Non-linear Structural Analysis as Real-Time Animation

Borrowing from the Arcade

Kirk Martini
University of Virginia

Key words: Structural analysis, interactive design, animation

Abstract: The paper describes a computational method commonly used in interactive computer graphics and games, and demonstrates its application to structural engineering problems, using a prototype program called *Arcade*. The method enables a new model of interaction in structural analysis, where the simulated structure responds to user input in real time, in the same way that computer games respond. The method shows good engineering accuracy in simple verification problems involving the non-linear phenomena of buckling and beam yielding. The method offers the potential to make non-linear, time-history analysis a much more common method in engineering practice, and to bring a greater emphasis on non-linear, dynamic structural behaviour in structural education.

1. OVERVIEW

One of the significant recent trends in interactive computer graphics is the incorporation of physics and elasticity, particularly in games (Heckler 1996). A common approach is to model objects as a system of particle masses connected by elastic springs, commonly known as *particle systems* (Witkin and Baraff 1997b). Particle systems provide a simple means to model deformable objects which exhibit visually realistic physics at interactive frame rates on conventional personal computers.

Particle systems are related to computer-based structural analysis methods used in the design of buildings, where the springs of the particle system correspond to finite elements used to model structural members.

Although both methods can be used to analyse structures, the particle system approach has received little attention in building structural analysis, primarily because the particle approach is computationally less efficient by orders of magnitude, and is more prone to numeric instability.

However as computation speed and capacity continue to increase, the computational shortcomings of the particle approach become less significant, and it is now becoming practical to consider the application of the particle system approach to structural analysis. The following discussion describes a project to apply the particle system approach to structural engineering problems. The discussion takes the following steps:

- Describe and compare the computational approaches and user interfaces traditionally used in structural engineering and in computer graphics particle systems.
- Describe an implementation prototype, called *Arcade*, which applies the particle system approach to engineering problems, presenting illustrative examples and verifying engineering accuracy.
- Identify potential impacts of the particle system approach on education and practice in structural engineering and architecture.

2. STRUCTURAL ANALYSIS BACKGROUND

The objective of any structural analysis is to determine the response of a structure to an action, where the action may include applied forces, support movements, temperature changes, etc. One of the key characteristics of analysis is the treatment of time. Analysis can be categorised into *static* analysis, which completely neglects time, or *dynamic* analysis, which accounts for time effects. Dynamic analysis can further be broken down into *time history* analysis, which determines the response of a structure at a series of points through time, or *modal* analysis, which determines the response of a structure as a combination of harmonic modes of vibration.

Another key characteristic of analysis is whether it is *linear* or *non-linear*. Linear analysis assumes that the response of a structure is proportional to the action; e.g. if the load on a structure is doubled, then the stresses and movements of the structure also double. In general, linear analysis is highly unrealistic, because there are many important phenomena which result in non-linear behaviour, these include the following:

- **Large displacements**: Linear analysis assumes that differences between the geometry of the deformed and undeformed structure are negligible.

- **Material yielding**: Linear analysis assumes that material remains elastic under at all load levels, without permanent deformation or fracture.
- **Contact phenomena**: Linear analysis assumes that all connections work equally well in tension or compression, however this assumption does not hold for connections where one object simply bears against another, since bearing can transfer compressive stress by not tension.
- **Buckling**: Linear analysis assumes that all members work equally well in tension or compression, neglecting the tendency of slender members to buckle in compression.

Structural engineering practice in building design extensively uses linear static and dynamic modal analysis methods. While these methods are effective in designing safe structures, they are less effective in simulating the real behaviour of structures; this is particularly true with respect to failure and collapse, which typically include any or all of large displacements, material yielding, contact phenomena, and buckling. Such simulation requires non-linear time-history analysis.

Although non-linear time-history analysis is still considered an advanced and somewhat exotic method in structural engineering practice, it is commonplace in the world of computer animation. For animation, time history analysis is essential to calculate the state of the subject at each animation frame. Non-linear analysis is necessary to model the large displacements that naturally occur with the subjects in an animation. In addition, interactive computer games require that the analysis be done in real time, so that the structure can respond immediately and realistically to input from the player.

