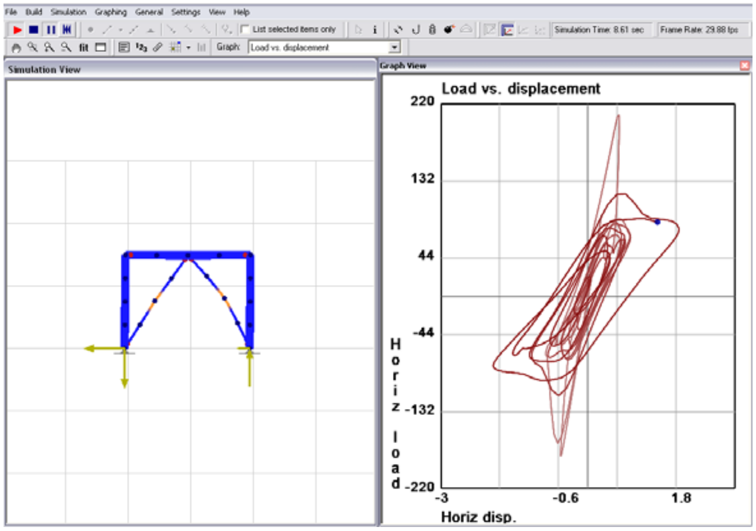
Frame Rate: 30.04 fps

Graph: Load vs. displacement

Figure 5. Cyclic loading of a chevron braced frame.

The graph shows the high initial strength and stiffness of the frame, which then degrades quickly as the braces buckle.

Figure 6 shows a later slide in this set: the same frame now subjected to an earthquake ground motion. The load displacement graph shows the reduction in strength and stiffness that occurs after buckling of the braces.



Simulation Time: 8.61 sec

Frame Rate: 29.88 fps

Graph: Load vs. displacement

Figure 6. Earthquake loading of a chevron braced frame.

This slide set continues with similar examples for a moment resisting frame and an eccentric-braced frame. In contrast to these relatively advanced topics, the next example slide set demonstrates an elementary topic: two-dimensional rigid body statics.

Two-Dimensional Rigid Body Statics

The ability to visualize free body diagrams is one of the most important skills an engineering student can acquire, and often one of the most troublesome. The following series of figures shows excerpts from an Arcade slide set used in teaching rigid body statics.

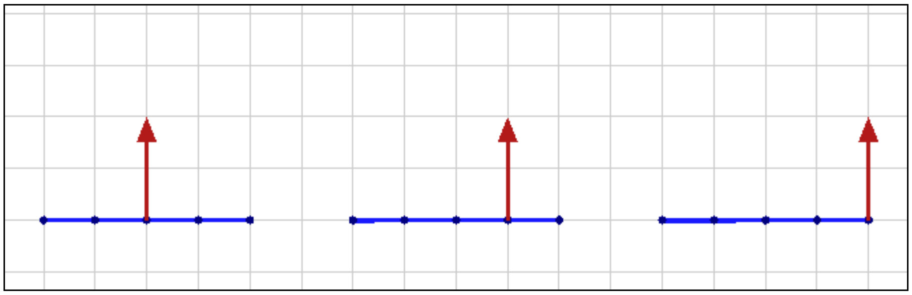


Figure 7: Three bodies with equal forces applied in different locations.

Figure 7 shows the first slide, three free-floating bar structures, each composed of very stiff beam elements. With this slide on the screen, the question is put to the class: how will each bar move? The question is answered by starting the animation, illustrated in the stop-action rendering in figure 8.

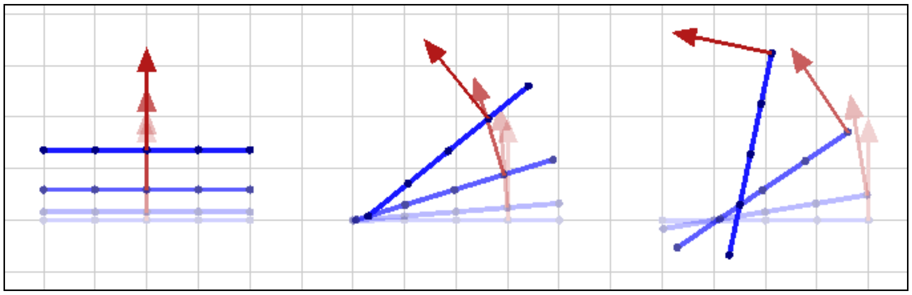


Figure 8. Stop-action rendering of animation.

The example illustrates that the body moves in a straight line when the force's line of action goes through the center of mass, but rotates when the line of action is away from the center of mass, with the rotation increasing as the distance from the center of mass to the line of action increases.