Although the particle system approach has several shortcomings as a general-purpose method for structural analysis, its ability to support non-linear, real-time, interactive analysis opens new possibilities, particularly in providing physical insight into highly non-linear phenomena through a very different user interface model.

3. USER INTERFACE MODELS IN STRUCTURAL ANALYSIS SOFTWARE

From the early days of computer punch cards to the present day of graphic user interfaces, the conceptual model for structural analysis has retained the following three fundamental stages:

1. **Modelling**: Create a numeric model representing the structure and the actions on it.

2. **Analysis**: Perform the calculations to determine the structure's response to the actions.
3. **Review**: Examine the results of the calculation describing the response of the structure to the actions.

In the early days of structural analysis, the modelling phase consisted of manually preparing tables of numbers for input, and the review phase consisted of scanning tables of numbers from the computer printout. Over the past fifteen years, the modelling and review stages have become far more graphic and interactive, and the analysis stage has become much faster and able to accommodate larger and more complex structural models. Despite those enormous improvements, the process has retained its three-stage batch-processing organisation: modelling, analysis, review, corresponding to input, process, output. In this organisation, seeing the effect of a change to the structure or its loading requires modifying the input, repeating the analysis, and repeating the review of the output. There is interactivity within the stages of modelling and review, but not in the modelling-analysis-review cycle.

It is useful to compare this model with that of interactive computer games. In games, the modelling-analysis-review cycle runs in a constantly repeating loop of real-time computation. Changes to the model are implemented through player actions with a mouse or other controller. The response of the structure is analysed immediately and its display on the screen corresponds to the review. The cycle runs at sufficient speed to provide a graphic update of the analysis results 15 to 30 times per second.

The following discussion describes an implementation prototype, called *Arcade*, whose objective is to merge the computational approaches commonly used in computer games and structural engineering, achieving the real-time interactivity of games while incorporating more sophisticated elements than simple springs, plus sufficient accuracy for engineering applications.

4. COMPUTATIONAL APPROACH

A particle system models a structure as a collection of particle masses connected by springs. The springs will be called *elements* in this discussion, consistent with terminology in structural engineering, and the particles will be called *nodes*. The analysis uses a time-step simulation, where the position and velocity of each node are known at the beginning of each step. The acceleration at the beginning of the step is calculated by first calculating the forces on the node, and then dividing by the node's mass. This calculation is done for each of the node's degrees of freedom, which for this two-dimensional application are horizontal and vertical translation plus rotation. Figure 1 shows the calculation loop performed with each time step:

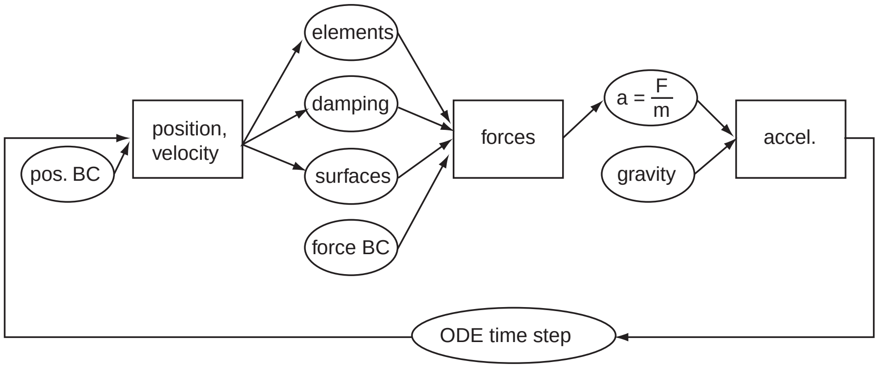

Figure 1. Calculation loop for a particle system The calculation begins with nodal position and velocity at the beginning of the step, then calculating forces, then calculating acceleration, then moving to position and velocity for the next step.

The figure shows that forces on a particle may arise from several different sources, the following are examples included in the *Arcade* implementation (several others are possible):

- **Elements**: Element forces arise from distortions of elements to which the particle is connected. Elements may be simple linear springs connecting two particles, where the force is proportional to the change in spring length, or more complex beam-type elements which transfer shear and moment, and model material yielding.
- **Damping**: Damping forces can be applied directly to a node based on its velocity. With high damping, the nodes behave as if they are moving through a thick fluid.
- **Surfaces**: It is relatively simple to model an elastic planar surface such as the ground to support particles using a *soft contact* approach (Cundall and Hart, 1992) where forces are applied to the particle based on its penetration into the surface. More sophisticated models can include friction and adhesion.
- **Force Boundary Condition**: Although rarely used in computer games, structural engineering commonly uses force boundary conditions where a force is applied to directly to a structure as an external load (e.g. applying a 100 kilo-Newtons force directly to a particle).

When all the forces are calculated on each particle, the acceleration of each particle can be calculated using Newton's second law, dividing the mass of the particle by the acceleration for each translation degree of freedom, and the mass moment of inertia for the rotational degree of freedom. When the elements are simple springs which work only in tension of compression, it is not necessary to consider rotational degrees of freedom, since particle rotation does not effect the spring force. For flexural beam-type elements, which curve and bend, it is necessary to consider angular acceleration and mass moment of inertia.

With the acceleration of each degree of freedom known, the velocity and position at the next time step can be solved using numeric methods for solving the system ordinary differential equations. In concept, this can be solved using Euler's method as follows:

$$
\begin{aligned}
v_{n+1} &= a_n \Delta_t \\
p_{n+1} &= v_n \Delta_t
\end{aligned}
$$

Where a_n, v_n, and p_n, are the acceleration, velocity, and position respectively of a nodal degree of freedom at time step n, and Δ_t is the time step increment. The calculation of the position for the next time step also accounts for displacement boundary conditions imposed on nodes. Typically such boundary conditions are used to model structural supports where the movement of the node is limited in one or more degrees of freedom.

In practice, Euler's method is ineffective because it requires an extremely small time step to achieve sufficient accuracy, and other more sophisticated methods are commonly applied (Witkin and Baraff, 1997a). The Arcade program uses an implicit method based on the trapezoid rule of integration, requiring the system of differential equations to be evaluated three times per time step. Simulation using the particle approach typically requires a time step increment in the range of 0.5 to 0.1 milliseconds. With three evaluations of the equations per time step, that increment range means the program will typically evaluate the system of equations 6000 to 30000 times per second.

5. EXAMPLES AND VERIFICATION

5.1 Interactivity

Figure 2 shows a multiple-exposure screen shot which illustrates the interactive nature of the program. The figure shows a truss structure which is supported at the top and bottom joints to the far left. The mouse is used to select the node at the lower right, the mouse is then clicked and dragged. The dragging adds a temporary element which links the selected node to the mouse. As the mouse is moved, the node at the link follows it, exerting a force on the structure.

Note the use of a linking element to exert forces on the node with the mouse rather than using the mouse directly. When the mouse is used directly on a structure where the elements are very stiff, the high velocities and accelerations of the mouse can generate extremely large forces in the members and lead to numeric instability. Using a relatively flexible link element provides an effective cushion for the mouse forces.

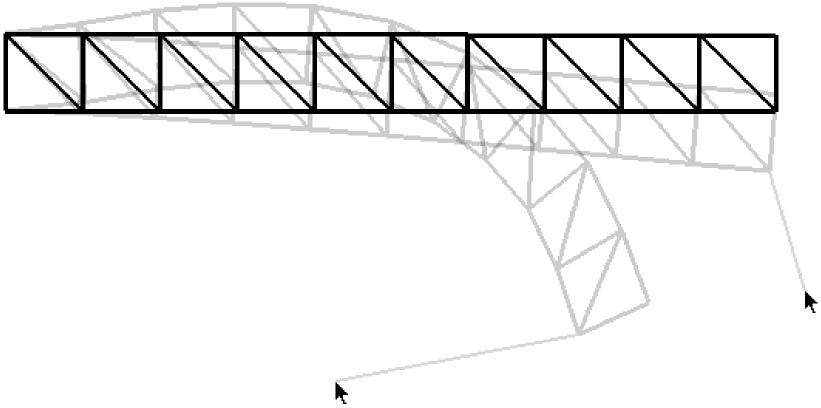

Figure 2. Truss structure loaded interactively via the mouse. The black outline shows the unloaded configuration. The grey shows steps of loading.