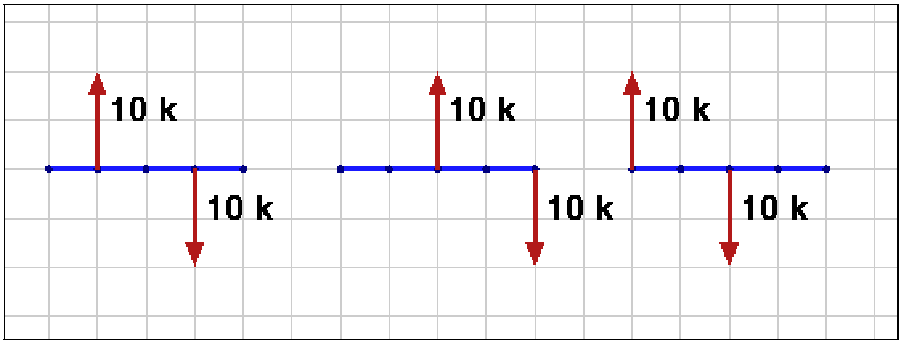


Figure 9. Three bodies with equal couples applied in different locations.

Figure 9 shows the next slide: three bodies, each with a clockwise couple of 10 kips (44.5 kN) applied at distance of 10 feet (3 m). Again, with this slide on the screen, the question is put to the class: how will each body move?

Even though students have studied physics, there are many students who expect each body to rotate about the midpoint between the two forces, rather than about the center of gravity. Figure 10 shows a stop-motion rendering of the animation.

One probable reason that many students answer this question incorrectly is they have little experience with situations where forces act on a free floating body: a rare situation in daily life. As students see numerous examples and work with the Arcade program, their ability to answer questions about free bodies with unbalanced forces visibly improves.

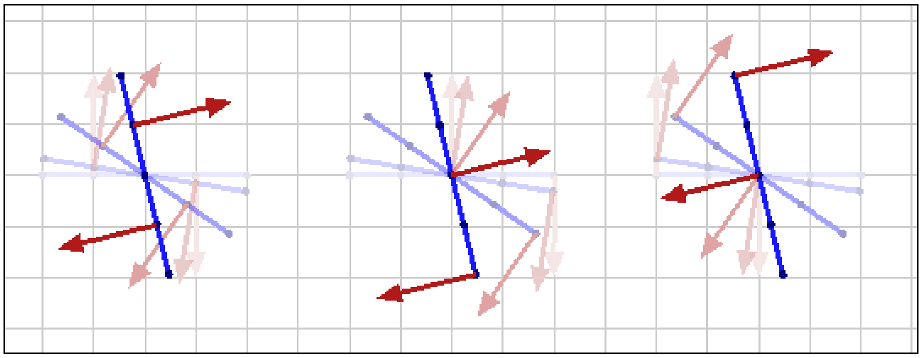


Figure 10: Stop-action rendering of animation.
The movement of each body is the same.

Summary of Arcade in Lecture Presentations

These examples demonstrate the range of applicability of the software. It can address elementary principles of physics and statics, as well as advanced concepts such as second-order, inelastic behavior of structural frames. Of course, it is reasonable to point out that much of the content in the slide set discussed above could be created as pre-scripted multi-media animations using programs such as Macromedia Flash to animate analysis results, rather than using the analysis program to draw the animation. There are two points to consider. First, when a student asks a question in lecture about the consequences of changing some aspect of a model, it is often possible to make the change on the spot and see the result. This pedagogically effective technique is impossible with a pre-scripted animation. Second, with Arcade it is possible to give assignments where students can modify and build models on their own, rather than simply looking at animated models created by others. This is not to say the scripted animations are not valuable, but they are not the same as interactive simulation. Arcade-based assignments are discussed below.

Arcade in Student Assignments

In addition to using Arcade in lecture demonstrations, Arcade has also been used for student assignments. The following discussion describes two such assignments: one on elementary statics, and one on bridge design for single member redundancy.

Elementary Statics: Rigid Body Equilibrium

In this problem, students begin with the model shown in figure 11, where a free-floating body has a system of unbalanced forces. The task is to add exactly two forces to the body which put it in equilibrium. The correct answer is any two forces that produce a counterclockwise couple of 800 kip-feet (1084 kN-m). The grid in the figure has a module of 10 feet (3.05 meters).

The problem has two notable characteristics. First, unlike most problems in a statics course, there is no single correct answer; there are an infinite number of couple configurations that produce the required balancing moment. Second, the program gives instant feedback on a proposed answer: if the answer is correct, the model remains stationary, otherwise the model moves. The type of movement gives a hint about how the answer is incorrect: clockwise movement means their couple is too small, counterclockwise movement means it is too big.