In addition to applying forces with the mouse, the program also allows the mass of a node and the global damping to be manipulated interactively. Many other options are possible. The computational approach allows any property of the model to be modified in real time through sliders or other devices and for the effects on structural behaviour to be displayed immediately.

5.2 Element types

One of the primary goals of the program is to introduce element types which are commonly used structural engineering, in particular elements that model material yielding. The elements implemented in Arcade are similar to those used in the DRAIN-2D program developed by Powell (1973), one of the earliest applications of non-linear analysis in structural engineering and still widely used for non-linear earthquake analysis of building frames. Figure 3 shows a the force-deformation characteristics of the non-linear truss element used in the example of figure 2.

Like a simple spring, the element resists only tension and compression forces along its axis, depending on its change in length. Unlike a simple spring, the element can model permanent deformations when the load exceeds a specified yield load (indicated by P_y on the graph). After yielding, the bar unloads on the elastic stiffness, and then yields again on load reversal as shown on the graph.

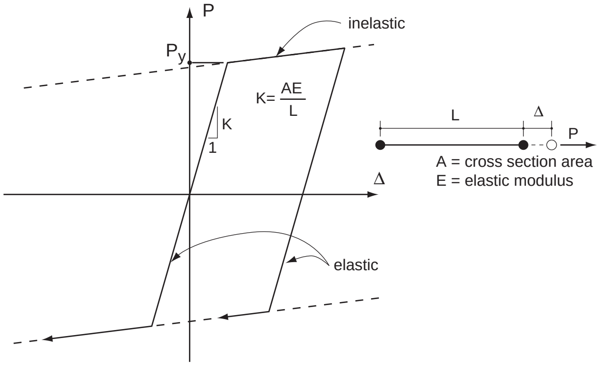

Figure 3. Force-deformation relationship for non-linear truss element

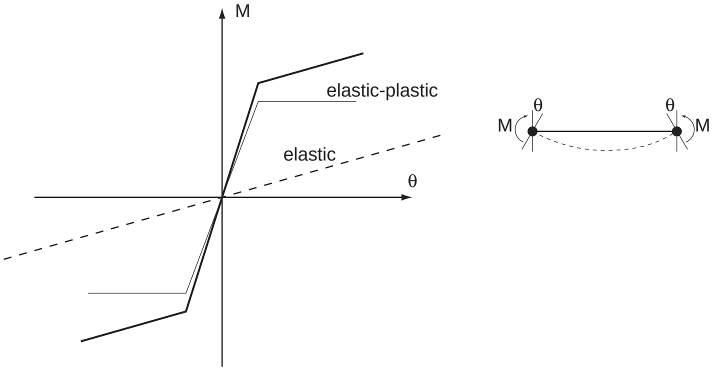

Figure 4. Force-displacement relationship for flexural component of beam element

The other element is a beam-column element which, in addition to resisting axial tension and compression, also resists shear and moment, which depend on the bending of the element (this component is called the *flexural* component). The flexural component also includes material yielding as shown in figure 4. The flexural behaviour is modelled by using two element components working in parallel: a completely elastic component, and an elastic-plastic component. Together, the two elements simulate an initial elastic range, followed by a plastic range with hardening. This concept is based on the DRAIN-2D beam element (Powell 1973), although it is simplified since it does not account for axial yielding, or for interaction between axial and flexural yielding. Figure 5 illustrates the behaviour of the inelastic beam element, as well as a contact surface. The figure shows frames from an animation where a circular structure viewed in elevation falls under the force of gravity and strikes a horizontal surface. The impact with the surface effectively crushes the structure. Note the elements rendered in grey in the third and fourth frames, indicating that the element is on the inelastic branch of it's load-deformation curve.

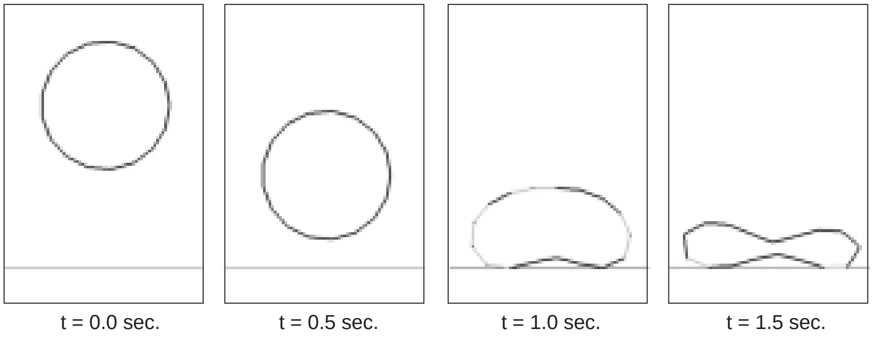

Figure 5. Circular structure, modelled with inelastic beam elements, falling under gravity and striking a horizontal surface. The grey rendering of some elements indicates that they are on the inelastic branch of the load-deformation curve.

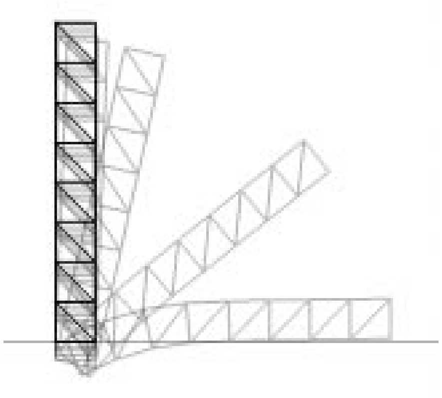

Figure 6. A tower sinking under its own weight into a viscous medium. The exposures are taken at 5-second intervals

The example from figure 5 illustrates the ability of a contact surface to simulate a stiff surface. In addition to stiffness, these surfaces can also model viscosity. Figure 6 shows a multi-exposure rendering of a tower structure bearing on a surface which has no elastic stiffness, but high viscosity, simulating the effect of the tower sinking into soft mud under its own weight. When viscosity is used in combination with elastic stiffness, the surface becomes sticky, similar to a wall or floor coated with adhesive.

5.3 Engineering accuracy

Computer graphics and games naturally emphasise visual qualities over physical accuracy (Blum and Thumrugati 1997), and some applications deliberately produce inaccurate results by reducing the material stiffness far below realistic values (O'Brien and Hodgins 1999), e.g. modelling a metal as if were hard plastic. The stiffness reduction has two advantages: first, it reduces the velocity and acceleration of deformations, making the action easier to see; and second, it allows the simulation calculations to remain stable with a much larger time step, reducing the computational effort required, and allowing more complex models to be rendered at interactive frame rates.

One of the challenges of applying the particle system approach to engineering problems is achieving accurate results on realistically stiff structures at interactive frame rates. Figure 7 shows a verification example based on the Euler buckling formula. The figure shows a column which is fixed against all movement at the base, and restrained against horizontal movement at the top. The column is subjected to a gradually increasing vertical load, plus a moment at the top equal to 0.02 times the vertical load.

The purpose of the small moment is to introduce slight curvature into the column, since a perfectly straight column can carry load beyond the buckling limit. It is modelled using eight elements of equal length, and nine nodes. The structural steel column has the following properties: length = 12.7 m (500 in.) area = 400 cm^2 (62 in^2), moment of inertia = 42870 cm^4 (1030 in^4). The Euler buckling load is then 10.7 MN (2406 kips). When the column is loaded over a period of ten seconds, it exhibits clear buckling deformations at 103% of the theoretical buckling load, and extremely large deformations at 104% of the buckling load. The deformed shapes are shown in the figure. The accuracy of the results are encouraging, although the stiffness of the structure required a relatively small time step of 0.1 milliseconds in order to maintain numeric stability.