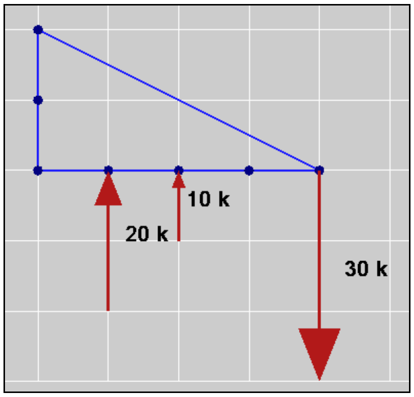
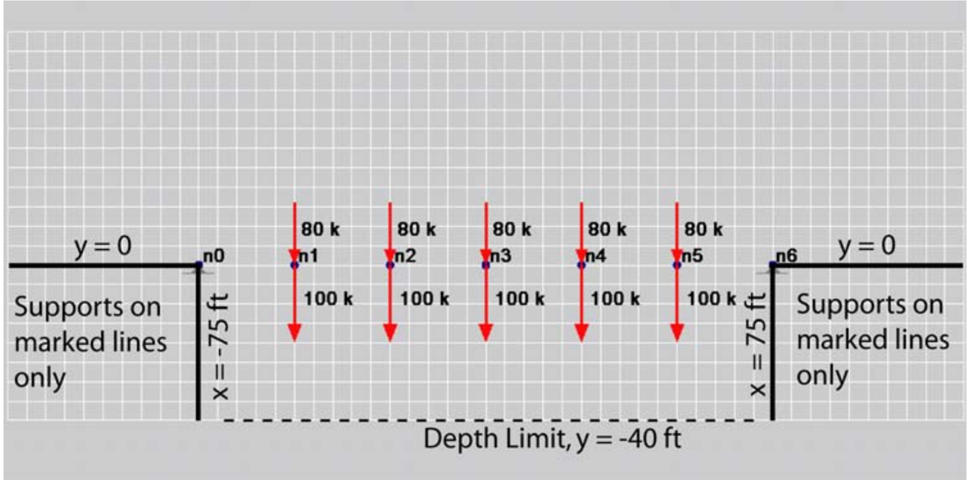


Figure 11. Body with unbalanced forces. Students must add two forces which put the body in equilibrium

Bridge Design: Single Member Redundancy

In the bridge design problem, architecture students in an introductory structures class begin with a starter file that defines basic geometry and loading. They design their bridge according to a problem statement, summarized as follows:

> *The figure below shows the portion of the model defined in the starting file, plus annotation describing constraints. Design a model which meets the performance criteria and has minimum self weight.*



> *Add nodes, elements, and supports to the model as follows:*

__Elements__: Use only truss-2 elements (which model yielding and fracture). You may use any cross section area.
__Supports__: Supports may be added only to nodes which lie on the lines marked in the figure above.

There are three performance criteria for the model:

__Stiffness__: Under full dead and live service load, the deflection at midspan should not exceed Span/500.
__Strength__: Under an overload of 1.2 times the dead load and 1.6 times the live load, no member should exceed the yield stress of 50 ksi.
__Redundancy__: The structure should have single-member redundancy, meaning that any member can be removed from the loaded structure without causing complete collapse.

The primary distinguishing characteristic of this problem is the last requirement concerning redundancy. This requirement is checked on each model using *Arcade's* "bomb tool"; clicking an element with the bomb tool during a simulation immediately removes the element from the model.

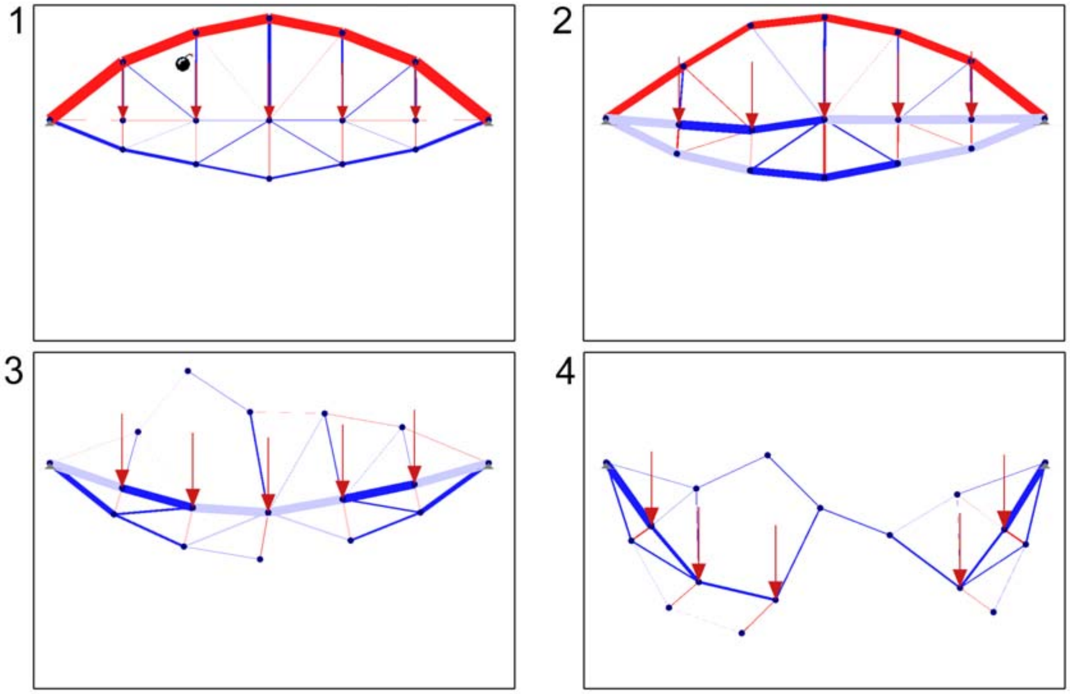


Figure 12. Image sequence showing the collapse of a student-designed bridge when the bomb tool (shown in frame 1) is used to remove an element during simulation.

Figure 12 shows a sequence of four images that span approximately two seconds of time as the structure collapses. In the rendering, the thickness of the members indicates the magnitude of the axial force, with the red color indicating compression and blue tension. The colors are rendered in pale shades when the member is yielding. In the first image, the first vertical member left of center is clicked with the bomb tool, removing it from the model. The next image shows subsequent yielding in several members near both supports. The third frame shows that other members have disappeared; this is because the program removes elements from the model when their tension strain exceeds a user-specified limit, modeling tension fracture. The fourth frame shows complete collapse; this model clearly did not pass the redundancy criterion.

It is important to note that the analysis is unrealistic, since the truss elements do not model buckling. It would be possible to account for buckling by modeling each truss member as multiple beam elements; however, that would make the process of generating and modifying design proposals much more tedious. Since the assignment is aimed at understanding behavior at a qualitative level, the simpler truss elements are used, with students fully aware of that limitation.

The testing of the student-designed models is done as a group demonstration, where the whole class sees the testing of all models. Testing a model typically requires 10 to 12 analyses, one for each element that is removed, and an analysis takes 5 to 10 seconds of simulation, so that each model may take about two minutes of testing, plus time for discussion of the model's behavior. It is particularly useful to have students try to tell whether a model will have a redundancy problem, and where the problem may be.

Redundancy has received significant attention in structural engineering since the disasters at Oklahoma City in 1995 and the World Trade center in 2001. This problem offers an opportunity to work with and discuss redundancy criteria, particularly the fact that single member redundancy may not be an effective criterion, since it encourages a design with many light members. Such a design may actually be more vulnerable to attack than one with fewer, heavier members (Nair 2004). It is clear that redundancy criteria and design to resist blast attack will become increasingly important in structural design practice, and structural education needs to find effective ways to teach it. This problem is a step in addressing that need, and much work remains to be done.

Closing

The current speed and capacity of computers enables a new type of structural analysis program which is not simply a bigger, faster and more colorful version of earlier analysis programs, but one which is based on a fundamentally different computational method. The ability of the physics engine approach to model large displacement and unstable structures makes it suitable to a broad range of topics. The conventional thinking in structural education has been that students need to learn statics before using structural analysis programs. The Arcade program has demonstrated that an analysis program can be effective in teaching elementary statics, and can continue to serve as a useful tool throughout the structures curriculum into advanced topics such as inelastic second-order frame behavior.

Acknowledgements

The work presented in this paper was 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

Hecker, C. (1996) “Physics the Next Frontier”, *Game Developer Magazine*, October/November, p. 12-20.

Martini, K. (2002) “A Particle-System Approach to Real-Time Non-Linear Analysis”, *Proceedings of the 7th National Conference on Earthquake Engineering*, [CD-ROM] Earthquake Engineering Research Institute, 8 pages.

Nair, R.S. (2004) “Progressive Collapse Basics”, *Modern Steel Construction*. AISC, 44(3), 37-42.

Salmon, C.G., Johnson. J.E. (1996) *Steel Structures: Design and Behavior*, Prentice Hall. New Jersey.

Witkin, A.D. (1997) “Particle System Dynamics” *Physically Based Modeling: Principles and Practice*.
<http://www.cs.cmu.edu/afs/cs/user/baraff/www/sigcourse/>