Figure 8 shows another example using the inelastic beam element to compared results with simple plastic theory. The figure shows a beam which is completely fixed at the left end, and restrained against vertical movement at the right end, with a concentrated load applied at mid-span. The beam is modelled as elastic perfectly plastic, meaning that it has zero stiffness in the inelastic range. The beam the following properties: span 12.7 m (500 in.), moment of inertia 112661 cm^4 (2750 in^4), area 272 cm^2 (42.1 cm^2), plastic modulus 5277 cm^3 (322 in^3), yield stress 345 MPa (50 ksi). The analysis model uses six equal-length elements and seven nodes.

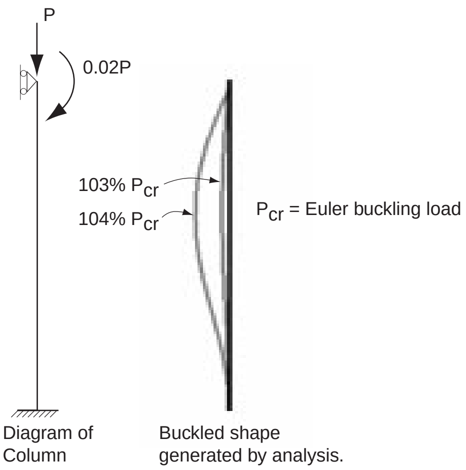

Figure 7. Euler buckling of a column, fixed at the base and pinned at the top. The deformed shapes show the position of the column at 103% and 104% of the Euler Buckling load

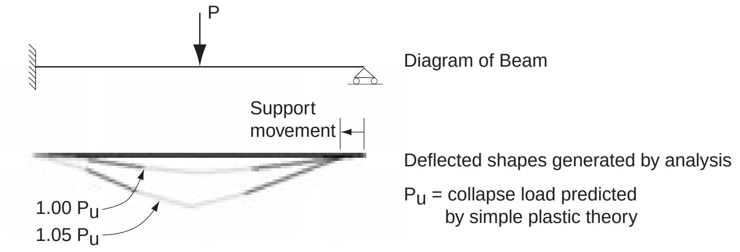

Figure 8. Beam structure with a fixed support at the left end and vertical movement restrained at the right end. The inelastic structure stabilises at 105% of the collapse load because the span shortens by 5%. (note that the displacements are not magnified)

According to simple plastic theory, the collapse load for this beam is 860 kN (193 kips) (Neal 1977). In analysis, when the beam is slowly loaded to the collapse load, the collapse mechanism forms as expected, initiated by yielding at the support, followed by yielding at mid-span. In theory, when both these regions have yielded, the beam becomes a mechanism and cannot resist further load, but the analysis shows that the beam does not collapse, it stabilises, continuing to carry the load. When the analysis increased the load to 5% greater than the collapse load, the beam experiences very large displacements, but does not collapse completely. This result seems at odds with the prediction of simple plastic theory, until the deformed shape of the beam is examined closely. As figure 8 shows, the deformation of the beam causes the right end of the beam to move to the left, effectively shortening the span.. At 5% over the collapse load, the displacement of the right support is 68.6 cm (27 in), which is 5.4% of the total span. This shortening of the span increases the beam's ability to carry load proportionately. Taking the deformation into account, which simple plastic theory normally does not do, the results are quite sensible.

6. POTENTIAL IMPACT

The primary strength of interactive non-linear time-history analysis is that it is the analysis method which most closely mimics real life: reality is an interactive, non-linear, time-history. It's greatest potential is in giving a truer picture of the non-linear behaviour of structures. In the engineering profession, this type of analysis can give designers deeper understanding of a structural concept, although it will certainly not replace static analysis and modal dynamic analysis, which are efficient and effective design tools for many tasks. The particle system approach has the potential to make non-linear time-history analysis a more commonplace, mainstream method which can join more conventional design tools in the practising engineer's toolbox.

There are potentially more profound impacts in structural education in engineering and architecture. In engineering, non-linear, dynamic analysis is typically not taught at the undergraduate level, and only sparingly at the graduate level, and is typically not included in architectural education at all. Yet, many important structural phenomena, particularly those related to collapse, are non-linear and dynamic. Interactive non-linear analysis has the potential to enable structural education emphasise understanding structural behaviour, bring deeper meaning to the study of analysis procedures and building code requirements.

7. EFFICIENCY ISSUES

For many problems, non-linear time history analysis is monumentally inefficient. In any problem that can be solved by static linear theory, such as the deflection of a stiff elastic beam under load, the computer essentially needs to perform one solution of the equilibrium equations to determine the structural response. Using non-linear time history, the solution to such a problem could take three to four seconds of simulation time, with a time step in the range of 0.1 to 0.5 milliseconds, performing three solutions of the equations per time step. Determining the response, then, requires solving the system of equations in the range of 18000 to 120000 times, compared to one solution for a linear static analysis.

This gross inefficiency is certainly one of the reasons that the particle system approach was not adopted in the early days of computer-based structural analysis, because computers were weak and processing cycles were scarce. The processing demands of the method made it completely impractical. The rapid growth of computer power now creates a radically different environment. Common computers are able to perform a complete static analysis in fraction of second for realistic structural models. As computing power continues to grow exponentially, computational efficiency will become a decreasingly important factor in choosing an analysis strategy.

8. FUTURE WORK

The key areas for future development are expanding to three dimensions and incorporating more sophisticated elements. For the truss element, the expansion to three dimensions is quite simple, since the element neglects rotations of the nodes. Rendering is also simple in three dimensions, since the system uses the OpenGL rendering library, which is already three dimensional. The key area for development is in the three dimensional beam element, which becomes much more complex since it must account for the three-dimensional rotation of the nodes.

Another area for development are the interactivity controls. Grabbing structures with the mouse is easy to understand, but offers very little control. Sliders, dials, and haptic force-feedback devices offer opportunities for more controlled interaction.

9. SUMMARY

Many computer games now incorporate real time simulation of physics in order to achieve visual realism. As computational speed and capacity continue to increase, it is reasonable to expect that the physical simulations in computer games will continue to become more sophisticated, and not far in the future, computer games will probably incorporate advanced engineering finite element methods simply for visual realism (e.g. O'Brien and Hodgins 1999). As the gaming industry borrows methods from engineers, it is reasonable for engineers to consider what they can borrow from the gaming industry. One of the lessons to learn from games is the importance of interactivity, and the ability to perform real-time, non-linear physical simulation. Such analysis will clearly not replace other methods used in structural engineering, but has the potential to complement them, adding an important tool to the structural engineering repertoire. The potential impact is particularly important in structural education, which currently places little emphasis on the non-linear and dynamic phenomena that commonly characterise structural failure.

10. REFERENCES

Blum, Mike, U. Thumrugati, 1997, "Using Dynamics in Disney's Production Environment", presentation slides for Siggraph '97 Course *Physically Based Modeling: Principles and Practice. http://www.cs.cmu.edu/afs/cs/user/baraff/www/sigcourse/slidesg.pdf*

Cundall, Peter A., R.D. Hart, 1992, "Numerical Modelling of Discontinua", *Engineering Computations*, vol. 9, p. 101-113.

Hecker, Chris, "Physics, The Next Frontier", 1996, *Game Developer,* October/November, p. 12-20.

Neal, B. G., 1977, *The Plastic Methods of Structural Analysis*, Chapman and Hall, London.

O'Brien, James F., J. K. Hodgins, 1999, “Graphical Modeling and Animation of Brittle Fracture”, *Proceedings of the ACM Siggraph Conference on Computer Graphics*, p. 137-146.

Powell, Graham H., 1973, *DRAIN-2D User's Guide*, UCB/EERC-73/22: Earthquake Engineering Research Center, University of California, Berkeley.

Witkin, Andrew, D. Baraff, 1997a, "Differential Equation Basics" lecture notes for Siggraph 1997 Course *Physically Based Modeling: Principles and Practice* *http://www.cs.cmu.edu/afs/cs/user/baraff/www/sigcourse/notesb.pdf*

Witkin, Andrew, D. Baraff, 1997b, "Particle System Dynamics" lecture notes for Siggraph 1997 Course *Physically Based Modeling: Principles and Practice* *http://www.cs.cmu.edu/afs/cs/user/baraff/www/sigcourse/notesc.pdf*